VIPA System 300S+

CPU | 314-6CF23 | Manual

HB140 | CPU | 314-6CF23 | GB | 16-36

SPEED7 CPU 314ST

VIPA
A YASKAWA COMPANY

VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telephone: +49 9132 744-0
Fax: +49 9132 744-1864
Email: info@vipa.com
Internet: www.vipa.com

313-6CF23_000_CPU 314ST,1,GB -

Table of contents

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

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

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.com

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

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

CE Conformity Declaration

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

Conformity Information

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

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

EMail: documentation@vipa.de

Technical support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Tel.: +49 9132 744-1150 (Hotline)

EMail: support@vipa.de

1.2 About this manual

Objective and contents

This manual describes the SPEED7 CPU-SC 314-6CF23 of the System 300S from VIPA. It contains a description of the construction, project implementation and usage.

Product	Order number	as of state:		
		CPU-HW	CPU-FW	DPM-FW
CPU 314ST	314-6CF23	01	V3.7.5	V3.3.5

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

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

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

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

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

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

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

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

2.2 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 314-6CF23

Overview

The CPU 314-6CF23 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.
- The CPU has a parallel SPEED-Bus that enables the additional connection of up to 10 modules from the SPEED-Bus periphery. While the standard peripheral modules are plugged at the right side of the CPU, the SPEED-Bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.
- The CPU has digital and analog input output components. If there is no hardware configuration available, the in- and output areas starting with address 1024 are mapped to the address range of the CPU.

  The following components are integrated:
  - Analog input: 4x12Bit, 1xPt100
  - Analog output: 2x12Bit
  - Digital input: 8xDC 24V with interrupt capability, 4 counter
  - Digital input/output: 8xDC 24V, 0.5A
- Modules and CPUs of the System 300S from VIPA 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 memory card.
- The CPU 314-6CF23 is configured as CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3) from Siemens.

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 2MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 512kbyte
  - There is the possibility to extend the work memory to its maximum printed capacity 2MB by means of a memory extension card.

SPEED-Bus

- The SPEED-Bus is a 32bit parallel bus developed from VIPA.
- Via the SPEED-Bus you may connect up to 10 SPEED-Bus modules to your CPU.
- In opposite to the "standard" backplane bus where the modules are plugged-in at the right side of the CPU by means of single bus connectors, the modules at the SPEED-Bus are plugged-in at the left side of the CPU via a special SPEED-Bus rail.
- VIPA delivers profile rails with integrated SPEED-Bus for 2, 6, or 10 SPEED-Bus peripheral modules with different lengths.

Each SPEED-Bus rail has a slot for an external power supply. The deployment of this external power supply at the CPU 314-6CF23 is not permitted.

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 VIPA SPEEDBUS.GSD.

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		

Conformity and approval		
UL		Refer to Technical data
others		
RoHS	2011/65/EU	Product is lead-free; 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, installation class 3 *

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

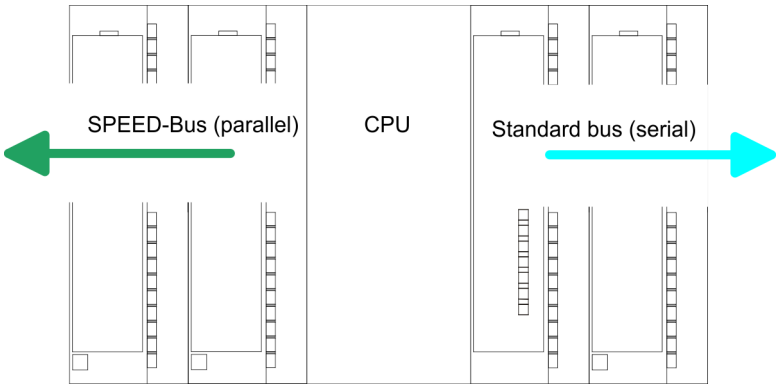
3 Assembly and installation guidelines

3.1 Overview

General

This CPU is provided with a parallel SPEED-Bus that enables the additional connection of up to 10 modules from the SPEED-Bus periphery. While the standard peripheral modules are plugged-in at the right side of the CPU, the SPEED-Bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.

VIPA delivers profile rails with integrated SPEED-Bus for 2, 6 or 10 SPEED-Bus peripheral modules with different lengths.



Serial Standard bus

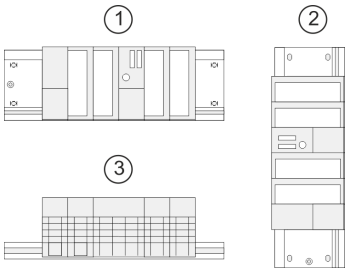
The single modules are directly installed on a profile rail and connected via the backplane bus coupler. Before installing the modules you have to clip the backplane bus coupler to the module from the backside. The backplane bus couplers are included in the delivery of the peripheral modules.

Parallel SPEED-Bus

With SPEED-Bus the bus connection happens via a SPEED-Bus rail integrated in the profile rail at the left side of the CPU. Due to the parallel SPEED-Bus not all slots must be occupied in sequence.

Assembly possibilities

You may assemble the System 300 horizontally, vertically or lying. Please regard the allowed environment temperatures:



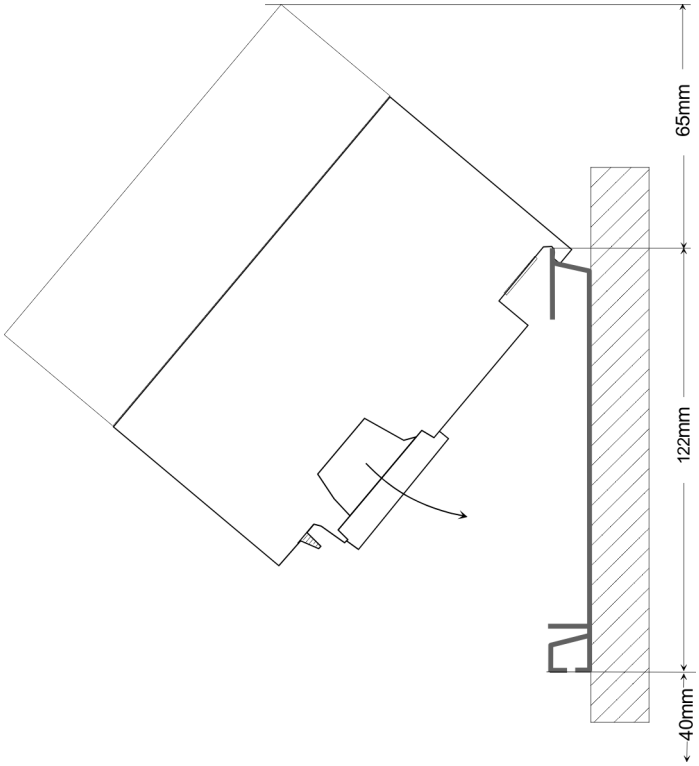
1 horizontal assembly: from 0 to 60°C
2 vertical assembly: from 0 to 50°C
3 lying assembly: from 0 to 55°C

3.2 Installation dimensions

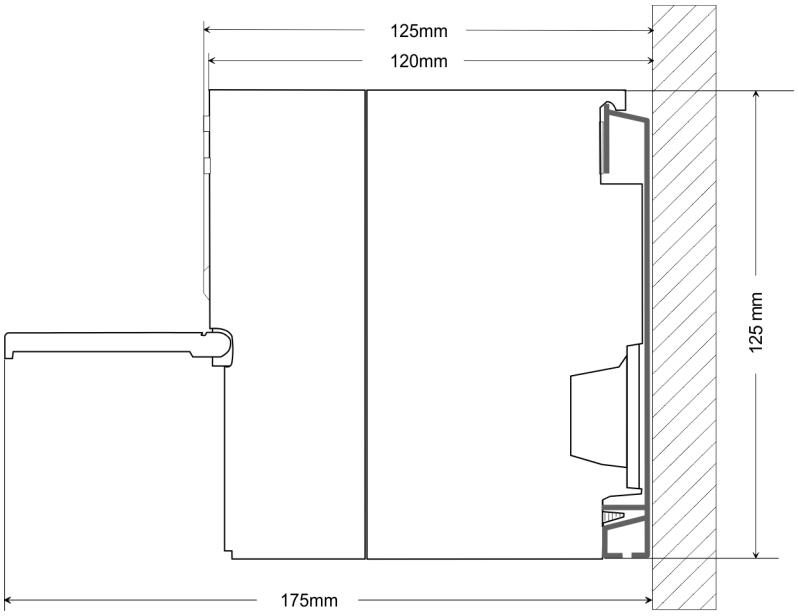
Dimensions Basic enclosure

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

Dimensions



Installation dimensions



3.3 Assembly SPEED-Bus

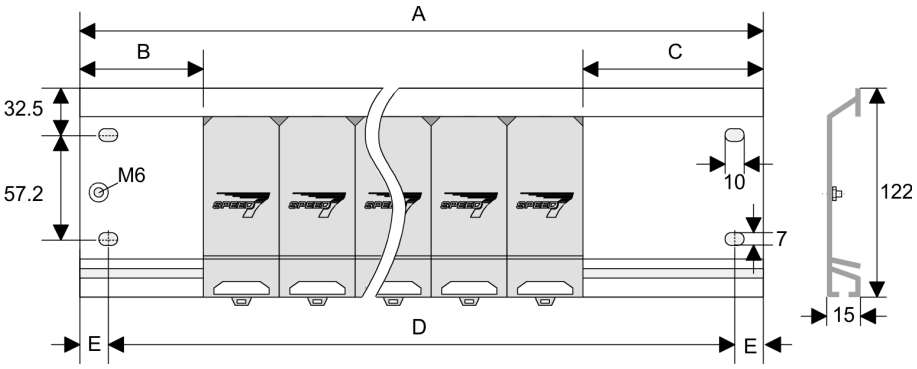
Pre-manufactured SPEED-Bus profile rail

For the deployment of SPEED-Bus modules, a pre-manufactured SPEED-Bus rail is required. This is available mounted on a profile rail with 2, 6 or 10 extension slots.

Dimensions

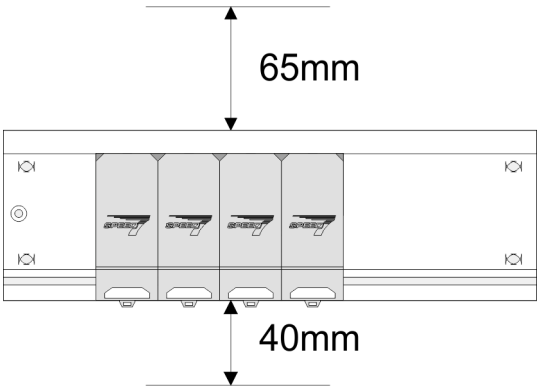
Order number	Number of modules SPEED-Bus/Standard bus	A	B	C	D	E
391-1AF10	2/6	530	100	268	510	10
391-1AF30	6/2	530	100	105	510	10
391-1AF50	10/0	530	20	20	510	10
391-1AJ10	2/15	830	22	645	800	15
391-1AJ30	6/11	830	22	480	800	15
391-1AJ50	10/7	830	22	320	800	15

Measures in mm

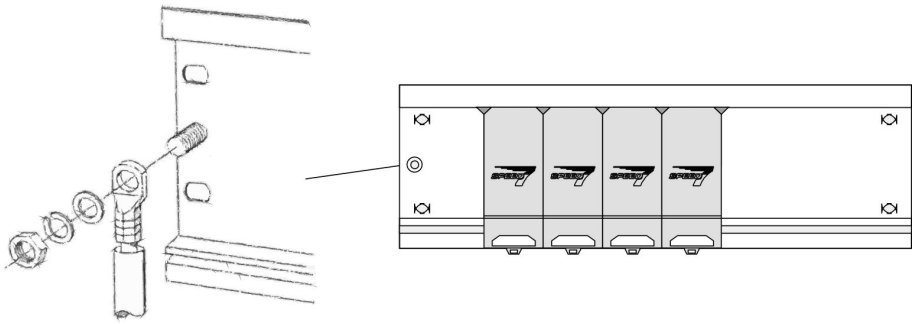


Installation of the profile rail

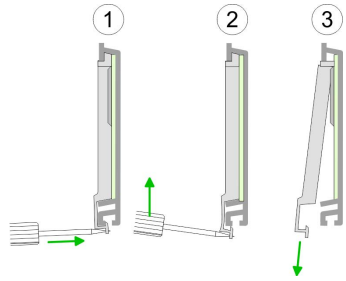
1. Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail. Please look for a low-impedance connection between profile rail and background.



2. Connect the profile rail with the protected earth conductor. The minimum cross-section of the cable to the protected earth conductor has to be 10mm^2.

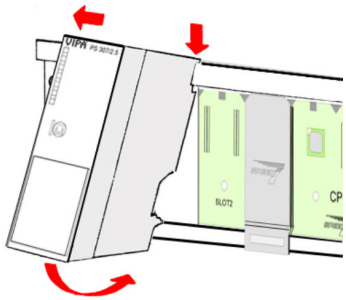


Installation SPEED-Bus module



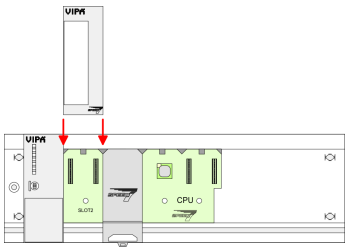
1. Dismantle the according protection flaps of the SPEED-Bus slot with a screw driver (open and pull down).

For the SPEED-Bus is a parallel bus, not every SPEED-Bus slot must be used in series. Leave the protection flap installed at an unused SPEED-Bus slot.



2. At deployment of a DC 24V power supply, install it at the shown position at the profile rail at the left side of the SPEED-Bus and push it to the left to the isolation bolt of the profile rail.

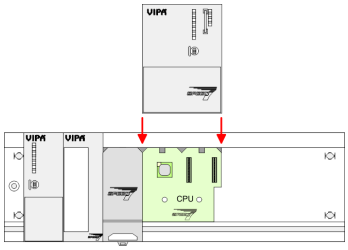
3. Fix the power supply by screwing.



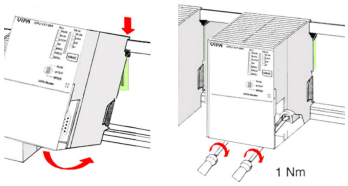
4. To connect the SPEED-Bus modules, plug it between the triangular positioning helps to a slot marked with "SLOT ..." and pull it down.

5. Fix the CPU by screwing.

Installation CPU without Standard-Bus-Modules

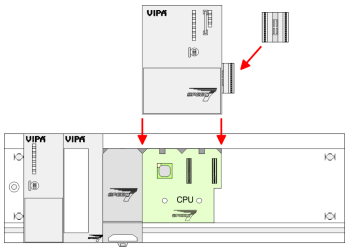


1. To deploy the SPEED7-CPU exclusively at the SPEED-Bus, plug it between the triangular positioning helps to the slot marked with "CPU SPEED7" and pull it down.

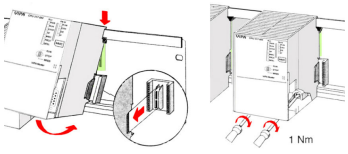


2. Fix the CPU by screwing.

Installation CPU with Standard-Bus-Modules

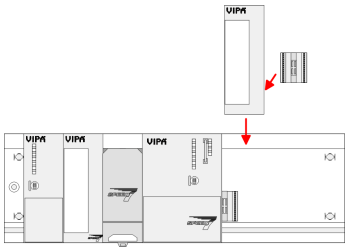


1. If also standard modules shall be plugged, take a bus coupler and click it at the CPU from behind like shown in the picture. Plug the CPU between the triangular positioning helps to the slot marked with "CPU SPEED7" and pull it down.



2. Fix the CPU by screwing.

Installation Standard-Bus-Modules



Repeat this procedure with the peripheral modules, by clicking a backplane bus coupler, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus coupler of the last module and bolt it.

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!

3.4 Assembly standard bus

General

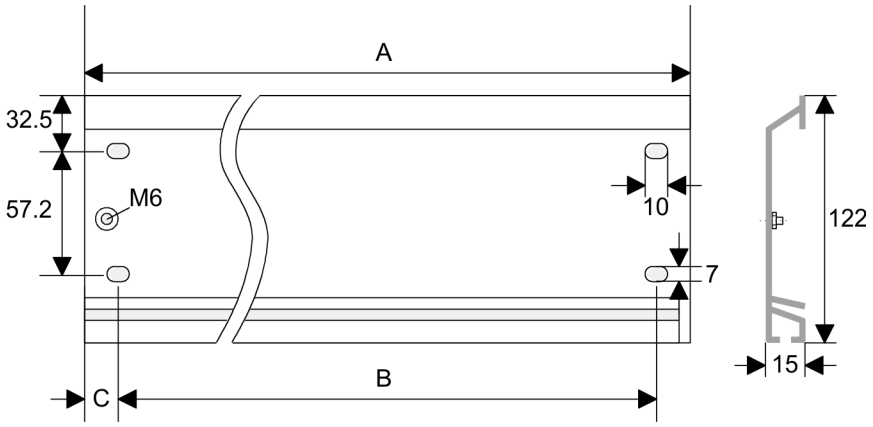
The single modules are directly installed on a profile rail and connected via the backplane bus connector. Before installing the modules you have to clip the backplane bus connector to the module from the backside. The backplane bus connector is delivered together with the peripheral modules.

Profile rail

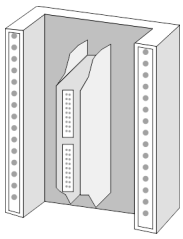
Order number	A	B	C
390-1AB60	160	140	10
390-1AE80	482	466	8.3
390-1AF30	530	500	15
390-1AJ30	830	800	15
390-9BC00*	2000	Drillings only left	15

*) Unit pack: 10 pieces

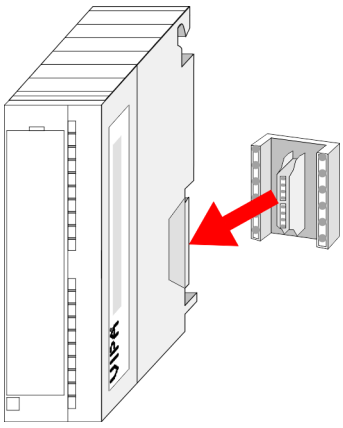
Measures in mm



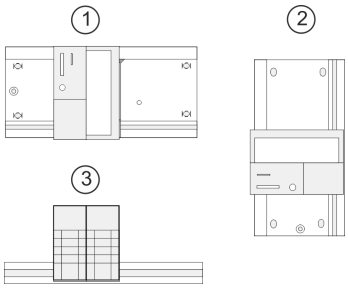
Bus connector



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.



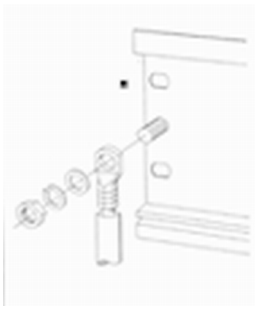
Assembly possibilities



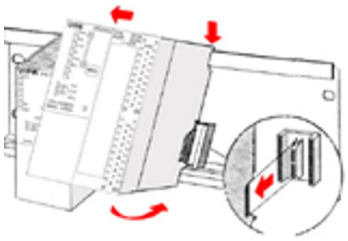
Please regard the allowed environment temperatures:

- horizontal assembly: from 0 to 60°C
- vertical assembly: from 0 to 50°C
- lying assembly: from 0 to 55°C

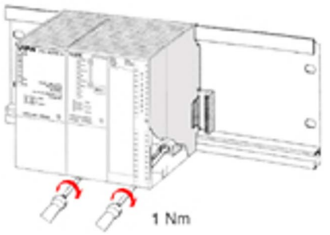
Approach



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.

3.5 Cabling

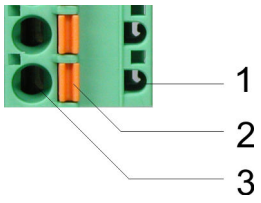
CAUTION!

- 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!

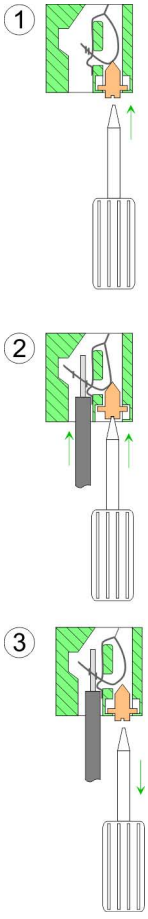
CageClamp technology (green)

For the cabling of power supply of a CPU, a green plug with Cage-Clamp 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

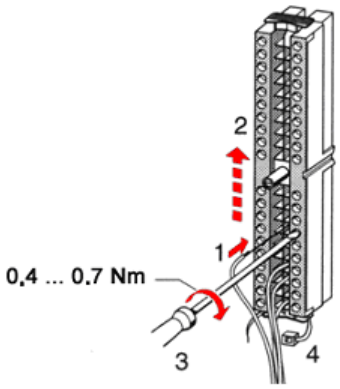
3. By removing the screwdriver the wire is connected safely with the plug connector via a spring.

Front connector I/O periphery

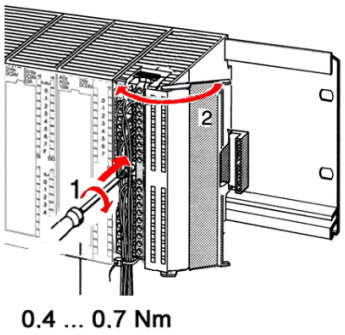
The 40pole front connector with the order number 392-1AM00 is in the delivery.

1. Open the front flap of your I/O module.
2. Bring the front connector in cabling position.

   For this you plug the front connector on the module until it locks. In this position the front connector juts out of the module and has no contact yet.
3. De-isolate your wires. If needed, use core end cases.
4. If you want to lead out your cables from the bottom of the module, start with the cabling from bottom to top, res. from top to bottom, if the cables should be led out at the top.
5. Bolt also the connection screws of not cabled screw clamps.



6. Put the included cable binder around the cable bundle and the front connector.
7. Fix the cable binder for the cable bundle.



8. Bolt the fixing screw of the front connector.
9. Now the front connector is electrically connected with your module.
10. Close the front flap.
11. Fill out the labeling strip to mark the single channels and push the strip into the front flap.

3.6 Installation guidelines

General

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

What does EMC mean?

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

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

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

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

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

There are:

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

Basic rules for EMC

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

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

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

Isolation of conductors

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

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  - the conduction of a potential compensating line is not possible.
  - analog signals (some mV respectively µA) are transferred.
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.

- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

CAUTION!

Please regard at installation!

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

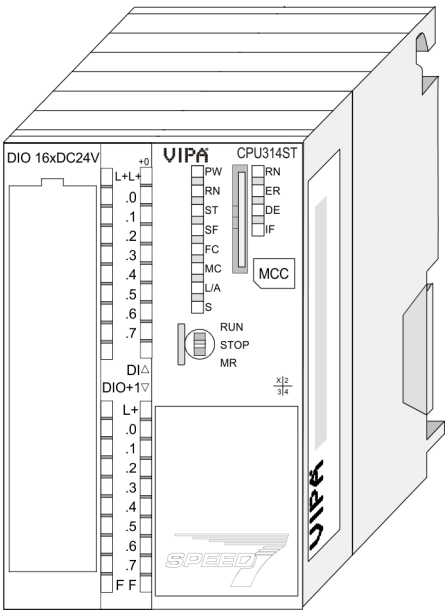
Remedy: Potential compensation line

4 Hardware description

4.1 Properties

CPU 314-6CF23

- SPEED7 technology and SPEED-Bus integrated
- 512kbyte work memory integrated (256kbyte code, 256kbyte data)
- Work memory expandable to max. 2Mbyte (1Mbyte code, 1Mbyte data)
- 2Mbyte load memory
- PROFIBUS DP master integrated (DP-V0, DP-V1)
- 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
- Fast digital I/Os: DI 8xDC24V / DIO 8xDC 24V, 0.5A
- Analog I/Os: AI 4x12Bit / AO 2x12Bit / AI 1xRTD
- 4 counter (100kHz)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte



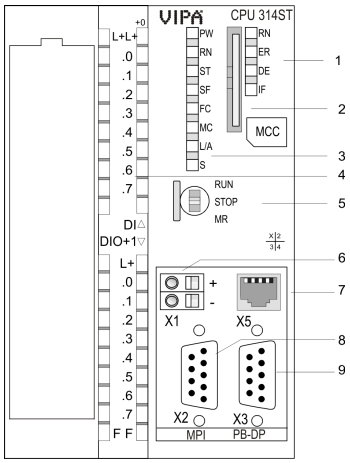
Ordering data

Type	Order number	Description
CPU 314ST	314-6CF23	SPEED-Bus, MPI interface, card slot, real time clock, Ethernet interface for PG/OP, PROFIBUS DP master, DI 8xDC24V / DIO 8xDC24V, 0.5A, AI 4x12Bit, U, I / AO 2x12Bit, U, I / AI 1xRTD, 4 Counter

4.2 Structure

4.2.1 General

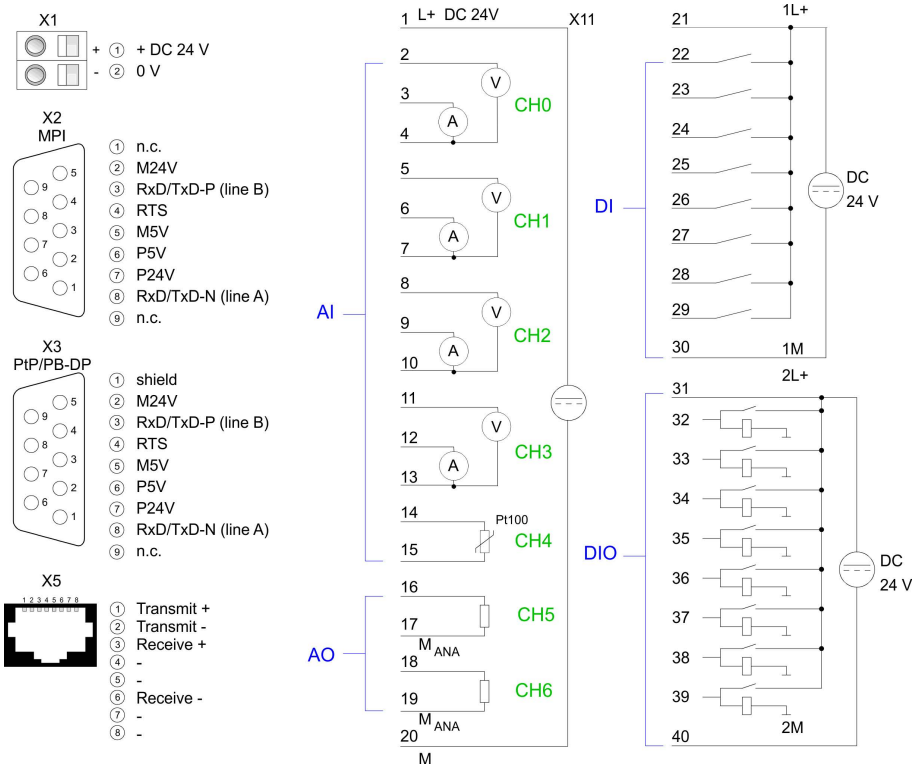
CPU 314-6CF23



1 LEDs of the integrated PROFIBUS DP master
2 Storage media slot (lockable)
3 LEDs of the CPU part
4 LEDs of the I/O part
5 Operating mode switch CPU
6 X1: Slot for DC 24V power supply
7 X5: Ethernet PG/OP channel
8 X2: MPI interface
9 X3: PB-DP/PtP interface

The components 6 - 9 are under the front flap!

4.2.2 Interfaces



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.

Each SPEED-Bus rail has a slot for an external power supply. This allows you to raise the maximum current at the back plane bus. The deployment of this external power supply at the CPU 314-6CF23 is not permitted!

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.

4.2.3 Memory management

Memory

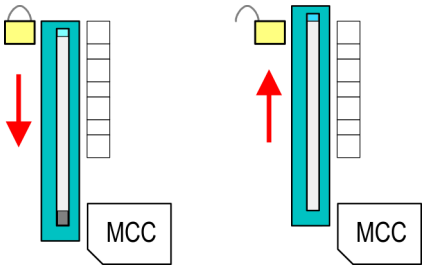
The CPU has an integrated memory. Information about the capacity of the memory may be found at the front of the CPU. The memory is divided into the following parts:

- Load memory 2MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 512kbyte
  - There is the possibility to extend the work memory to its maximum printed capacity 2MB by means of a memory extension card.

4.2.4 Storage media slot

- Via this slot as external storage medium for applications and firmware you may use a memory card (MMC respectively SD).
- The VIPA storage media 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 storage medium 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.

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



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!

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

As soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.

RN (RUN)	ST (STOP)	SF (SFAIL)	FC (FRCE)	MC (MMC)	Meaning
green	yellow	red	yellow	yellow	
Boot-up after PowerON					
●	BB*	●	●	●	* Blinking with 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.
BB	●	X	X	X	CPU is in start-up state, the RUN LED blinks during operating OB100 at least for 3s.
●	○	○	X	X	CPU is in state RUN without error.
X	X	●	X	X	There is a system fault. More information may be found in the diagnostics buffer of the CPU.
X	X	X	●	X	Variables are forced.
X	X	X	X	●	Access to the memory card.
X	BB*	○	○	○	* Blinking with 10Hz: Configuration is loaded.
Overall reset					
○	BB	X	X	X	Overall reset is requested.
○	BB*	X	X	X	* Blinking with 10Hz: Overall reset is executed.
Factory reset					
●	●	○	○	○	Factory reset is executed.
○	●	●	●	●	Factory reset finished without error.

RN (RUN)	ST (STOP)	SF (SFAIL)	FC (FRCE)	MC (MMC)	Meaning
Firmware update					
○	●	BB	BB	●	The alternate blinking indicates that there is new firmware on the memory card.
○	○	BB	BB	●	The alternate blinking indicates that a firmware update is executed.
○	●	●	●	●	Firmware update finished without error.
○	BB*	BB*	BB*	BB*	* Blinking with 10Hz: Error during Firmware update.

on: ● | off: ○ | blinking (2Hz): BB | not relevant: X

LEDs Ethernet PG/OP channel L/A, S

The green L/A-LED (Link/Activity) indicates the physical connection of the Ethernet PG/OP channel to Ethernet. Irregular flashing of the L/A-LED indicates communication of the Ethernet PG/OP channel via Ethernet.

If the green S-LED (Speed) is on, the Ethernet PG/OP has a communication speed of 100MBit/s otherwise 10MBit/s.

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)	ER (ERR)	DE	IF	Meaning
green	red	green	red	
○	○	○	○	Master has no project, this means the interface is deactivated respectively PtP is active.
●	○	○	○	Master has bus parameters and is in RUN without slaves.
●	○	BB	○	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, at least 1 slave is missing.
●	●	BB	○	CPU is in STOP, at least 1 slave is missing.
○	○	○	●	Initialization error at faulty parametrization.
○	●	○	●	Waiting state for start command from CPU.

on: ● | off: ○ | blinking (2Hz): BB

Slave operation

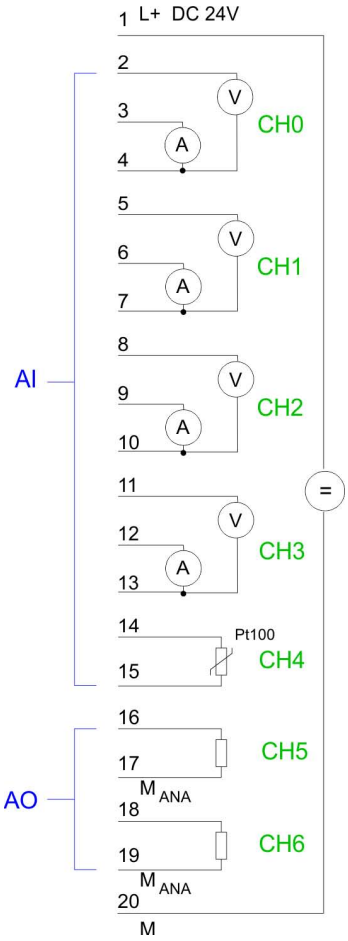
RN (RUN)	ER (ERR)	DE	IF	Meaning
green	red	green	red	
○	○	○	○	Slave has no project respectively PtP is active.
BB	○	○	○	Slave is without master.
BB*	○	BB*	○	* Alternate flashing at configuration faults.
●	○	●	○	Slave exchanges data between master.
on: ● | off: ○ | blinking (2Hz): BB				

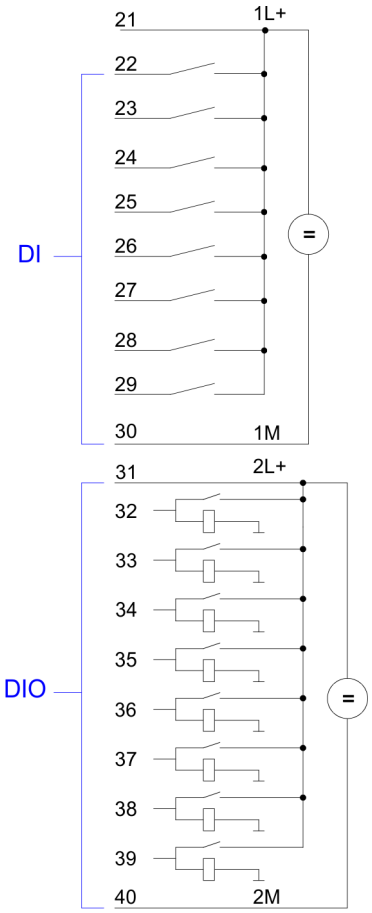
4.2.8 In-/Output range CPU 314-6CF23

Overview CPU 314-6CF23

The CPU 314-6CF23 has the following analog and digital in- and output ranges integrated in one casing:

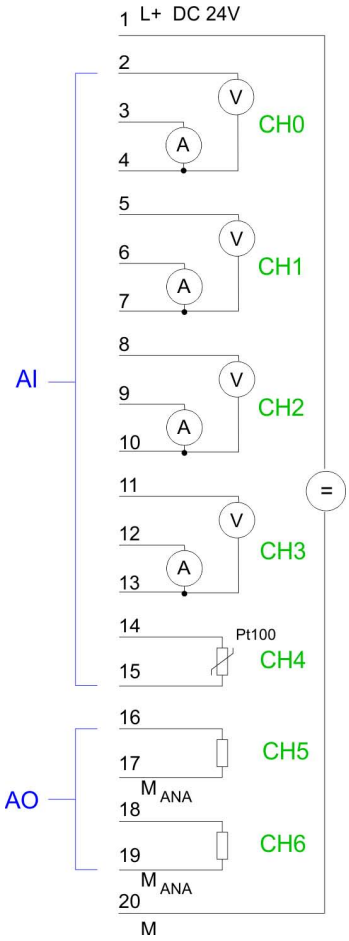
- Analog input: 4x12Bit, 1xPt100
- Analog output: 2x12Bit
- Digital input: 8xDC 24V, interrupt capable, 4 counter
- Digital in-/output: 8xDC 24V, 0.5A



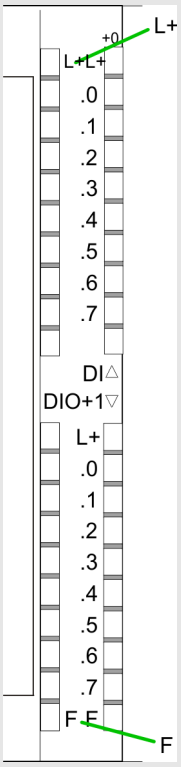


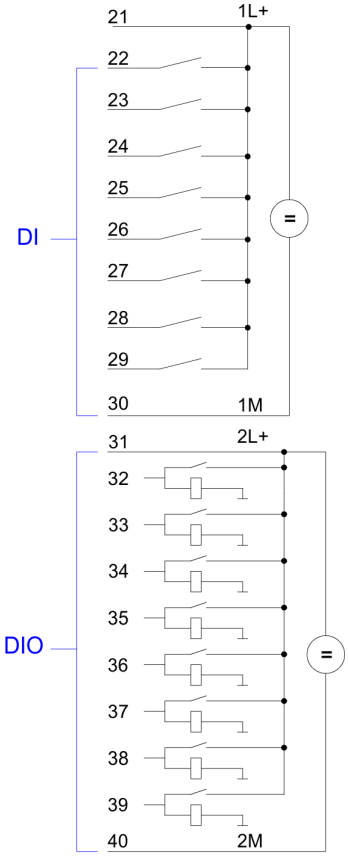
CAUTION!

Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+. Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal. A thus connected output remains active even with shut down supply voltage. Non-observance may cause damages of the module.



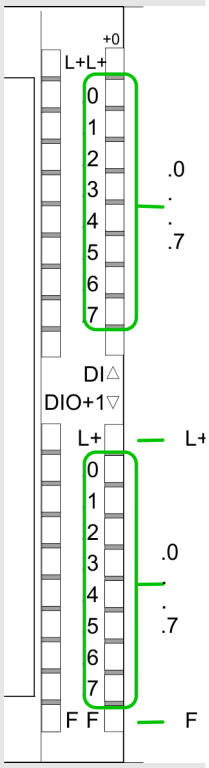
CPU 314-6CF23: Analog part pin assignment and status indicator

Pin	Assignment	LEDs	Description
1	Power supply DC 24V AIO		**1L+** LED (green) Supply voltage available **F** LED (red) Sum error
2	Voltage measurement channel 0		
3	Current measurement channel 0		
4	Ground channel 0		
5	Voltage measurement channel 1		
6	Current measurement channel 1		
7	Ground channel 1		
8	Voltage measurement channel 2		
9	Current measurement channel 2		
10	Ground channel 2		
11	Voltage measurement channel 3		
12	Current measurement channel 3		
13	Ground channel 3		
14	Pt 100 channel 4		
15	Pt 100 channel 4		
16	Output + channel 5		
17	Ground output channel 5		
18	Output + channel 6		
19	Ground output channel 6		
20	Ground power supply AIO		



CPU 314-6CF23: Digital part pin assignment and status indicator

Pin	Assignment
1	Power supply +DC 24 V DI
2	I+0.0 / Counter 0(A)
3	I+0.1 / Counter 0(B)
4	I+0.2 / Gate0/Latch0/Reset0
5	I+0.3 / Counter 1(A)
6	I+0.4 / Counter 1(B)
7	I+0.5 / Gate1/Latch1/Reset1
8	I+0.6 / Counter 2(A)
9	I+0.7 / Counter 2(B)
10	Ground DI
11	Power supply +DC 24 V DIO
12	I/Q+1.0 / Gate2/Latch2/Reset2
13	I/Q+1.1 / Counter 3(A)
14	I/Q+1.2 / Counter 3(B)
15	I/Q+1.3 / Gate3/Latch3/Reset3
16	I/Q+1.4 / OUT0/Latch0/Reset0
17	I/Q+1.5 / OUT1/Latch1/Reset1
18	I/Q+1.6 / OUT2/Latch2/Reset2
19	I/Q+1.7 / OUT3/Latch3/Reset3
20	Ground DIO



LEDs	Description
	DI:
	.0 7
	LED (green)
	I+0.0 to I+0.7
	starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated
	DIO:
	2L+
	LED (green)
	Supply voltage available for DIO
	.0 7
	LED (green)
	I/Q+1.0 to I/Q +1.7
	on at active output/input
	F
	LED (red)
	Overload or short circuit error

4.3 Technical data

Order no.	314-6CF23
Type	CPU 314ST/DPM
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)	300 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	-

Order no.	314-6CF23
Power loss	14 W
Technical data digital inputs	
Number of inputs	8
Cable length, shielded	1000 m
Cable length, unshielded	600 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	✓
Current consumption from load voltage L+ (without load)	70 mA
Rated value	DC 24 V
Input voltage for signal "0"	DC 0...5 V
Input voltage for signal "1"	DC 15...28.8 V
Input voltage hysteresis	-
Frequency range	-
Input resistance	-
Input current for signal "1"	6 mA
Connection of Two-Wire-BEROs possible	✓
Max. permissible BERO quiescent current	1.5 mA
Input delay of "0" to "1"	parameterizable 2.56µs - 40ms
Input delay of "1" to "0"	parameterizable 2.56µs - 40ms
Number of simultaneously utilizable inputs horizontal configuration	8
Number of simultaneously utilizable inputs vertical configuration	8
Input characteristic curve	IEC 61131-2, type 1
Initial data size	34 Byte
Technical data digital outputs	
Number of outputs	8
Cable length, shielded	1000 m
Cable length, unshielded	600 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	-
Current consumption from load voltage L+ (without load)	30 mA

Order no.	314-6CF23
Total current per group, horizontal configuration, 40°C	4 A
Total current per group, horizontal configuration, 60°C	3 A
Total current per group, vertical configuration	3 A
Output voltage signal "1" at min. current	L+ (-0.8 V)
Output voltage signal "1" at max. current	L+ (-0.8 V)
Output current at signal "1", rated value	0.5 A
Output current, permitted range to 40°C	5 mA to 0.6 A
Output current, permitted range to 60°C	5 mA to 0.6 A
Output current at signal "0" max. (residual current)	100 µA
Output delay of "0" to "1"	100 µs
Output delay of "1" to "0"	100 µs
Minimum load current	-
Lamp load	5 W
Parallel switching of outputs for redundant control of a load	possible
Parallel switching of outputs for increased power	not possible
Actuation of digital input	✓
Switching frequency with resistive load	max. 2.5 kHz
Switching frequency with inductive load	max. 0.5 Hz
Switching frequency on lamp load	max. 2.5 kHz
Internal limitation of inductive shut-off voltage	L+ (-52 V)
Short-circuit protection of output	yes, electronic
Trigger level	1 A
Number of operating cycle of relay outputs	-
Switching capacity of contacts	-
Output data size	18 Byte
Technical data analog inputs	
Number of inputs	5
Cable length, shielded	200 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	✓

Order no.	314-6CF23
Current consumption from load voltage L+ (without load)	85 mA
Voltage inputs	✓
Min. input resistance (voltage range)	120 kΩ
Input voltage ranges	-10 V ... +10 V 0 V ... +10 V
Operational limit of voltage ranges	+/-0.3%
Operational limit of voltage ranges with SFU	-
Basic error limit voltage ranges	+/-0.3%
Basic error limit voltage ranges with SFU	-
Destruction limit voltage	max. 15V
Current inputs	✓
Max. input resistance (current range)	85 Ω
Input current ranges	-20 mA ... +20 mA 0 mA ... +20 mA +4 mA ... +20 mA
Operational limit of current ranges	+/-0.3%
Operational limit of current ranges with SFU	-
Basic error limit current ranges	+/-0.2%
Radical error limit current ranges with SFU	-
Destruction limit current inputs (electrical current)	max. 50mA
Destruction limit current inputs (voltage)	max. 15V
Resistance inputs	✓
Resistance ranges	0 ... 600 Ohm
Operational limit of resistor ranges	+/-0.4%
Operational limit of resistor ranges with SFU	-
Basic error limit	+/-0.2%
Basic error limit with SFU	-
Destruction limit resistance inputs	max. 15V
Resistance thermometer inputs	✓
Resistance thermometer ranges	Pt100 Pt1000 Ni100 Ni1000

Order no.	314-6CF23
Operational limit of resistance thermometer ranges	+/-0.6%
Operational limit of resistance thermometer ranges with SFU	-
Basic error limit thermoresistor ranges	+/-0.4%
Basic error limit thermoresistor ranges with SFU	-
Destruction limit resistance thermometer inputs	max. 15V
Thermocouple inputs	-
Thermocouple ranges	-
Operational limit of thermocouple ranges	-
Operational limit of thermocouple ranges with SFU	-
Basic error limit thermoelement ranges	-
Basic error limit thermoelement ranges with SFU	-
Destruction limit thermocouple inputs	-
Programmable temperature compensation	-
External temperature compensation	-
Internal temperature compensation	-
Technical unit of temperature measurement	°C
Resolution in bit	12
Measurement principle	Sigma-Delta
Basic conversion time	6 ms
Noise suppression for frequency	80 dB
Initial data size	10 Byte
Technical data analog outputs	
Number of outputs	2
Cable length, shielded	200 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	✓
Current consumption from load voltage L+ (without load)	-
Voltage output short-circuit protection	-
Voltage outputs	✓
Min. load resistance (voltage range)	1 kΩ

Order no.	314-6CF23
Max. capacitive load (current range)	1 µF
Max. inductive load (current range)	30 mA
Output voltage ranges	-10 V ... +10 V 0 V ... +10 V
Operational limit of voltage ranges	+/-0.4%
Basic error limit voltage ranges with SFU	+/-0.3%
Destruction limit against external applied voltage	max. 15V
Current outputs	✓
Max. in load resistance (current range)	500 Ω
Max. inductive load (current range)	10 mH
Typ. open circuit voltage current output	16 V
Output current ranges	-20 mA ... +20 mA 0 mA ... +20 mA +4 mA ... +20 mA
Operational limit of current ranges	+/-0.4%
Radical error limit current ranges with SFU	+/-0.3%
Destruction limit against external applied voltage	max. 15V
Settling time for ohmic load	0.2 ms
Settling time for capacitive load	0.5 ms
Settling time for inductive load	0.75 ms
Resolution in bit	12
Conversion time	1 ms
Substitute value can be applied	yes
Output data size	4 Byte
Technical data counters	
Number of counters	4
Counter width	32 Bit
Maximum input frequency	100 kHz
Maximum count frequency	100 kHz
Mode incremental encoder	✓
Mode pulse / direction	✓
Mode pulse	✓
Mode frequency counter	-

Order no.	314-6CF23
Mode period measurement	-
Gate input available	✓
Latch input available	✓
Reset input available	✓
Counter output available	✓
Load and working memory	
Load memory, integrated	2 MB
Load memory, maximum	2 MB
Work memory, integrated	512 KB
Work memory, maximal	2 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
Status information, alarms, diagnostics	
Status display	yes
Interrupts	yes
Process alarm	yes, parameterizable
Diagnostic interrupt	yes, parameterizable
Diagnostic functions	yes
Diagnostics information read-out	possible
Supply voltage display	green LED
Group error display	red SF LED
Channel error display	red LED per group
Isolation	
Between channels	✓
Between channels of groups to	8
Between channels and backplane bus	✓

Order no.	314-6CF23
Between channels and power supply	-
Max. potential difference between circuits	DC 75 V/ AC 50 V
Max. potential difference between inputs (Ucm)	-
Max. potential difference between Mana and Mintern (Uiso)	-
Max. potential difference between inputs and Mana (Ucm)	-
Max. potential difference between inputs and Mintern (Uiso)	-
Max. potential difference between Mintern and outputs	-
Insulation tested with	DC 500 V
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	0 .. 512
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	0 .. 512
S7 times remanence adjustable	not retentive
Data range and retentive characteristic	
Number of flags	8192 Byte
Bit memories retentive characteristic adjustable	0 .. 8192
Bit memories retentive characteristic preset	MB0 .. MB15
Number of data blocks	4095
Max. data blocks size	64 KB
Max. local data size per execution level	1024 Byte
Blocks	
Number of OBs	23
Number of FBs	2048
Number of FCs	2048

Order no.	314-6CF23
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
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	no
Address areas (I/O)	
Input I/O address area	8192 Byte
Output I/O address area	8192 Byte
Input process image maximal	2048 Byte
Output process image maximal	2048 Byte
Digital inputs	65536
Digital outputs	65536
Digital inputs central	1032
Digital outputs central	1032
Integrated digital inputs	8
Integrated digital outputs	8
Analog inputs	1024
Analog outputs	1024
Analog inputs, central	261
Analog outputs, central	258
Integrated analog inputs	5
Integrated analog outputs	2
Communication functions	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	4
Size of GD packets, max.	22 Byte
S7 basic communication	✓

Order no.	314-6CF23
S7 basic communication, user data per job	76 Byte
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
S7 communication, user data per job	160 Byte
Number of connections, max.	32
PWM data	
PWM channels	-
PWM time basis	-
Period length	-
Minimum pulse width	-
Type of output	-
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.	314-6CF23
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	
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.	1 KB
Address range outputs, max.	1 KB
User data inputs per slave, max.	244 Byte
User data outputs per slave, max.	244 Byte
Functionality PROFIBUS slave	
PG/OP channel	✓
Routing	✓

Order no.	314-6CF23
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Automatic detection of transmission speed	-
Transfer memory inputs, max.	244 Byte
Transfer memory outputs, max.	244 Byte
Address areas, max.	32
User data per address area, max.	32 Byte
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

Order no.	314-6CF23
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Housing	
Material	PPE
Mounting	Rail System 300
Mechanical data	
Dimensions (WxHxD)	80 mm x 125 mm x 120 mm
Weight	480 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	in preparation
KC certification	in preparation

5 Deployment CPU 314-6CF23

5.1 Assembly

Information about assembly and cabling: Chapter 3 'Assembly and installation guidelines' on page 17

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. Modules at the SPEED-Bus are also taken into account at the automatic address allocation. Here the digital I/Os are stored beginning with address 128 and analog I/Os, FMs and CPs beginning with address 2048.

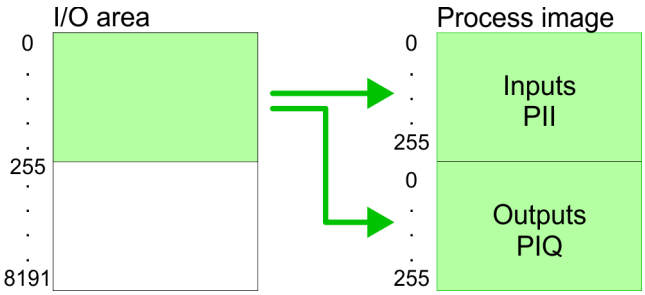
5.3.2 Addressing

Backplane bus periphery

The CPU 314-6CF23 provides an I/O area (address 0 ... 8191) and a process image of the in- and outputs (each address 0 ... 255). The process image stores the signal states of the lower address (0 ... 255) additionally in a separate memory area.

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 314-6CF23.

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 314-6CF23 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.

Further 10 modules at the SPEED-Bus may be connected. CPs and DP masters that are additionally virtual configured at the standard bus are taken into the count of 32 modules at the standard bus.

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.

CAUTION!
Please take care not to configure a double address assignment at connection via external PROFIBUS DP masters - required for project engineering of a SPEED-Bus system! At external DP master systems, the Siemens hardware configurator does not execute an address check!

Automatic addressing

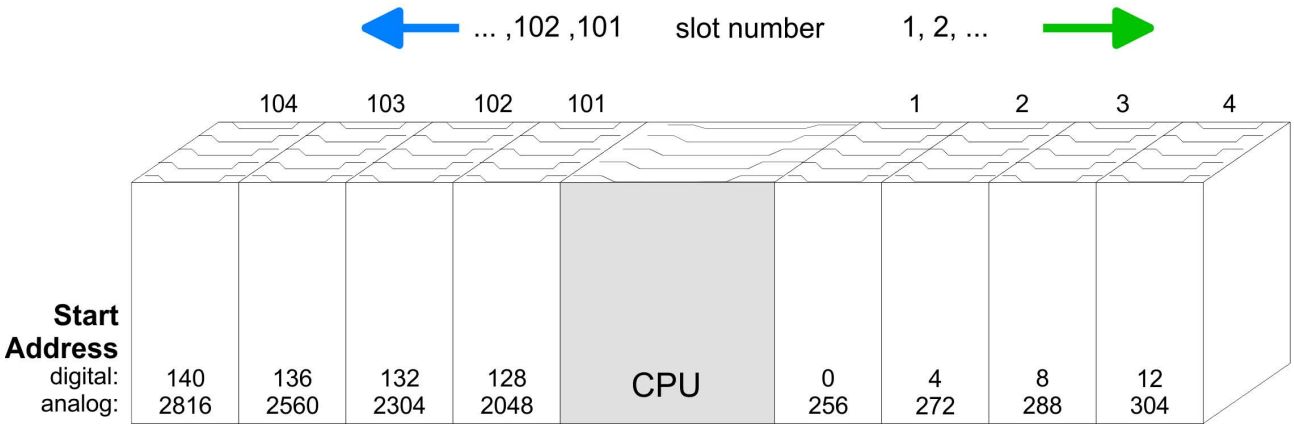
If you do not like to use a hardware configuration, an automatic addressing comes into force. At the automatic address allocation DIOs occupy depending on the slot location always 4byte and AIOs, FMs, CPs always 16byte at the standard bus and 256byte at the SPEED-Bus. Depending on the slot location the start address from where on the according module is stored in the address range is calculated with the following formulas:

Standard-Bus

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

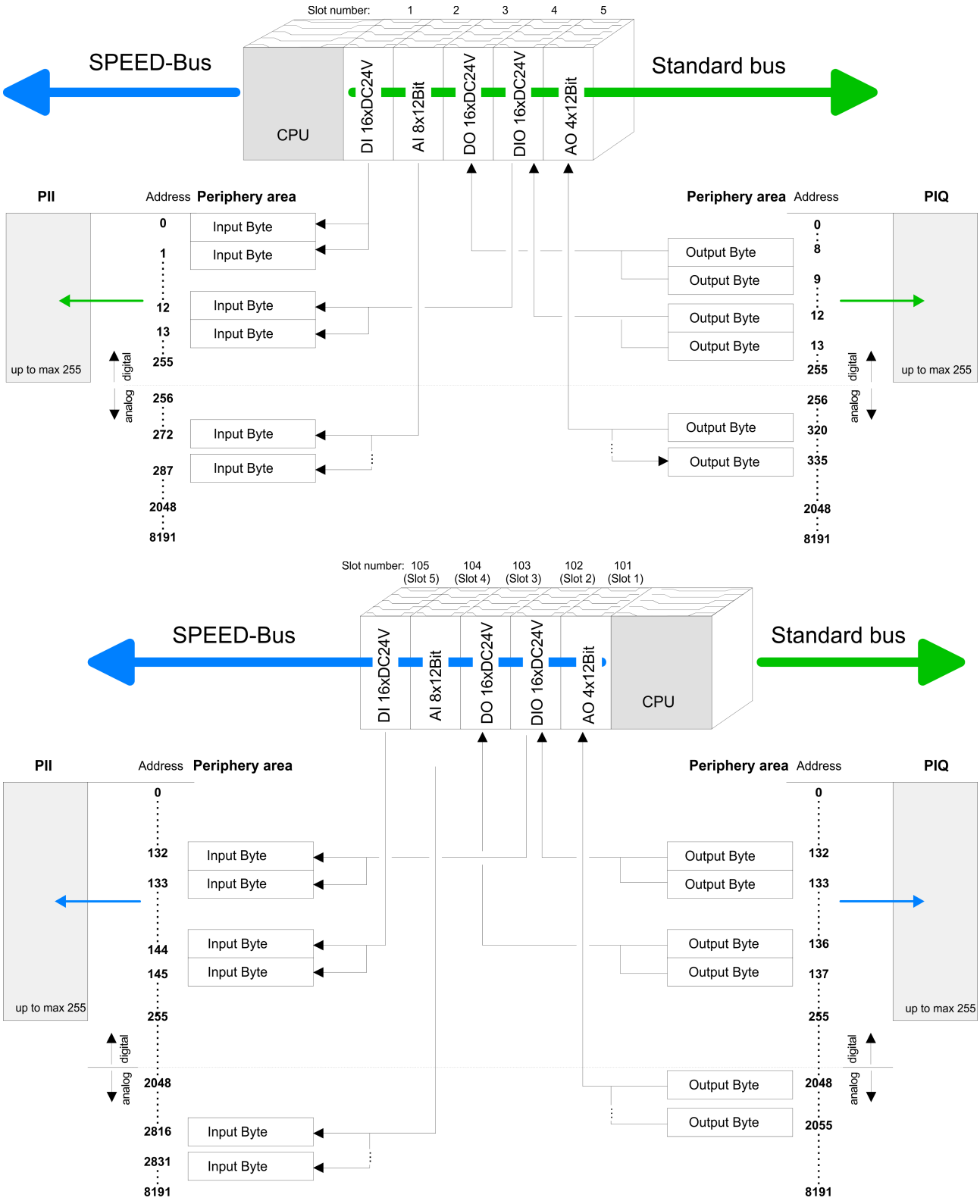
SPEED-Bus

- DIOs: Start address = 4×(slot-101)+128
- AIOs, FMs, CPs: Start address = 256×(slot-101)+2048



Example for automatic address allocation

The following sample shows the functionality of the automatic address allocation separated for standard bus and SPEED-Bus:



5.3.3 Address assignment I/O part

Overview

- By including the SPEEDBUS.GSD in your hardware configurator, the module is at your disposal in the hardware catalog. After the installation of the GSD you will find the CPU 314-6CF23 at *'Additional field devices ➔ I/O ➔ VIPA_SpeedBus'*.
- In case there is no hardware configuration available, the in- and output areas starting at address 1024 are shown in the address range of the CPU.
- For the data input a range of 48byte and for the data output a range of 24byte is available

Input area

Addr.	Name	Byte	Function
+0	DI_0	1	Digital input I+0.0 ... I+0.7
+1	DI_1	1	Digital input I+1.0 ... I+1.7
+2	-	2	reserved
+4	AI_CH0	2	Analog input CH0
+6	AI_CH1	2	Analog input CH1
+8	AI_CH2	2	Analog input CH2
+10	AI_CH3	2	Analog input CH3
+12	AI_CH4	2	Analog input CH4
+14	-	2	reserved
+16	CVCL_0	4	Counter/Latch value 0
+20	-	2	reserved
+22	ISTS_0	2	Input status counter 0
+24	CVCL_1	4	Counter/Latch value 1
+28	-	2	reserved
+30	ISTS_1	2	Input status counter 1
+32	CVCL_2	4	Counter/Latch value 2
+36	-	2	reserved
+38	ISTS_2	2	Input status counter 2
+40	CVCL_3	4	Counter/Latch value 3
+44	-	2	reserved
+46	ISTS_3	2	Input status counter 3

Output area

Addr.	Name	Byte	Function
+0	-	1	reserved
+1	DO_1	1	Digital output Q+1.0 ... Q+1.7
+2	-	2	reserved
+4	AO_CH0	2	Analog output CH0
+6	AO_CH1	2	Analog output CH1
+8	-	2	reserved
+10	OSTS_0	2	Output status counter 0
+12	-	2	reserved
+14	OSTS_1	2	Output status counter 1
+16	-	2	reserved
+18	OSTS_2	2	Output status counter 2
+20	-	2	reserved
+22	OSTS_3	2	Output status counter 3

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	

To be compatible with the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'* number 2 the CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3) from Siemens.
4. The integrated PROFIBUS DP master (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 *CPU-TYPE_318*. After this perform a power cycle.
  - *Chapter 5.19 'CMD - auto commands' on page 95*

```
CMD_START
CPUTYPE_318
CMD_END
```

