YASKAWA

VIPA System 300S+

CPU-SC | 312-5BE23 | Manual

HB140 | CPU-SC | 312-5BE23 | en | 18-01

SPEED7 CPU 312SC

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

VIPA CONTROLS

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

312-5BE23_000_CPU-SC 312SC,3,EN - © 2018

Table of contents

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.com

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

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

CE Conformity Declaration

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

Conformity Information

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

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

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

Technical support

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

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

1.2 About this manual

Objective and contents

This manual describes the SPEED7 CPU-SC 312-5BE23 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
CPU 312SC	312-5BE23	01	V3.7.3

Target audience

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

Structure of the manual

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

Guide to the document

The following guides are available in the manual:

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

Availability

The manual is available in:

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

Icons Headings

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

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

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

Supplementary information and useful tips.

1.3 Safety information

Applications conforming with specifications

The system is constructed and produced for:

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

DANGER!
This device is not certified for applications in

– in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

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

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

Disposal

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

2 Basics

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 312-5BE23

Overview

The CPU 312-5BE23 bases upon the SPEED7 technology. This supports the CPU at programming and communication by means of co-processors that causes a power improvement for highest needs.

- The CPU is programmed in STEP®7 from Siemens. For this you may use the SIMATIC Manager or TIA Portal from Siemens. Here the instruction set of the S7-400 from Siemens is used.
- Modules and CPUs of the System 300S from 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 storage module.
- The CPU is configured as CPU 312C (6ES7 312-5BE03-0AB0/V2.6) 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 1MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 128kbyte
  - There is the possibility to extend the work memory to its maximum printed capacity 1MB by means of a memory extension card.

Integrated PtP function

The CPU has a RS485 interface, which is fix set to PtP communication (**p**oint **t**o **p**oint). This supports the serial process connection to different source or destination systems.

Integrated Ethernet PG/OP channel

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

Operation Security

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

Dimensions/ Weight

Dimensions of the basic enclosure:

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

Integrated power supply

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

2.4 General data

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

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

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

General data

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

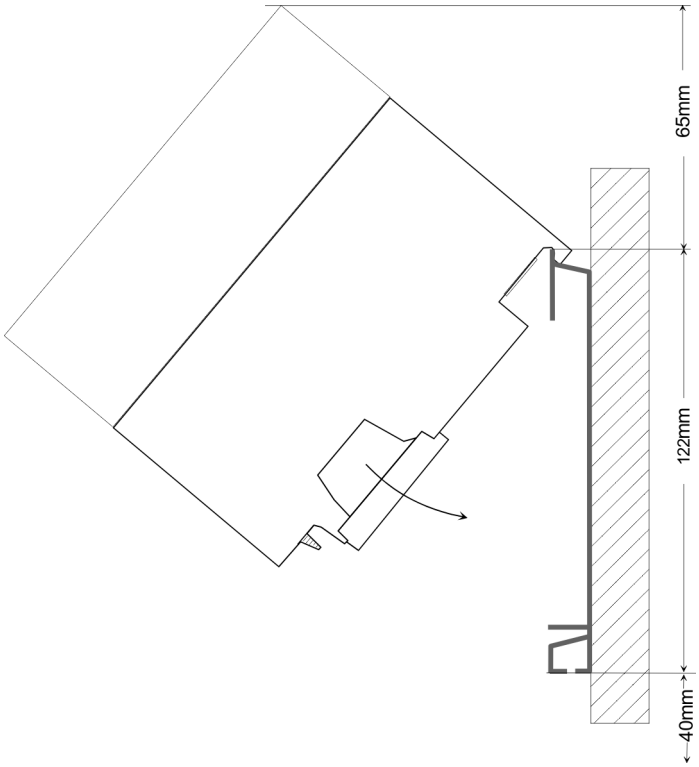
EMC	Standard		Comment
Emitted interference	EN 61000-6-4		Class A (Industrial area)
Noise immunity zone B	EN 61000-6-2		Industrial area
		EN 61000-4-2	ESD 8kV at air discharge (degree of severity 3), 4kV at contact discharge (degree of severity 2)
		EN 61000-4-3	HF field immunity (casing) 80MHz … 1000MHz,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

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

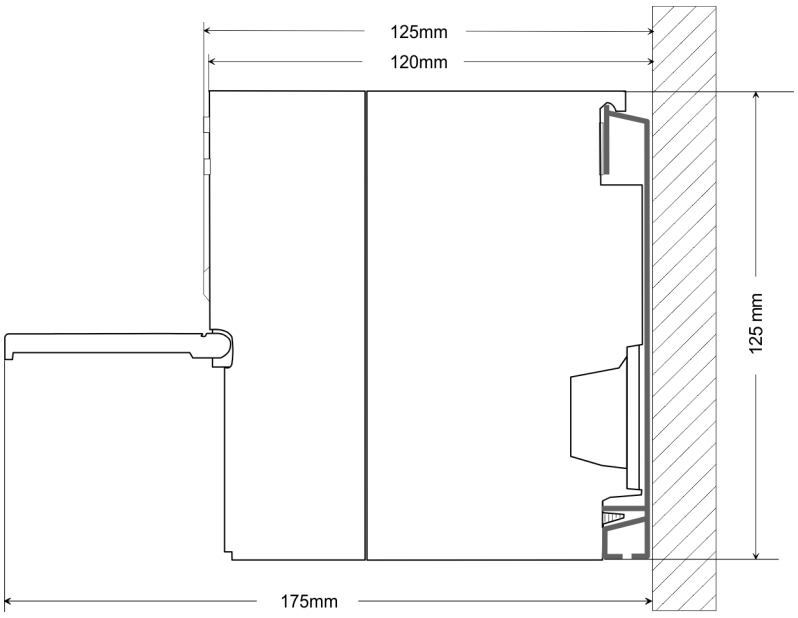
3 Assembly and installation guidelines

3.1 Installation dimensions

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



Installation dimensions



3.2 Assembly standard bus

General

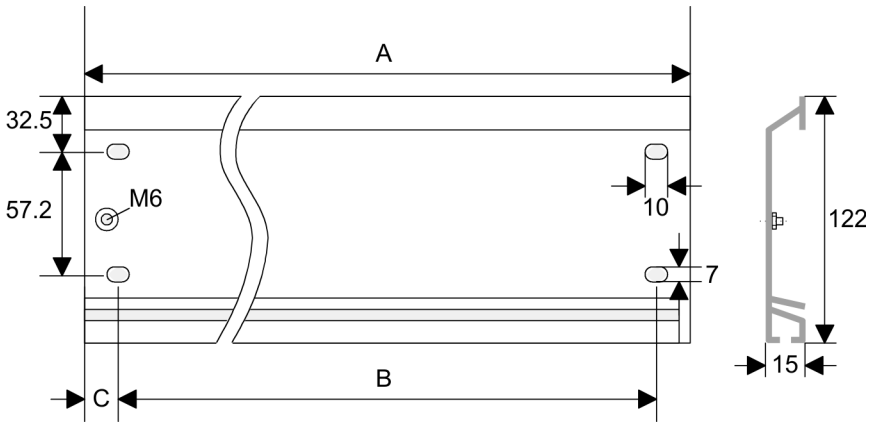
The single modules are directly installed on a profile rail and connected via the backplane bus connector. Before installing the modules you have to clip the backplane bus connector to the module from the backside. The backplane bus connector is delivered together with the peripheral modules.

Profile rail

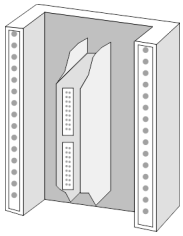
Order number	A	B	C
390-1AB60	160	140	10
390-1AE80	482	466	8.3
390-1AF30	530	500	15
390-1AJ30	830	800	15
390-9BC00*	2000	Drillings only left	15

*) Unit pack: 10 pieces

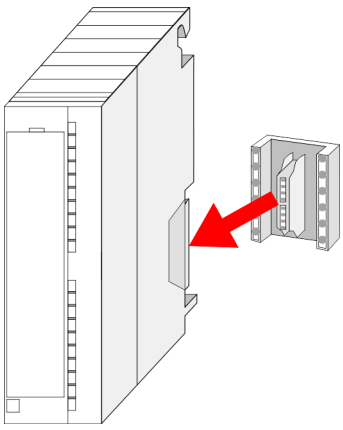
Measures in mm



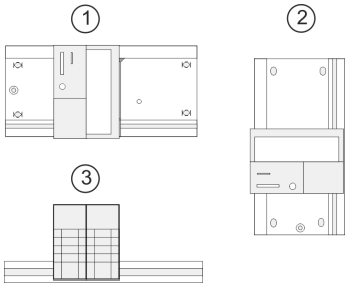
Bus connector



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



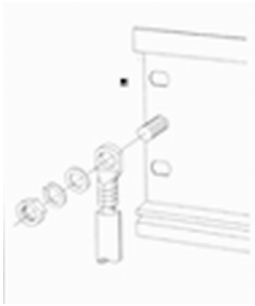
Assembly possibilities



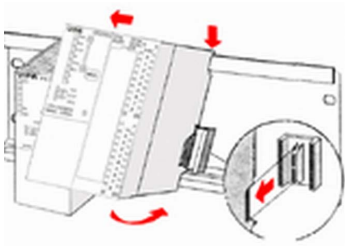
Please regard the allowed environment temperatures:

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

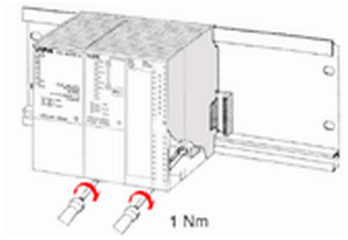
Approach



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



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



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

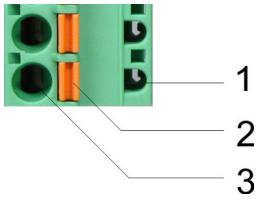
3.3 Cabling

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

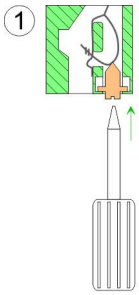
CageClamp technology (green)

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

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

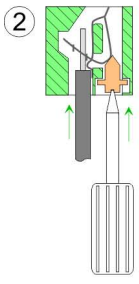


1 Test point for 2mm test tip
2 Locking (orange) for screwdriver
3 Round opening for wires

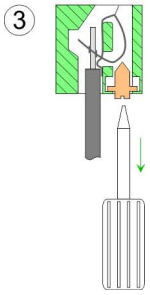


The picture on the left side shows the cabling step by step from top view.

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



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



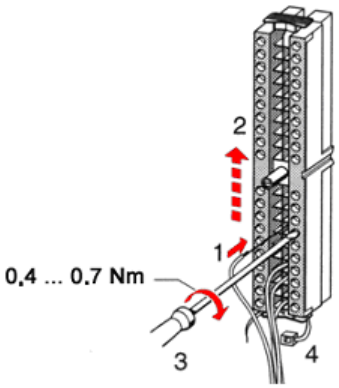
3. By removing the screwdriver the wire is connected safely with the plug connector via a spring.

Front connector I/O periphery

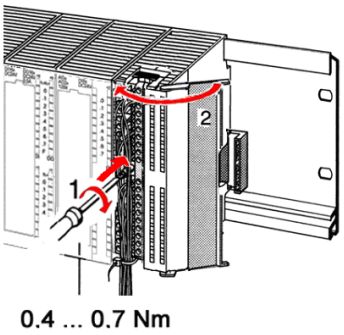
The 40pole front connector with the order number 392-1AM00 is in the delivery.

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.

3.4 Installation guidelines

General

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

What does EMC mean?

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

The components of 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.

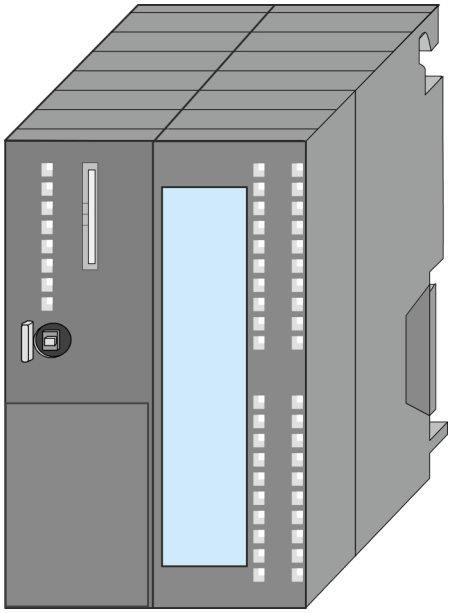
Remedy: Potential compensation line

4 Hardware description

4.1 Properties

CPU 312-5BE23

- SPEED7 technology integrated
- 128kbyte work memory integrated (64kbyte code, 64kbyte data)
- Work memory expandable to max. 1MB (512kbyte code, 512kbyte data)
- 1MB load memory
- RS485 interface for 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
- Digital I/Os: DI 16xDC24V / DO 8xDC 24V, 0.5A
- 2 channels for counter, frequency measurement and pulse width modulation
- I/O address range digital/analog 1024byte
- 512 timer
- 512 counter
- 8192 flag byte



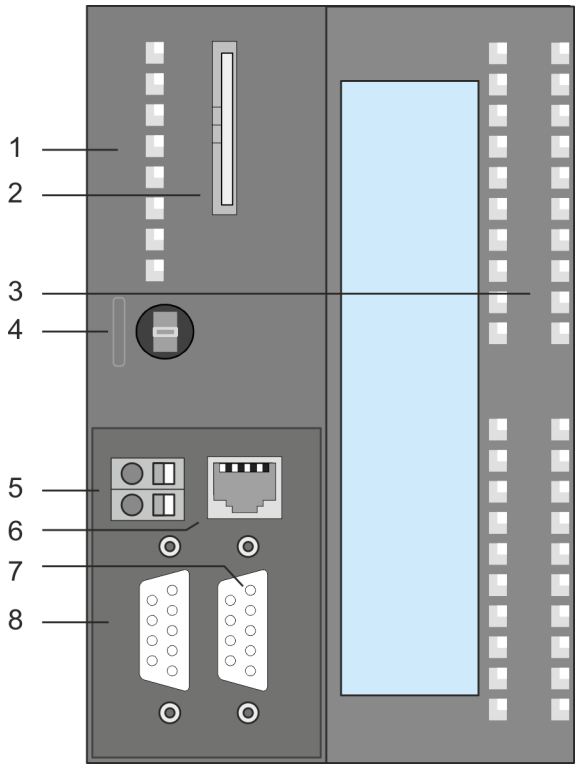
Ordering data

Type	Order number	Description
312SC	312-5BE23	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, DI 16xDC24V / DO 8xDC24V, 0.5A, 2 channels technological functions

4.2 Structure

4.2.1 General

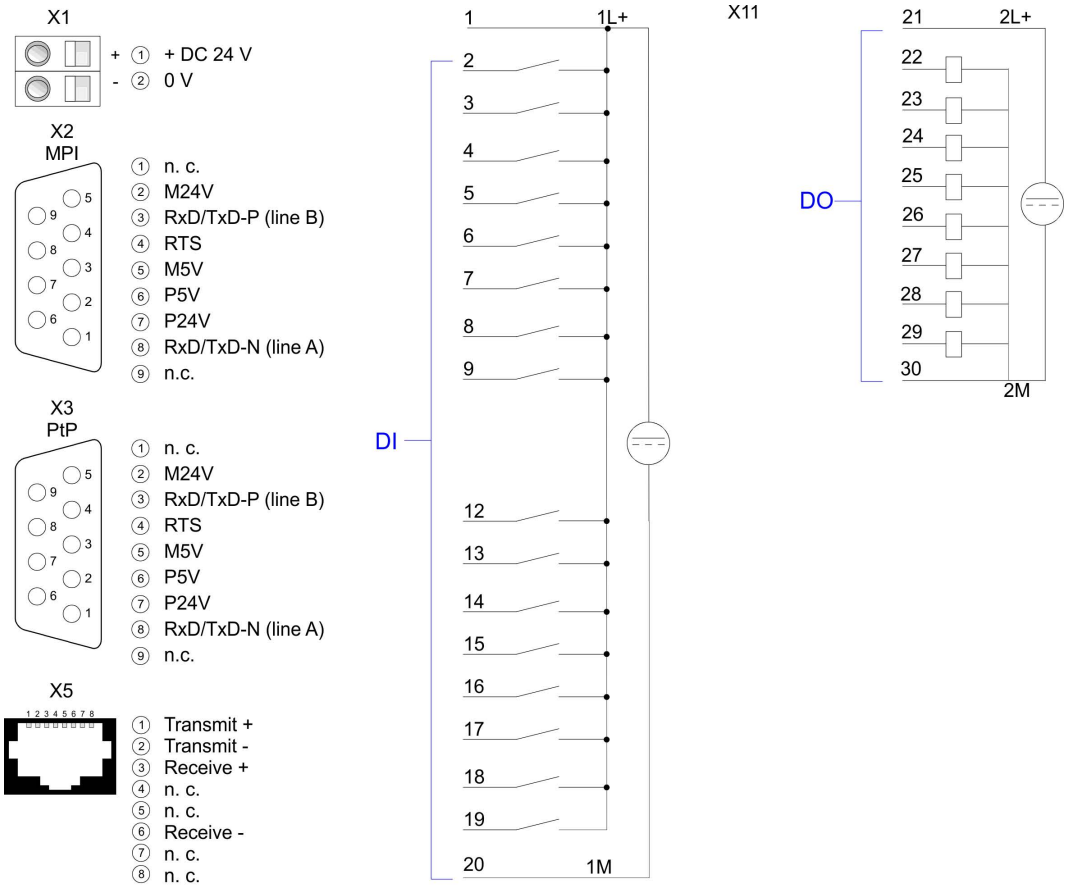
CPU 312-5BE23



1 LEDs of the CPU part
2 Storage media slot (lockable)
3 LEDs of the I/O part
4 Operating mode switch CPU
5 Slot for DC 24V power supply
6 Twisted pair interface for Ethernet PG/OP channel
7 PtP interface
8 MPI interface

The components 5 - 8 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.

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.

X3: PtP interface

9pin SubD jack:

The CPU has a PtP interface. The interface is fix set to PtP communication.

- 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 PtP communication is configured during run-time by means of the SFC 216 (SER_CFG). The communication happens by means of the SFC 217 (SER_SND) and SFC 218 (SER_RCV).

X5: Ethernet PG/OP channel

8pin RJ45 jack:

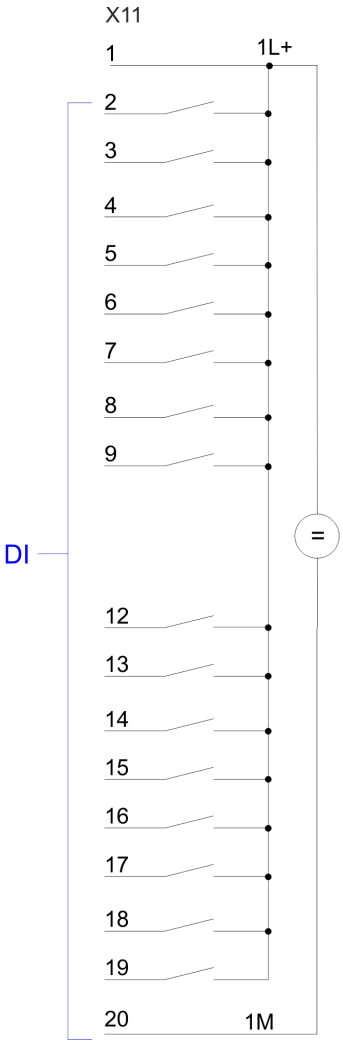
- The RJ45 jack serves the interface to the Ethernet PG/OP channel.
- This interface allows you to program res. remote control your CPU, to access the internal web site or to connect a visualization.
- Configurable connections are not possible.
- For online access to the CPU via Ethernet PG/OP channel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

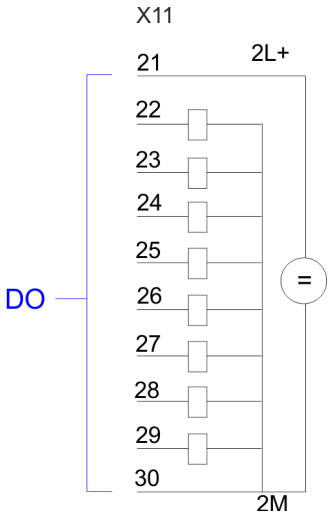
4.2.3 In-/Output area CPU 312-5BE23

Overview

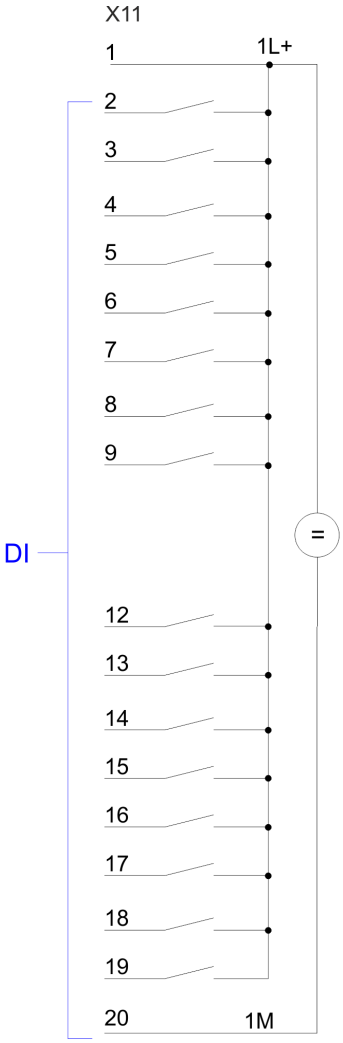
The CPU 312-5BE23 has the following digital in and output channels integrated in one casing:

- Digital Input: 16xDC 24V, with interrupt capability
- Digital Output: 8xDC 24V, 0.5A
- Technological functions: 2 channels



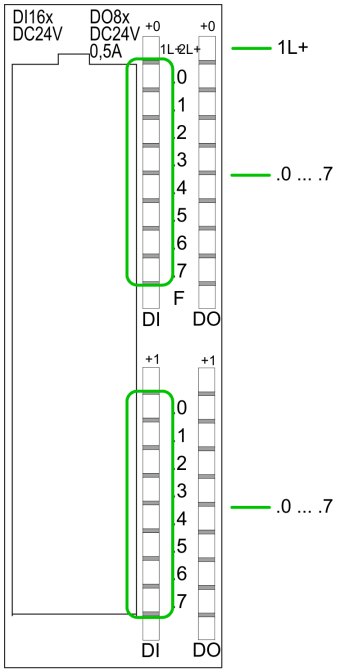


CAUTION!
Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+.



