YASKAWA

System 300S+

CPU | 315-4NE23 | Manual

HB140 | CPU | 315-4NE23 | en | 18-02

SPEED7 CPU 315SN

YASKAWA Europe GmbH
Hauptstraße 185
65760 Eschborn
Germany
Tel.: +49 6196 569-300
Fax: +49 6196 569-398
Email: info@yaskawa.eu.com
Internet: www.yaskawa.eu.com

315-4NE23_000_CPU 315SN,4,EN - © 2018

Table of contents

1 General ... 6
1.1 Copyright © YASKAWA Europe GmbH ... 6
1.2 About this manual ... 7
1.3 Safety information ... 8
2 Basics ... 9
2.1 Safety information for users ... 9
2.2 Operating structure of a CPU ... 10
2.2.1 General ... 10
2.2.2 Applications ... 10
2.2.3 Operands ... 10
2.3 CPU 315-4NE23 ... 12
2.4 General data ... 14
2.4.1 Use in difficult operating conditions ... 15
3 Assembly and installation guidelines ... 16
3.1 Installation dimensions ... 16
3.2 Assembly standard bus ... 17
3.3 Cabling ... 19
3.4 Installation guidelines ... 20
4 Hardware description ... 22
4.1 Properties ... 22
4.2 Structure ... 23
4.2.1 General ... 23
4.2.2 Interfaces ... 23
4.2.3 Memory management ... 25
4.2.4 Slot for storage media ... 25
4.2.5 Battery backup for clock and RAM ... 25
4.2.6 Operating mode switch ... 26
4.2.7 LEDs ... 27
4.3 Technical data ... 29
5 Deployment CPU 315-4NE23 ... 37
5.1 Assembly ... 37
5.2 Start-up behavior ... 37
5.3 Addressing ... 38
5.3.1 Overview ... 38
5.3.2 Addressing Backplane bus I/O devices ... 38
5.4 Hardware configuration - CPU ... 40
5.4.1 Switching CPU type to CPU 318-2AJ00 ... 40
5.5 Hardware configuration - I/O modules ... 41
5.6 Hardware configuration - Ethernet PG/OP channel ... 41
5.7 Hardware configuration - Communication ... 43
5.8 Setting standard CPU parameters ... 43
5.8.1 Parameterization via Siemens CPU ... 43
5.8.2 Parameters CPU ... 43
5.8.3 Parameters for DP ... 46
5.8.4 Parameters for MPI/DP ... 46
5.9 Setting Yaskawa specific CPU parameters ... 47
5.9.1 Proceeding ... 47

5.9.2 Yaskawa specific parameters ... 48
5.10 Project transfer ... 51
5.10.1 Transfer via MPI/PROFIBUS ... 51
5.10.2 Transfer via Ethernet ... 53
5.10.3 Transfer via memory card ... 53
5.11 Accessing the web server ... 54
5.12 Operating modes ... 59
5.12.1 Overview ... 59
5.12.2 Function security ... 61
5.13 Overall reset ... 62
5.14 Firmware update ... 63
5.15 Reset to factory settings ... 65
5.16 Deployment storage media - MMC, MCC ... 66
5.17 Extended know-how protection ... 68
5.18 CMD - auto commands ... 68
5.19 Diagnostic entries ... 70
5.20 Control and monitoring of variables with test functions ... 71
6 Deployment PtP communication ... 73
6.1 Fast introduction ... 73
6.2 Principle of the data transfer ... 74
6.3 Deployment of RS485 interface for PtP ... 75
6.4 Parametrization ... 77
6.4.1 FC/SFC 216 - SER_CFG - Parametrization PtP ... 77
6.5 Communication ... 77
6.5.1 FC/SFC 217 - SER_SND - Send to PtP ... 77
6.5.2 FC/SFC 218 - SER_RCV - Receive from PtP ... 78
6.6 Protocols and procedures ... 78
6.7 Modbus - Function codes ... 82
6.8 Modbus - Example communication ... 86
7 Deployment PROFIBUS communication ... 88
7.1 Overview ... 88
7.2 Fast introduction ... 88
7.3 Hardware configuration - CPU ... 90
7.3.1 Switching CPU type to CPU 318-2AJ00 ... 90
7.4 Deployment as PROFIBUS DP master ... 91
7.5 Deployment as PROFIBUS DP slave ... 92
7.6 PROFIBUS installation guidelines ... 94
7.7 Commissioning and Start-up behavior ... 97
8 Deployment Ethernet communication - productive ... 98
8.1 Basics - Industrial Ethernet in automation ... 98
8.2 Basics - ISO/OSI reference model ... 99
8.3 Basics - Terms ... 101
8.4 Basics - Protocols ... 101
8.5 Basics - IP address and subnet ... 105
8.6 Basics - MAC address and TSAP ... 107
8.7 Fast introduction ... 108
8.8 Commissioning and initialization ... 109
8.9 Hardware configuration - CP 343 ... 110
8.9.1 Time-of-day synchronization ... 111

8.10 Configure connections ... 112
8.10.1 Overview ... 112
8.10.2 Siemens NetPro ... 112
8.10.3 Connection type - S7 ... 115
8.10.4 Connection type - Send/Receive ... 117
8.11 Configure Open Communication ... 126
8.12 NCM diagnostic - Help for error diagnostic ... 128
8.13 Coupling to other systems ... 130
9 WinPLC7 ... 134
9.1 System conception ... 134
9.2 Installation ... 134
9.3 Example project engineering ... 135
9.3.1 Job definition ... 135
9.3.2 Project engineering ... 136
9.3.3 Test the PLC program in the *Simulator* ... 142
9.3.4 Transfer PLC program to CPU and its execution ... 143
10 Configuration with TIA Portal ... 145
10.1 TIA Portal - Work environment ... 145
10.1.1 General ... 145
10.1.2 Work environment of the TIA Portal ... 146
10.2 TIA Portal - Hardware configuration - CPU ... 147
10.3 TIA Portal - Hardware configuration - I/O modules ... 148
10.4 TIA Portal - Hardware configuration - Ethernet PG/OP channel ... 149
10.5 TIA Portal - Setting VIPA specific CPU parameters ... 151
10.6 TIA Portal - Yaskawa-Include library ... 154
10.7 TIA Portal - Project transfer ... 155
Appendix ... 157
A System specific event IDs ... 159
B Integrated blocks ... 207
C SSL partial list ... 211

Copyright © YASKAWA Europe GmbH

1 General

1.1 Copyright © YASKAWA Europe GmbH

All Rights Reserved

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

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

For permission to reproduce or distribute, please contact: YASKAWA Europe GmbH, European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany

Tel.: +49 6196 569 300
Fax.: +49 6196 569 398
Email: info@yaskawa.eu.com
Internet: www.yaskawa.eu.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, YASKAWA Europe 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 representative of YASKAWA Europe GmbH.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of YASKAWA Europe GmbH.

SPEED7 is a registered trademark of YASKAWA Europe 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.

Document support

Contact your local representative of YASKAWA Europe GmbH if you have errors or questions regarding the content of this document. You can reach YASKAWA Europe GmbH via the following contact:

Email: Documentation.HER@yaskawa.eu.com

Technical support

Contact your local representative of YASKAWA Europe GmbH if you encounter problems or have questions regarding the product. If such a location is not available, you can reach the Yaskawa customer service via the following contact:

YASKAWA Europe GmbH,
European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany
Tel.: +49 6196 569 500 (hotline)
Email: support@yaskawa.eu.com

1.2 About this manual

Objective and contents

This manual describes the SPEED7 CPU 315-4NE23 of the CPU from Yaskawa. It contains a description of the construction, project implementation and usage.

Product	Order no.	as of state:			
		CPU-HW	CPU-FW	DPM-FW	CP-FW
CPU 315SN	315-4NE23	01	V3.7.5	V3.3.5	V3.1.1

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

2.1 Safety information for users

Handling of electrostatic sensitive modules

The 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 Operating structure of a CPU

2.2.1 General

The CPU contains a standard processor with internal program memory. In combination with the integrated SPEED7 technology the unit provides a powerful solution for process automation applications within the System 300S family. A CPU supports the following modes of operation:

- cyclic operation
- timer processing
- alarm controlled operation
- priority based processing

Cyclic processing

Cyclicprocessing represents the major portion of all the processes that are executed in the CPU. Identical sequences of operations are repeated in a never-ending cycle.

Timer processing

Where a process requires control signals at constant intervals you can initiate certain operations based upon a **timer**, e.g. not critical monitoring functions at one-second intervals.

Alarm controlled processing

If a process signal requires a quick response you would allocate this signal to an **alarm controlled** procedure. An alarm can activate a procedure in your program.

Priority based processing

The above processes are handled by the CPU in accordance with their **priority**. Since a timer or an alarm event requires a quick reaction, the CPU will interrupt the cyclic processing when these high-priority events occur to react to the event. Cyclic processing will resume, once the reaction has been processed. This means that cyclic processing has the lowest priority.

2.2.2 Applications

The program that is present in every CPU is divided as follows:

- System routine
- User application

System routine

The system routine organizes all those functions and procedures of the CPU that are not related to a specific control application.

User application

This consists of all the functions that are required for the processing of a specific control application. The operating modules provide the interfaces to the system routines.

2.2.3 Operands

The following series of operands is available for programming the CPU:

- Process image and periphery
- Bit memory
- Timers and counters
- Data blocks

Process image and periphery

The user application can quickly access the process image of the inputs and outputs PIO/PII. You may manipulate the following types of data:

- individual Bits
- Bytes
- Words
- Double words

You may also gain direct access to peripheral modules via the bus from user application. The following types of data are available:

- Bytes
- Words
- Blocks

Bit Memory

The bit memory is an area of memory that is accessible by means of certain operations. Bit memory is intended to store frequently used working data.

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

Timers and counters

In your program you may load cells of the timer with a value between 10ms and 9990s. As soon as the user application executes a start-operation, the value of this timer is decremented by the interval that you have specified until it reaches zero.

You may load counter cells with an initial value (max. 999) and increment or decrement these when required.

Data Blocks

A data block contains constants or variables in the form of bytes, words or double words. You may always access the current data block by means of operands.

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

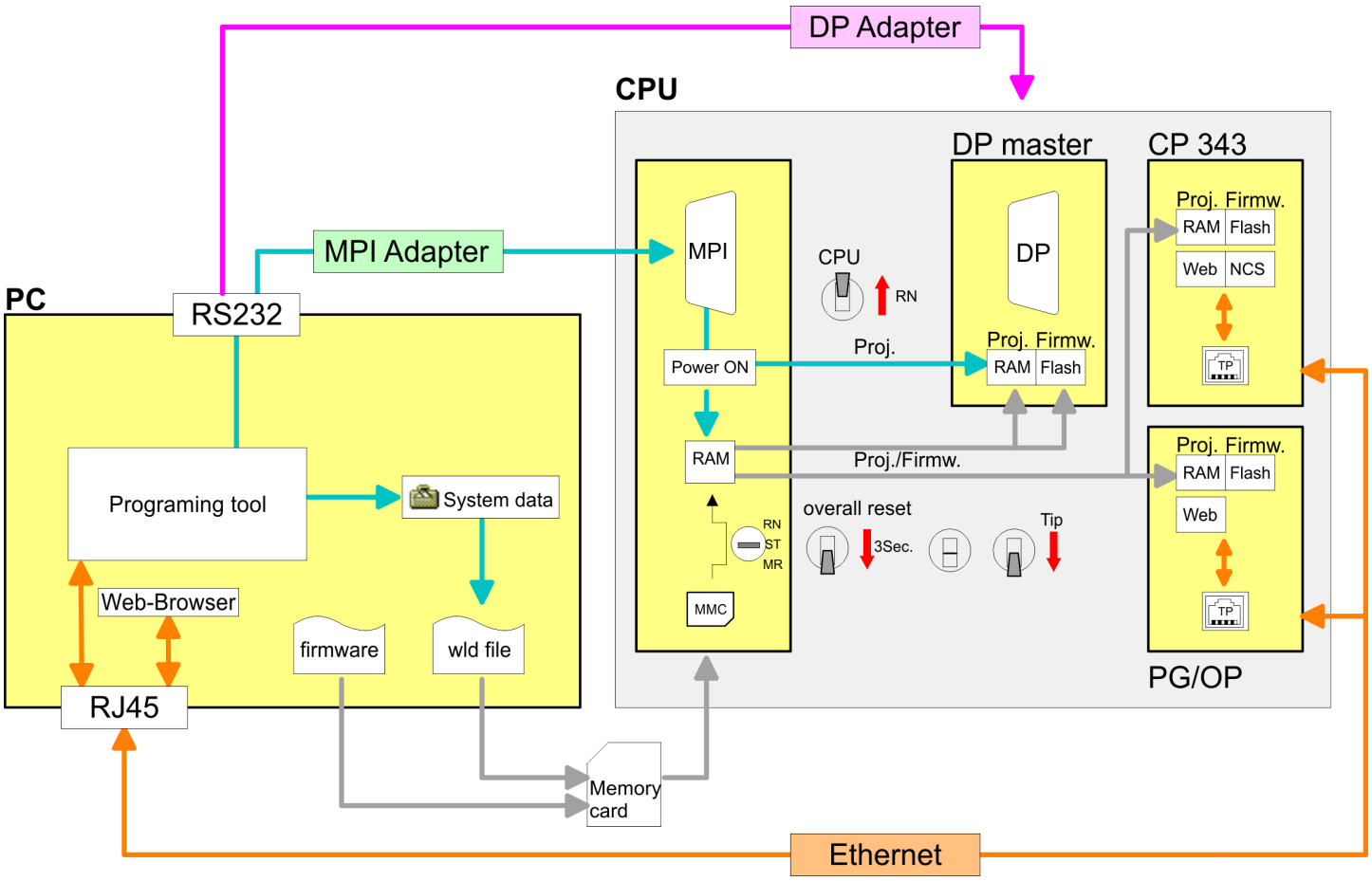
2.3 CPU 315-4NE23

Overview

The CPU 315-4NE23 bases upon the SPEED7 technology. This supports the CPU at programming and communication by means of co-processors that causes a power improvement for highest needs.

- The CPU is programmed in STEP®7 from Siemens. For this you may use the SIMATIC Manager or TIA Portal from Siemens. Here the instruction set of the S7-400 from Siemens is used.
- Modules and CPUs of the System 300S from Yaskawa and Siemens may be used at the bus as a mixed configuration.
- The user application is stored in the battery buffered RAM or on an additionally pluggable storage module.
- The CPU is configured as CPU 317-2DP (6ES7 317-2AK14 V3.3) from Siemens.

Access



*Please always use the **CPU 317-2DP (6ES7 317-2AK14 V3.3)** from Siemens of the hardware catalog to configure this CPU from Yaskawa. For the project engineering, a thorough knowledge of the Siemens SIMATIC Manager and the hardware configurator from Siemens is required!*

Memory

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 4MB
- Code memory (50% of the work memory)

- Data memory (50% of the work memory)
- Work memory 1MB
  - There is the possibility to extend the work memory to its maximum printed capacity 4MB by means of a memory extension card.

Integrated PROFIBUS DP master/slave respectively PtP functionality

The CPU has a PROFIBUS/PtP interface with a fix pinout. After an overall reset the interface is deactivated. By appropriate configuration, the following functions for this interface may be enabled:

- PROFIBUS DP master operation: Configuration via PROFIBUS sub module with *'Operation mode'* master in the hardware configuration.
- PROFIBUS DP slave operation: Configuration via PROFIBUS sub module with *'Operation mode'* slave in the hardware configuration.
- PtP functionality: Configuration as virtual PROFIBUS master system by including the Yaskawa SPEEDBUS.GSD.

Integrated CP 343

The integrated CP 343 offers you a communication processor. This serves PG/OP channels and by Siemens NetPro respectively by user program configurable productive connections.

Integrated Ethernet PG/OP channel

The CPU has an Ethernet interface for PG/OP communication. After assigning IP address parameters with your configuration tool, via the "PLC" functions you may directly access the Ethernet PG/OP channel and program res. remote control your CPU. You may also access the CPU with a visualization software via these connections.

Operation Security

- Wiring by means of spring pressure connections (CageClamps) at the front connector
- Core cross-section 0.08...2.5mm^2
- Total isolation of the wiring at module change
- Potential separation of all modules to the backplane bus

Dimensions/ Weight

Dimensions of the basic enclosure:

- 2tier width: (WxHxD) in mm: 80x125x120

Integrated power supply

The CPU comes with an integrated power supply. The power supply is to be supplied with DC 24V. By means of the supply voltage, the internal electronic is supplied as well as the connected modules via backplane bus. The power supply is protected against inverse polarity and overcurrent.

2.4 General data

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

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

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

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.

2.4.1 Use in difficult operating conditions

Without additional protective measures, the products must not be used in locations with difficult operating conditions; e.g. due to:

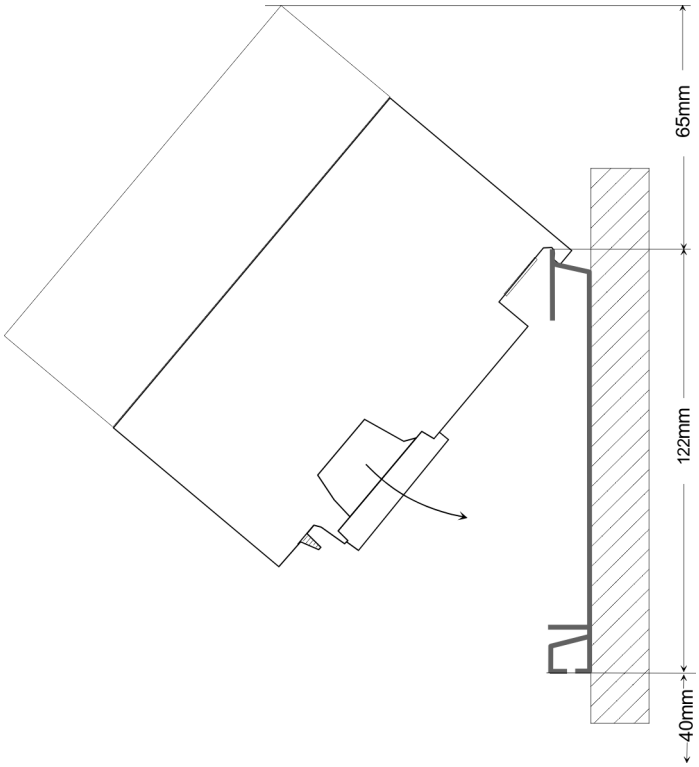
- *dust generation*
- *chemically active substances (corrosive vapors or gases)*
- *strong electric or magnetic fields*

3 Assembly and installation guidelines

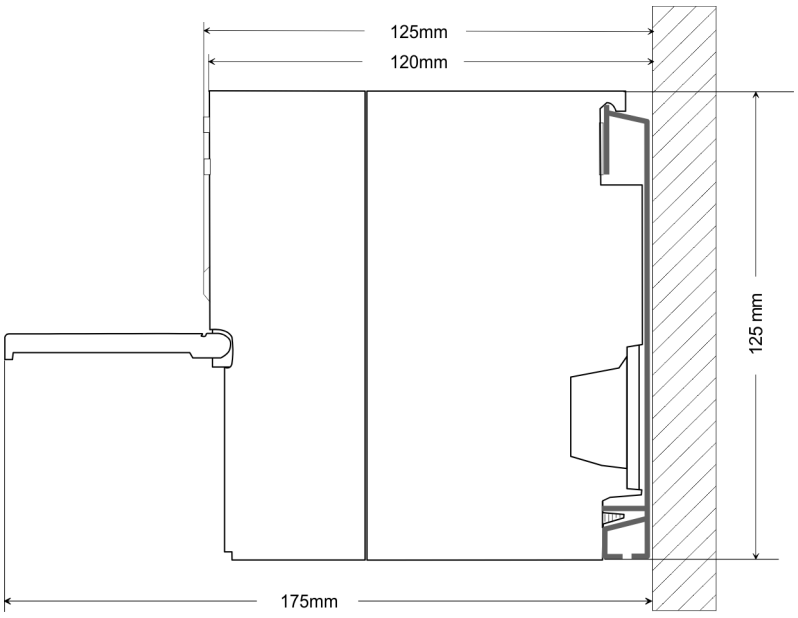
3.1 Installation dimensions

Dimensions Basic enclosure

2tier width (WxHxD) in mm: 80 x 125 x 120



Installation dimensions



3.2 Assembly standard bus

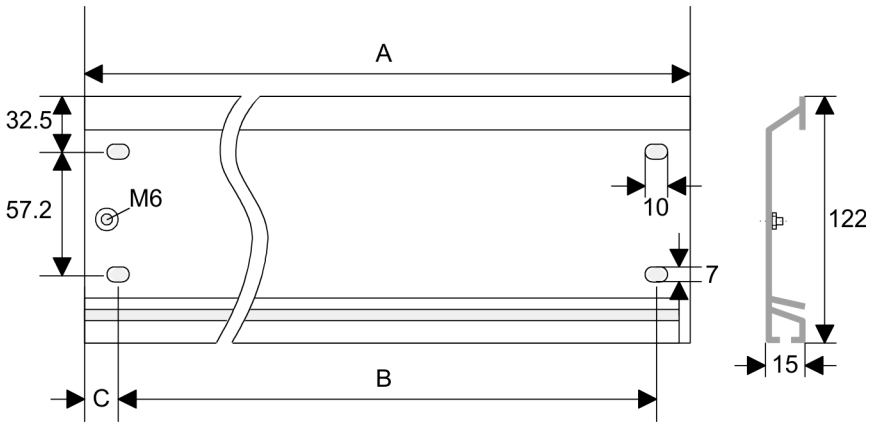
General

The single modules are directly installed on a profile rail and connected via the backplane bus connector. Before installing the modules you have to clip the backplane bus connector to the module from the backside. The backplane bus connector is delivered together with the peripheral modules.

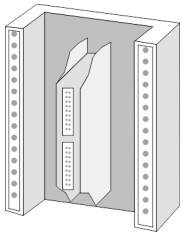
Profile rail

Order number	A	B	C
390-1AB60	160	140	10
390-1AE80	482	466	8.3
390-1AF30	530	500	15
390-1AJ30	830	800	15
390-9BC00*	2000	Drillings only left	15
*) Unit pack: 10 pieces			

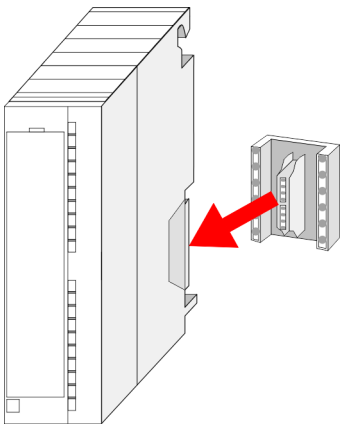
Measures in mm



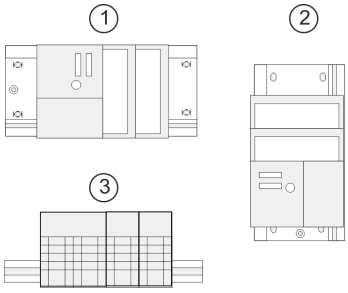
Bus connector



For the communication between the modules the System 300S uses a backplane bus connector. Backplane bus connectors are included in the delivering of the peripheral modules and are clipped at the module from the backside before installing it to the profile rail.



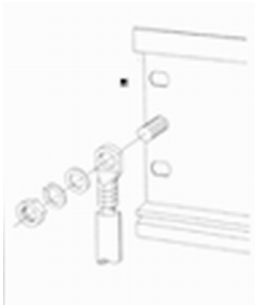
Assembly possibilities



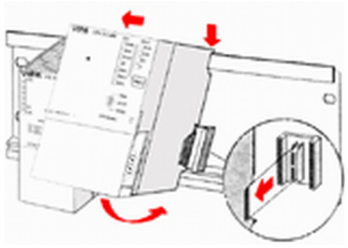
Please regard the allowed environment temperatures:

1 horizontal assembly: from 0 to 60°C
2 vertical assembly: from 0 to 50°C
3 lying assembly: from 0 to 55°C

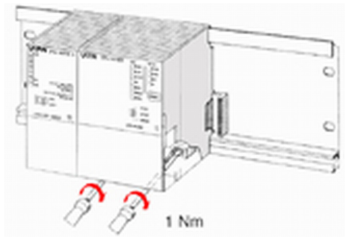
Approach



1. Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail.
2. If the background is a grounded metal or device plate, please look for a low-impedance connection between profile rail and background.
3. Connect the profile rail with the protected earth conductor. For this purpose there is a bolt with M6-thread.
4. The minimum cross-section of the cable to the protected earth conductor has to be 10mm^2.



5. Stick the power supply to the profile rail and pull it to the left side to the grounding bolt of the profile rail.
6. Fix the power supply by screwing.
7. Take a backplane bus connector and click it at the CPU from the backside like shown in the picture.
8. Stick the CPU to the profile rail right from the power supply and pull it to the power supply.



9. Click the CPU downwards and bolt it like shown.
10. Repeat this procedure with the peripheral modules, by clicking a backplane bus connector, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus connector of the last module and bolt it.

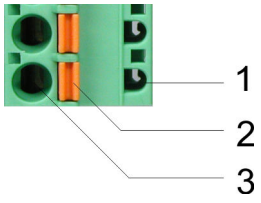
3.3 Cabling

CAUTION!
- The power supplies must be released before installation and repair tasks, i.e. before handling with the power supply or with the cabling you must disconnect current/voltage (pull plug, at fixed connection switch off the concerning fuse)!
- Installation and modifications only by properly trained personnel!

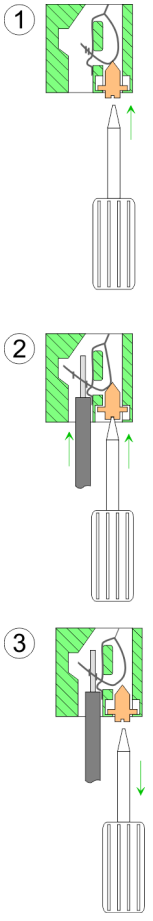
CageClamp technology (green)

For the cabling of power supply of a CPU, a green plug with CageClamp technology is deployed. The connection clamp is realized as plug that may be clipped off carefully if it is still cabled.

Here wires with a cross-section of 0.08mm² to 2.5mm² may be connected. You can use flexible wires without end case as well as stiff wires.



1 Test point for 2mm test tip
2 Locking (orange) for screwdriver
3 Round opening for wires



The picture on the left side shows the cabling step by step from top view.

1. For cabling you push the locking vertical to the inside with a suiting screwdriver and hold the screwdriver in this position.

2. Insert the de-isolated wire into the round opening. You may use wires with a cross-section from 0.08mm² to 2.5mm²

3. By removing the screwdriver the wire is connected safely with the plug connector via a spring.

3.4 Installation guidelines

General

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

What does EMC mean?

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

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

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

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

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

There are:

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

Basic rules for EMC

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

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

- Proof the correct fixing of the lead isolation.
  - Data lines must be shielded.
  - Analog lines must be shielded. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Cables for frequency inverters, servo and stepper motors must be shielded.
  - 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.

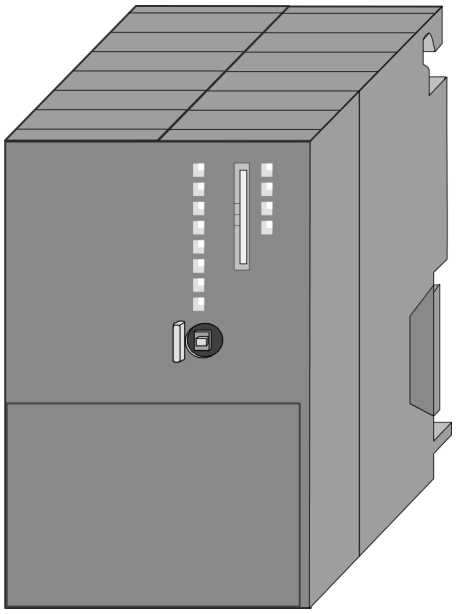
Remedy: Potential compensation line

4 Hardware description

4.1 Properties

CPU 315-4NE23

- SPEED7 technology integrated
- 1MB work memory integrated (512kByte code, 512kByte data)
- Work memory expandable to max. 4MB (2MB code, 2MB data)
- 4MB load memory
- PROFIBUS DP master integrated (DP-V0, DP-V1)
- CP 343 communication processor integrated
  ↳ *Chap. 4.3 'Technical data' page 29*
  - Productive connections via Siemens NetPro
  - Productive connections via user program
  - PG/OP connections
- RS485 interface configurable for PROFIBUS DP master respectively PtP communication
- Ethernet PG/OP interface integrated
- MPI interface
- Slot for external memory cards (lockable)
- Status LEDs for operating state and diagnostics
- Real-time clock battery buffered
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte



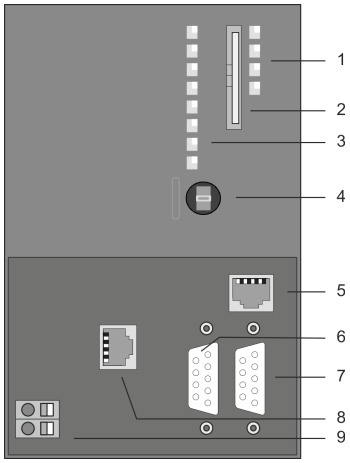
Ordering data

Type	Order number	Description
CPU 315SN	315-4NE23	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, PROFIBUS DP master, CP 343

4.2 Structure

4.2.1 General

CPU 315-4NE23

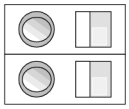


1 LEDs of the integrated PROFIBUS DP master
2 Storage media slot (lockable)
3 LEDs of the CPU part
4 Operating mode switch CPU
5 Twisted pair interface for Ethernet PG/OP channel
6 MPI interface
7 PROFIBUS DP/PtP interface
8 Twisted Pair interface for CP 343
9 Slot for DC 24V power supply

The components 5 - 9 are under the front flap!

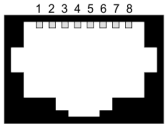
4.2.2 Interfaces

X1



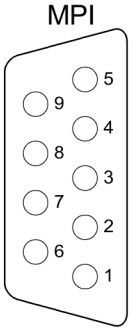
+ ① + DC 24 V
- ② 0 V

X5



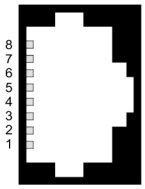
① Transmit +
② Transmit -
③ Receive +
④ n. c.
⑤ n. c.
⑥ Receive -
⑦ n. c.
⑧ n. c.

X2
MPI



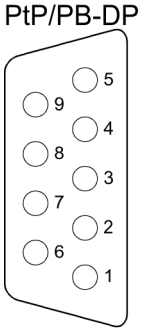
① n.c.
② M24V
③ RxD/TxD-P (line B)
④ RTS
⑤ M5V
⑥ P5V
⑦ P24V
⑧ RxD/TxD-N (line A)
⑨ n.c.

X8



① Transmit +
② Transmit -
③ Receive +
④ n. c.
⑤ n. c.
⑥ Receive -
⑦ n. c.
⑧ n. c.

X3
PtP/PB-DP



① shield
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.

X1: Power supply

The CPU has an integrated power supply:

- The power supply has to be provided with DC 24V. For this serves the double DC 24V slot, that is underneath the flap.
- Via the power supply not only the internal electronic is provided with voltage, but by means of the backplane bus also the connected modules.
- The power supply is protected against polarity inversion and overcurrent.
- The internal electronic is galvanically connected with the supply voltage.

X2: MPI interface

9pin SubD jack:

- The MPI interface serves for the connection between programming unit and CPU.
- By means of this the project engineering and programming happens.
- MPI serves for communication between several CPUs or between HMIs and CPU.
- Standard setting is MPI Address 2.

X5: Ethernet PG/OP channel

8pin RJ45 jack:

- The RJ45 jack serves the interface to the Ethernet PG/OP channel.
- This interface allows you to program res. remote control your CPU, to access the internal web site or to connect a visualization.
- Configurable connections are not possible.
- For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this.

X3: PROFIBUS/PtP interface with configurable functionality

9pin SubD jack:

The CPU has a PROFIBUS/PtP interface with a fix pinout. After an overall reset the interface is deactivated. By appropriate configuration, the following functions for this interface may be enabled:

- PROFIBUS DP master operation
  - Configuration via PROFIBUS sub module X1 (MPI/DP) with *'Operation mode'* master in the hardware configuration.
- PROFIBUS DP slave operation
  - Configuration via PROFIBUS sub module X1 (MPI/DP) with *'Operation mode'* slave in the hardware configuration.
- 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.
  - Here the following protocols are supported: ASCII, STX/ETX, 3964R, USS and Modbus-Master (ASCII, RTU).
  - The activation of the PtP functionality happens by embedding the SPEEDBUS.GSD from VIPA in the hardware catalog. After the installation the CPU may be configured in a PROFIBUS master system and here the interface may be switched to PtP communication.

Communication processor CP 343 X8

8pin RJ45 jack:

- Communication processor CP 343 for connection to Ethernet
- Productive connections via Siemens NetPro
- Productive connections via user program
- PG/OP connections

Number of connections ↬ Chap. 4.3 'Technical data' page 29

4.2.3 Memory management

Memory

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 4MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 1MB
  - There is the possibility to extend the work memory to its maximum printed capacity 4MB by means of a memory extension card.