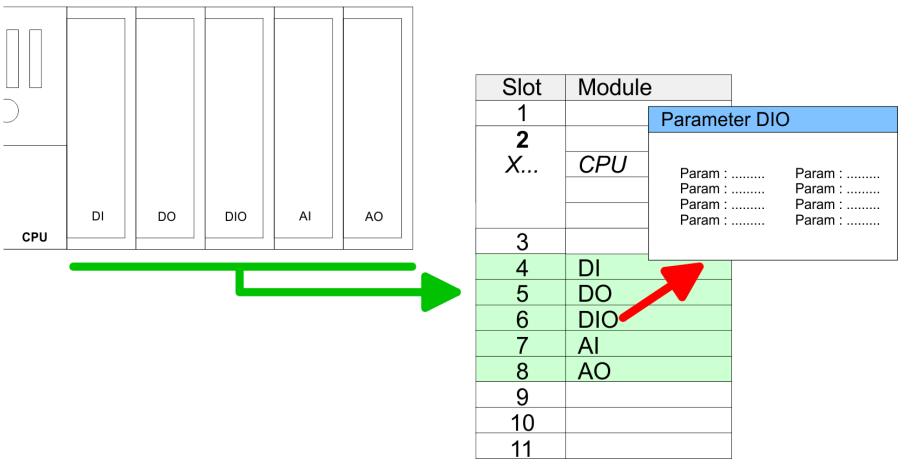
- CPU type original
  - The reset to the original type takes place with the CMD auto command *CPUTYPE_ORIGINAL* respectively by *Chapter 5.15 'Reset to factory settings' on page 91*.
  - *Chapter 5.19 'CMD - auto commands' on page 95*

```
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 314-6CF23 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

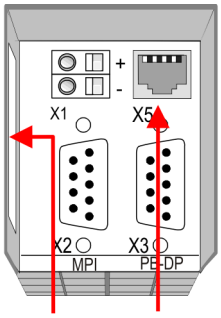
1. Install your System 300S with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the Ethernet jack of the Ethernet PG/OP channel to Ethernet

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**

→ 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 V5.5 + SP1 with the following proceeding:

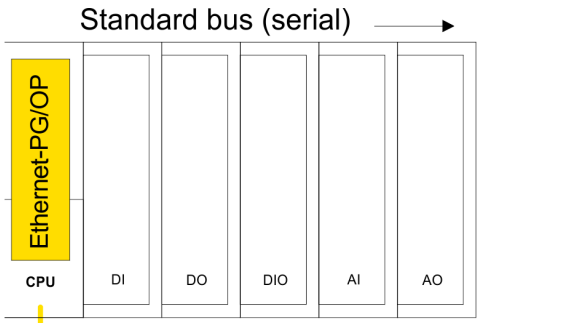
1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the 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 317-2AK14-0AB0 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.



Standard bus

Slot	Module
1	
2 *X...*	**CPU**
3	
4	DI
5	DO
6	DIO
7	AI
8	AO
9	343-1EX11
10	
11	

5.7 Hardware configuration - SPEED-Bus

5.7.1 Preconditions

Since the VIPA specific CPU parameters may be set and the modules at the SPEED-Bus may be configured, the installation of the SPEEDBUS.GSD from VIPA in the hardware catalog is necessary. The CPU and its SPEED-Bus modules may be configured in a PROFIBUS master 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:

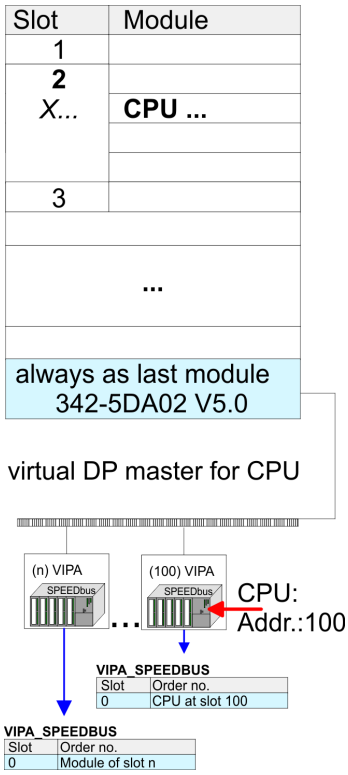
Name	Language
SPEEDBUS.GSD	German (default)
SPEEDBUS.GSG	German
SPEEDBUS.GSE	English

The GSD files may be found at www.vipa.com at the "Service" part.

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

1. Browse to www.vipa.com
2. Click to *'Service ➔ Download ➔ GSD- and EDS-Files ➔ Profibus'*
3. Download the file Cx000023_Vxxx.
4. Extract the file to your work directory. The SPEEDBUS.GSD is stored in the directory VIPA_System_300S.
5. Start the hardware configurator from Siemens.
6. Close every project.
7. Select *'Options ➔ Install new GSD-file'*.
8. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from VIPA may now be found in the hardware catalog at PROFIBUS-DP / Additional field devices / I/O / VIPA_SPEEDBUS.

5.7.2 Proceeding



The embedding of the CPU 314-6CF23 and its modules at the SPEED-Bus happens by means of a virtual PROFIBUS master system with the following approach:

1. Perform a hardware configuration for the CPU. *Chapter 5.4 'Hardware configuration - CPU' on page 57*
2. Since the SPEED-Bus modules are to be linked as a virtual PROFIBUS system, configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Link the DP master to a new PROFIBUS net and switch it to DP master operating mode.
3. To this master system you assign every SPEED-Bus module as "VIPA_SPEEDBUS" slave starting with the CPU. Here the PROFIBUS address corresponds to the slot no. Beginning with 100 for the CPU. Place at slot 0 of every slave the assigned module

Due to the fact that some SPEED-Bus CPs from VIPA are similar in project engineering and parametrization to the corresponding CP from Siemens, for each SPEED-Bus CP a corresponding Siemens CP is to be placed and linked at the standard bus.

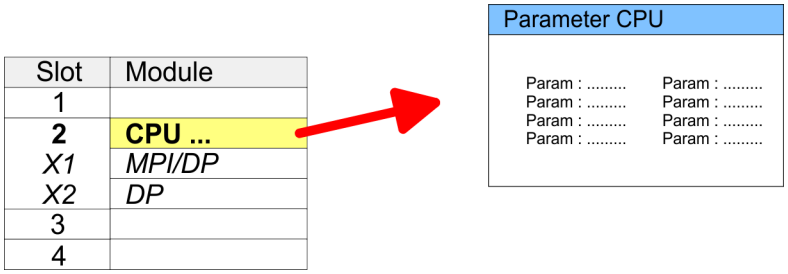
More information about the configuration of the according SPEED-Bus module may be found in the according manual.

5.8 Setting standard CPU parameters

5.8.1 Parameterization via Siemens CPU

Parameterization via Siemens CPU

Since the CPU 314-6CF23 is to be configured as Siemens CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3) in the Siemens hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 317-2DP during hardware configuration. Via a double-click on the CPU 317-2DP 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 317-2AK14-0AB0 V3.3).
- Order No. / Firmware: Order number and firmware are identical to the details in the "hardware catalog" window.
- Name: The Name field provides the short description of the CPU. If you change the name the new name appears in the Siemens SIMATIC Manager.
- Plant designation: Here is the possibility to specify a plant designation for the CPU. This plant designation identifies parts of the plant according to their function. Its structure is hierarchic according to IEC 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 VIPA 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.vipa.com at the "Service" part.

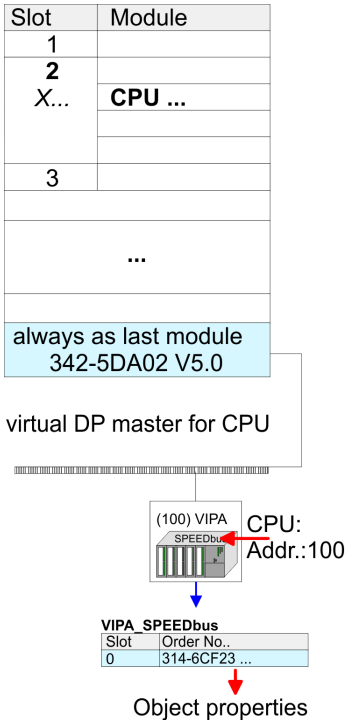
The integration of the SPEEDBUS.GSD takes place with the following proceeding:

1. Browse to www.vipa.com
2. Click to *'Service ➔ Download ➔ GSD- and EDS-Files ➔ Profibus'*
3. Download the file Cx000023_Vxxx.
4. Extract the file to your work directory. The SPEEDBUS.GSD is stored in the directory VIPA_System_300S.
5. Start the hardware configurator from Siemens.
6. Close every project.
7. Select *'Options ➔ Install new GSD-file'*.
8. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.

   ⇨ The SPEED7 CPUs and modules of the System 300S from VIPA 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 314-6CF23 happens by means of a virtual PROFIBUS master system with the following approach:



1. Perform a hardware configuration for the CPU. *Chapter 5.4 'Hardware configuration - CPU' on page 57*
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 314-6CF23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 314-6CF23 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 VIPA specific parameters

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

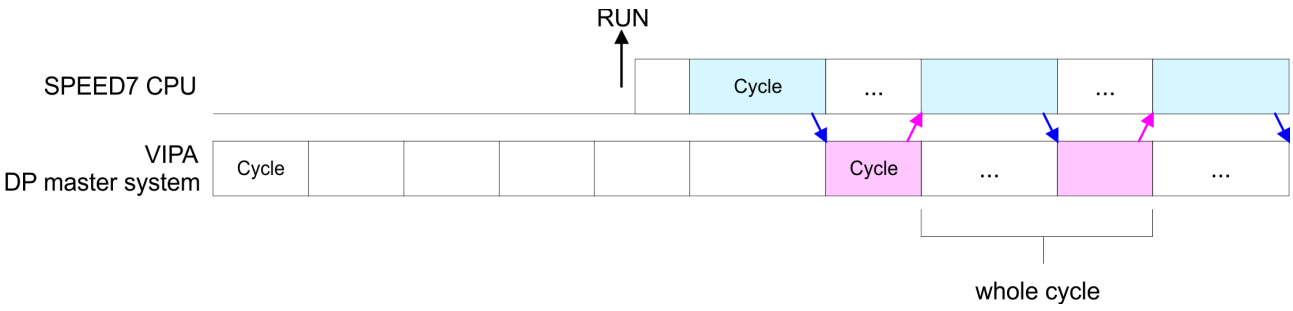
5.9.2.1.1 Synchronization between master system and CPU

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 VIPA 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.

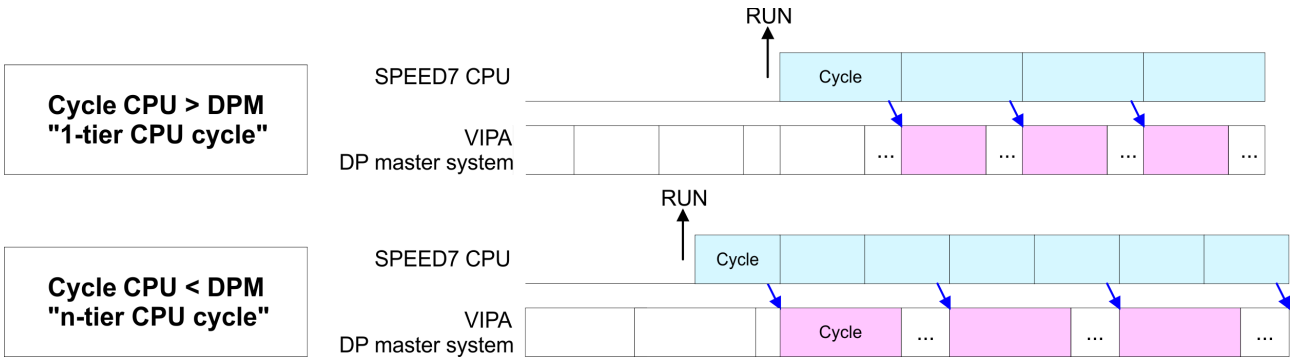
PROFIBUS DP SyncInOut

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.



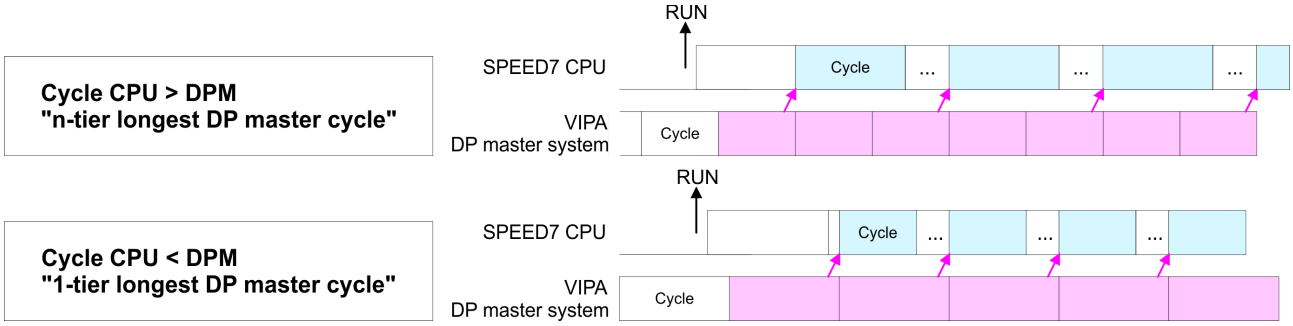
PROFIBUS DP SyncOut

In this operating mode the cycle time of the VIPA 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.



PROFIBUS-DP SyncIn

In the operating mode PROFIBUS DP SyncIn the CPU cycle is synchronized to the cycle of the VIPA PROFIBUS DP master system. Here the CPU cycle depends on the VIPA 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.



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 there is the following interface:

- X2: MPI interface
- X3: PROFIBUS interface

Net structure

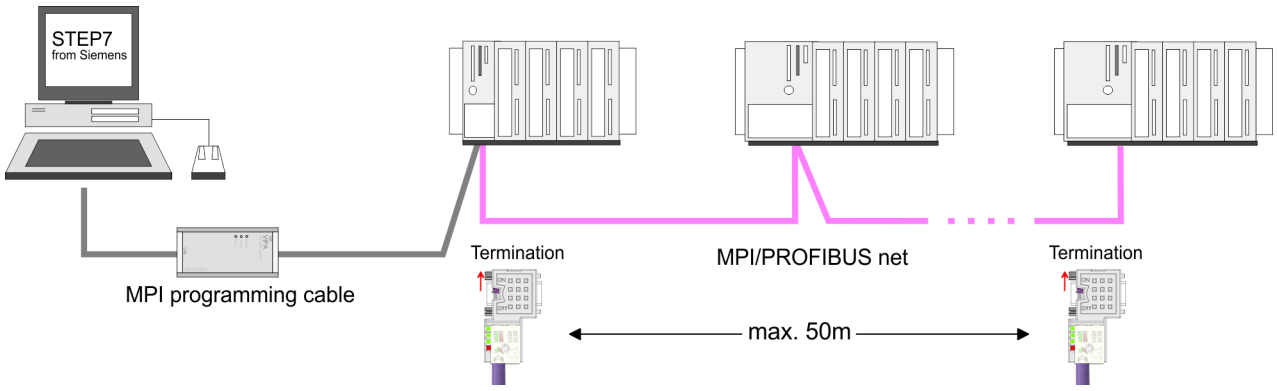
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Please consider with the CPU 314-6CF23 that the total extension of the MPI net does not exceed 50m. Per default the MPI net runs with 187.5kbaud. VIPA CPUs are delivered with MPI address 2.

MPI programming cable

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

Terminating resistor

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



Approach transfer via MPI interface

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Load your project in the SIMATIC Manager from Siemens.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].
5. Set in the register MPI the transfer parameters of your MPI net and type a valid *address.*
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400Baud for the MPI programming cable from VIPA.
8. Via *'PLC ➔ Load to module'* via MPI to the CPU and save it on a memory card via *'PLC ➔ Copy RAM to ROM'* if one is plugged.

Proceeding Transfer via PROFIBUS interface

1. Connect your PC to the PB-DP jack X3 of your CPU via a MPI programming cable.
2. Load your project in the Siemens SIMATIC Manager.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (PROFIBUS)"; if appropriate you have to add it first, then click at [Properties].
5. Set in the register PROFIBUS the transfer parameters of your PROFIBUS net and enter a valid *PROFIBUS address*. The *PROFIBUS address* must be assigned to the DP master by a project before.
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable from VIPA.
8. Transfer your project via *'PLC ➔ Load to module'* via PROFIBUS to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

Transfer via PROFIBUS is available by DP master, if projected as master and assigned with a PROFIBUS address before.

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

Initialization

So that you may access the Ethernet PG/OP channel you have to assign IP address parameters by means of the "initialization".
Chapter 5.6 'Hardware configuration - Ethernet PG/OP channel' on page 59

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

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

- S7PROG.WLD
- AUTOLOAD.WLD

With *'File ➔ Memory Card File ➔ New'* in the Siemens SIMATIC Manager a new wld file may be created. After the creation copy the blocks from the project blocks folder and the *System data* into the wld file.

Transfer memory card → CPU

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

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

A short lightning up of the 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 is controlled by means of the block area of the Siemens SIMATIC Manager *'PLC ➔ Copy RAM to ROM'*. The MC LED lights up during the write access. When the LED expires, the write process is finished.

If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostic Buffer" you reach the diagnostic window. ↯ *Chapter 5.20 'Diagnostic entries' on page 97*

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. *Chapter 5.6 'Hardware configuration - Ethernet PG/OP channel' on page 59*

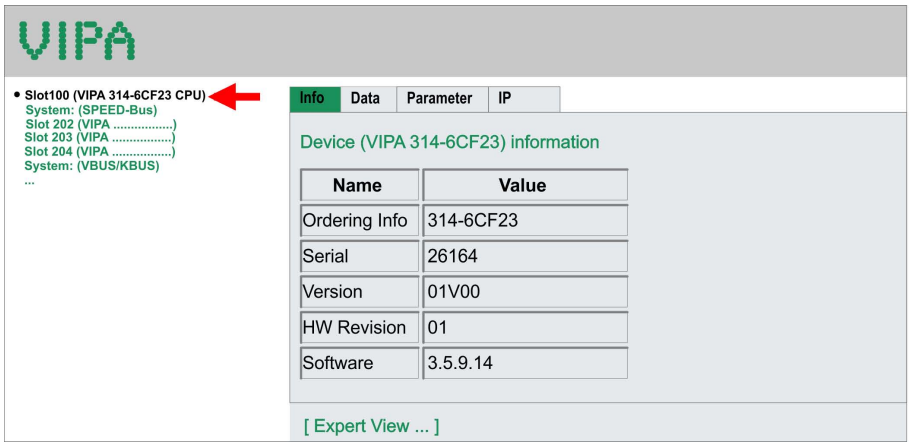
It is assumed that there is a connection between PC and CPU with Internet browser via the Ethernet PG/OP channel. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.

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)	DPM-async	■ 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 / 2097152 Bytes	CPU: Information to memory configuration Load memory, working memory (code/data)
WorkMemCode	0 / 262144 Bytes	
WorkMemData	0 / 262144 Bytes	
PG/OP Network Information		
Device Name	VIPA 314-6CF23 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	314-6CF23 V3.7.5 V.... Px000305.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.

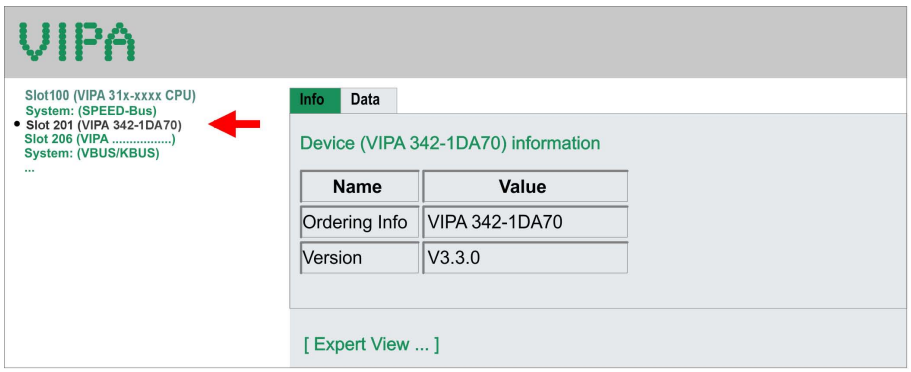
Parameter

Currently nothing is displayed here.

IP

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

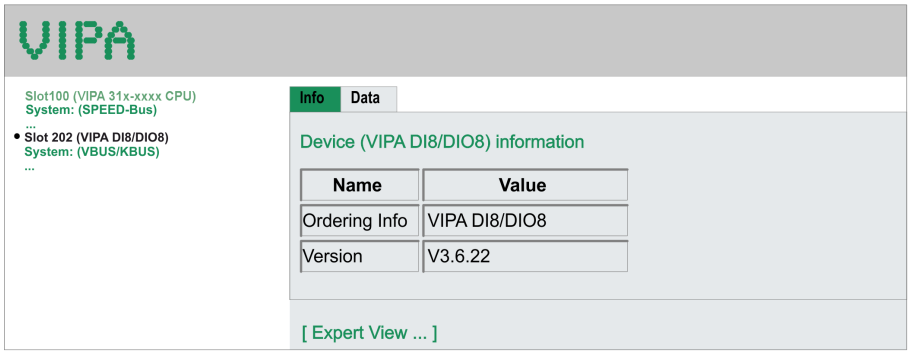
Info - Overview **DP master**



Info - Expert View

Internal Information	**Slot 201**	VIPA 342-1DA70
Module Type	0xCB2C0010	
Module Firmware Information		
PRODUCT	VIPA 342-1DA70 V3.3.5.0 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 **CPU component: Digitale I/O**



Info - Expert View

Internal Information		Slot 202
Module Type	0x4FD30000	Information for support
Module Firmware Information		
PRODUCT	VIPA DI8/DIO8 V3.2.9.0	Name, firmware version

Data - Input data

Offset	Width	Value (dec)	Value (hex)
124	1	0	00
125	1	0	00

Data - Output data

Offset	Width	Value (dec)	Value (hex)	New Value (hex)
124	1	0	00	00
125	1	0	00	00

Info - Overview

CPU component: Analog I/O

Info - Expert View

Internal Information		Slot 203
Module Type	0x55DD0002	Information for support
Module Firmware Information		
BB000432	V1.1.2.0	
PRODUCT	VIPA AI5/AO2 V1.1.2.0 Px000073.pkg	Name, firmware version
Hx000041	V1.6.0.0	

Data - AI5 (10byte)

Offset	Width	Value (dec)	Value (hex)
752	1	255	ff
753	1	240	f0
754	1	0	00
755	1	0	00
756	1	0	00
757	1	0	00
758	1	0	00
759	1	0	00
760	1	127	7f
761	1	255	ff
55dd0002			

Data - AO2 (4byte)

Offset	Width	Value (dec)	Value (hex)	New Value (hex)
752	1	0	00	
753	1	0	00	
754	1	0	00	
755	1	0	00	

Info - Overview

CPU component: counter

Info - Expert View

Internal Information		**Slot 204**
Module Type	0x38C00000	Information for support
Module Firmware Information		
PRODUCT	VIPA 4 COUNTER V3.6.22.0	Name, firmware version

Data - Input data (16byte)

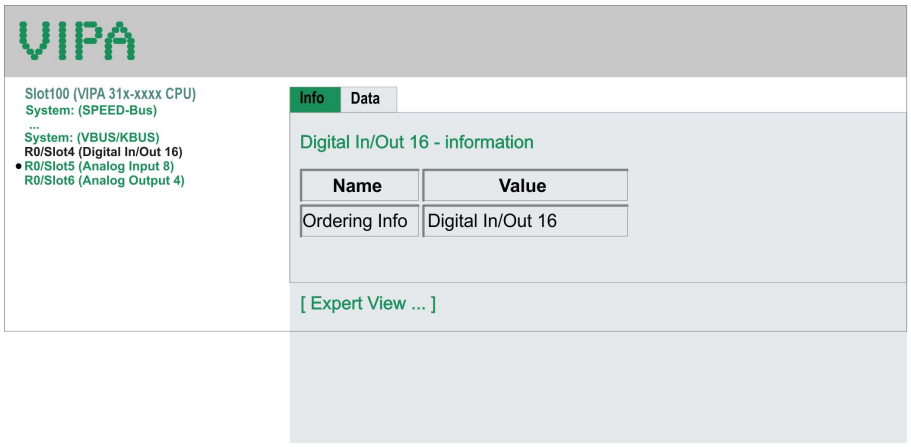
Offset	Width	Value (dec)	Value (hex)
768	1	0	00
769	1	0	00
770	1	0	00
771	1	0	00
772	1	0	00
773	1	0	00
774	1	0	00
775	1	0	00
776	1	0	00
777	1	0	00
778	1	0	00
779	1	0	00
780	1	0	00
781	1	0	00
782	1	0	00
783	1	0	00

Data - Output data (16byte)

Offset	Width	Value (dec)	Value (hex)
768	1	0	00
769	1	0	00
770	1	0	00
771	1	0	00
772	1	0	00
773	1	0	00
774	1	0	00
775	1	0	00
776	1	0	00
777	1	0	00
778	1	0	00
779	1	0	00
780	1	0	00
781	1	0	00
782	1	0	00
783	1	0	00

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 VIPA 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 lock) is set.
	central digital outputs	The outputs are disabled.
	central analog outputs	The outputs are disabled. ■ Voltage outputs issue 0V ■ Current outputs 0...20mA issue 0mA ■ Current outputs 4...20mA issue 4mA If configured also substitute values may be issued.
	decentral outputs	Same 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. You have 2 options to initiate an overall reset:

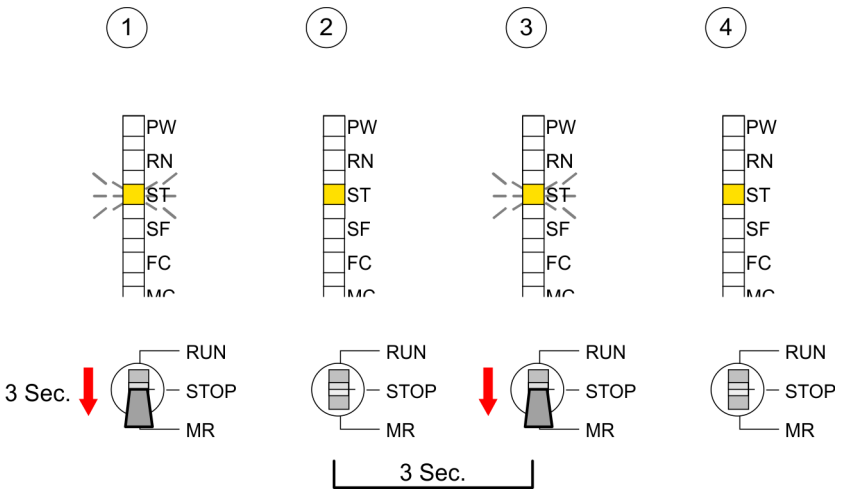
- initiate the overall reset by means of the operating mode switch
- initiate the overall reset by means of the Siemens SIMATIC Manager

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. The operating mode of the CPU is to be switched to STOP. 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 ST-LED blinks (overall reset procedure).
4. The overall reset has been completed when the STOP-LED is on permanently.
   ⇨ The ST-LED is on. The following figure illustrates the above procedure:



Overall reset by means of the Siemens SIMATIC Manager

- Precondition The operating mode of the CPU is to be switched to STOP. You may place the CPU in STOP by the menu command *'PLC ➔ Operating mode'*.
- Overall reset: You may request the overall reset by means of the menu command *'PLC ➔ Clean/Reset'*. In the dialog window you may place your CPU in STOP state and start the overall reset if this has not been done as yet. The ST-LED blinks during the overall reset procedure. 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 memory card, the CPU attempts to reload this project from memory card.

⇨ 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.

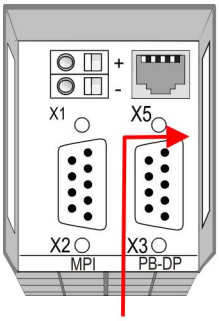
Reset to factory setting

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! *Chapter 5.15 'Reset to factory settings' on page 91*

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.vipa.com

The latest firmware versions are to be found in the service area at www.vipa.com. For example the following files are necessary for the firmware update of the CPU 314-6CF23and its components with hardware release 1:

- 314-6CF23, Hardware release 1: Px000305.pkg
- PROFIBUS-DP master: Px000182.pkg
- DI/DO/AIO: Px000244.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. *Chapter 5.11 'Accessing the web server' on page 76*

Load firmware and transfer it to memory card

- Go to www.vipa.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.

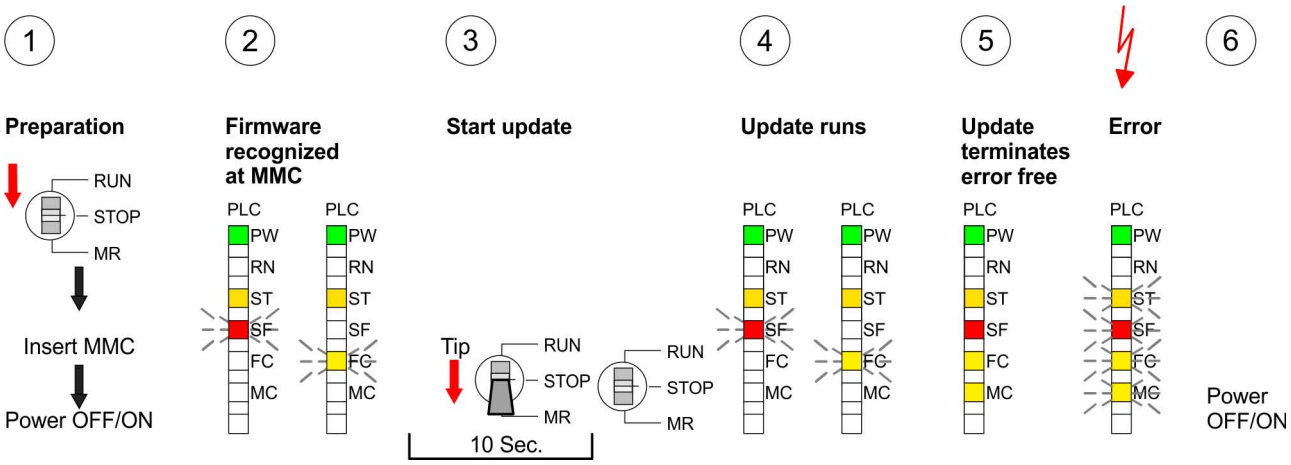
CAUTION!

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 *Chapter 5.15 'Reset to factory settings' on page 91*.

Transfer firmware from memory card into CPU

1. Switch the operating mode switch of your CPU in position STOP. Turn off the voltage supply. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card. Turn on the voltage supply.
2. 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 on the memory card.
3. You start the transfer of the firmware as soon as you tip the operating mode switch downwards to MR within 10s.
4. During the update process, the LEDs SF and FC are alternately blinking and MC LED is on. This may last several minutes.
5. The update is successful finished when the LEDs PW, ST, SF, FC and MC are on. If they are blinking fast, an error occurred.
6. Turn Power OFF and ON. Now it is checked by the CPU, whether further current firmware versions are available at the memory card. If so, again the LEDs SF and FC flash after a short start-up period. Continue with point 3.

   ⇨ If the LEDs do not flash, the firmware update is ready. Now a *factory reset* should be executed. After that the CPU is ready for duty.



5.15 Reset to factory settings

Proceeding

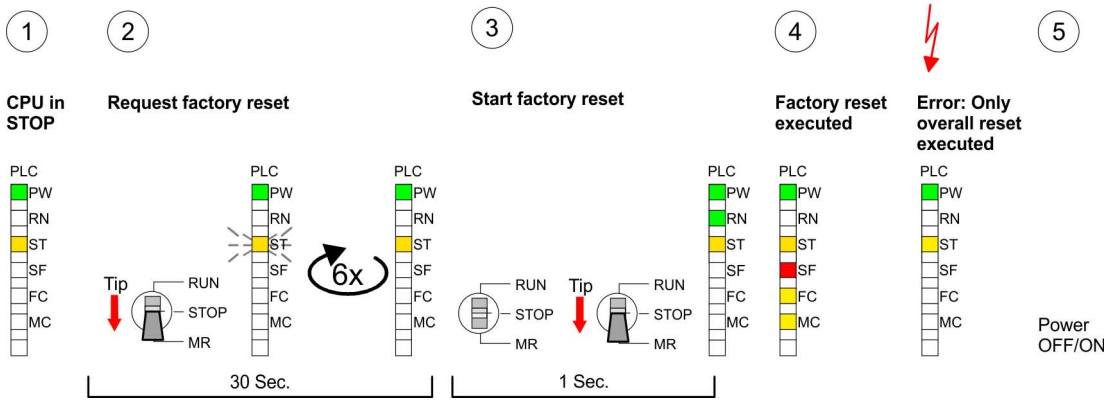
With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.

Please note that here also the IP address of the Ethernet PG/OP channel is set to 0.0.0.0 and the MPI address is reset to the address 2!

A reset to factory setting may also be executed by the CMD auto command *FACTORY_RESET*. *Chapter 5.19 'CMD - auto commands' on page 95*

1. Switch the CPU to STOP.
2. Push the operating mode switch down to position MR for 30s. Here the ST LED flashes. After a few seconds the ST LED changes to static light. Now the ST LED changes between static light and flashing. Starting here count the static light states.
3. After the 6. static light release the operating mode switch and tip it downwards to MR. Now the RN LED lights up once. This means that the RAM was deleted completely.
4. For the confirmation of the resetting procedure the LEDs PW, ST, SF, FC and MC get ON. If not, the factory reset has failed and only an overall reset was executed. In this case you can repeat the procedure. A factory reset can only be executed if the stop LED has static light for exactly 6 times.
5. The end of factory reset is shown by static light of the LEDs PW, ST, SF, FC and MC. Switch the power supply off and on.

The proceeding is shown in the following Illustration:



After the firmware update you always should execute a Reset to factory setting.

5.16 Slot for storage media

Overview

At the front of the CPU there is a slot for storage media. Via this slot as external storage medium for applications and firmware you may use a memory card (MMC respectively SD). You can cause the CPU to load a project automatically respectively to execute a command file by means of pre-defined file names.

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

Accessing the storage medium

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

- After overall reset
  - The CPU checks if there is a project S7PROG.WLD. If exists the project is automatically loaded.
  - The CPU checks if there is a project PROTECT.WLD with protected blocks. If exists the project is automatically loaded. These blocks are stored in the CPU until the CPU is reset to factory setting or an empty PROTECT.WLD is loaded
  - The CPU checks if a MCC memory extension card is put. If exists the memory extension is enabled, otherwise a memory expansion, which was activated before, is de-activated.
- After PowerON
  - The CPU checks if there is a project AUTOLOAD.WLD. If exists an overall reset is established and the project is automatically loaded.
  - The CPU checks if there is a command file with VIPA_CMD.MMC. If exists the command file is loaded and the containing instructions are executed.
  - After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file). If exists this is indicated by blinking of the LEDs and the firmware may be installed by an update request.
- Once in STOP
  - If a storage medium is put, which contains a command file VIPA_CMD.MMC, the command file is loaded and the containing instructions are executed.

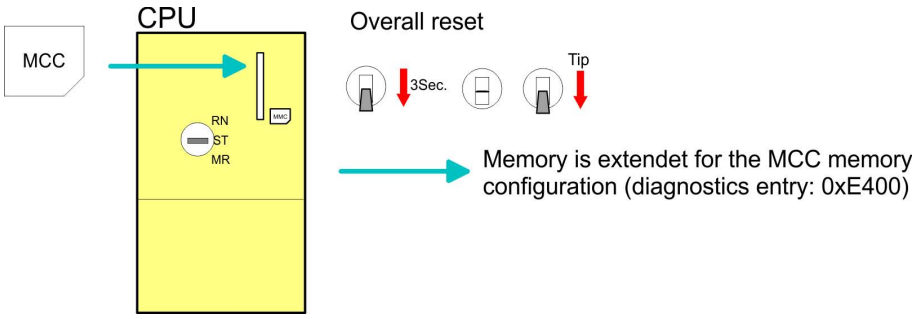
5.17 Memory extension

Overview

In front of the CPU there is a slot for storage media. For this, a MCC memory extension card is available from VIPA. The MCC is a specially prepared MMC (**M**ulti**m**edia **C**ard). 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.

Proceeding

To extend the memory, plug the memory card into the card slot at the CPU labelled with "MCC" and execute an overall reset.



If the memory expansion on the memory card 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 via the integrated web page or with the Siemens SIMATIC Manager at Module Information - "Memory".

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.

Behavior

When the MCC memory configuration has been taken over you may find the diagnostic entry 0xE400 in the diagnostic buffer of the CPU.

After pulling the MCC the entry 0xE401 appears in the diagnostic 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.

The remaining time after pulling the MCC is always been shown with the parameter *MCC-Trial-Time* on the web page.

After re-plugging the MCC, the SF LED extinguishes and 0xE400 is entered into the diagnostic buffer. You may reset the memory configuration of your CPU to the initial status at any time by executing an overall reset without MCC.

5.18 Extended know-how protection

Overview

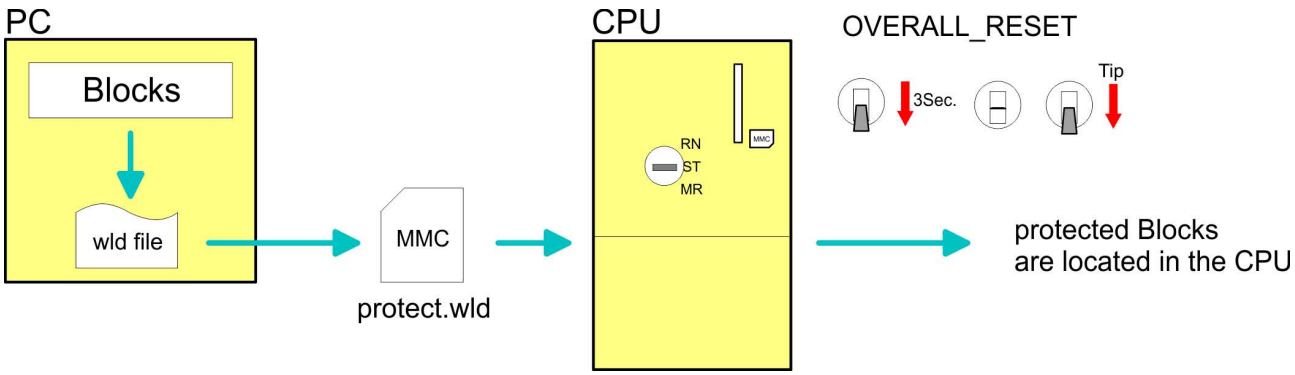
Besides the "standard" Know-how protection the SPEED7-CPUs from VIPA provide an "extended" know-how protection that serves a secure block protection for accesses of 3. persons.

Standard protection

The standard protection from Siemens transfers also protected blocks to the PG but their content is not displayed. But with according manipulation the Know-how protection is not guaranteed.

Extended protection

The "extended" know-how protection developed by VIPA offers the opportunity to store blocks permanently in the CPU. At the "extended" protection you transfer the protected blocks into a WLD-file named protect.wld. By plugging the memory card and following 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.

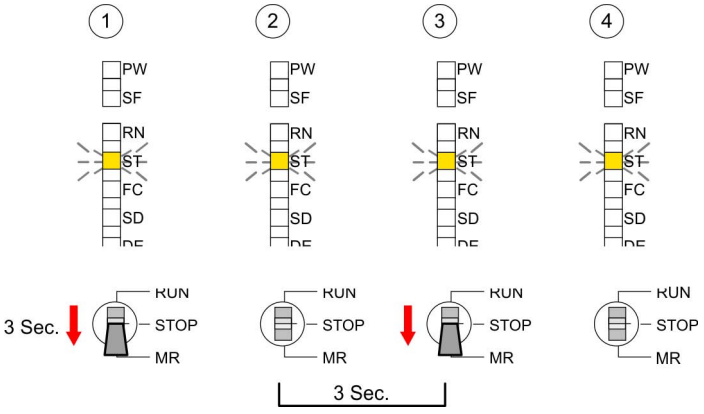


Protect blocks with protect.wld

Create a new wld-file in your project engineering tool with *'File ➔ Memory Card file ➔ New'* and rename it to "protect.wld". Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.

Transfer protect.wld to CPU with overall reset

Transfer the file protect.wld to a memory card, plug the memory card into the CPU and execute an overall reset with the following approach:



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

Protection behavior

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 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 out of all protected blocks. Delete all networks in the blocks so that these only contain the variable definitions in the according symbolism.

5.19 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

Command	Description	Diagnostics entry
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 ☞ *Chapter 5.15 'Reset to factory settings' on page 91* 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 ☞ *Chapter 5.15 'Reset to factory settings' on page 91* 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.

5.20 Diagnostic entries

Accessing diagnostic data

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostics buffer, the VIPA CPUs support some additional specific entries as Event-IDs.
- To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostics Buffer" you reach the diagnostics window.
- The current content of the diagnostic buffer is stored at the memory card by means of the CMD DIAGBUF. ↳ *Chapter 5.19 'CMD - auto commands' on page 95*
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

Overview of the system specific event IDs

Event ID	Description
0x115C	Vendor-specific interrupt (OB 57) at EtherCAT
	OB: OB number
	ZInfo1: Logical address of the slave that triggered the interrupt
	ZInfo2: Interrupt type
	0x00: Reserved
	0x01: Diagnostic interrupt (incoming)
	0x02: Hardware interrupt
	0x03: Pull interrupt
	0x04: Plug interrupt
	0x05: Status interrupt
	0x06: Update interrupt
	0x07: Redundancy interrupt
	0x08: Controlled by the supervisor
	0x09: Enabled
	0x0A: Wrong sub module plugged
	0x0B: Restoration of the sub module
	0x0C: Diagnostic interrupt (outgoing)
	0x0D: Cross traffic connection message
	0x0E: Neighbourhood change message
	0x0F: Synchronisation message (bus)
	0x10: Synchronisation message (device)
	0x11: Network component message
	0x12: Clock synchronisation message (bus)
	0x1F: Pull interrupt module
	ZInfo3: CoE error code
0xE003	Error on accessing the periphery
	ZInfo1 : Transfer type
	ZInfo2 : Periphery address
	ZInfo3 : Slot
0xE004	Multiple configuration of a periphery address
	ZInfo1 : Periphery address
	ZInfo2 : Slot
0xE005	Internal error - Please contact the hotline!
0xE007	Configured in-/output bytes do not fit into periphery area
0xE008	Internal error - Please contact the hotline!
0xE009	Error on accessing the standard backplane bus
0xE010	There is a undefined module at the backplane bus

Event ID	Description
	ZInfo2 : Slot
	ZInfo3 : Type ID
0xE011	Master project engineering at slave CPU not possible or wrong slave configuration
0xE012	Error at parametrization
0xE013	Error at shift register access to standard bus digital modules
0xE014	Error at Check_Sys
0xE015	Error at access to the master
	ZInfo2 : Slot of the master
	ZInfo2 : Page frame master
0xE016	Maximum block size at master transfer exceeded
	ZInfo1 : Periphery address
	ZInfo2 : Slot
0xE017	Error at access to integrated slave
0xE018	Error at mapping of the master periphery
0xE019	Error at standard back plane bus system recognition
0xE01A	Error at recognition of the operating mode (8 / 9 bit)
0xE01B	Error - maximum number of plug-in modules exceeded
0xE020	Error - Interrupt information undefined
	ZInfo2 : Slot
	ZInfo3 : Not relevant to the user
	DatID : Interrupt type
0xE030	Error of the standard bus
0xE033	Internal error - Please contact the hotline!
0xE0B0	SPEED7 is not stoppable (e.g. undefined BCD value at timer)
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xE0C0	Not enough space in work memory for storing code block (block size exceeded)
0xE0CB	Error at SSL access
	ZInfo1 : Error
	4: SSL wrong
	5: Sub-SSL wrong
	6: Index wrong
	ZInfo2 : SSL ID
	ZInfo3 : Index
0xE0CC	Communication errors
	ZInfo1 : Error code

Event ID	Description
	1: Wrong priority
	2: Buffer overflow
	3: Telegram format error
	4: Wrong SSL request (SSL ID not valid)
	5: Wrong SSL request (SSL sub ID invalid)
	6: Wrong SSL request (SSL-Index not valid)
	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 relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xE0CE	Error: Timeout at sending of the 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 relevant to the user
0xE200	Memory card writing finished (Copy Ram2Rom)
	PK : Not relevant to the user
	OB : Not relevant to the user
0xE210	Memory card reading finished (reload after overall reset)
	ZInfo1 : Not relevant to the user
	PK : Not relevant to the user
	OB : Not relevant to the user
0xE21E	Memory card reading: Error at reload (after overall reset), error in block header

Event ID	Description
	ZInfo1 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB
	ZInfo2 : Block number
	ZInfo3 : Block length
0xE21E	Memory card reading: Error at reload (after overall reset), file "Protect.wld" too big
	OB : Not relevant to the user
0xE21F	Memory card reading: Error at reload (after overall reset), checksum error at reading
	PK : Not relevant to the user
	OB : Not relevant to the user
	ZInfo1 : Not relevant to the user
	ZInfo2 : BstTyp
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB

Event ID	Description
	ZInfo3 : BstNr
0xE300	Internal flash writing finished (Copy Ram2Rom)
0xE310	Internal flash writing finished (reload after battery failure)
0xE400	FSC card was plugged
	DatID : FeatureSet Trialtime in minutes
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
0xE401	FSC card was removed
	DatID : FeatureSet Trialtime in minutes
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
	ZInfo3 : Source of the FSC
	0: CPU
	1: Card
0xE402	A configured functionality is not activated
	ZInfo1 : FCS ErrorCode
	1: The PROFIBUS functionality is disabled The interface acts further as MPI interface
	2: The EtherCAT functionality is not enabled
	3: The number of configured axis is not enabled
0xE403	FSC can not be activated in this CPU
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
0xE404	FeatureSet deleted due to CRC error
	DatID : Not relevant to the user
0xE405	The trial time of a feature set or MMC has expired
	DatID : Not relevant to the user
0xE410	A CPU feature set was activated
	DatID : Not relevant to the user
0xE500	Memory management: Deleted block without corresponding entry in BstList

Event ID	Description
	ZInfo2 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB
	ZInfo3 : Block no.
0xE501	Parser error
	ZInfo3 : SDB number
	ZInfo1 : ErrorCode
	1: Parser error: SDB structure
	2: Parser error: SDB is not a valid SDB type.
	ZInfo2 : SDB type
0xE502	Invalid block type in protect.wld
	ZInfo2 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB

Event ID	Description
	ZInfo3 : Block number
0xE503	Inconsistency of code size and block size in work memory