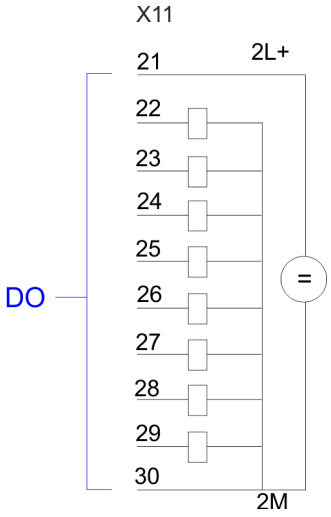
Pin assignment X11: DI

Pin	Assignment
1	1L+ Power supply +DC 24V
2	I+0.0 / Channel 0 (A) / Pulse
3	I+0.1 / Channel 0 (B) / Direction
4	I+0.2 / Channel 0 HW gate
5	I+0.3 / Channel 1 (A) / Pulse
6	I+0.4 / Channel 1 (B) / Direction
7	I+0.5 / Channel 1 HW gate
8	I+0.6
9	I+0.7
10	not used
11	not used
12	I+1.0
13	I+1.1
14	I+1.2
15	I+1.3
16	I+1.4 / Channel 0 Latch
17	I+1.5 / Channel 1 Latch
18	I+1.6
19	I+1.7
20	Ground 1M DI



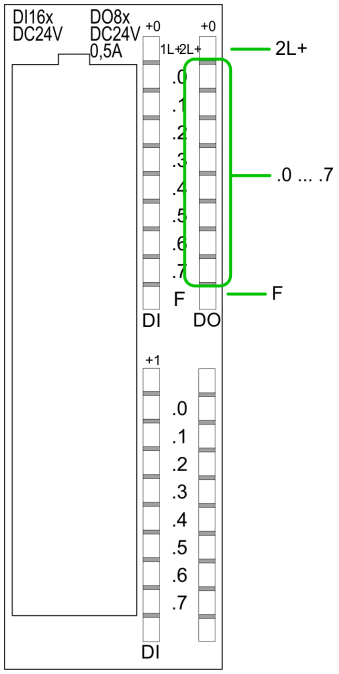
Status indication X11: DI

- 1L+
  - LED (green)
    Supply voltage available for DI
- .0 7
  - LEDs (green)
    I+0.0 ... I+0.7
    I+1.0 ... I+1.7
    Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated



Pin assignment X11: DO

Pin	Assignment
21	2L+ Power supply +DC 24V
22	O+0.0 / Channel 0 Output
23	O+0.1 / Channel 1 Output
24	Q+0.2
25	Q+0.3
26	Q+0.4
27	Q+0.5
28	Q+0.6
29	Q+0.7
30	Ground 2M DO
31 ... 40	not used



Status indication X11: DO

- 2L+
  - LED (green)
    Supply voltage available for DO
- .0 7
  - LEDs (green)
    Q+0.0 ... Q+0.7
    The according LED is on at active output
- F
  - LED (red)
    Overload or short circuit error

4.2.4 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 1MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 128kbyte
  - There is the possibility to extend the work memory to its maximum printed capacity 1MB by means of a memory extension card.

4.2.5 Slot for storage media

At this slot the following storage media can be plugged:

- SD respectively MCC (**M**ulti**m**edia **c**ard)
  - External memory card for programs and firmware.
- MCC - **M**emory **c**onfiguration **c**ard
  - External memory card (MMC) for programs and firmware with the possibility to unlock additional work memory.
  - The additional memory can be purchased separately. ↬ *Chapter 5.15 'Deployment storage media - MMC, MCC' on page 70*
  - To activate the corresponding card is to be installed and an *Overall reset* is to be established. ↬ *Chapter 5.12 'Overall reset' on page 65*

4.2.6 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.

4.2.7 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.

4.2.8 LEDs

LEDs CPU

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

Ethernet PG/OP channel

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

4.3 Technical data

Please consider with the configuration with the Siemens TIA Portal the number of timer and counters is limited to the maximum possible number of the corresponding Siemens CPU.

Order no.	312-5BE23
Type	VIPA CPU 312SC
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)	135 mA
Current consumption (rated value)	500 mA
Inrush current	11 A
I^2t	0.7 A^2s
Max. current drain at backplane bus	3 A
Max. current drain load supply	-
Power loss	8 W
Technical data digital inputs	
Number of inputs	16
Cable length, shielded	1000 m
Cable length, unshielded	600 m

Order no.	312-5BE23
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"	0.1 / 0.35 ms
Input delay of "1" to "0"	0.1 / 0.35 ms
Number of simultaneously utilizable inputs horizontal configuration	16
Number of simultaneously utilizable inputs vertical configuration	16
Input characteristic curve	IEC 61131-2, type 1
Initial data size	2 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)	100 mA
Total current per group, horizontal configuration, 40°C	3 A
Total current per group, horizontal configuration, 60°C	2 A
Total current per group, vertical configuration	2 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)	0.5 mA
Output delay of "0" to "1"	100 µs

Order no.	312-5BE23
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	1 Byte
Technical data analog inputs	
Number of inputs	-
Cable length, shielded	-
Rated load voltage	-
Reverse polarity protection of rated load voltage	-
Current consumption from load voltage L+ (without load)	-
Voltage inputs	-
Min. input resistance (voltage range)	-
Input voltage ranges	-
Operational limit of voltage ranges	-
Operational limit of voltage ranges with SFU	-
Basic error limit voltage ranges	-
Basic error limit voltage ranges with SFU	-
Destruction limit voltage	-
Current inputs	-
Max. input resistance (current range)	-
Input current ranges	-
Operational limit of current ranges	-
Operational limit of current ranges with SFU	-
Basic error limit current ranges	-
Radical error limit current ranges with SFU	-

Order no.	312-5BE23
Destruction limit current inputs (electrical current)	-
Destruction limit current inputs (voltage)	-
Resistance inputs	-
Resistance ranges	-
Operational limit of resistor ranges	-
Operational limit of resistor ranges with SFU	-
Basic error limit	-
Basic error limit with SFU	-
Destruction limit resistance inputs	-
Resistance thermometer inputs	-
Resistance thermometer ranges	-
Operational limit of resistance thermometer ranges	-
Operational limit of resistance thermometer ranges with SFU	-
Basic error limit thermoresistor ranges	-
Basic error limit thermoresistor ranges with SFU	-
Destruction limit resistance thermometer inputs	-
Thermocouple inputs	-
Thermocouple ranges	-
Operational limit of thermocouple ranges	-
Operational limit of thermocouple ranges with SFU	-
Basic error limit thermoelement ranges	-
Basic error limit thermoelement ranges with SFU	-
Destruction limit thermocouple inputs	-
Programmable temperature compensation	-
External temperature compensation	-
Internal temperature compensation	-
Technical unit of temperature measurement	-
Resolution in bit	-
Measurement principle	-
Basic conversion time	-
Noise suppression for frequency	-
Initial data size	-
Technical data analog outputs	
Number of outputs	-
Cable length, shielded	-

Order no.	312-5BE23
Rated load voltage	-
Reverse polarity protection of rated load voltage	-
Current consumption from load voltage L+ (without load)	-
Voltage output short-circuit protection	-
Voltage outputs	-
Min. load resistance (voltage range)	-
Max. capacitive load (current range)	-
Max. inductive load (current range)	-
Output voltage ranges	-
Operational limit of voltage ranges	-
Basic error limit voltage ranges with SFU	-
Destruction limit against external applied voltage	-
Current outputs	-
Max. in load resistance (current range)	-
Max. inductive load (current range)	-
Typ. open circuit voltage current output	-
Output current ranges	-
Operational limit of current ranges	-
Radical error limit current ranges with SFU	-
Destruction limit against external applied voltage	-
Settling time for ohmic load	-
Settling time for capacitive load	-
Settling time for inductive load	-
Resolution in bit	-
Conversion time	-
Substitute value can be applied	-
Output data size	-
Technical data counters	
Number of counters	2
Counter width	32 Bit
Maximum input frequency	10 kHz
Maximum count frequency	10 kHz
Mode incremental encoder	✓
Mode pulse / direction	✓
Mode pulse	✓
Mode frequency counter	✓

Order no.	312-5BE23
Mode period measurement	✓
Gate input available	✓
Latch input available	✓
Reset input available	-
Counter output available	✓
Load and working memory	
Load memory, integrated	1024 KB
Load memory, maximum	1024 KB
Work memory, integrated	128 KB
Work memory, maximal	1024 KB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
Hardware configuration	
Racks, max.	1
Modules per rack, max.	8
Number of integrated DP master	0
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
Diagnostic interrupt	yes
Diagnostic functions	no
Diagnostics information read-out	possible
Supply voltage display	green LED
Group error display	red SF LED
Channel error display	red LED per group
Isolation	
Between channels	✓
Between channels of groups to	16
Between channels and backplane bus	✓
Between channels and power supply	-
Max. potential difference between circuits	DC 75 V/ AC 50 V

Order no.	312-5BE23
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.	0.02 µs
Word instruction, min.	0.02 µs
Double integer arithmetic, min.	0.02 µs
Floating-point arithmetic, min.	0.12 µs
Timers/Counters and their retentive characteristics	
Number of S7 counters	512
S7 counter remanence	adjustable 0 up to 128
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	adjustable 0 up to 128
S7 times remanence adjustable	not retentive
Data range and retentive characteristic	
Number of flags	8192 Byte
Bit memories retentive characteristic adjustable	adjustable 0 up to 128
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	510 Byte
Blocks	
Number of OBs	15
Number of FBs	2048
Number of FCs	2048
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

Order no.	312-5BE23
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	1024 Byte
Output I/O address area	1024 Byte
Input process image maximal	128 Byte
Output process image maximal	128 Byte
Digital inputs	272
Digital outputs	264
Digital inputs central	272
Digital outputs central	264
Integrated digital inputs	16
Integrated digital outputs	8
Analog inputs	64
Analog outputs	64
Analog inputs, central	64
Analog outputs, central	64
Integrated analog inputs	0
Integrated analog outputs	0
Communication functions	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	4
Size of GD packets, max.	22 Byte
S7 basic communication	✓
S7 basic communication, user data per job	76 Byte
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
S7 communication, user data per job	160 Byte
Number of connections, max.	32
PWM data	
PWM channels	2
PWM time basis	0.1 ms / 1 ms

Order no.	312-5BE23
Period length	4...65535 / 1...65535 * time base
Minimum pulse width	0...0.5 * Period duration
Type of output	Highside with 1.1kOhm pulldown
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, non-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	-
DP slave	-
Point-to-point interface	✓
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Functionality MPI	
Number of connections, max.	32
PG/OP channel	✓
Routing	-
Global data communication	✓
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-

Order no.	312-5BE23
Transmission speed, min.	19.2 kbit/s
Transmission speed, max.	187.5 kbit/s
Functionality PROFIBUS master	
Number of connections, max.	-
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.	-
Transmission speed, max.	-
Number of DP slaves, max.	-
Address range inputs, max.	-
Address range outputs, max.	-
User data inputs per slave, max.	-
User data outputs per slave, max.	-
Functionality PROFIBUS slave	
Number of connections, max.	-
PG/OP channel	-
Routing	-
S7 communication	-
S7 communication as server	-
S7 communication as client	-
Direct data exchange (slave-to-slave communication)	-
DPV1	-
Transmission speed, min.	-
Transmission speed, max.	-
Automatic detection of transmission speed	-
Transfer memory inputs, max.	-
Transfer memory outputs, max.	-
Address areas, max.	-
User data per address area, max.	-

Order no.	312-5BE23
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	-
RS485 interface	✓
Connector	Sub-D, 9-pin, female
Transmission speed, min.	150 bit/s
Transmission speed, max.	115.5 kbit/s
Cable length, max.	500 m
Point-to-point protocol	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	✓
Modbus master protocol	✓
Modbus slave protocol	-
Special protocols	-
Functionality RJ45 interfaces	
Type	X5
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
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
Net weight	410 g
Weight including accessories	-
Gross weight	-
Environmental conditions	

Order no.	312-5BE23
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	yes
KC certification	yes

5 Deployment CPU 312-5BE23

5.1 Assembly

Information about assembly and cabling: Chapter 3 'Assembly and installation guidelines' on page 15

5.2 Start-up behavior

Turn on power supply

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

Default boot procedure, as delivered

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

Boot procedure with valid configuration in the CPU

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

Boot procedure with empty battery

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

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

5.3 Addressing

5.3.1 Overview

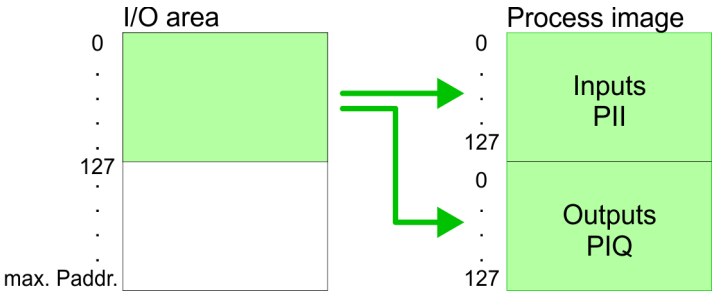
To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. At the start-up of the CPU, this assigns automatically peripheral addresses for digital in-/output modules starting with 0 and ascending depending on the slot location. If no hardware project engineering is available, the CPU stores at the addressing analog modules to even addresses starting with 256.

5.3.2 Addressing Backplane bus I/O devices

The CPU 312-5BE23 provides an I/O area (address 0 ... max. Peripheral address) and a process image of the In- and Outputs (each address 0 ... 127). The process image stores the signal states of the lower address (0 ... 127) 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 may be addressed by the CPU 312-5BE23. The extension by means of extension racks is not possible.

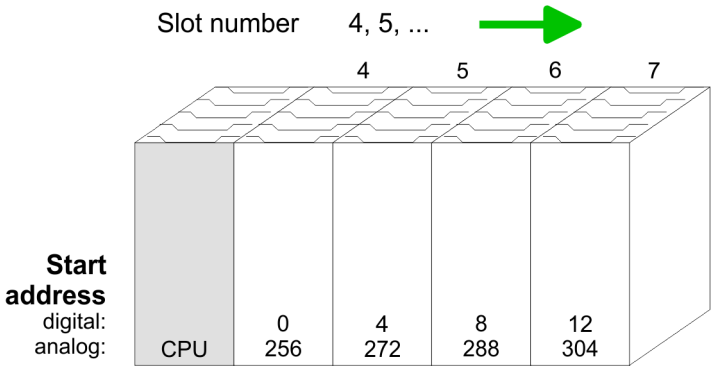
Define addresses by hardware configuration

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

Automatic addressing

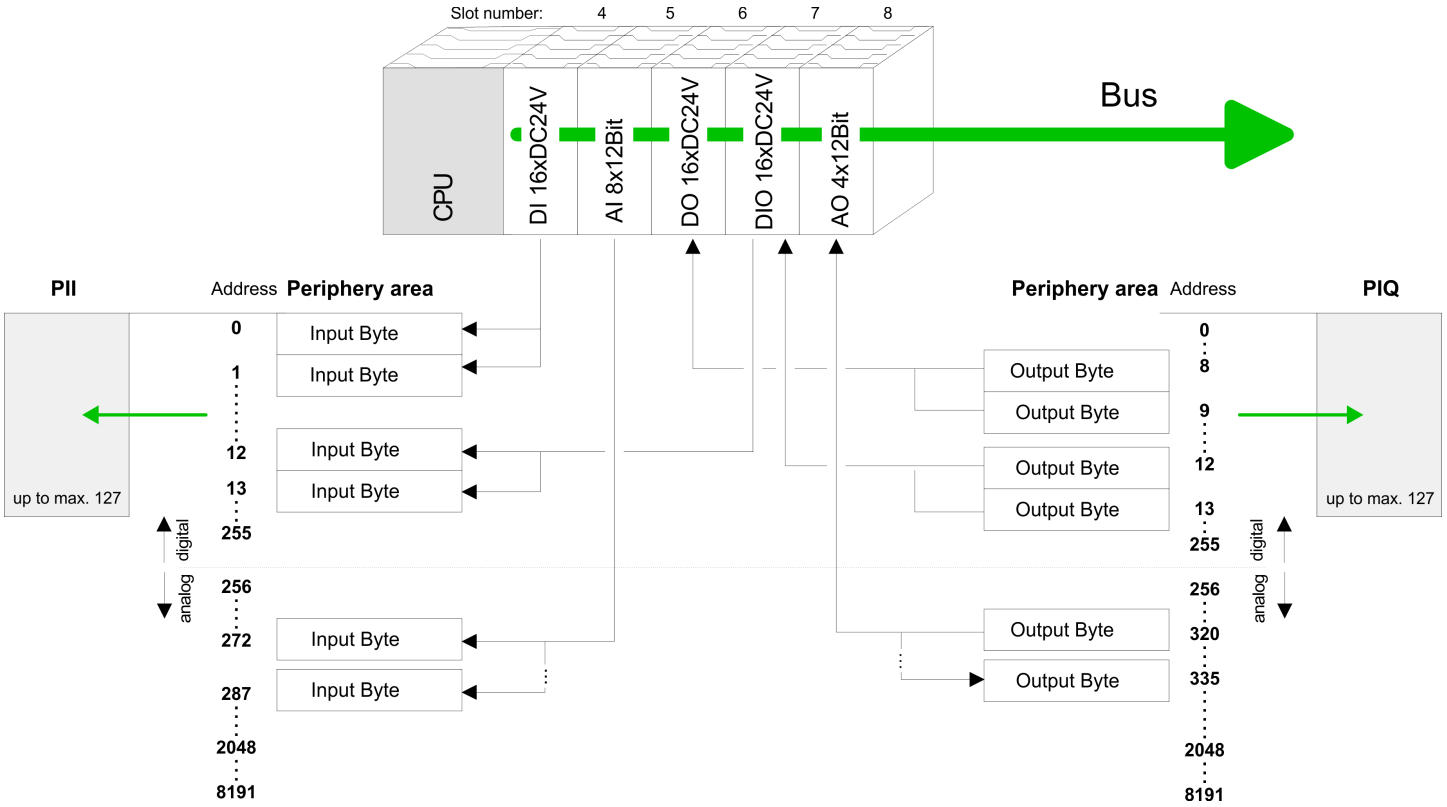
If you do not like to use a hardware configuration, an automatic addressing comes into force. At the automatic address allocation DIOs occupy depending on the slot location always 4byte and AIOs, FMs, CPs always 16byte at the bus. Depending on the slot location the start address from where on the according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

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



Example for automatic address allocation

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



5.4 Address assignment

Input area

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital input I+0.0 ... I+0.7
	125	Byte	Digital input I+1.0 ... I+1.7
Counter	768	DInt	Channel 0: Counter value / Frequency value
	772	DInt	Channel 1: Counter value / Frequency value
	776	DInt	reserved
	780	DInt	reserved

Output area

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital output Q+0.0 ... Q+0.7
Counter	768	DWord	reserved
	772	DWord	reserved
	776	DWord	reserved
	780	DWord	reserved

5.5 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 312C**
2.2	*DI10/DO6*
2.4	*Count*
3	

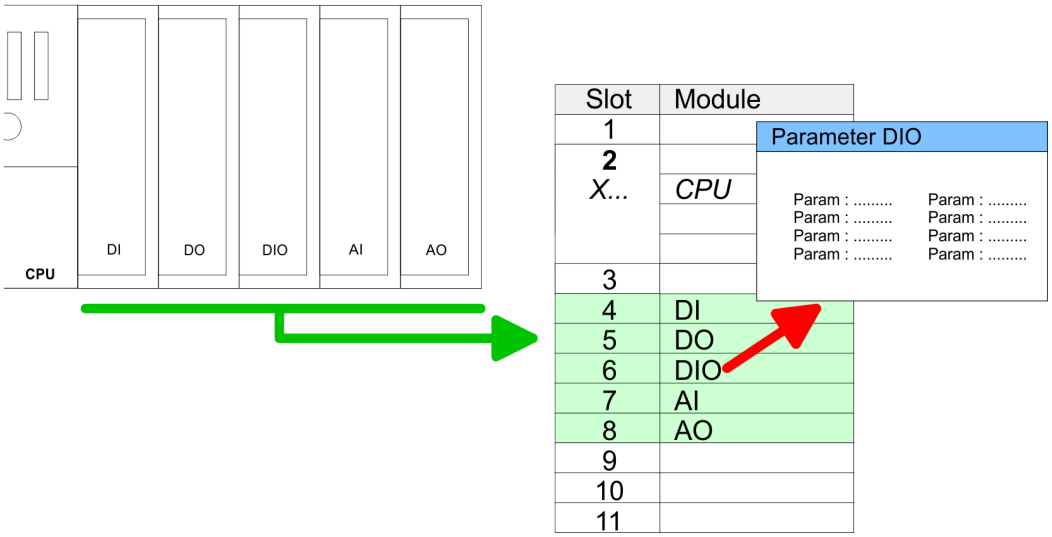
To be compatible with the Siemens SIMATIC Manager the following steps should be executed:

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 Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6).