4.2.4 Slot for storage media

At this slot the following storage media can be plugged:

- SD respectively MCC (**M**ulti**m**edia **c**ard)
  - External memory card for programs and firmware.
- MCC - **M**emory **c**onfiguration **c**ard
  - External memory card (MMC) for programs and firmware with the possibility to unlock additional work memory.
  - The additional memory can be purchased separately. ↬ *Chap. 5.16 'Deployment storage media - MMC, MCC' page 66*
  - To activate the corresponding card is to be installed and an *Overall reset* is to be established. ↬ *Chap. 5.13 'Overall reset' page 62*

4.2.5 Battery backup for clock and RAM

A rechargeable battery is installed on every CPU to safeguard the contents of the RAM when power is removed. This battery is also used to buffer the internal clock. The rechargeable battery is maintained by a charging circuit that receives its power from the internal power supply and that maintain the clock and RAM for a max. period of 30 days.

- *Please connect the CPU at least for 24 hours to the power supply, so that the internal accumulator/battery is loaded accordingly.*
- *Please note that in case of repeated discharge cycles (charging/ buffering) can reduce the buffer time continuously. Only after a charging time of 24 hours there is a buffer for max. 30 days.*

CAUTION!

- After a power reset and with an empty battery the CPU starts with a BAT error and executes an overall reset. The loading procedure is not influenced by the BAT error.
- The BAT error can be deleted again, if once during power cycle the time between switching on and off the power supply is at least 30sec. and the battery is fully loaded. Otherwise with a short power cycle the BAT error still exists and an overall reset is executed.

4.2.6 Operating mode switch

- 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 (Memory Reset), you request an overall reset with following load from memory card, if a project there exists.

4.2.7 LEDs

LEDs CPU

RN (RUN) green	ST (STOP) yellow	SF (SFAIL) red	FC (FRCE) yellow	MC (MMC) yellow	Meaning
Boot-up after PowerON - as soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.					
■	◪ 10Hz	■	■	■	Firmware is loaded.
■	■	■	■	■	Initialization: Phase 1
■	■	■	■	□	Initialization: Phase 2
■	■	■	□	□	Initialization: Phase 3
□	■	■	□	□	Initialization: Phase 4
Operation					
□	■	X	X	X	CPU is in STOP state.
◪ 2Hz	■	X	X	X	CPU is in start-up state. As long as the OB 100 is processed, the RUN LED blinks for at least 3s.
■	□	□	X	X	CPU is in state RUN without error.
X	X	■	X	X	There is a system fault. More information can be found in the diagnostics buffer of the CPU.
X	X	X	■	X	Variables are forced.
X	X	X	X	■	Accessing the memory card
X	◪ 10Hz	□	□	□	Configuration is loaded.
Overall reset					
□	◪ 2Hz	X	X	X	Overall reset is requested
□	◪ 10Hz	X	X	X	Overall reset is executed.
Factory reset					
■	■	□	□	□	Reset to factory setting is executed.
□	■	■	■	■	Reset to factory setting finished without error
Firmware update					
□	■	◪ 2Hz	◪ 2Hz	■	The alternate blinking indicates that there is new firmware on the memory card.
□	□	◪ 2Hz	◪ 2Hz	■	The alternate blinking indicates that a firmware update is executed.
□	■	■	■	■	Firmware update finished without error.
□	◪ 10Hz	◪ 10Hz	◪ 10Hz	◪ 10Hz	Error during Firmware update.
not relevant: X					

Ethernet PG/OP channel

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

LEDs PROFIBUS/PtP interface X3

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

Master operation

RN (RUN) green	ER (ERR) red	DE green	IF red	Meaning
□	□	□	□	Master has no project, this means the interface is deactivated respectively PtP is active.
■	□	□	□	Master has bus parameters and is in RUN without slaves.
■	□	◪ 2Hz	□	Master is in "clear" state (safety state). The inputs of the slaves may be read. The outputs are disabled.
■	□	■	□	Master is in "operate" state, this means data exchange between master and slaves. The outputs may be accessed.
■	■	■	□	CPU is in RUN state, at least 1 slave is missing.
■	■	◪ 2Hz	□	CPU is in STOP, at least 1 slave is missing.
□	□	□	■	Initialization error at faulty parametrization.
□	■	□	■	Wait state for start command from CPU.

Slave operation

RN (RUN) green	ER (ERR) red	DE green	IF red	Meaning
□	□	□	□	Slave has no configuration respectively PtP is active.
◪ 2Hz	□	□	□	Slave is without master.

RN (RUN) green	ER (ERR) red	DE green	IF red	Meaning
2Hz		2Hz		Alternate blinking at configuration faults.
				Slave exchanges data with the master.

4.3 Technical data

Order no.	315-4NE23
Type	CPU 315SN
SPEED-Bus	-
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)	270 mA
Current consumption (rated value)	1 A
Inrush current	5 A
I^2t	0.5 A^2s
Max. current drain at backplane bus	2.5 A
Max. current drain load supply	-
Power loss	8.5 W
Load and working memory	
Load memory, integrated	4 MB
Load memory, maximum	4 MB
Work memory, integrated	1 MB
Work memory, maximal	4 MB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
Hardware configuration	
Racks, max.	4
Modules per rack, max.	8 in multiple-, 32 in a single-rack configuration
Number of integrated DP master	1
Number of DP master via CP	4
Operable function modules	8
Operable communication modules PtP	8
Operable communication modules LAN	8

Order no.	315-4NE23
Status information, alarms, diagnostics	
Status display	yes
Interrupts	no
Process alarm	no
Diagnostic interrupt	no
Diagnostic functions	no
Diagnostics information read-out	possible
Supply voltage display	green LED
Group error display	red SF LED
Channel error display	none
Command processing times	
Bit instructions, min.	0.01 µs
Word instruction, min.	0.01 µs
Double integer arithmetic, min.	0.01 µs
Floating-point arithmetic, min.	0.06 µs
Timers/Counters and their retentive characteristics	
Number of S7 counters	512
S7 counter remanence	adjustable 0 up to 512
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	adjustable 0 up to 512
S7 times remanence adjustable	not retentive
Data range and retentive characteristic	
Number of flags	8192 Byte
Bit memories retentive characteristic adjustable	adjustable 0 up to 8192
Bit memories retentive characteristic preset	MB0 .. MB15
Number of data blocks	4095
Max. data blocks size	64 KB
Number range DBs	1 ... 4095
Max. local data size per execution level	1024 Byte
Max. local data size per block	1024 Byte
Blocks	
Number of OBs	23
Maximum OB size	64 KB
Total number DBs, FBs, FCs	-
Number of FBs	2048

Order no.	315-4NE23
Maximum FB size	64 KB
Number range FBs	0 ... 2047
Number of FCs	2048
Maximum FC size	64 KB
Number range FCs	0 ... 2047
Maximum nesting depth per priority class	8
Maximum nesting depth additional within an error OB	4
Time	
Real-time clock buffered	✓
Clock buffered period (min.)	6 w
Type of buffering	Vanadium Rechargeable Lithium Battery
Load time for 50% buffering period	20 h
Load time for 100% buffering period	48 h
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	Slave
Address areas (I/O)	
Input I/O address area	8192 Byte
Output I/O address area	8192 Byte
Process image adjustable	✓
Input process image preset	256 Byte
Output process image preset	256 Byte
Input process image maximal	2048 Byte
Output process image maximal	2048 Byte
Digital inputs	65536
Digital outputs	65536
Digital inputs central	1024
Digital outputs central	1024
Integrated digital inputs	-
Integrated digital outputs	-
Analog inputs	4096
Analog outputs	4096
Analog inputs, central	256
Analog outputs, central	256

Order no.	315-4NE23
Integrated analog inputs	-
Integrated analog outputs	-
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
Functionality Sub-D interfaces	
Type	X2
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	-
DP slave	-
Point-to-point interface	-
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Type	X3
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	-
MP²I (MPI/RS232)	-
DP master	yes
DP slave	yes
Point-to-point interface	✓

Order no.	315-4NE23
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Functionality MPI	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓
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	-
Activation/deactivation of DP slaves	✓
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Number of DP slaves, max.	124
Address range inputs, max.	8 KB
Address range outputs, max.	8 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	✓

Order no.	315-4NE23
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
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	-
RS485 interface	✓
Connector	Sub-D, 9-pin, female
Transmission speed, min.	150 bit/s
Transmission speed, max.	115.5 kbit/s
Cable length, max.	500 m
Point-to-point protocol	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	✓
Modbus master protocol	✓
Modbus slave protocol	-
Special protocols	-
Functionality RJ45 interfaces	
Type	X5
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓

Order no.	315-4NE23
Number of connections, max.	4
Productive connections	-
Fieldbus	-
Type	X8
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	32
Productive connections	✓
Fieldbus	-
Ethernet communication CP	
Number of configurable connections, max.	8
Number of productive connections by Siemens NetPro, max.	8
S7 connections	BSEND, BRCV, GET, PUT, Connection of active and passive data handling
User data per S7 connection, max.	32 KB
TCP-connections	SEND, RECEIVE, FETCH PASSIV, WRITE PASSIV, Connection of active and passive data handling
User data per TCP connection, max.	64 KB
ISO-connections	SEND, RECEIVE, FETCH PASSIV, WRITE PASSIV, Connection of active and passive data handling
User data per ISO connection, max.	8 KB
ISO on TCP connections (RFC 1006)	SEND, RECEIVE, FETCH PASSIV, WRITE PASSIV, Connection of active and passive data handling
User data per ISO on TCP connection, max.	32 KB
UDP-connections	SEND and RECEIVE
User data per UDP connection, max.	2 KB
UDP-multicast-connections	SEND and RECEIVE (max. 8 Multicast groups)
UDP-broadcast-connections	SEND
Ethernet open communication	
Number of connections, max.	8
ISO on TCP connections (RFC 1006)	TSEND, TRCV, TCON, TDISCON
User data per ISO on TCP connection, max.	8 KB
TCP-Connections native	TSEND, TRCV, TCON, TDISCON
User data per native TCP connection, max.	8 KB

Order no.	315-4NE23
User data per ad hoc TCP connection, max.	1460 Byte
UDP-connections	TUSEND, TURCV
User data per UDP connection, max.	1472 Byte
Housing	
Material	PPE
Mounting	Rail System 300
Mechanical data	
Dimensions (WxHxD)	80 mm x 125 mm x 120 mm
Net weight	430 g
Weight including accessories	-
Gross weight	-
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	yes
KC certification	yes

5 Deployment CPU 315-4NE23

5.1 Assembly

Information about assembly and cabling: Chap. 3 'Assembly and installation guidelines' page 16

5.2 Start-up behavior

Turn on power supply

After the power supply has been switched on, the CPU changes to the operating mode the operating mode lever shows.

Default boot procedure, as delivered

When the CPU is delivered it has been reset. After a STOP→RUN transition the CPU switches to RUN without program.

Boot procedure with valid configuration in the CPU

The CPU switches to RUN with the program stored in the battery buffered RAM.

Boot procedure with empty battery

- The accumulator/battery is automatically loaded via the integrated power supply and guarantees a buffer for max. 30 days. If this time is exceeded, the battery may be totally discharged. This means that the battery buffered RAM is deleted.
- In this state, the CPU executes an overall reset. If a memory card is plugged, program code and data blocks are transferred from the memory card into the work memory of the CPU. If no memory card is plugged, the CPU transfers permanent stored "protected" blocks into the work memory if available.
- Depending on the position of the operating mode switch, the CPU switches to RUN, if OB 81 exists, res. remains in STOP. This event is stored in the diagnostic buffer as: "Start overall reset automatically (unbuffered PowerON)".

CAUTION!
After a power reset and with an empty battery the CPU starts with a BAT error and executes an overall reset. The BAT error can be deleted again, if once during power cycle the time between switching on and off the power supply is at least 30sec. and the battery is fully loaded. Otherwise with a short power cycle the BAT error still exists and an overall reset is executed.

5.3 Addressing

5.3.1 Overview

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. At the start-up of the CPU, this assigns automatically peripheral addresses for digital in-/output modules starting with 0 and ascending depending on the slot location.

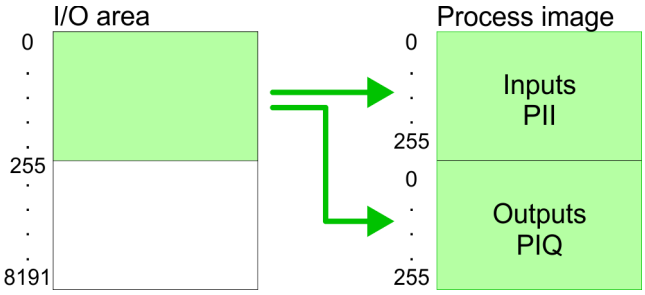
If no hardware project engineering is available, the CPU stores at the addressing analog modules to even addresses starting with 256.

5.3.2 Addressing Backplane bus I/O devices

The CPU 315-4NE23 provides an I/O area (address 0 ... 8191) and a process image of the In- and Outputs (each address 0 ... 255).

The process image this divided into two parts:

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



The process image is updated automatically when a cycle has been completed.

Max. number of pluggable modules

Maximally 8 modules per row may be configured by the CPU 315-4NE23.

For the project engineering of more than 8 modules you may use line interface connections. For this you set in the hardware configurator the module IM 360 from the hardware catalog to slot 3 of your 1. profile rail. Now you may extend your system with up to 3 profile rails by starting each with an IM 361 from Siemens at slot 3. Considering the max total current with the CPU 315-4NE23 from VIPA up to 32 modules may be arranged in a row. Here the installation of the line connections IM 360/361 from Siemens is not required.

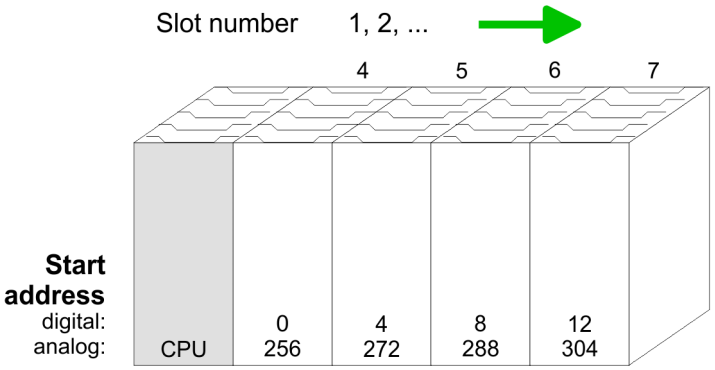
Define addresses by hardware configuration

You may access the modules with read res. 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

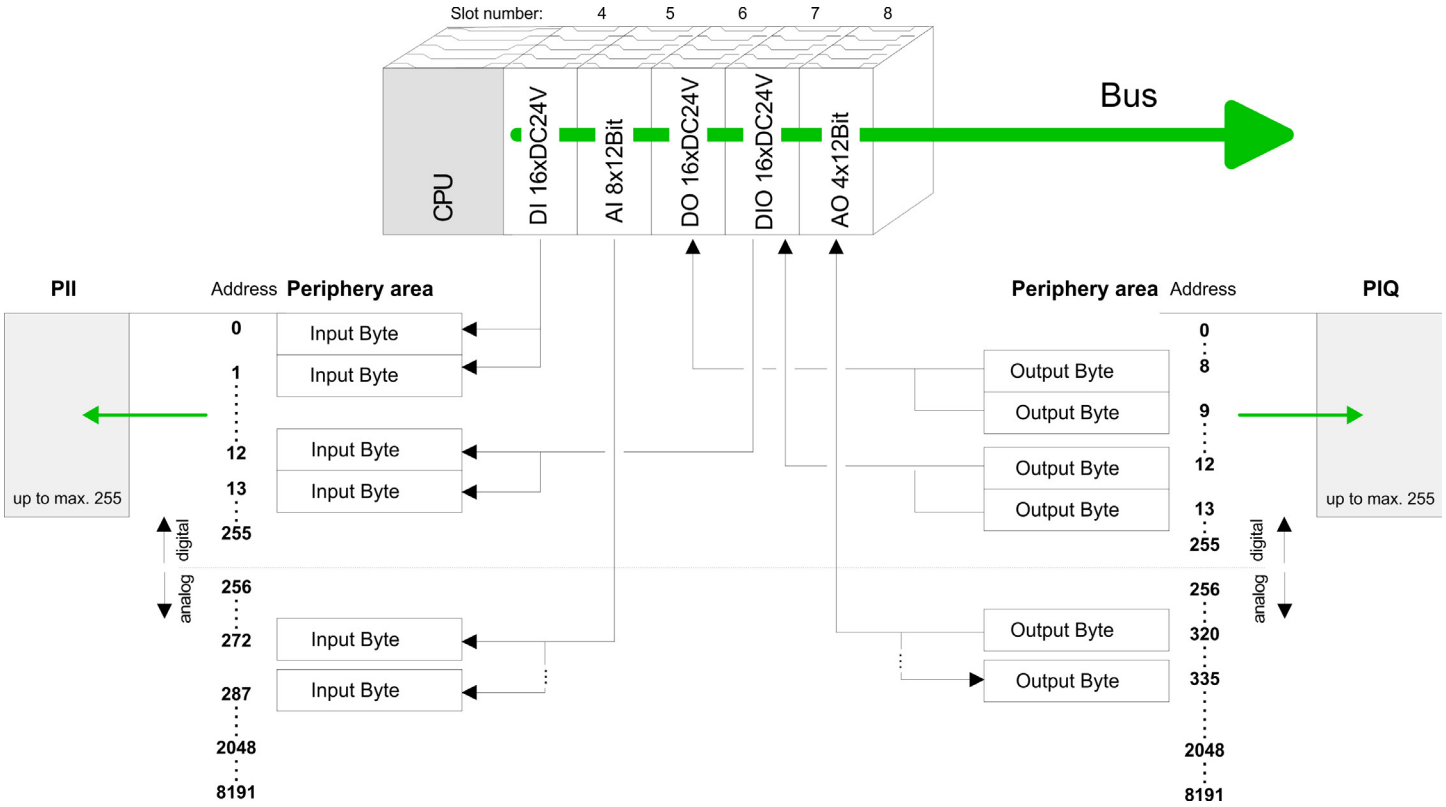
If you do not like to use a hardware configuration, an automatic addressing comes into force. At the automatic address allocation DIOs occupy depending on the slot location always 4byte and AIOs, FMs, CPs always 16byte at the bus. Depending on the slot location the start address from where on the according module is stored in the address range is calculated with the following formulas:

- DIOs: Start address = 4×(slot -1)
- AIOs, FMs, CPs: Start address = 16×(slot -1)+256



Example for automatic address allocation

The following sample shows the functionality of the automatic address allocation:



5.4 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. 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.

*Please consider that this SPEED7-CPU has 4 ACCUs. After an arithmetic operation (+I, -I, *I, /I, +D, -D, *D, /D, MOD, +R, -R, *R, /R) the content of ACCU 3 and ACCU 4 is loaded into ACCU 3 and 2. This may cause conflicts in applications that presume an unmodified ACCU 2.*

For more information may be found in the manual "VIPA Operation list SPEED7" at "Differences between SPEED7 and 300V programming".

Proceeding

Slot	Module
1	
2	**CPU 317-2DP**
X1	*MPI/DP*
X2	*DP*
3	

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 Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3).
4. The integrated PROFIBUS DP master (jack X3) is to be configured and connected via the sub module *"X2 (DP)"*.

5.4.1 Switching CPU type to CPU 318-2AJ00

Overview

To use projects, which were configured with the Siemens CPU type 318-2AJ00, you can switch from original CPU type to CPU type 318-2AJ00 by means of a CMD auto command. The setting is retained even after power cycle, firmware update or battery failure. With reset to factory settings respectively with the corresponding CMD auto command the CPU type is reset to the original CPU type.

Switching

- CPU type 318
  - Switching takes place with the CMD auto command *CPUTYPE_318*. After this perform a power cycle.
  - *Chap. 5.18 'CMD - auto commands' page 68*

```
CMD_START
CPUTYPE_318
CMD_END
```

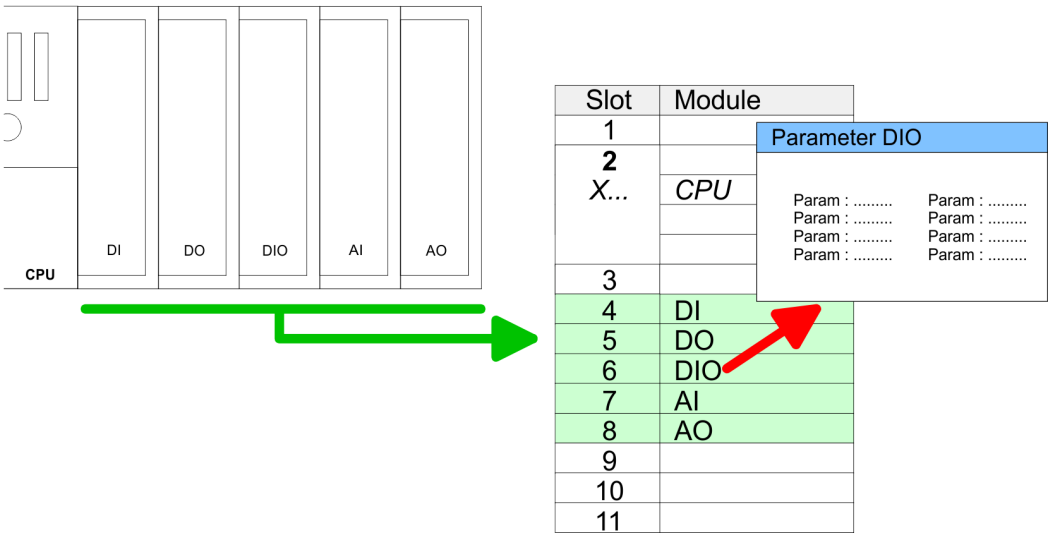
- CPU type original
  - The reset to the original type takes place with the CMD auto command *CPUTYPE_ORIGINAL* respectively by *Chap. 5.15 'Reset to factory settings' page 65*.
  - *Chap. 5.18 'CMD - auto commands' page 68*

```
CMD_START
CPUTYPE_ORIGINAL
CMD_END
```

5.5 Hardware configuration - I/O modules

Hardware configuration of the modules

After the hardware configuration place the System 300 modules in the plugged sequence starting with slot 4.



Parametrization

For parametrization double-click during the project engineering at the slot overview on the module you want to parameterize. In the appearing dialog window you may set the wanted parameters. 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.

Bus extension with IM 360 and IM 361

For the project engineering of more than 8 modules you may use line interface connections. For this you set in the hardware configurator the module IM 360 from the hardware catalog to slot 3 of your 1. profile rail. Now you may extend your system with up to 3 profile rails by starting each with an IM 361 from Siemens at slot 3. Considering the max. total current with the VIPA SPEED7 CPUs up to 32 modules may be arranged in a row. Here the installation of the line connections IM 360/361 from Siemens is not required.

5.6 Hardware configuration - Ethernet PG/OP channel

Overview

The CPU 315-4NE23 has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU. The PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc. With the first start-up respectively after an overall reset the Ethernet PG/OP channel does not have any IP address. For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this by means of the Siemens SIMATIC Manager. This is called "initialization".

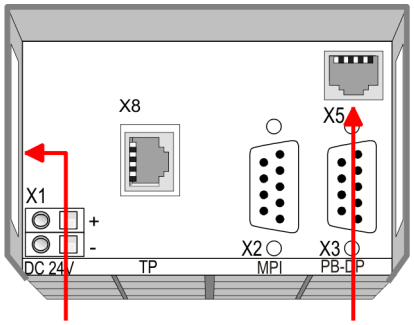
Assembly and commissioning

1. Install your System 300S with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the Ethernet jack 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:



Ethernet address **PG/OP channel**

1. Ethernet PG/OP channel
2. CP 343

→ Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This always may be found as 1. address under the front flap of the CPU on a sticker on the left side.

Assign IP address parameters

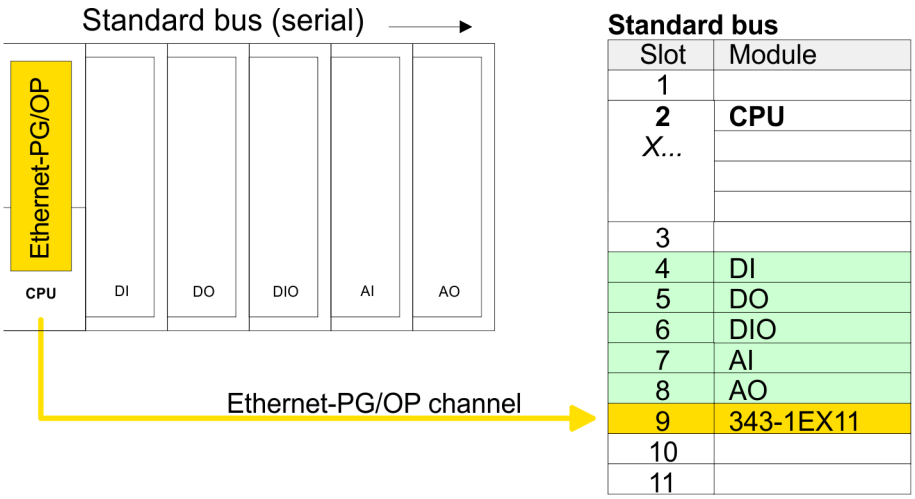
You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.3 & SP3 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 1. label beneath the front flap of the CPU.
4. Choose if necessary the known MAC address of the list of found stations.
5. Either type in the IP configuration like IP address, subnet mask and gateway.
6. Confirm with [Assign IP configuration].

   ⇨ Direct after the assignment the Ethernet PG/OP channel may be reached online by these address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.

Take IP address parameters in project

1. Open the Siemens hardware configurator und configure the Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3).
2. Configure the modules at the standard bus.
3. For the Ethernet PG/OP channel you have to configure a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX11 0XE0) always below the really plugged modules.
4. Open the property window via double-click on the CP 343-1EX11 and enter for the CP at *'Properties'* the IP address data, which you have assigned before.
5. Assign the CP to a *'Subnet'*. Without assignment the IP address data are not used!
6. Transfer your project.



5.7 Hardware configuration - Communication

The hardware configuration is described at the following pages:

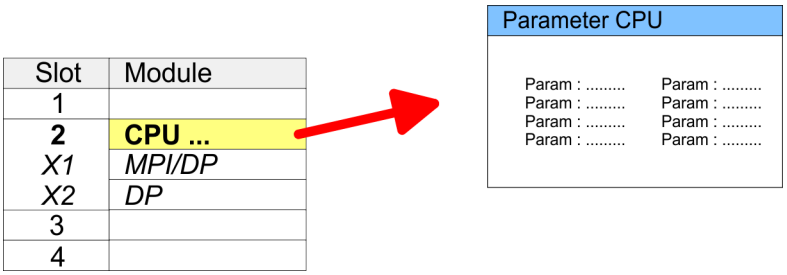
- *Chap. 7.4 'Deployment as PROFIBUS DP master' page 91*
- *Chap. 7.5 'Deployment as PROFIBUS DP slave' page 92*
- *Chap. 6.3 'Deployment of RS485 interface for PtP' page 75*
- *Chap. 8.9 'Hardware configuration - CP 343' page 110*

5.8 Setting standard CPU parameters

5.8.1 Parameterization via Siemens CPU

Parameterization via Siemens CPU

Since the CPU is to be configured as Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3) in the Siemens hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 317-2 DP during hardware configuration. Via a double-click on the CPU 317-2 DP the parameter window of the CPU may be accessed. Using the registers you get access to every standard parameter of the CPU.



5.8.2 Parameters CPU

Supported parameters

The CPU does not evaluate each parameter, which may be set at the hardware configuration. The following parameters are supported by the CPU at this time:

General

- Short description: The short description of the Siemens CPU is CPU 317-2DP (6ES7 317-2AK14 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 1346-1.
- 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 of 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 ready message by modules [100ms]
- Transfer of parameters to modules [100ms]: The maximum time for the transfer of parameters to parameterizable modules. 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: This parameter is not relevant.
- 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.
- 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: Here the priorities may be specified according to which the time-of-day interrupt is processed. With priority "0" the corresponding OB is deactivated.
- Active: Activate the check box of the time-of-day interrupt OBs if these are to be automatically started on complete restart.
- 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. With priority "0" the corresponding interrupt is deactivated.
- 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: Here the number of diagnostics are displayed, which may be stored in the diagnostics buffer (circular buffer).
- 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

5.8.3 Parameters for DP

The properties dialog of the PROFIBUS part is opened via a double click to the sub module DP.

General

- Short description: Here the short description "DP" for PROFIBUS DP is specified.
- Order no.: Nothing is shown here.
- Name: Here "DP" is shown. If you change the name, the new name appears in the Siemens SIMATIC Manager.
- Interface: The PROFIBUS address is shown here.
- Properties: With this button the properties of the PROFIBUS DP interface may be preset.
- Comment: You can enter the purpose of the PROFIBUS interface.

Address

- Diagnostics: A diagnostics address for PROFIBUS DP is to be preset here. In the case of an error the CPU is informed via this address.
- Operating mode: Here the operating mode of the PROFIBUS part may be preset. More may be found at chapter "Deployment PROFIBUS Communication".
- Configuration: Within the operating mode "DP-Slave" you may configure your slave system. More may be found at chapter "Deployment PROFIBUS communication".
- Clock: These parameters are not supported.

5.8.4 Parameters for MPI/DP

The properties dialog of the MPI interface is opened via a double click to the sub module MPI/DP.

General

- Short description: Here the short description "MPI/DP" for the MPI interface is specified.
- Order no.: Nothing is shown here.
- Name: At *Name* "MPI/DP" for the MPI interface is shown. If you change the name, the new name appears in the Siemens SIMATIC Manager.
- Type: Please regard only the type "MPI" is supported by the VIPA CPU.
- Interface: Here the MPI address is shown.
- Properties: With this button the properties of the MPI interface may be preset.
- Comment: You can enter the purpose of the MPI interface.

Address

- Diagnostics: A diagnostics address for the MPI interface is to be preset here. In the case of an error the CPU is informed via this address.
- Operating mode, Configuration, Clock: These parameters are not supported.

5.9 Setting Yaskawa specific CPU parameters

5.9.1 Proceeding

Overview

Except of the VIPA specific CPU parameters the CPU parameterization takes place in the parameter dialog of the CPU from Siemens. With installing of the SPEEDBUS.GSD the VIPA specific parameters may be set during hardware configuration. Here the following parameters may be accessed:

- Function RS485 X3 (PtP, Synchronization between DP master and CPU)
- Token Watch
- Number remanence flag, timer, counter
- Priority OB 28, OB 29, OB 33, OB 34
- Execution OB 33, OB 34
- Phase offset OB 33, OB 34
- Call OB 80 on cyclic interrupt error

Requirements

Since the VIPA specific CPU parameters may be set, the installation of the SPEEDBUS.GSD from VIPA in the hardware catalog is necessary. The CPU may be configured in a PROFIBUS master system and the appropriate parameters may be set after installation.

Installation of the SPEEDBUS.GSD

The GSD (Geräte-Stamm-Datei) is online available in the following language versions. Further language versions are available on inquires:

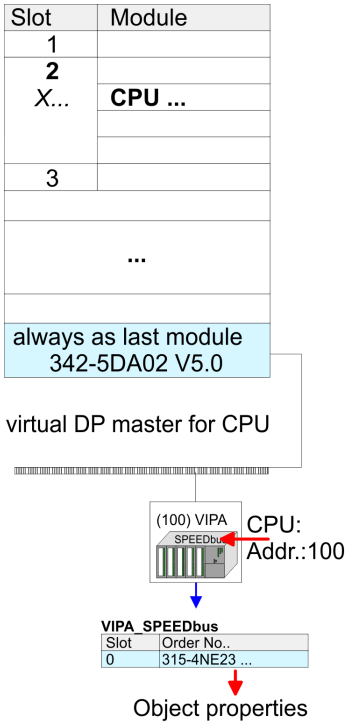
Name	Language
SPEEDBUS.GSD	German (default)
SPEEDBUS.GSG	German
SPEEDBUS.GSE	English

The GSD files may be found at www.yaskawa.eu.com at the service area.

The integration of the SPEEDBUS.GSD takes place with the following proceeding:

1. Go to the service area of www.yaskawa.eu.com.
2. Load from the download area at *'Config files ➔ PROFIBUS'* the according file for your System 300S.
3. Extract the file to your work directory.
4. Start the hardware configurator from Siemens.
5. Close every project.
6. Select *'Options ➔ Install new GSD-file'*.
7. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from Yaskawa may now be found in the hardware catalog at PROFIBUS-DP / Additional field devices / I/O / VIPA_SPEEDBUS.

Hardware configuration



The embedding of the CPU 315-4NE23 happens by means of a virtual PROFIBUS master system with the following approach:

1. Perform a hardware configuration for the CPU. *Chap. 5.4 'Hardware configuration - CPU' page 40*
2. Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".
3. Connect the slave system "VIPA_SPEEDbus". After installing the SPEEDBUS.GSD this may be found in the hardware catalog at Profibus-DP / Additional field devices / I/O / VIPA / VIPA_SPEEDBUS.
4. For the slave system set the PROFIBUS address 100.
5. Configure at slot 0 the VIPA CPU 315-4NE23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 315-4NE23 the properties dialog of the CPU may be opened.

The hardware configuration, which is shown here, is only required, if you want to customize the VIPA specific parameters.

5.9.2 Yaskawa specific parameters

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

5.9.2.1 Function RS485 X3

Using this parameter the RS485 interface may be switched to PtP communication (**p**oint **t**o **p**oint) respectively the synchronization between DP master system and CPU may be set:

Deactivated	Deactivates the RS485 interface.
PtP	With this operating mode the PROFIBUS DP master is deactivated and 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.
PROFIBUS DP async	PROFIBUS DP master operation asynchronous to CPU cycle The RS485 interface is preset at default to PROFIBUS DP async. Here CPU cycle and cycles of every Yaskawa PROFIBUS DP master run independently.
PROFIBUS DP syncIn	The CPU is waiting for DP master input data.