	ZInfo1 : Code size
	ZInfo2 : Block size (high word)
	ZInfo3 : Block size (low word)
0xE504	Additional information for CRC error in work memory
	ZInfo2 : Block address (high word)
	ZInfo3 : Block address (low word)
0xE505	Internal error - Please contact the hotline!
0xE604	Multiple parametrization 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 : Slot of the interface
	ZInfo2 : Number configured connections
	ZInfo3 : Number of allowed connections
0xE610	Onboard PROFIBUS/MPI: Bus error fixed
	ZInfo1 : Interface
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xE701	Internal error - Please contact the hotline!
0xE703	Internal error - Please contact the hotline!
0xE710	Onboard PROFIBUS/MPI: Bus error occurred
	ZInfo1 : Interface
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xE720	Internal error - Please contact the hotline!
0xE721	Internal error - Please contact the hotline!
0xE722	Internal error - Please contact the hotline!
0xE723	Internal error - Please contact the hotline!
0xE780	Internal error - Please contact the hotline!
0xE801	CMD - Auto command: CMD_START recognized and successfully executed
0xE802	CMD - Auto command: CMD_End recognized and successfully executed
0xE803	CMD - Auto command: WAIT1SECOND recognized and successfully executed

Event ID	Description
0xE804	CMD - Auto command: WEBPAGE recognized and successfully executed
0xE805	CMD - Auto command: LOAD_PROJECT recognized and successfully executed
0xE806	CMD - Auto command: SAVE_PROJECT recognized and successfully executed
	ZInfo3 : Status
	0: Error
	1: OK
	0x8000: Wrong password
0xE807	CMD - Auto command: FACTORY_RESET recognized and successfully executed
0xE808	Internal error - Please contact the hotline!
0xE809	Internal error - Please contact the hotline!
0xE80A	Internal error - Please contact the hotline!
0xE80B	CMD - Auto command: DIAGBUF recognized and successfully executed
	ZInfo3 : Status
	0: OK
	0xFE81: File create error
	0xFEA1: File write error
	0xFEA2: Odd address when reading
0xE80C	Internal error - Please contact the hotline!
0xE80D	Internal error - Please contact the hotline!
0xE80E	CMD - Auto command: SET_NETWORK recognized and successfully executed
0xE80F	Internal error - Please contact the hotline!
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 recognized
0xE816	CMD - Auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal error - Please contact the hotline!
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

Event ID	Description
0xE82A	CMD - Auto command: CPUTYPE_318 recognized and successfully executed
	ZInfo3 : Error code
	0: No Error
	1: Command not possible
	2: Error on storing the attribute
0xE82B	CMD - Auto command: CPUTYPE_ORIGINAL recognized and successfully executed
	ZInfo3 : Error code
	0: No Error
	1: Command not possible
	2: Error on storing the attribute
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 missing
0xE8FF	CMD - Auto command: Error: Error while reading CMD file (memory card error)
0xE901	Check sum error
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	DatID : Not relevant to the user
0xE902	Internal error - Please contact the hotline!
0xEA00	Internal error - Please contact the hotline!
0xEA01	Internal error - Please contact the hotline!
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	ZInfo1 : Slot
	ZInfo2 : Type ID set
	ZInfo3 : Type ID
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA03	SBUS: Communication error between CPU and IO controller
	ZInfo1 : Slot
	ZInfo2 : Status
	0: OK
	1: Error
	2: Empty
	3: Busy
	4: Timeout
	5: Internal blocking
	6: Too many frames
	7: Not connected

Event ID	Description
	8: Unknown
	PK : Not relevant to the user
	DatID : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
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: Parametrized input data width unequal to plugged input data width
	ZInfo1 : Parametrized input data width
	ZInfo2 : Slot
	ZInfo3 : Input data width of the plugged module
0xEA09	SBUS: Parametrized output data width unequal to plugged output data width
	ZInfo1 : Parametrized output data width
	ZInfo2 : Slot
	ZInfo3 : Output data width of the plugged module
0xEA10	SBUS: Input periphery address outside the periphery area

Event ID	Description
	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 at writing record set
	ZInfo1 : Slot
	ZInfo2 : Record set number
	ZInfo3 : Record set length
0xEA14	SBUS: Multiple parametrization of a periphery address (diagnostics address)
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	ZInfo3 : Data width
0xEA15	Internal error - Please contact the hotline!
0xEA18	SBUS: Error at mapping of the master periphery
	ZInfo2 : Slot of the master
0xEA19	Internal error - Please contact the hotline!
0xEA1A	SBUS: Error at access to the FPGA address table
	ZInfo2 : HW slot
	ZInfo3 : Table
	0: Reading
	1: Writing
	PK : Not relevant to the user
	DatID : Not relevant to the user
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 faulty configured.
0xEA22	Error - RS485 interface X2 - Value exceeds the limits
	ZInfo2 : Project engineering for X2
0xEA23	Error - RS485 interface X3 - Value exceeds the limits
	ZInfo2 : Project engineering for X3
0xEA24	Error - Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used.
	ZInfo2 : Project engineering for X2
	ZInfo3 : Project engineering for X3
0xEA30	Internal error - Please contact the hotline!

Event ID	Description
0xEA40	Internal error - Please contact the hotline!
0xEA41	Internal error - Please contact the hotline!
0xEA50	PROFINET IO controller: Error in the configuration
	ZInfo1 : Rack/slot of the controller
	ZInfo2 : Device no.
	ZInfo3 : Slot at the device
	OB : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA51	PROFINET IO CONTROLLER: There is no PROFINET IO controller at the configured slot
	ZInfo1 : Rack/slot of the controller
	ZInfo2 : Recognized ID at the configured slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA53	PROFINET IO CONTROLLER: PROFINET configuration: There are too many PROFINET IO devices configured
	ZInfo1 : Number of configured devices
	ZInfo2 : Slot
	ZInfo3 : Maximum possible number of devices
0xEA54	PROFINET IO controller: IO controller reports multiple parametrization of a periphery address
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	ZInfo3 : Data width
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA61	Internal error - Please contact the hotline!
0xEA62	Internal error - Please contact the hotline!
0xEA63	Internal error - Please contact the hotline!
0xEA64	PROFINET IO controller/EtherCAT-CP: Error in the configuration
	ZInfo1 : Too many devices
	ZInfo1 : Too many devices per second
	ZInfo1 : Too many input bytes per ms
	ZInfo1 : Too many output bytes per ms
	ZInfo1 : Too many input bytes per ms
	ZInfo1 : Too many output bytes per device
	ZInfo1 : Too many productive connections
	ZInfo1 : Too many input bytes in the process image
	ZInfo1 : Too many output bytes in the process image
	ZInfo1 : Configuration not available

Event ID	Description
	ZInfo1 : Configuration not valid
	ZInfo1 : Refresh time too short
	ZInfo1 : Cycle time too big
	ZInfo1 : Not valid device number
	ZInfo1 : CPU is configured as I device
	ZInfo1 : Use different method to obtain IP address Is not supported for the IP address of the controller
	ZInfo2 : Incompatible configuration (SDB version not supported)
	ZInfo2 : EtherCAT: EoE configured but not supported
	ZInfo2 : DC parameter not valid
0xEA65	Internal error - Please contact the hotline!
0xEA66	PROFINET error in communication stack
	PK : Rack/slot
	OB : StackError.Service
	DatID : StackError.DeviceRef
	ZInfo1 : StackError.Error.Code
	ZInfo2 : StackError.Error.Detail
	ZInfo3 : StackError.Error.AdditionalDetail
	ZInfo3 : StackError.Error.AreaCode
0xEA67	PROFINET IO controller: Error reading record set
	PK : Error type
	0: Record set error local
	1: Record set error stack
	2: Record set error station
	OB : Rack/slot of the controller
	DatID : Device
	ZInfo1 : Record set number
	ZInfo2 : Record set handle (caller)
	ZInfo3 : Internal error code from PN stack
0xEA68	PROFINET IO controller: Error at writing record set
	PK : Error type
	0: Record set error local
	1: Record set error stack
	2: Record set error station
	OB : Rack/slot of the controller
	DatID : Device
	ZInfo1 : Record set number
	ZInfo2 : Record set handle (caller)
	ZInfo3 : Internal error code from PN stack

Event ID	Description
0xEA69	Internal error - Please contact the hotline!
0xEA6A	PROFINET IO controller: Service error in communication stack
	PK : Rack/slot
	OB : Service ID
	ZInfo1 : ServiceError.Code
	ZInfo2 : ServiceError.Detail
	ZInfo3 : StackError.Error.AdditionalDetail
	ZInfo3 : ServiceError.AreaCode
0xEA6B	PROFINET IO controller: Faulty vendor ID
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6C	PROFINET IO controller: Faulty device ID
	ZInfo1 : Device ID
	PK : Rack/slot
	OB : Operation mode

Event ID	Description
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xEA6D	PROFINET IO controller: No empty Name
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE

Event ID	Description
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6E	PROFINET IO controller: Waiting for RPC answer
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6F	PROFINET IO controller: PROFINET module deviation
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode

Event ID	Description
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA70	PROFINET stack error in configuration
	ZInfo1 : UnsupportedApiError.slot
	ZInfo2 : UnsupportedApiError.subslot
	OB : UnsupportedApiError.api
	PK : Rack Slot No
	DatID : UnsupportedApiError.deviceID
0xEA71	Internal PROFINET error - Please contact the hotline!
0xEA81	Internal error - Please contact the hotline!
0xEA82	Internal error - Please contact the hotline!
0xEA83	Internal error - Please contact the hotline!
0xEA91	Internal error - Please contact the hotline!
0xEA92	Internal error - Please contact the hotline!
0xEA93	Internal error - Please contact the hotline!
0xEA97	Internal error - Please contact the hotline!
0xEA98	Timeout at waiting for reboot of a SBUS module (server)
	PK : Not relevant to the user
	DatID : Not relevant to the user

Event ID	Description
	ZInfo3 : Slot
0xEA99	Error at file reading via SBUS
	ZInfo3 : Slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : File version of the SBUS module (if not equal to 0)
	ZInfo1 : File version at MMC/SD (if not equal 0)
0xEAA0	Internal error - Please contact the hotline!
0xEAB0	Link mode not valid
	ZInfo1 : Diagnostics address of the master
	ZInfo2 : Current connection mode
	0x01: 10Mbit half-duplex
	0x02: 10Mbit full-duplex
	0x03: 100Mbit half-duplex
	0x04: 100Mbit full-duplex
	0x05: Link mode undefined
	0x06: Auto Negotiation
	OB : Current operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xEAC0	Internal error - Please contact the hotline!

Event ID	Description
0xEAD0	Error in configuration SyncUnit
0xEB02	SLIO bus: Present configuration does not match the 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	SLIO error: IO mapping
	ZInfo1 : Type of error
	0x01: SDB parser error
	0x02: Configured address already used
	0x03: Mapping error
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Slot (0=not be determined)
0xEB05	SLIO error: Bus structure for Isochron process image not suitable
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Slot (0=not be determined)
0xEB10	SLIO error: Bus error
	ZInfo1 : Type of error
	0x60: Bus enumeration error
	0x80: General error
	0x81: Queue execution error
	0x82: Error interrupt
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEB11	SLIO error during bus initialization
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEB20	SLIO error: Interrupt information undefined
0xEB21	SLIO error: Accessing configuration data
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xEC03	EtherCAT: Error in configuration
	ZInfo1 : Error code
	1: Number of slaves is not supported.
	2: Master system ID not valid

Event ID	Description
	3: Slot not valid
	4: Master configuration not valid
	5: Master type not valid
	6: Slave diagnostic address invalid
	7: Slave address not valid
	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)
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Error code higher 2 bytes
	ZInfo3 : Error code lower 2 bytes
0xEC04	EtherCAT Multiple configuration of a periphery address
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEC05	EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive
	PK : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting

Event ID	Description
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	DatID : Not relevant to the user
	ZInfo1 : Station address of the EtherCAT device
	ZInfo2 : Errorcode
	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 Beckhoff 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 timeout, 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 relevant to the user
0xEC10	EtherCAT: Restoration bus with its slaves
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address

Event ID	Description
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC11	EtherCAT: Restoration bus with missing slaves
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC12	EtherCAT: Restoration slave
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp

Event ID	Description
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AL Statuscode
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC30	EtherCAT: Topology OK
	ZInfo2 : Diagnostics address of the master
0xEC50	EtherCAT: DC out of sync
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : DC State Change
	0: DC master out of sync
	1: DC slaves out of Sync
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xED10	EtherCAT: Bus failure
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT

Event ID	Description
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostic address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED12	EtherCAT: Slave failure
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AlStatusCode
	0x0000: No Error
	0x0001: Unspecified error
	0x0011: Invalid requested status change
	0x0012: Unknown requested status

Event ID	Description
	0x0013: Bootstrap not supported
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration
	0x0020: Slave needs cold start
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED20	EtherCAT: Bus state change without calling OB86
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp

Event ID	Description
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED21	EtherCAT: Faulty bus status change
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Error code
	0x0008: Busy
	0x000B: Invalid parameters
	0x000E: Invalid status
	0x0010: Timeout
	0x0004: Abbort (master state change)

Event ID	Description
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED22	EtherCAT: Slave state change without calling OB86
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AlStatusCode
	0x0000: No Error
	0x0001: Unspecified error
	0x0011: Invalid requested status change
	0x0012: Unknown requested status
	0x0013: Bootstrap not supported
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration

Event ID	Description
	0x0020: Slave needs cold start
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED23	EtherCAT: Timeout while changing the master status to OP, after CPU has changed to RUN
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power

Event ID	Description
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : Master status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : There is an EtherCAT configuration
	0: There is no EC configuration
	1: There is an EC configuration
	ZInfo3 : DC in sync
	0: not in sync
	1: in sync
0xED30	EtherCAT: Topology deviation
	ZInfo2 : Diagnostics address of the master
0xED31	EtherCAT: Overflow of the interrupt queue
	ZInfo2 : Diagnostics address of the master
0xED50	EtherCAT: DC slaves in sync
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : DC State change
	0: Master
	1: Slave
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING

Event ID	Description
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xED60	EtherCAT: Diagnostics buffer CP: Slave state change
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Slave address
	ZInfo3 : AlStatusCode
	0x0000: No Error

Event ID	Description
	0x0001: Unspecified error
	0x0011: Invalid requested status change
	0x0012: Unknown requested status
	0x0013: Bootstrap not supported
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration
	0x0020: Slave needs cold start
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Restoration slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status

Event ID	Description
0xED61	EtherCAT: Diagnostics buffer CP: CoE emergency
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	DatID : Error code
	ZInfo1 : Error register
	ZInfo1 : MEF-Byte1
	ZInfo2 : MEF-Byte2
	ZInfo2 : MEF-Byte3
	ZInfo3 : MEF-Byte4
	ZInfo3 : MEF-Byte5
0xED62	EtherCAT: Diagnostics buffer CP: Error on SDO access
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	DatID : Subindex
	ZInfo1 : Index
	ZInfo2 : SDOErrorCode (high word)
	ZInfo3 : SDOErrorCode (low word)
0xED63	EtherCAT: Diagnostics buffer CP: Error in the response to an INIT command
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	ZInfo1 : Error type
	1: No response
	2: Validation error
	3: INIT command failed, requested station could not be reached
	0: Not defined
0xED70	EtherCAT: Diagnostics buffer CP: Twice HotConnect group found
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT

Event ID	Description
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : Diagnostics address of the master
	ZInfo2 : EtherCAT station address
0xEE00	Additional information at UNDEF_OPCODE
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Not relevant to the user
	DatID : Not relevant to the user
0xEE01	Internal error - Please contact the hotline!
0xEEEE	CPU was completely overall reset, since after PowerON the start-up could not be finished
0xEF00	Internal error - Please contact the hotline!
0xEF01	Internal error - Please contact the hotline!
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!
0xEFFF	Internal error - Please contact the hotline!

5.21 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.

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

The processing of the states may be interrupted by means of jump commands or by timer and process-related interrupts. 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. 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. For this reason, jumps or time and process alarms can result in the value displayed during program execution remaining at 0 for the items below:

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

'PLC ➔ Monitor/Modify Variables'

This test function returns the condition of a selected operand (inputs, outputs, flags, data word, counters or timers) at the end of program execution. This information is obtained from the process image of the selected operands. During the "processing check" or in operating mode STOP the periphery is read directly from the inputs. Otherwise only the process image of the selected operands is displayed.

- Control of outputs
  - It is possible to check the wiring and proper operation of output modules.
  - You can set outputs to any desired status with or without a control program. The process image is not modified but outputs are no longer inhibited.
- Control of 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.
  - Process variables are controlled asynchronously to the execution sequence of the program.

6 Deployment I/O periphery

6.1 Overview

General

At the CPU 314-6CF23 the analog and digital in-/output channels are together in a 2tier casing. The following components are integrated:

- Analog input
  - 4xU/Ix12bit
  - 1xPt100
- Analog output
  - 2xU/Ix12bit
- Digital input
  - 16(8)xDC24V with parametrizable counter functions
- Digital output
  - 0(8)xDC24V 1A
- Counter
  - max. 4 counter with the operating mode endless, single or periodic count

Project engineering

If there is no hardware configuration available, the in- and output areas starting with address 1024 are mapped to the address range of the CPU. In the following these areas are more described. Otherwise the project engineering takes place after installing the SPEEDBUS.GSD in the Siemens SIMATIC Manager.

Counter

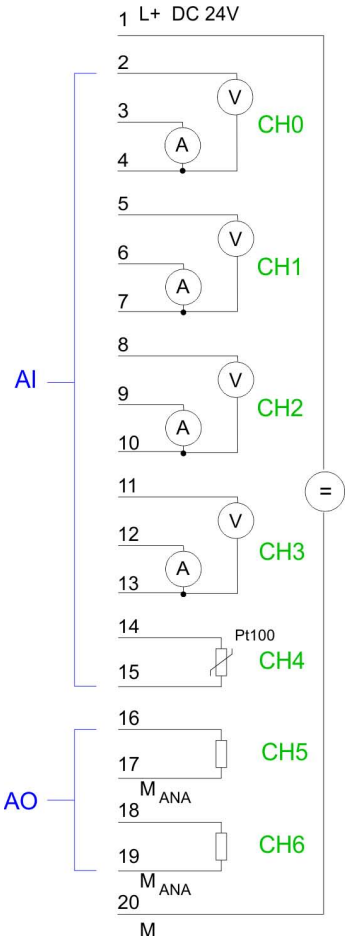
The counters used here are endless counter, where the control happens via the digital input channels. For the counter you may configure interrupts via hardware configuration that may influence the corresponding digital output channel.

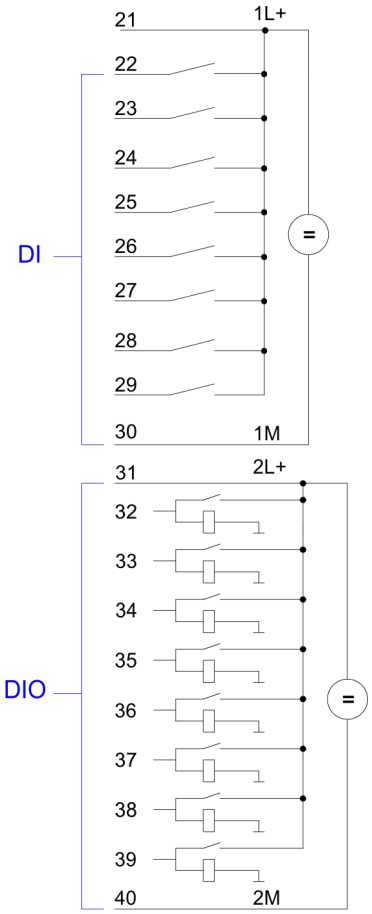
6.2 In-/Output range CPU 314-6CF23

Overview CPU 314-6CF23

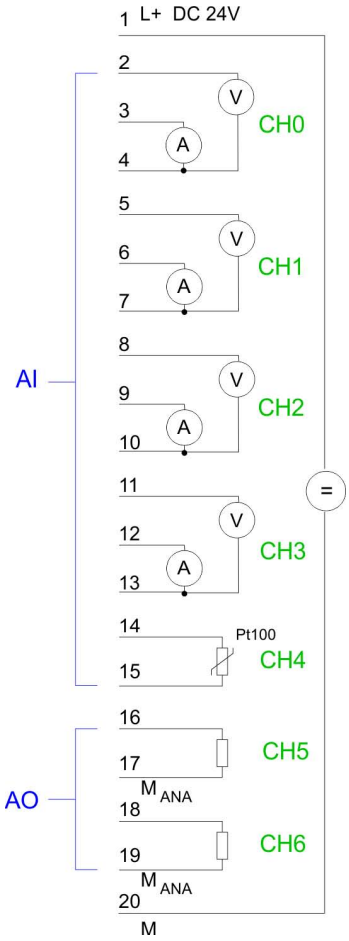
The CPU 314-6CF23 has the following analog and digital in- and output ranges integrated in one casing:

- Analog input: 4x12Bit, 1xPt100
- Analog output: 2x12Bit
- Digital input: 8xDC 24V, interrupt capable, 4 counter
- Digital in-/output: 8xDC 24V, 0.5A

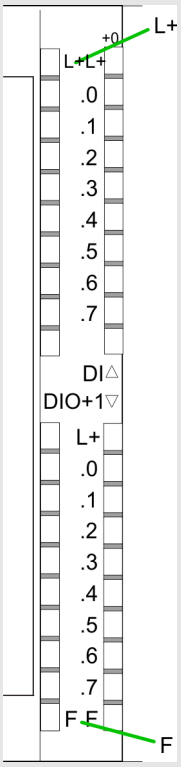


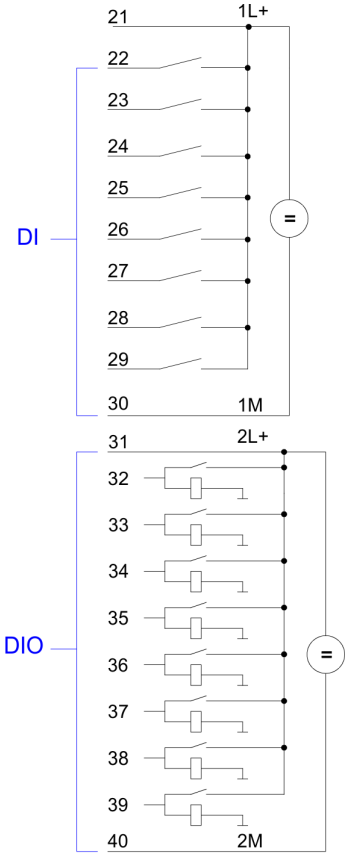


CAUTION!
Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+. Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal. A thus connected output remains active even with shut down supply voltage. Non-observance may cause damages of the module.



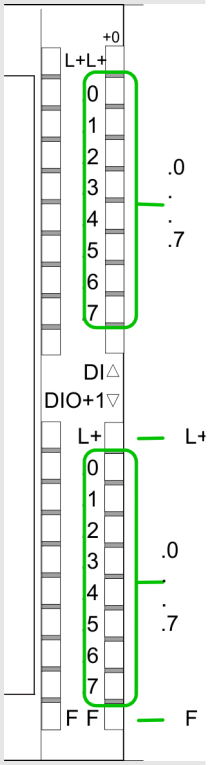
CPU 314-6CF23: Analog part pin assignment and status indicator

Pin	Assignment	LEDs	Description
1	Power supply DC 24V AIO		**1L+** LED (green) Supply voltage available **F** LED (red) Sum error
2	Voltage measurement channel 0		
3	Current measurement channel 0		
4	Ground channel 0		
5	Voltage measurement channel 1		
6	Current measurement channel 1		
7	Ground channel 1		
8	Voltage measurement channel 2		
9	Current measurement channel 2		
10	Ground channel 2		
11	Voltage measurement channel 3		
12	Current measurement channel 3		
13	Ground channel 3		
14	Pt 100 channel 4		
15	Pt 100 channel 4		
16	Output + channel 5		
17	Ground output channel 5		
18	Output + channel 6		
19	Ground output channel 6		
20	Ground power supply AIO		



CPU 314-6CF23: Digital part pin assignment and status indicator

Pin	Assignment
1	Power supply +DC 24 V DI
2	I+0.0 / Counter 0(A)
3	I+0.1 / Counter 0(B)
4	I+0.2 / Gate0/Latch0/Reset0
5	I+0.3 / Counter 1(A)
6	I+0.4 / Counter 1(B)
7	I+0.5 / Gate1/Latch1/Reset1
8	I+0.6 / Counter 2(A)
9	I+0.7 / Counter 2(B)
10	Ground DI
11	Power supply +DC 24 V DIO
12	I/Q+1.0 / Gate2/Latch2/Reset2
13	I/Q+1.1 / Counter 3(A)
14	I/Q+1.2 / Counter 3(B)
15	I/Q+1.3 / Gate3/Latch3/Reset3
16	I/Q+1.4 / OUT0/Latch0/Reset0
17	I/Q+1.5 / OUT1/Latch1/Reset1
18	I/Q+1.6 / OUT2/Latch2/Reset2
19	I/Q+1.7 / OUT3/Latch3/Reset3
20	Ground DIO



DI:

.0 7

LED (green)

I+0.0 to I+0.7

starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated

DIO:

2L+

LED (green)

Supply voltage available for DIO

.0 7

LED (green)

I/Q+1.0 to I/Q +1.7

on at active output/input

F

LED (red)

Overload or short circuit error

6.3 Address assignment I/O part

Overview

- By including the SPEEDBUS.GSD in your hardware configurator, the module is at your disposal in the hardware catalog. After the installation of the GSD you will find the CPU 314-6CF23 at *'Additional field devices ➔ I/O ➔ VIPA_SpeedBus'*.
- In case there is no hardware configuration available, the in- and output areas starting at address 1024 are shown in the address range of the CPU.
- For the data input a range of 48byte and for the data output a range of 24byte is available

Input area

Addr.	Name	Byte	Function
+0	DI_0	1	Digital input I+0.0 ... I+0.7
+1	DI_1	1	Digital input I+1.0 ... I+1.7
+2	-	2	reserved
+4	AI_CH0	2	Analog input CH0

Addr.	Name	Byte	Function
+6	AI_CH1	2	Analog input CH1
+8	AI_CH2	2	Analog input CH2
+10	AI_CH3	2	Analog input CH3
+12	AI_CH4	2	Analog input CH4
+14	-	2	reserved
+16	CVCL_0	4	Counter/Latch value 0
+20	-	2	reserved
+22	ISTS_0	2	Input status counter 0
+24	CVCL_1	4	Counter/Latch value 1
+28	-	2	reserved
+30	ISTS_1	2	Input status counter 1
+32	CVCL_2	4	Counter/Latch value 2
+36	-	2	reserved
+38	ISTS_2	2	Input status counter 2
+40	CVCL_3	4	Counter/Latch value 3
+44	-	2	reserved
+46	ISTS_3	2	Input status counter 3

Output area

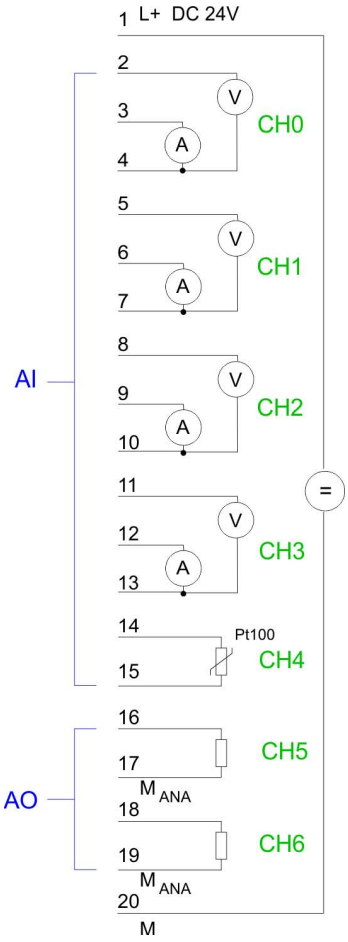
Addr.	Name	Byte	Function
+0	-	1	reserved
+1	DO_1	1	Digital output Q+1.0 ... Q+1.7
+2	-	2	reserved
+4	AO_CH0	2	Analog output CH0
+6	AO_CH1	2	Analog output CH1
+8	-	2	reserved
+10	OSTS_0	2	Output status counter 0
+12	-	2	reserved
+14	OSTS_1	2	Output status counter 1
+16	-	2	reserved
+18	OSTS_2	2	Output status counter 2
+20	-	2	reserved
+22	OSTS_3	2	Output status counter 3

6.4 Analog part

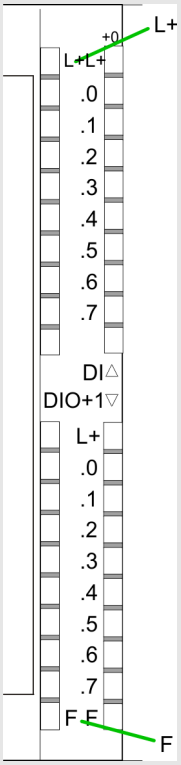
Overview

The analog part consists of 4 input, 1 Pt100 and 2 output channels. 10byte input and 4byte output data of the process image are used by the analog part. The channels of the module are galvanically separated from the SPEEDBus via DC/DC transducer and opto couplers.

> **CAUTION!**
> Temporarily not used analog inputs with activated channel must be connected to the concerning ground. To avoid measuring errors, you should connect only one measuring type per channel.



CPU 314-6CF23: Analog part pin assignment and status indicator

Pin	Assignment	LEDs	Description
1	Power supply DC 24V AIO		**1L+** LED (green) Supply voltage available **F** LED (red) Sum error
2	Voltage measurement channel 0		
3	Current measurement channel 0		
4	Ground channel 0		
5	Voltage measurement channel 1		
6	Current measurement channel 1		
7	Ground channel 1		
8	Voltage measurement channel 2		
9	Current measurement channel 2		
10	Ground channel 2		
11	Voltage measurement channel 3		
12	Current measurement channel 3		
13	Ground channel 3		
14	Pt 100 channel 4		
15	Pt 100 channel 4		
16	Output + channel 5		
17	Ground output channel 5		
18	Output + channel 6		
19	Ground output channel 6		
20	Ground power supply AIO		

Access to the analog part

- By including the SPEEDBUS.GSD in your hardware configurator, the module is at your disposal in the hardware catalog. After the installation of the GSD you will find the CPU 314-6CF23 at *'Additional field devices ➔ I/O ➔ VIPA_SpeedBus'*.
- If there is no hardware configuration available, the in- and output areas starting with address 1024 are mapped to the address range of the CPU.

- For every channel the measuring data are stored as word in the data input range.
- For the output you have to enter a value as word into the data output range.

Used area *'Input area' on page 135*

Address	Access	Assignment
+4	Word	Analog input CH0
+6	Word	Analog input CH1
+8	Word	Analog input CH2
+10	Word	Analog input CH3
+12	Word	Analog input CH4

Used area *'Output area' on page 136*

Address	Access	Assignment
+4	Word	Analog output CH0
+6	Word	Analog output CH1

Numeric notation in Siemens S7 format

The analog values are represented in two's complement format. Depending on the parametrized transformation speed the lowest value bits of the measuring value are irrelevant. With increasing sampling rate, the resolution decreases. The following table lists the resolution in dependence of the sampling rate. The lowest value irrelevant bits of the output value are marked with "X".

Resolution	Analog value															
	High byte (byte 0)								Low byte (byte 1)							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Significance	sign	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
15bit + sign	sign	Relevant output value (at 3.7 ... 30Hz)														
14bit + sign	sign	Relevant output value (at 60Hz)														X
13bit + sign	sign	Relevant output value (at 120Hz)													X	X
11bit + sign	sign	Relevant output value (at 170Hz)											X	X	X	X
9bit + sign	sign	Relevant output value (at 200Hz)									X	X	X	X	X	X

Algebraic sign bit (sign)

For the sign bit is valid:

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

Behavior at errors

As soon as a measuring value exceeds overrange respectively unterrange, the following value is returned:

- Measured value > overrange
  - 32767 (7FFFh)
- Measured value < unterrange
  - -32768 (8000h)

At wire break, parametrization error or deactivated analog part the measuring value 32767 (7FFFh) is returned.

De-activate analog part

With this record set 9Eh you may de-activate the digital respectively analog part. Please regard that in spite of the de-activation of the digital res. analog part, the process image for both components remains reserved. The record set has the following structure:

Byte	Bit 7 ... 0
0...1	■ Bit 15 ... 0: Module selection – 0000h: Activate digital and analog part (default) – 0001h: De-activate digital part – 0002h: De-activate analog part

Digital/Analog conversion

In the following all measuring ranges are listed that are supported by the analog part. The here listed formulas allow you to transform an evaluated measuring value (digital value) to a value assigned to the measuring range and vice versa.

Measuring range	Voltage (U)	Decimal (D)	Hex	Range	Conversion
±10V	11.76V	32511	7EFFh	overrange	$D = 27648 \cdot \frac{U}{10}$ $U = D \cdot \frac{10}{27648}$
	10V	27648	6C00h	nominal range	
	5V	13824	3600h		
	0V	0	0000h		
	-5V	-13824	CA00h		
	-10V	-27648	9400h		
	-11.76	-32512	8100h	underrange	

Measuring range	Voltage (U)	Decimal (D)	Hex	Range	Conversion
0 ... 10V	11.76V	32511	7EFFh	overrange	$D = 27648 \cdot \frac{U}{10}$ $U = D \cdot \frac{10}{27648}$
	10V	27648	6C00h	nominal range	
	5V	13824	3600h		
	0V	0	0000h		
	-1.76V	-4864	ED00h	underrange	

Measuring range	Current (I)	Decimal (D)	Hex	Range	Conversion
0 ... 20mA	23.52mA	32511	7EFFh	overrange	$D = 27648 \cdot \frac{I}{20}$ $I = D \cdot \frac{20}{27648}$
	20mA	27648	6C00h	nominal range	
	10mA	13824	3600h		
	0mA	0	0000h		
	-3.52	-4864	ED00h	underrange	

Measuring range	Current (I)	Decimal (D)	Hex	Range	Conversion
4 ... 20mA	22.81mA	32511	7EFFh	overrange	$D = 27648 \cdot \frac{I-4}{16}$ $I = D \cdot \frac{16}{27648} + 4$
	20mA	27648	6C00h	nominal range	
	12mA	13824	3600h		
	4mA	0	0000h		
	1.19mA	-4864	ED00h	underrange	

Measuring range	Current (I)	Decimal (D)	Hex	Range	Conversion
±20mA	23.52mA	32511	7EFFh	overrange	$D = 27648 \cdot \frac{I}{20}$ $I = D \cdot \frac{20}{27648}$
	20mA	27648	6C00h	nominal range	
	10mA	13824	3600h		
	0mA	0	0000h		
	-10mA	-13824	CA00h		
	-20mA	-27648	9400h		
	-23.52mA	-32512	8100h	underrange	

6.5 Analog part - Parametrization

Parameter data

18Byte of parameter data are available for the configuration. By using the record set B4h of the SFC 55 "WR_PARM" you may alter the parametrization in the module during runtime. The time needed until the new parametrization is valid can last up to 50ms. During this time, the measuring value output is 7FFFFh. The following table shows the structure of the parameter data:

Record set B4h

Byte	Bit 7 ... 0	Default
0	Wire break recognition ■ Bit 0: Wire break recognition channel 0 ■ Bit 1: Wire break recognition channel 1 ■ Bit 2: Wire break recognition channel 2 ■ Bit 3: Wire break recognition channel 3 ■ Bit 4: Wire break recognition channel 4 – 0 = Off (de-activated) – 1 = On (activated) ■ Bit 7 ... 5: reserved	00h
1	■ Bit 4 ... 0: reserved ■ Bit 5: Reaction at CPU_STOP channel 5 – 0 = Set replacement value * – 1 = Store last value ■ Bit 6: Reaction at CPU_STOP channel 6 – 0 = Set replacement value * – 1 = Store last value ■ Bit 7: reserved	00h
2	Channel 0: Function *'Input area (channel 0 ... 3)' on page 143*	19h
3	Channel 1: Function *'Input area (channel 0 ... 3)' on page 143*	19h
4	Channel 2: Function *'Input area (channel 0 ... 3)' on page 143*	19h
5	Channel 3: Function *'Input area (channel 0 ... 3)' on page 143*	19h
6	Channel 4: Function *'Input area (channel 4)' on page 144*	00h
7	Channel 0: Measuring cycle *'Structure measuring cycle byte:' on page 143*	00h
8	Channel 1: Measuring cycle *'Structure measuring cycle byte:' on page 143*	00h

Byte	Bit 7 ... 0	Default
9	Channel 2: Measuring cycle ↳ *'Structure measuring cycle byte:' on page 143*	00h
10	Channel 3: Measuring cycle ↳ *'Structure measuring cycle byte:' on page 143*	00h
11	Channel 4: Measuring cycle ↳ *'Input area (channel 4)' on page 144*	00h
12	Channel 5: Function ↳ *'Output area (channel 5, channel 6)' on page 145*	19h
13	Channel 6: Function ↳ *'Output area (channel 5, channel 6)' on page 145*	19h
14	Channel 5: High byte substitute value	00h
15	Channel 5: Low byte substitute value	00h
16	Channel 6: High byte substitute value	00h
17	Channel 6: Low byte substitute value	00h

*) If you want to get 0A respectively 0V as output value at CPU-STOP, you have to set the replacement value E500h.

Parameter

- Wire break recognition
  - The bits 0 ... 4 of byte 0 allow you to activate the wire break recognition for the input channels.
  - The wire break recognition is only available for the current measuring range of 4...20mA and thermo resistance measurement.
  - A wire break is recognized, a diagnostic entry is made and displayed by the SF LED when the current during current measuring (4 ... 20mA) sinks under 1.18mA respectively at thermo resistance measurement the resistance becomes endless.
  - If additionally a diagnostic interrupt is activated, a diagnostic message is sent to the superordinated system.
- Diagnostic interrupt
  - The diagnostic interrupt is global released for the digital and analog part.
    ↳ *Chapter 6.10 'Counter - Parametrization' on page 155*
    In case of an error like e.g. wire break, the superordinated system receives record set 0. For a channel specific diagnostic you may then call record set 1.

- CPU-Stop reaction and substitute value
  - Via byte 14 ... 17 you predefine a substitute value for the output channel as soon as the CPU switches to Stop.
  - By setting bit 5 respectively 6, the last output value remains in the output at CPU Stop. A reset sets the replacement value.
- Function no.
  - Here you set the function no. of your measuring respectively output function for every channel. These can be found in the corresponding function no. assignment from the table for the input respectively output area.
- Measuring cycle
  - Here you may set the transducer velocity for every input channel. Please regard that a higher transducer velocity causes a lower resolution because of the lower integration time. The data transfer format remains unchanged. Only the lower bits (LSBs) are not longer relevant for the analog value.

Structure measuring cycle byte:

Byte	Bit 7 ... 0	Default
7 ... 11	■ Bit 3 ... 0: Velocity per channel – 0000: 15 conversions/s, resolution: 16 – 0001: 30 conversions/s, resolution: 16 – 0010: 60 conversions/s, resolution: 15 – 0011: 120 conversions/s, resolution: 14 – 0100: 170 conversions/s, resolution: 12 – 0101: 200 conversions/s, resolution: 10 – 0110: 3.7 conversions/s, resolution: 16 – 0111: 7.5 conversions/s, resolution: 16 ■ Bit 7 ... 4: reserved	00h

Function no. assignment

Input area (channel 0 ... 3)

No.	Function	Input area
19h	Voltage ±10V S7 format from Siemens	±11.76V 11.76V= End overdrive region (32511) -10...10V = Nominal range (-27648 ... 27648) -11.76V = End underdrive region (-32512) two's complement
18h	Voltage 0...10V S7 format from Siemens	0...11.76V 11.76V = End overdrive region (32511) 0...10V = Nominal range (0 ... 27648) no underdrive region available

No.	Function	Input area
24h	Current ±20mA S7 format from Siemens	±23.52mA 23.52mA = End overdrive region (32511) -20...20mA = Nominal range (-27648 ... 27648) -23.52mA = End underdrive region (-32512) two's complement
23h	Current 4...20mA S7 format from Siemens	1.185...22.81mA 22.81mA = End overdrive region (32511) 4...20mA = Nominal range (0 ... 27648) 1.185mA = End underdrive region (-4864) two's complement
22h	Current 0...20mA S7 format from Siemens	0...23.52mA 23.52mA = End overdrive region (32511) 0...20mA = Nominal range (0 ... 27648) no underdrive region available
00h	Channel not active (turned off)	

Input area (channel 4)

No.	Function	Input area
82h	Pt100 in 2wire mode	-240...1000°C 1000°C = End overdrive region (10000) -200...+850°C = Nominal range (-2000 ... 8500) -240°C = End underdrive region (-2400) two's complement
85h	Pt100 in 2wire mode	-240...600°C 600°C = End overdrive region (6000) -200...+500°C = Nominal range (-2000 ... 5000) -240°C = End underdrive region (-2400) two's complement
83h	NI100 in 2wire mode	-105...295°C 295°C = End overdrive region (2950) -50...+250°C = Nominal range (-500 ... 2500) -105°C = End underdrive region (-1050) two's complement
86h	NI1000 in 2wire mode	-105...270°C 270°C = End overdrive region (2700) -50...+250°C = Nominal range (-500 ... 2500) -105°C = End underdrive region (-1050) two's complement

No.	Function	Input area
46h	Resistance measurement 600Ω 2wire	0...705.5Ω 705.5Ω = End overdrive region (32511) 0...600Ω = Nominal range (0 ... 27648) no underdrive region available
00h	Channel not active (turned off)	

Output area (channel 5, channel 6)

No.	Function	Output area
19h	Voltage ±10V S7 format from Siemens	±11.76V 11.76V= End overdrive region (32511) -10...10mA = Nominal range (-27648 ... 27648) -11.76V = End underdrive region (-32512) two's complement
18h	Voltage 0...10V S7 format from Siemens	0...11.76V 11.76V = End overdrive region (32511) 0...10V = Nominal range (0 ... 27648) no underdrive region available
24h	Current ±20mA S7 format from Siemens	±23.52mA 23.52mA = End overdrive region (32511) -20...20mA = Nominal range (-27648 ... 27648) -23.52mA = End underdrive region (-32512) two's complement
23h	Current 4...20mA S7 format from Siemens	0...22.81mA 22.81mA = End overdrive region (32511) 4...20mA = Nominal range (0 ... 27648) 0mA = End underdrive region (-6912) two's complement
22h	Current 0...20mA S7 format from Siemens	0...23.52mA 23.52mA = End overdrive region (32511) 0...20mA = Nominal range (0 ... 27648) no underdrive region available
00h	Channel not active (turned off)	

Leaving the defined range, the output is 0V respectively 0A!

6.6 Analog part - Diagnostic functions

Overview

As soon as you've activated the diagnostic interrupt release in the parametrization, the following events can release a diagnostic interrupt:

- Wire break
- Parametrization error
- Measuring range underflow
- Measuring range overflow

At a pending diagnostic the CPU interrupts the user application and jumps to the OB 82 for diagnostic$_{incoming}$. This OB allows you with an according programming to monitor detailed diagnostic information via the SFCs 51 or 59 and to react to it. After the execution of the OB 82 the user application processing is continued. The diagnostic data is consistent until leaving the OB 82. After error correction automatically a diagnostic$_{going}$ occurs if the diagnostic interrupt release is still active. In the following the record sets for diagnostic$_{incoming}$ and diagnostic$_{going}$ are specified.

Record sets

Record set 0 - diagnostic$_{incoming}$

Byte	Bit 7 ... 0
0	■ Bit 0: set at module failure ■ Bit 1: 0 (fix) ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 4: set at external auxiliary supply missing ■ Bit 6 ... 5: reserved ■ Bit 7: set at error in parametrization
1	■ Bit 3 ... 0: Module class – 0101b analog module ■ Bit 4: Channel information available ■ Bit 7 ... 5: reserved
2	00h (fix)
3	00h (fix)

After error correction automatically a diagnostic$_{going}$ occurs if the diagnostic interrupt release is still active.

Record set 0 - diagnostic$_{outgoing}$

Byte	Bit 7 ... 0
0	00h (fix)
1	■ Bit 3 ... 0: Module class – 0101b analog module ■ Bit 4: Channel information available ■ Bit 7 ... 5: reserved

Byte	Bit 7 ... 0
2	00h (fix)
3	00h (fix)

Record set 1 - channel specific diagnostic$_{incoming}$ (Byte 0 ... 14)

Byte	Bit 7 ... 0
1...3	Content record set 0 *'Record set 0 - diagnostic$_{incoming}$' on page 146*
4	■ Bit 6 ... 0: Channel type (here 74h) – 70h: Digital input – 71h: Analog input – 72h: Digital output – 73h: Analog output – 74h: Analog input/output ■ Bit 7: 0 (fix)
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 07h)
7	■ Bit 0: Channel error channel 0 ■ Bit 1: Channel error channel 1 ■ Bit 2: Channel error channel 2 ■ Bit 3: Channel error channel 3 ■ Bit 4: Channel error channel 4 ■ Bit 5: Channel error channel 5 ■ Bit 6: Channel error channel 6 ■ Bit 7: 0 (fix)
8	■ Bit 0: Parametrization error channel 0 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: 0 (fix) ■ Bit 4: Wire break channel 0 ■ Bit 5: 0 (fix) ■ Bit 6: Measuring range underflow channel 0 ■ Bit 7: Measuring range overflow channel 0
9	■ Bit 0: Parametrization error channel 1 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: 0 (fix) ■ Bit 4: Wire break channel 1 ■ Bit 5: 0 (fix) ■ Bit 6: Measuring range underflow channel 1 ■ Bit 7: Measuring range overflow channel 1

Byte	Bit 7 ... 0
10	■ Bit 0: Parametrization error channel 2 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: 0 (fix) ■ Bit 4: Wire break channel 2 ■ Bit 5: 0 (fix) ■ Bit 6: Measuring range underflow channel 2 ■ Bit 7: Measuring range overflow channel 2
11	■ Bit 0: Parametrization error channel 3 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: 0 (fix) ■ Bit 4: Wire break channel 3 ■ Bit 5: 0 (fix) ■ Bit 6: Measuring range underflow channel 3 ■ Bit 7: Measuring range overflow channel 3
12	■ Bit 0: Parametrization error channel 4 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: 0 (fix) ■ Bit 4: Wire break channel 4 ■ Bit 5: 0 (fix) ■ Bit 6: Measuring range underflow channel 4 ■ Bit 7: Measuring range overflow channel 4
13	■ Bit 0: Parametrization error channel 5 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: Short circuit channel 5 ■ Bit 4: Wire break channel 5 ■ Bit 7 ... 5: 0 (fix)
14	■ Bit 0: Parametrization error channel 6 ■ Bit 1: 0 (fix) ■ Bit 2: 0 (fix) ■ Bit 3: Short circuit channel 6 ■ Bit 4: Wire break channel 6 ■ Bit 7 ... 5: 0 (fix)

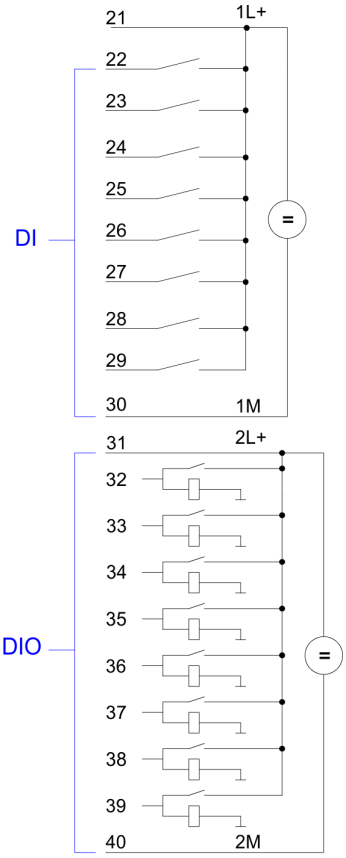
6.7 Digital part

Overview

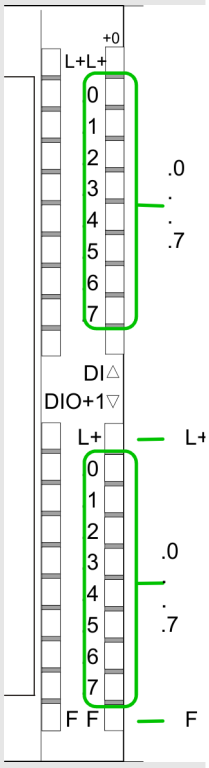
- The digital part consists of 8 input and 8 in-/output channels. Each of these channels shows its state via a LED.
- By means of the parametrization you may assign interrupt properties to every digital input.
- Additionally you may parametrize the digital inputs as counter with max. 100kHz.

- The output channels provide a diagnostic function, i.e. as soon as an output is active, the concerning input is set to "1".
- At a short circuit at the load, the input is set to "0" and the error may be recognized by evaluating the input.
- The DIO area has to be provided with external DC 24V.

CAUTION!
Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+. Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal. A thus connected output remains active even with shut down supply voltage. Non-observance may cause damages of the module.



CPU 314-6CF23: Digital part pin assignment and status indicator

Pin	Assignment	LEDs	Description
1	Power supply +DC 24 V DI		*DI:* **.0 ... .7** LED (green) I+0.0 to I+0.7 starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated *DIO:* **2L+** LED (green) Supply voltage available for DIO **.0 ... .7** LED (green) I/Q+1.0 to I/Q +1.7 on at active output/input **F** LED (red) Overload or short circuit error
2	I+0.0 / Counter 0(A)		
3	I+0.1 / Counter 0(B)		
4	I+0.2 / Gate0/Latch0/Reset0		
5	I+0.3 / Counter 1(A)		
6	I+0.4 / Counter 1(B)		
7	I+0.5 / Gate1/Latch1/Reset1		
8	I+0.6 / Counter 2(A)		
9	I+0.7 / Counter 2(B)		
10	Ground DI		
11	Power supply +DC 24 V DIO		
12	I/Q+1.0 / Gate2/Latch2/Reset2		
13	I/Q+1.1 / Counter 3(A)		
14	I/Q+1.2 / Counter 3(B)		
15	I/Q+1.3 / Gate3/Latch3/Reset3		
16	I/Q+1.4 / OUT0/Latch0/Reset0		
17	I/Q+1.5 / OUT1/Latch1/Reset1		
18	I/Q+1.6 / OUT2/Latch2/Reset2		
19	I/Q+1.7 / OUT3/Latch3/Reset3		
20	Ground DIO		

Access to the digital part

- By including the SPEEDBUS.GSD in your hardware configurator, the module is at your disposal in the hardware catalog. After the installation of the GSD you will find the CPU at *'Additional field devices ➔ I/O ➔ VIPA_SpeedBus'*. 314-6CF23.
- If there is no hardware configuration available, the in- and output areas starting with address 1024 are mapped to the address range of the CPU.
- For each input bit the status is stored in the data input area.
- For the output you have to enter a value into the data output are.

Used area *'Input area' on page 135*

Addr.	Name	Byte	Function
+0	DI_0	1	Digital input I+0.0 ... I+0.7
+1	DI_1	1	Digital input I+1.0 ... I+1.7
+16	CVCL_0	4	Counter value / latch value counter 0
+20	-	2	reserved
+22	ISTS_0	2	Input status counter 0
+24	CVCL_1	4	Counter value / latch value counter 1
+28	-	2	reserved
+30	ISTS_1	2	Input status counter 1
+32	CVCL_2	4	Counter value / latch value counter 2
+36	-	2	reserved
+38	ISTS_2	2	Input status counter 2
+40	CVCL_3	4	Counter value / latch value counter 3
+44	-	2	reserved
+46	ISTS_3	2	Input status counter 3

Used area *'Output area' on page 136*

Addr.	Name	Byte	Function
+0	-	1	reserved
+1	DO_1	1	Digital output Q+1.0 ... Q+1.7
+10	OSTS_0	2	Output status counter 0
+12	-	2	reserved
+14	OSTS_1	2	Output status counter 1
+16	-	2	reserved
+18	OSTS_2	2	Output status counter 2
+20	-	2	reserved
+22	OSTS_3	2	Output status counter 3

6.8 Counter - Fast introduction

Overview

- The CPU 314-6CF23 has 4 parametrizable counters integrated that may be controlled separately.
- Each counter is controlled via an internal gate.
- The counter behavior and the assignment of the inputs is configurable for each counter.
- During the count process the counter signal is recognized and evaluated.
- Every counter occupies one double word in the input address area with the counter register and in the input and output area one word for the status.

Pre-set respectively parametrize counter

By including the SPEEDBUS.GSD you may pre-set all counter parameters via a hardware configuration. Except of the parameter in record set 0, you may change parameters during runtime by using the SFC 55, 56, 57 and 58. For this you have to transfer the wanted parameters via record set to the counter by using the according SFC in the user application. Here you may define among others:

- Interrupt behavior
- Assignment I/O (Gate, Latch, Reset, OUT)
- Input filter
- Counter operating mode respectively behavior
- Start value for load value, end value and comparison value register

Chapter 6.10 'Counter - Parametrization' on page 155

Control counter

The counter is controlled via the internal gate (I-gate). The I-gate is the sum of hardware- (HW) and software-gate (SW), where the HW-gate evaluation may be deactivated via the parametrization.

'Gate function' on page 168

Read counter

Depending on the status setting, the counter register contains the current counter value (input status bit 0 = 0) or the current latch value (input status bit 0 = 1). By setting the output status bit 8 the current latch value is transferred to the counter register in the input area. By setting the output status bit 8, the current counter value is transferred.

Chapter 6.9 'Counter - In-/output area' on page 152

Counter status word

Besides of the counter register in the input area you may find a status word for every counter in the in- respectively output area. You may monitor the status or influence the counter by setting according bits like e.g. activate the SW gate.

'ISTS_X Input status' on page 154,

'OSTS_X Output status word' on page 154

Counter inputs (connections)

For not all inputs are available at the same time, you may set the input assignment for every counter via the parametrization.

↳ *'CPU 314-6CF23: Digital part pin assignment and status indicator ' on page 135*

For each counter the following inputs are available:

- $Counter_x$ (A)
  - Pulse input for counter signal respectively track A of an encoder. Here you may connect encoder with 1-, 2- or 4-tier evaluation.
- $Counter_x$ (B)
  - Direction signal respectively track B of the encoder. Via the parametrization you may invert the direction signal.

The following inputs may be assigned to a pin at the module via parametrization:

- $Gate_x$
  - This input allows you to open the HW gate with a high peek and thus start a count process.
- $Latch_x$
  - With a positive edge at $Latch_x$ the current counter value is stored in a memory that you may read if needed.
- $Reset_x$
  - As long as $Reset_x$ is applied with a positive level the counter is still reset to the load value.

Counter outputs

Every counter has an assigned output channel. ↳ *'Record set 0 - Counter mode' on page 156*

The following behavior for the output channel can be set via parametrization:

- No comparison: output is not controlled
- Counter value ≥ comparison value: output is set
- Counter value ≤ comparison value: output is set
- Counter value = comparison value: output is set

Maximum count frequency

The maximum count frequency is 100kHz, independent from the number of activated counters.

6.9 Counter - In-/output area

Access to the digital part

- By including the SPEEDBUS.GSD in your hardware configurator, the module is at your disposal in the hardware catalog. After the installation of the GSD you will find the CPU at *'Additional field devices ➔ I/O ➔ VIPA_SpeedBus'*. 314-6CF23.
- If there is no hardware configuration available, the in- and output areas starting with address 1024 are mapped to the address range of the CPU.
- For each input bit the status is stored in the data input area.
- For the output you have to enter a value into the data output are.

Used area *'Input area' on page 135*

Addr.	Name	Byte	Function
+0	DI_0	1	Digital input I+0.0 ... I+0.7
+1	DI_1	1	Digital input I+1.0 ... I+1.7
+16	CVCL_0	4	Counter value / latch value counter 0
+20	-	2	reserved
+22	ISTS_0	2	Input status counter 0
+24	CVCL_1	4	Counter value / latch value counter 1
+28	-	2	reserved
+30	ISTS_1	2	Input status counter 1
+32	CVCL_2	4	Counter value / latch value counter 2
+36	-	2	reserved
+38	ISTS_2	2	Input status counter 2
+40	CVCL_3	4	Counter value / latch value counter 3
+44	-	2	reserved
+46	ISTS_3	2	Input status counter 3

Used area *'Output area' on page 136*

Addr.	Name	Byte	Function
+0	-	1	reserved
+1	DO_1	1	Digital output Q+1.0 ... Q+1.7
+10	OSTS_0	2	Output status counter 0
+12	-	2	reserved
+14	OSTS_1	2	Output status counter 1
+16	-	2	reserved
+18	OSTS_2	2	Output status counter 2
+20	-	2	reserved
+22	OSTS_3	2	Output status counter 3

Counter value counter$_X$ The *counter value* always contains the current value of the counter.

Latch value counter$_X$ An edge 0-1 at the digital Latch input stores the current *counter value* in as *latch value*.

ISTS_x Input status

The status word in the input area has the following structure:

Bit	Name	Function
0	COUNT_LTCH	■ 0: Value in the input image is counter value ■ 1: Value in the input image is latch value
1	CTRL_C_DO	Is set when the digital output is enabled.
2	STS_SW_GATE	■ 0: Software gate (SW gate) is not active ■ 1: Software gate (SW gate) is active
3	reserved	reserved
4	STS_HW_GATE	■ 0: Hardware gate (HW gate) is not active ■ 1: Hardware gate (HW gate) is active
5	STS_I_GATE	■ 0: Internal gate (I gate) is not active ■ 1: Internal gate (I gate) is active
6	STS_DO	■ 0: Counter output (DO) = "0" ■ 1: Counter output (DO) = "1"
7	STS_C_DN	Is set at counter direction backwards.
8	STS_C_UP	Is set at counter direction forward.
9	STS_CMP*	Is set when *counter value* = *comparison value*. If comparison is parametrized never, the bit is never set.
10	STS_END*	Is set when *counter value* = *end value*.
11	STS_OFLW*	Is set at overflow.
12	STS_UFLW*	Is set at underflow.
13	STS_ZP*	Is set at zero-crossing.
14	STS_L	■ 0: Latch input is not active ■ 1: Latch input is active
15	NEW_L	Is set if value in the latch register has changed.

*) The bits remain set until reset with RES_STS (output status: bit 6).

OSTS_x Output status word

After setting a bit in the output status word this is immediately reset. Please regard that setting and resetting of a function at the output status word takes place with different bits.

The status word in the output area has the following structure:

Bit	Name	Function
0	GET_C_VAL	By setting the current counter value is transferred to the process image.
1	SET_C_DO	By setting the digital output (DO) is enabled for the counter. Then the output may only be controlled by the counter.

Bit	Name	Function
2	SET_SW_GATE	By setting the software gate is set (not allowed in OB 100).
3	reserved	-
4	reserved	-
5	SET_C_VAL	By setting the counter may be temporarily set to a value, which was pre-set via record set (9A+x)h before.
6	RES_STS	By setting the status bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP are reset.
7	reserved	-
8	GET_L_VAL	By setting the latch value is transferred to the process image.
9	RES_C_DO	By setting the digital output (DO) is disabled for the counter. Then the output may only be controlled by the process image.
10	RES_SW_GATE	By setting the software gate is reset.
12	reserved	-
...	...	...
15	reserved	-

6.10 Counter - Parametrization

Overview

The parametrization takes place in the hardware configurator. Here, parameter data are transferred existing of the following components:

Byte	Record set	Description
16	0h	Counter mode C0 ... C3
4	7Fh	Diagnostics interrupt
16	80h	Edge selection for process interrupt
32	81h	Filter value I+0.0 ... I+1.7
16	82 ... 86h	C0: Comparison, set, end value, hysteresis, pulse
16	87h	C0: Sum parameter (comparison, set, end value, hysteresis and pulse)
16	88 ... 8Ch	C1: Comparison, set, end value, hysteresis, pulse
16	8Dh	C1: Sum parameter (comparison, set, end value, hysteresis and pulse)
16	8E ... 92h	C2: Comparison, set, end value, hysteresis, pulse
16	93h	C2: Sum parameter (comparison, set, end value, hysteresis and pulse)
16	94 ... 98h	C3: Comparison, set, end value, hysteresis, pulse
16	99h	C3: Sum parameter (comparison, set, end value, hysteresis and pulse)

Byte	Record set	Description
4	9Ah	C0: Count value that is transferred to counter by setting bit 5 in the output status word
4	9Bh	C1: Count value that is transferred to counter by setting bit 5 in the output status word
4	9Ch	C2: Count value that is transferred to counter by setting bit 5 in the output status word
4	9Dh	C3: Count value that is transferred to counter by setting bit 5 in the output status word
2	9Eh	Activate respectively de-activate analog/digital part

Except of the parameter in record set 0, you may transfer the other parameters during runtime by using the SFC 55, 56, 57 and 58 to the digital part. For this you have to transfer the wanted parameters via record set to the counter by using the according SFC in the user application.

Record set 0 - Counter mode

Byte	Bit 7...0
0...3	Counter mode C0
4...7	Counter mode C1
8...11	Counter mode C2
12...15	Counter mode C3

- Via the record set 0 you may preset a counter mode for every counter as double word.
- Record set 0 may not be transferred during runtime.

The double word for the counter mode has the following structure:

Byte	Bit 7 ... 0
0	■ Bit 2 ... 0: Signal evaluation – 000b = Counter de-activated At de-activated counter the further parameter settings for this counter are ignored and the according I/O channel is set as "normal" output if this should be used as output. – 001b = Encoder 1-tier (at counter$_X$ (A_X) and counter$_X$ (B_X)) – 010b = Encoder 2-tier (at counter$_X$ (A_X) and counter$_X$ (B_X)) – 011b = Encoder 4-tier (at counter$_X$ (A_X) and counter$_X$ (B_X)) – 100b = Pulse/direction (pulse at counter$_X$ (A_X) and direction at counter$_X$ (B_X)) ■ Bit 6 ... 3: C_X Input (function of the counter input as gate, latch or reset) – 0000b = de-activated (counter starts at set SW gate) – 0001b = Gate$_X$ The input of counter$_X$ serves as gate. High level at gate activates the HW gate. The counter may only start when HW and SW gate are set. – 0010b = Monoflop * – 0100b = Latch$_X$ (edge 0-1 at input saves counter value of counter$_X$) – 1000b = Reset$_X$ (positive level at input resets counter$_X$) ■ Bit 7: Gate function (internal gate) – 0 = abort (count process starts again at load value) – 1 = interrupt (count process continues with counter value)
1	■ Bit 2 ... 0: Output set (OUT_X of counter$_X$ is set when condition is met) – 000b = never – 001b = counter value ≥ comparison value – 010b = counter value ≤ comparison value – 100b = counter value = comparison value ■ Bit 3: Count direction – 0 = Count direction inverted: OFF (count direction at B_X not inverted) – 1 = Count direction inverted: ON (count direction at B_X inverted) ■ Bit 7 ... 4: reserved

Byte	Bit 7 ... 0
2	■ Bit 5 ... 0: Counter function *Chapter 6.11 'Counter - Functions' on page 162* – 000000b = count endless – 000001b = once: forward – 000010b = once: backwards – 000100b = once: no main direction – 001000b = periodic: forward – 010000b = periodic: backwards – 100000b = periodic: no main direction ■ Bit 7 ... 6: C_X In-/Output (Function of the counter I/O as OUT, Latch or Reset) – 00b = O: OUT_X (at comparison function) – 01b = I: $Latch_X$ (edge 0-1 saves counter value of $counter_X$) – 10b = I: $Reset_X$ (Positive level resets $counter_X$)