5.6 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.

Maximum 8 modules addressable

Maximally 8 modules may be addressed by the CPU 312-5BE23. The extension by means of extension racks is not possible!

5.7 Hardware configuration - Ethernet PG/OP channel

Overview

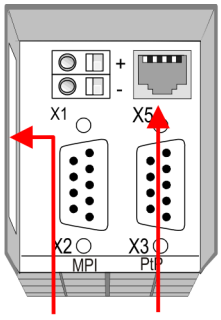
The CPU 312-5BE23 has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU. The PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc. With the first 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.

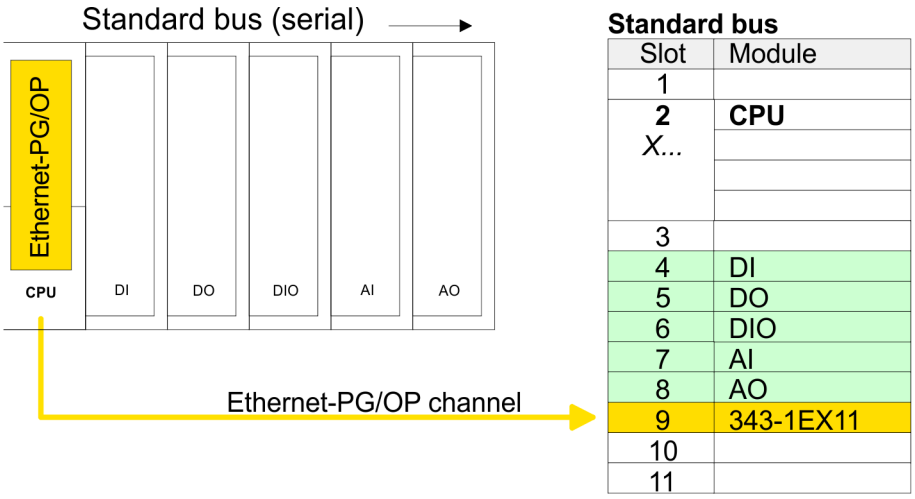
Assign IP address parameters

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 V 5.3 & SP3 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 1. label beneath the front flap of the CPU.
4. Choose if necessary the known MAC address of the list of found stations.
5. Either type in the IP configuration like IP address, subnet mask and gateway.
6. Confirm with [Assign IP configuration].
   - ⇨ Direct after the assignment the Ethernet PG/OP channel may be reached online by these address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.

Take IP address parameters in project

1. Open the Siemens hardware configurator und configure the Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6).
2. Configure the modules at the standard bus.
3. For the Ethernet PG/OP channel you have to configure a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX11 0XE0) always below the really plugged modules.
4. Open the property window via double-click on the CP 343-1EX11 and enter for the CP at *'Properties'* the IP address data, which you have assigned before.
5. Assign the CP to a *'Subnet'*. Without assignment the IP address data are not used!
6. Transfer your project.

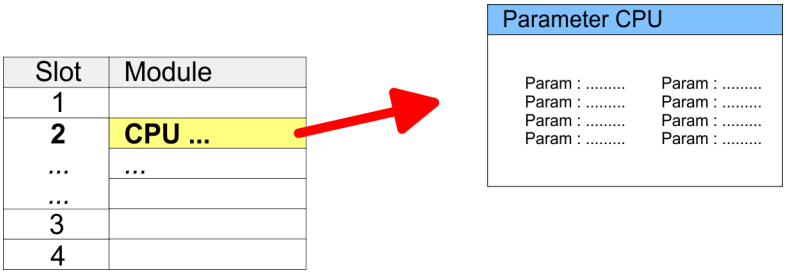


5.8 CPU parametrization

5.8.1 Parametrization via Siemens CPU

Parameterization via Siemens CPU 312C

Since the CPU is to be configured as Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6) in the Siemens hardware configurator, the parameters of the CPU 312-5BE23 may be set with "Object properties" of the CPU 312C during hardware configuration. Via a double-click on the CPU 312C the parameter window of the CPU may be accessed. Using the registers you get access to every parameter of the CPU.



Description of the parameters of the sub modules 'DI10/DO6' and 'Counter' ↬ *Chapter 6 'Deployment I/O periphery' on page 77*

5.8.2 CPU parameters

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 312-5BE03 is CPU 312C.
- 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.
- Interface: Here is the address of the MPI interface.
- Properties: By means of this button you can change the properties of the MPI interface.
- 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. 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.
- 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:
  This parameter is not relevant.
- 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: The priority may not be modified.
- 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.

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

5.9 Project transfer

Overview

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

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

5.9.1 Transfer via MPI

General

For transfer via MPI there is the following interface:

- X2: MPI interface

Net structure

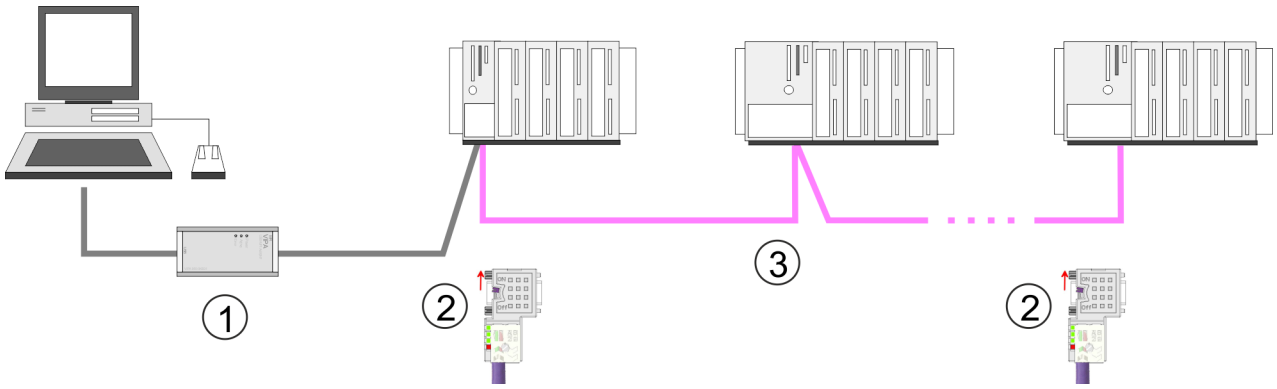
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

MPI programming cable

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

Terminating resistor

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



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

Proceeding transfer via MPI interface

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Load your project in the Siemens SIMATIC Manager.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].
5. Set in the register MPI the transfer parameters of your MPI net and type a valid *address*.
6. Switch to the register *Local connection*.

7. Set the COM port of the PC and the transfer rate 38400baud for the MPI programming cable.
8. Transfer your project via *'PLC ➔ Load to module'* via MPI to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

5.9.2 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 5.7 'Hardware configuration - Ethernet PG/OP channel' on page 49*

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.

5.9.3 Transfer via memory card

Proceeding transfer via memory card

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

- S7PROG.WLD
- AUTOLOAD.WLD

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

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

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

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

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

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

5.10 Accessing the web server

Access to the web server

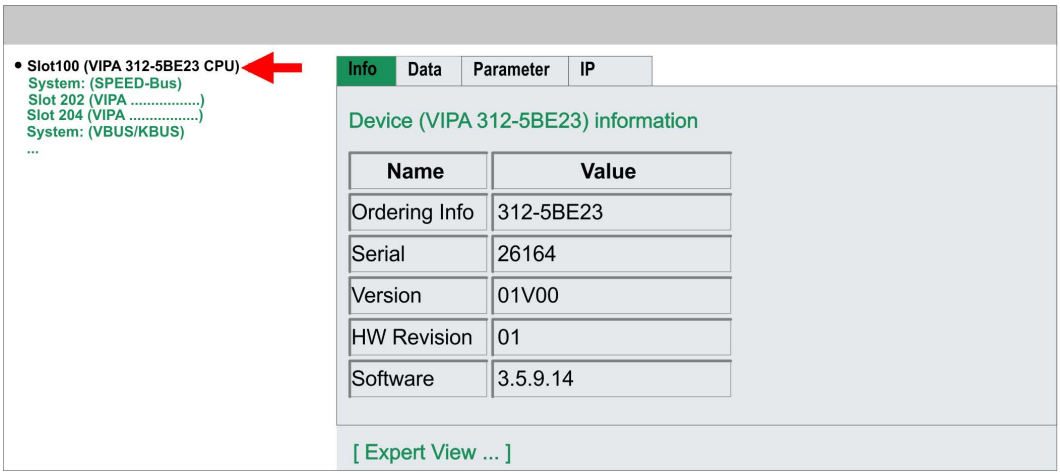
There is a web server, which can be accessed via the IP address of the Ethernet PG/OP channel with an Internet browser. At the web page information about the CPU and its connected modules can be found. *Chapter 5.7 'Hardware configuration - Ethernet PG/OP channel' on page 49*

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



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

Info - Expert View

Runtime Information		
Operation Mode	STOP	CPU: Status information
Mode Switch	RUNP	
System Time	01.09.09 00:35:30:812	CPU: Date, time
OB1-Cycle Time	cur = 0us, min = 0us, max = 0us, avg = 0us	CPU: Cyclic time: min = minimum cur = current max = maximum avg = average
Interface Information		
X2 (RS485/COM1)	MPI	Operating mode RS485 ■ MPI: MPI operation
X3 (RS485/COM2)	PtP	■ 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 / 524288 Bytes	CPU: Information to memory configuration Load memory, working memory (code/data)
WorkMemCode	0 / 248968 Bytes	
WorkMemData	0 / 248968 Bytes	
PG/OP Network Information		
Device Name	VIPA 312-5BE23 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	VIPA 312-5BE23 V3.7.3 Px000275.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

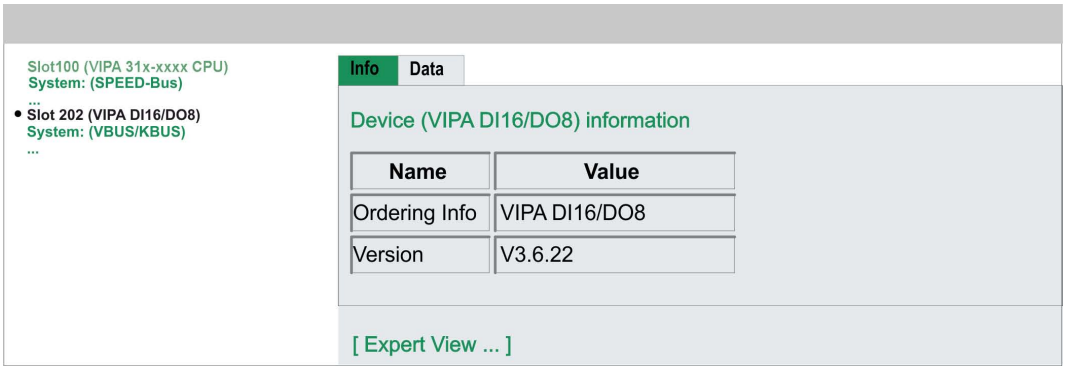
Currently nothing is displayed here.

IP

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

Info - Overview

CPU component: Digitale I/O



Info - Expert View

Internal Information		**Slot 202**
Module Type	0x4FD30000	Information for support
Module Firmware Information		
PRODUCT	VIPA DI16/DO8 V3.2.9.0	Name, firmware version

Data - Input data

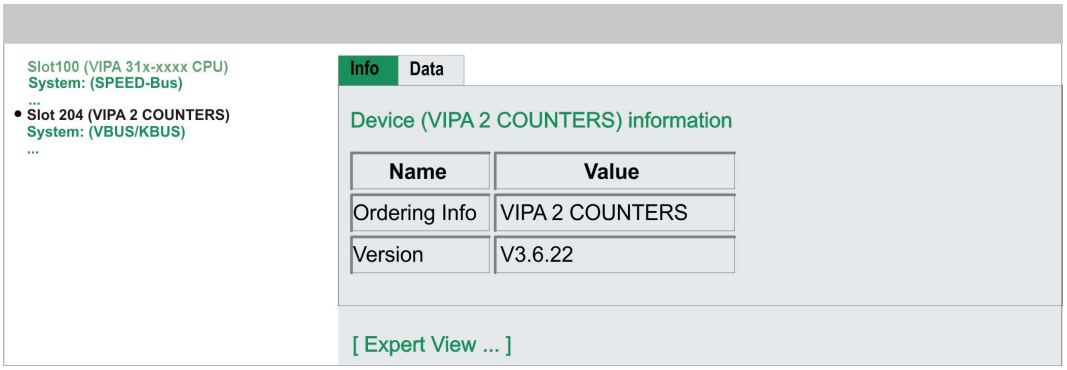
Offset	Width	Value (dec)	Value (hex)
124	1	0	00
125	1	0	00

Data - Output data

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

Info - Overview

CPU component: counter



Info - Expert View

Internal Information		**Slot 204**
Module Type	0x38C00000	Information for support
Module Firmware Information		
PRODUCT	VIPA 2 COUNTER V3.6.22	Name, firmware version

Data - Input data (16byte)

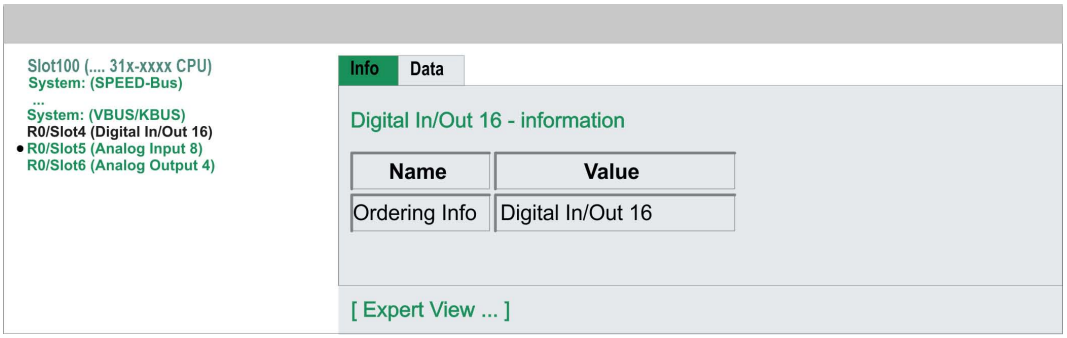
Offset	Width	Value (dec)	Value (hex)
768	1	0	00
769	1	0	00
770	1	0	00
771	1	0	00
772	1	0	00
773	1	0	00
774	1	0	00
775	1	0	00
776	1	0	00
777	1	0	00
778	1	0	00
779	1	0	00
780	1	0	00
781	1	0	00
782	1	0	00
783	1	0	00

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

5.11 Operating modes

5.11.1 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.

5.11.2 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

5.12 Overall reset

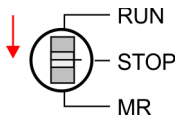
Overview

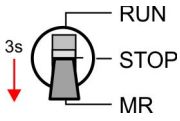
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

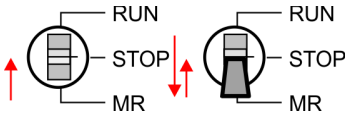
- Overall reset by means of the operating mode switch
- Overall reset by means of a configuration tool like e.g. the Siemens SIMATIC Manager

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

Overall reset by means of the operating mode switch







Proceeding

1. Your CPU must be in STOP mode. For this switch the operating mode switch of the CPU to STOP.
   ⇨ The ST LED is on.
2. Switch the operating mode switch to MR position for about 3 seconds.
   ⇨ The ST LED changes from blinking to permanently on.
3. Place the operating mode switch in the position STOP and switch it to MR and quickly back to STOP within a period of less than 3 seconds.
   ⇨ The overall reset is carried out. Here the ST LED flashes.
4. The overall reset has been completed when the ST LED is permanently on.

Overall reset by means of the Siemens SIMATIC Manager

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

1. For an overall reset the CPU must be switched to STOP state. You may place the CPU in STOP by the menu command *'PLC ➔ Operating mode'*.
2. You may request the overall reset by means of the menu command *'PLC ➔ Clean/Reset'*.
   ⇨ A dialog window opens. Here you can bring your CPU in STOP state, if not already done, and start the overall reset. During the overall reset procedure the the ST LED blinks. When the ST LED is on permanently the overall reset procedure has been completed.

Automatic reload

If there is a project S7PROG.WLD on the 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.

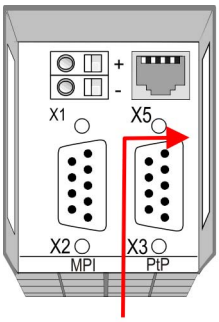
Reset to factory setting

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 5.14 'Reset to factory settings' on page 69*

5.13 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 312-5BE23 and its components with hardware release 01:

- 312-5BE23, Hardware release 01: Px000275.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

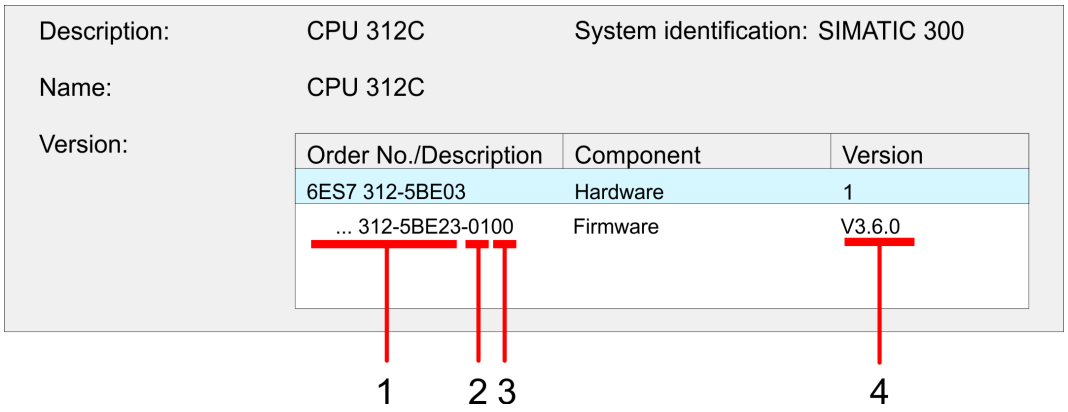
The CPU has an integrated website that monitors information about firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web site. The CPU has an integrated website that monitors information about firmware version of the SPEED7 components. The Ethernet PG/OP 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 5.10 'Accessing the web server' on page 56*

Determine CPU firmware version with module information

1. First establish an online connection to the CPU.
2. To show the module information you have to select *'PLC ➔ Module information'* in the Siemens SIMATIC Manager.