PROFIBUS DP syncOut	The DP master system is waiting for CPU output data.
PROFIBUS DP syncInOut	CPU and DP master system are waiting on each other and form thereby a cycle.
Default: PROFIBUS DP async	

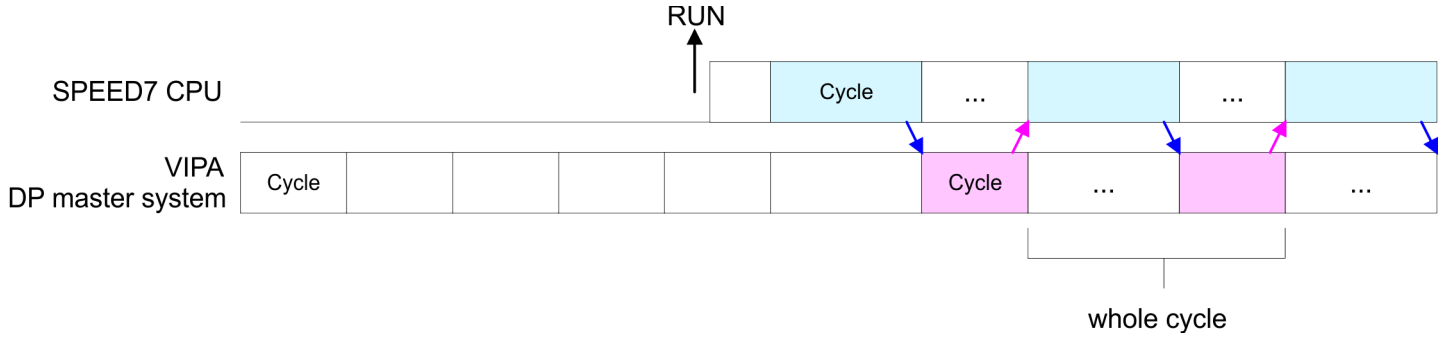
5.9.2.1.1 Synchronization between master system and CPU

Overview

Normally the cycles of CPU and DP master run independently. The cycle time of the CPU is the time needed for one OB1 cycle and for reading respectively writing the inputs respectively outputs. The cycle time of a DP master depends among others on the number of connected slaves and the baud rate, thus every plugged DP master has its own cycle time. Due to the asynchronism of CPU and DP master the whole system gets relatively high response times. The synchronization behavior between every Yaskawa PROFIBUS DP master and the CPU may be configured by means of a hardware configuration as shown above. The different modes for the synchronization are in the following described.

PROFIBUS DP SyncInOut

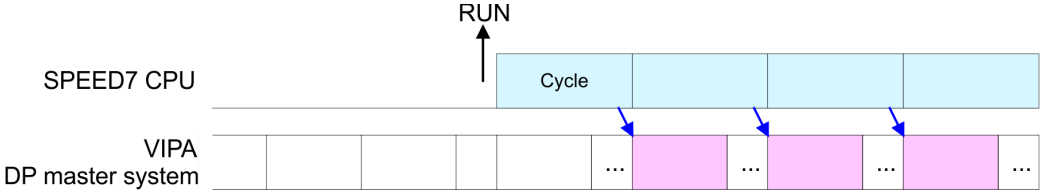
In PROFIBUS DP SyncInOut mode CPU and DP master system are waiting on each other and form thereby a cycle. Here the whole cycle is the sum of the longest DP master cycle and CPU cycle. By this synchronization mode you receive global consistent in-/output data, since within the total cycle the same input and output data are handled successively by CPU and DP master system. If necessary the time of the Watchdog of the bus parameters should be increased at this mode.



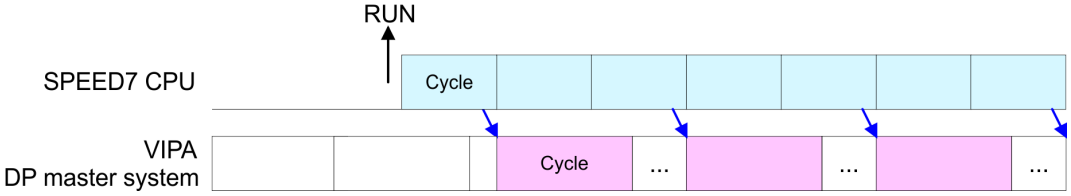
PROFIBUS DP SyncOut

In this operating mode the cycle time of the Yaskawa DP master system depends on the CPU cycle time. After CPU start-up the DP master gets synchronized. As soon as their cycle is passed they wait for the next synchronization impulse with output data of the CPU. So the response time of your system can be improved because output data were directly transmitted to the DP master system. If necessary the time of the Watchdog of the bus parameters should be increased at this mode.

Cycle CPU > DPM "1-tier CPU cycle"



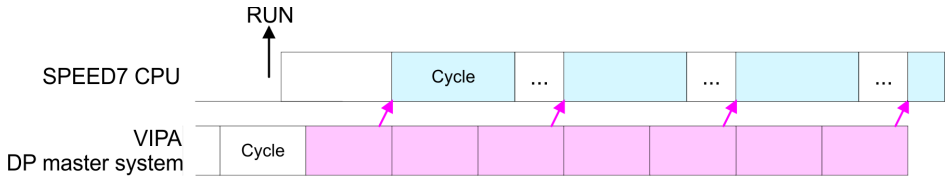
Cycle CPU < DPM "n-tier CPU cycle"



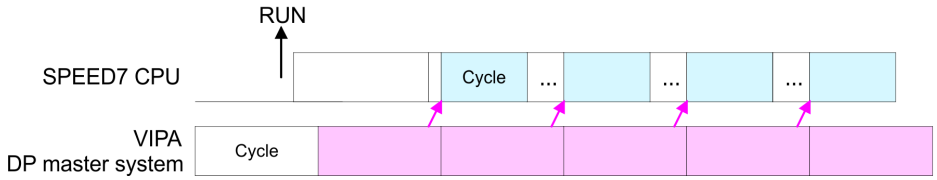
PROFIBUS-DP SyncIn

In the operating mode PROFIBUS DP SyncIn the CPU cycle is synchronized to the cycle of the Yaskawa PROFIBUS DP master system. Here the CPU cycle depends on the Yaskawa DP master with the longest cycle time. If the CPU gets into RUN it is synchronized with each PROFIBUS DP master. As soon as the CPU cycle is passed, it waits for the next synchronization impulse with input data of the DP master system. If necessary the Scan Cycle Monitoring Time of the CPU should be increased.

Cycle CPU > DPM
"n-tier longest DP master cycle"



Cycle CPU < DPM
"1-tier longest DP master cycle"



5.9.2.2 Token Watch

By presetting the PROFIBUS bus parameters within the hardware configuration a token time for the PROFIBUS results. The token time defines the duration until the token reaches the DP master again. Per default this time is supervised. Due to this monitoring disturbances on the bus can affect a reboot of the DP master. Here with the parameter Token Watch the monitoring of the token time can be switched off respectively on.

Default: On

5.9.2.3 Number remanence flag

Here the number of flag bytes may be set. With 0 the value Retentive memory > Number of memory bytes starting with MB0 set at the parameters of the Siemens CPU is used. Otherwise the adjusted value (1 ... 8192) is used. Default: 0

5.9.2.4 Phase offset and execution of OB 33 and OB 34

The CPU offers additional cyclic interrupts, which interrupt the cyclic processing in certain distances. Point of start of the time interval is the change of operating mode from STOP to RUN. To avoid that the cyclic interrupts of different cyclic interrupt OBs receive a start request at the same time and so a time out may occur, there is the possibility to set a phase offset respectively a time of execution.

- The *phase offset* (0 ... 60000ms) serves for distribution processing times for cyclic interrupts across the cycle. Default: 0
- The time intervals, in which the cyclic interrupt OB should be processed may be entered with *execution* (1 ... 60000ms). Default: OB 33: 500ms, OB 34: 200ms

5.9.2.5 Priority of OB 28, OB 29, OB 33 and OB 34

The priority fixes the order of interrupts of the corresponding interrupt OB. Here the following priorities are supported: 0 (Interrupt-OB is deactivated), 2, 3, 4, 9, 12, 16, 17, 24. Default: 24

5.9.2.6 Call OB 80 on cyclic interrupt error

Once during a cyclic interrupt OB (OB 28, 29, 32 ... 35) the same cyclic interrupt is requested, the interrupt requests are collected and processed sequentially. Via the parameter *'OB 80 for cyclic interrupt'* you can set here for the corresponding cyclic interrupt group that on a cyclic interrupt instead of the sequential processing the OB 80 is to be called. With this parameter you have the following settings:

- Deactivated (default)
  - At a cyclic interrupt error the interrupt requests are collected and processed sequentially.
- for OB...
  - At a cyclic interrupt error of the corresponding cyclic interrupt OB, the OB 80 is called.

5.10 Project transfer

Overview

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

- Transfer via MPI/PROFIBUS
- Transfer via Ethernet
- Transfer via memory card

5.10.1 Transfer via MPI/PROFIBUS

General

For transfer via MPI/PROFIBUS the CPU has the following interface:

- X2: MPI interface
- X3: PROFIBUS interface

Net structure

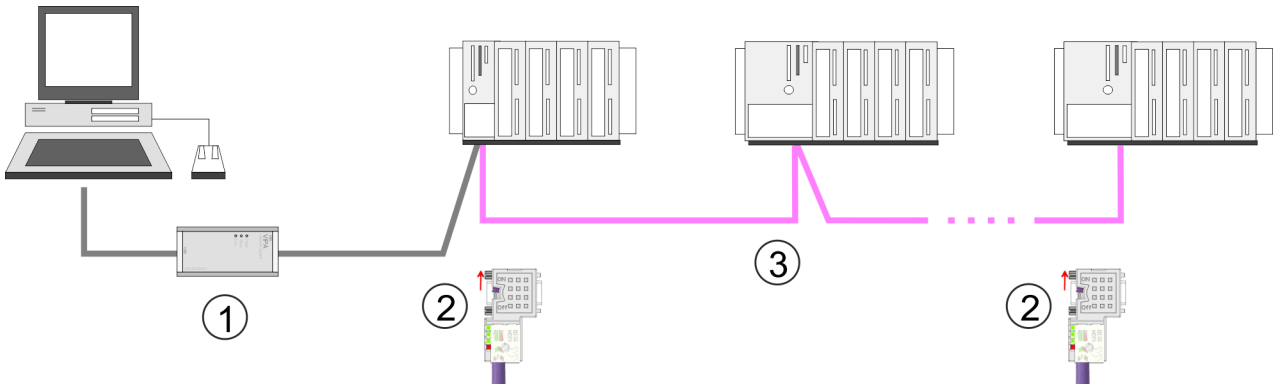
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

MPI programming cable

The MPI programming cables are available at Yaskawa 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 interface

1. Connect your PC to the MPI jack 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 (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 PC and the transfer rate 38400baud for the MPI programming cable.
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

1. Connect your PC to the PROFIBUS jack 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 PC and the transfer rate 38400baud for the MPI programming cable.
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.

Within selecting the slave mode you have additionally to select the option "Test, commissioning, routing".

5.10.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X5: Ethernet PG/OP channel
- X8: CP 343 communication processor

Initialization

So that you may access the Ethernet PG/OP channel you have to assign IP address parameters by means of the "initialization". ↬ *'Assign IP address parameters' page 42*,

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.

5.10.3 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.

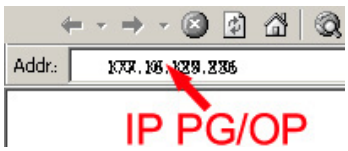
   S7PROG.WLD is read from the memory card after overall reset.

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

   The short flashing of the MC LED of the CPU indicates the transfer process. 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.

5.11 Accessing the web server

Access to the web server



There is a web server, which can be accessed via the IP address of the Ethernet PG/OP channel with an Internet browser. At the web page information about the CPU and its connected modules can be found. *Chap. 5.6 'Hardware configuration - Ethernet PG/OP channel' page 41*

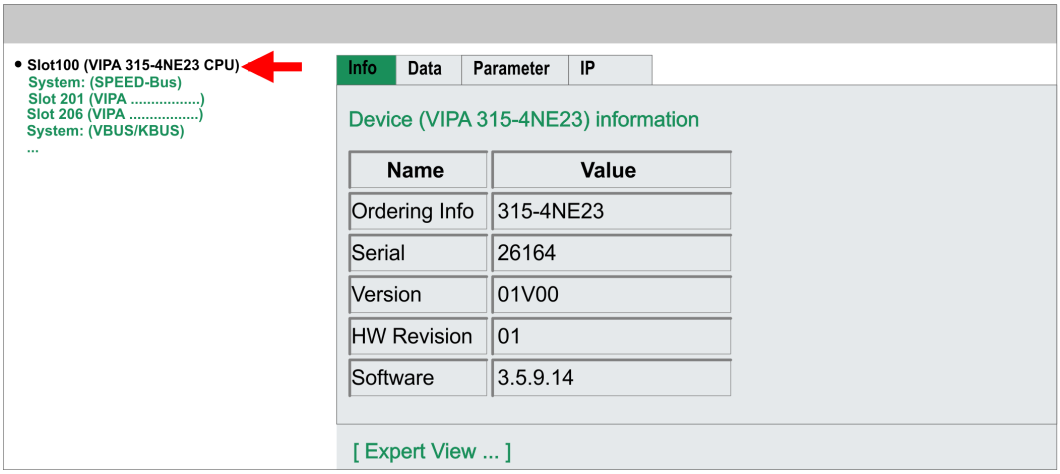
It is assumed that there is a connection between PC and CPU with Internet browser via the Ethernet PG/OP channel. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.

Structure of the web page

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

Info - Overview

CPU



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

Info - Expert View

Runtime Information		
Operation Mode	STOP	CPU: Status information
Mode Switch	RUNP	
System Time	01.09.09 00:35:30:812	CPU: Date, time
OB1-Cycle Time	cur = 0us, min = 0us, max = 0us, avg = 0us	CPU: Cyclic time: min = minimum cur = current max = maximum avg = average
Interface Information		
X2 (RS485/COM1)	MPI	Operating mode RS485 ■ MPI: MPI operation

X3 (RS485/COM2)	PtP	■ DPM: DP master operation or PtP: point to point operation
X5	PG/OP Ethernet Port	
Card Information		
Type	SD	
Product S/N	6BC34010	
Size	493617152 bytes	
Free	492355584 bytes	
Active Feature Set Information		
Status	Memory Extension present	
Memory Usage		
LoadMem	0 / 4194304 Bytes	CPU: Information to memory configuration Load memory, working memory (code/data)
WorkMemCode	0 / 524288 Bytes	
WorkMemData	0 / 524288 Bytes	
PG/OP Network Information		
Device Name	Yaskawa 315-4NE23 CPU	Ethernet PG/OP channel:
IP Address	172.16.129.210	Address information
Subnet Mask	255.255.255.0	
Gateway Address	172.16.129.210	
MAC Address	00:20:D5:77:30:36	
CPU Firmware Information		
File System	V1.0.2	Information for the support
PRODUCT	Yaskawa 315-4NE23 V3.7.5 Px000307.pkg	Name, firmware version, package
HARDWARE	V0.1.0.0 5679H-V20 HX000027.110	CPU: Information for the support
Bx000227	V6.6.29.255	
Ax000086	V1.2.1.0	
Ax000056	V0.2.2.0	
fx000007.wld	V1.1.8.0	
ARM Processor Load		
Last Value	0%	
Maximum load	41%	

Data

Currently nothing is displayed here.

Parameter

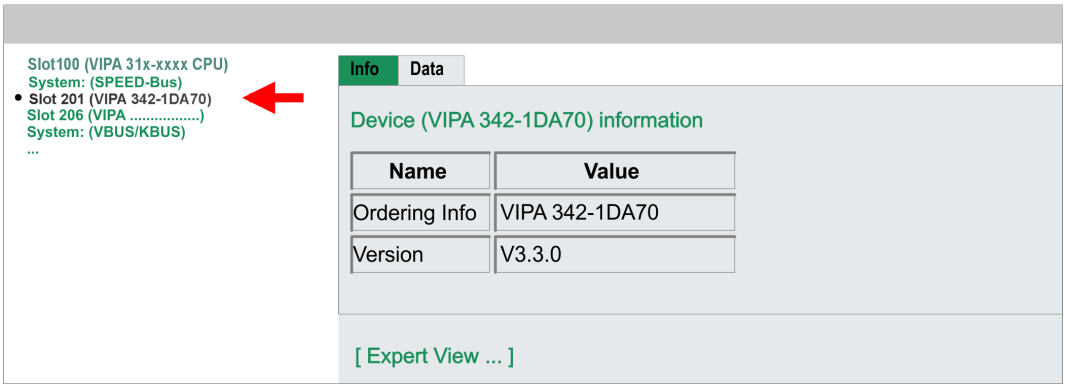
Currently nothing is displayed here.

IP

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

Info - Overview

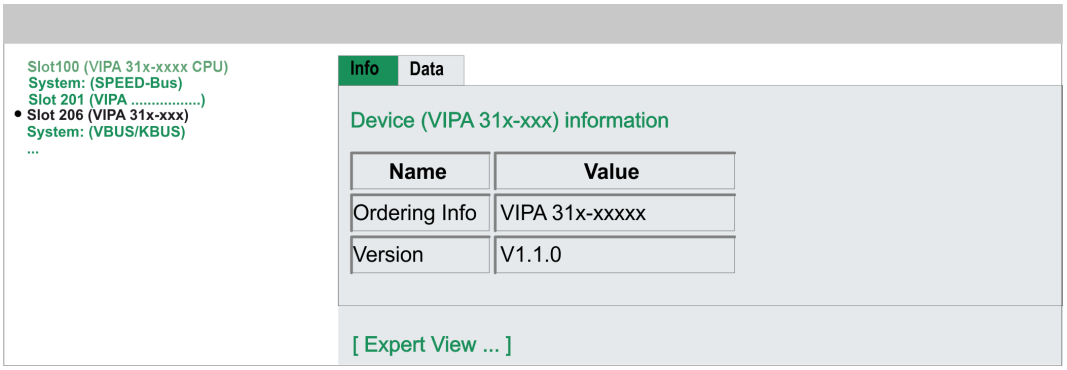
DP master



Info - Expert View

Internal Information	**Slot 201**	Yaskawa 342-1DA70
Module Type	0xCB2C0010	
Module Firmware Information		
PRODUCT	Yaskawa 342-1DA70 V3.3.5 Px000182.pkg	Name, firmware-version, package
BB000218	V5.3.0.0	Information for support
AB000068	V4.1.7.0	
Runtime Information		
Cycle Time	cur = 0us, min = 65535000us, max = 0us, avg = 0us, cnt = 0	CPU cycle time: min = minimal cur = current max = maximal

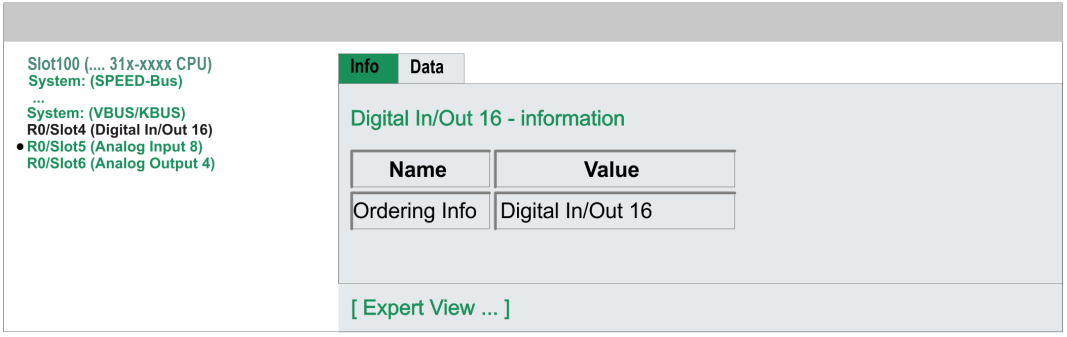
Info - Overview **CP 343**



Info - Expert View

Internal Information		**CPU component: CP 343**
Module Type	0xACDB0080	Information for support
Module Firmware Information		
Bb000429	V1.1.0.12	
AB000125	V0.1.0.3	
PRODUCT	Yaskawa 315-4NE23 V3.1.1 Px000299.pkg	
Hx000075	V1.1.0.0	

Info - Overview **VBUS - Digital In/Out 16**



Data - Input data

Offset	Width	Value (dec)	Value (hex)
0	1	0	00
1	1	0	00

Data - Output data

Offset	Width	Value (dec)	Value (hex)	New Value (hex)
0	1	0	00	00
1	1	0	00	00

5.12 Operating modes

5.12.1 Overview

The CPU can be in one of 4 operating modes:

- Operating mode STOP
- Operating mode START-UP
- 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.
- Outputs are inhibited, i.e. 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, i.e. outputs are inhibited.
- 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 off

When the CPU has completed the START-UP OB, it assumes the operating mode RUN.

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.
- The BASP-signal (outputs inhibited) is deactivated, i.e. all digital 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.

5.12.2 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 Yaskawa CPUs are developed function secure and have the following system properties:

Event	concerns	Effect
RUN → STOP	general	BASP (Befehls-Ausgabe-Sperre, i.e. command output disable) is set.
	central digital outputs	The outputs are disabled.
	central analog outputs	The outputs are disabled. ■ Voltage outputs issue 0V ■ Current outputs 0...20mA issue 0mA ■ Current outputs 4...20mA issue 4mA If configured also substitute values may be issued.
	decentral outputs	Same behavior 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 once be read by the decentralized station and the recent values are put at disposal.
RUN	general	The program execution happens cyclically and can therefore be foreseen: Read PII → OB 1 → Write PIO.

PII: Process image inputs, PIO: Process image outputs

5.13 Overall reset

Overview

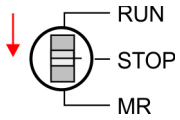
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. If you have assigned IP address data to your CP 343, these remain until there is a new PowerON.

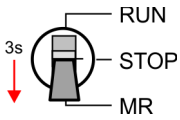
You have 2 options to initiate an overall reset:

- initiate the overall reset by means of the operating mode switch
- initiate the overall reset by means of the Siemens SIMATIC Manager

You should always issue 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 blinks. When the ST LED is on permanently the overall reset procedure has been completed.

Automatic reload

If there is a project S7PROG.WLD on the MMC, the CPU attempts to reload this project from MMC.

→ The MC 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.

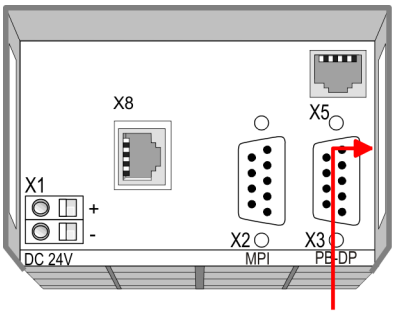
Reset to factory setting

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. 5.15 'Reset to factory settings' page 65*

5.14 Firmware update

Overview

- There is the opportunity to execute a firmware update for the CPU and its components via memory card. For this an accordingly prepared memory card must be in the CPU during the startup.
- So a firmware files can be recognized and assigned with startup, a pkg file name is reserved for each updateable component an hardware release, which begins with "px" and differs in a number with six digits. The pkg file name of every updateable component may be found at a label right down the front flap of the module.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file on 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.



Firmware package and Version

Latest firmware at www.yaskawa.eu.com

The latest firmware versions are to be found in the service area at www.yaskawa.eu.com. For example the following files are necessary for the firmware update of the CPU 315-4NE23 and its components with hardware release 1:

- 315-4NE23, Hardware release 01: Px000307.pkg
- PROFIBUS DP master: Px000182.pkg
- Ethernet-CP 343: Px000299.pkg

CAUTION!

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

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

Display the firmware version of the SPEED7 system via Web Site

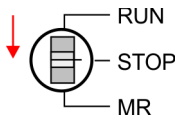
The CPU has an integrated website that monitors information about firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web site. The CPU has an integrated website that monitors information about firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web site. *'PLC ➔ Assign Ethernet Address'*. After that you may access the PG/OP channel with a web browser via the IP address of the project engineering. *Chap. 5.11 'Accessing the web server' page 54*

Load firmware and transfer it to memory card

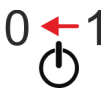
- Go to www.yaskawa.eu.com
- Click on *'Service ➔ Download ➔ Firmware'*.
- Navigate via *'System 300S ➔ CPU'* to your CPU and download the zip file to your PC.
- Extract the zip file and copy the extracted pkg files to your memory card.

CAUTION!
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update! After the firmware update you should execute a *Chap. 5.15 'Reset to factory settings' page 65*.

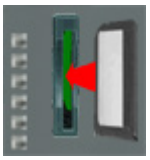
Transfer firmware from memory card into CPU



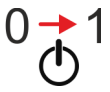
1. Switch the operating mode switch of your CPU in position STOP.



2. Turn off the power supply.

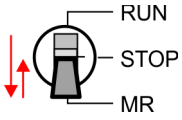


3. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card.



4. Turn on the power supply.

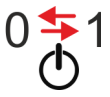
⇨ After a short boot-up time, the alternate blinking of the LEDs SF and FC shows that at least a more current firmware file was found at the memory card.



5. You start the transfer of the firmware as soon as you tip the operating mode switch downwards to MR within 10s and then leave the switch in STOP position.

⇨ During the update process, the LEDs SF and FC are alternately blinking and the MC LED is on. This may last several minutes.

6. The update is successful finished when the LEDs PW, ST, SF, FC and MC are on. If they are blinking fast, an error occurred.



7. Turn power OFF and ON.

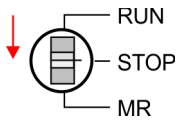
⇨ Now it is checked by the CPU, whether further firmware updates are to be executed. If so, again the LEDs SF and FC flash after a short start-up period. Continue with step 5. If the LEDs do not flash, the firmware update is finished.

8. Now execute a *Reset to factory setting*. After that the CPU is ready for duty.
Chap. 5.15 'Reset to factory settings' page 65

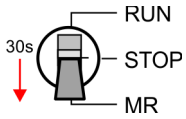
5.15 Reset to factory settings

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. 5.18 'CMD - auto commands' page 68*



1. Switch the CPU to STOP.



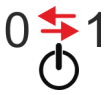
2. Push the operating mode switch down to position MR for 30 seconds. Here the ST LED blinks. After a few seconds the ST LED changes to static light. Now the ST LED changes between static light and blinking. Start here to count the static light of the ST LED.

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

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

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

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



5. Turn power OFF and ON.

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

5.16 Deployment storage media - MMC, MCC

Overview

At this slot the following storage media can be plugged:

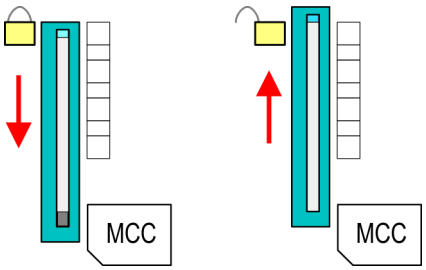
- SD respectively MMC (**M**ulti**m**edia **c**ard)
  - External memory card for programs and firmware.
- MCC - **M**emory **c**onfiguration **c**ard
  - External memory card (MMC) for programs and firmware with the possibility to unlock additional work memory.
  - The additional memory can be purchased separately.
  - To activate the corresponding card is to be installed and an *overall reset* is to be established. *Chap. 5.13 'Overall reset' page 62*

To avoid malfunctions, you should use memory cards of Yaskawa. These correspond to the industrial standard. A list of the currently available memory cards can be found at www.yaskawa.eu.com

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

MMC

- The MMCs of Yaskawa are pre-formatted with the PC format FAT and can be accessed via a card reader.
- After PowerON respectively an overall reset the CPU checks, if there is a memory card plugged with data valid for the CPU.
- Push the memory card into the slot until it snaps in leaded by a spring mechanism. This ensures contacting. By sliding down the sliding mechanism, a just installed memory 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!

Please note that the write protection function of SD cards is not evaluated!

MCC

- The MCC is a MMC with the possibility to unlock additional work memory.
- By plugging the MCC into the MCC slot and then an overall reset the according memory expansion is released. There may only one memory expansion be activated at one time.
- On the MCC there is the file memory.key. This file may not be altered or deleted.

- You may use the MCC also as "normal" MMC for storing your project.
- If the memory expansion on the MCC exceeds the maximum extendible memory range of the CPU, the maximum possible memory of the CPU is automatically used.
- You may determine the recent memory extension and the remaining time after pulling the MCC via the integrated web page. *Chap. 5.11 'Accessing the web server' page 54*
- When the MCC memory configuration has been taken over you may find the diagnostics entry 0xE400 in the diagnostics buffer of the CPU.
- After pulling the MCC the entry 0xE401 appears in the diagnostics buffer, the SF-LED is on and after 72 hours the CPU switches to STOP. A reboot is only possible after plugging-in the MCC again or after an overall reset.
- After re-plugging the MCC, the SF LED extinguishes and 0xE400 is entered into the diagnostics buffer. You may reset the memory configuration of your CPU to the initial status at any time by executing an overall reset without MCC.

CAUTION!
Please regard that the MCC must remain plugged when you've executed the memory expansion at the CPU. Otherwise the CPU switches to STOP after 72 hours. The MCC cannot be exchanged with a MCC of the same memory configuration. The activation code is fixed to the MCC by means of an unique serial number. Here the functionality 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 MCC is plugged. If so, the according additional memory is unlocked.
- The CPU checks whether a project S7PROG.WLD exists. If so, it is automatically loaded.

After PowerON

- The CPU checks whether a project AUTOLOAD.WLD exists. If so, an overall reset is executed and the project is automatically loaded.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists. If so the command file is loaded and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chap. 5.14 'Firmware update' page 63*

Once in STOP state

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

The FC/SFC 208 ... FC/SFC 215 and FC/SFC 195 allow you to include the memory card access into your user application. More can be found in the manual operation list (HB00_OPL_SP7) of your CPU.

5.17 Extended know-how protection

Overview

Besides the "standard" Know-how protection the SPEED7 CPUs from Yaskawa 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 Yaskawa 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. 5.13 'Overall reset' page 62*

   ⇨ 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.

5.18 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 stuck
- After each PowerON

Command file

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

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

Commands

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

Command	Description	Diagnostics entry
CMD_START	In the first line *CMD_START* is to be located.	0xE801
	There is a diagnostic entry if *CMD_START* is missing	0xE8FE
WAIT1SECOND	Waits about 1 second.	0xE803
WEBPAGE	The current web page of the CPU is stored at the memory card as" webpage.htm".	0xE804
LOAD_PROJECT	The function "Overall reset and reload from MMC" 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
SET_MPI_ADDRESS	This lets you adjust the MPI interface on the value that follows the command. The setting is retained even after power cycle, firmware update or battery failure. With ↬ *Chap. 5.15 'Reset to factory settings' page 65* you get the default setting.	0xE814
CPUTYPE_318	To use projects, which were configured with the CPU type 318-2AJ00, with this command you can switch from original CPU type to CPU type 318-2AJ00. The setting is retained even after power cycle, firmware update or battery failure.	0xE82A
CPUTYPE_ORIGINAL	With ↬ *Chap. 5.15 'Reset to factory settings' page 65* or with this command the CPU type is reset to the original CPU type.	0xE82B
CMD_END	In the last line *CMD_END* is to be located.	0xE802

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

Example 1

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj.wld	Execute an overall reset and load "proj.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
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)
SET_MPI_ADDRESS 4	MPI address 4 is set (0xE814)
CPUTYPE_318	Switches the CPU type to CPU 318-2AJ00 (0xE82A)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
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.

5.19 Diagnostic entries

Accessing diagnostic data ↳ *Appendix A 'System specific event IDs' page 159*

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostics buffer, the Yaskawa 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. 5.18 'CMD - auto commands' page 68*
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

5.20 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 / ACCU 1
- ACCU 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.

6 Deployment PtP communication

6.1 Fast introduction

General

The CPU has a PROFIBUS/PtP interface with a fix pinout. After an overall reset the interface is deactivated. By appropriate configuration the PtP function (**p**oint **t**o **p**oint) 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.
  - The activation of the PtP functionality happens by embedding the SPEEDBUS.GSD from VIPA in the hardware catalog. After the installation the CPU may be configured in a PROFIBUS master system and here the interface may be switched to PtP communication.

Protocols

The protocols res. 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.

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

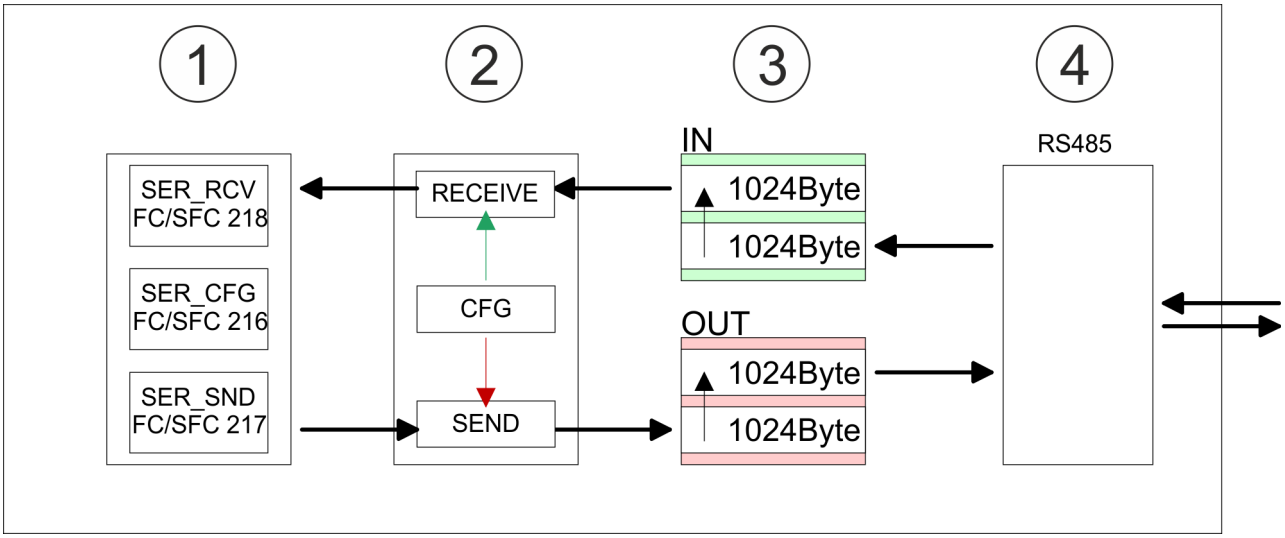
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from Yaskawa.