3	■ Bit 5 ... 0: Interrupt behavior: – Bit 0: Process interrupt HW gate open Bit 1: Process interrupt HW gate closed Bit 2: Process interrupt overflow Bit 3: Process interrupt underrun Bit 4: Process interrupt comparison value Bit 5: Process interrupt end value By setting the bits you may activate the wanted process interrupts. ■ Bit 7 ... 6: reserved

*) This is not supported at this time.

Record set 7Fh - Diagnostic interrupt

Byte	Bit 15...0
0...1	■ Diagnostic interrupt – 0000h = de-activated – 0001h = activated
2...3	■ reserved

- Here you activate respectively de-activate the diagnostic function.
- A diagnostic interrupt occurs when during a process interrupt execution another process interrupt is initialized for the same event.

Record set 80h - Edge selection

Byte	Bit 7...0
0	■ Bit 1 ... 0: Edge selection I+0.0 ■ Bit 7 ... 2: reserved
1	■ Bit 1 ... 0: Edge selection I+0.1 ■ Bit 7 ... 2: reserved
2	■ Bit 1 ... 0: Edge selection I+0.2 ■ Bit 7 ... 2: reserved

Byte	Bit 7...0
3	■ Bit 1 ... 0: Edge selection I+0.3 ■ Bit 7 ... 2: reserved
4	■ Bit 1 ... 0: Edge selection I+0.4 ■ Bit 7 ... 2: reserved
5	■ Bit 1 ... 0: Edge selection I+0.5 ■ Bit 7 ... 2: reserved
6	■ Bit 1 ... 0: Edge selection I+0.6 ■ Bit 7 ... 2: reserved
7	■ Bit 1 ... 0: Edge selection I+0.7 ■ Bit 7 ... 2: reserved
8	■ Bit 1 ... 0: Edge selection I+1.0 ■ Bit 7 ... 2: reserved
9	■ Bit 1 ... 0: Edge selection I+1.1 ■ Bit 7 ... 2: reserved
10	■ Bit 1 ... 0: Edge selection I+1.2 ■ Bit 7 ... 2: reserved
11	■ Bit 1 ... 0: Edge selection I+1.3 ■ Bit 7 ... 2: reserved
12	■ Bit 1 ... 0: Edge selection I+1.4 ■ Bit 7 ... 2: reserved
13	■ Bit 1 ... 0: Edge selection I+1.5 ■ Bit 7 ... 2: reserved
14	■ Bit 1 ... 0: Edge selection I+1.6 ■ Bit 7 ... 2: reserved
15	■ Bit 1 ... 0: Edge selection I+1.7 ■ Bit 7 ... 2: reserved

- Via this record set you may activate a process interrupt for I+0.0 ... I+1.7 and define for which edge type of the input signal a process interrupt is thrown.
- Edge selection
  - 00b = de-activated
  - 01b = Hardware interrupt at edge 0-1
  - 10b = Hardware interrupt at edge 1-0
  - 11b = Hardware interrupt at edge 0-1 and 1-0

Record set 81h - Input filter

Byte	Bit 15...0
0...1	■ Input filter I+0.0 in 2.56µs
2...3	■ Input filter I+0.1 in 2.56µs
4...5	■ Input filter I+0.2 in 2.56µs
6...7	■ Input filter I+0.3 in 2.56µs

Byte	Bit 15...0
8...9	■ Input filter I+0.4 in 2.56µs
10...11	■ Input filter I+0.5 in 2.56µs
12...13	■ Input filter I+0.6 in 2.56µs
14...15	■ Input filter I+0.7 in 2.56µs
16...17	■ Input filter I+1.0 in 2.56µs
18...19	■ Input filter I+1.1 in 2.56µs
20...21	■ Input filter I+1.2 in 2.56µs
22...23	■ Input filter I+1.3 in 2.56µs
24...25	■ Input filter I+1.4 in 2.56µs
26...27	■ Input filter I+1.5 in 2.56µs
28...29	■ Input filter I+1.6 in 2.56µs
30...31	■ Input filter I+1.7 in 2.56µs

- This record set allows you to preset an input filter in steps of 2.56µs steps for I+0.0 ... I+1.7.
- By preceding a filter you define how long an input signal must be present before it is recognized as "1" signal. With the help of filters you may e.g. filter signal peaks at a blurred input signal.
- The entry happens as a factor of 2.56µs and is within the range 1 ... 16000 i.e. 2.56µs ... 40.96ms.

Record set 82 ... 99h - Counter parameter

Counter 0 87h	Counter 1 8Dh	Counter 2 93h	Counter 3 99h	Type	Function
82h	88h	8Eh	94h	Double word	comparison value
83h	89h	8Fh	95h	Double word	Load value
84h	8Ah	90h	96h	Double word	End value
85h	8Bh	91h	97h	Word	Hysteresis
86h	8Ch	92h	98h	Word	Pulse

- Each of the counter parameters has an assigned record set depending on the counter number.
- Additionally for every counter the parameter are summoned in one record set.
- For every counter the record sets have the same structure.

Functions *Chapter 6.12 'Counter - Additional functions' on page 167*

- Comparison value
  - Via the parametrization you may preset a comparison value that may influence the counter output respectively throw a process interrupt when compared with the recent counter value.
  - The behavior of the output respectively the process interrupt has to be set via the record set 0.
- Load value, end value
  - You may define a main counting direction for every counter via the parametrization.
  - If "none" or "endless" is chosen, the complete counting range is available:

    Lower counter limit: - 2 147 483 648 (-2^{31})

    Upper counter limit: + 2 147 483 648 ($-2^{31}-1$)
  - Otherwise you may set an upper and a lower limit by setting a *load value* as start and an *end value*.
- Hysteresis
  - The hysteresis serves the avoidance of many toggle processes of the output and the interrupt, if the counter value is in the range of the comparison value.
  - You may set a range of 0 to 255.
  - The settings 0 and 1 deactivate the hysteresis.
  - The hysteresis influences zero run, comparison, over- and underflow.
- Pulse (Pulse duration)
  - The pulse duration tells for what time the output is set when the parametrized comparison criterion is reached respectively overstepped.
  - The pulse duration can be set in steps of 2.048ms between 0 and 522.24ms.
  - If the pulse duration = 0, the output is set active until the comparison condition is not longer fulfilled.

Record set 9A ... 9Dh - Set counter value temporary

A register can be preset using record set (9A+x)h. The current counter value is replaced by the register value by setting bit 5 of the output status word without any influence to the load value.

Record set 9Eh - Module selection

Byte	Bit 15...0
0...1	■ Module selection – 0000h = Digital/analog part activated (default) – 0001h = Digital part activated – 0002h = Analog part activated

- Using this record set you can de-activate the digital respectively analog part.
- Please regard that in spite of the de-activation of the digital respectively analog part, the process image for both components remains reserved.

6.11 Counter - Functions

Overview

You may count forward and backwards and choose between the following counter functions:

- Count endless, e.g. distance measuring with incremental encoder
- Count once, e.g. count to a maximum limit
- Count periodic, e.g. count with repeated counter process

In the operating modes "Count once" and "Count periodic" you may define a counter range as start and end value via the parameterization. For every counter additional parameterizable functions are available like gate function, comparison, hysteresis and process interrupt.

Main counting direction

Via the parameterization you have the opportunity to define a main counting direction for every counter. If "none" is chosen, the complete counting range is available:

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

Main counting direction forward

Upper restriction of the count range. The counter counts 0 res. *load value* in positive direction until the parameterized *end value* -1 and jumps then back to the load value with the next following encoder pulse.

Main counting direction backwards

Lower restriction of the count range. The counter counts from the parameterized start- res. *load value* in negative direction to the parameterized *end value* +1 and jumps then back to the start value with the next following encoder pulse.

Gate function abort/ interrupt

If the HW gate is enabled, only the HW gate may be influenced by the gate functions. An opening and closing of the SW gate aborts or interrupts the count process.

Abort count process

The count process starts after closing and restart of the gate beginning with the *load value*.

Interrupt count process

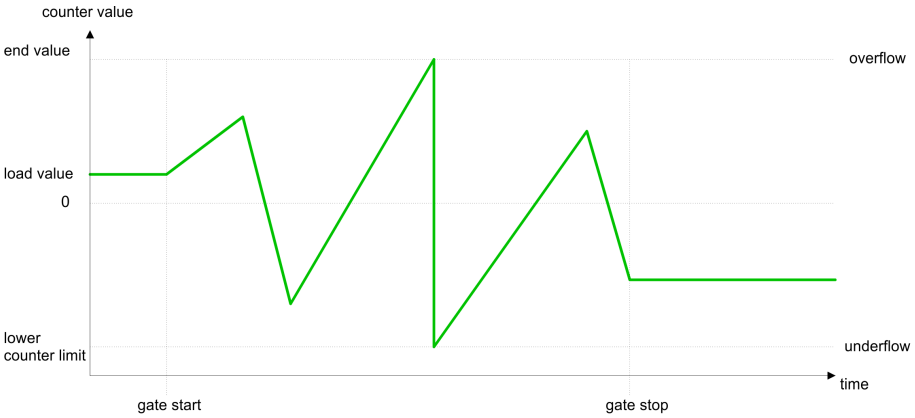
The count process continuous after closing and restart of the gate beginning with the last recent counter value.

Count continuously

In this operating mode, the counter counts from the load value. When the counter counts forward and reaches the upper count limit and another counting pulse in positive direction arrives, it jumps to the lower count limit and counts from there on. When the counter counts backwards and reaches the lower count limit and another counting pulse in negative direction arrives, it jumps to the upper count limit and counts from there on. The count limits are set to the maximum count range.

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

With overflow or underflow the status bits STS_OFLW respectively STS_UFLW are set. These bits remain set until these are reset with RES_STS. If enabled additionally a process interrupt is triggered.



Count Once

No main counting direction

- The counter counts once starting with the *load value*.
- You may count forward or backwards.
- The count limits are set to the maximum count range.
- At over- or underflow at the count limits, the counter jumps to the according other count limit and the internal gate is automatically closed and the status bits STS_OFLW respectively STS_UFLW are set. If enabled additionally a process interrupt is triggered.
- To restart the count process, you have to re-open the internal gate.
- At interrupting gate control, the count process continuous with the last recent *counter value*.
- At aborting gate control, the counter starts with the *load value*.

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

Interrupting gate control:

Aborting gate control:

Main counting direction forward

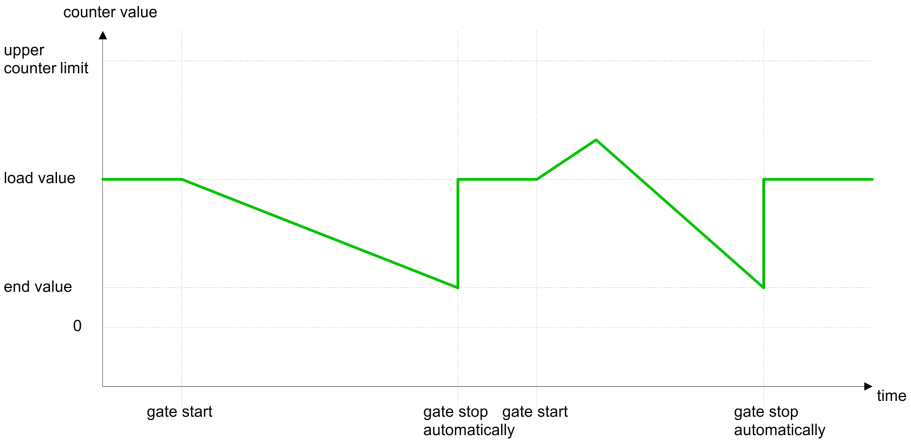
- The counter counts starting with the *load value*.
- When the counter reaches the end value -1 in positive direction, it jumps to the load value at the next positive count pulse and the gate is automatically closed.
- To restart the count process, you must create a positive edge of the gate. The counter starts with the load value.

Limits	Valid value range
Limit value	-2 147 483 647 (-2^{31} +1) to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})

Main counting direction backwards

- The counter counts backwards starting with the *load value*.
- When the counter reaches the end value +1 in negative direction, it jumps to the load value at the next negative count pulse and the gate is automatically closed.
- To restart the count process, you must create a positive edge of the gate. The counter starts with the load value.

Limits	Valid value range
Limit value	-2 147 483 648 (-2^{31}) to +2 147 483 646 (2^{31} -2)
Upper count limit	+2 147 483 647 (2^{31} -1)

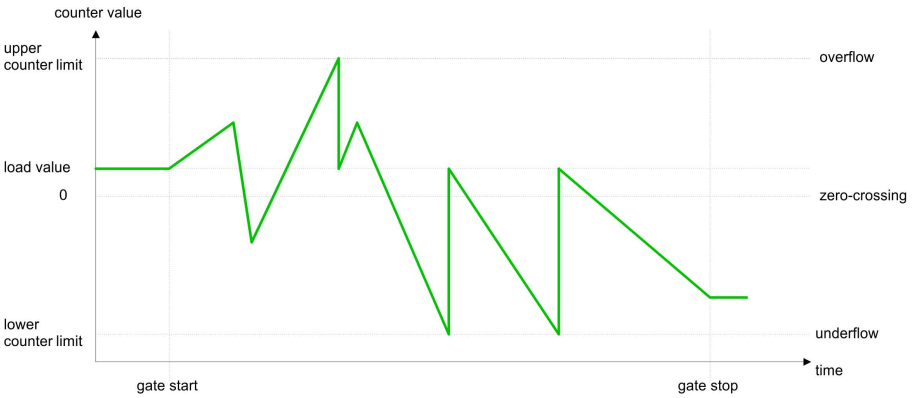


Count Periodically

No main counting direction

- The counter counts forward or backwards starting with the *load value*.
- At over- or underrun at the count limits, the counter jumps to the according other count limit and counts from there on.
- The count limits are set to the maximum count range.

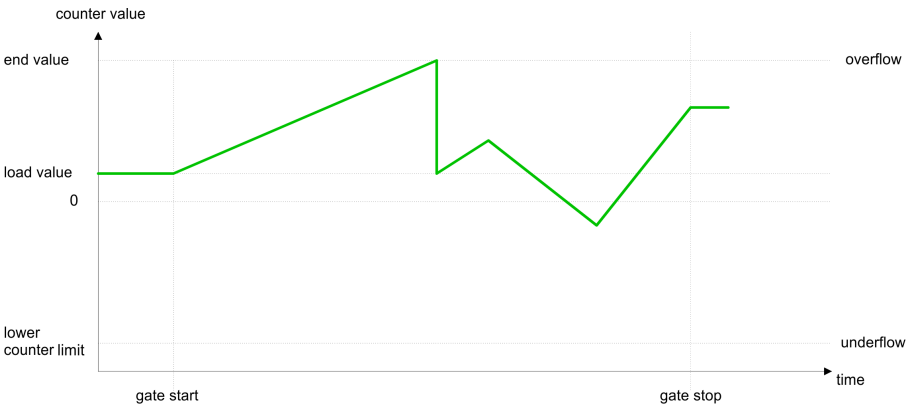
Limits	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31} -1)



Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the end value –1 in positive direction, it jumps to the load value at the next positive count pulse.

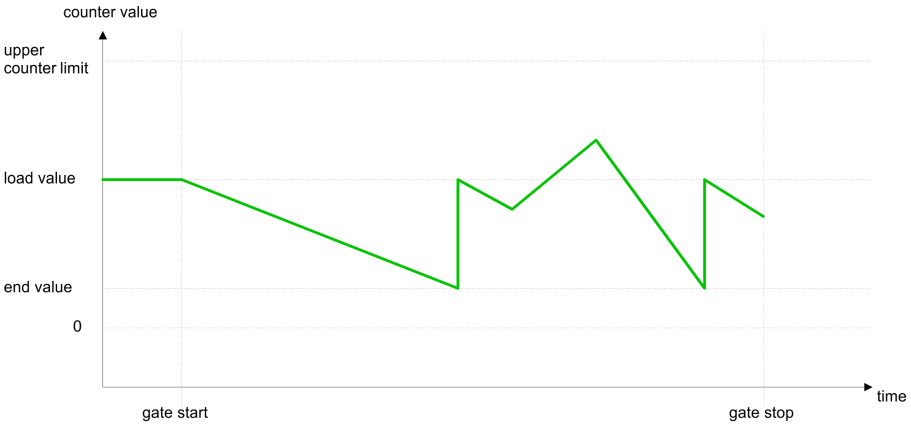
Limits	Valid value range
Limit value	-2 147 483 647 (-2^{31} +1) to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})



Main counting direction backwards

- The counter counts backwards starting with the *load value*.
- When the counter reaches the *end value* +1 in negative direction, it jumps to the *load value* at the next negative count pulse.
- You may exceed the upper count limit.

Limits	Valid value range
Limit value	-2 147 483 648 (-2^{31}) to +2 147 483 646 (2^{31} -2)
Upper count limit	+2 147 483 647 (2^{31} -1)



6.12 Counter - Additional functions

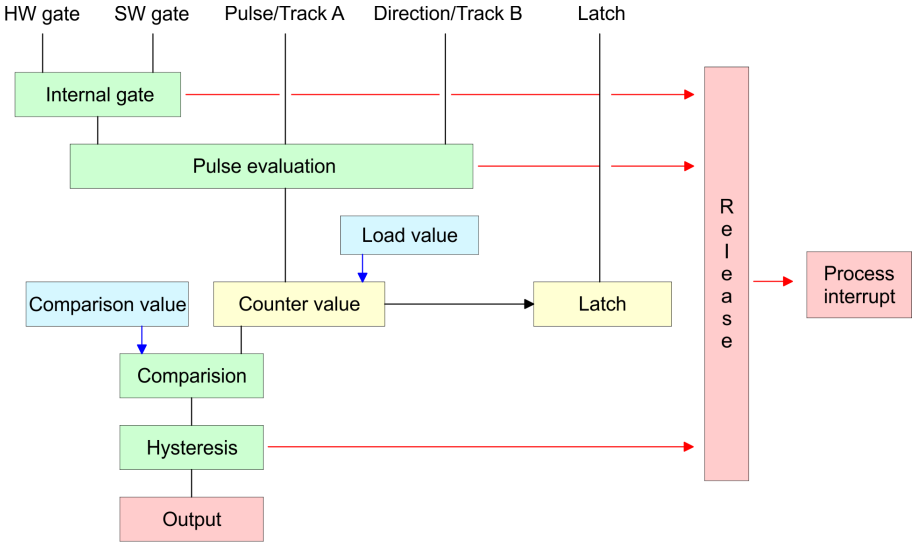
Overview

The following additional functions may be set via the parametrization of the counter:

- Gate function
  - The gate function is used to start , stop and interrupt a counting function.
- Latch function:
  - An edge 0-1 at the digital Latch input stores the current *counter value* in the latch register.
- Comparison
  - You may set a *comparison value* that de-activates a digital output respectively releases a process interrupt depending on the *counter value*.
- Hysteresis:
  - By specifying a *hysteresis* you can prevent e.g. a high output toggling and/or triggering interrupts, when the value of an encoder signal varies to the *comparison value*.

Schematic structure

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



Gate function

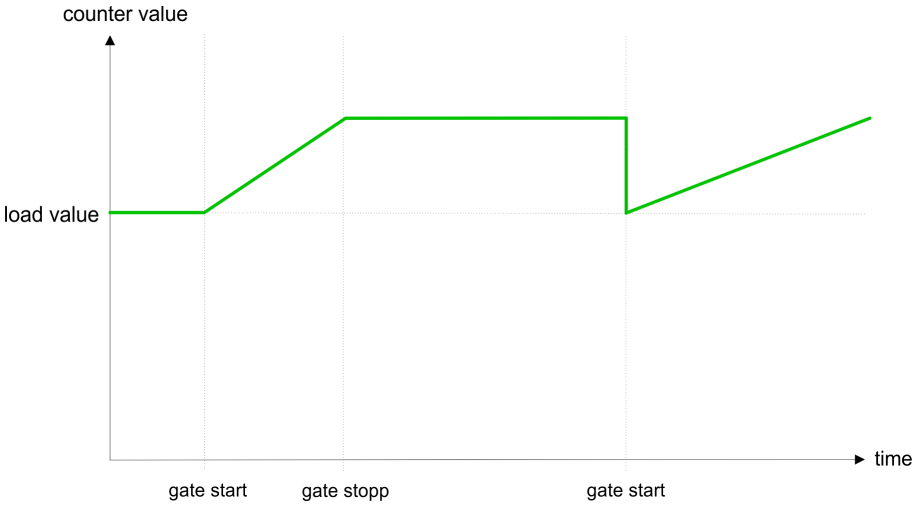
The activation respectively de-activation of a counter happens via an internal gate (I-gate). The I-gate consists of a software gate (SW-gate) and a hardware gate (HW-gate). The SW-gate is opened (activated) via your user application by setting the output status bit 2 for the according counter. The SW-gate is closed (de-activated) by setting the output status bit 10. The HW-gate is controlled via the digital $Gate_x$ input. The parametrization allows you to de-activate the consideration of the HW-gate so that the counter activation can take place only via the SW-gate. The following states influence the I-gate:

SW-gate	HW-gate	influences the I-gate
0	with edge 0-1	0
1	with edge 0-1	1
with edge 0-1	1	1
with edge 0-1	0	0
with edge 0-1	de-activated	1

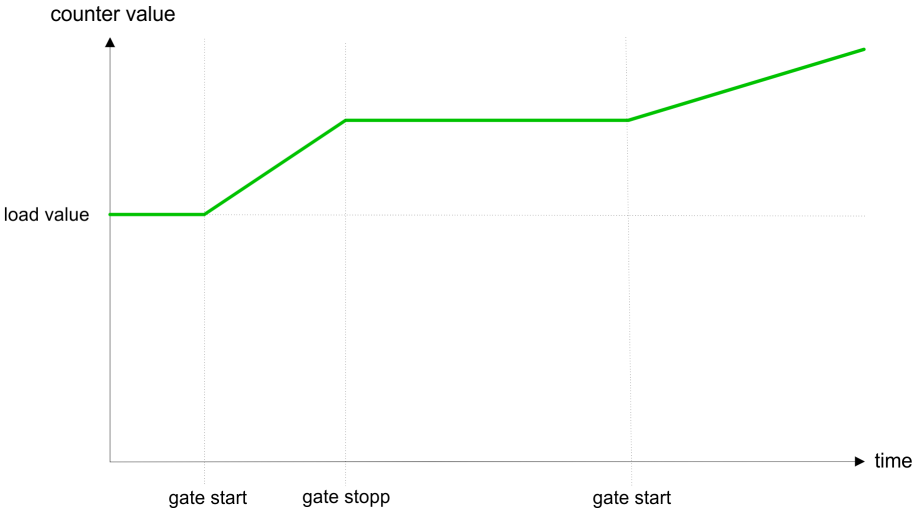
Gate function abort and interrupt

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

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



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



Gate control abort, interrupt

Gate control via *SW-gate*, aborting (parametrization: record set 0, byte 0, bit 7 ... 3 = 00000b)

SW-gate	HW-gate	Reaction counter
Edge 0-1	de-activated	Restart with *Load value*

Gate control via *SW-gate*, interrupting (parametrization: record set 0, byte 0, bit 7 ... 3 = 10000b)

SW-gate	HW-gate	Reaction counter
Edge 0-1	de-activated	Continue

Gate control via *SW/HW-gate*, aborting (parametrization: record set 0, byte 0, bit 7 ... 3 = 00001b)

SW-gate	HW-gate	Reaction counter
Edge 0-1	1	Continue
1	Edge 0-1	Restart with *Load value*

Gate control via *SW/HW-gate*, interrupting (parametrization: record set 0, byte 0, bit 7 ... 3 = 10001b)

SW-gate	HW-gate	Reaction counter
Edge 0-1	1	Continue
1	Edge 0-1	Continue

Gate control "Count once"

Gate control via *SW/HW gate*, operating mode "count once": If the internal gate has been closed automatically it may only be opened again under the following conditions:

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

Latch function

- As soon as during a count process a positive edge is recognized at the "Latch" input of a counter, the recent counter value is stored in the according latch register.
- You may access the latch register via the input image. For this set bit 8 of the output status word.
- At a new latch value additionally bit 15 is set in the input status word.
- By setting bit 8 in the output status word you may read the recent latch value of the according counter and reset the bit 15 of the input status word.

Comparison

You pre-define the behavior of the counter output via the parametrization:

- Output never switches
  - The output remains unaffected by the counter and is set as standard output.
- Output switches when counter value ≥ comparison value
  - The output remains set as long as the *counter value* is higher or equal *comparison value*.

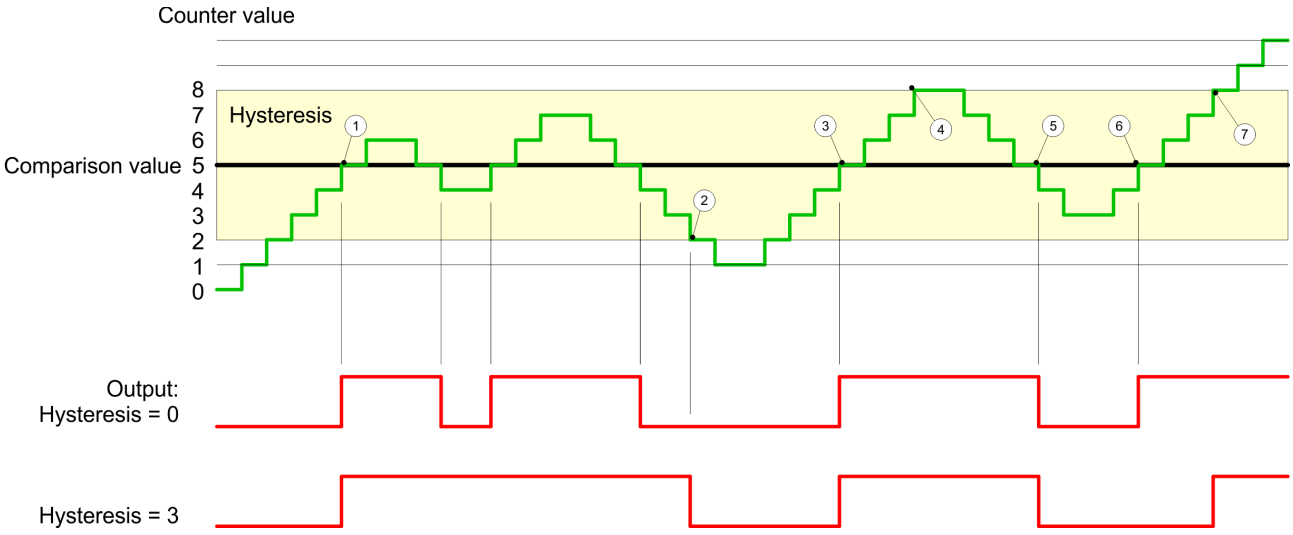
- Output switches when counter value ≤ comparison value
  - The output remains set as long as the *counter value* is lower or equal *comparison value*.
- Output set at comparison value (pulse at comparison value)
  - When the counter reaches the *comparison value* the output is set for the parametrized *pulse duration*.
  - If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled. When you've set a main counting direction the output is only set at reaching the comparison value from the main counting direction.
- Pulse duration
  - The pulse duration defines how long the output is to be set. It can be pre-set in steps of 2.048ms between 0 and 522.24ms.
  - The *pulse duration* starts with the setting of the according digital output.
  - The inaccuracy of the *pulse duration* is less than 2.048ms.
  - There is no past triggering of the *pulse duration* when the comparison value has been left and reached again during pulse output.

Hysteresis

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

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

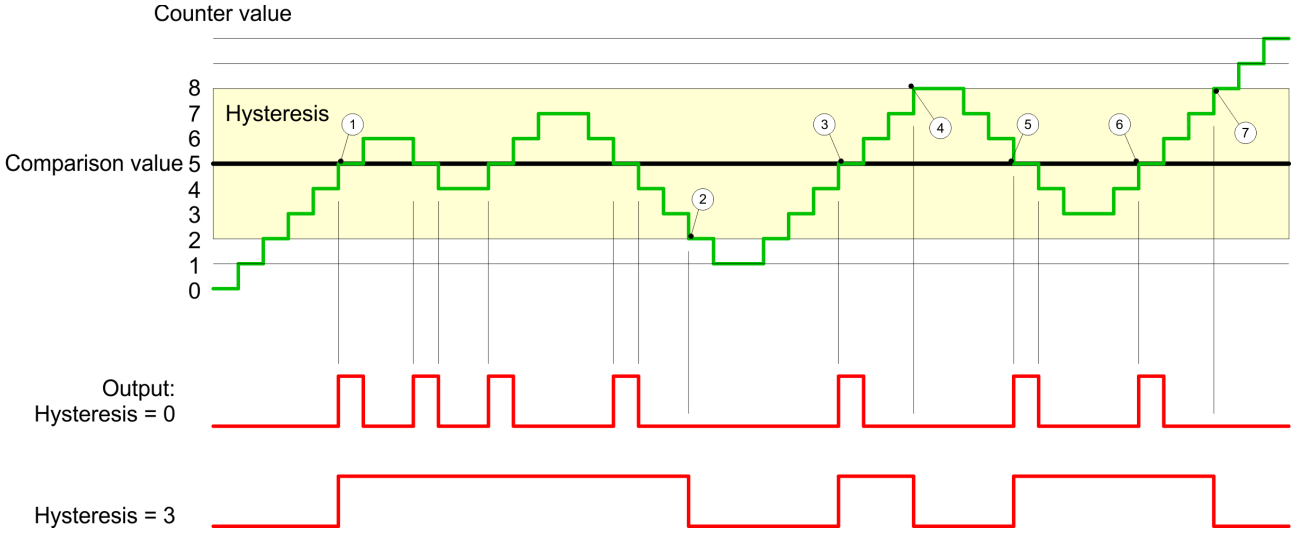
Effect at counter value ≥ comparison value



1 *Counter value* ≥ *comparison value* → output is set and *hysteresis* activated
2 Leave *hysteresis* range → output is reset
3 *Counter value* ≥ *comparison value* → output is set and *hysteresis* activated
4 Leave *hysteresis* range, output remains set for *counter value* ≥ *comparison value*
5 *counter value* < *comparison value* and *hysteresis* active → output is reset
6 *counter value* ≥ *comparison value* → output is not set for *hysteresis* active
7 Leave *hysteresis* range, output remains set for *counter value* ≥ *comparison value*

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

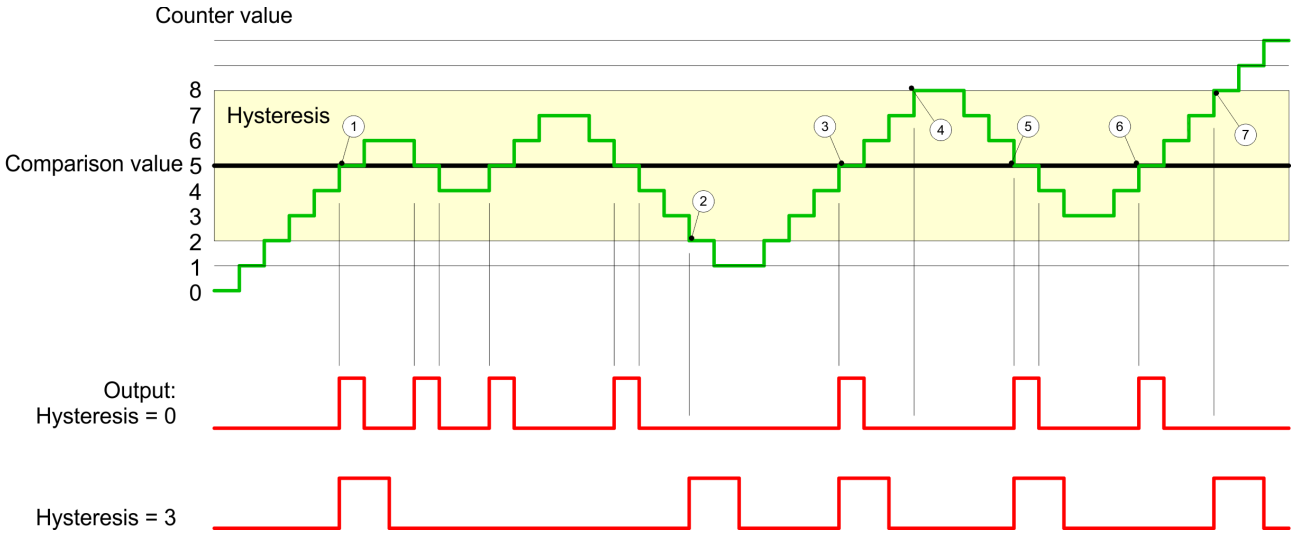
Effect at pulse at comparison value with pulse duration Zero



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

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

Effect at pulse at comparison value with pulse duration not zero



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

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

6.13 Counter - Diagnostic and interrupt

Overview

The parametrization allows you to define the following trigger for a process interrupt that may initialize a diagnostic interrupt:

- Status changes at an input
- Status changes at the HW-gate
- Over- respectively underflow or reaching an end value
- Reaching a comparison value

6.13.1 Process interrupt

Function

A process interrupt causes a call of the OB 40. Within the OB 40 you may find the logical basic address of the module that initialized the process interrupt by using the *local word 6*. More detailed information about the initializing event is to find in the *local double word 8*.

The *local double word 8* of the OB 40 has the following structure:

Local byte	Bit 7...0
8	■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7
9	■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7

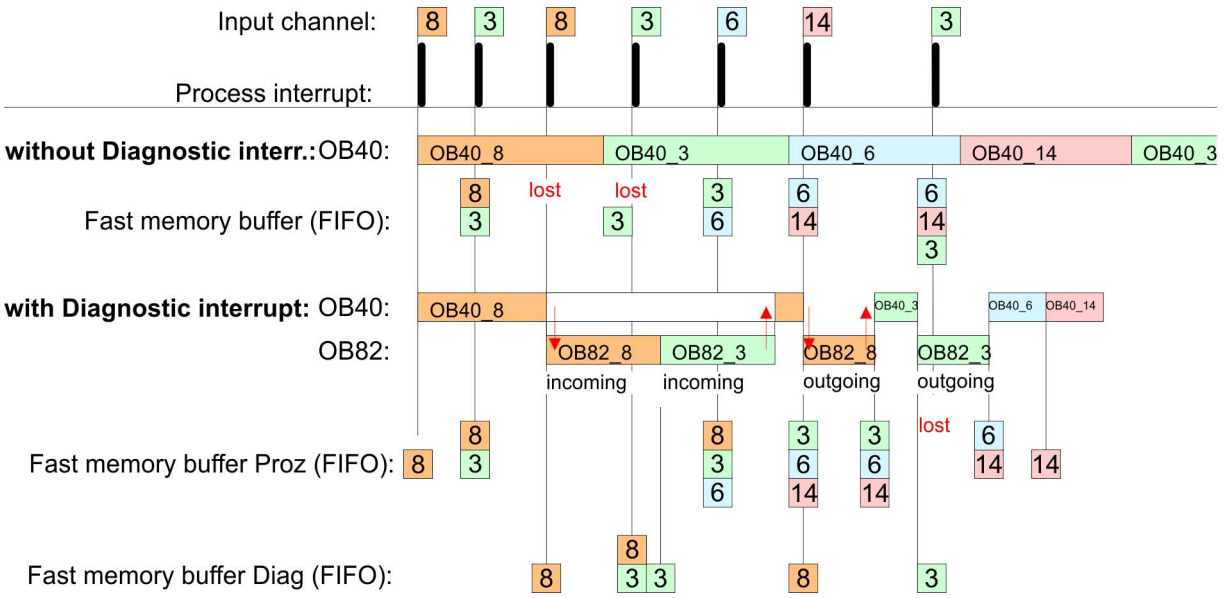
Local byte	Bit 7...0
10	■ Bit 0: Gate counter 0 open (activated) ■ Bit 1: Gate counter 0 closed ■ Bit 2: Over-/underflow/end value counter 0 ■ Bit 3: Counter 0 reached comparison value ■ Bit 4: Gate counter 1 open (activated) ■ Bit 5: Gate counter 1 closed ■ Bit 6: Over-/underflow/end value counter 1 ■ Bit 7: Counter 1 reached comparison value
11	■ Bit 0: Gate counter 2 open (activated) ■ Bit 1: Gate counter 2 closed ■ Bit 2: Over-/underflow/end value counter 2 ■ Bit 3: Counter 2 reached comparison value ■ Bit 4: Gate counter 3 open (activated) ■ Bit 5: Gate counter 3 closed ■ Bit 6: Over-/underflow/end value counter 3 ■ Bit 7: Counter 3 reached comparison value

6.13.2 Diagnostic interrupt

Function

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

Example:



Diagnostic interrupt processing

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

Record set 0 Diagnostic$_{incoming}$

Byte	Bit 7...0
0	■ Bit 0: set at module failure ■ Bit 1: 0 (fix) ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 4: set when external auxiliary supply is missing ■ Bit 7 ... 5: 0 (fix)
1	■ Bit 3 ... 0: Module class – 0101b: Analog – 1111b: Digital ■ Bit 4: Channel information present ■ Bit 7 ... 5: 0 (fix)

Byte	Bit 7...0
2	■ Bit 3 ... 0: 0 (fix) ■ Bit 4: Failure module internal supply voltage (output overload) ■ Bit 7 ... 5: 0 (fix)
3	■ Bit 5 ... 0: 0 (fix) ■ Bit 6: Process interrupt lost ■ Bit 7: 0 (fix)

Record set 0 Diagnostic$_{outgoing}$

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

Byte	Bit 7...0
0	■ Bit 0: set at module failure ■ Bit 1: 0 (fix) ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 4: set when external auxiliary supply is missing ■ Bit 7 ... 5: 0 (fix)
1	■ Bit 3 ... 0: Module class – 0101b: Analog – 1111b: Digital ■ Bit 4: Channel information present ■ Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	00h (fix)

Diagnostic record set 1

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

Byte	Bit 7...0
0 ... 3	Content record set 0 ↯ *'Record set 0 Diagnostic$_{incoming}$' on page 176*
4	■ Bit 6 ... 0: Channel type (here 70h) – 70h: Digital input – 71h: Analog input – 72h: Digital output – 73h: Analog output – 74h: Analog input/output ■ Bit 7: More channel types present – 0: no – 1: yes

Byte	Bit 7...0
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)
7	■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7 ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) ■ Bit 4: Error in channel group 4 (counter 0) ■ Bit 5: Error in channel group 5 (counter 1) ■ Bit 6: Error in channel group 6 (counter 2) ■ Bit 7: Error in channel group 7 (counter 3)
8	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix)
9	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix)
10	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... Input I+1.3 ■ Bit 7: 0 (fix)

Byte	Bit 7...0
11	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix)
12	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... gate counter 0 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 0 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value Counter 0 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 0 reached comparison value ■ Bit 7: 0 (fix)
13	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... gate counter 1 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 1 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value Counter 1 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 1 reached comparison value ■ Bit 7: 0 (fix)

Byte	Bit 7...0
14	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... gate counter 2 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 2 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value Counter 2 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 2 reached comparison value ■ Bit 7: 0 (fix)
15	Diagnostic interrupt due to process interrupt lost at... ■ Bit 0: ... gate counter 3 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 3 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value Counter 3 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 3 reached comparison value ■ Bit 7: 0 (fix)

7 Deployment PtP communication

7.1 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 "Operation list".

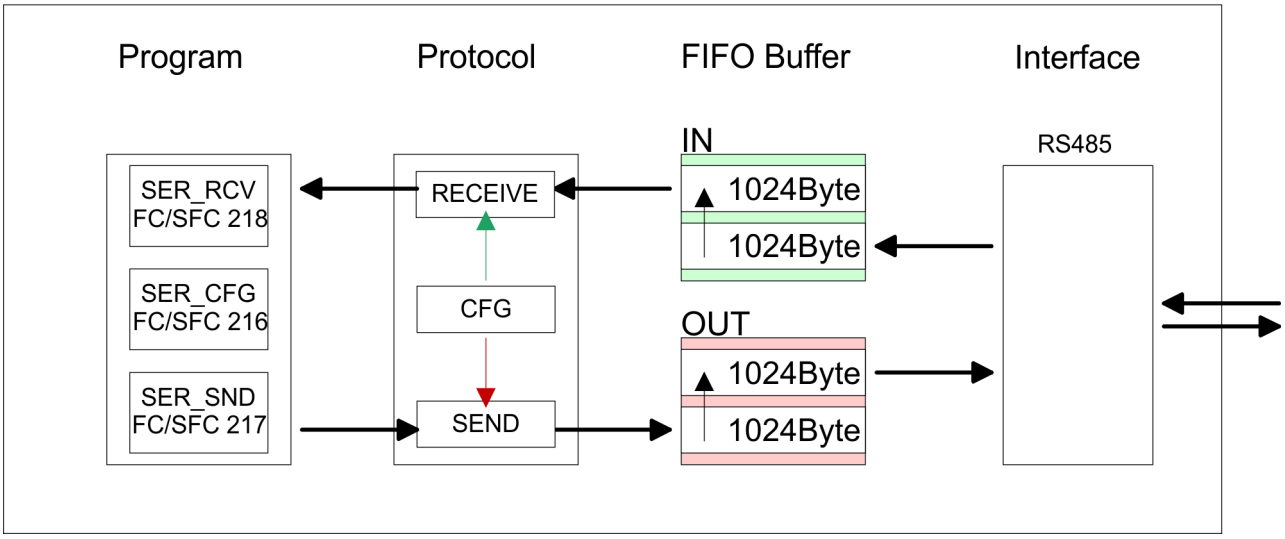
7.2 Principle of the data transfer

Overview

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.

RS485 PtP communication



7.3 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:

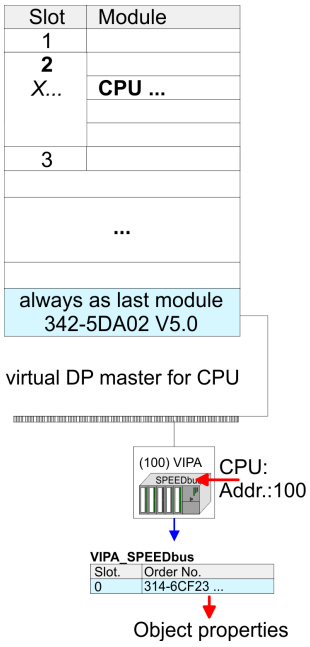
Name	Language
SPEEDBUS.GSD	German (default)
SPEEDBUS.GSG	German
SPEEDBUS.GSE	English

The GSD files may be found at www.vipa.com at the "Service" part.

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

1. Browse to www.vipa.com
2. Click to *'Service ➔ Download ➔ GSD- and EDS-Files ➔ Profibus'*
3. Download the file Cx000023_Vxxx.
4. Extract the file to your work directory. The SPEEDBUS.GSD is stored in the directory VIPA_System_300S.
5. Start the hardware configurator from Siemens.
6. Close every project.
7. Select *'Options ➔ Install new GSD-file'*.
8. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from VIPA may now be found in the hardware catalog at PROFIBUS-DP / Additional field devices / I/O / VIPA_SPEEDBUS.

Proceeding



The embedding of the CPU 314-6CF23 happens by means of a virtual PROFIBUS master system with the following approach:

1. Perform a hardware configuration for the CPU. *Chapter 5.4 'Hardware configuration - CPU' on page 57*
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 314-6CF23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 314-6CF23 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

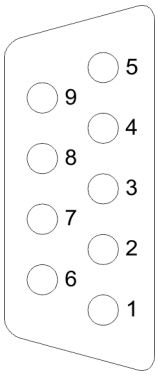
1. By double clicking the CPU 314-6CF23 placed in the slave system the properties dialog of the CPU may be opened.
2. Switch the Parameter *'Function RS485 X3'* to *'PtP'*.

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

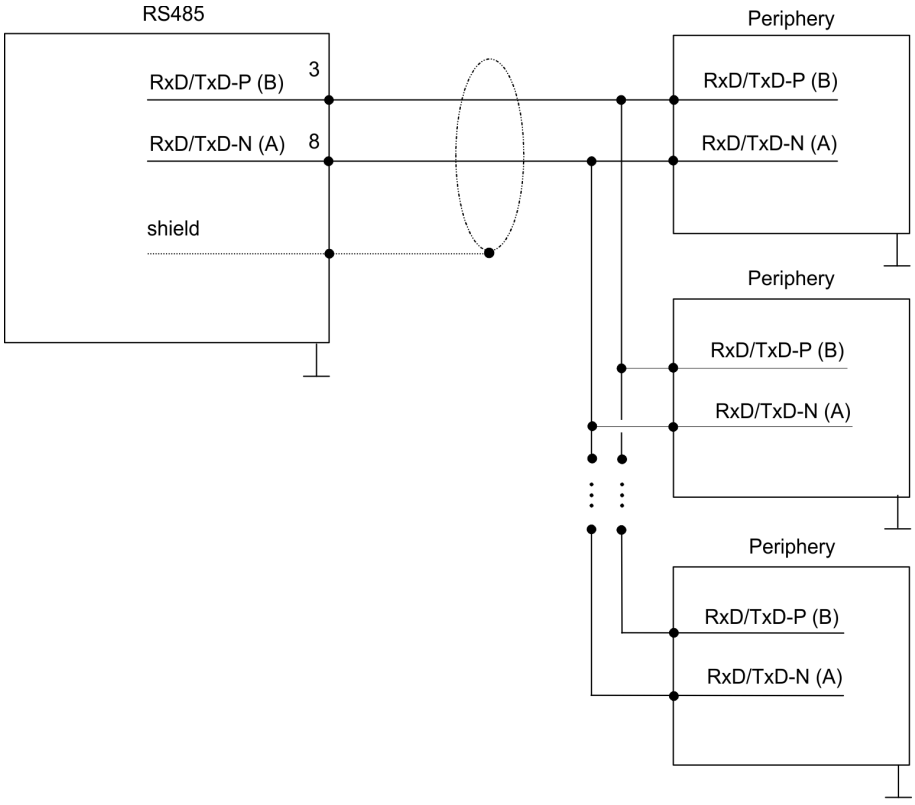
RS485

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



7.4 Parametrization

7.4.1 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.

7.5 Communication

7.5.1 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.

7.5.2 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 "Operation list".

7.6 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 parameterized 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. An according Receive_ASCII FB may be found within the VIPA library in the service area of www.vipa.com.

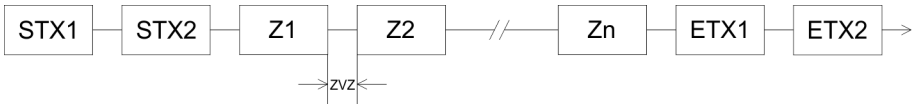
STX/ETX

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

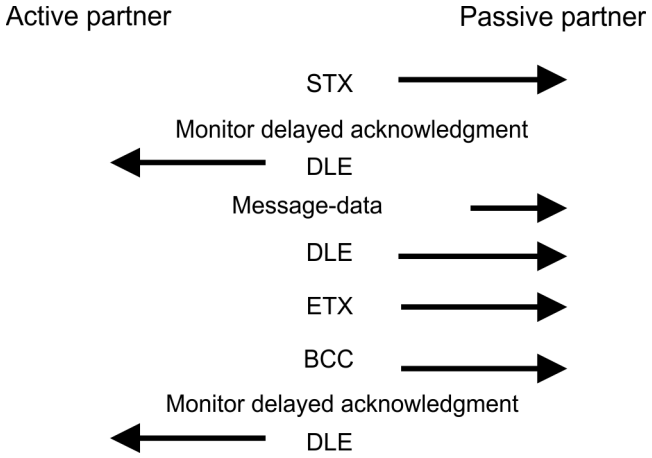
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

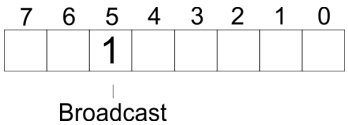
Slave master telegram

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

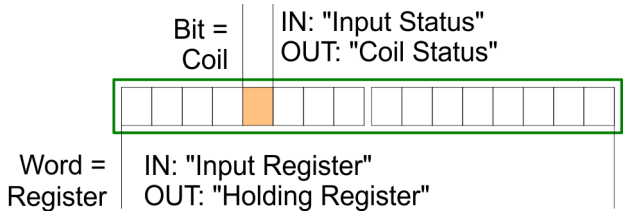
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

7.7 Modbus - Function codes

Naming convention

Modbus has some naming conventions:



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

Range definitions

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

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

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

0x - Bit area for master output data

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

1x - Bit area for master input data

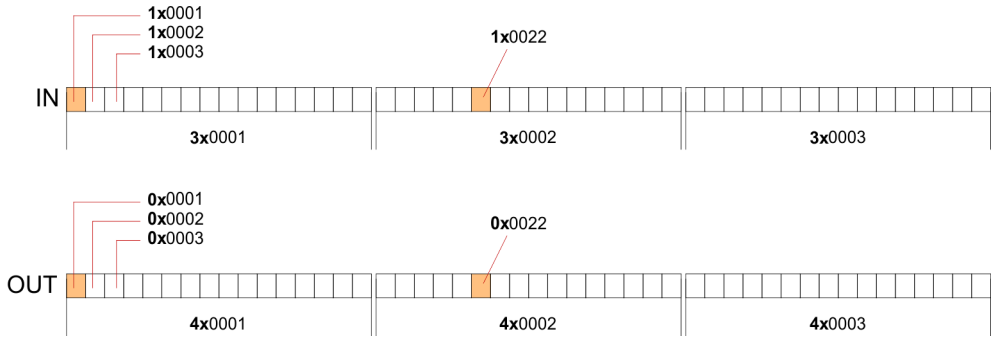
Access via function code 02h

3x - word area for master input data

Access via function code 04h

4x - word area for master output data

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



A description of the function codes follows below.

Overview

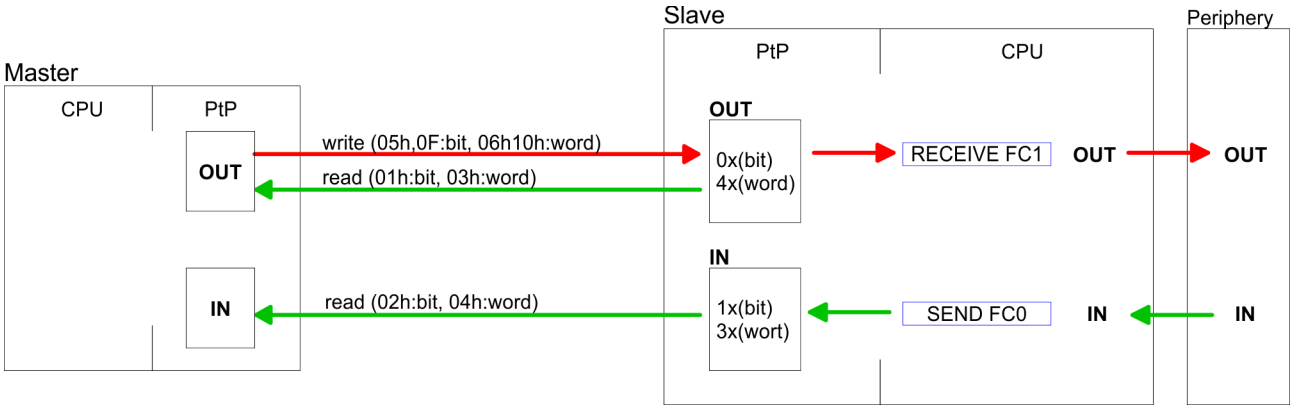
With the following Modbus function codes a Modbus master can access a Modbus slave: With the following Modbus function codes a Modbus master can access a Modbus slave. The description always takes place from the point of view of the master:

Code	Command	Description
01h	Read n bits	Read n bits of master output area 0x
02h	Read n bits	Read n bits of master input area 1x
03h	Read n words	Read n words of master output area 4x
04h	Read n words	Read n words master input area 3x

Code	Command	Description
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	Func-tion 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	Func-tion code	Address 1. word	Number of words	Number of bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1word	1word	1word	1word
					max. 125words			

Respond telegram

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

7.8 Modbus - Example communication

Overview

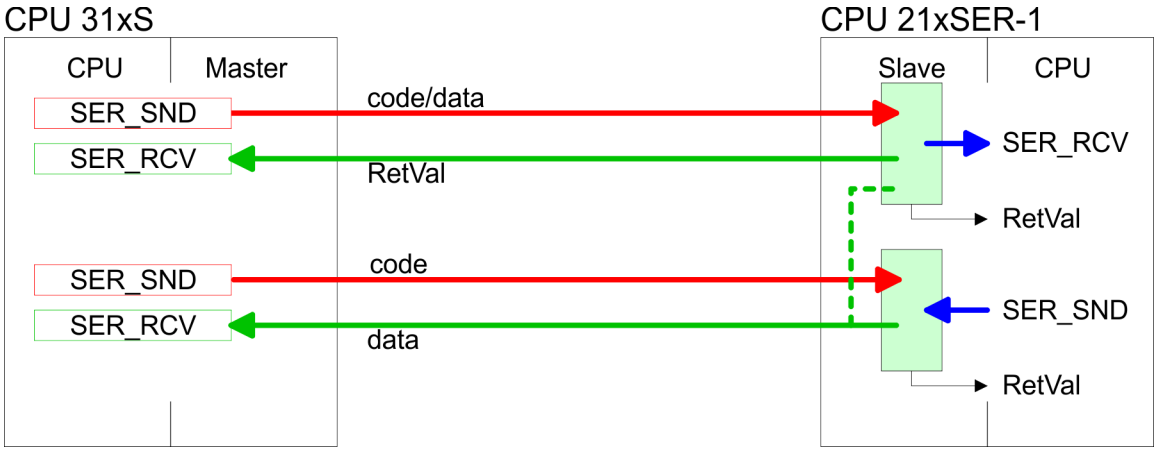
The example establishes a communication between a master and a slave via Modbus. The following combination options are shown:

- 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:



8 Deployment PROFIBUS communication

8.1 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.

8.2 Fast introduction

Overview

The PROFIBUS DP master is to be configured in the hardware configurator. Here the configuration happens by means of the sub module 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** ↳ *Chapter 5.10 'Project transfer' on page 72*

To be compatible to the Siemens SIMATIC Manager, the CPU 314-6CF23 from VIPA is to be configured as

CPU 317-2DP (6ES7 317-2AK14-0AB0 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 314-6CF23 is always to be configured as 1. module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens.

8.3 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	

To be compatible with the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'* number 2 the CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3) from Siemens.
4. The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X2 (DP).

8.3.1 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 *CPU-TYPE_318*. After this perform a power cycle.
  - *Chapter 5.19 'CMD - auto commands' on page 95*