3. Via the register *'General'* the window with hardware and firmware version may be selected.

⇨ Due to software-technical reasons there is something different of the VIPA CPU 312-5BE23 to the CPU 312C from Siemens:



1 VIPA order number (VIPA 312-5BE23)
2 Hardware release (01)
3 Internal hardware version (00)
4 Firmware version (V3.6.0)

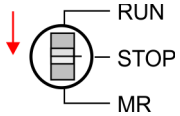
Every register of the module information dialog is supported by the VIPA CPUs. More about these registers may be found in the online help of the Siemens SIMATIC manager.

Load firmware and transfer it to memory card

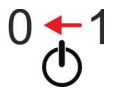
- Go to www.vipa.com
- Click on *'Service ➔ Download ➔ Firmware'*.
- Navigate via *'System 300S ➔ CPU'* to your CPU and download the zip file to your PC.
- Extract the zip file and copy the extracted pkg files to your memory card.

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 5.14 'Reset to factory settings' on page 69*.

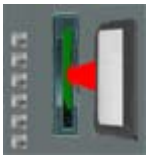
Transfer firmware from memory card into CPU



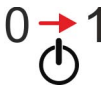
1. Switch the operating mode switch of your CPU in position STOP.



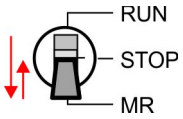
2. Turn off the power supply.



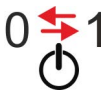
3. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card.



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



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

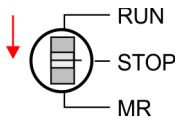


7. Turn power OFF and ON.
   ⇨ Now it is checked by the CPU, whether further firmware updates are to be executed. If so, again the LEDs SF and FC flash after a short start-up period. Continue with step 5. If the LEDs do not flash, the firmware update is finished.
8. Now execute a *Reset to factory setting*. After that the CPU is ready for duty.
   ↳ *Chapter 5.14 'Reset to factory settings' on page 69*

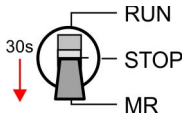
5.14 Reset to factory settings

Proceeding

- With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.
- Please regard that the MPI address is also reset to default 2 and the IP address of the Ethernet PG/OP channel is reset to 0.0.0.0!
- A factory reset may also be executed by the command *FACTORY_RESET*.
  ↳ *Chapter 5.17 'CMD - auto commands' on page 73*



1. Switch the CPU to STOP.



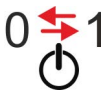
2. Push the operating mode switch down to position MR for 30 seconds. Here the ST LED blinks. After a few seconds the ST LED changes to static light. Now the ST LED changes between static light and blinking. Start here to count the static light of the ST LED.

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

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

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

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



5. Turn power OFF and ON.

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

5.15 Deployment storage media - MMC, MCC

Overview

At this slot the following storage media can be plugged:

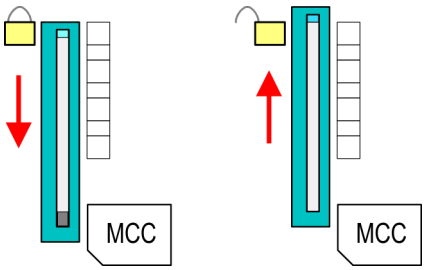
- SD respectively MMC (**M**ulti**m**edia **c**ard)
  - External memory card for programs and firmware.
- MCC - **M**emory **c**onfiguration **c**ard
  - External memory card (MMC) for programs and firmware with the possibility to unlock additional work memory.
  - The additional memory can be purchased separately.
  - To activate the corresponding card is to be installed and an *overall reset* is to be established. *Chapter 5.12 'Overall reset' on page 65*

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

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

MMC

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



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

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

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

MCC

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

- You may use the MCC also as "normal" MMC for storing your project.
- If the memory expansion on the MCC exceeds the maximum extendible memory range of the CPU, the maximum possible memory of the CPU is automatically used.
- You may determine the recent memory extension and the the remaining time after pulling the MCC via the integrated web page. *Chapter 5.10 'Accessing the web server' on page 56*
- When the MCC memory configuration has been taken over you may find the diagnostics entry 0xE400 in the diagnostics buffer of the CPU.
- After pulling the MCC the entry 0xE401 appears in the diagnostics buffer, the SF-LED is on and after 72 hours the CPU switches to STOP. A reboot is only possible after plugging-in the MCC again or after an overall reset.
- After re-plugging the MCC, the SF LED extinguishes and 0xE400 is entered into the diagnostics buffer. You may reset the memory configuration of your CPU to the initial status at any time by executing an overall reset without MCC.

CAUTION!
Please regard that the MCC must remain plugged when you've executed the memory expansion at the CPU. Otherwise the CPU switches to STOP after 72 hours. The MCC cannot be exchanged with a MCC of the same memory configuration. The activation code is fixed to the MCC by means of an unique serial number. Here the functionality as an external memory card is not affected.

Accessing the storage medium

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

After overall reset

- The CPU checks if a MCC is plugged. If so, the according additional memory is unlocked.
- The CPU checks whether a project S7PROG.WLD exists. If so, it is automatically loaded.

After PowerON

- The CPU checks whether a project AUTOLOAD.WLD exists. If so, an overall reset is executed and the project is automatically loaded.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists. If so the command file is loaded and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chapter 5.13 'Firmware update' on page 66*

Once in STOP state

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

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

5.16 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.
  - With the "extended" protection you transfer the protected blocks to a memory card into a WLD-file named protect.wld.
  - By plugging the memory card and then an overall reset the blocks in the protect.wld are permanently stored in the CPU.
  - You may protect OBs, FBs and FCs.
  - When back-reading the protected blocks into the PG, exclusively the block header are loaded. The block code that is to be protected remains in the CPU and cannot be read.

Protect blocks with protect.wld

1. Create a new wld file in your project engineering tool with *'File ➔ Memory Card file ➔ New'*.
2. Rename the wld file to "protect.wld".
3. Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.
4. Transfer the file protect.wld to a memory card.
5. Plug the memory card into the CPU and execute an *overall reset*. *Chapter 5.12 'Overall reset' on page 65*

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

Protection behaviour

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

Change respectively delete protected blocks

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

Usage of protected blocks

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

5.17 CMD - auto commands

Overview

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

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

Command file

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

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

Commands

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

Command	Description	Diagnostics entry
CMD_START	In the first line *CMD_START* is to be located.	0xE801
	There is a diagnostic entry if *CMD_START* is missing	0xE8FE
WAIT1SECOND	Waits about 1 second.	0xE803
WEBPAGE	The current web page of the CPU is stored at the memory card as" webpage.htm".	0xE804
LOAD_PROJECT	The function "Overall reset and reload from MMC" is executed. The wld file located after the command is loaded else "s7prog.wld" is loaded.	0xE805
SAVE_PROJECT	The recent project (blocks and hardware configuration) is stored as "s7prog.wld" at the memory card. If the file just exists it is renamed to "s7prog.old". If your CPU is password protected so you have to add this as parameter. Otherwise there is no project written. Example: *SAVE_PROJECT* password	0xE806
FACTORY_RESET	Executes "factory reset".	0xE807
DIAGBUF	The current diagnostics buffer of the CPU is stored as "diagbuff.txt" at the memory card.	0xE80B
SET_NETWORK	IP parameters for Ethernet PG/OP channel may be set by means of this command. The IP parameters are to be given in the order IP address, subnet mask and gateway in the format x.x.x.x each separated by a comma. Enter the IP address if there is no gateway used.	0xE80E
SET_MPI_ADDRESS	This lets you adjust the MPI interface on the value that follows the command. The setting is retained even after power cycle, firmware update or battery failure. With *Chapter 5.14 'Reset to factory settings' on page 69* you get the default setting.	0xE814
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)
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.

5.18 Diagnostic entries

Accessing diagnostic data

↳ *Appendix A 'System specific event IDs' on page 167*

- 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 5.17 'CMD - auto commands' on page 73*
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

5.19 Control and monitoring of variables with test functions

Overview

- For troubleshooting purposes and to display the status of certain variables you can access certain test functions via the menu item **Debug** of the Siemens SIMATIC Manager.
- The status of the operands and the RLO can be displayed by means of the test function *'Debug ➔ Monitor'*.
- The status of the operands and the RLO can be displayed by means of the test function *'PLC ➔ Monitor/Modify Variables'*.

'Debug ➔ Monitor'

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

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

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

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

'PLC ➔ Monitor/Modify Variables'

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

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

CAUTION!

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

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

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

6 Deployment I/O periphery

6.1 Overview

Hardware

At the CPU 312-5BE23 the connectors for digital in-/output and technological functions are integrated to a 2tier casing.

Project engineering

The project engineering takes place in the Siemens SIMATIC manager as Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6). Here the CPU 312-5BE23 is parameterized by the "Properties" dialog of the CPU 312C. For parameterization of the digital I/O periphery and the technological functions the corresponding submodule of the CPU 312C may be used.

I/O periphery

The integrated I/Os of the 312-5BE23 may be used for technological functions or as standard I/Os. Technological functions and standard I/Os may be used simultaneously with appropriate hardware. Read access to inputs used by technological functions is possible. Write access to used outputs is not possible.

Technological functions

Up to 2 channels may be parameterized as technological function. The parameterization of the appropriate channel is made in the hardware configurator by the *count* submodule of the CPU 312C. There are the following technological functions:

- Continuous count
- Single count
- Periodic count
- Frequency measurement
- Pulse width modulation (PWM)

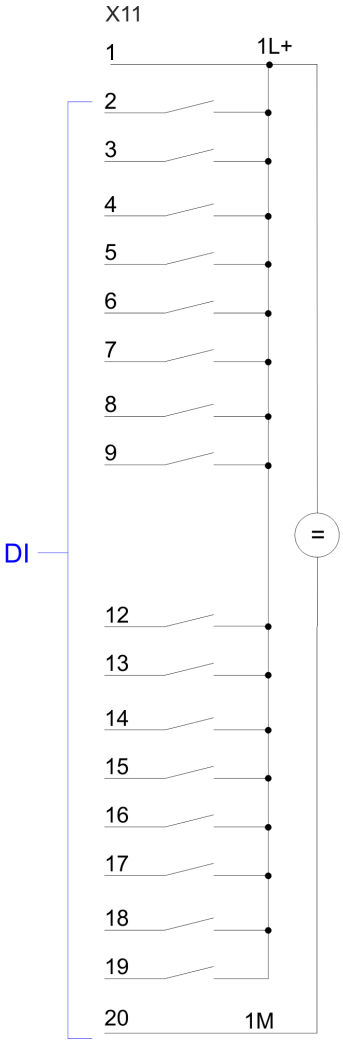
The controlling of the corresponding counter mode happens by means of the SFB COUNT (SFB 47) of the user program.

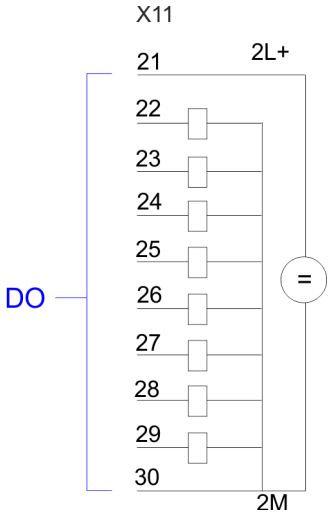
6.2 In-/Output area CPU 312-5BE23

Overview

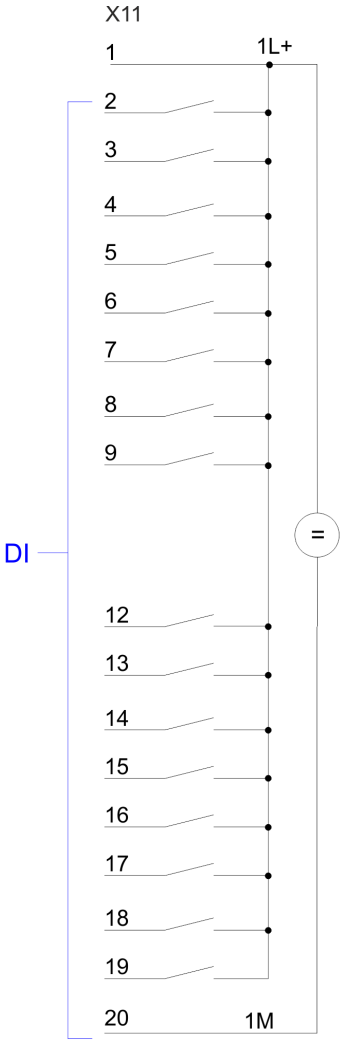
The CPU 312-5BE23 has the following digital in and output channels integrated in one casing:

- Digital Input: 16xDC 24V, with interrupt capability
- Digital Output: 8xDC 24V, 0.5A
- Technological functions: 2 channels



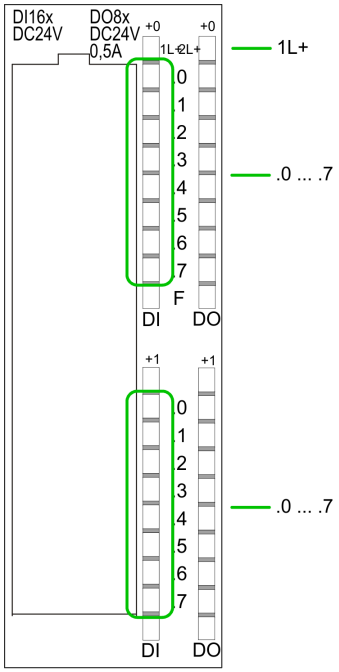


CAUTION!
Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+.



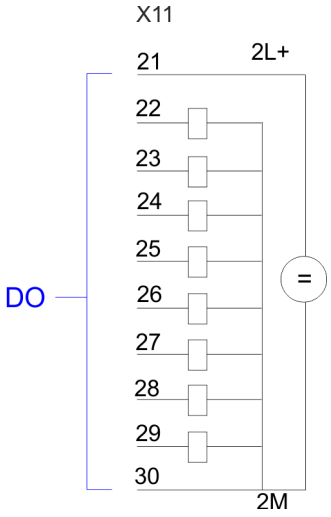
Pin assignment X11: DI

Pin	Assignment
1	1L+ Power supply +DC 24V
2	I+0.0 / Channel 0 (A) / Pulse
3	I+0.1 / Channel 0 (B) / Direction
4	I+0.2 / Channel 0 HW gate
5	I+0.3 / Channel 1 (A) / Pulse
6	I+0.4 / Channel 1 (B) / Direction
7	I+0.5 / Channel 1 HW gate
8	I+0.6
9	I+0.7
10	not used
11	not used
12	I+1.0
13	I+1.1
14	I+1.2
15	I+1.3
16	I+1.4 / Channel 0 Latch
17	I+1.5 / Channel 1 Latch
18	I+1.6
19	I+1.7
20	Ground 1M DI



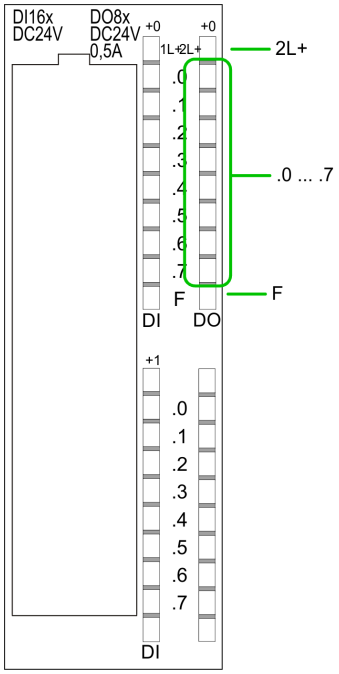
Status indication X11: DI

- 1L+
  - LED (green)
    Supply voltage available for DI
- .0 7
  - LEDs (green)
    I+0.0 ... I+0.7
    I+1.0 ... I+1.7
    Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated



Pin assignment X11: DO

Pin	Assignment
21	2L+ Power supply +DC 24V
22	O+0.0 / Channel 0 Output
23	O+0.1 / Channel 1 Output
24	Q+0.2
25	Q+0.3
26	Q+0.4
27	Q+0.5
28	Q+0.6
29	Q+0.7
30	Ground 2M DO
31 ... 40	not used



Status indication X11: DO

- 2L+
  - LED (green)
    Supply voltage available for DO
- .0 7
  - LEDs (green)
    Q+0.0 ... Q+0.7
    The according LED is on at active output
- F
  - LED (red)
    Overload or short circuit error

6.3 Address assignment

Input range

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital Input I+0.0 ... I+0.7
	125	Byte	Digital Input I+1.0 ... I+1.7
Counter	768	DInt	Channel 0: Count value / Frequency value
	772	DInt	Channel 1: Count value / Frequency value
	776	DInt	reserved
	780	DInt	reserved

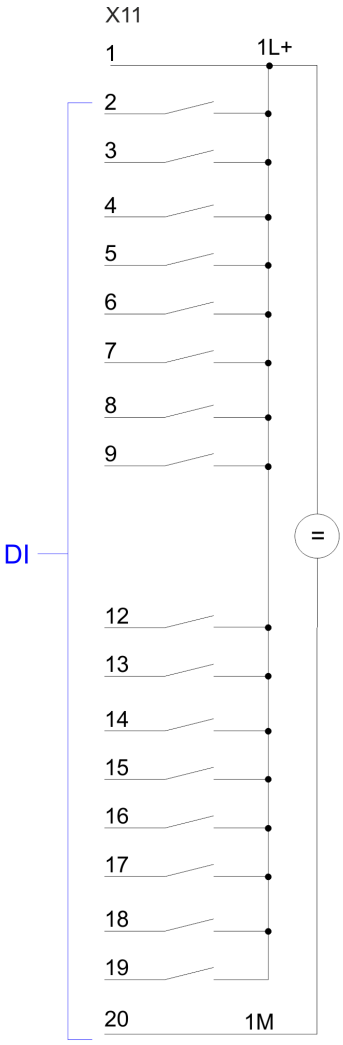
Output range

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital Output Q+0.0 ... Q+0.7
Counter	768	DWort	reserved
	772	DWort	reserved
	776	DWort	reserved
	780	DWort	reserved

6.4 Digital part

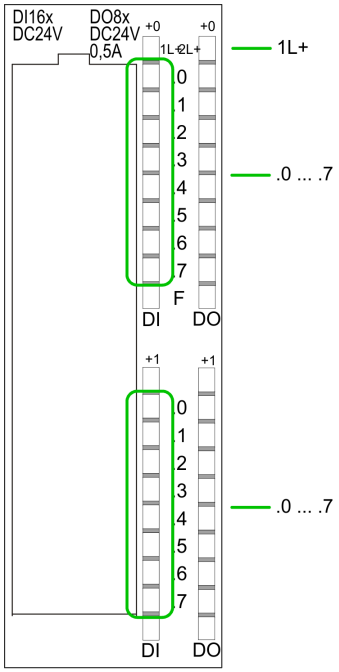
312-5BE23

The digital part consists of 16 input -, 8 output channels and 2 channels for technological functions. Each of these digital input- respectively output channels show its state via a LED. By means of the parameterization you may assign interrupt properties to the inputs I +0.0 to I+1.1.



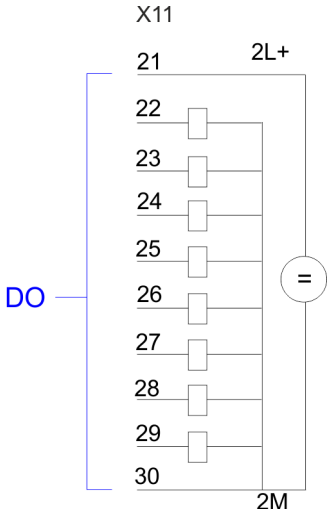
Pin assignment X11: DI

Pin	Assignment
1	1L+ Power supply +DC 24V
2	I+0.0 / Channel 0 (A) / Pulse
3	I+0.1 / Channel 0 (B) / Direction
4	I+0.2 / Channel 0 HW gate
5	I+0.3 / Channel 1 (A) / Pulse
6	I+0.4 / Channel 1 (B) / Direction
7	I+0.5 / Channel 1 HW gate
8	I+0.6
9	I+0.7
10	not used
11	not used
12	I+1.0
13	I+1.1
14	I+1.2
15	I+1.3
16	I+1.4 / Channel 0 Latch
17	I+1.5 / Channel 1 Latch
18	I+1.6
19	I+1.7
20	Ground 1M DI



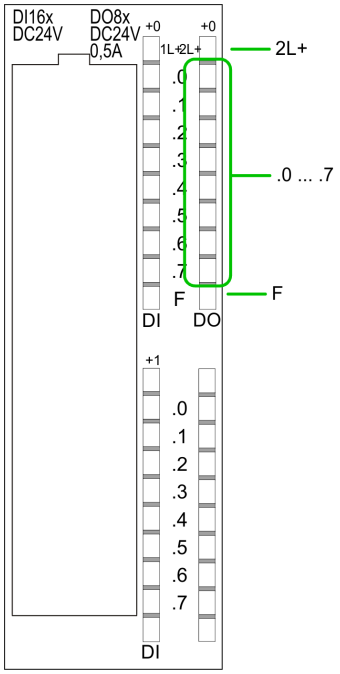
Status indication X11: DI

- 1L+
  - LED (green)
    Supply voltage available for DI
- .0 7
  - LEDs (green)
    I+0.0 ... I+0.7
    I+1.0 ... I+1.7
    Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated



Pin assignment X11: DO

Pin	Assignment
21	2L+ Power supply +DC 24V
22	O+0.0 / Channel 0 Output
23	O+0.1 / Channel 1 Output
24	Q+0.2
25	Q+0.3
26	Q+0.4
27	Q+0.5
28	Q+0.6
29	Q+0.7
30	Ground 2M DO
31 ... 40	not used



Status indication X11: DO

- 2L+
  - LED (green)
    Supply voltage available for DO
- .0 7
  - LEDs (green)
    Q+0.0 ... Q+0.7
    The according LED is on at active output
- F
  - LED (red)
    Overload or short circuit error

6.4.1 Access to the I/O area

The 312-5BE23 creates in its peripheral area an area for input respectively output data. Without a hardware configuration the in the following specified default addresses are used.