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

6.3 Deployment of RS485 interface for PtP

Activate RS485 to PtP operation

Per default, the RS485 interface is deactivated. Via hardware configuration the RS485 interfaces may be switched to PtP operation (**p**oint **t**o **p**oint) via the parameter *Function RS485* of the *Properties.*

Requirements

Since the VIPA specific CPU parameters may be set, the installation of the SPEEDBUS.GSD from VIPA in the hardware catalog is necessary. The CPU may be configured in a PROFIBUS master system and the appropriate parameters may be set after installation.

Installation of the SPEEDBUS.GSD

The GSD (Geräte-Stamm-Datei) is online available in the following language versions. Further language versions are available on inquires:

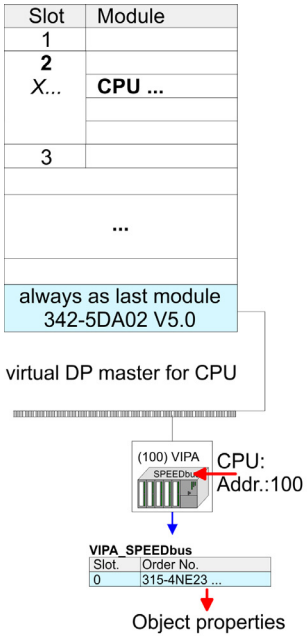
Name	Language
SPEEDBUS.GSD	German (default)
SPEEDBUS.GSG	German
SPEEDBUS.GSE	English

The GSD files may be found at www.yaskawa.eu.com at the service area.

The integration of the SPEEDBUS.GSD takes place with the following proceeding:

1. Go to the service area of www.yaskawa.eu.com.
2. Load from the download area at *'Config files ➔ PROFIBUS'* the according file for your System 300S.
3. Extract the file to your work directory.
4. Start the hardware configurator from Siemens.
5. Close every project.
6. Select *'Options ➔ Install new GSD-file'*.
7. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from Yaskawa may now be found in the hardware catalog at PROFIBUS-DP / Additional field devices / I/O / VIPA_SPEEDBUS.

Proceeding



The embedding of the CPU 315-4NE23 happens by means of a virtual PROFIBUS master system with the following approach:

1. Perform a hardware configuration for the CPU ☞ *Chap. 5.4 'Hardware configuration - CPU' page 40*
2. Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".
3. Connect the slave system "VIPA_SPEEDbus". After installing the SPEEDBUS.GSD this may be found in the hardware catalog at PROFIBUS DP / Additional field devices / I/O / VIPA / VIPA_SPEEDBUS.
4. For the slave system set the PROFIBUS address 100.
5. Configure at slot 0 the VIPA CPU 315-4NE23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 315-4NE23 the properties dialog of the CPU may be opened.

As soon as the project is transferred together with the PLC user program to the CPU, the parameters will be taken after start-up.

The hardware configuration, which is shown here, is only required, if you want to customize the VIPA specific parameters.

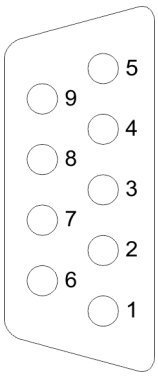
Setting PtP parameters

1. By double clicking the CPU 315-4NE23 placed in the slave system the properties dialog of the CPU may be opened.
2. Switch the Parameter *'Function RS485 X3'* to *'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

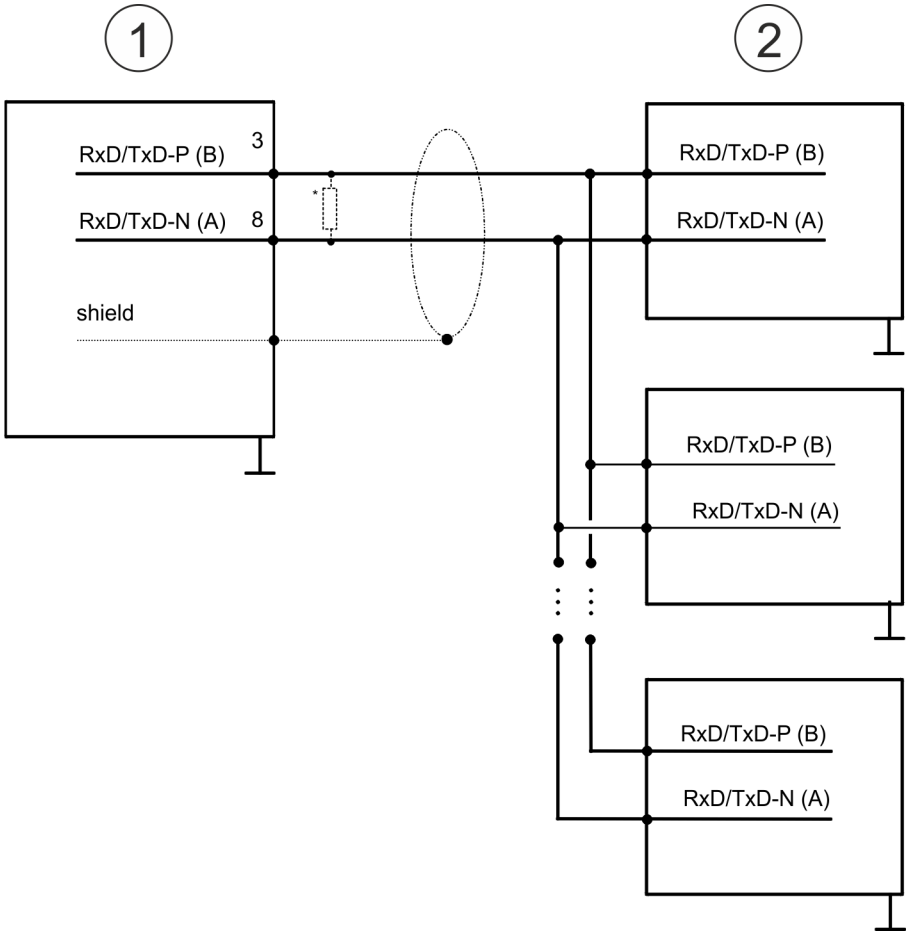
RS485



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 .*

6.4 Parametrization

6.4.1 FC/SFC 216 - SER_CFG - Parametrization PtP

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

6.5 Communication

6.5.1 FC/SFC 217 - SER_SND - Send to PtP

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

6.5.2 FC/SFC 218 - SER_RCV - Receive from PtP

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

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

6.6 Protocols and procedures

Overview

The CPU supports the following protocols and procedures:

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

ASCII

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

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

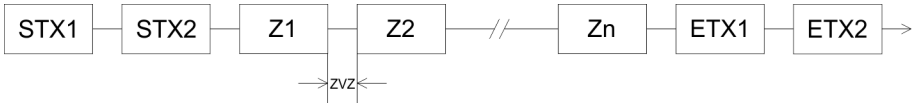
STX/ETX

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

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

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

Message structure:



3964

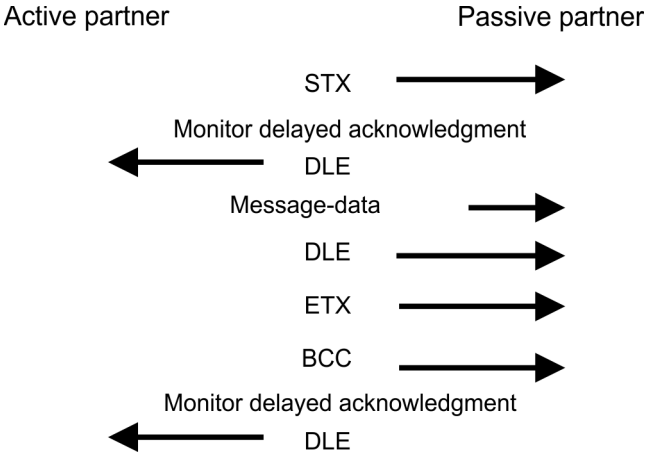
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

The procedure employs the following control characters:

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

You may transfer a maximum of 255byte per message.

Procedure



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

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

USS

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

The following features characterize the USS protocol:

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

It is essential:

- You may connect 1 master and max. 31 slaves at the bus
- The single slaves are addressed by the master via an address sign in the telegram.
- The communication happens exclusively in half-duplex operation.
- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.

The telegrams for send and receive have the following structure:

Master slave telegram

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

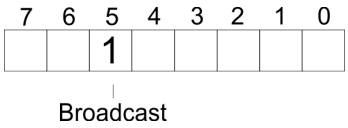
Slave master telegram

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

with

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

Broadcast with set bit 5 in ADR byte



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

Modbus

- The Modbus protocol is a communication protocol that fixes a hierarchic structure with one master and several slaves.
- Physically, Modbus works with a serial half-duplex connection. There are no bus conflicts occurring, because the master can only communicate with one slave at a time.
- After a request from the master, this waits for a preset delay time for an answer of the slave. During the delay time, communication with other slaves is not possible.

- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.
- The request telegrams send by the master and the respond telegrams of a slave have the following structure:

Telegram structure

Start sign	Slave address	Function Code	Data	Flow control	End sign

Broadcast with slave address = 0

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

ASCII, RTU mode

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

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

Supported Modbus protocols

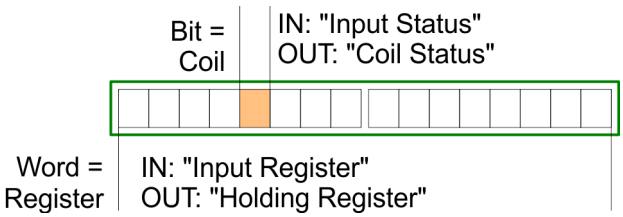
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

6.7 Modbus - Function codes

Naming convention

Modbus has some naming conventions:



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

Range definitions

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

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

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

0x - Bit area for master output data

Access via function code 01h, 05h, 0Fh

1x - Bit area for master input data

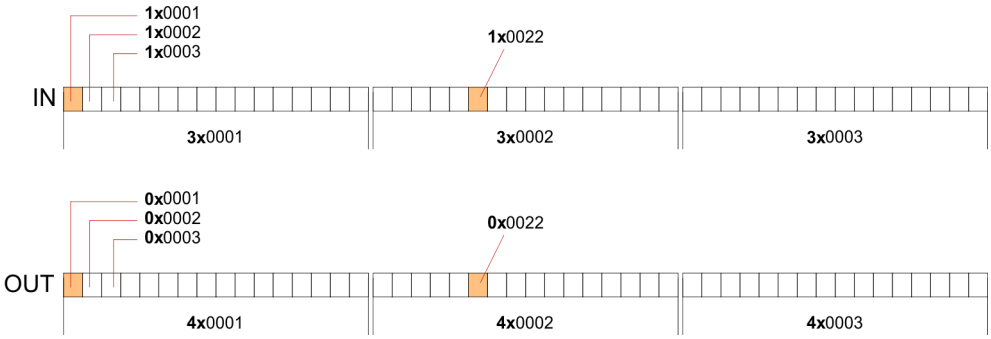
Access via function code 02h

3x - word area for master input data

Access via function code 04h

4x - word area for master output data

Access via function code 03h, 06h, 10h



A description of the function codes follows below.

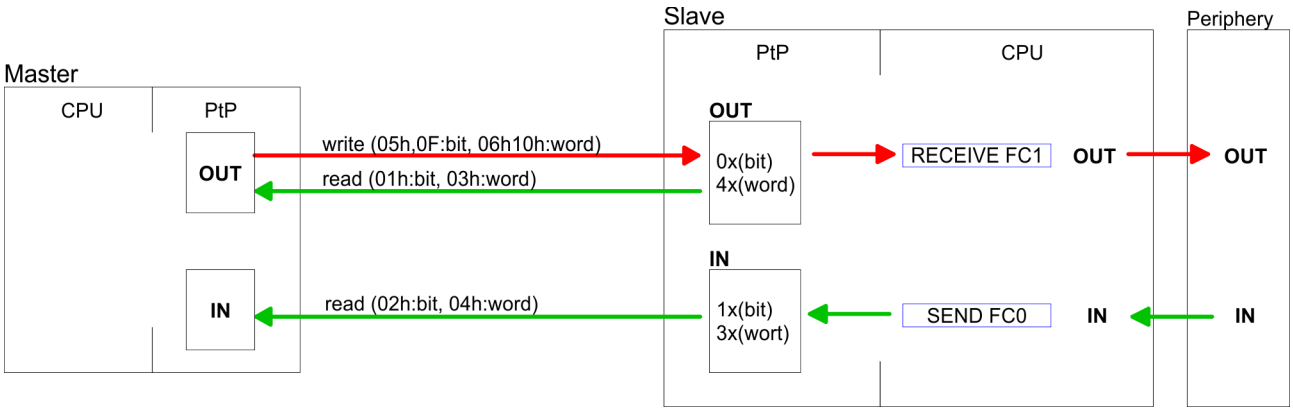
Overview

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

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

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

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



Respond of the slave

If the slave announces an error, the function code is send back with an "ORed" 80h.

Without an error, the function code is sent back.

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

Byte sequence in a word

1 word	
High-byte	Low-byte

Check sum CRC, RTU, LRC

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

Read n bits 01h, 02h

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

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

Command telegram

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

Respond telegram

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

Read n words 03h, 04h

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

04h: Read n words master input area 3x

Command telegram

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

Respond telegram

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

Write 1 bit 05h

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

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

"Status bit" = 0000h → Bit = 0

"Status bit" = FF00h → Bit = 1

Command telegram

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

Respond telegram

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

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

Command telegram

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

Respond telegram

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

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

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

Command telegram

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

Respond telegram

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

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

Command telegram

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

Respond telegram

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

6.8 Modbus - Example communication

Overview

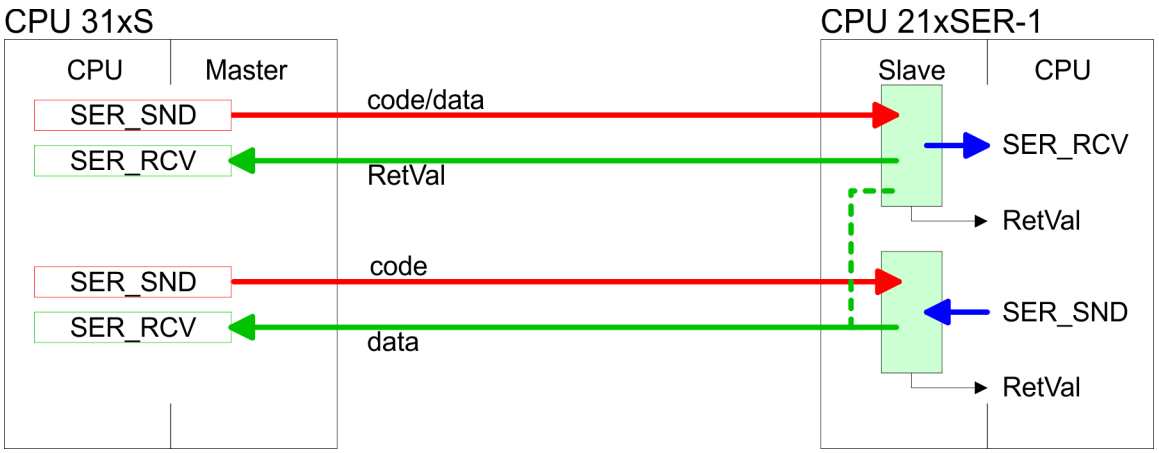
The example establishes a communication between a master and a slave via Modbus. The following combination options are shown:

- CPU 31xS as Modbus RTU master
- CPU 21xSER-1 as Modbus RTU slave
- Siemens SIMATIC Manager and possibilities for the project transfer
- Modbus cable connection

Approach

1. Assemble a Modbus system consisting of a CPU 31xS as Modbus master and a CPU 21xSER-1 as Modbus slave and Modbus cable.
2. Execute the project engineering of the master! For this you create a PLC user application with the following structure:
   - OB 100:
     Call SFC 216 (configuration as Modbus RTU master) with timeout setting and error evaluation.
   - OB 1:
     Call SFC 217 (SER_SND) where the data is send with error evaluation. Here you have to build up the telegram according to the Modbus rules. Call SFC 218 (SER_RECV) where the data is received with error evaluation.
3. Execute the project engineering of the slave! The PLC user application at the slave has the following structure:
   - OB 100:
     Call SFC 216 (configuration as Modbus RTU slave) with timeout setting and Modbus address in the DB and error evaluation.
   - OB 1:
     Call SFC 217 (SER_SND) for data transport from the slave CPU to the output buffer. Call SFC 218 (SER_RECV) for the data transport from the input buffer to the CPU. Allow an according error evaluation for both directions.

Structure for the according PLC programs for master and slave:



OB100:
Start
SFC 216
SER_CFG
(DB: Timeout)
RetVal 8xxxh ?
Y
Error evaluation
N
OB1:
SFC 217
SER_SND
(with slave no., code, range)
RetVal 700xh?
Y
N
RetVal 8xxxh / 90xxh ?
Y
Error evaluation
N
RetVal 2001h ?
Y
N
RetVal 2000h ?
Y
SFC 218
SER_RCV
N
RetVal 0000h ?
Y
N
SER_SND
RetVal 2001h ?
N
Data evaluation
J
RetVal 8xxxh ?
Y
Error evaluation
N
Start
SFC 216
SER_CFG
(DB: Timeout, slave address)
RetVal 8xxxh ?
Y
Error evaluation
N
SFC 217
SER_SND
RetVal 9001h ?
Y
Error evaluation
N
RetVal 0000h ?
Y
N
SFC 218
SER_RCV
RetVal 0000h ?
Y
Data evaluation
N
RetVal 8xxxh ?
Y
Error evaluation
N

7 Deployment PROFIBUS communication

7.1 Overview

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. Therefore 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 res. overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the ER-LED is on and 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.

7.2 Fast introduction

Overview

The PROFIBUS DP master is to be configured in the hardware configurator. Here the configuration happens by means of the sub module *X2 (DP)* of the Siemens CPU.

Steps of configuration

For the configuration of the PROFIBUS DP master please follow the following approach:

- **Hardware configuration - CPU**
- **Deployment as DP master** or **Deployment as DP slave**
- **Transfer of the complete project to CPU** *Chap. 5.10 'Project transfer' page 51*

With the Siemens SIMATIC Manager the CPU 315-4NE23 from VIPA is to be configured as

CPU 317-2DP (6ES7 317-2AK14 V3.3).

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X2 (DP). The Ethernet PG/OP channel of the 315-4NE23 is always to be configured as 1. module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens.

The CP 343 of the CPU is always to be configured below the before configured PG/OP channel also as CP343-1 (343-1EX11).

7.3 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. 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.

*Please consider that this SPEED7-CPU has 4 ACCUs. After an arithmetic operation (+I, -I, *I, /I, +D, -D, *D, /D, MOD, +R, -R, *R, /R) the content of ACCU 3 and ACCU 4 is loaded into ACCU 3 and 2. This may cause conflicts in applications that presume an unmodified ACCU 2.*

For more information may be found in the manual "VIPA Operation list SPEED7" at "Differences between SPEED7 and 300V programming".

Proceeding

Slot	Module
1	
2	**CPU 317-2DP**
X1	*MPI/DP*
X2	*DP*
3	

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 Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3).
4. The integrated PROFIBUS DP master (jack X3) is to be configured and connected via the sub module *"X2 (DP)"*.

7.3.1 Switching CPU type to CPU 318-2AJ00

Overview

To use projects, which were configured with the Siemens CPU type 318-2AJ00, you can switch from original CPU type to CPU type 318-2AJ00 by means of a CMD auto command. The setting is retained even after power cycle, firmware update or battery failure. With reset to factory settings respectively with the corresponding CMD auto command the CPU type is reset to the original CPU type.

Switching

- CPU type 318
  - Switching takes place with the CMD auto command *CPUTYPE_318*. After this perform a power cycle.
  - *Chap. 5.18 'CMD - auto commands' page 68*

```
CMD_START
CPUTYPE_318
CMD_END
```

- CPU type original
  - The reset to the original type takes place with the CMD auto command *CPUTYPE_ORIGINAL* respectively by *Chap. 5.15 'Reset to factory settings' page 65*.
  - *Chap. 5.18 'CMD - auto commands' page 68*

```
CMD_START
CPUTYPE_ORIGINAL
CMD_END
```

7.4 Deployment as PROFIBUS DP master

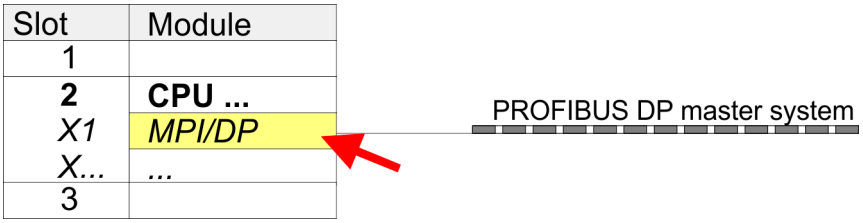
Precondition

The hardware configuration described before was established.

Proceeding

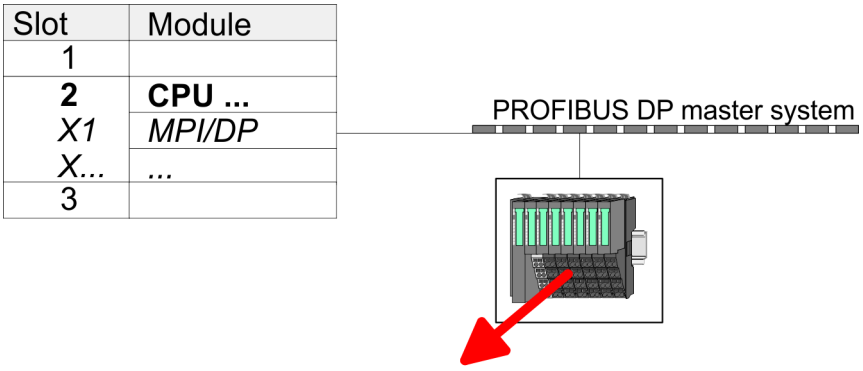
1. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
2. Set Interface type to "PROFIBUS"
3. Connect to PROFIBUS and preset an address (preferably 2) and confirm with [OK].
4. Switch at Operating mode to "DP master" and confirm the dialog with [OK]. A PROFIBUS DP master system is inserted.

   ⇨ 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, parameterize the modules.
5. Save, compile and transfer your project.



Slot	Module	Order number	
1	...		
2	Module		
3	...		
4			
5			
...			

7.5 Deployment as PROFIBUS DP slave

Fast introduction

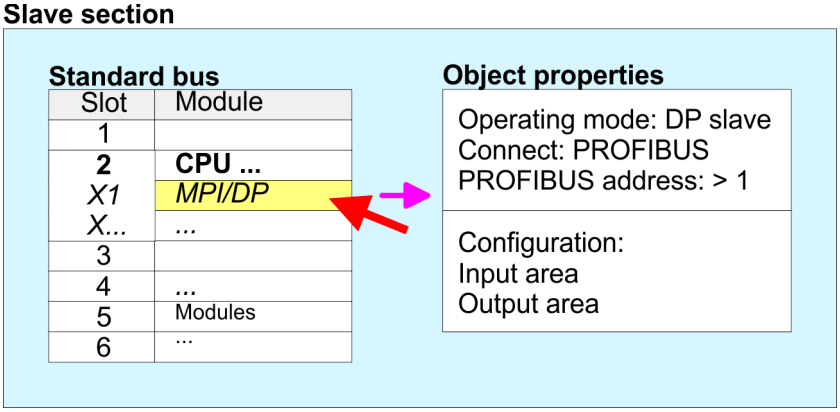
In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

1. Configure a station with a CPU with operating mode DP slave.
2. Connect to PROFIBUS and configure the in-/output area for the slave section.
3. Save and compile your project.
4. Configure another station with another CPU with operating mode DP master.
5. Connect to PROFIBUS and configure the in-/output ranges for the master section.
6. Save, compile and transfer your project to your CPU.

Project engineering of the slave section

1. Start the Siemens SIMATIC Manager and configure a CPU as described at "Hardware configuration - CPU".
2. Designate the station as "...DP slave".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
5. Set Interface type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 3) and confirm with [OK].
7. Switch at Operating mode to "DP slave" .
8. Via Configuration you define the in-/output address area of the slave CPU, which are to be assigned to the DP slave.
9. Save, compile and transfer your project to your CPU.

Slave section



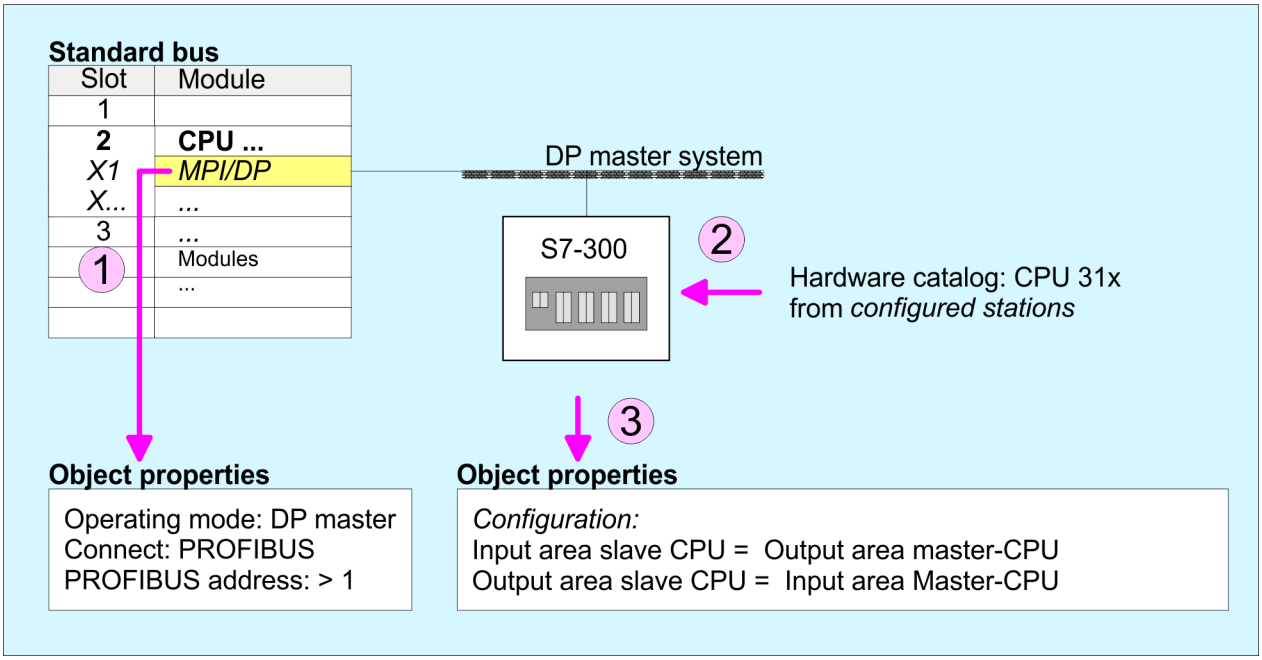
Project engineering master section

DP master and DP slave are in the same project

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



DP master and DP slave are in different projects

1. Create a new project, add a 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 *'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. For further configuration, install the GSD file from the appropriately configured Siemens slave CPU.
9. Choose via *'Additional field devices ➔ PLC ➔ SIMATIC'* the Siemens slave CPU.
10. Connect your slave system to the master system by dragging the slave CPU via PROFIBUS onto the master system.
11. Via the slots configure the I/O area of your slave system.
12. Save, compile and transfer your project to your CPU.

7.6 PROFIBUS installation guidelines

PROFIBUS in general

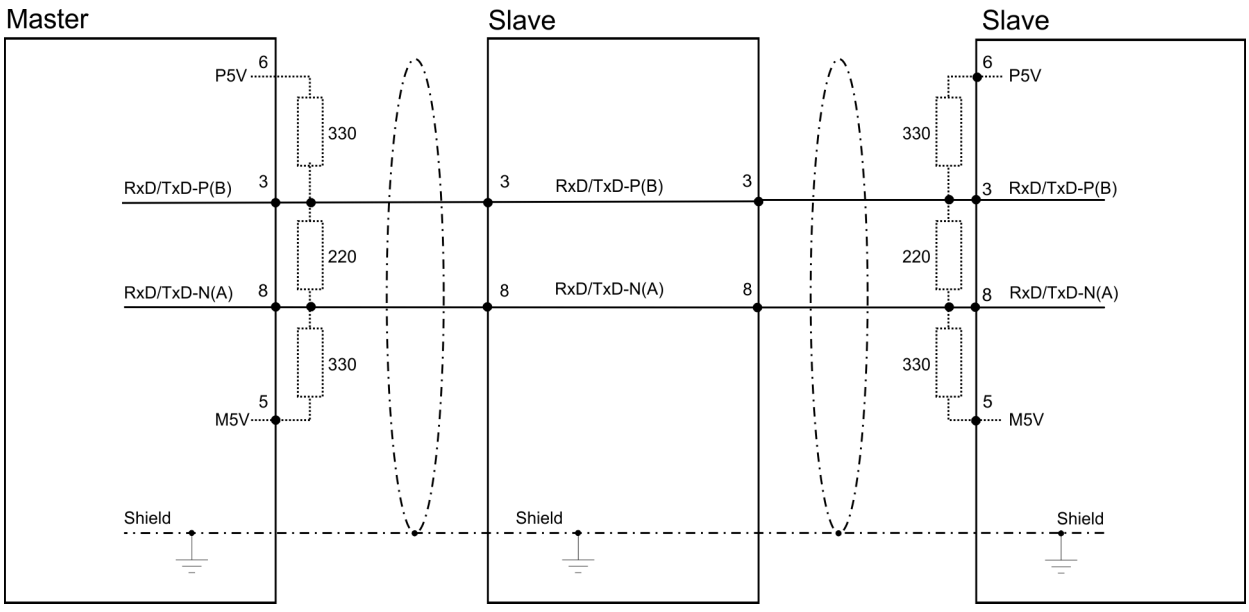
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- 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

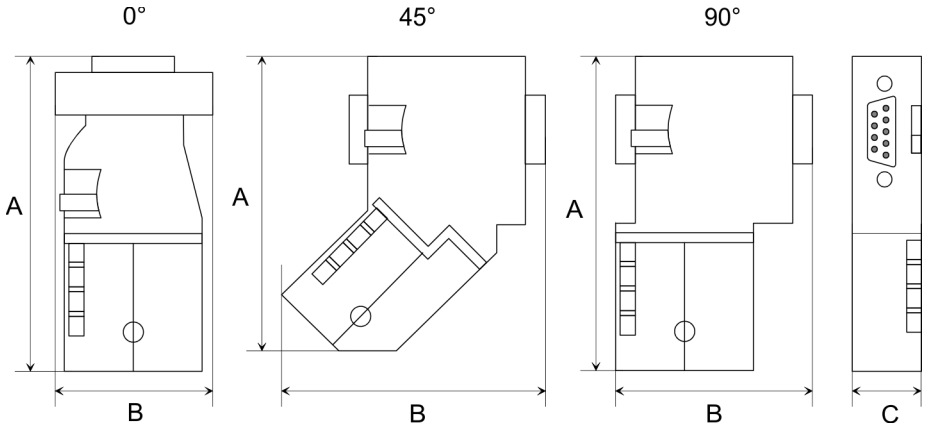
The following picture illustrates the terminating resistors of the respective start and end station.



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 "Easy-Conn" from Yaskawa. 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.

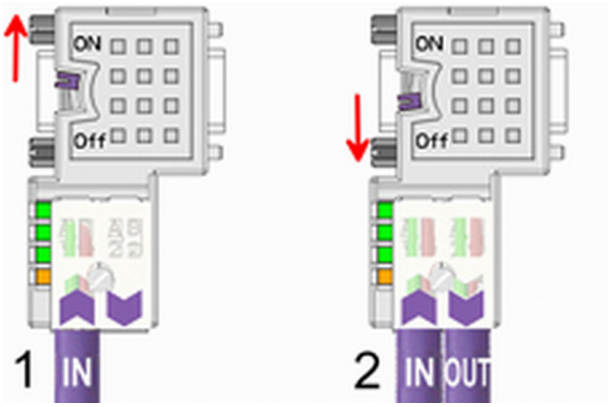
With the order no. 905-6AA00 Yaskawa offers the "EasyStrip" de-isolating tool that makes the connection of the EasyConn much easier.

Dimensions in mm

Termination with "Easy-Conn"

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

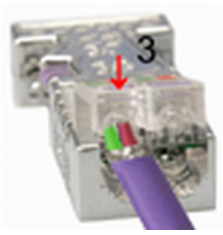
CAUTION!
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

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!