```
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 *Chapter 5.15 'Reset to factory settings' on page 91*.
  - *Chapter 5.19 'CMD - auto commands' on page 95*

```
CMD_START
CPUTYPE_ORIGINAL
CMD_END
```

8.4 Deployment as PROFIBUS DP master

Precondition

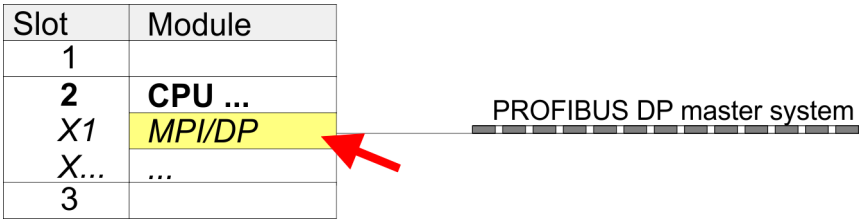
The hardware configuration described before was established.

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.

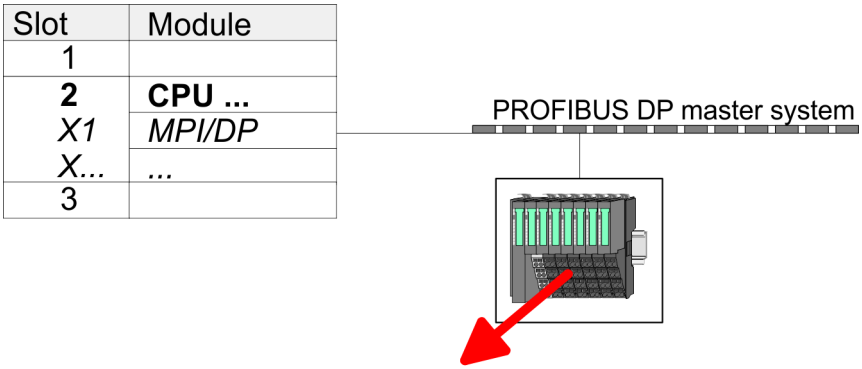
1. For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.

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			
...			

8.5 Deployment as PROFIBUS DP slave

Fast introduction

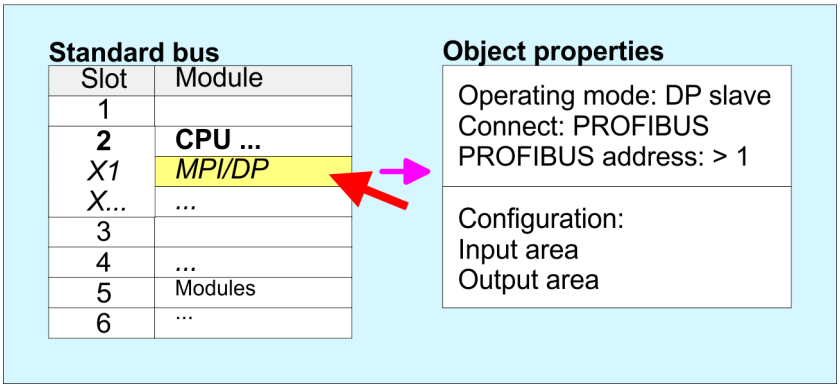
In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

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



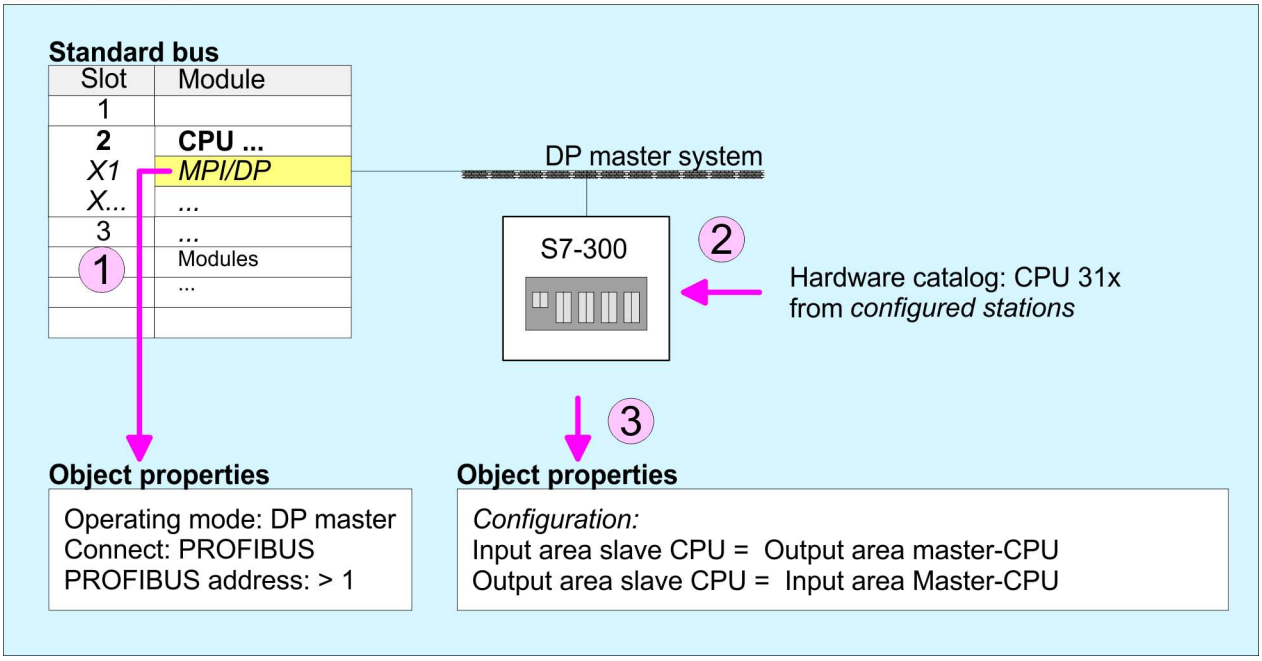
Project engineering master section

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.

8.6 PROFIBUS installation guidelines

PROFIBUS in general

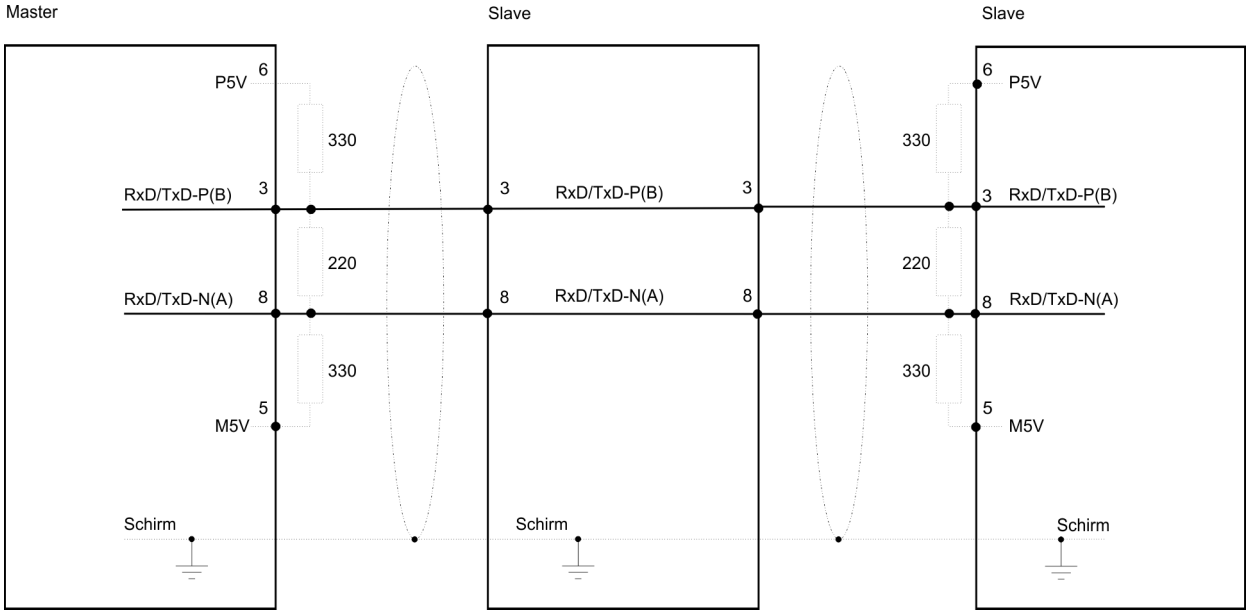
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.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

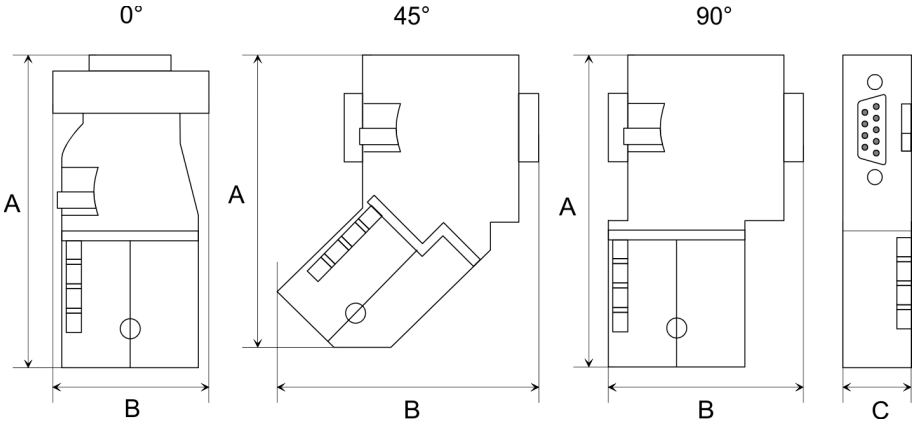
The following picture illustrates the terminating resistors of the respective start and end station.



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 "EasyConn" from VIPA. This is a bus connector with switchable terminating resistor and integrated bus diagnostic.



Dimensions in mm	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8

To connect this EasyConn plug, please use the standard PROFIBUS cable type A (EN50170). Starting with release 5 you also can use highly flexible bus cable:

Lapp Kabel order no: 2170222, 2170822, 2170322.

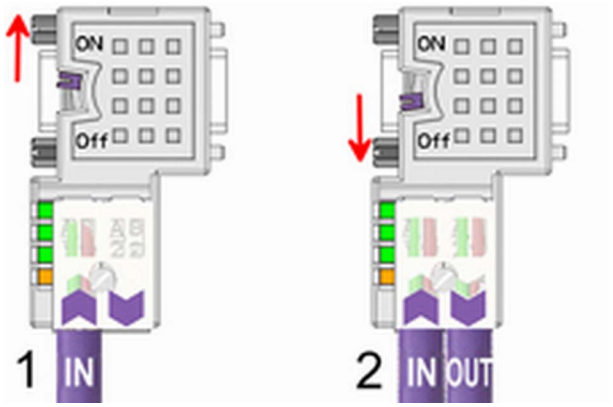
With the order no. 905-6AA00 VIPA offers the "Easy-Strip" de-isolating tool that makes the connection of the EasyConn much easier.

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!

8.7 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 VIPA 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.

9 WinPLC7

9.1 System conception

General

WinPLC7 is a programming and simulation software from VIPA 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 VIPA. 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 VIPA. Without any activation with the *demo version* the CPUs 11x of the System 100V from VIPA may be configured. To configure the SPEED7 CPUs a license for the "profi" version is necessary. This may be online be received from VIPA and activated.

There are the following sources to get WinPLC7:

- Online
  - At www.vipa.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.