6.4.1.1 Address assignment

Input range

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital Input I+0.0 ... I+0.7
	125	Byte	Digital Input I+1.0 ... I+1.7
Counter	768	DInt	Channel 0: Count value / Frequency value
	772	DInt	Channel 1: Count value / Frequency value
	776	DInt	reserved
	780	DInt	reserved

Output range

Sub module	Default address	Access	Assignment
DI10/DO6	124	Byte	Digital Output Q+0.0 ... Q+0.7
Counter	768	DWort	reserved
	772	DWort	reserved
	776	DWort	reserved
	780	DWort	reserved

6.4.2 Parameterization - Digital part

Parameter data

Parameters of the digital part may be set by means of the *DI10/DO6* submodule of the CPU 312C from Siemens during hardware configuration. In the following all parameters are specified, which may be used with the hardware configuration of the digital periphery.

General

This provides the short description of the digital periphery. At *Comment* information about the module such as purpose may be entered.

Addresses

At this register the start address of the in-/output periphery may be set.

Inputs

Here there are the following adjustment possibilities:

- Hardware interrupt
  - A hardware interrupt may be optionally triggered on the rising or falling edge of an input. A diagnostic interrupt is only supported together with hardware interrupt lost. Select with the arrow keys the input and activate the desired hardware interrupt.
- Input delay
  - The input delay may be configured per channel in groups of four. Please note that in the parameter window only the value 0.1ms may be set. At the other values 0.35ms is internally used for input delay.

Outputs

There are no parameters for the digital output channels.

6.5 Counter

6.5.1 Counter - Fast introduction

Overview

The CPU 312-5BE23 has in-/outputs, which may be used for technological functions respectively as standard periphery. Technological functions and standard I/O may be used simultaneously with appropriate hardware. Read access to inputs used by technological functions is possible. Write access to used outputs is not possible. The parameterization of the corresponding channel is made in the hardware configurator by means of the *Count* submodule of the CPU 312C from Siemens. Now the following technological functions at 2 channels are at the disposal:

- Continuous count, e.g. for position decoding with Incremental encoder
- Single count, e.g. for unit decoding to a maximum limit
- Periodical count, e.g. for applications with repeated counting operations

Independent of the number of activated counters for the CPU 312-5BE23 the maximum frequency amounts to 10kHz.

The controlling of the appropriate modes of operation is made from the user program by the SFB COUNT (SFB 47).

Pin assignment

↳ *Chapter 6.4 'Digital part' on page 82*

Preset respectively parameterize counter

The counter signal is detected and evaluated during counting operation. Every counter occupies one double word in the input range for the *counter register*. In the operating modes "single count" and "periodical count" an end respectively start value may be defined according to the counting direction up respectively down. Each counter has parameterizable additional functions as gate function, latch function, comparison value, hysteresis and hardware interrupt. Each counter parameter may be set by the *Count* submodule of the Siemens CPU 312C. Here is defined among others:

- Interrupt behavior
- Max. frequency
- Counter mode respectively behaviour
- Stat, end, comparison value and hysteresis

Parameterization

1. Start the Siemens SIMATIC Manager with your project and open the hardware configurator.
2. Place a profile rail.
3. Configure at slot 2 the corresponding CPU from Siemens CPU 312C.
4. Open the dialog window "Properties" by a double click to the *Count* submodule of the CPU.
5. As soon as an operating mode to the corresponding channel is selected, a dialog window for this operating mode is created and displayed and filled with default parameters.
6. Execute the wished parameterization.
7. Save the project with *'Station ➔ Save and compile'*.
8. Transfer the project to the CPU.

Controlling the counter functions

The SFB COUNT (SFB 47) should cyclically be called (e.g. OB 1) for controlling the counter functions. The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored. Among others the SFB 47 contains a request interface. Hereby you get read and write access to the registers of the appropriate counter. So that a new job may be executed, the previous job must have be finished with JOB_DONE = TRUE. Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Writing accesses to outputs of the instance DB is not permissible.

You must not call an SFB you have configured in your program in another program section under another priority class, because the SFB must not interrupt itself. Example: It is not allowed to call the same SFB both in OB 1 and in the interrupt OB.

Controlling the counter

The counter is controlled by the internal gate (i gate). The i gate is the result of logic operation of hardware gate (HW gate) and software gate (SW gate), where the HW gate evaluation may be deactivated by the parameterization.

HW gate:	open (activate):	Edge 0-1 at hardware gate$_x$ input of the module
	close (deactivate):	Edge 1-0 at hardware gate$_x$ input of the module

SW gate:	open (activate):	In application program by setting SW_GATE of the SFB 47
	close (deactivate):	In application program by resetting SW_GATE of the SFB 47

Read counter

The counter values may be read by the output parameter COUNTVAL of the SFB 47. There is also the possibility for direct access to the counter values by means of the input address of the *Count* submodule.

Counter inputs (Connections)

There are the following possibilities for connection to the technological functions:

- 24V incremental encoder, equipped with two tracks with 90° phase offset
- 24V pulse generator with direction signal
- 24V proximity switch (e.g. BERO or light barrier)

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

- *Channel$_x$ (A)*

  Pulse input for count signal res. track A of an encoder. Here you may connect encoder with 1-, 2- or 4-tier evaluation.
- *Channel$_x$ (B)*

  Direction signal res. track B of the encoder. Via the parameterization you may invert the direction signal.
- *Hardware gate$_x$*

  This input allows you to open the HW gate with a high peek and thus start a count process. The usage of the HW gate may be parameterized.
- *Latch$_x$*

  With an edge 0-1 at Latch$_x$ the recent counter value is stored in a memory that you may read at need.

Counter outputs

Every counter has an assigned output channel. The following behavior for the output channel may be set via parameterization:

- No comparison: Output is not controlled and is switched in the same way as a normal output.
- Count value ≥ comparison value: Output is set as long as counter value ≥ comparison value.
- Count value ≤ comparison value: Output is set as long as counter value ≤ comparison value.
- Pulse at comparison value: You can specify a pulse period for adaptation to the actuators you are using. The output is set for the given pulse duration, as soon as the counter reached the comparison value. If you have parameterized a main count direction the output is only set when reaching the comparison value from the main counting direction. The maximum pulse duration may amount to 510ms. By setting 0 as pulse duration the output gets set as long as the comparison conditions are fulfilled.

6.5.1.1 Parameter overview

In the following the parameters are listed which may be used for counter configuration during hardware configuration.

General

Here the short description of the counter function may be found. At *Comment* information about the module such as purpose may be entered.

Addresses

Here the start address of the in- output periphery is set.

Basic parameters

Here the interrupts the counter functions should trigger may be selected. You have the following options:

- None: There is no interrupt triggered.
- Process: The counting function triggers a hardware interrupt.
- Diagnostics and Process: With the 312-5BE23 the diagnostic interrupt of the digital in-/output periphery is only supported in connection with "hardware interrupt lost".

Count

Parameters	Description	Range of values	Default
Main count direction	■ *None:* No restriction of the counting range ■ *Up:* Restricts the up-counting range. Counter starts at 0 or load value, counts in positive direction up to the declaration end value -1 and then jumps back to load value at the next positive transducer pulse. ■ *Down:* Restricts the down-counting range. The Counter starts at the declared start value or load value in negative direction, counts to 1 and then jumps to start value at the next negative encoder pulse.	■ None ■ Up ■ Down (not with continuous count)	■ None
End value/ Start value	*End value*, with up-count as default. *Start value*, with down-count as default.	2...2147483647 (2^{31}-1)	2147483647 (2^{31}-1)
Gate function	■ *Cancel count:* The count starts when the gate opens and resumes at the load value when the gate opens again. ■ *Stop count:* The count is interrupted when the gate closes and resumed at the last actual value when the gate opens again.	■ Abort the count operation ■ Interrupt the count operation	Cancel count
Comparison value	The count value is compared with the comparison value. see also the parameter "Characteristics of the output": ■ No main direction of count ■ Up-count as default ■ Down-count as default	-2^{31} to +2^{31}-1 -2^{31} to End value 1 to +2^{31}-1	0
Hysteresis	A hysteresis is used to eliminate frequent output jitter if the count value lies within the range of the comparison value. 0 and 1 means: Hysteresis switched off	0 to 255	0
max. frequency: counting signals/hardware gate	You can set the maximum frequency of the track A/pulse, track B/direction and hardware gate signals in fixed steps.	10, 5, 2, 1kHz	10kHz
max. frequency: Latch	You can set the maximum frequency of the latch signal in fixed steps.	10, 5, 2, 1kHz	10kHz

Parameters	Description	Range of values	Default
Signal evaluation	The count and direction signals are connected to the input. A rotary transducer is connected to the input (single, dual or quadruple evaluation).	■ Pulse/Direction ■ Rotary encoder single ■ Rotary encoder, double ■ Rotary encoder quadruple	Pulse/Direction
Hardware gate	In the activated state the Gate control is made via SWgate and HW-gate, otherwise via SW-gate only.	■ activated ■ deactivated	deactivated
Count direction inverted	In the activated state the "direction" input signal is inverted.	■ activated ■ deactivated	deactivated
Characteristics of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter.	■ No comparison ■ Count ≥ comparison value ■ Count ≤ comparison value ■ Pulse at comparison value	No comparison
Pulse duration	With the setting "Characteristics of the output: Pulse at comparison value" the pulse duration of the output signal may be specified. Only even values are possible. The value is internal multiplied with 1.024ms.	0 to 510	0
Hardware interrupt: Hardware gate opening	In the activated state a hardware interrupt is generated when the hardware gate opens while the software gate is open.	■ activated ■ deactivated	
Hardware interrupt: Hardware gate closing	In the activated state a hardware interrupt is generated when the hardware gate closes while the software gate is open.	■ activated ■ deactivated	deactivated
Hardware interrupt: On reaching comparator	In the activated state a hardware interrupt is triggered on reaching the comparator (reaction) value. The process interrupt may only be released if in addition the value of "Characteristics of the output" is not "no comparison".	■ activated ■ deactivated	deactivated
Hardware interrupt: Overflow	In the activated state a hardware interrupt is generated in the event of an overflow (exceeding the upper count limit).	■ activated ■ deactivated	deactivated
Hardware interrupt: Underflow	In the activated state a hardware interrupt is generated in the event of an underflow (undershooting the lower count limit).	■ activated ■ deactivated	deactivated

6.5.2 SFB 47 - COUNT - Counter controlling

Description

The SFB 47 is a specially developed block for compact CPUs for controlling of the counters. The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored. With the SFB COUNT (SFB 47) you have following functional options:

- Start/Stop the counter via software gate *SW_GATE*
- Enable/control digital output DO
- Read the status bit
- Read the actual count and latch value
- Request to read/write internal counter registers

Parameters

Name	Data type	Address (Instance DB)	Default value	Comment
LADDR	WORD	0.0	300h	This parameter is not evaluated. Always the internal I/O periphery is addressed.
CHANNEL	INT	2.0	0	Channel number
SW_GATE	BOOL	4.0	FALSE	Enables the Software gate
CTRL_DO	BOOL	4.1	FALSE	Enables the output False: Standard Digital Output
SET_DO	BOOL	4.2	FALSE	Parameter is not evaluated
JOB_REQ	BOOL	4.3	FALSE	Initiates the job (edge 0-1)
JOB_ID	WORD	6.0	0	Job ID
JOB_VAL	DINT	8.0	0	Value for write jobs
STS_GATE	BOOL	12.0	FALSE	Status of the internal gate
STS_STRT	BOOL	12.1	FALSE	Status of the hardware gate
STS_LTCH	BOOL	12.2	FALSE	Status of the latch input
STS_DO	BOOL	12.3	FALSE	Status of the output
STS_C_DN	BOOL	12.4	FALSE	Status of the down-count Always indicates the last direction of count. After the first SFB call *STS_C_DN* is set FALSE.
STS_C_UP	BOOL	12.5	FALSE	Status of the up-count Always indicates the last direction of count. After the first SFB call *STS_C_UP* is set TRUE.
COUNTVAL	DINT	14.0	0	Actual count value
LATCHVAL	DINT	18.0	0	Actual latch value
JOB_DONE	BOOL	22.0	TRUE	New job can be started
JOB_ERR	BOOL	22.1	FALSE	Job error
JOB_STAT	WORD	24.0	0	Job error ID

Local data only in instance DB

Name	Data type	Address (Instance DB)	Default value	Comment
RES00	BOOL	26.0	FALSE	reserved
RES01	BOOL	26.1	FALSE	reserved
RES02	BOOL	26.2	FALSE	reserved
STS_CMP	BOOL	26.3	FALSE	Comparator Status * Status bit *STS_CMP* indicates that the comparison condition of the comparator is or was reached. *STS_CMP* also indicates that the output was set. (*STS_DO* = TRUE).
RES04	BOOL	26.4	FALSE	reserved
STS_OFLW	BOOL	26.5	FALSE	Overflow status *
STS_UFLW	BOOL	26.6	FALSE	Underflow status *
STS_ZP	BOOL	26.7	FALSE	Status of the zero mark * The bit is only set when counting without main direction. Indicates the zero mark. This is also set when the counter is set to 0 or if is start counting.
JOB_OVAL	DINT	28.0		Output value for read request.
RES10	BOOL	32.0	FALSE	reserved
RES11	BOOL	32.1	FALSE	reserved
RES_STS	BOOL	32.2	FALSE	Reset status bits: Resets the status bits: STS_CMP, STS_OFLW, STS_ZP. The SFB must be twice called to reset the status bit.

*) Reset with RES_STS

Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Writing accesses to outputs of the instance DB is not permissible.

Counter request interface

To read/write counter registers the request interface of the SFB 47 may be used. So that a new job may be executed, the previous job must have be finished with *JOB_DONE* = TRUE.

Proceeding

The deployment of the request interface takes place at the following sequence:

1. Edit the following input parameters:

Name	Data type	Address (DB)	Default	Comment
JOB_REQ	BOOL	4.3	FALSE	Initiates the job (edges 0-1) *
JOB_ID	WORD	6.0	0	Job ID: 00h Job without function 01h Writes the *count value* 02h Writes the *load value* 04h Writes the *comparison value* 08h Writes the *hysteresis* 10h Writes the *pulse duration* 20h Writes the *end value* 82h Reads the *load value* 84h Reads the *comparison value* 88h Reads the *hysteresis* 90h Reads the *pulse duration* A0h Reads the *end value*
JOB_VAL	DINT	8.0	0	Value for write jobs

*) State remains set also after a CPU STOP-RUN transition.

2. Call the SFB. The job is processed immediately. *JOB_DONE* only applies to SFB run with the result FALSE. *JOB_ERR* = TRUE if an error occurred. Details on the error cause are indicated at *JOB_STAT*.

Name	Data type	Address (DB)	Default	Comment
JOB_DONE	BOOL	22.0	TRUE	New job can be started
JOB_ERR	BOOL	22.1	FALSE	Job error
JOB_STAT	WORD	24.0	0000h	Job error ID 0000h No error 0121h *Comparison value* too low 0122h *Comparison value* too high 0131h *Hysteresis* too low 0132h *Hysteresis* too high 0141h *Pulse duration* too low 0142h *Pulse duration* too high 0151h *Load value* too low 0152h *Load value* too high 0161h *Count value* too low 0162h *Count value* too high 01FFh Invalid *job ID*

3. A new job may be started with *JOB_DONE* = TRUE.

4. A value to be read of a read job may be found in *JOB_OVAL* in the instance DB at address 28.

Permitted value range for JOB_VAL

Continuous count:

Job	Valid range
Writing *counter* directly	-2147483647 (-2^{31}+1) ... +2147483646 (2^{31}-2)
Writing the *load value*	-2147483647 (-2^{31}+1) ... +2147483646 (2^{31}-2)
Writing *comparison value*	-2147483648 (-2^{31}) ... +2147483647 (2^{31}-1)
Writing *hysteresis*	0 ... 255
Writing *pulse duration**	0 ... 510ms

Single/periodic count, no main count direction:

Job	Valid range
Writing *counter* directly	-2147483647 (-2^{31}+1) ... +2147483646 (2^{31}-2)
Writing the *load value*	-2147483647 (-2^{31}+1) ... +2147483646 (2^{31}-2)
Writing *comparison value*	-2147483648 (-2^{31}) ... +2147483647 (2^{31}-1)
Writing *hysteresis*	0 ... 255
Writing *pulse duration**	0 ... 510ms

Single/periodic count, main count direction up:

Job	Valid range
End value	2 ... +2147483646 (2^{31}-1)
Writing *counter* directly	-2147483648 (-2^{31}) ... *end value* -2
Writing the *load value*	-2147483648 (-2^{31}) ... *end value* -2
Writing *comparison value*	-2147483648 (-2^{31}) ... *end value* -1
Writing *hysteresis*	0 ... 255
Writing *pulse duration**	0 ... 510ms

Single/periodic count, main count direction down:

Job	Valid range
Writing *counter directly*	2 ... +2147483647 (2^{31}-1)
Writing the *load value*	2 ... +2147483647 (2^{31}-1)
Writing *comparison value*	1 ... +2147483647 (2^{31}-1)
Writing *hysteresis*	0 ... 255

Job	Valid range
Writing *pulse duration**	0 ... 510ms

*) Only even values allowed. Odd values are automatically rounded.

Latch function

As soon as during a count process an edge 0-1 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 *LATCHVAL* of the SFB 47.

A just in *LATCHVAL* loaded value remains after a STOP-RUN transition.

6.5.3 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

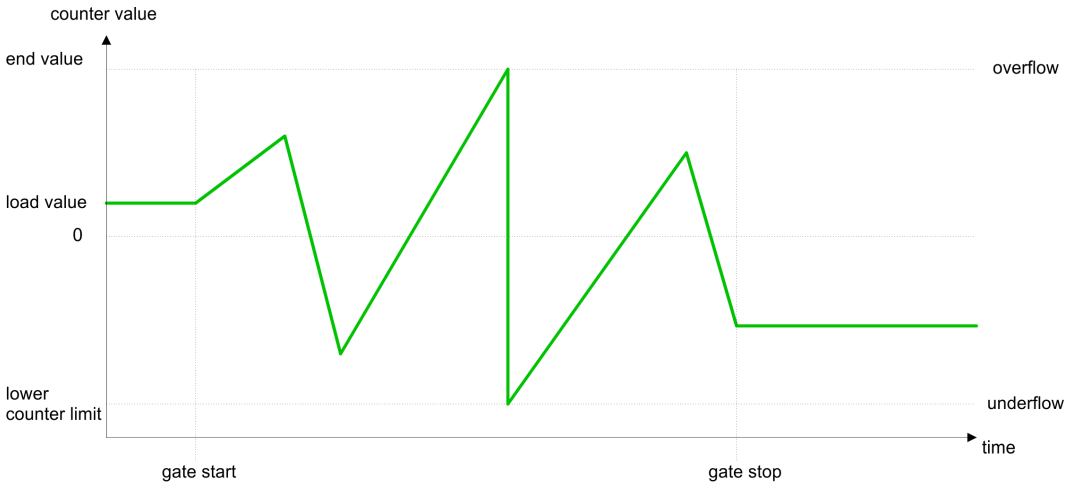
The count process continuous after closing and restart of the gate beginning with the last recent counter value.