7.7 Commissioning and Start-up behavior

Start-up on delivery

In delivery the CPU is overall reset. The PROFIBUS part is deactivated and its LEDs are off after Power ON.

Online with bus parameter without slave project

The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via PROFIBUS by means of his PROFIBUS address. In this state the CPU can be accessed via PROFIBUS to get configuration and DP slave project.

Slave configuration

If the master has received valid configuration data, he switches to *Data Exchange* with the DP slaves. This is indicated by the DE-LED.

CPU state controls DP master

After PowerON respectively a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master. Dependent on the CPU state the following behavior is shown by the DP master:

- Master behavior at CPU STOP
  - The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.
  - DP slaves with fail safe mode were provided with output telegram length "0".
  - DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.
  - The input data of the DP slaves were further cyclically transferred to the input area of the CPU.
- Master behavior at CPU RUN
  - The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is on.
  - Every connected DP slave is cyclically attended with an output telegram containing recent output data.
  - The input data of the DP slaves were cyclically transferred to the input area of the CPU.

Adjusting the "Watchdog" time

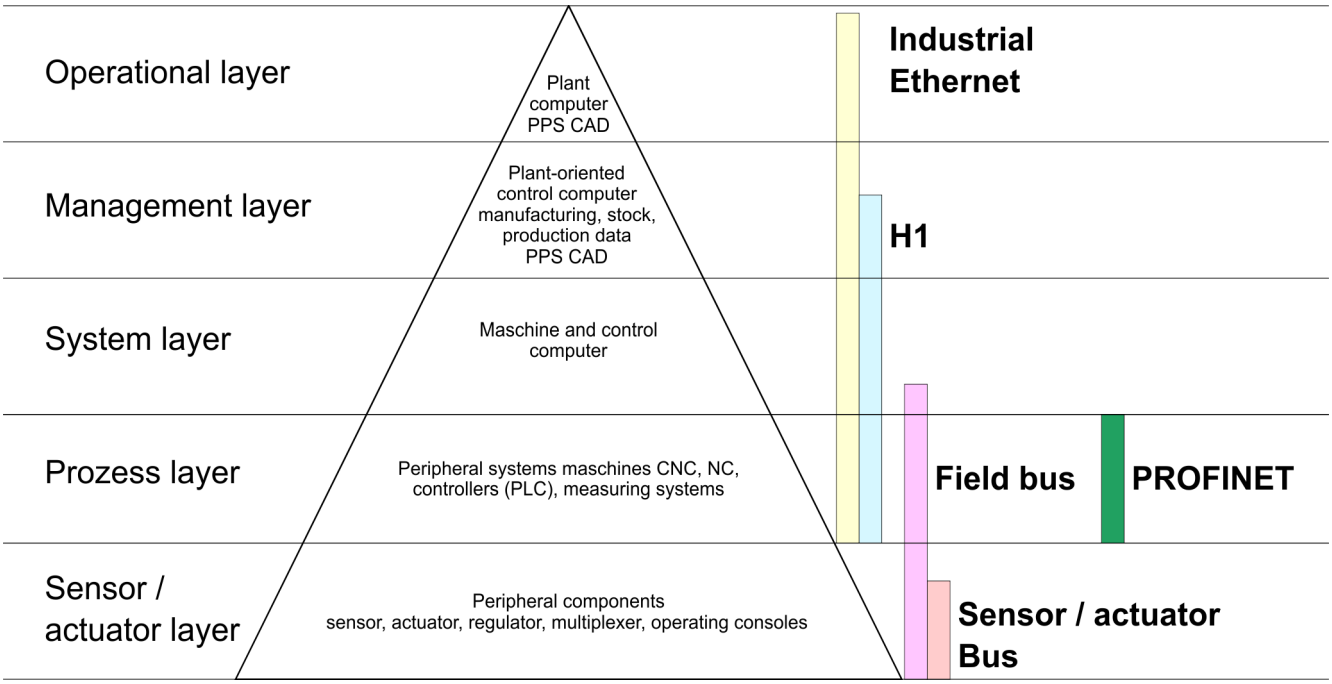
Due to the system the calculation of the bus rotation time in the Siemens SIMATIC Manager differs from the real bus rotation time of a Yaskawa DP master. For this reason, with many DP slaves and on a high transfer rate, the watchdog time should accordingly be adjusted. Especially on error in the PROFIBUS communication, with transfer rates up to 1.5Mbit/s, you should increase the watchdog time by factor 3 and with higher transfer rates (6Mbit/s respectively 12Mbit/s) by factor 6.

8 Deployment Ethernet communication - productive

8.1 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.

8.2 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.

8.3 Basics - Terms

Network (LAN)

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

Twisted Pair

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

Hub (repeater)

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

Switch

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

8.4 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. 8.2 'Basics - ISO/OSI reference model' page 99*

The following protocols are used:

- Communication connections
  - Siemens S7 connections
  - TCP/IP
  - UDP
  - RFC1006 (ISO-ON-TCP)
  - ISO transport (once H1)
- 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 Yaskawa 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 Yaskawa.

TCP/IP

TCP/IP protocols are available on all major systems. At the bottom end this applies to simple PCs, through to the typical mini-computer up to mainframes. For the wide spread of Internet accesses and connections, TCP/IP is often used to assemble heterogeneous system pools. TCP/IP (**T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol) collects a various range of protocols and functions. TCP and IP are only two of the protocols required for the assembly of a complete architecture.

- TCP/IP
  - The application layer provides programs like "FTP" and "Telnet" for the PC. The application layer of the Ethernet CP is defined with the user application using the standard handling blocks. These user applications use the transport layer with the protocols TCP or UDP for the data transfer which themselves communicate via the IP protocol with the Internet layer.
  - Besides of the IP address ports are used for the addressing. A port address should be within the range of 2000...65535.
  - Not depending on the used protocol, the PLC needs the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) for data transfer.
- IP
  - IP covers the network layer (Layer 3) of the ISO/OSI layer model.
  - The purpose of IP is to send data packages from on PC to another passing several other PCs. These data packages are referred to as datagrams. The IP doesn't neither guarantee the correct sequence of the datagrams nor the delivery at the receiver.
  - For the unambiguous identification between sender and receiver 32Bit addresses (IP addresses) are used that are normally written as four octets (exactly 8bit), e.g. 172.16.192.11. These Internet addresses are defined and assigned worldwide, thus every user may communicate with all other TCP/IP users.
  - One part of the address specifies the network; the rest serves the identification of the participants inside the network. The boarder between the network and the host part is variable and depends on the size of the network.
  - To save IP addresses, so called NAT router are used that have one official IP address and cover the network. Then the network can use any IP address.
- TCP
  - TCP bases directly on the IP and thus covers the transport layer (layer 4) of the ISO/OSI layer model.
  - TCP is a connection orientated end-to-end protocol and serves the logic connection between two partners.
  - TCP guarantees the correct sequence and reliability of the data transfer. Therefore you need a relatively large protocol overhead that slows down the transfer speed
  - Every datagram gets a header of at least 20Byte. This header also contains a sequence number identifying the series. This has the consequence that the single datagrams may reach the destination on different ways through the network.
  - Using TCP connections, the telegram length is not transmitted. This means that the recipient has to know how many bytes belong to a message.
  - To transfer data with variable length you may begin the user data with the length information and evaluate this at the counter station.

UDP

UDP (**U**ser **D**atagramm **P**rotocol) is a connection free transport protocol. It has been defined in the RFC768 (**R**equest **f**or **C**omment). Compared to TCP, it has much fewer characteristics. The addressing happens via port numbers. UDP is a fast unsafe protocol for it doesn't neither care about missing data packages nor about their sequence.

ISO-on-TCP RFC1006

The TCP transport service works stream orientated. This means that data packages assembled by the user not necessarily have to receive the partner in the same packaging. Depending on the data amount, packages may though come in the correct

sequence but differently packed. This causes that the recipient may not recognize the package borders anymore. For example you may send 2x 10Byte packages but the counter station receives them as 20Byte package. But for most of the applications the correct packaging is important. Due to this you need another protocol above TCP. This purpose is defined in the protocol extension RFC1006 (ISO-on-TCP).

- RFC1006 describes the function of an ISO transport interface (ISO 8072) basing upon the transport interface TCP (RFC793).
- The basic protocol of RFC1006 is nearly identical to TP0 (Transport Protocol, Class 0) in ISO 8073.
- For RFC1006 is run as protocol for TCP, the decoding takes place in the data section of the TCP package.
- In contrast to TCP here the receipt of a telegram is confirmed.
- Instead of ports TSAPs are used for the addressing besides of the IP address. The TSAP length may be 1 ... 16 characters. The entry may happen in ASCII or Hex format.
- Not depending on the used protocol, the PLC needs the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) for data transfer.
- Contrary to TCP different telegram lengths can be received using RFC1006.

ISO transport (once H1)

With ISO transport connections there is the possibility for program and event controlled communication via Industrial Ethernet. Here data blocks may be exchanged bi-directional. The ISO transport connection offers services for a safety transfer of data by means of configured connections. Large data blocks may be transferred by means of blocking. The transmission reliability is very high by the automatic repetition, by additional block test mechanisms and by the receipt acknowledgement at the receiver side.

- The ISO transport service (ISO 8073 class 4) corresponds to the transport layer (Layer 4) of the ISO/OSI reference model.
- ISO transport connections are exclusively transferred via Industrial Ethernet and they are optimized for the deployment in a closed manufacturing area.
- The receipt of data is acknowledged by the partner station. Here different telegram lengths may be processed.
- For deployment of the ISO transport connections they must be enabled in the Ethernet properties of the CP within your project. Here there is the possibility to assign the CP to a MAC address.
- The addressing happens by MAC address (Ethernet address) and TSAPs (**T**ransport **S**ervice **A**ccess **P**oint).
- The data transfer is made by the services SEND/RECEIVE and FETCH/WRITE.
- Not depending on the used protocol, the PLC needs the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) for data transfer.

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.

8.5 Basics - IP address and subnet

IP address structure

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

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

Range: 000.000.000.000 to 255.255.255.255

Net-ID, Host-ID

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

Subnet mask

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

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

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

Address at first start-up

At the first start-up of the CPU, the Ethernet PG/OP channel and the CP 343 do not have an IP address.

Information about the assignment of IP address data to the Ethernet PG/OP channel may be found in *Chap. 5.6 'Hardware configuration - Ethernet PG/OP channel' page 41*.

Information about the assignment of IP address data to the CP 343 may be found in *Chap. 8.8 'Commissioning and initialization' page 109*

Address classes

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

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

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

Private IP networks

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

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

Reserved Host-IDs

Some Host-IDs are reserved for special purposes.

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

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

8.6 Basics - MAC address and TSAP

MAC address

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

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

TSAP

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

Address parameters

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

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

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

8.7 Fast introduction

Overview

At the first start-up respectively at an over all reset with an PowerON again, the Ethernet PG/OP channel and CP 343 do not have any IP address. These may only be reached via its MAC address. IP address parameters may be assigned to the corresponding component by means of the MAC addresses, which may be found on labels beneath the front flap with the sequence 1. address PG/OP channel and beneath address of the CP 343. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

Steps of configuration

For the configuration of the CP 343 for productive connections please follow the following approach:

- Assembly and commissioning
- Hardware configuration - CPU
- Configure connections
  - Communication 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

With the Siemens SIMATIC Manager the CPU 315-4NE23 from VIPA is to be configured as

CPU 317-2DP (6ES7 317-2AK14 V3.3).

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X2 (DP). The Ethernet PG/OP channel of the 315-4NE23 is always to be configured as 1. module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens.

The CP 343 of the CPU is always to be configured below the before configured PG/OP channel also as CP343-1 (343-1EX11).

8.8 Commissioning and initialization

Assembly and commissioning

1. Install your System 300S with your CPU.
2. Wire the system by connecting cables for voltage supply and signals
3. Connect your CP 343 with Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time, the CP is in idle. At the first commissioning res. after an overall reset of the CPU, the CP 343 and the Ethernet PG/OP channel have no IP address.

Assign IP address parameters

Please regard this function is available with CP firmware version 1.7.4 and up.

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.3 & SP3 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 2. 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. 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
6. Confirm with [Assign IP configuration].

Directly after the assignment the CP 343 is online reachable using the set IP address data.

Since the IP address data, which were assigned here, are deleted at PowerOFF, you have to take them to a project by means of the hardware configuration.

8.9 Hardware configuration - CP 343

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. 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

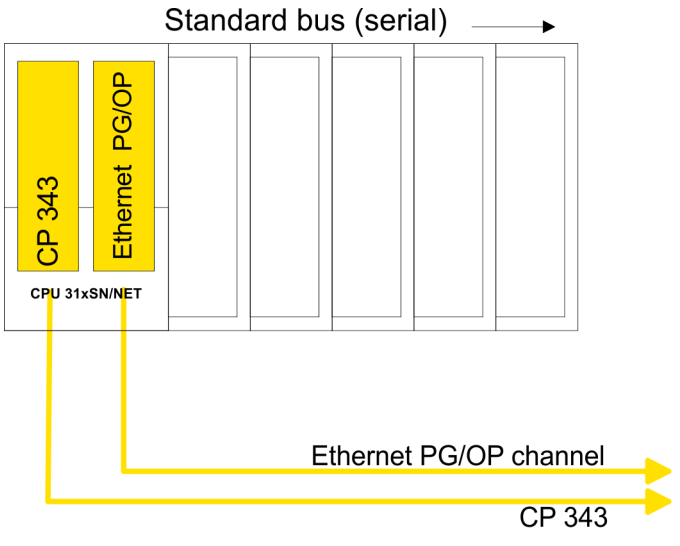
Slot	Module
1	
2	**CPU 317-2DP**
X1	*MPI/DP*
X2	*DP*
3	

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 Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3).
4. The integrated PROFIBUS DP master (jack X3) is to be configured and connected via the sub module *"X2 (DP)"*.

Project engineering Ethernet PG/OP channel and CP 343

1. Place for the internal Ethernet PG/OP channel always as 1. module below the really plugged modules a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX11 0XE0).
2. Configure the integrated CP 343 of the CPU also as CP 343-1 (343-1EX11) but always below the before configured CP 343-1.



Standard bus

Slot	Module
1	
2	**CPU ...**
X...	
3	
4	...
5	...
6	Modules
7	...
8	...
9	343-1EX11
10	343-1EX11
11	

3. Open the property window of the CP via double-click on the CP.
4. Enter at *'General'* a device name. The device name must be unique at the Ethernet subnet.
5. On [Properties] for the CP enter the IP address, subnet mask and gateway and select the wanted subnet.

8.9.1 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.

Please note that the integrated CP only supports the functions described in this manual, even if the Siemens CP used for the configuration offers further functions! Since NTP is not supported by the Siemens CP 343-1EX11, the Siemens CP 343-1EX21 must be used to configure the NTP method.

Slot	Module
1	
2 *X...*	**CPU 31...**
9	
10	343-1EX21
11	
...	

1. Configure the Siemens CP 343-1EX21 for the integrated Yaskawa CP 343.
2. Open the properties dialog via double-click on the CP 343-1EX21.
3. Select the tab *'Time-of-day synchronization'*.
4. Activate the NTP method by enabling *'Activate NTP time-of-day synchronization'*.
5. Click at [Add] and add the corresponding NTP server.
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. 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.

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 Yaskawa.

When used with the Siemens SIMATIC Manager, there are the following limitations:

- The parameter *'Time-of-day synchronization on the full minute'* is not supported.
- The parameter *'Accept time-of-day from synchronized NTP servers'* is not supported.
- The parameter *'Forwarding time-of-day to station'* is not supported.

8.10 Configure connections

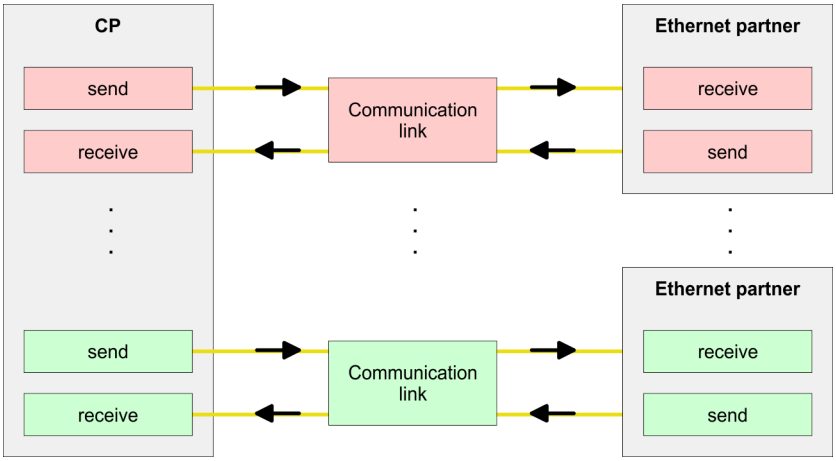
8.10.1 Overview

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 Yaskawa handling blocks. To use this blocks, configured communication connections are always necessary in the active station.

Properties communication connection

The following properties are characterizing a communication connection:

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



Requirements

- Siemens SIMATIC Manager V 5.3 SP3 or higher and SIMATIC NET are installed.
- With the hardware configuration the CP was assigned with IP address data by the properties.

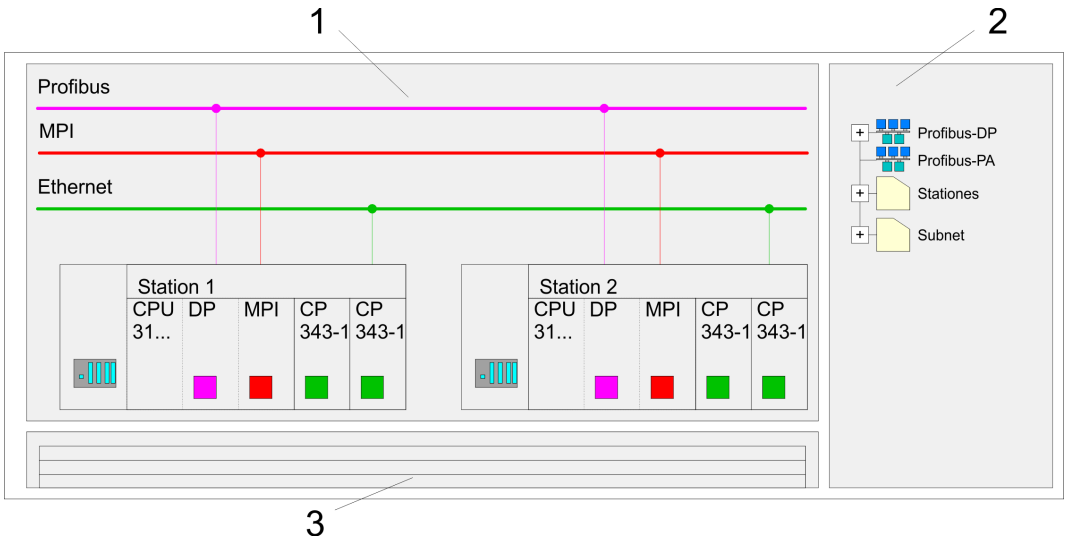
Every station outside of the recent project must be configured as replacement objects like e.g. Siemens "SIMATIC S5" or "other station" or with the object "In unknown project". When creating a connection you may also choose the partner type "unspecified" and set the required remote parameter directly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

8.10.2 Siemens NetPro

Work environment of NetPro

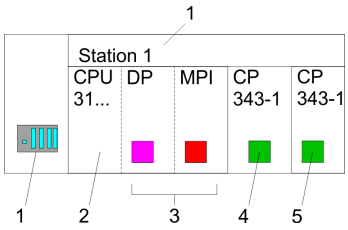
For the project engineering of connections, a thorough knowledge with NetPro from Siemens is required! The following passage only describes the basic usage of NetPro. More detailed information about NetPro is to be found in the according online manual res. documentation. Start NetPro by clicking on a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

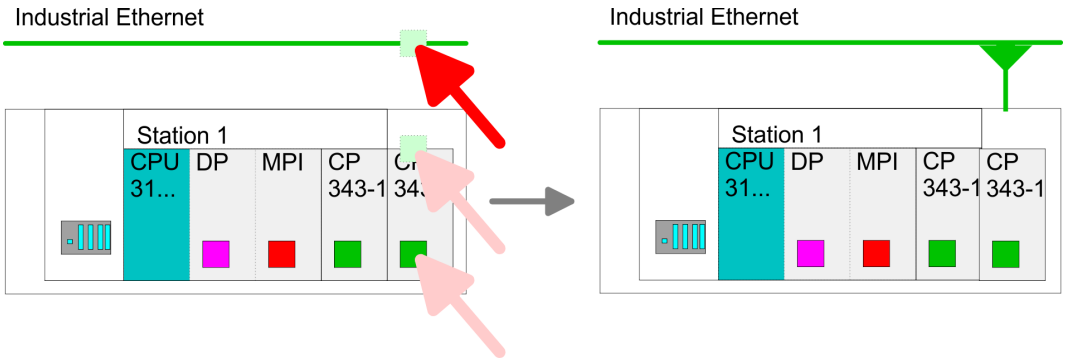


You receive the following graphical display for every PLC station and their component. By selecting the single components, the context menu offers you several functions:

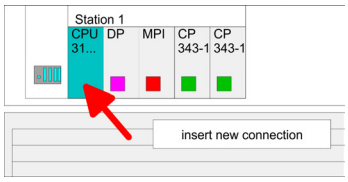
1 *Station:* This includes a PLC station with rack, CPU and communication components. Via the context menu you may configure a station added from the net objects and its concerning components in the hardware configurator. After returning to NetPro, the new configured components are shown.
2 *CPU:* A 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 shows the communication components that are available in your CPU. For the NET-CPU is configured as Siemens CPU, the internal components do not show the CP. Due to this, the internal CPs of the NET-CPU must be configured as external CPs behind the really plugged modules. The CPs are then also shown in NetPro as external CPs (4, 5) in the station.
4 *Ethernet PG/OP channel:* The internal Ethernet PG/OP channel must always be configured as external CP in the hardware configuration. This CP only serves the PG/OP communication. Configurable connections are not possible.
5 The internal CP 343 must always be configured as external 2. CP in the hardware configuration after the Ethernet PG/OP channel.

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

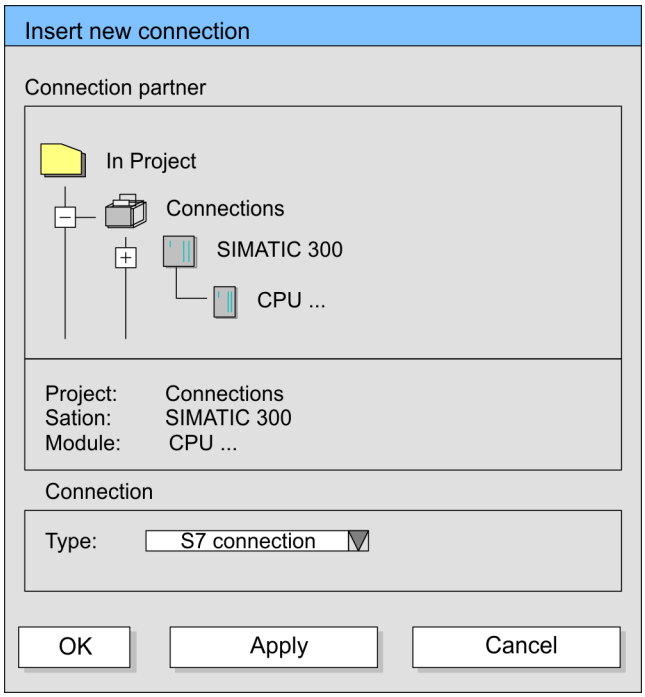


1. For the project engineering of connections, open the connection list by selecting the according CPU. Open *'Context menu ➔ Insert new connection'*:

- 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*.
- All broadcast stations
  - Exclusive at UDP connections you may send to every reachable participant.
  - The receipt of user data is not possible.
  - The broadcast participants are specified by one port and one broadcast address at sender and receiver.
  - Per default, broadcasts that are only serving the Ethernet communication, like e.g. ARP-Requests (Search MAC <> IP address), are received and accordingly processed.
  - For the identification of the broadcast participants within the net, you have to define a valid broadcast address as partner IP during project engineering of a broadcast connection.
  - Additionally to the broadcast address you have to set a common port for sender and receiver.
- All multicast stations
  - By selecting *'All Multicast stations'* you define that UDP telegrams have to be sent respectively received by all participants of a multicast group.
  - In contrast to broadcast here a reception is possible.
  - For the identification of the multicast participants within the net, you have to define one valid multicast group address and one port for sender and receiver. The maximum number of multicast circles, which are supported by the CP, is identical to the maximum number of connections.
- Connection types

  The following connection types are available for communication:
  - *Siemens S7 connections*, *Send/Receive connections* (TCP, ISO-on-TCP and ISO transport) for secured data transfer of data blocks between two Ethernet stations.
  - *UDP* for not secured data transfer of data blocks between two Ethernet stations.

2. Select 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.

8.10.3 Connection type - S7

Siemens S7 connection

- For data transfer with Siemens S7 connections the FB/SFB Yaskawa 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 Yaskawa 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 Yaskawa 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 Yaskawa 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 Yaskawa 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	Sending data in blocks: FB/SFB 12 BSEND sends data to a remote partner FB/SFB of the type BRCV (FB/SFB 13). The data area to be transmitted is segmented. Each segment is sent individually to the partner. The last segment is acknowledged by the partner as it is received, independently of the calling up of the corresponding FB/SFB/FB BRCV. 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.
FB/SFB 13	BRCV	Receiving data in blocks: The FB/SFB 13 BRCV can receive data from a remote partner FB/SFB of the type BSEND (FB/SFB 12). The parameter R_ID of both FB/SFBs must be identical. After each received data segment an acknowledgement is sent to the partner FB/SFB and the LEN parameter is updated.
FB/SFB 14	GET	Remote CPU read: The FB/SFB 14 GET can be used to read data from a remote CPU. The respective CPU must be in RUN mode or in STOP mode.
FB/SFB 15	PUT	Remote CPU write: The FB/SFB 15 PUT can be used to write data to a remote CPU. The respective CPU may be in RUN mode or in STOP mode.

8.10.4 Connection type - Send/Receive

Send/Receive connections

At the PLC side for data transfer with these connections the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) are to be used.

Send/receive connections are the following connections:

- TCP (SEND-RECEIVE, FETCH-WRITE PASSIVE)
- ISO-on-TCP (SEND-RECEIVE, FETCH-WRITE PASSIVE)
- ISO-on-TCP (SEND-RECEIVE, FETCH-WRITE PASSIVE)
- UDP (SEND-RECEIVE)

The following parameters define a connection end point:

Station A				Station B
remote port:	→	TCP connection	→	local port
local port	←		←	remote port
IP address A				IP address B

Station A				Station B
remote TSAP	→	ISO TCP connection	→	local TSAP
local TSAP	←		←	remote TSAP
IP address A				IP address B

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

Station A				Station B
remote port	→	UDP connection	→	local port
local port	←		←	remote port
IP address A				IP address B

Combination options with the different operating modes

Connection partner	Connection type	Connection establishment	Connection	Operating mode
specified in NetPro (in current project)	TCP / ISO-on-TCP / ISO-Transport	active/passive	specified	SEND/RECEIVE
	UDP	-		
unspecified in NetPro (in current project)	TCP / ISO-on-TCP / ISO-Transport	active	specified	SEND/RECEIVE
		passive	part specified (Port/TSAP)	SEND/RECEIVE FETCH PASSIVE
			unspecified	WRITE PASSIVE
	UDP	-	specified	SEND/RECEIVE
unspecified in NetPro (in unknown project)	TCP / ISO-on-TCP / ISO-Transport	active	specified (connection name in an other project)	SEND/RECEIVE
		passive		SEND/RECEIVE FETCH PASSIVE WRITE PASSIVE
	UDP	-		SEND/RECEIVE

Connection partner	Connection type	Connection establishment	Connection	Operating mode
All broadcast stations	UDP	-	specified (Port, Broadcast addr.)	SEND
All multicast stations	UDP	-	specified (Port, Multicast group)	SEND/RECEIVE

In the following each relevant parameters of the different connection types are described:

- *General*
  In this tab the general connection parameters are listed, which identify the local connection end point.
  - *ID*
    This entry is identical to the entry of the connection table. The value may always be changed. Please also regard to adjust the ID parameter of the call interface of the FC.
  - *Name*
    This field contains the name of the connection. The name is generated by the system and may be changed on every time.
  - *Via CP [Route]*
    Here is the CP listed, which should be used for connection. With the button [Route] the appropriate CP may be selected for communication. Do not select the 1. CP of the route for communication connections. The 1. CP is always the Ethernet PG/OP channel, which does not support configurable connections.
  - *Active connection establishment*
    If activated the connection to the partner is active established by the local station. Here the partner is to be specified in the tab *'Addresses'*. At an unspecified connection the connection is passive established.
- *Block parameters*
  - Here the parameters *ID* and *LADDR* for your user program are shown. Both are parameters, which are to be preset if you use the FC 5 and FC 6 (AG_SEND, AG_RECEIVE). Please always use the VIPA FCs, which you may receive from VIPA.
- *Addresses*
  The Addresses tab displays the relevant local and remote address information as proposed values. Depending on the kind of communication the address information may remain unspecified.
  - *Port*
    Ports respectively port addresses are defining the access point to the user application within the station/CPU. These must be unique. A port address should be within the range of 2000...65535.
  - *TSAP*
    ISO-on-TCP and ISO transport support TSAP lengths (**T**ransport **S**ervice **A**ccess **P**oint) of 1...16byte. You may enter the TSAP in ASCII or hexadecimal format. The calculation of the length happens automatically.

- *Options*
  Depending on the specification of the connecting partner the following *operating modes* may be set respectively displayed:
  – SEND/RECEIVE
    The SEND/RECEIVE interface allows the program-controlled communication to any partner station via a configured connection. Here the data transfer happens by a call from your user application. The FC5 and FC6 that are part of the VIPA block library are serving as interface. This enables your PLC to send messages depending on process events.
  – FETCH/WRITE PASSIVE
    With the help of FETCH/WRITE services partner systems have the direct access to memory areas of the CPU. These are "passive" communication connections that have to be configured. The connections are "actively" established by the connection partner.
  – FETCH PASSIVE (request data)
    FETCH allows a partner system to request data.
  – WRITE PASSIVE (write data)
    This allows a partner system to write data in the data area of the CPU.
- *Overview*
  Here every configured connections of this station and its partner are displayed. These data are information and may not be changed.

– *If a CP is exchanged by another one, this must at least provide the same services and must at least have the same version level. Only this can guarantee the connections configured via the CP to remain consistent and useable.*
– *By appropriate shift respectively delete activities in the Siemens SIMATIC manager connections may lose the allocation to the CP. These connections are marked with "!" at ID of the overview.*

8.10.4.1 FC 5 - AG_SEND / FC 6 - AG_RECV - CP 343 communication

Overview

The two blocks serve the processing of the Ethernet-CP 343 connection commands on the side of the PLC. Including these blocks in the cycle block OB1 you may send and receive data cyclically.

Within these blocks, the SFCs 205 and 206 are called that are stored as special function blocks in the CPU.

Please regard that you may only use the SEND/RECV-FCs from VIPA in your user application for the communication with VIPA-CPs. At a change to VIPA-CPs in an already existing project, the present AG_SEND / AG_LSEND res. AG_RECV / AG_LRECV may be replaced by AG_SEND res. AG_RECV from VIPA without adjustment. Due to the fact that the CP automatically adjusts itself to the length of the data to transfer, the L variant of SEND res. RECV is not required for VIPA CPs.

Communication blocks

For the communication between CPU and Ethernet-CP 343, the following FCs are available:

AG_SEND (FC 5)

This block transfers the user data from the data area given in *SEND* to the CP specified via *ID* and *LADDR*. As data area you may set a PA, bit memory or data block area. When the data area has been transferred without errors, "order ready without error" is returned.

AG_RECV (FC 6)

The block transfers the user data from the CP into a data area defined via *RECV*. As data area you may set a PA, bit memory or data block area. When the data area has been transferred without errors, "order ready without error" is returned.

Status displays

The CP processes send and receive commands independently from the CPU cycle and needs for this transfer time. The interface with the FC blocks to the user application is here synchronized by means of acknowledgements/receipts.

For status evaluation the communication blocks return parameters that may be evaluated directly in the user application.

These status displays are updated at every block call.

Deployment at high communication load

Do not use cyclic calls of the communication blocks in OB 1. This causes a permanent communication between CPU and CP. Program instead the communication blocks within a time OB where the cycle time is higher res. event controlled.