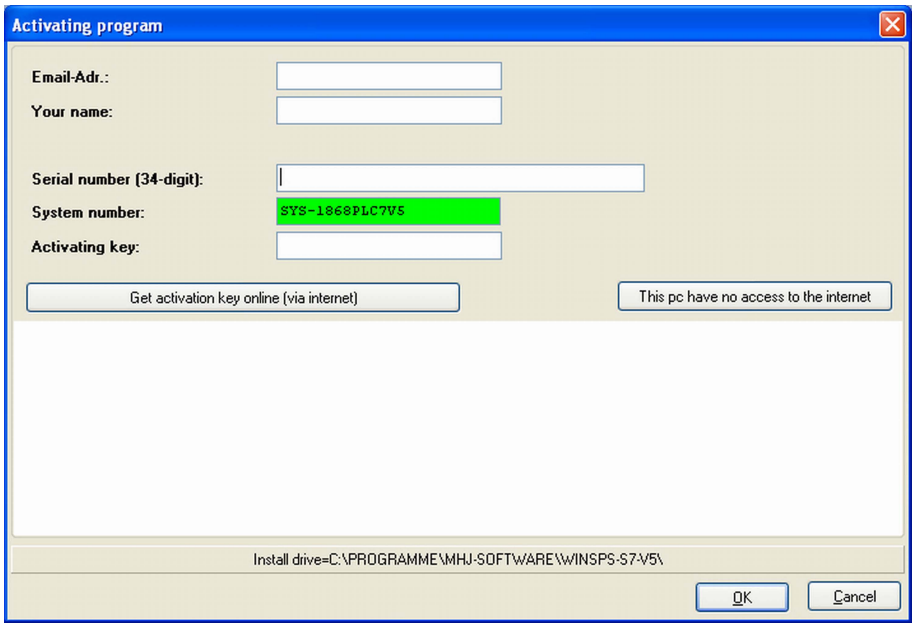
Installation WinPLC7 Demo

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

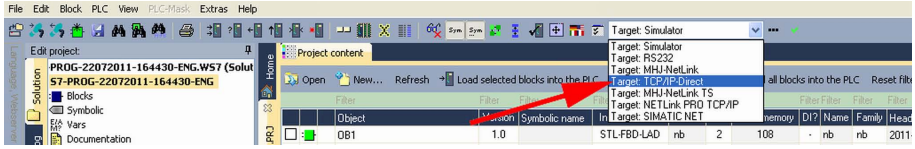
- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.
- One SPEED7 CPU and one digital output module are installed and cabled.
- The Ethernet PG/OP channel of the CPU is connected to your Ethernet network. Your CPU may be connected to your PC with an Ethernet 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

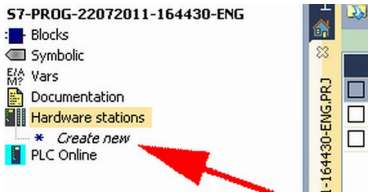
1. Start WinPLC7 ("Profi" version)
2. Create and open a new project with [Create a new solution].

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.

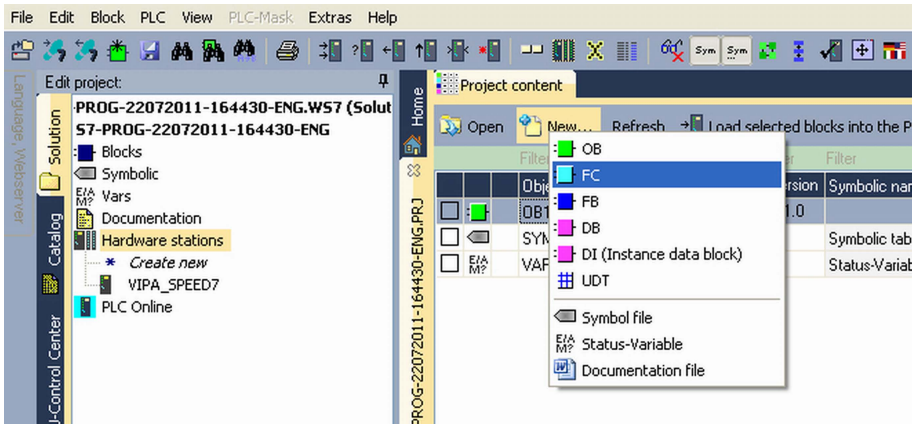
The hardware configuration is finished, now and the CPU may always be accessed by the IP parameters as well by means of WinPLC7.

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'*.

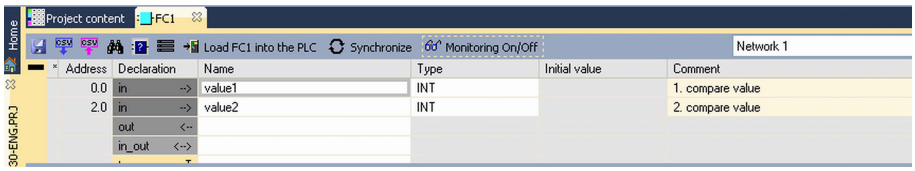


2. Enter "FC1" as block and confirm with [OK].
   ⇨ The editor for FC 1 is called.

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".

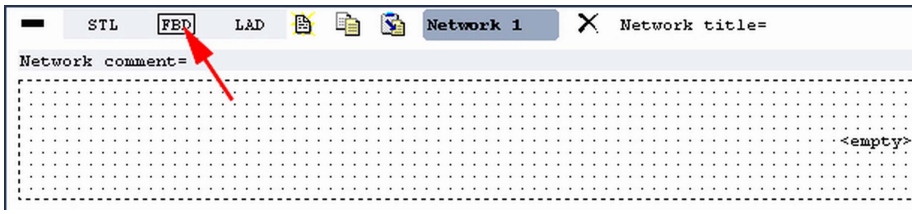
1. Select the *'in →'* row at the *'parameter table'* and enter at the field *'Name'* "value1". Press the *[Return]* key.
   ⇨ The cursor jumps to the column with the data type.
2. The data type may either directly be entered or be selected from a list of available data types by pressing the *[Return]* 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:



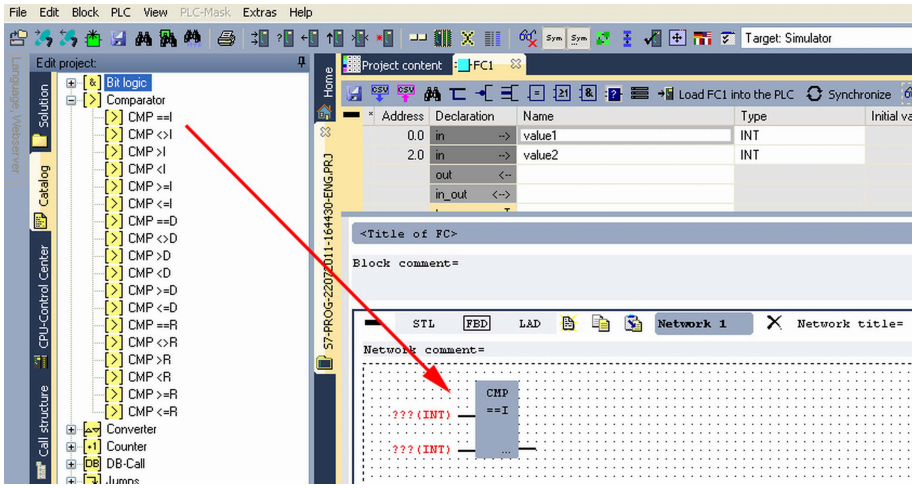
Enter the program

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 "#".

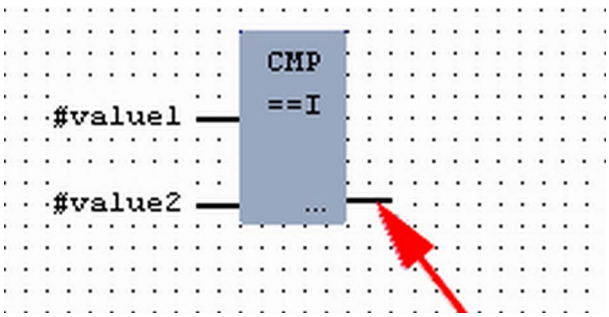
5. Type in "#" and press the *[Return]* key.

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*.

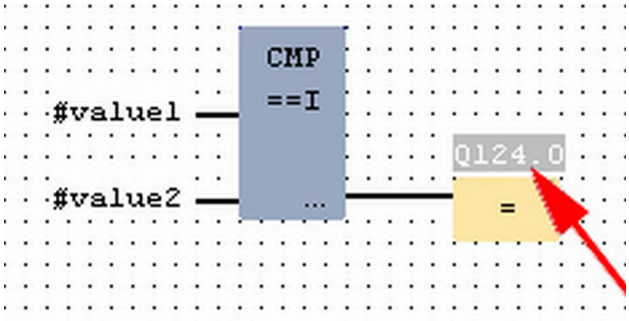
The allocation to the corresponding output, here Q 124.0, takes place with the following proceeding:

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.

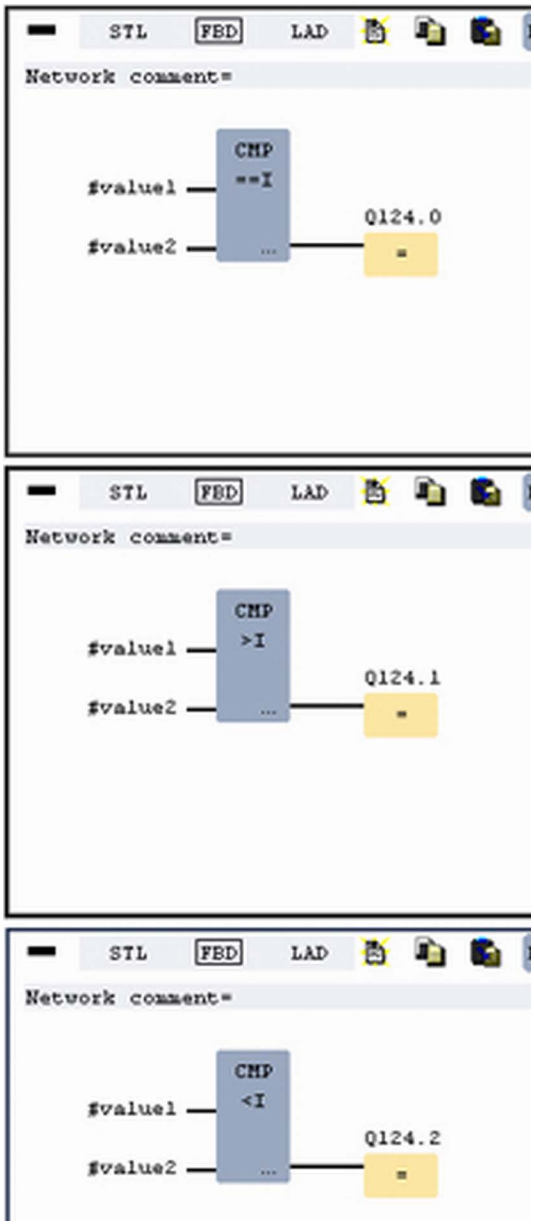


⇨ Network1 is finished, now.

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:

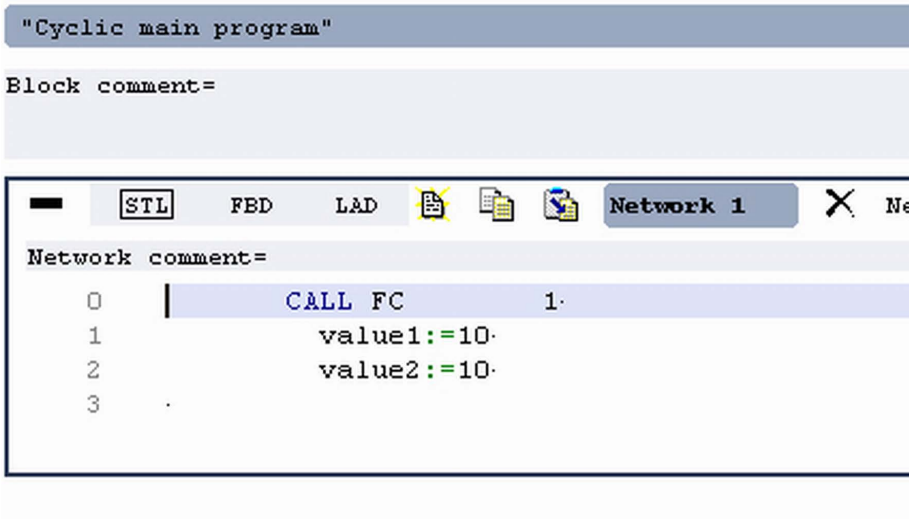
1. Move your mouse at an arbitrary position on the editor window and press the right mouse key.
2. Select at *'context menu ➔ Insert new network'*.
   ⇨ A dialog field is opened to enter the position and number of the networks.
3. Proceed as described for "Network 1".
4. Save the FC 1 with *'File ➔ Save 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:



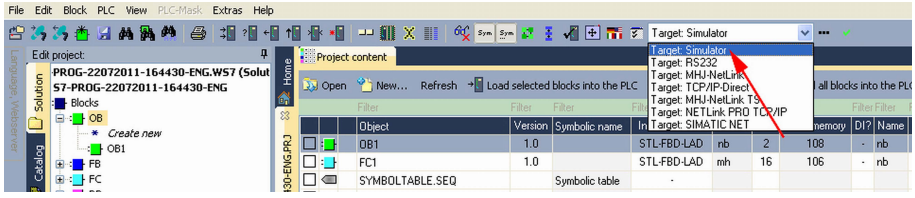
5. Save the OB 1 with respectively press *[Strg]+[S]*.

9.3.3 Test the PLC program in the *Simulator*

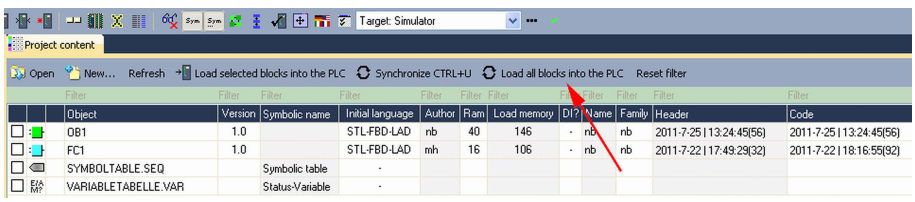
Proceeding

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.

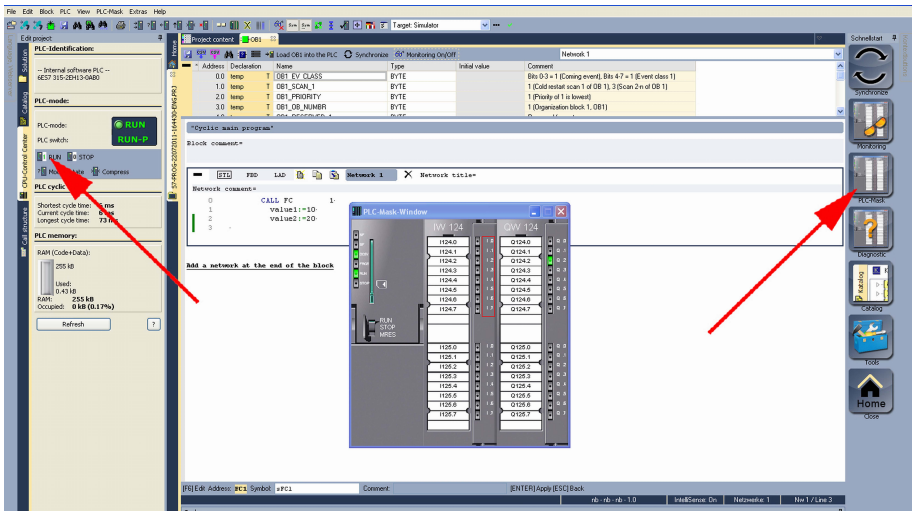
1. Open the *PLC mask* with *'view ➔ PLC mask'*.

   ⇨ 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.



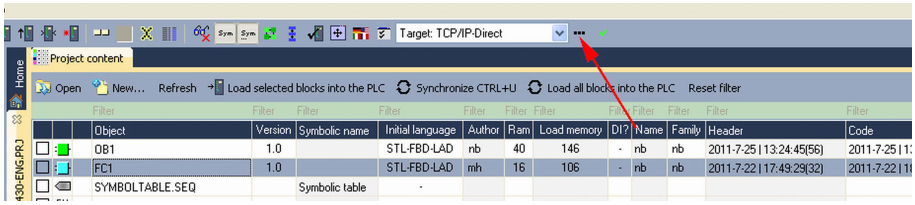
9.3.4 Transfer PLC program to CPU and its execution

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

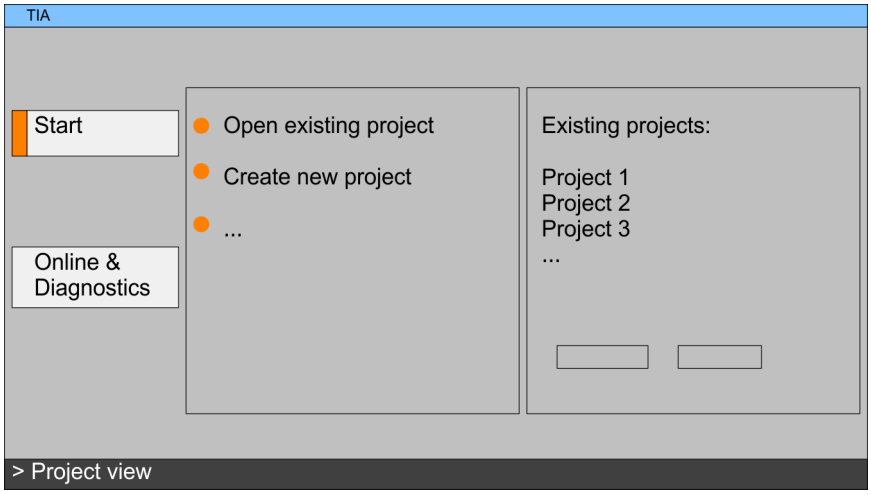
In this chapter the project engineering of the VIPA CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a VIPA CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means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

Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.

Starting the TIA Portal

To start the Siemens TIA Portal with Windows select *'Start ➔ Programs ➔ Siemens Automation ➔ TIA ...'*

Then the TIA Portal opens with the last settings used.



Exiting the TIA Portal

With the menu *'Project ➔ Exit'* in the *'Project view'* you may exit the TIA Portal. Here there is the possibility to save changes of your project before.

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

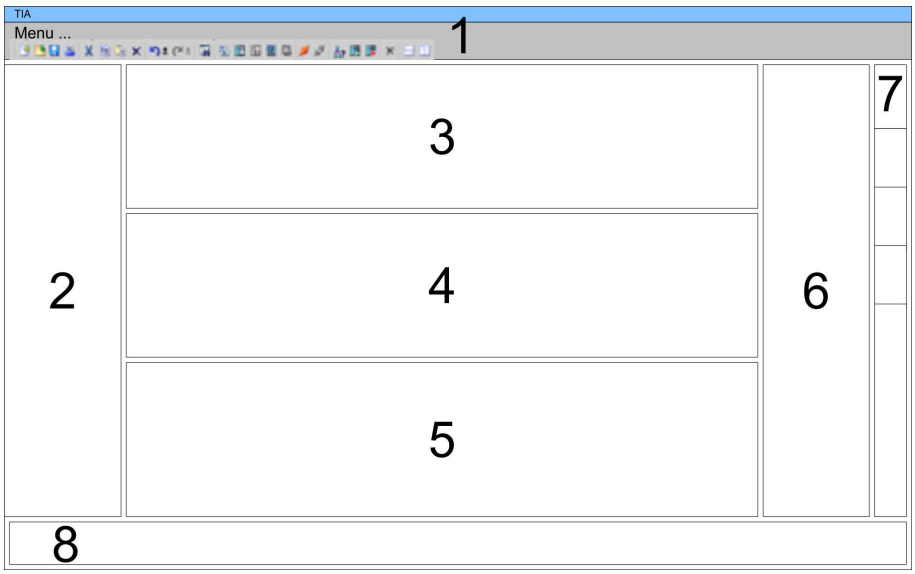
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

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

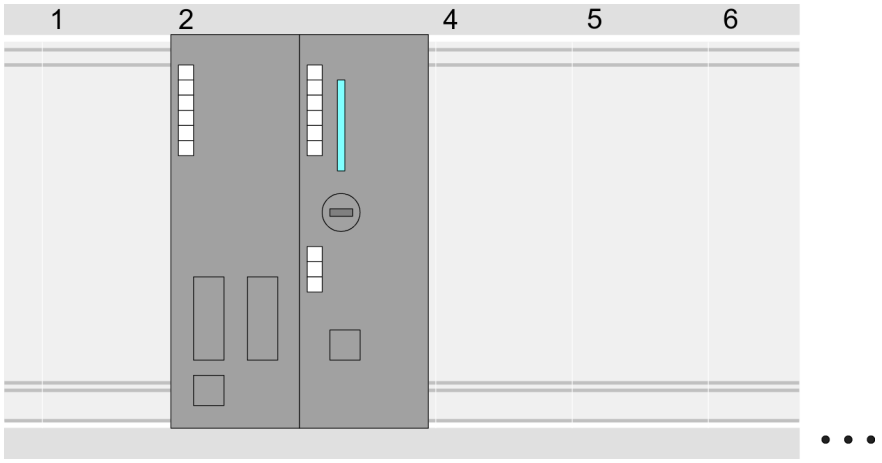
Configuration Siemens CPU

With the Siemens TIA Portal the CPU 314-6CF23 from VIPA is to be configured as CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3) from Siemens.

1. Start the Siemens TIA Portal.
2. Create a new project in the *Portal view* with *'Create new project'*.
3. Switch to the *Project view*.
4. Click in the *Project tree* at *'Add new device'*.
5. Select the following CPU in the input dialog:

   SIMATIC S7-300 > CPU 317-2DP (6ES7 317-2AK14-0AB0 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	
...		...		...	

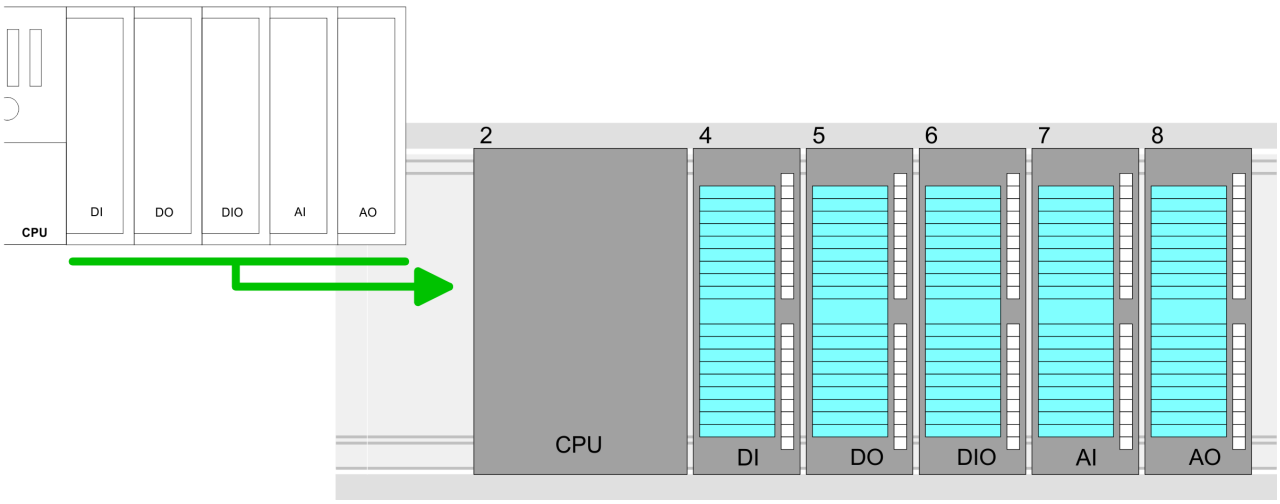
Setting standard CPU parameters

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. *Chapter 5.8.2 'Parameters CPU' on page 64*

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.

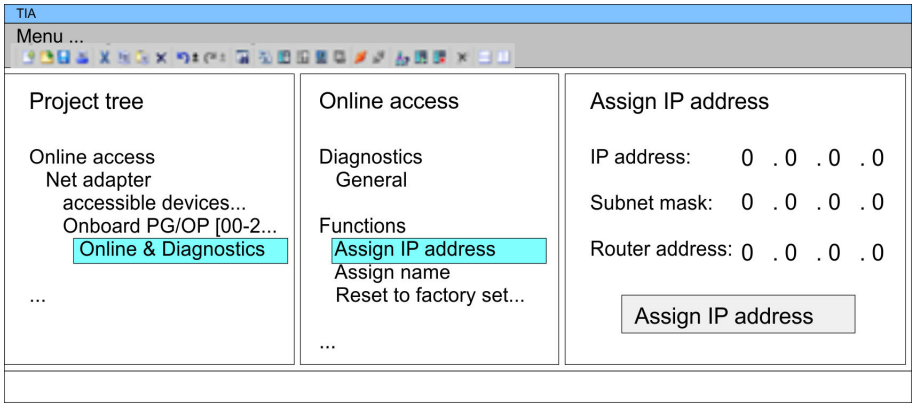
Assign IP address parameters

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.

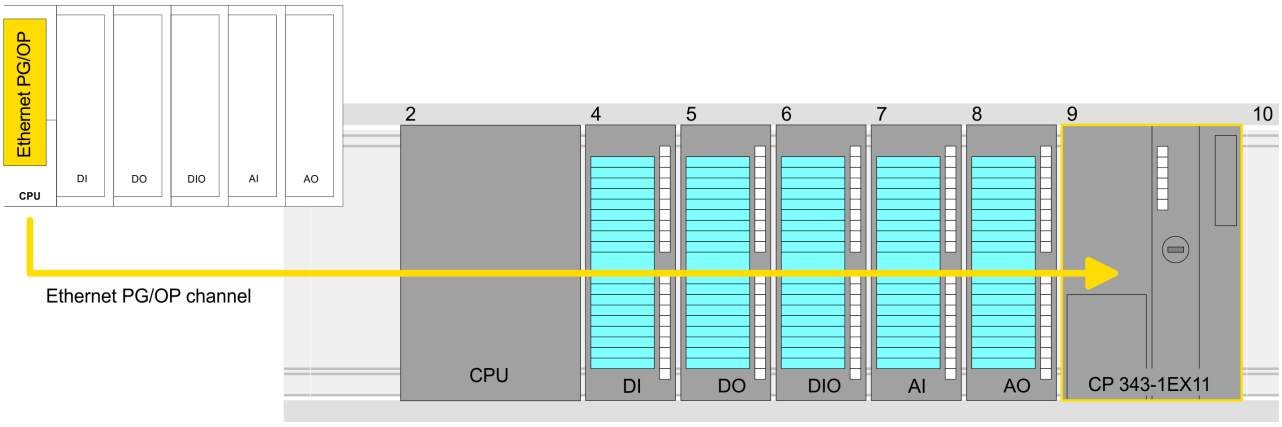
2. If not already done, configure in the *'Device configuration'* a Siemens CPU 317-2DP (6ES7 317-2AK14-0AB0 V3.3).

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.vipa.com at the "Service" part.

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

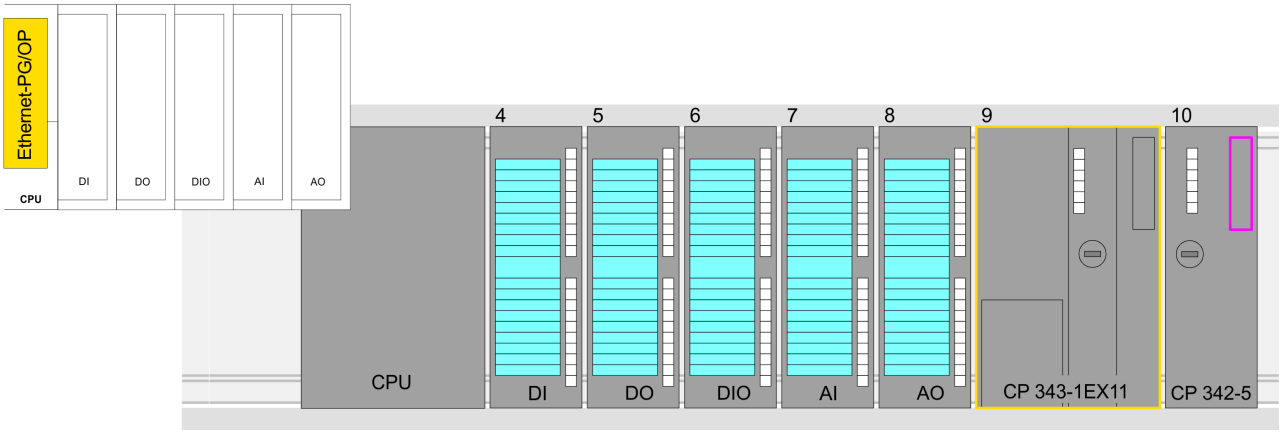
1. Browse to www.vipa.com
2. Click to *'Service ➔ Download ➔ GSD- and EDS-Files ➔ Profibus'*
3. Download the file Cx000023_Vxxx.
4. Extract the file to your work directory. The SPEEDBUS.GSD is stored in the directory VIPA_System_300S.
5. Start the hardware configurator from Siemens.
6. Close every project.
7. Select *'Options ➔ Install new GSD-file'*.
8. Navigate to the directory VIPA_System_300S and select **SPEEDBUS.GSD** an.
   ⇨ The SPEED7 CPUs and modules of the System 300S from VIPA 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 314-6CF23 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".



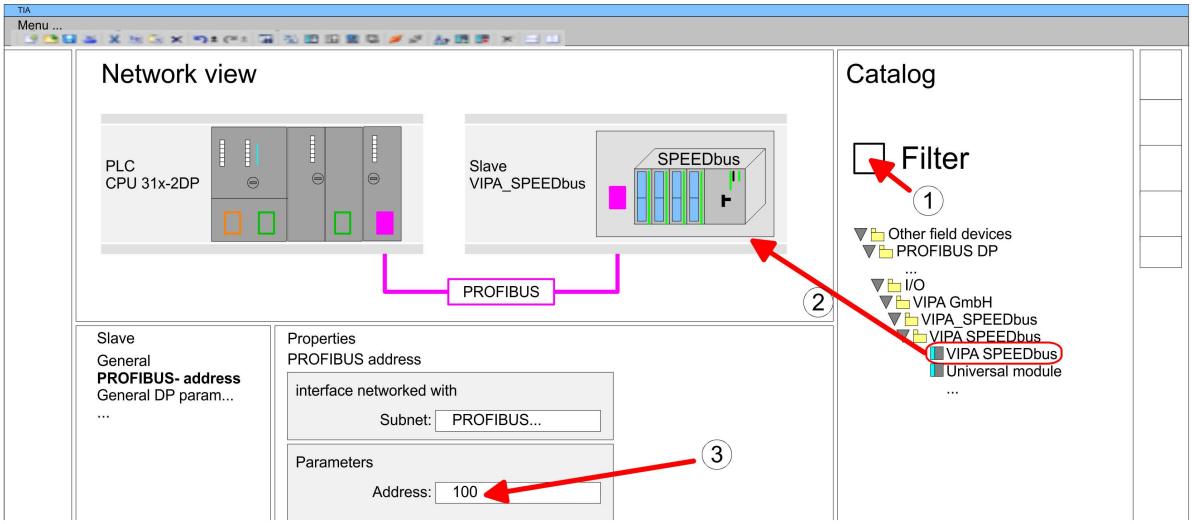
Device overview

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 314-6CF23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 314-6CF23 the properties dialog of the CPU is showed.

   Device overview

Module	...	Slot	...	Type	...
Slave ...		0		VIPA SPEEDbus	
314-6CF23		1		314-6CF23 ...	
		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 - VIPA-Include library

Overview

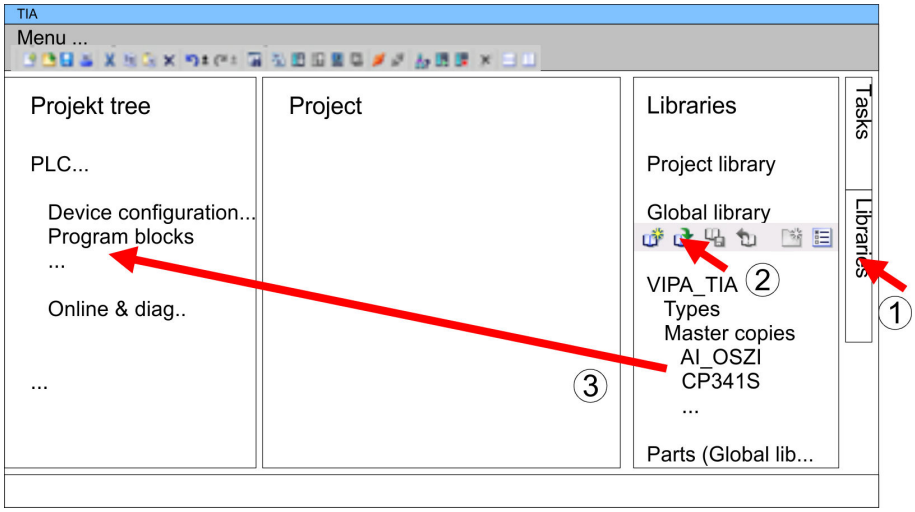
- The VIPA specific blocks can be found in the "Service" area of www.vipa.com as library download file at *Downloads > VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use VIPA specific blocks you have to import them into your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

Unzip ...TIA_Vxx.zip

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

Open library and transfer blocks into the project

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



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

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. ↳ *Chapter 5.19 'CMD - auto commands' on page 95*

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries, you select *Online & Diagnostics* in the Siemens TIA Portal. Here you can access the "Diagnostics buffer". ↳ *Chapter 5.20 'Diagnostic entries' on page 97*