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)

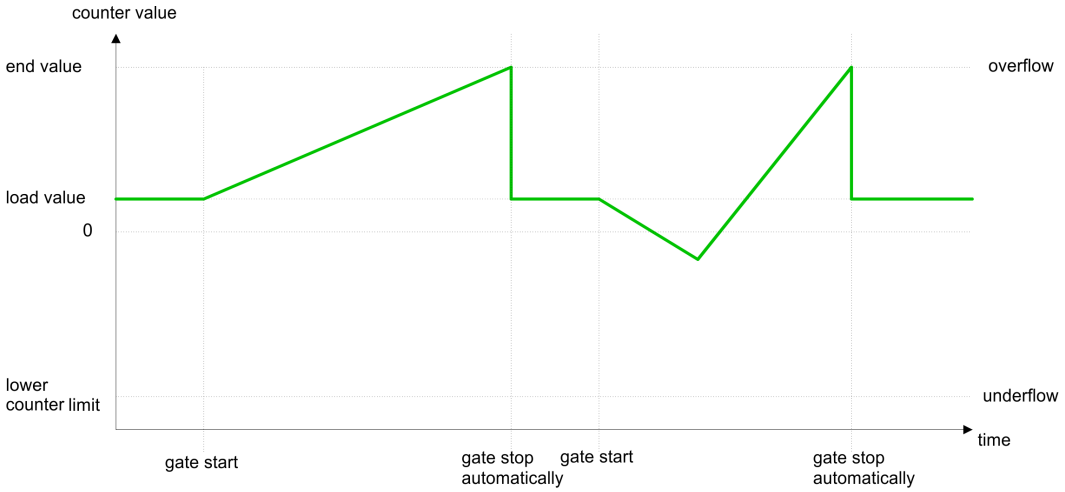
Interrupting gate control:

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.

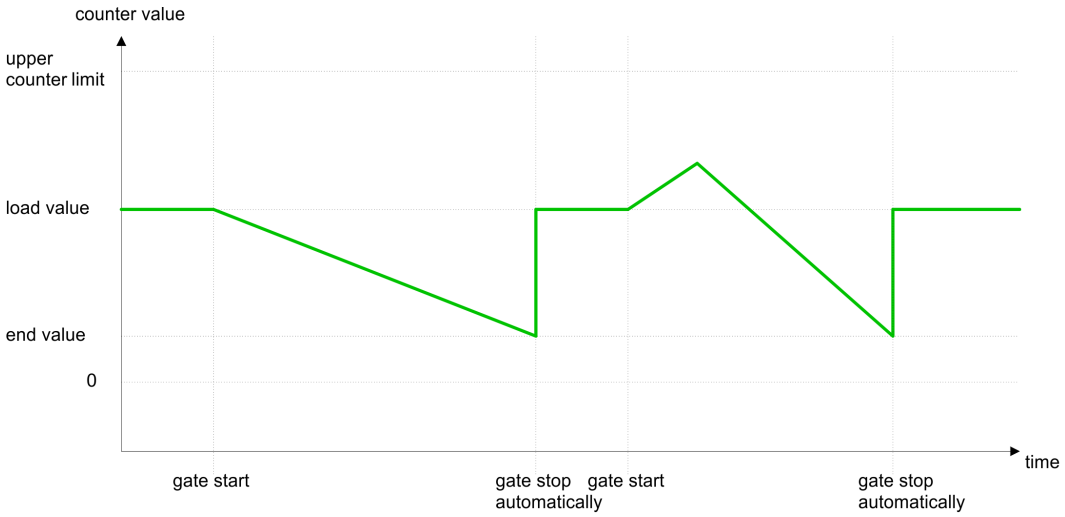
Limits	Valid value range
Limit value	-2 147 483 647 (-2^{31} +1) to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})



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)

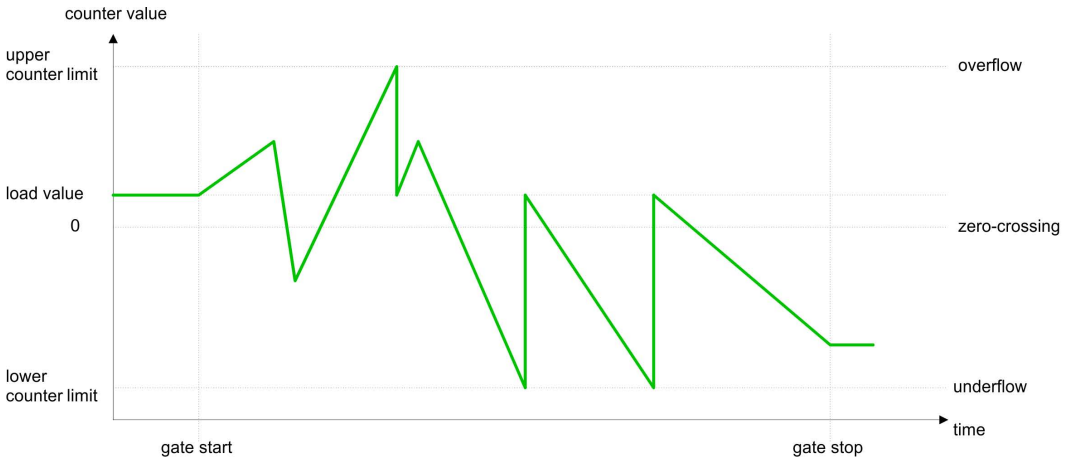


Count Periodically

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.

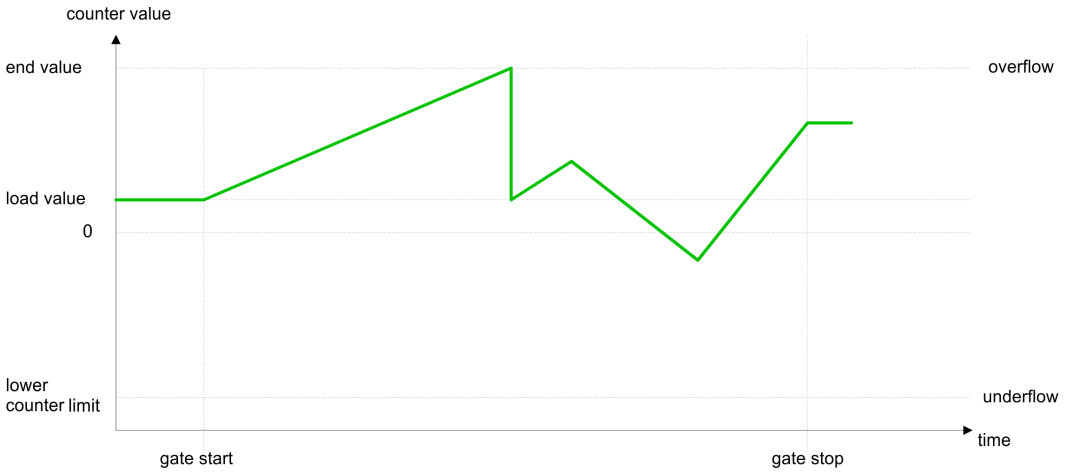
Limits	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31} -1)



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.

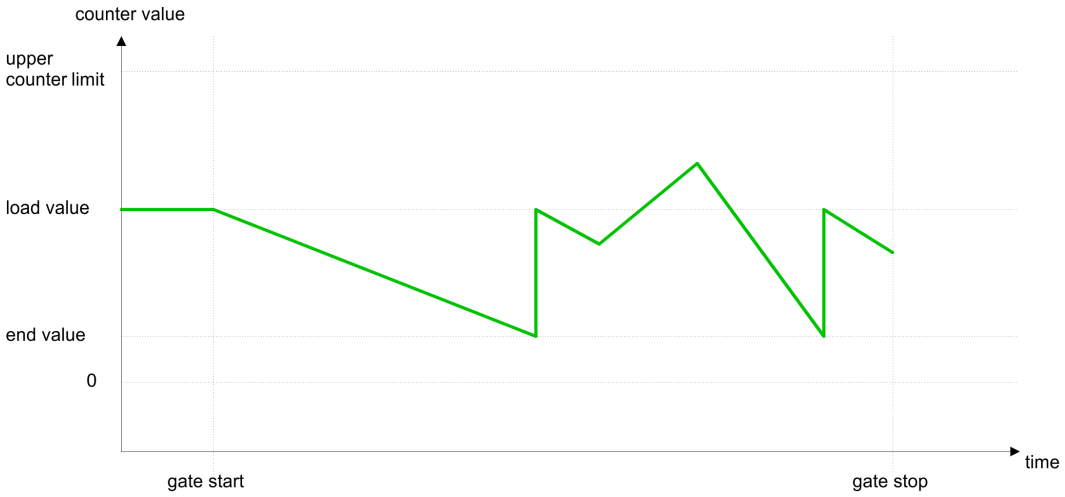
Limits	Valid value range
Limit value	-2 147 483 647 (-2^{31} +1) to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})



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)



6.5.4 Counter - Additional functions

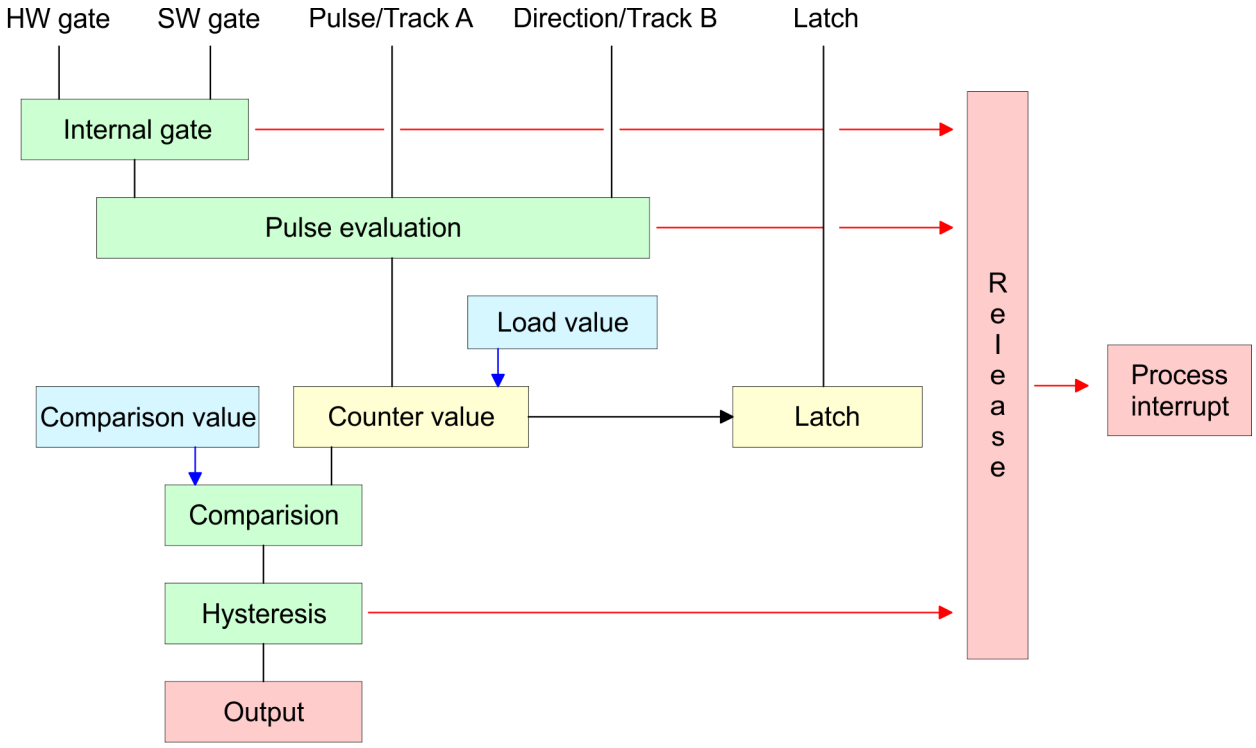
Overview

The following additional functions may be set via the parameterization for every counter:

- Gate function
  The gate function serves the start, stop and interrupt of a count function.
- Latch function
  An edge 0-1 at the digital input "Latch" stores the recent counter value in the latch register.
- Comparison
  You may set a comparison value that activates res. de-activates a digital output res. releases a hardware interrupt depending on the counter value.
- Hysteresis
  The setting of a hysteresis avoids for example a high output toggling when the value of an encoder signal shifts around a comparison value.

Schematic structure

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



Gate function

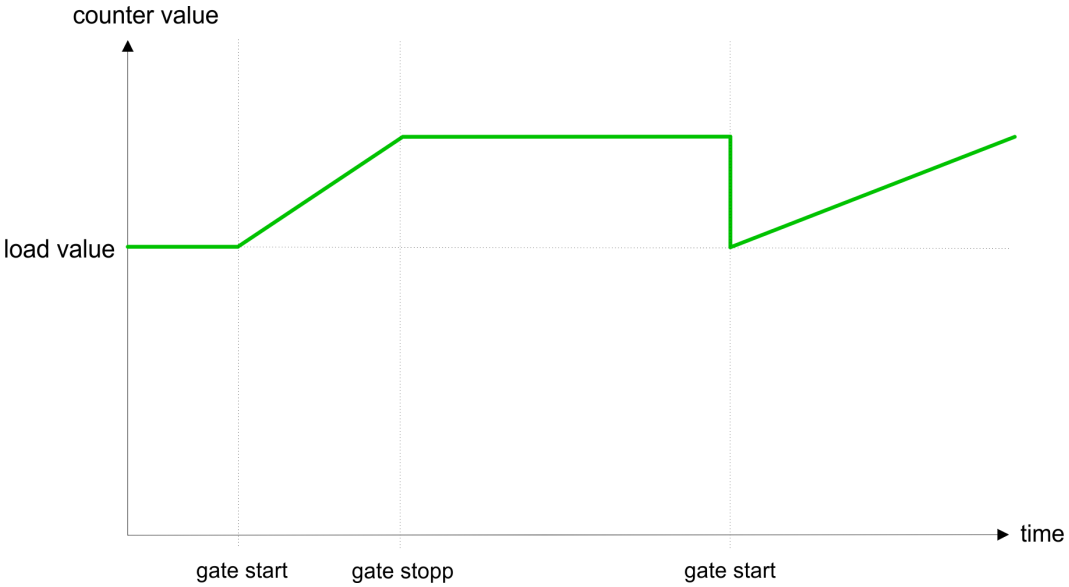
The counter is controlled by the internal gate (i gate). The i gate is the result of logic operation of hardware gate (HW gate) and software gate (SW gate), where the HW gate evaluation may be deactivated by the parameterization.

HW gate:	open (activate):	Edge 0-1 at hardware gate$_x$ input of the module
	close (deactivate):	Edge 1-0 at hardware gate$_x$ input of the module
SW gate:	open (activate):	In application program by setting SW_GATE of the SFB 47
	close (deactivate):	In application program by resetting SW_GATE of the SFB 47

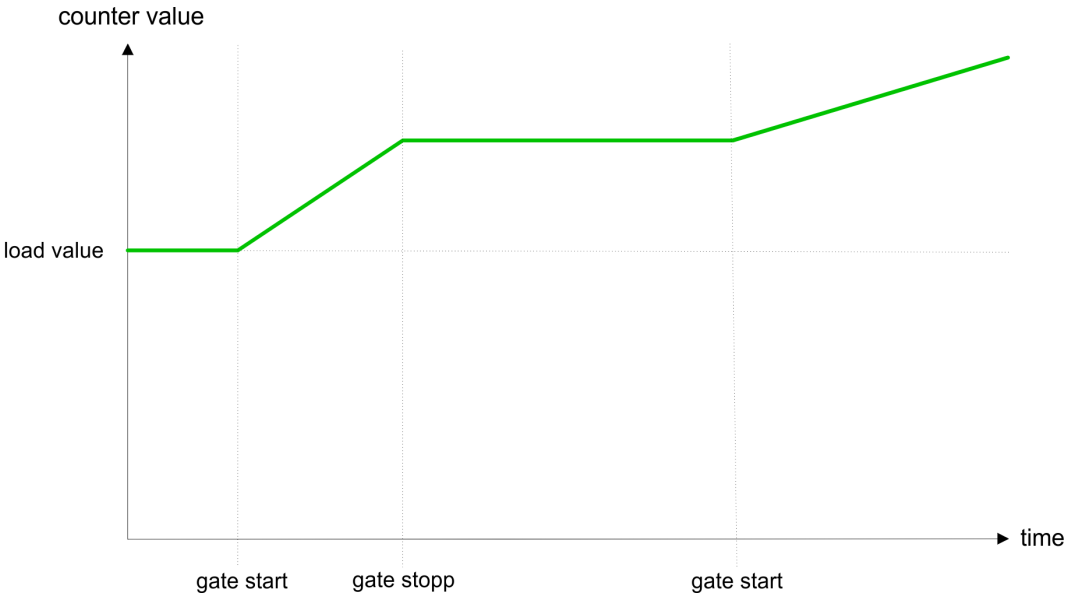
Gate function cancel and stop

The parameterization defines if the gate cancels or stops the counter process.

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



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



Gate control abort, interruption

How the CPU should react at opening of the SW gate may be set with the parameter *Gate function*. The usage of the HW gate may be determined by the parameter *Hardware gate*.

Gate control via *SW gate*, canceling (HW gate deactivated, gate function: Cancel count)

SW gate	HW gate	Reaction Counter
edge 0-1	deactivated	Restart with load value

Gate control via *SW gate*, stopping (HW gate deactivated, gate function: Stop count)

SW gate	HW gate	Reaction Counter
edge 0-1	deactivated	Continue

Gate control via *SW/HW gate*, canceling (HW gate activated, gate function: Cancel count)

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*, stopping (HW gate activated, gate function: Stop count)

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)	edge 0-1	1

Latch function

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

Comparator

In the CPU a comparison value may be stored that is assigned to the digital output, to the status bit "Status Comparator" STS_CMP and to the hardware interrupt. The digital output may be activated depending on the count value and comparison value. A comparison value may be entered by the parameter assignment screen form respectively by the request interface of the SFB 47.

Characteristics of the output

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

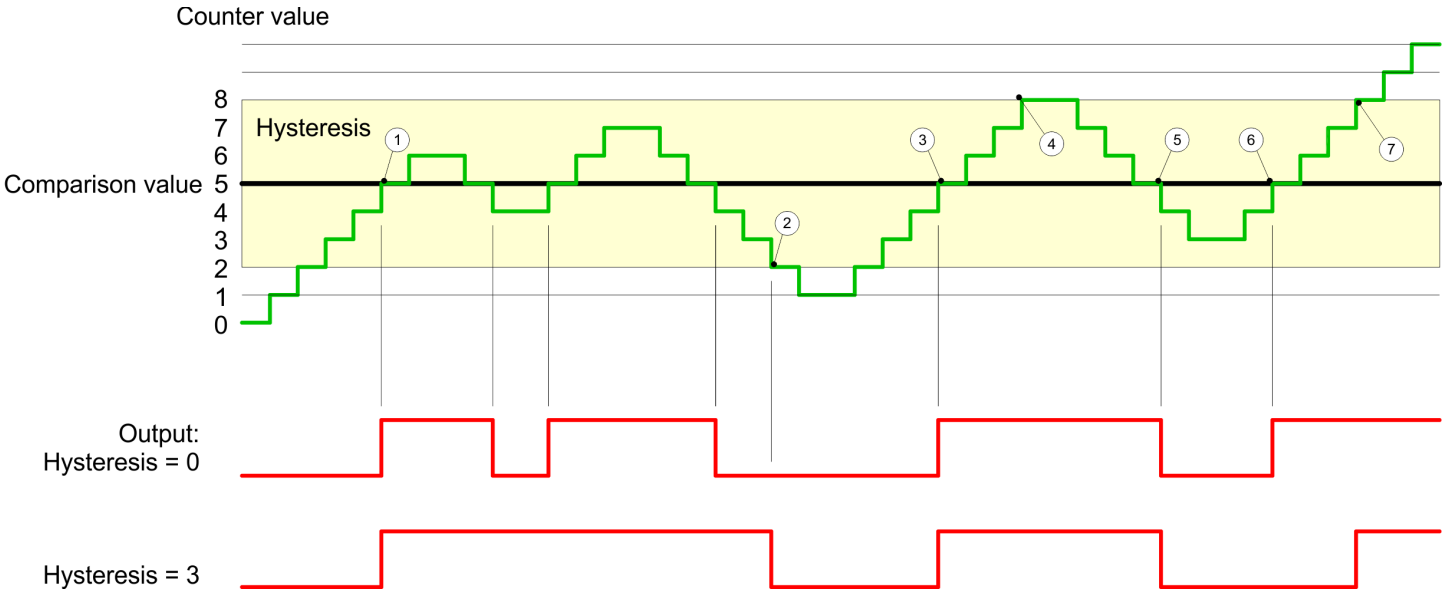
- No comparison
  The output is set as normal output. The SFB input parameter CTRL_DO is effect less. The status bits STS_DO and STS_CMP (Status comparator in the instance DB) remain reset.
- Count ≥ comparison value respectively Count ≤ comparison value
  The output remains set as long as the counter value is higher or equal comparison value respectively lower or equal comparison value. For this the control bit must be set. The comparison result is shown by the status bit STS_CMP. This status bit may only be reset if the comparison condition is no longer fulfilled.
- Pulse at comparison value
  When the counter reaches the comparison value the output is set for the parameterized pulse duration. If you have configured a main count direction the output is only activated when the comparison value is reached with the specified main count direction. For this the control bit CTRL_DO should be set first. The status of the digital output may be shown by the status bit ST_DO. The comparison result is shown by the status bit STS_CMP. This status bit may only be reset if the pulse duration has run off. comparison condition is no longer fulfilled. With pulse time = 0 the output is as set as the comparison condition is fulfilled.
  Pulse duration
  For adaptation to the used actors a pulse duration may be specified. The pulse duration defines how long the output should be set. It may be preset in steps of 2ms between 0 and 510ms. The pulse duration starts with the setting of the according digital output. The inaccuracy of the pulse duration is less than 1ms. There is no past triggering of the pulse duration when the comparison value has been left and reached again during pulse output. A change of the pulse period during runtime is not applied until the next pulse.