FC call is faster than CP transfer time

If a block is called a second time in the user application before the data of the last time is already completely send res. received, the FC block interface reacts like this:

AG_SEND

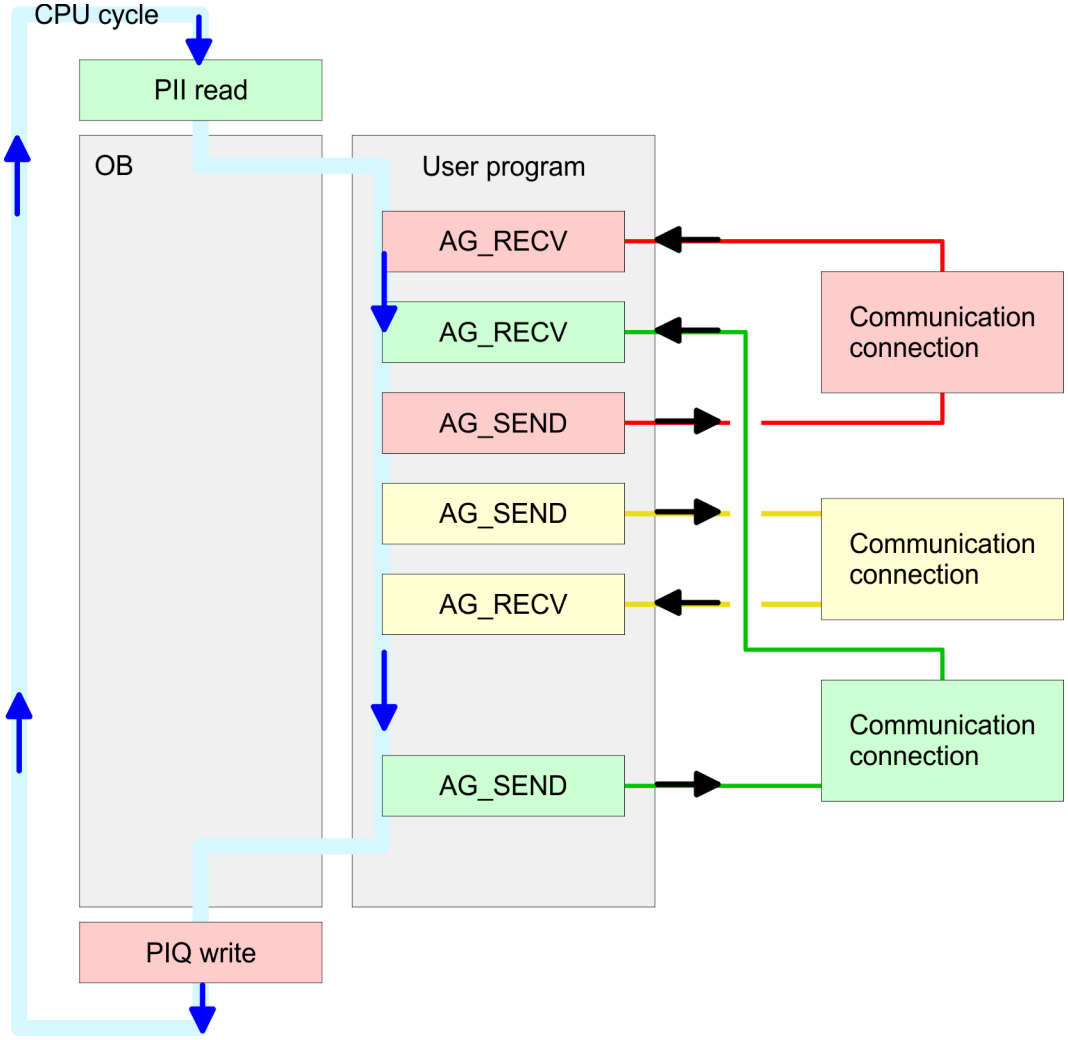
No command is accepted until the data transfer has been acknowledged from the partner via the connection. Until this you receive the message "Order running" before the CP is able to receive a new command for this connection.

AG_RECV

The order is acknowledged with the message "No data available yet" as long as the CP has not received the receive data completely.

AG_SEND, AG_RECV in the user application

The following illustration shows a possible sequence for the FC blocks together with the organizations and program blocks in the CPU cycle:



The FC blocks with concerning communication connection are summed up by colour. Here you may also see that your user application may consist of any number of blocks. This allows you to send or receive data (with AG_SEND res. AG_RECV) event or program driven at any wanted point within the CPU cycle. You may also call the blocks for **one** communication connection several times within one cycle.

AG_SEND (FC 5)

By means of AG_SEND the data to send are transferred from the CPU to an Ethernet CP 343.

Parameters

Parameter	Declaration	Data type	Description
ACT	INPUT	BOOL	Activation of the sender 0: Updates *DONE*, *ERROR* and *STATUS* 1: The data area defined in *SEND* with the length *LEN* is send
ID	INPUT	INT	Connection number 1 ... 16 (identical with *ID* of NetPro)
LADDR	INPUT	WORD	Logical basic address of the CP (identical with *LADDR* of NetPro)
SEND	INPUT	ANY	Data area

Parameter	Declaration	Data type	Description
LEN	INPUT	INT	Number of bytes from data area to transfer
DONE	OUTPUT	BOOL	Status parameter for the order 0: Order running 1: Order ready without error
ERROR	OUTPUT	BOOL	Error message 0: Order running (at *DONE* = 0) 0: Order ready without error (at *DONE* = 1) 1: Order ready with error
STATUS	OUTPUT	WORD	Status message returned with *DONE* and *ERROR*. More details are to be found in the following table.

AG_RECV (FC 6)

With the 1. call of AG_RECV a receive buffer for the communication between CPU and an Ethernet CP 343 is established. From now on received data are automatically stored in this buffer. As soon as after calling AG_RECV the return value of *NDR* = 1 is returned, valid data are present.

Since with a further call of AG_RECV the receive buffer is established again for the receipt of new data, you have to save the previous received data.

Parameters

Parameter	Declaration	Data type	Description
ID	INPUT	INT	Connection number 1 ... 16 (identical with *ID* of NetPro)
LADDR	INPUT	WORD	Logical basic address of the CP (identical with *LADDR* of NetPro)
RECV	INPUT	ANY	Data area for the received data.
NDR	OUTPUT	BOOL	Status parameter for the order 0: Order running 1: Order ready data received without error
ERROR	OUTPUT	BOOL	Error message 0: Order running (at *NDR* = 0) 0: Order ready without error (at *NDR* = 1) 1: Order ready with error
STATUS	OUTPUT	WORD	Status message returned with *NDR* and *ERROR*. More details are to be found in the following table.
LEN	OUTPUT	INT	Number of bytes that have been received

DONE, ERROR, STATUS

The following table shows all messages that can be returned by the Ethernet CP 343 after a SEND res. RECV command.

A "-" means that this message is not available for the concerning SEND res. RECV command.

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Description
1	-	0	0000h	Order ready without error.
-	1	0	0000h	New data received without error.
0	-	0	0000h	No order present.
-	0	0	8180h	No data available yet.
0	0	0	8181h	Order running
0	0	1	8183h	No CP project engineering for this order.
0	-	1	8184h	System error
-	0	1	8184h	System error (destination data area failure).
0	-	1	8185h	Parameter *LEN* exceeds source area *SEND*.
	0	1	8185h	Destination buffer (RECV) too small.
0	0	1	8186h	Parameter *ID* invalid (not within 1 ...16).
0	-	1	8302h	No receive resources at destination station, receive station is not able to process received data fast enough res. has no receive resources reserved.
0	-	1	8304h	The connection is not established. The send command shouldn't be sent again before a delay time of > 100ms.
-	0	1	8304h	The connection is not established. The receive command shouldn't be sent again after a delay time of > 100ms.
0	-	1	8311h	Destination station not available under the defined Ethernet address.
0	-	1	8312h	Ethernet error in the CP.
0		1	8F22h	Source area invalid, e.g. when area in DB not present Parameter *LEN* < 0
-	0	1	8F23h	Source area invalid, e.g. when area in DB not present Parameter *LEN* < 0
0	-	1	8F24h	Range error at reading a parameter.
-	0	1	8F25h	Range error at writing a parameter.
0	-	1	8F28h	Orientation error at reading a parameter.
-	0	1	8F29h	Orientation error at writing a parameter.
-	0	1	8F30h	Parameter is within write protected 1. recent data block
-	0	1	8F31h	Parameter is within write protected 2. recent data block
0	0	1	8F32h	Parameter contains oversized DB number.
0	0	1	8F33h	DB number error
0	0	1	8F3Ah	Area not loaded (DB)
0	-	1	8F42h	Acknowledgment delay at reading a parameter from peripheral area.
-	0	1	8F43h	Acknowledgment delay at writing a parameter from peripheral area.

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Description
0	-	1	8F44h	Address of the parameter to read locked in access track
-	0	1	8F45h	Address of the parameter to write locked in access track
0	0	1	8F7Fh	Internal error e.g. invalid ANY reference e.g. parameter *LEN* = 0.
0	0	1	8090h	Module with this module start address not present or CPU in STOP.
0	0	1	8091h	Module start address not within double word grid.
0	0	1	8092h	reference contains type setting unequal BYTE.
-	0	1	80A0h	Negative acknowledgment at reading the module.
0	0	1	80A4h	reserved
0	0	1	80B0h	Module doesn't recognize record set.
0	0	1	80B1h	The length setting (in parameter *LEN*) is invalid.
0	0	1	80B2h	reserved
0	0	1	80C0h	Record set not readable.
0	0	1	80C1h	The set record set is still in process.
0	0	1	80C2h	Order accumulation.
0	0	1	80C3h	The operating sources (memory) of the CPU are temporarily occupied.
0	0	1	80C4h	Communication error (occurs temporarily; a repetition in the user application is reasonable).
0	0	1	80D2h	Module start address is wrong.

Status parameter at reboot At a reboot of the CP, the output parameters are set as follows:

- DONE = 0
- NDR = 0
- ERROR = 0
- STATUS = 8180h (at AG_RECV)
  STATUS = 8181h (at AG_SEND)

8.11 Configure Open Communication

Connection-oriented protocols

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

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

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

Connection-less protocol

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

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

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

Handling blocks

Those in the following listed UTDs and FBs serve for "open communication" with other Ethernet capable communication partners via your user program. These blocks are part of the 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.

UDTs

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

FBs

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

8.12 NCM diagnostic - Help for error diagnostic

NCM diagnostics - Help for error diagnostics

This page shall help you with the error diagnostics. The following check list shows a number of typical problems and their probable causes:

Question	Remedy with "no"
CPU in RUN?	■ Check DC 24V power supply. ■ Set the operating mode switch to position RUN. ■ Check PLC program and transfer it again.
AG_SEND, AG_RECV in user application?	These 2 blocks are required in the user application for the data transfer between CP and CPU. Both blocks must also be called with a passive connection.
Is CP able to connect?	■ Check Ethernet cable (at a point-to-point connection a crossed Ethernet cable is to be used). ■ Check IP address.
Can data be transferred?	■ Check Port no. for read and write. ■ Check source and destination areas. ■ Check if the 2. CP is selected in the route. ■ Enlarge the receive respectively send buffer defined via the ANY pointer.
Is the complete data block sent at ISO-on-TCP?	■ Check the LEN parameter at AG_SEND. ■ Set the receive respectively send buffer defined via the ANY pointer to the required size.

Siemens NCM S7 diagnostic

The CP supports the Siemens NCM diagnostic tool. The NCM diagnostic tool is part of the Siemens SIMATIC manager. This tool delivers information about the operating state of the communication functions of the online CPs dynamically.

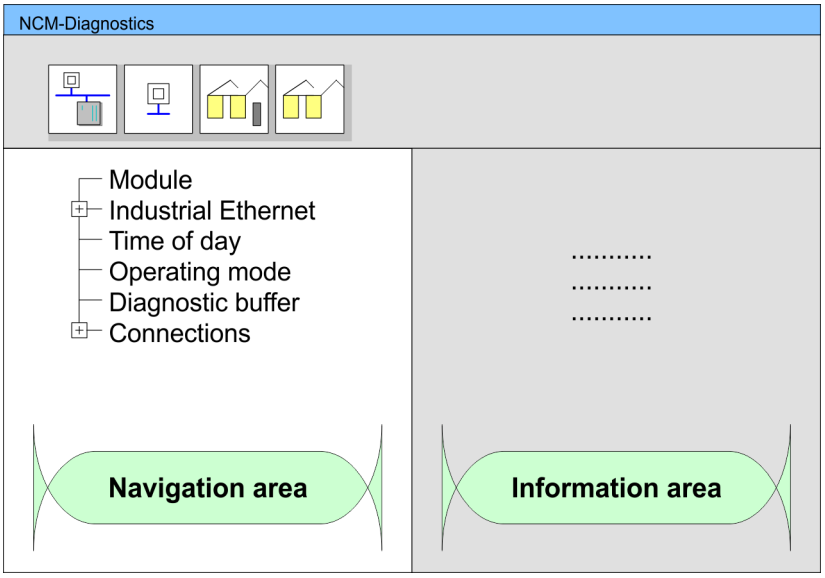
The following diagnostic functions are available:

- Check operating state at Ethernet
- Read the diagnostic buffer of the CP
- Diagnostics of connections

Start NCM diagnostic

The diagnostic tool is started by *'Windows-START menu ➔ SIMATIC ➔ ... NCM S7 ➔ Diagnostic'*.

Structure



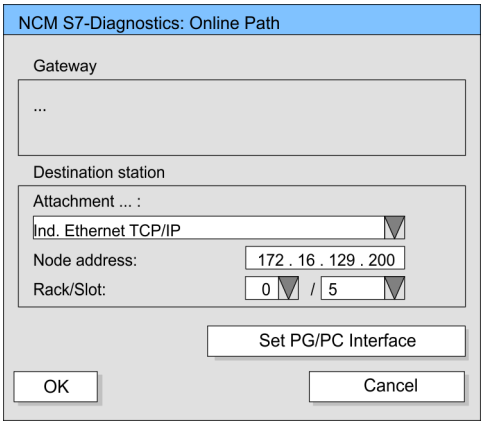
The working surface of the diagnostic tool has the following structure:

- The *'navigation area'* at the left side contains the hierarchical listed diagnostic objects. Depending on CP type and configured connections there is an adjusted object structure in the navigation area.
- The *'information area'* at the right side always shows the result of the navigation function you chose in the navigation area.

No diagnostic without connection

A diagnostic always requires an online connection to the CP you want to control. For this click at the symbol bar.

The following dialog window appears:



Set the following parameters at *'Destination station'*:

- *Attachment ..:*
  Ind. Ethernet TCP/IP
- *Node addr.:*
  Enter the IP address of the CP
- *Rack/slot:*
  Enter the module rack and slot of the CP 343 that you've placed at the 2. slot. Set your PG/PC interface to "TCP/IP -> Network card .... ". Via [OK] you start the online diagnostic.

Read diagnostic buffer

The CP has a diagnostic buffer. This has the architecture of a ring memory and may store up to 100 diagnostic messages. The NCM diagnostic allows you to monitor and evaluate the diagnostic messages via the diagnostic object *Diagnostic buffer*. Via a double click on a diagnostic message the NCM diagnostic shows further information.

Approach for diagnostic

You execute a diagnostic by clicking on a diagnostic object in the navigation area. More functions are available via the menu and the symbol bar.

Please always control the preconditions for an operative communication using the check list. *‘NCM diagnostics - Help for error diagnostics’ page 128*

For the aimed diagnostic deployment the following approach is convenient:

1. Start diagnostic.
2. Open the dialog for the online connection with , enter connection parameters and establish the online connection with [OK].
3. Identify the CP and check the recent state of the CP via module status.
4. Check the connections for particularities like:
   - Connection status
   - Receive status
   - Send status
5. Control and evaluate the diagnostic buffer of the CP via *‘Diagnostic buffer’*.
6. As needed, alter project engineering respectively programming and restart diagnostic.

8.13 Coupling to other systems

Overview

The operating mode FETCH/WRITE supported at TCP res. ISO-on-TCP can be used for accesses of partner devices to the PLC system memory. To be able to use this access also for example for implementation in PC applications you have to know the telegram structure for orders. The specific headers for request and acknowledgement telegrams have per default a length of 16Byte and are described at the following pages.

ORG format

The organization format is the abbreviated description of a data source or a data destination in a PLC environment. The available ORG formats are listed in the following table. The ERW-identifier is used for the addressing of data blocks. In this case the data block number is entered into this identifier. The start address and quantity provide the address for the memory area and they are stored in HIGH-/LOW- format (Motorola-formatted addresses)

Description	Type	Range
ORG identifier	BYTE	1...x
ERW identifier	BYTE	1...255
Start address	HILOWORD	0...y
Length	HILOWORD	1...z

The following table contains a list of available ORG-formats. The "length" must not be entered as -1 (FFFFh).

ORG identifier 01h-04h

CPU area	DB	MB	EB	AB
ORG identifier	01h	02h	03h	04h
Description	Source/destination data from/into data Block in main memory.	Source/destination data from/into flag memory area	Source/destination data from/into process image of the inputs (PII).	Source/destination data from/into process image of the outputs (PIQ).
ERW identifier (DBNO)	DB, from where the source data is retrieved or to where the destination data is transferred.	irrelevant	irrelevant	irrelevant
Start address significance	DBB-No., from where the data is retrieved or where the data is saved.	MB-No., from where the data is retrieved or where the data is saved.	IB-No., from where the data is retrieved or where the data is saved.	QB-No., from where the data is retrieved or where the data is saved.
Length significance	Length of the source/ destination data block in words.	Length of the source/ destination data block in bytes	Length of the source/ destination data block in bytes	Length of the source/ destination data block in bytes

ORG identifier 05h-07h

CPU area	PB	ZB	TB
ORG identifier	05h	06h	07h
Description	source/destination data from/ into peripheral modules. Input module for source data, output module for destination data.	source/destination data from/ into counter cells.	Source/destination data from/ into timer cells.
ERW identifier (DBNO)	irrelevant	irrelevant	irrelevant
Start address Significance	PB-No., from where the data can be retrieved or where it is saved.	ZB-No., from where the data can be retrieved or where it is saved.	TB-No., from where the data can be retrieved or where it is saved.
Length Significance	Length of the source/destina- tion data block in bytes.	Length of the source/destina- tion data block in words (counter cell = 1 word).	Length of the source/destina- tion data block in words (counter cell = 1 word).

Transfer of blocks with numbers >255

ORG identifier 81h-FFh

To transfer data blocks of the number range 256 ... 32768 you may use the ORG identifier 81h-FFh. For the setting of a DB No. >255 needs a length of one word, the $DBNO_{new}$ is assembled from the content of the ORG identifier and the DBNO. $DBNO_{new}$ is created as word as follows:

$DBNO_{new}$

High-Byte								Low-Byte							
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	ORG identifier (0XXXXXXX)							DBNR (XXXXXXXX)							

If the highest bit of the ORG identifier is set, the Low-Byte of $DBNO_{new}$ is defined via DBNO and the High-Byte of $DBNO_{new}$ via ORG identifier, where the highest bit of the ORG identifier is eliminated. The following formula illustrates this:

$DBNO_{new}$ =256 x (ORG-identifier AND 7Fh) + DBNO

Structure of PLC-Header

For every FETCH and WRITE the CP generates PLC header for request and acknowledgment messages. Normally the length of these headers is 16Bytes and have the following structure:

WRITE

Request telegram Remote Station	***Acknowledgement telegram CP***
System ID = "S5" (Word)	System ID ="S5" (Word)
Length Header = 10h (Byte)	Length Header =10h (Byte)
ID OP-Code = 01h (Byte)	ID OP-Code =01h (Byte)
Length OP-Code = 03h (Byte)	Length OP-Code =03h (Byte)
OP-Code = 03h (Byte)	**OP-Code =04h (Byte)**
ORG block = 03h (Byte)	Ackn. block = 0Fh (Byte)
Length ORG block = 08h (Byte)	Length Ack. block = 03h (Byte)
ORG identifier (Byte)*	Error no. (Byte)
ERW identifier (Byte)	Empty block = FFh (Byte)
Start address (Word)	Length empty block = 07h (Byte)
Length (Word)	5 empty bytes attached
Empty block = FFh (Byte)	
Length empty block = 02h (Byte)	
Data up to 64kByte (only if error no.=0)	

FETCH

Request telegram Remote Station	***Acknowledgement telegram CP***
System ID = "S5" (Word)	System ID ="S5" (Word)
Length Header = 10h (Byte)	Length Header =10h (Byte)
ID OP-Code = 01h (Byte)	ID OP-Code =01h (Byte)
Length OP-Code = 03h (Byte)	Length OP-Code =03h (Byte)
OP-Code = 05h (Byte)	**OP-Code =06h (Byte)**
ORG block = 03h (Byte)	Ackn. block = 0Fh (Byte)
Length ORG block = 08h (Byte)	Length Ack. block = 03h (Byte)
ORG identifier (Byte)*	Error no. (Byte)
ERW identifier (Byte)	Empty block = FFh (Byte)
Start address (Word)	Length empty block = 07h (Byte)

Request telegram Remote Station	*Acknowledgement telegram CP*
Length (Word)	5 empty bytes attached
Empty block = FFh (Byte)	Data up to 64kByte
Length empty block = 02h (Byte)	(only if error no.=0)
*) More details to the data area is to be found at "ORG-Format" above.	

Please regard that in opposite to Siemens-S5 systems, the block addressing of these CPUs takes the start address as byte number and the length as number of words.

Messages of error no.

The following messages can be returned via *error no.*:

Error no.	Message
00h	No error occurred
01h	The defined area cannot be read res. written

9 WinPLC7

9.1 System conception

General

WinPLC7 is a programming and simulation software from Yaskawa for every PLC programmable with Siemens STEP®7. This tool allows you to create user applications in FBD, LAD and STL. Besides of a comfortable programming environment, WinPLC7 has an integrated simulator that enables the simulation of your user application at the PC without additional hardware. This "Soft-PLC" is handled like a real PLC and offers the same error behavior and diagnostics options via diagnostics buffer, USTACK and BSTACK.

Detailed information and programming samples may be found at the online help respectively in the online documentation of WinPLC7.

Alternatives

There is also the possibility to use according configuration tools from Siemens instead of WinPLC7 from Yaskawa. Here the proceeding is part of this manual.

System requirements

- Windows XP (SP3)
- Windows Vista
- Windows 7 (32 and 64 bit)
- Windows 8 (32 and 64 bit)

Source

You may receive a *demo version* from Yaskawa. Without any activation with the *demo version* the CPUs 11x of the System 100V from Yaskawa may be configured. To configure the SPEED7 CPUs a license for the "profi" version is necessary. This may be online be received from Yaskawa and activated.

There are the following sources to get WinPLC7:

- Online
  - At www.yaskawa.eu.com in the service area at Downloads a link to the current demo version and the updates of WinPLC7 may be found.
- CD
  - SW211C1DD: WinPLC7 Single license, CD, with documentation in German
  - SW211C1ED: WinPLC7 Single license, CD, with documentation in English

9.2 Installation

Precondition

The project engineering of a SPEED7 CPU from VIPA with WinPLC7 is only possible using an activated "Profi" version of WinPLC7.

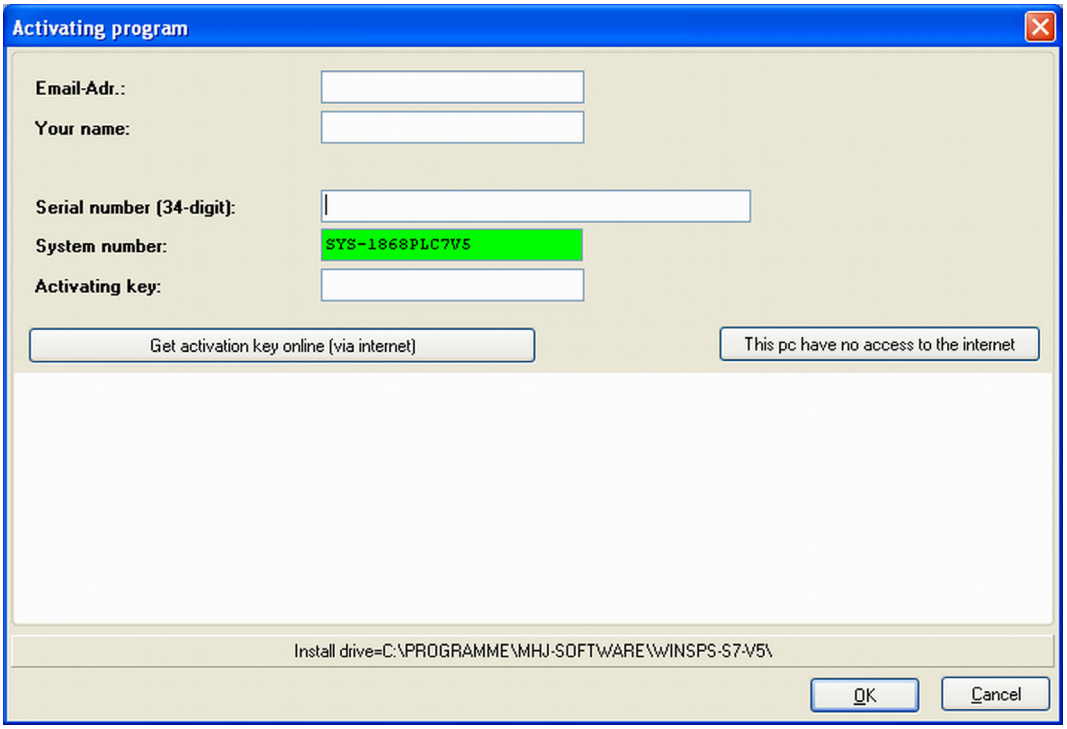
Installation WinPLC7 Demo

The installation and the registration of WinPLC7 has the following approach:

1. For installation of WinPLC7 start the setup program of the corresponding CD respectively execute the online received exe file.
2. Select the according language.
3. Accept the licensing agreement.
4. Set an installation directory and a group assignment and start the installation.

Activation of the "Profi" version

1. Start WinPLC7.
   ⇨ A *'Demo'* dialog is shown
2. Click at [Activate Software].
   ⇨ The following dialog for activation is shown:



3. Fill in the following fields:
   - Email-Addr.
   - Your Name
   - Serial number
     The serial number may be found on a label at the CD case of WinPLC7.
4. If your computer is connected to Internet you may online request the Activation Key by [Get activation key via Internet]. Otherwise click at [This PC has no access to the Internet] and follow the instructions.
   ⇨ With successful registration the activation key is listed in the dialog window respectively is sent by email.
5. Enter this at *'Activation code'* and click at [OK].
   ⇨ Now, WinPLC7 is activated as "Profi" version.

Installation of WinPCAP for station search via Ethernet

To find a station via Ethernet (accessible nodes) you have to install the WinPCAP driver. This driver may be found on your PC in the installation directory at WinSPS-S7-V5/ WinPcap_... .exe. Execute this file and follow the instructions.

9.3 Example project engineering

9.3.1 Job definition

In the example a FC 1 is programmed, which is cyclically called by the OB 1. By setting of 2 comparison values (value1 and value2) during the FC call, an output of the PLC-System should be activated depending on the comparison result.

Here it should apply:

- if value1 = value2 activate output Q 124.0
- if value1 > value2 activate output Q 124.1
- if value1 < value2 activate output Q 124.2

Precondition

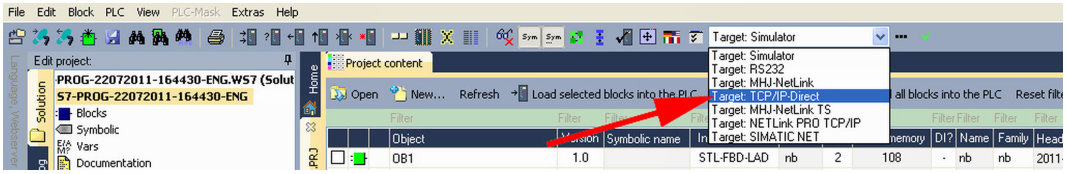
- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.
- One SPEED7 CPU and one digital output module are installed and cabled.
- The Ethernet PG/OP channel of the CPU is connected to your Ethernet network. Your CPU may be connected to your PC with an Ethernet cable either directly or via hub/ switch.
- WinPCap for station search via Ethernet is installed.
- The power supply of the CPU and the I/O periphery are activated and the CPU is in STOP state.

9.3.2 Project engineering

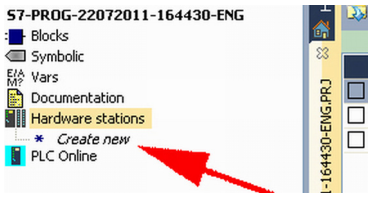
1. Start WinPLC7 ("Profi" version)
2. Create and open a new project with [Create a new solution].

Hardware configuration

1. For the call of the hardware configurator it is necessary to set WinPLC7 from the Simulator-Mode to the Offline-Mode. For this and the communication via Ethernet set "Target: TCP/IP Direct".



2. Double click to *'Hardware stations'* and here at *'Create new'*.



3. Enter a station name. Please consider that the name does not contain any spaces.
4. After the load animation choose in the register Select PLC-System the system "VIPA SPEED7" and click to [Create]. A new station is created.
5. Save the empty station with *[Strg]+[S]*.
6. By double click or drag&drop the according VIPA CPU in the hardware catalog at *'CPU SPEED7'* the CPU is inserted to your configuration.
7. For output place a digital output module, assign the start address 124 and save the hardware configuration.

Establish online access via Ethernet PG/OP channel:

1. Open the CPU-Properties, by double clicking to the CPU at slot 2 in the hardware configurator.
2. Click to the button [Ethernet CP-Properties (PG/OP-channel)].
   ⇨ The dialog *'Properties CP343'* is opened.
3. Chose the register *'Common Options'*.
4. Click to [Properties Ethernet].
5. Choose the subnet *'PG_OP_Ethernet'*.
6. Enter a valid IP address-and a subnet mask. You may get this from your system administrator.
7. Close every dialog window with [OK].
8. Select, if not already done, *'Target: External TCP/IP direct'*.
9. Open with *'Online ➔ Send configuration to the CPU'* a dialog with the same name.
10. Click to [Accessible nodes]. Please regard to use this function it is necessary to install WinPCap before!
11. Choose your network card and click to [Determining accessible nodes].
    ⇨ After a waiting time every accessible station is listed. Here your CPU with IP 0.0.0.0 is listed, too. To check this the according MAC address is also listed. This MAC address may be found at a label beneath the front flap of the CPU.
12. For the temporary setting of an IP address select you CPU and click to [Temporary setting of the IP parameters]. Please enter the same IP parameters, you configured in the CPU properties and click to [Write Parameters].
13. Confirm the message concerning the overall reset of the CPU.
    ⇨ The IP parameters are transferred to the CPU and the list of accessible stations is refreshed.
14. Select your CPU and click to [Confirm].
    ⇨ Now you are back in the dialog "Send configuration".

Transfer hardware configuration

- Choose your network card and click to [Send configuration].
  ⇨ After a short time a message is displayed concerning the transfer of the configuration is finished.

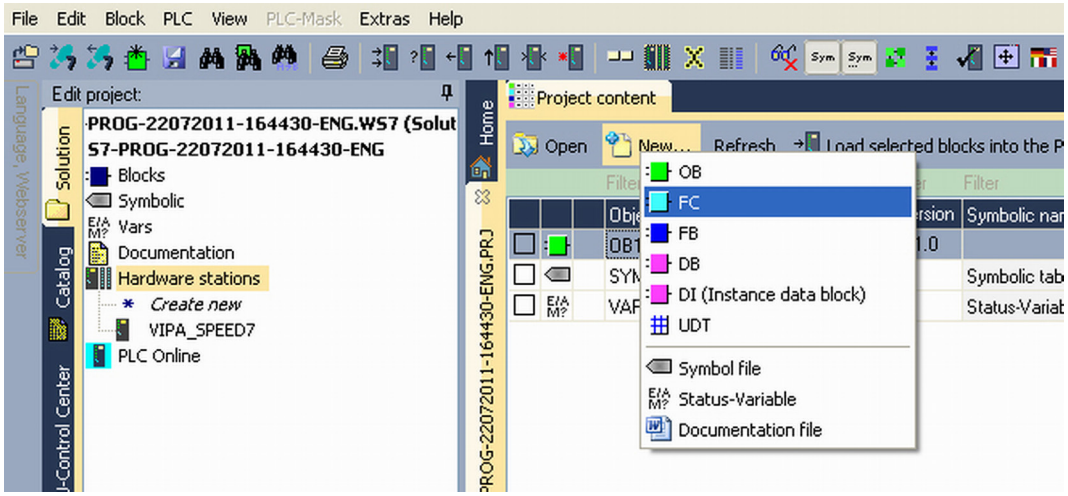
The hardware configuration is finished, now and the CPU may always be accessed by the IP parameters as well by means of WinPLC7.

Usually the online transfer of the hardware configuration happens within the hardware configurator. With 'File ➔ Save active station in the WinPL7 sub project' there is also the possibility to store the hardware configuration as a system file in WinPLC7 to transfer it from WinPLC7 to the CPU.

Programming of the FC 1

The PLC programming happens by WinPLC7. Close the hardware configurator and return to your project in WinPLC7. The PLC program is to be created in the FC 1.

1. In *'Project content'* choose *'New ➔ FC'*.



2. Enter "FC1" as block and confirm with [OK].

   ⇨ The editor for FC 1 is called.

Creating parameters

In the upper part of the editor there is the parameter table. In this example the 2 integer values *value1* and *value2* are to be compared together. Since both values are read only by the function, these are to be defined as "in".

1. Select the *'in →'* row at the *'parameter table'* and enter at the field *'Name'* "value1". Press the *[Return]* key.

   ⇨ The cursor jumps to the column with the data type.

2. The data type may either directly be entered or be selected from a list of available data types by pressing the *[Return]* key. Set the data type to INT and press the *[Return]* key.

   ⇨ Now the cursor jumps to the *'Comment'* column.

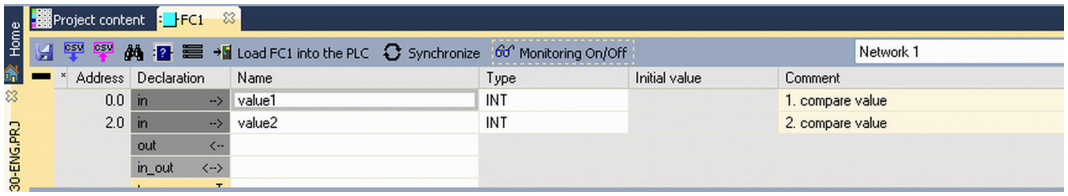
3. Here enter "1. compare value" and press the *[Return]* key.