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:

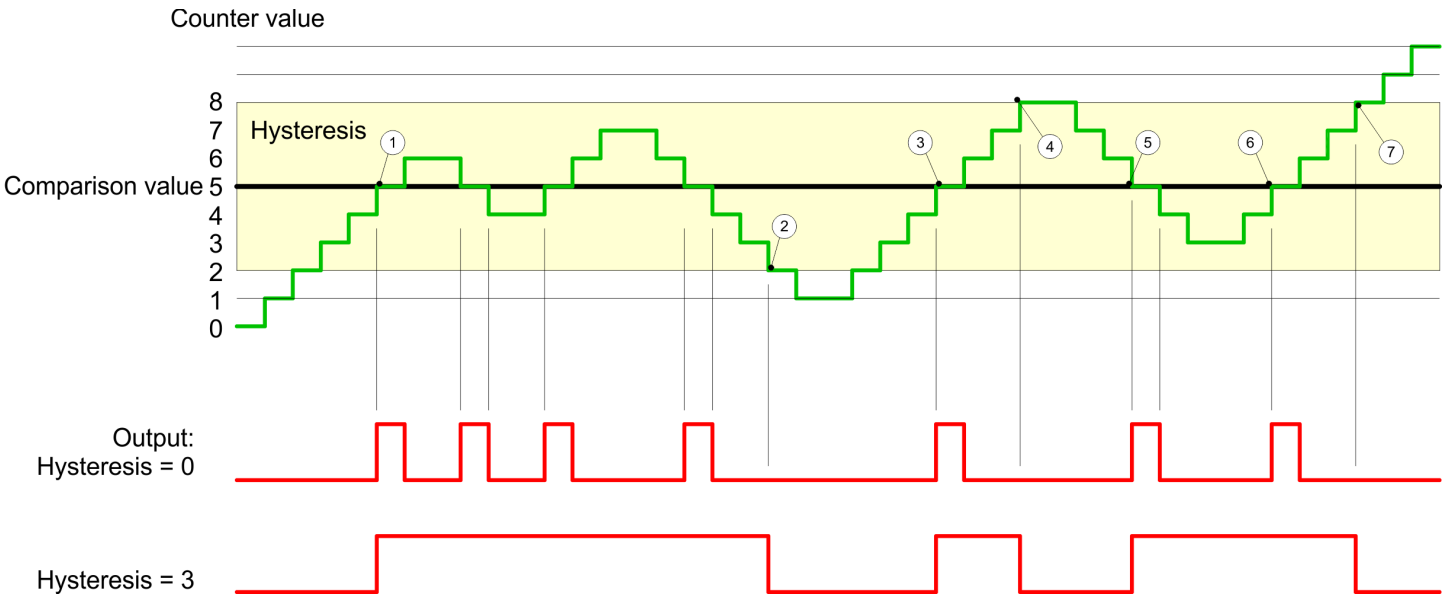
Effect at counter value ≥ comparison value



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

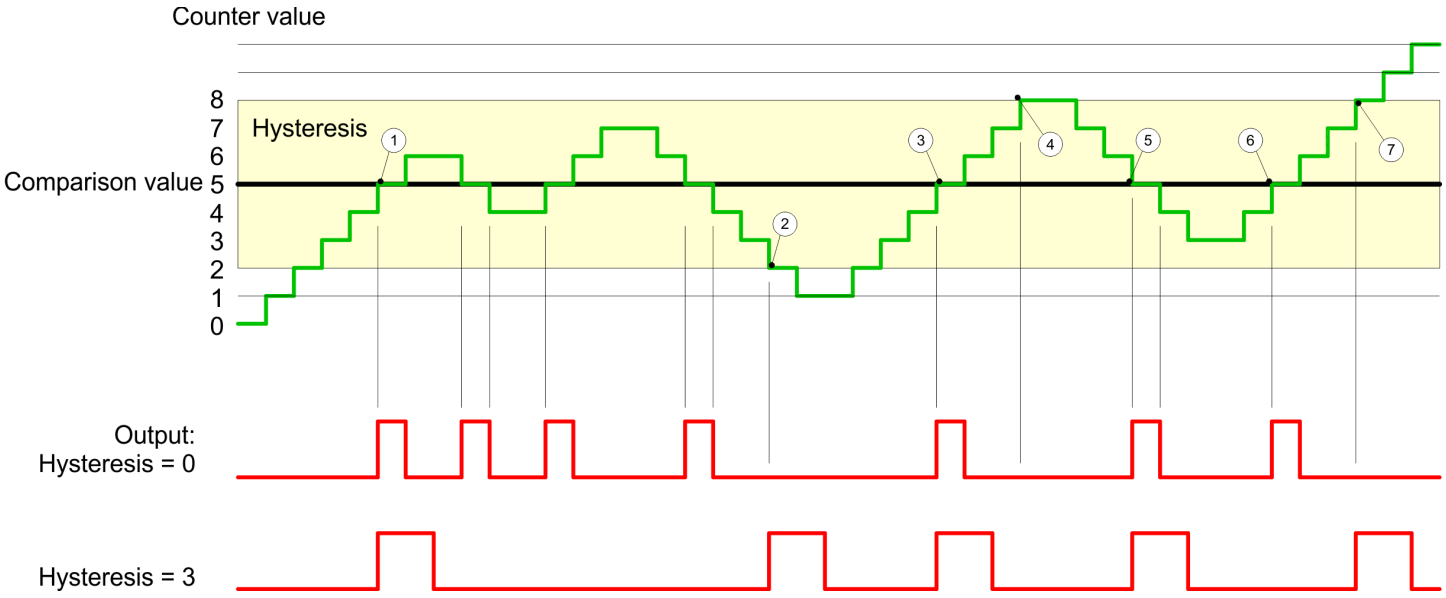


1 *Counter value* = *comparison value* → output is set and *hysteresis* activated
2 Leave *hysteresis* range → output is reset and *counter value* < *comparison value*

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



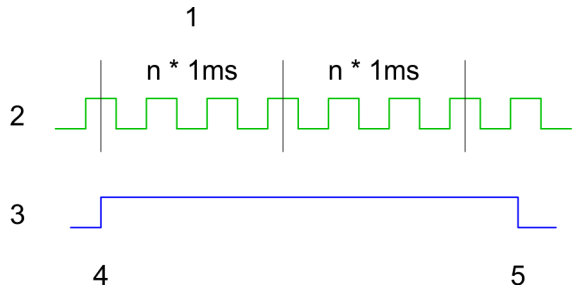
1 *Counter value* = *comparison value* → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is activated and the counting direction stored
2 Leaving the *hysteresis* range contrary to the stored counting direction → pulse of the parameterized *pulse duration* is put out, the *hysteresis* 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.

6.6 Frequency measurement

6.6.1 Overview

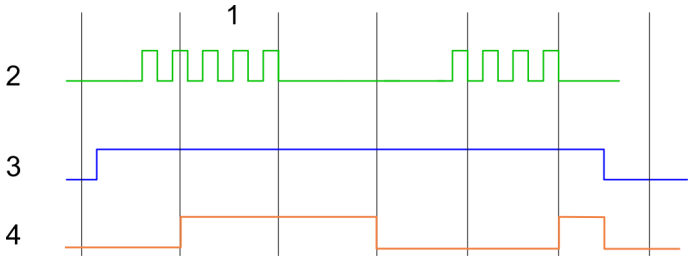
In this operating mode the CPU counts the incoming pulses during a specified integration time and outputs them as frequency value. You can set a value for the integration time between 10ms and 10000ms, in steps of 1 ms. You can set the integration time in the parameter assignment screen forms or you can edit them in the job interface of the SFB FREQUENC (SFB 48).



1 Integration time
2 Count pulse
3 Internal gate (SW gate)
4 Start of frequency measurement
5 Stop of frequency measurement

Measuring procedure

The measurement is carried out during the integration time and is updated after the integration time has expired If the period of the measured frequency exceeds the assigned integration time, this means there was no rising edge during the measurement, a value of 0 is returned. The calculated frequency value is supplied in "mHz" units. You can read out this value with the SFB parameter *MEAS_VAL*. The number of activated channels does not influence the max. frequency, which is defined in the technical data.



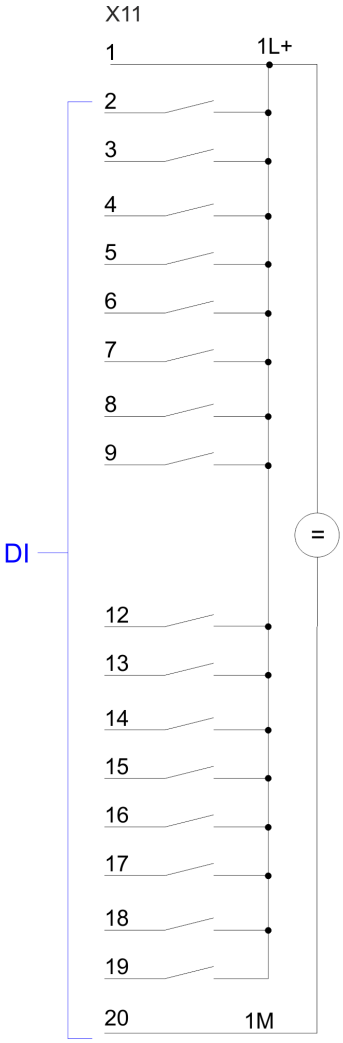
1 Integration time
2 Count pulse
3 Internal gate (SW gate)
4 Calculated frequency

During frequency measurement the count function at the same channel is deactivated.

6.6.2 Inputs for the frequency measurement

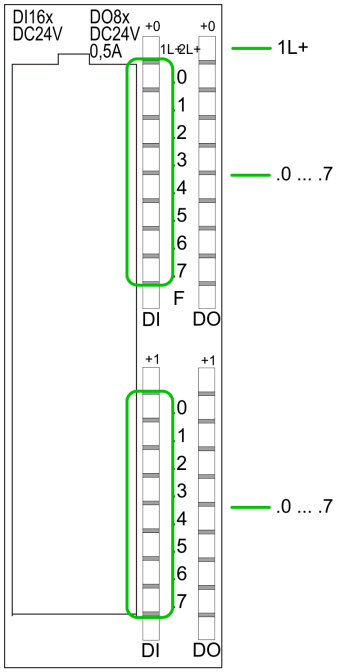
For frequency measurement, connect your signal to be measured at input B.

- Channel 0: Pin 3
- Channel 1: Pin 6



Pin assignment X11: DI

Pin	Assignment
1	1L+ Power supply +DC 24V
2	I+0.0 / Channel 0 (A) / Pulse
3	I+0.1 / Channel 0 (B) / Direction
4	I+0.2 / Channel 0 HW gate
5	I+0.3 / Channel 1 (A) / Pulse
6	I+0.4 / Channel 1 (B) / Direction
7	I+0.5 / Channel 1 HW gate
8	I+0.6
9	I+0.7
10	not used
11	not used
12	I+1.0
13	I+1.1
14	I+1.2
15	I+1.3
16	I+1.4 / Channel 0 Latch
17	I+1.5 / Channel 1 Latch
18	I+1.6
19	I+1.7
20	Ground 1M DI



Status indication X11: DI

- 1L+
  - LED (green)
    Supply voltage available for DI
- .0 7
  - LEDs (green)
    I+0.0 ... I+0.7
    I+1.0 ... I+1.7
    Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated

6.6.3 Parameterization

1. Start the Siemens SIMATIC Manager with your project and open the hardware configurator.
2. Place a profile rail.
3. Configure on slot 2 the Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6).
4. Open the dialog window "Properties" by a double click to the *Count* submodule of the CPU.
5. As soon as you select the operating mode "Frequency measurement" to the corresponding channel, a dialog window for the frequency measurement is created and displayed and filled with default parameters.
6. Execute the wished parameterization.
7. Store the project with *'Station ➔ Save and compile'*.
8. Transfer the project to the CPU.

Parameter overview

In the following the parameters are listed which may be used for frequency measurement configuration during hardware configuration. Parameters, which are not listed here, are ignored by the CPU.

- *General*
  Here the short description of the counter function may be found. At Comment information about the module such as purpose may be entered.
- *Addresses*
  Here the start address of the counter function is set.

- *Basic parameters*
  Here the interrupts, the counter component should trigger, may be selected. You have the following options:
  – None: There is no interrupt triggered.
  – Process: The counter component triggers a hardware interrupt.
  – Diagnostics and Process: With the CPU the diagnostic interrupt of the digital in-/output periphery is only supported in connection with "hardware interrupt lost".
- *Frequency measurement*
  The following parameters are relevant for frequency measurement. Parameters, which are not listed here, are ignored by the CPU.
  – *Integration time:*
    Integration time for frequency measurement
    Range of values: 10 ... 10000ms
  – *Hardware interrupt:*
    End of measurement (End of integration time)
    When activated, with each end of the integration time, a hardware interrupt is triggered.

6.6.4 SFB 48 - FREQUENC - Frequency measurement

Description

The SFB 48 is a specially developed block for compact CPUs for frequence measurement.

- The SFB FREQUENC should cyclically be called (e.g. OB 1) for controlling the frequency measurement.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.
- Among others the SFB 48 contains a request interface. Hereby you get read and write access to the registers of the frequency meter.
- So that a new job may be executed, the previous job must have be finished with *JOB_DONE* = TRUE.
- Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Writing accesses to outputs of the instance DB is not permissible.
- With the SFB FREQUENC (SFB 48) you have following functional options:
  – Start/Stop the frequency meter via software gate SW_GATE
  – Read the status bit
  – Read the evaluated frequency
  – Request to read/write internal registers of the frequency meter.

Parameters

Name	Declaration	Data type	Address (Inst.-DB)	Default value	Comment
LADDR	INPUT	WORD	0.0	300h	This parameter is not evaluated. Always the internal I/O periphery is addressed.
CHANNEL	INPUT	INT	2.0	0	Channel number
SW_GATE	INPUT	BOOL	4.0	FALSE	Enables the Software gate
JOB_REQ	INPUT	BOOL	4.3	FALSE	Initiates the job (edge 0-1)
JOB_ID	INPUT	WORD	6.0	0	Job ID
JOB_VAL	INPUT	DINT	8.0	0	Value for write jobs

Name	Declaration	Data type	Address (Inst.-DB)	Default value	Comment
STS_GATE	OUTPUT	BOOL	12.0	FALSE	Status of the internal gate
MEAS_VAL	OUTPUT	DINT	14.0	0	Evaluated frequency
JOB_DONE	OUTPUT	BOOL	22.0	TRUE	New job can be started.
JOB_ERR	OUTPUT	BOOL	22.1	FALSE	Job error
JOB_STAT	OUTPUT	WORD	24.0	0	Job error ID

Local data only in instance DB

Name	Data type	Address (Instance DB)	Default	Comment
JOB_OVAL	DINT	28.0	-	Output value for read request.

Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Writing accesses to outputs of the instance DB is not permissible.

Frequency meter request interface

To read/write the registers of the frequency meter the request interface of the SFB 48 may be used.

So that a new job may be executed, the previous job must have be finished with *JOB_DONE* = TRUE.

Proceeding

The deployment of the request interface takes place at the following sequence:

→ Edit the following input parameters:

Name	Data type	Address (DB)	Default	Comment
JOB_REQ	BOOL	4.3	FALSE	Initiates the job (edges 0-1)
JOB_ID	WORD	6.0	0	Job ID: 00h Job without function 04h Writes the integration time 84h Read the integration time
JOB_VAL	DINT	8.0	0	Value for write jobs. Permitted value for integration time: 10 ... 10000ms

→ Call the SFB. The job is processed immediately. *JOB_DONE* only applies to SFB run with the result FALSE. *JOB_ERR* = TRUE if an error occurred. Details on the error cause are indicated at *JOB_STAT*.

Name	Data type	Address (DB)	Default	Comment
JOB_DONE	BOOL	22.0	TRUE	New job can be started
JOB_ERR	BOOL	22.1	FALSE	Job error
JOB_STAT	WORD	24.0	0000h	Job error ID 0000h No error 0221h Integration time too low 0222h Integration time too high 02FFh Invalid job ID 8001h Parameter error 8009h Channel no. not valid

1. A new job may be started with *JOB_DONE* = TRUE.
2. A value to be read of a read job may be found in *JOB_OVAL* in the instance DB at address 28.

Channel no. not valid

(8009h and Parameter error 8001h)

If you have preset a CHANNEL number greater than 3, the error "Channel no. not valid " (8009h) is reported. if you have preset a CHANNEL number greater than the maximum channel number of the CPU, "Parameter error" (8001h) is reported.

Controlling frequency meter

The frequency meter is controlled by the internal gate (I gate). The I gate is identical to the software gate (SW gate).

SW gate:

open (activate): In the user program by setting *SW_GATE* of SFB 48

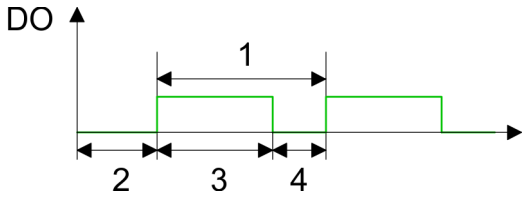
close (deactivate): In the user program by resetting *SW_GATE* of SFB 48

6.7 Pulse width modulation - PWM

6.7.1 Overview

PWM

With the pulse width modulation (PWM) by presetting of time parameters the CPU evaluates a pulse sequence with according pulse/break ratio and issues it via the according output channel.



1 Period duration
2 ON delay
3 Pulse duration
4 Pulse pause

During pulse width modulation the count function at the same channel is deactivated.

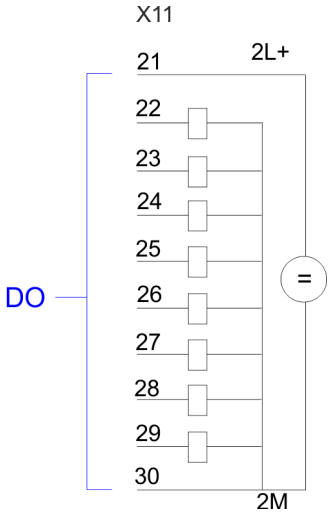
PWM Outputs

For pulse width modulation connect your actuators to the following pins:

Channel 0: Pin 22

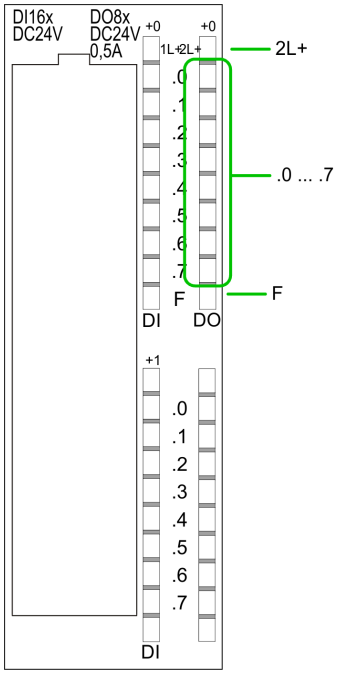
Channel 1: Pin 23

Connect the common ground to pin 30.



Pin assignment X11: DO

Pin	Assignment
21	2L+ Power supply +DC 24V
22	O+0.0 / Channel 0 Output
23	O+0.1 / Channel 1 Output
24	Q+0.2
25	Q+0.3
26	Q+0.4
27	Q+0.5
28	Q+0.6
29	Q+0.7
30	Ground 2M DO
31 ... 40	not used



Status indication X11: DO

- 2L+
  - LED (green)
    Supply voltage available for DO
- .0 7
  - LEDs (green)
    Q+0.0 ... Q+0.7
    The according LED is on at active output
- F
  - LED (red)
    Overload or short circuit error

6.7.2 Parameterization

1. Start the Siemens SIMATIC Manager with your project and open the hardware configurator.
2. Place a profile rail.
3. Configure at slot 2 the Siemens CPU 312C (6ES7 312-5BE03-0AB0/V2.6).
4. Open the dialog window "Properties" by a double click to the *Count* submodule of the CPU.
5. As soon as you select the operating mode "Pulse width modulation" to the corresponding channel, a dialog window for the pulse width modulation is created and displayed and filled with default parameters.
6. Execute the wished parameterization.
7. Save the project with *'Station ➔ Save and compile'*.
8. Transfer the project to the CPU.