   ⇨ A new *'in →'* row is created and the cursor jumps to *'Name'*.

4. Proceed for *value2* in the same way as described for *value1*.

5. Save the block. A note that the interface of the block was changed may be acknowledged with [Yes].

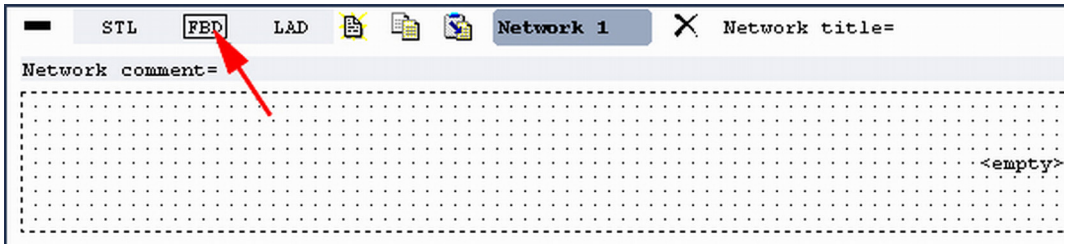
   ⇨ The parameter table shows the following entries, now:



Enter the program

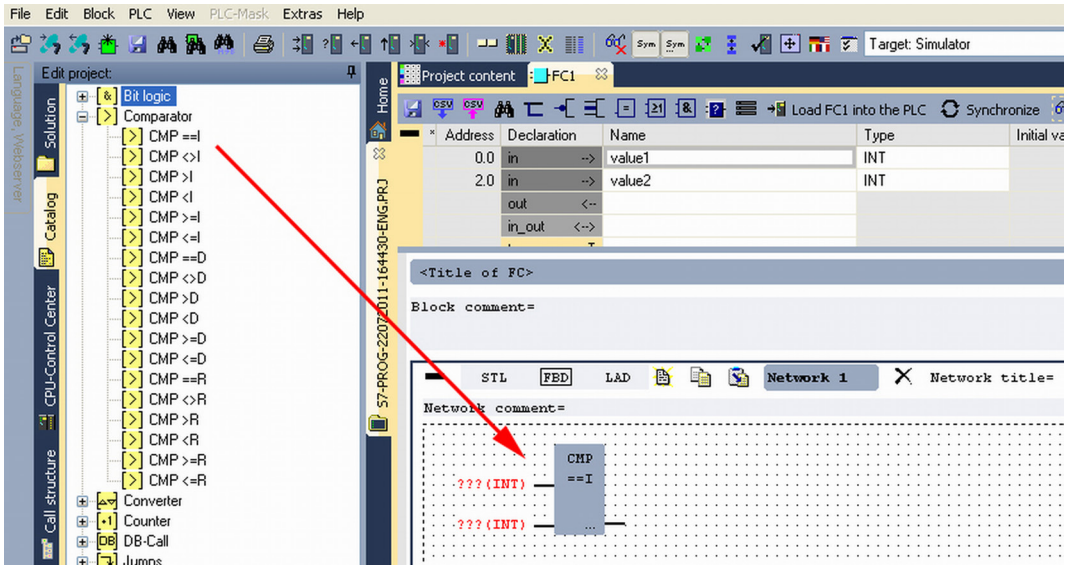
As requested in the job definition, the corresponding output is activated depending on the comparison of *value1* and *value2*. For each comparison operation a separate network is to be created.

1. The program is to be created as FBD (function block diagram). Here change to the FBD view by clicking at *'FBD'*.



2. Click to the input field designated as "<empty>". The available operations may be added to your project by drag&drop from the *hardware catalog* or by double click at them in the *hardware catalog*.

3. Open in the *catalog* the category "Comparator" and add the operation *'CMP==I'* to your network.



4. Click to the input left above and insert *value1*. Since these are block parameters a selection list of block parameters may be viewed by entering "#".

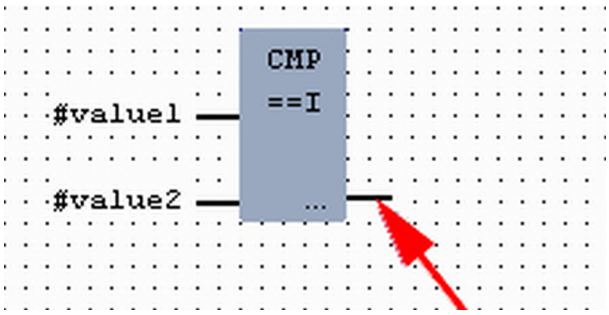
5. Type in "#" and press the *[Return]* key.

6. Choose the corresponding parameter of the list and confirm it with the *[Return]* key.

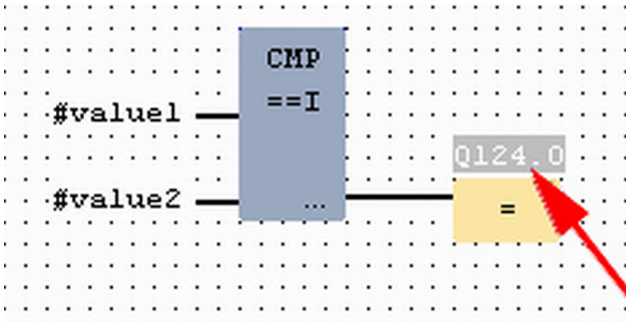
7. Proceed in the same way with the parameter *value2*.

The allocation to the corresponding output, here Q 124.0, takes place with the following proceeding:

1. Click to the output at the right side of the operator.



2. Open in the *catalog* the category *'Bit logic'* and select the function *'--[=]'*. The inserting of *'--[=]'* corresponds to the WinPLC7 shortcut *[F7]*.
3. Insert the output Q 124.0 by clicking to the operand.



⇨ Network1 is finished, now.

Adding a new network

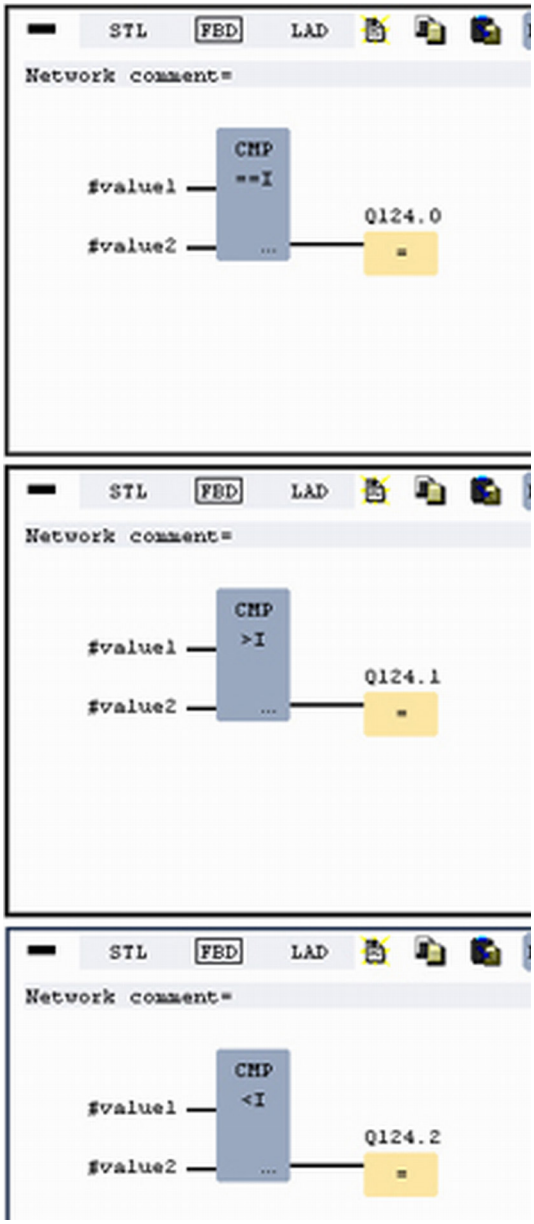
For further comparisons the operations "CMP>I" at Q 124.1 and "CMP<I" at Q 124.2 are necessary. Create a network for both operations with the following proceeding:

1. Move your mouse at an arbitrary position on the editor window and press the right mouse key.
2. Select at *'context menu ➔ Insert new network'*.

   ⇨ A dialog field is opened to enter the position and number of the networks.
3. Proceed as described for "Network 1".

4. Save the FC 1 with *'File ➔ Save content of focused window'* respectively press *[Strg]+[S]*.

   ⇨ After you have programmed the still missing networks, the FC 1 has the following structure:



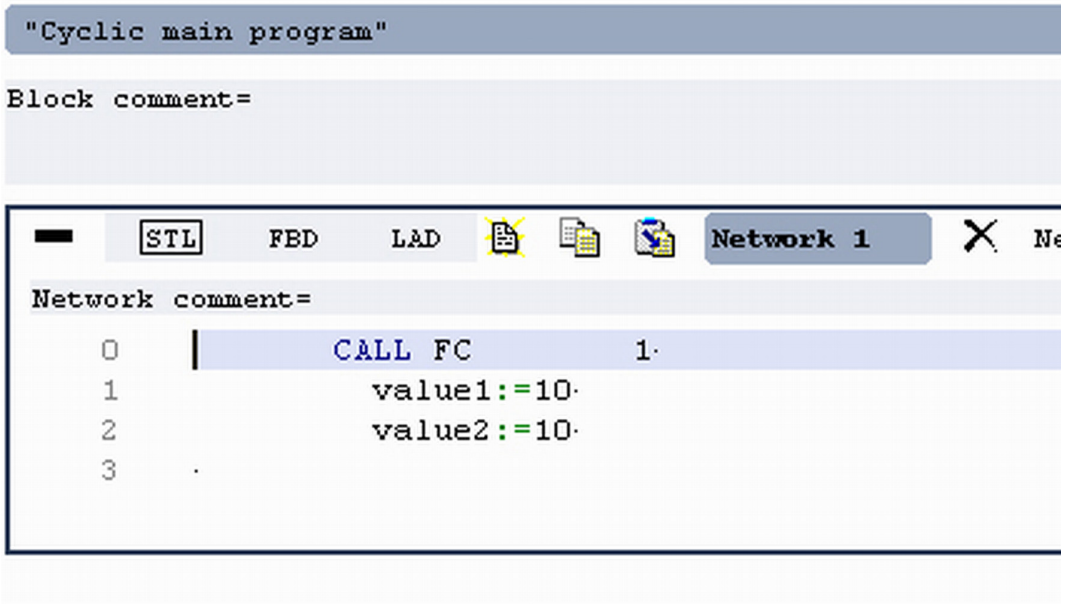
Creating the block OB 1

The FC 1 is to be called from the cycle OB 1.

1. Go to OB 1, which was automatically created with starting the project.
2. Go to *'Project content'* or to *'Solution'* and open the OB 1 by a double click.
3. Change to the STL view.

4. Type in "Call FC 1" and press the *[Return]* key.

   ⇨ The FC parameters are automatically displayed and the following parameters are assigned:



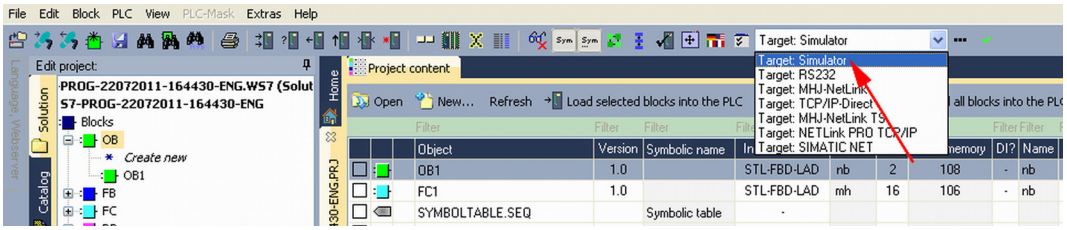
5. Save the OB 1 with respectively press *[Strg]+[S]*.

9.3.3 Test the PLC program in the *Simulator*

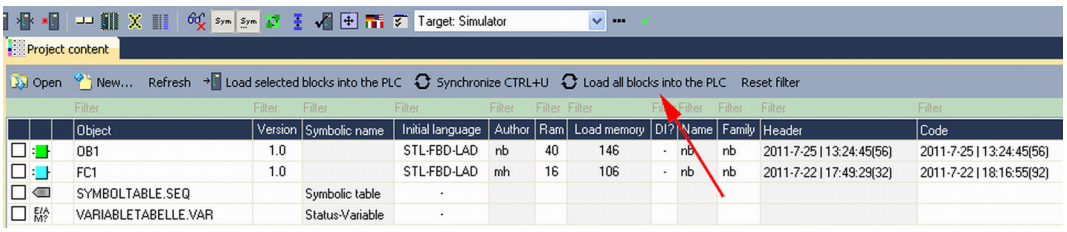
Proceeding

With WinPLC7 there is the possibility to test your project in a *Simulator*.

1. Here select *'Target: Simulator'*.



2. Transfer the blocks to the simulator with [Load all blocks into the PLC].



3. Switch the CPU to RUN, by clicking at *'RUN'* in the *'CPU Control Center'* of *'Edit project'*.

   ⇨ The displayed state changes from STOP to RUN.

4. To view the process image select *'View ➔ Display process image window'* or click at .

   ⇨ The various areas are displayed.

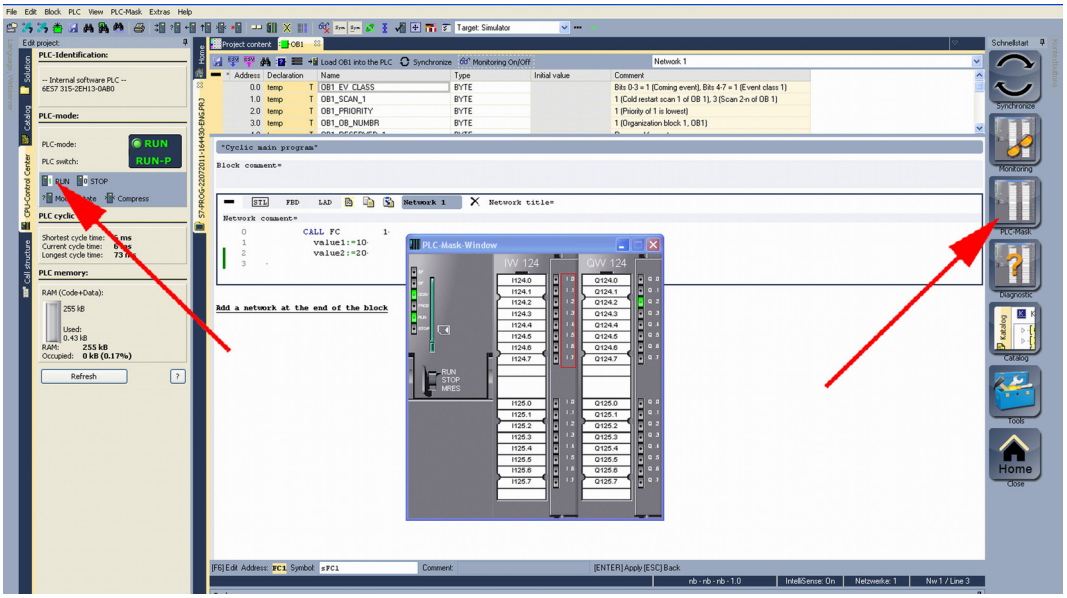
5. Double click to the process image and enter at *'Line 2'* the address PQB 124. Confirm your input with [OK]. A value marked by red color corresponds to a logical "1".

6. Open the OB 1.
7. Change the value of one variable, save the OB 1 and transfer it to the simulator.
   ⇨ According to your settings the process image changes immediately. The status of your blocks may be displayed with *'Block ➔ Monitoring On/Off'*.

Visualization via PLC mask

A further component of the simulator is the *PLC mask*. Here a CPU is graphically displayed, which may be expanded by digital and analog peripheral modules. As soon as the CPU of the simulator is switched to RUN state, inputs may be activated by mouse and outputs may be displayed.

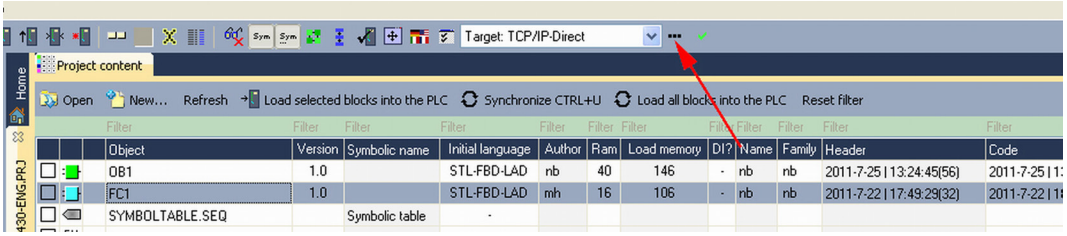
1. Open the *PLC mask* with *'view ➔ PLC mask'*.
   ⇨ A CPU is graphically displayed.
2. Double-click to the output module, open its properties dialog and enter the Module address 124.
3. Switch the operating mode switch to RUN by means of the mouse.
   ⇨ Your program is executed and displayed in the simulator, now.



9.3.4 Transfer PLC program to CPU and its execution

Proceeding

1. For transfer to the CPU set the transfer mode to "Target: TCP/IP-Direct".
2. If there are more network adapters in your PC, the network adapter may be selected via *'Extras ➔ Select network adapter'*.
3. For presetting the Ethernet data click to [...] and click to [Accessible nodes].



4. Click at [Determining accessible nodes].
   ⇨ After a waiting time every accessible station is listed.

5. Choose your CPU, which was provided with TCP/IP address parameters during the hardware configuration and click to [Confirm].
6. Close the dialog *'Ethernet properties'* with [OK].
7. Transfer your project to your CPU with *'PLC ➔ Send all blocks'*.
8. Switch your CPU to RUN state.
9. Open the OB 1 by double click.
10. Change the value of one variable, save the OB 1 and transfer it to the CPU.

    ⇨ According to your settings the process image changes immediately. The status of your blocks may be displayed with *'Block ➔ Monitoring On/Off'*.

10 Configuration with TIA Portal

10.1 TIA Portal - Work environment

10.1.1 General

General

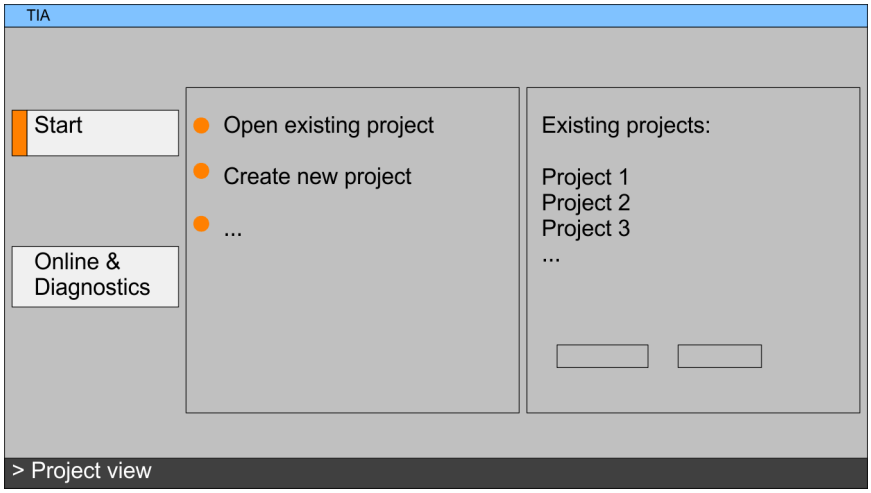
In this chapter the project engineering of the Yaskawa CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a Yaskawa CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means **T**otally **i**ntegrated **A**utomation from Siemens. Here your Yaskawa 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.

10.1.2 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

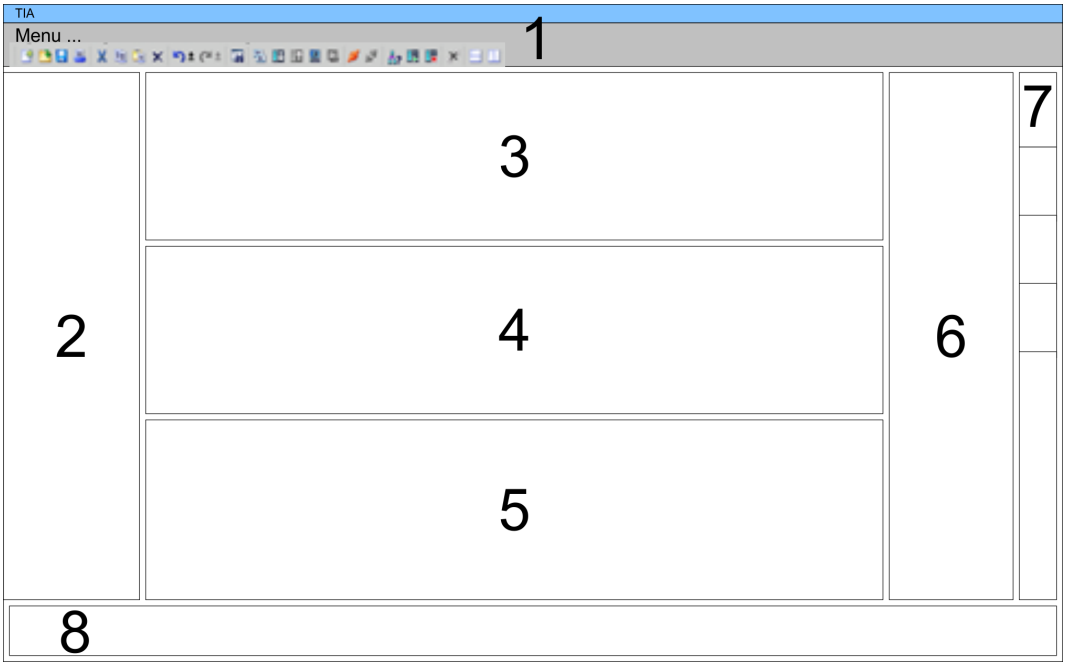
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

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

10.2 TIA Portal - Hardware configuration - CPU

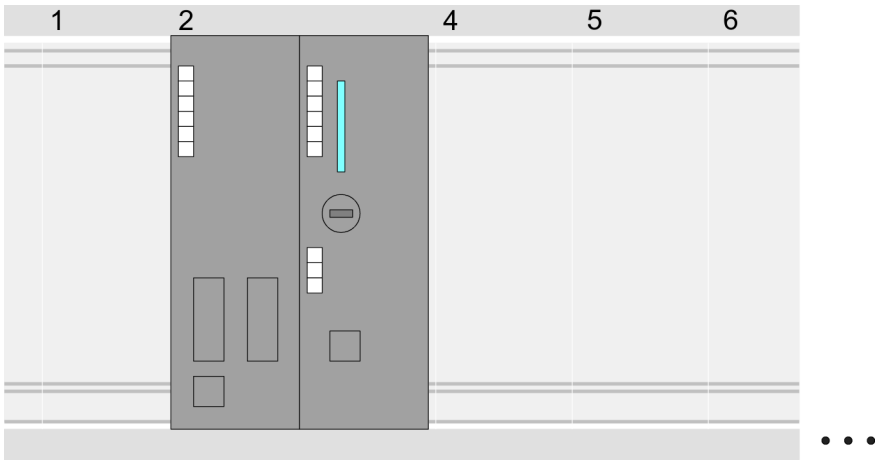
Configuration Siemens CPU

With the Siemens TIA Portal the CPU 315-4NE23 from VIPA is to be configured as CPU 317-2DP (6ES7 317-2AK14 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 317-2DP (6ES7 317-2AK14 V3.3)

   ⇨ The CPU is inserted with a profile rail.



Device overview:

Module	...	Slot	...	Type	...
PLC ...		2		CPU 317-2DP	
MPI/DP interface		2 X1		MPI/DP interface	
DP interface		2 X2		DP interface	
...		...		...	

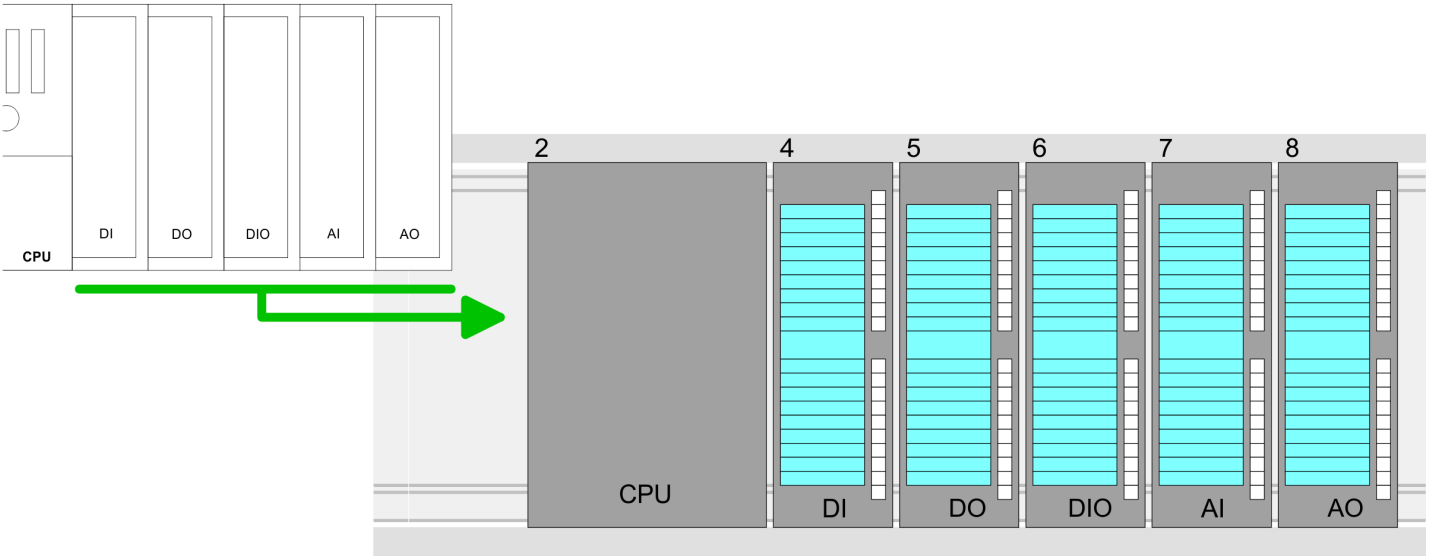
Setting standard CPU parameters

Since the CPU from VIPA is configured as Siemens CPU, so the setting of the 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. 5.8.2 'Parameters CPU' page 43*

10.3 TIA Portal - Hardware configuration - I/O modules

Hardware configuration of the modules

After the hardware configuration of the CPU place the System 300 modules at the bus in the plugged sequence. For this drag&drop the according module from the Hardware catalog to the according position of the profile rail in the *Project area* or in the *Device overview*



Device overview

Module	...	Slot	...	Type	...
PLC...		2		CPU ...	
...		...		...	
		3			
DI...		4		DI...	
DO...		5		DO...	
DIO...		6		DIO...	
AI...		7		AI...	
AO...		8		AO...	

Parametrization

For parametrization click in the *Project area* respectively in the *Device overview* on the module you want to parameterize. The parameters of the module appear in the Properties dialog. Here you can make your parameter settings.

10.4 TIA Portal - Hardware configuration - Ethernet PG/OP channel

Overview

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

- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address.
- For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this. This is called "initialization".
- This can be done with the Siemens TIA Portal.

Assembly and commissioning

1. Install your System 300S with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the Ethernet jack of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization" via Online functions

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

→ Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found as 1. address under the front flap of the CPU on a sticker on the left side.

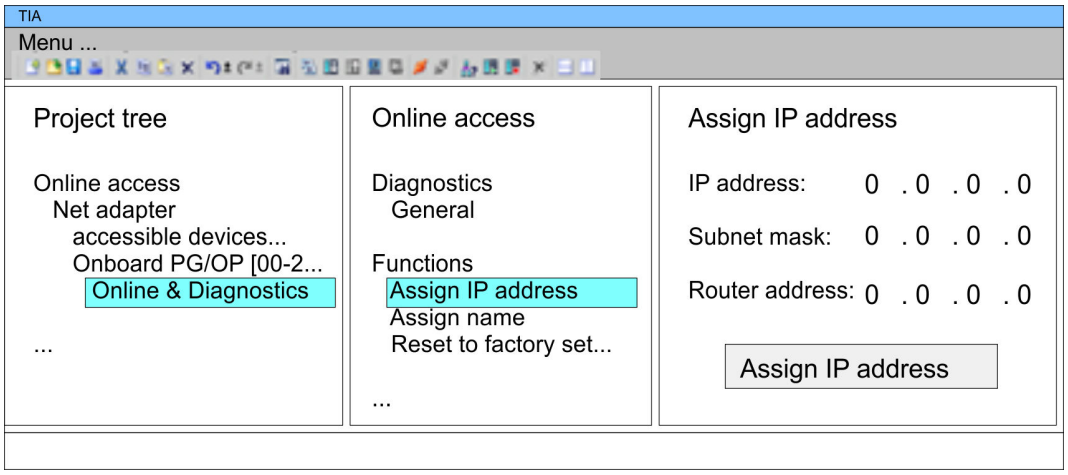
Assign IP address parameters

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens TIA Portal with the following proceeding:

1. Start the Siemens TIA Portal.
2. Switch to the *'Project view'*.
3. Click in the *'Project tree'* at *'Online access'* and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.
4. To get the stations and their MAC address, use the *'Accessible device'*. The MAC address can be found at the 1. label beneath the front flap of the CPU.
5. Choose from the list the module with the known MAC address (Onboard PG/OP [MAC address]) and open with "Online & Diagnostics" the diagnostics dialog in the Project area.
6. Navigate to *Functions* > *Assign IP address*. Type in the IP configuration like IP address, subnet mask and gateway.

7. Confirm with [Assign IP configuration].

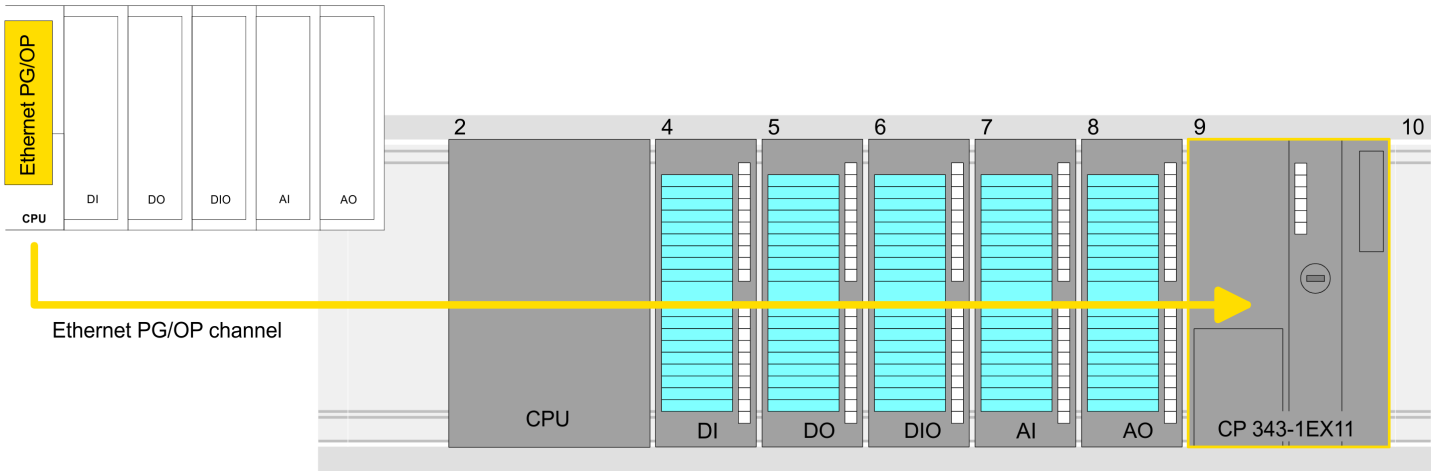
   ⇨ Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.



Due to the system you may get a message that the IP address could not be assigned. This message can be ignored.

Take IP address parameters in project

1. Open your project.
2. If not already done, configure in the *'Device configuration'* a Siemens CPU 317-2DP (6ES7 317-2AK14 V3.3).
3. Configure the System 300 modules.
4. For the Ethernet PG/OP channel you have to configure a Siemens CP 343-1 (6GK7 343-1EX11 0XE0) always as last module after the really plugged modules.
5. Open the "Property" dialog by clicking on the CP 343-1EX11 and enter for the CP at "Properties" at "Ethernet address" the IP address data, which you have assigned before.
6. Transfer your project.



Device overview:

Module	...	Slot	...	Type	...
PLC...		2		CPU ...	
...		...		...	
		3			
DI...		4		DI...	
DO...		5		DO...	
DIO...		6		DIO...	
AI...		7		AI...	
AO...		8		AO...	
CP 343-1		9		CP 343-1	

10.5 TIA Portal - Setting VIPA specific CPU parameters

Requirements

Since the VIPA specific CPU parameters may be set, the installation of the SPEEDBUS.GSD from VIPA in the hardware catalog is necessary. The CPU may be configured in a PROFIBUS master system and the appropriate parameters may be set after installation.

Installation of the SPEEDBUS.GSD

The GSD (Geräte-Stamm-Datei) is online available in the following language versions. Further language versions are available on inquires:

Name	Language
SPEEDBUS.GSD	German (default)
SPEEDBUS.GSG	German
SPEEDBUS.GSE	English

The GSD files may be found at www.yaskawa.eu.com at the service area.

The integration of the SPEEDBUS.GSD takes place with the following proceeding:

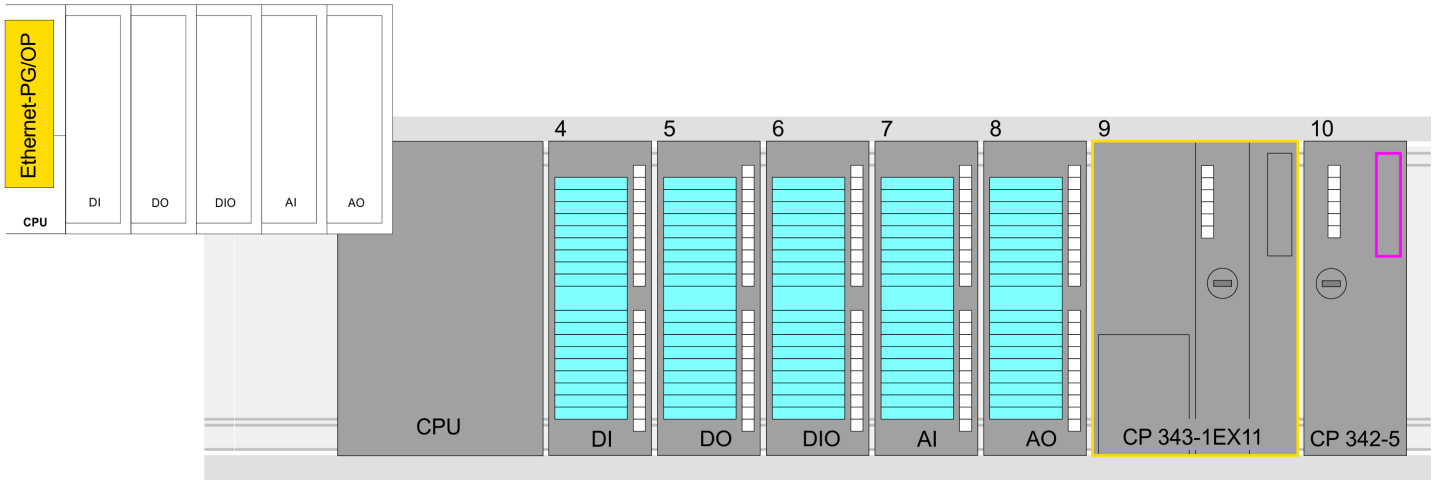
1. Go to the service area of www.yaskawa.eu.com.
2. Load from the download area at *'Config files ➔ PROFIBUS'* the according file for your System 300S.
3. Extract the file to your work directory.
4. Start the hardware configurator from Siemens.
5. Close every project.
6. Select *'Options ➔ Install new GSD-file'*.
7. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from Yaskawa may now be found in the hardware catalog at PROFIBUS-DP / Additional field devices / I/O / VIPA_SPEEDBUS.

Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

Proceeding

The embedding of the CPU 315-4NE23 happens by means of a virtual PROFIBUS master system with the following approach:

1. Start the Siemens TIA Portal.
2. Configure in the Device configuration the according Siemens CPU.
3. Configure your System 300 modules.
4. Configure your Ethernet PG/OP channel always as last module after the really plugged modules.
5. Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".



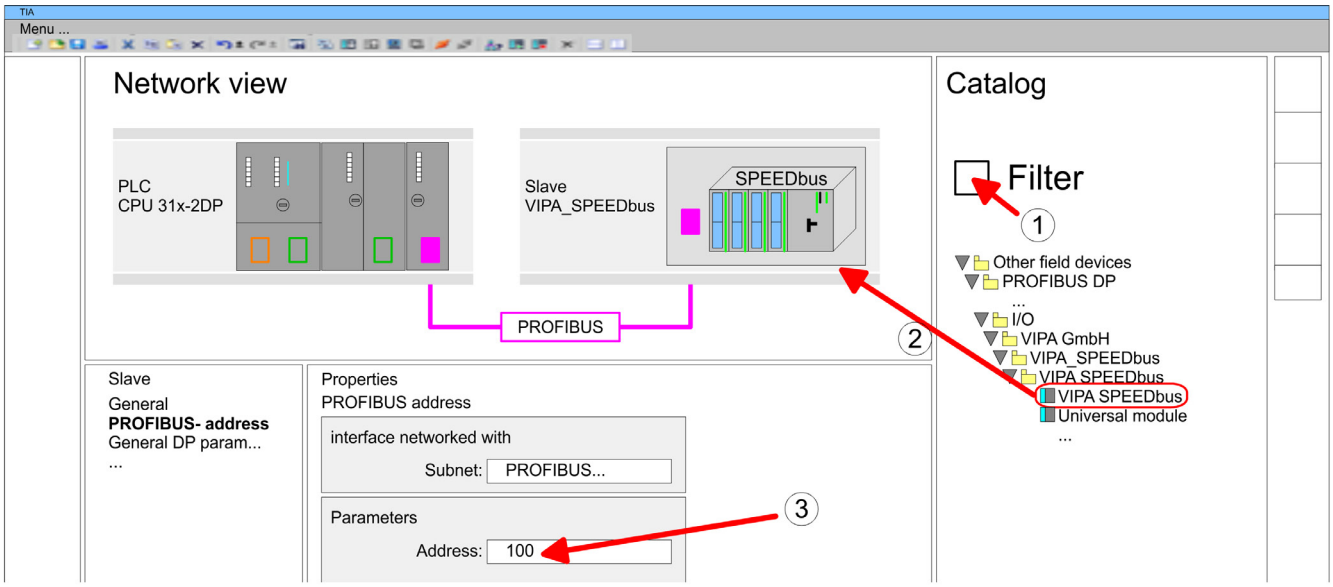
Device overview

Module	...	Slot	...	Type	...
PLC...		2		CPU ...	
...		...		...	
		3			
DI...		4		DI...	
DO...		5		DO...	
DIO...		6		DIO...	
AI...		7		AI...	
AO...		8		AO...	
CP 343-1		9		CP 343-1	
CP 342-5		10		CP 342-5	

Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

Connect VIPA_SPEEDbus

1. Switch in the *Project area* to *Network view*.
2. Connect the slave system "VIPA_SPEEDbus". After installing the SPEEDBUS.GSD this may be found in the hardware catalog at: *Other field devices > PROFIBUS DP > I/O > VIPA GmbH > VIPA_SPEEDbus.*
3. Set for the SPEEDbus slave system the PROFIBUS address 100.



4. Click at the slave system and open the *'Device overview'* in the *Project area*.
5. Configure at slot 1 the VIPA CPU 315-4NE23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 315-4NE23 the properties dialog of the CPU is showed.

Device overview

Module	...	Slot	...	Type	...
Slave ...		0		VIPA SPEEDbus	
315-4NE23		1		315-4NE23 ...	
		2			

⇨ As soon as the project is transferred together with the PLC user program to the CPU, the parameters will be taken after start-up.

10.6 TIA Portal - Yaskawa-Include library

Overview

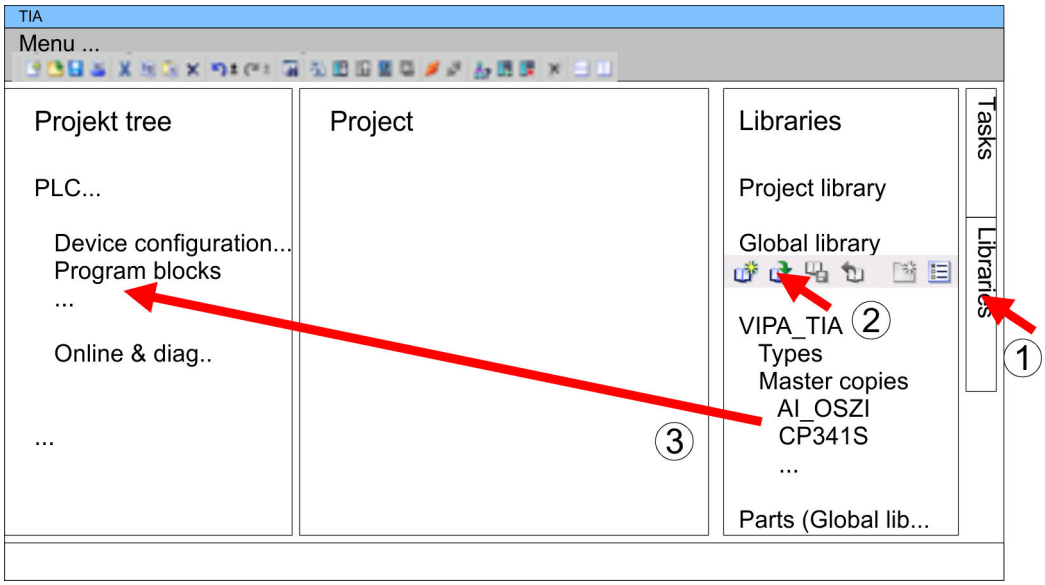
- The Yaskawa specific blocks can be found in the "Service" area of www.yaskawa.eu.com as library download file at *Downloads* > *VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use Yaskawa specific blocks you have to import them into your project.
  Execute the following steps:
  – Load an unzip the file ...TIA_V*xx*.zip (note TIA Portal version)
  – Open library and transfer blocks into the project

Unzip ...TIA_Vxx.zip

Start your un-zip application with a double click on the file TIA_V*xx*.zip and copy all the files and folders in a work directory for the Siemens TIA Portal.

Open library and transfer blocks into the project

1. Start the Siemens TIA Portal with your project.
2. Switch to the *Project view*.
3. Choose "Libraries" from the task cards on the right side.
4. Click at "Global libraries".
5. Click at "Open global libraries".
6. Navigate to your directory and load the file ...TIA.al*xx*.



7. Copy the necessary blocks from the library into the "Program blocks" of the *Project tree* of your project. Now you have access to the Yaskawa specific blocks via your user application.

10.7 TIA Portal - Project transfer

Overview

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

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

Transfer via MPI

Currently the VIPA programming cables for transfer via MPI are not supported. This is only possible with the programming cable from Siemens.

1. Establish a connection to the CPU via MPI with an appropriate programming cable. Information may be found in the corresponding documentation of the programming cable.
2. Switch-ON the power supply of your CPU and start the Siemens TIA Portal with your project.
3. Select in the *Project tree* your CPU and choose *'Context menu ➔ Download to device ➔ Hardware configuration'* to transfer the hardware configuration.
4. To transfer the PLC program choose *'Context menu ➔ Download to device ➔ Software'*. Due to the system you have to transfer hardware configuration and PLC program separately.

Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X5: 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".

Please consider to use the same IP address data in your project for the CP 343-1.

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet jack to your Ethernet.
2. Open your project with the Siemens TIA Portal.
3. Click in the *Project tree* at *Online access* and choose here by a double-click your network card, which is connected to the Ethernet PG/OP interface.
4. Select in the *Project tree* your CPU and click at [Go online].
5. Set the access path by selecting "PN/IE" as type of interface, your network card and the according subnet. Then a net scan is established and the corresponding station is listed.
6. Establish with [Connect] a connection.
7. Click to *'Online ➔ Download to device'*.

   ⇨ The according block is compiled and by a request transferred to the target device. Provided that no new hardware configuration is transferred to the CPU, the entered Ethernet connection is permanently stored in the project as transfer channel.

Transfer via memory card

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

- S7PROG.WLD
- AUTOLOAD.WLD

1. Create in the Siemens TIA Portal a wld file with *'Project ➔ Memory card file ➔ New'*.
   ⇨ The wld file is shown in the *Project tree* at "SIMATIC Card Reader" as "Memory card file".
2. Copy the blocks from the *Program blocks* to the wld file. Here the hardware configuration data are automatically copied to the wld file as "System data".

Transfer memory card → CPU

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

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

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

Transfer CPU → Memory card

When a memory card has been installed, the write command stores the content of the RAM as S7PROG.WLD on the memory card. The write command can be found in the Siemens TIA Portal in the Task card "Online tools" in the command area at "Memory" as button [Copy RAM to ROM]. The MC LED blinks during the write access. When the LED expires, the write process is finished. If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Please note that in the Siemens TIA Portal with some CPU types the [Copy RAM to ROM] button is not available. Instead please use the CMD auto command SAVE PROJECT. *Chap. 5.18 'CMD - auto commands' page 68*

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries, you select *Online & Diagnostics* in the Siemens TIA Portal. Here you can access the "Diagnostics buffer". *Chap. 5.19 'Diagnostic entries' page 70*

Appendix

Content

A System specific event IDs 159
B Integrated blocks 207
C SSL partial list 211

A System specific event IDs

Event IDs *Chap. 5.19 'Diagnostic entries' page 70*

Event ID	Description
0x115C	Manufacture interrupt for EtherCAT / PROFINET IO
	OB: OB number
	ZINFO1: Logical address of the slave station that triggered the interrupt
	ZINFO2: Interrupt type
	0: Reserved
	1: Diagnostic interrupt (incoming)
	2: Process interrupt
	3: Pull interrupt
	4: Plug interrupt
	5: Status interrupt
	6: Update interrupt
	7: Redundancy interrupt
	8: Controlled by the supervisor
	9: Enabled
	10: Wrong sub module plugged
	11: Recurrence of the sub module
	12: Diagnostic interrupt (outgoing)
	13: Cross traffic connection message
	14: Neighbourhood change message
	15: Synchronisation message (bus)
	16: Synchronisation message (device)
	17: Network component message
	18: Clock synchronisation message (bus)
	31: Pull interrupt component
	32: Vendor-specific interrupt min.
	33: Vendor-specific interrupt topology change
	127: Vendor-specific interrupt max.
	ZINFO3: CoE error code
0xE003	Error in access to periphery
	ZINFO1: Transfer type
	ZINFO2: Periphery address
	ZINFO3: Slot
0xE004	Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot

Event ID	Description
0xE005	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE007	Configured input/output bytes do not fit in the periphery area
0xE008	Internal error - Please contact the hotline!
0xE009	Error on accessing the standard backplane bus
0xE010	Non-defined component recognised at the standard backplane bus
	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
	ZINFO2: Slot
	ZINFO3: Not user relevant
	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

Event ID	Description
	ZINFO1: Error
	4: SSL wrong
	5: Sub-SSL wrong
	6: Index wrong
	ZINFO2: SZL-ID
	ZINFO3: Index
0xE0CC	Communication error
	ZINFO1: Error code
	1: Wrong priority
	2: Buffer overrun
	3: Telegram format error
	4: Wrong SSL request (SSL-ID invalid)
	5: Wrong SSL request (SSL-Sub-ID invalid)
	6: Wrong SSL request (SSL-Index invalid)
	7: Wrong value
	8: Wrong return value
	9: Wrong SAP
	10: Wrong connection type
	11: Wrong sequence number
	12: Faulty block number in the telegram
	13: Faulty block type in the telegram
	14: Inactive function
	15: Wrong size in the telegram
	20: Error in writing on MMC
	90: Faulty buffer size
	98: Unknown error
	99: Internal error
0xE0CD	Error at DP-V1 job management
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE0CE	Error: Time out when sending i-slave diagnostics
0xE100	Memory card access error
0xE101	Memory card error file system
0xE102	Memory card error FAT
0xE104	Memory card error at saving
	ZINFO3: Not user relevant

Event ID	Description
0xE200	Memory card writing finished (Copy Ram2Rom)
	OB: Not user relevant
	PK: Not user relevant
0xE210	Memory card reading finished (reload after memory reset)
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1 - 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

Event ID	Description
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3: Block number
0xE300	Internal flash writing completed (copy Ram2Rom)
0xE310	Internal flash reading completed (recharging after battery failure)
0xE400	FSC card was plugged
	OB: FSC activated from this slot (PK)
	OB: The inserted FSC is the activated FSC
	OB: The inserted FSC is compatible with the CPU
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00

Event ID	Description
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE401	FSC card was removed
	OB: Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30

Event ID	Description
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
	DatID: FeatureSet Trialtime in minutes
0xE402	A configured functionality is not activated. The configuration is accepted, but the PLC can not go to RUN.
	ZINFO1: Required FSC: PROFIBUS
	ZINFO1: Required FSC: MOTION

Event ID	Description
	ZINFO2: Number of released axes
	ZINFO3: Number of configured axes
0xE403	FSC can not be activated in this CPU
	OB: FCS error code
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00

Event ID	Description
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE404	Feature set deleted due to CRC error
0xE405	The trial time of a feature set/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

Event ID	Description
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
	DatID: FeatureSet Trialtime in minutes
0xE406	The inserted feature set is corrupt
	PK: FSC source
	0: CPU
	1: Card
0xE410	A CPU feature set was activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30

Event ID	Description
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE500	Memory management: Deleted block without corresponding entry in BstList
	ZINFO2: Block type

Event ID	Description
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3: Block number
0xE501	Parser error
	ZINFO1: Error code
	1: Parser error: SDB structure
	2: Parser error: SDB is not a valid SDB type
	ZINFO2: SDB type
	ZINFO3: SDB number
0xE502	Error in protect.wld
	ZINFO2: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3: Block number

Event ID	Description
0xE503	Inconsistency of code sizes and block sizes in the working memory
	ZINFO1: Code size
	ZINFO2: Block size (high word)
	ZINFO3: Block size (low word)
0xE504	Additional information for CRC error in the working memory
	ZINFO2: Block address (high word)
	ZINFO3: Block address (low word)
0xE505	Internal error - Please contact the hotline!
	ZINFO1: Cause for MemDump
	0: Unknown
	1: Manual request
	2: Invalid OP value
	3: CRC code error
	4: Processor exception
	5: Processor exception with dump after reboot
	6: Block-CRC error
0xE604	Multiple configuration of a periphery address for Ethernet PG/OP channel
	ZINFO1: Periphery address
	ZINFO3: 0: periphery address is input, 1: periphery address is output
0xE605	Too many productive connections configured
	ZINFO1: Interface slot
	ZINFO2: Number of configured connections
	ZINFO3: Number of admissible connections
0xE610	On-board PROFIBUS/MPI: Bus error removed
	PK: Not user relevant
	ZINFO1: Interface
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE701	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE703	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO1: Master system ID
	ZINFO2: Slave address

Event ID	Description
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE705	Too many PROFIBUS slaves configured
	ZINFO1: Diagnostic address of the PROFIBUS master
	ZINFO2: Number of configured slaves
	ZINFO3: Number of admissible slaves
0xE710	On-board PROFIBUS/MPI: Bus error occurred
	PK: Not user relevant
	ZINFO1: Interface
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE720	Internal error - Please contact the hotline!
	ZINFO1: Slave no
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Master system ID
0xE721	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Master system ID
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE722	Internal error - Please contact the hotline!
	ZINFO1: Channel-Event
	0: Channel offline
	1: Bus error
	2: Internal error
	ZINFO2: Master system ID
	DatID: Not user relevant
0xE723	Internal error - Please contact the hotline!
	ZINFO1: Error code
	1: Parameter error
	2: Configuration error
	ZINFO2: Master system ID
	DatID: Not user relevant
0xE780	Internal error - Please contact the hotline!
0xE781	Address range exceeds process image limit
	ZINFO1: Address

Event ID	Description
	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 error - Please contact the hotline!
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE809	Internal error - Please contact the hotline!
	ZINFO3: Not user relevant
0xE80A	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80B	CMD - auto command: DIAGBUF recognized and executed
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80C	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80D	Internal error - Please contact the hotline!

Event ID	Description
0xE80E	CMD - auto command: SET_NETWORK recognized and executed
0xE80F	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE810	Internal error - Please contact the hotline!
0xE811	Internal error - Please contact the hotline!
0xE812	Internal error - Please contact the hotline!
0xE813	Internal error - Please contact the hotline!
0xE814	CMD - auto command: SET_MPI_ADDRESS identified
0xE816	CMD - auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal error - Please contact the hotline!
	ZINFO3: Not user relevant
0xE820	Internal message
0xE821	Internal message
0xE822	Internal message
0xE823	Internal message
0xE824	Internal message
0xE825	Internal message
0xE826	Internal message
0xE827	Internal message
0xE828	Internal message
0xE829	Internal message
0xE82A	CMD - auto command: CPUTYPE_318 recognized and executed
	ZINFO3: Error code
0xE82B	CMD - auto command: CPUTYPE_ORIGINAL recognized and executed
	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
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)

Event ID	Description
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
0xE910	PG/OP: Input peripheral address out of peripheral area
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
0xE911	PG/OP: Output peripheral address out of peripheral area
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
0xE920	Configuration error PROFINET
	ZINFO1 - Position 0: Error code
0xE980	Error when loading the WebVisu project file
0xE981	Error in the configuration of the WebVisu project
0xE982	Internal error of the WebVisu server
0xE983	Hardware configuration of the control is not loaded, WebVisu is not started
0xE984	WebVisu is blocked by the user, start of the WebVisu was prevented
0xE985	WebVisu was started
0xE986	WebVisu was stopped
0xE987	WebVisu was enabled by the user
0xE988	WebVisu was disabled by the user
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

Event ID	Description
	ZINFO1: Slot
	DatID: Not user relevant
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Type identifier target
	ZINFO3: Type identifier
	DatID: Not user relevant
0xEA03	SBUS: Communication error between CPU and IO controller
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	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

Event ID	Description
	7: Not Connected
	8: Unknown
	DatID: Not user relevant
0xEA04	SBUS: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA05	Internal error - Please contact the hotline!
0xEA07	Internal error - Please contact the hotline!
0xEA08	SBUS: Configured input data width not the same as the connected input data width
	ZINFO1: Configured input data width
	ZINFO2: Slot
	ZINFO3: Input data width of the connected component
0xEA09	SBUS: Configured output data width not the same as the connected output data width
	ZINFO1: Configured output data width
	ZINFO2: Slot
	ZINFO3: Output data width of the plugged component
0xEA10	SBUS: Input periphery address outside the periphery area
	ZINFO1: Periphery address
	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!

Event ID	Description
	PK: Not user relevant
	ZINFO2: HW slot
	ZINFO3: Interface type
	DatID: Not user relevant
0xEA1A	SBUS: Error in access to SBUS FPGA address table
	PK: Not user relevant
	ZINFO2: HW slot
	ZINFO3: Table
	0: Read
	1: Writing
	DatID: Not user relevant
0xEA20	Error: RS485 interface is not pre-set to PROFIBUS DP master bus a PROFIBUS DP master is configured
0xEA21	Error: Configuration RS485 interface X2/X3: PROFIBUS DP master is configured but missing
	ZINFO2: Interface X is configured incorrectly
0xEA22	Error: 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

Event ID	Description
	DatID: Line
0xEA50	PROFINET IO controller: Error in the configuration
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Slot at the device
	DatID: Not user relevant
0xEA51	PROFINET IO controller: There is no PROFINET IO controller at the configured slot
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Recognized type identifier at the configured slot
	DatID: Not user relevant
0xEA52	PROFINET IO controller: Too many configured PROFINET IO controllers
	PK: Not user relevant
	ZINFO1: Number of configured controllers
	ZINFO2: Slot of the excessively configured controller
	DatID: Not user relevant
0xEA53	PROFINET IO controller: 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
	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 configured
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Number of configured slots
0xEA56	PROFINET IO controller: Too many subslots configured
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Number of configured subslots
0xEA57	PROFINET IO controller: The port configuration in the virtual SLIO device has no effect.
0xEA61	Internal error - Please contact the hotline!

Event ID	Description
	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
	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.)

Event ID	Description
	ZINFO2 - Bit 2: DC parameter invalid
	ZINFO2 - Bit 3: I device configuration invalid (slot gap)
	ZINFO2 - Bit 4: MRP configuration invalid (client)
0xEA65	Internal error - Please contact the hotline!
	PK: Platform
	0: none
	8: CP
	9: Ethernet CP
	10: PROFINET CP
	12: EtherCAT CP
	16: CPU
	ZINFO1: ServiceID in which the error occurred
	ZINFO2: Command in which the error occurred
	1: Request
	2: Connect
	3: Error
0xEA66	PROFINET 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
	DatID: Device
0xEA68	PROFINET IO controller: Error writing dataset
	OB: Rack/slot of the controller
	PK: Error type
	0: Dataset error local

Event ID	Description
	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
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID

Event ID	Description
	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)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING

Event ID	Description
	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
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6F	PROFINET IO controller: PROFINET module deviation

Event ID	Description
	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
	ZINFO2: Not user relevant
0xEA72	Internal error - Please contact the hotline!
	OB: Connection number
	PK: Slot of the controller
	ZINFO1: Error cause

Event ID	Description
	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]
	ZINFO3: Line
	DatID: SvnRevision
0xEA91	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status

Event ID	Description
	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
0xEA97	Internal error - Please contact the hotline!
	ZINFO3: Slot
0xEA98	Error in file reading via SBUS
	PK: Not user relevant
	ZINFO3: Slot

Event ID	Description
	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
	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
	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 SLIO error: Preset configuration does not match actual configuration
	ZINFO1: Bit mask slots 1-16
	ZINFO2: Bit mask slots 17-32
	ZINFO3: Bit mask slots 33-48
	DatID: Bit mask slots 49-64
0xEB03	System SLIO error: IO mapping

Event ID	Description
	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	SLIO-Bus: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	DatID: Input
	DatID: Output
0xEB05	System SLIO 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 SLIO error: Timeout with the isochronous process image
0xEB10	System SLIO 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 SLIO error: Error during bus initialization
	PK: Not user relevant
	DatID: Not user relevant
0xEB20	System SLIO error: Interrupt information undefined
0xEB21	System SLIO error: Accessing configuration data
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC02	EtherCAT: configuration warning
	ZINFO1: Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid

Event ID	Description
	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 SLIO 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 SLIO device
	ZINFO2: Station number
	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: Not user relevant

Event ID	Description
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
	6: Error when sending the SDO request (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	7: Drive returns error in the SDO response (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	8: SDO time out, DC mode could not be determined (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC10	EtherCAT: Recurrence bus with all slaves
	ZINFO1 - Position 0: New status

Event ID	Description
	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: Recurrence 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
	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

Event ID	Description
	DatID: Input address
	DatID: Output address
0xEC12	EtherCAT: Recurrence 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
	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

Event ID	Description
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	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
	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

Event ID	Description
	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: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state

Event ID	Description
	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
	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

Event ID	Description
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
	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 station
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT

Event ID	Description
	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
	0: There is no EC configuration
	1: There is an EC configuration
	ZINFO3: DC in sync
	0: Not in sync
	1: In sync
0xED30	EtherCAT: Topology deviation

Event ID	Description
	ZINFO2: Diagnostic address of the master
0xED31	EtherCAT: Overflow of the interrupt queue
	ZINFO2: Diagnostic address of the master
0xED40	Bus cycle time infringement occurred
	ZINFO1: Logical address of the IO system
0xED50	EtherCAT: Distributed clocks (DC) in sync
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2: Diagnostic address of the master
	ZINFO3: DC state change
	0: Master
	1: Slave
0xED60	EtherCAT: Diagnostic buffer CP: Slave status change
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)

Event ID	Description
	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: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start

Event ID	Description
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Recurrence slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status
0xED61	EtherCAT: Diagnostic buffer CP: CoE emergency
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1 - 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)
	PK: EtherCAT station address (low byte)
	ZINFO1: Index
	ZINFO2: SDO error code (high word)
	ZINFO3: SDO error code (low word)
	DatID: Sub index

Event ID	Description
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
	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

Event ID	Description
	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
	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)

Event ID	Description
	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 Yaskawa.

OB	Name	Description
OB 1	CYCL_EXC	Program Cycle
OB 10	TOD_INT0	Time-of-day Interrupt
OB 20	DEL_INT0	Time delay interrupt
OB 21	DEL_INT1	Time delay interrupt
OB 28	CYC_INT_250us	Cyclic interrupt
OB 29	CYC_INT_500us	Cyclic interrupt
OB 32	CYC_INT2	Cyclic interrupt
OB 33	CYC_INT3	Cyclic interrupt
OB 34	CYC_INT4	Cyclic interrupt
OB 35	CYC_INT5	Cyclic interrupt
OB 40	HW_INT0	Hardware interrupt
OB 55	DP: STATUS ALARM	Status interrupt
OB 56	DP: UPDATE ALARM	Update interrupt
OB 57	DP: MANUFACTURE ALARM	Vendor specific interrupt
OB 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 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	Name	Description
SFB 5	TOF	Create turn-off delay
SFB 7	TIMEMESS	Time measurement
SFB 12	BSEND	Sending data in blocks
SFB 13	BRCV	Receiving data in blocks:
SFB 14	GET	Remote CPU read
SFB 15	PUT	Remote CPU write
SFB 31	NOTIFY8P	Messages without acknowledge display (8x)
SFB 32	DRUM	Realize a step-by-step switch
SFB 33	ALARM	Messages with acknowledgement display
SFB 34	ALARM_8	Messages without associated values (8x)
SFB 35	ALARM_8P	Messages with associated values (8x)
SFB 36	NOTIFY8	Messages without acknowledgement display
SFB 52	RDREC	Read record set
SFB 53	WRREC	Write record set
SFB 54	RALRM	Receiving an interrupt from a periphery module
SFB 238	EC_RWOD	Function is used internally
SFB 239	FUNC	Function is used internally
SFB 240	DPRAM	Function is used internally

SFC	Name	Description
SFC 0	SET_CLK	Set system clock
SFC 1	READ_CLK	Read system clock
SFC 2	SET_RTM	Set run-time meter
SFC 3	CTRL_RTM	Control run-time meter
SFC 4	READ_RTM	Read run-time meter
SFC 5	GADR_LGC	Logical address of a channel
SFC 6	RD_SINFO	Read start information
SFC 7	DP_PRAL	Triggering a hardware interrupt on the DP master
SFC 12	D_ACT_DP	Activating and deactivating of DP slaves
SFC 13	DPNRM_DG	Read diagnostic data of a DP salve
SFC 14	DPRD_DAT	Read consistent data
SFC 15	DPWR_DAT	Write consistent data
SFC 17	ALARM_SQ	ALARM_SQ
SFC 18	ALARM_SQ	ALARM_S
SFC 19	ALARM_SC	Acknowledgement state last alarm
SFC 20	BLKMOV	Block move
SFC 21	FILL	Fill a field

SFC	Name	Description
SFC 22	CREAT_DB	Create a data block
SFC 23	DEL_DB	Deleting a data block
SFC 24	TEST_DB	Test data block
SFC 25	COMPRESS	Compressing the User Memory
SFC 28	SET_TINT	Set time-of-day interrupt
SFC 29	CAN_TINT	Cancel time-of-day interrupt
SFC 30	ACT_TINT	Activate time-of-day interrupt
SFC 31	QRY_TINT	Query time-of-day interrupt
SFC 32	SRT_DINT	Start time-delay interrupt
SFC 33	CAN_DINT	Cancel time-delay interrupt
SFC 34	QRY_DINT	Query time-delay interrupt
SFC 36	MSK_FLT	Mask synchronous errors
SFC 37	MSK_FLT	Unmask synchronous errors
SFC 38	READ_ERR	Read error register
SFC 39	DIS_IRT	Disabling interrupts
SFC 40	EN_IRT	Enabling interrupts
SFC 41	DIS_AIRT	Delaying interrupts
SFC 42	EN_AIRT	Enabling delayed interrupts
SFC 43	RE_TRIGR	Re-trigger the watchdog
SFC 44	REPL_VAL	Replace value to ACCU1
SFC 46	STP	STOP the CPU
SFC 47	WAIT	Delay the application program
SFC 49	LGC_GADR	Read the slot address
SFC 51	RDSYSST	Read system status list SSL
SFC 52	WR_USMSG	Write user entry into diagnostic buffer
SFC 53	µS_TICK	Time measurement
SFC 54	RD_DPARM	Reading predefined parameters
SFC 55	WR_PARM	Write dynamic parameter
SFC 56	WR_DPARM	Write default parameter
SFC 57	PARM_MOD	Parametrize module
SFC 58	WR_REC	Write record set
SFC 59	RD_REC	Read record set
SFC 64	TIME_TCK	Read system time tick
SFC 65	X_SEND	Sending data
SFC 66	X_RCV	Receiving data
SFC 67	X_GET	Read data

SFC	Name	Description
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 75	SET_ADDR	Set PROFIBUS MAC 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
SFC 193	AI_OSZI	Oscilloscope-/FIFO function
SFC 194	DP_EXCH	Data exchange with CP 342S
SFC 195	FILE_ATT	Change file attributes
SFC 208	FILE_OPN	Open file
SFC 209	FILE_CRE	Create file
SFC 210	FILE_CLO	Close file
SFC 211	FILE_RD	Read file
SFC 212	FILE_WR	Write file
SFC 213	FILE_SEK	Position pointer
SFC 214	FILE_REN	Rename file
SFC 215	FILE_DEL	Delete file
SFC 216	SER_CFG	Parametrization PtP
SFC 217	SER_SND	Send to PtP
SFC 218	SER_RCV	Receive from PtP
SFC 219	CAN_TLGR	CANopen communication
SFC 227	TD_PRM	Parameterization of a text display
SFC 253	IBS_ACC	IBS communication
SFC 254	RW_SBUS	IBS communication

C SSL partial list

More information about this may be found in the manual "SPEED7 Operation List" from Yaskawa.

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
xy74h	Status of the LEDs
xy91h	Status information CPU
xy92h	Stations status information (DPM)
xy94h	Stations status information (DPM, PROFINET-IO and EtherCAT)
xy96h	Module status information (PROFIBUS DP, PROFINET-IO, EtherCAT)
xyA0h	Diagnostic buffer of the CPU
xyB1h	Module diagnostic information (record set 0)
xyB2h	Module diagnostic information (record set 1) via physical address
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