6.7.2.1 Parameter overview

In the following the parameters are listed which may be used for pulse width modulation configuration during hardware configuration. Parameters, which are not listed here, are ignored by the CPU.

General

Here the short description of the counter function may be found. At *Comment* information about the module such as purpose may be entered.

Addresses

Here the start address of the counter function is set.

Basic parameters

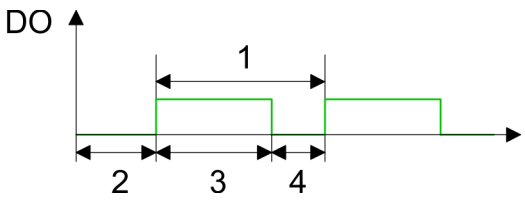
Here the interrupts, the counter function should trigger, may be selected. You have the following options:

- None: There is no interrupt triggered.
- Process: The counter component triggers a hardware interrupt.
- Diagnostics and Process: With the CPU the diagnostic interrupt of the digital in-/output periphery is only supported in connection with "hardware interrupt lost".

There are no interrupts with the pulse width function.

Pulse width modulation

The following parameters are relevant for pulse width modulation. Parameters, which are not listed here, are ignored by the CPU.



1 Period duration
2 ON delay
3 Pulse duration
4 Pulse pause

Output format

- Here select the range of values of the output value. With this the CPU calculates the pulse duration:

Output format	Range of values	Pulse duration
Per mil (Default)	0 ... 1000	(Output value / 1000) x Period duration
S7 analog value	0 ... 27648	(Output value / 27648) x Period duration

Time base

- Set the time base, which is valid for resolution and the range of values of period duration, minimum pulse duration and on-delay.
- If you have checked the "1ms" button times with a resolution of 1ms can be set.
- If you have checked the "0.1ms" button times with a resolution of 0.1ms can be set.
- Default: "0.1 ms"

On-delay

- Enter a value for the delay time between the start of an output sequence and the output of the pulse. The pulse sequence is output on the output channel after the on-delay has expired.
- Range of values: 0 ... 65535ms respectively 0 ... 6553.5ms

Period (Period duration)

- With the period duration the length of the output sequence (pulse duration/pulse pause) is defined
- Range of values: 1 ... 65535ms respectively 0.4 ... 6553.5ms
- Default value: 20000

Minimum pulse duration

- You can set a minimum pulse time for the attenuation of short output pulses/pauses. This suppresses all pulses or pauses shorter than the minimum pulse time. Thus you may filter very small pulses (spikes), which are not noted from the periphery anymore.
- Range of values:
  0 ... Period duration/2 * 1ms respectively
  2 ... Period duration/2 * 0.1ms
- Default: 2

6.7.3 SFB 49 - PULSE - Pulse width modulation

Description

The SFB 49 is a specially developed block for compact CPUs for *PWM* and *pulse train* output. With the SFB PULSE (SFB 49) the following functionalities are available:

- PWM (**P**uls**w**idth**m**odulation)
  - Start/Stop via software gate *SW_EN*
  - Enabling/controlling of the PWM output
  - Read status bits
  - Request to read/write the internal PWM registers
- Configurable pulse train output with a maximum of 2 drive jobs
  - Start/Stop via software gate *SW_EN*
  - Enabling/controlling of the pulse train output
  - Read status bits
  - Request to read/write the internal pulse train registers
- Configurable time base (1µs ... 1ms)

When using the block, the following must be observed:

- The SFB is cyclically to be called with the corresponding instance DB e.g. in OB 1.
- You have read and write access to the corresponding registers via the SFB 49 job interface.
- Per channel you may call the SFB in each case with the same instance DB. Write accesses to outputs of the instance DB is not permissible.
- So that a new job may be executed, the previous job must have be finished with *JOB_DONE* = TRUE.
- The switching between the modes takes place by the presetting of the pulse number (JOB_ID = 08h/09h). As soon as you specify a pulse number > 0, you switch to the pulse train mode, otherwise PWM is active.

Please note that some functions of this block are not available in all CPUs. If you call a functionality that is not supported, you receive the error message 04FFh 'Order no. invalid' as Return value. More about the supported functions can also be found in the 'Properties' of your CPU.

Parameter

Parameter	Declaration	Data type	Address (Inst.-DB)	Default Value	Comment
LADDR	INPUT	WORD	0.0	300h	This parameter is not evaluated. Always the internal I/O periphery is addressed.
CHANNEL	INPUT	INT	2.0	0	Channel number
SW_EN	INPUT	BOOL	4.0	FALSE	Enable software gate

Parameter	Declaration	Data type	Address (Inst.-DB)	Default Value	Comment
MAN_DO	INPUT	BOOL	4.1	FALSE	This parameter is not evaluated.
SET_DO	INPUT	BOOL	4.2	FALSE	This parameter is not evaluated.
OUTP_VAL	INPUT	INT	6.0	0	Output value ‘OUTP_VAL’ on page 116
JOB_REQ	INPUT	BOOL	8.0	FALSE	Job trigger (edge 0-1)
JOB_ID	INPUT	WORD	10.0	0	Job number ‘JOB_ID’ on page 116
JOB_VAL	INPUT	DINT	12.0	0	Value for write jobs
STS_EN	OUTPUT	BOOL	16.0	FALSE	Status internal gate
STS_STRT	OUTPUT	BOOL	16.1	FALSE	This parameter is reserved.
STS_DO	OUTPUT	BOOL	16.2	FALSE	This parameter is reserved.
JOB_DONE	OUTPUT	BOOL	16.3	TRUE	Status parameter ■ 0: The job has not started or is still running. ■ 1: Job has been executed. A new job can be started.
JOB_ERR	OUTPUT	BOOL	16.4	FALSE	Status parameter ■ 0: no error ■ 1: Error (see *JOB_STAT*)
JOB_STAT	OUTPUT	WORD	18.0	0	‘Return value JOB_STAT’ on page 121

OUTP_VAL

The *‘output format’* for PWM and pulse train can be set via the hardware configuration. Depending on the output format, there are the following range of values for the *output value*:

- Output in ‰
  - Range of values: 0 ... 1000
  - *Pulse duration* = (*OUTP_VAL* / 1000) x *period duration*
- Output format: S7 analog value
  - *Pulse duration* = (*OUTP_VAL* / 27648) x *period duration*
  - Range of values: 0 ... 27648

JOB_ID

Job number

- 00h: Job without function
- 01h: Write *period duration* for PWM and. 1 pulse train job
  Range of values in dependence of the time base:
  - 1ms: 1 ... 87
  - 0.1ms: 1 ... 870:
  - 10µs 2 ... 8700
  - 1µs: 20 ... 65535

- 02h: Write *on-delay*
  Range of values in dependence of the time base:
  – 1ms: 0 ... 65535
  – 0.1ms: 0 ... 65535
  – 10µs 0 ... 65535
  – 1µs: 0 ... 65535
- 04h: Write *minimum pulse duration*
  Range of values in dependence of the time base:
  – 1ms: 0 ... Period duration/2
  – 0.1ms: 0 ... Period duration/2
  – 10µs 0 ... Period duration/2
  – 1µs: 5 ... Period duration/2
- 08h: Write *number of pulses* for the 1. pulse train job
  Range of values:
  – 0 ... 8.388.607
- 09h: Write *number of pulses* for the 2. pulse train job
  Range of values:
  – 0 ... 8.388.607
- 0Ah: *Period duration* for writing 2. pulse train job
- 0Bh: Write *time base*
  – 00h: 0.1ms
  – 01h: 1ms
  – 02h: 1µs:
  – 03h: 10µs
- 0Ch 2. Attach pulse train job to the 1. pulse train job
  – With this job number, the duty factor for the 2. pulse train job is additionally to be specified via *OUTP_VAL*.
- 81h: Read *period duration* of PWM and 1. pulse train job
- 82h: Read *on-delay*
- 84h: Read *minimum pulse duration*
- 88h: Read *number of pulses* of the 1. pulse train job
- 89h: Read *number of pulses* of the 2. pulse train job
- 8Ah: Read *period duration* of the 2. pulse train job
- 8Bh Read *time base*
  – 00h: 0.1ms
  – 01h: 1ms
  – 02h: 1µs:
  – 03h: 10µs

JOB_VAL

Value for write jobs, which range of values depends on the according job:

-2147483648 (-2^{31}) ... +2147483647 ($2^{31}-1$)

Local data only in instance DB

Name	Data type	Address (Instance DB)	Default	Comment
JOB_OVAL	DINT	20.0	-	Output values for read jobs

Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here. Write accesses to outputs of the instance DB is not permissible.

Request interface

- To read/write the registers the request interface of the SFB 49 may be used.
- So that a new job may be executed, the previous job must have be finished with *JOB_DONE* = TRUE.
- With an edge 0-1 at *JOB_REQ*, you can always transfer a job, regardless of the state of *SW_EN* and *STS_EN*.
- Changes of the *period duration* and the *minimum pulse duration* will immediately take effect.
- Changes of the *on-delay* take effect with the next edge 0-1 of *SW_EN*.
- A running PWM output is not affected by setting pulse train specific values such as *pulse number* and *period duration* for the 2. pulse train job.

Controlling the output

Controlling the PWM output

The request interface is used according to the following sequence:

1. Call the SFB 49:

- *SW_EN* = FALSE
- *JOB_VAL* = Enter a value for the *period duration* here
- *JOB_ID* = 01h: Write *period duration* for PWM output.
- *JOB_REQ* = TRUE

⇨
- From*JOB_VAL* the period duration is transmitted to the PWM output.
- *JOB_DONE* is FALSE during the SFB run.
- On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

2. Call the SFB 49:

- *SW_EN* = FALSE
- *JOB_VAL* = Enter a value for the *on-delay* here
- *JOB_ID* = 02h: Write *on-delay* for PWM output.
- *JOB_REQ* = TRUE

⇨
- From *JOB_VAL* the *on-delay* is transmitted to the PWM output.
- *JOB_DONE* is FALSE during the SFB run.
- On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

3. Call the SFB 49:

- *SW_EN* = FALSE
- *JOB_VAL* = Enter a value for the *minimum pulse duration* here
- *JOB_ID* = 04h: Write *minimum pulse duration* for PWM output.
- *JOB_REQ* = TRUE

⇨
- From *JOB_VAL* the *minimum pulse duration* is transmitted to the PWM output.
- *JOB_DONE* is FALSE during the SFB run.
- On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

4. Call the SFB 49:

- *SW_EN* = TRUE (edge 0-1)
- *OUTP_VAL*: Specify a duty factor.

⇨
- The PWM output is started
- *STS_EN* goes to TRUE and remains in this state until SFB 49 is called with *SW_EN* = FALSE.
- On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

5. Call the SFB 49 cyclically:

- *SW_EN* = FALSE
- Via *STS_EN* you get the current status of the PWM output. With *OUTP_VAL* you can always change the duty factor.

6. As soon as *JOB_DONE* returns TRUE, you can change the PWM parameters by repeating the steps 1 to 5.

If values are changed during PWM output, the new values are only output with the beginning of a new period. A started period is always finished!

7. By resetting of *SW_EN* (*SW_EN* = FALSE) the output is immediately stopped.

8. With reading jobs, you can find the values to be read in the parameter *JOB_OVAL* in the instance DB at address 20.

Controlling the pulse train output

The request interface is used according to the following sequence:

1. Call the SFB 49:

- *SW_EN* = FALSE
- *JOB_VAL* = Enter a value for the *number of pulses* here.
- *JOB_ID* = 08h: Write *number of pulses* for the 1. pulse train job.
- *JOB_REQ* = TRUE

⇨
- From *JOB_VAL* the *number of pulses* for the 1. pulse train job is transmitted.
- *JOB_DONE* is FALSE during the SFB run.
- On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

2. Call the SFB 49:
   - *SW_EN* = FALSE
   - *JOB_VAL* = Enter a value for the *period duration* here.
   - *JOB_ID* = 01h: Write *period duration* for the 1. pulse train job.
   - *JOB_REQ* = TRUE

   ⇨
   - From *JOB_VAL* the *period duration* for the 1. pulse train job is transmitted.
   - *JOB_DONE* is FALSE during the SFB run.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

3. Optional for the 2. pulse train job: Call the SFB 49:
   - *SW_EN* = FALSE
   - *JOB_VAL* = Enter a value for the *number of pulses* here.
   - *JOB_ID* = 09h: Write *number of pulses* for the 2. pulse train job.
   - *JOB_REQ* = TRUE

   ⇨
   - The *number of pulses* for the 2. pulse train job is transmitted.
   - *JOB_DONE* is FALSE during the SFB run.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

4. Optional for the 2. pulse train job: Call the SFB 49:
   - *SW_EN* = FALSE
   - *JOB_VAL* = Enter a value for the *period duration* here.
   - *JOB_ID* = 0Ah: Write *period duration* for the 2. pulse train job.
   - *JOB_REQ* = TRUE

   ⇨
   - From *JOB_VAL* the *period duration* for the 2. pulse train job is transferred.
   - *JOB_DONE* is FALSE during the SFB run.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

5. Call the SFB 49:
   - *SW_EN* = TRUE (edge 0-1)
   - *OUTP_VAL*: Enter the duty factor such as 50%.

   ⇨
   - The 1. pulse train job is started and then if present the 2. pulse train job.
   - Via *STS_EN* you get the current status of the pulse train output. As long as the required number of pulses is output, *STS_EN* returns TRUE. *STS_EN* returns FALSE if either the requested number of pulses has been output or output with *SW_EN* = FALSE was terminated early.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

6. Call the SFB 49 cyclically:
   - *SW_EN* = FALSE
   - Via *STS_EN* you get the current status of the pulse train output.

7. As soon as *JOB_DONE* returns TRUE, you can transfer additional pulse train jobs by repeating the steps 1 to 6.

8. By resetting of *SW_EN* (*SW_EN* = FALSE) the output is immediately stopped.

9. With reading jobs, you can find the values to be read in the parameter *JOB_OVAL* in the instance DB at address 20.

Extend a running pulse train job

As long as only one pulse train job is defined and currently being processed, there is the possibility to attach a 2. pulse train job to the 1. pulse train job.

1. Call the SFB 49:
   - *SW_EN* = FALSE
   - *JOB_VAL* = Enter a value for the *number of pulses* here.
   - *JOB_ID* = 09h: Write *number of pulses* for the 2. pulse train job.
   - *JOB_REQ* = TRUE

   ⇨
   - From *JOB_VAL* the *number of pulses* for the 2. pulse train job is transmitted.
   - *JOB_DONE* is FALSE during the SFB run.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

2. Call the SFB 49:
   - *SW_EN* = FALSE
   - *JOB_VAL* = Enter a value for the *period duration* here.
   - *JOB_ID* = 0Ah: Write *period duration* for the 2. pulse train job.
   - *JOB_REQ* = TRUE

   ⇨
   - From *JOB_VAL* the *period duration* for the 2. pulse train job is transferred.
   - *JOB_DONE* is FALSE during the SFB run.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

3. Call the SFB 49:
   - *SW_EN* = TRUE (edge 0-1)
   - *JOB_ID* = 0Ch: Attach 2. pulse train job to the 1. pulse train job.
   - *OUTP_VAL*: Enter the duty factor such as 50%.

   ⇨
   - As long as the 1. pulse train job is still running, the 2. pulse train job is attached. Otherwise you receive the error message 0461h as *return value*.
   - Via *STS_EN* you get the current status of the pulse train output. As long as the required number of pulses is output, *STS_EN* returns TRUE. *STS_EN* returns FALSE if either the requested number of pulses has been output or output with *SW_EN* = FALSE was terminated early.
   - On error *JOB_ERR* = TRUE and the cause of the error is returned in *JOB_STAT*

Please note that a maximum of 2 pulse train jobs can be executed directly after another!

Return value *JOB_STAT*

The JOB_STAT return value gives you detailed information in the event of an error.

Value	Description
0000h	no error
0411h	*Period duration* too small
0412h	*Period duration* too big
0421h	*On-delay* too small
0422h	*On-delay* too big

Value	Description
0431h	*Minimum pulse duration* too small
0432h	*Minimum pulse duration* too big
0441h	*Number of pulses* too small
0442h	*Number of pulses* too big
0451h	Invalid *time base*
0461h	Pulse train job could not be attached
04FFh	Job number not valid You receive this error message e.g. if the corresponding functionality is not supported by your CPU.
8001h	Parametrization error You will get a parametrization error (8001h), if you have transmitted a channel number with *CHANNEL*, which is bigger than the max. available number of channels of the CPU.
8009h	Channel no. not valid You will get the return value channel no. not valid (8009h), if you have transmitted a channel number with *CHANNEL*, which is bigger than 3.

6.8 Diagnostic and interrupt

Overview

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

Counter function

- Edge at a digital input
- Opening the HW gate (at opened SW gate)
- Closing the HW gate (at opened SW gate)
- Reaching the comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at underrun lower counter limit

Frequency measurement

- Edge at a digital input
- End of measurement (end of the integration time)

Pulse width modulation

- There is no process interrupt available

6.8.1 Process interrupt

Activation

The activation of the process interrupt occurs only if at the same time the counter output is activated and "Diagnostics+Process" is set in the basic parameters.

Process interrupt

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*.

Local double word 8 of OB 40 at *counter function*

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 7 ... 2: reserved
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 7 ... 0: reserved

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

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 7 ... 2: reserved
10	■ Bit 0: End of measurement channel 0 (end of the integration time) ■ Bit 3 ... 1: reserved ■ Bit 4: End of measurement channel 1 (end of the integration time) ■ Bit 7 ... 5: reserved
11	■ Bit 7 ... 0: reserved

6.8.2 Diagnostic interrupt

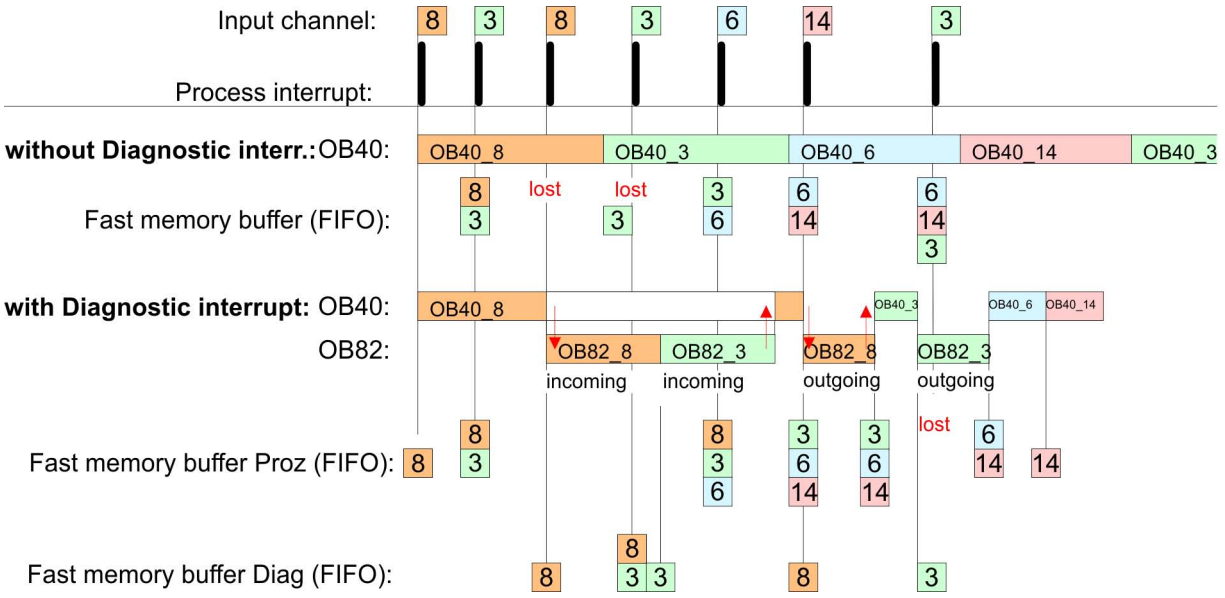
Activation

Please consider that diagnostic interrupts are enabled only if you have selected one of the technology functions (counting, frequency measurement, PWM) and set "Diagnostics+Process interrupt" in the basic parameters.

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 7 ... 4: 0 (fix)
1	■ Bit 3 ... 0: Module class – 0101b: Analog – 1111b: Digital ■ Bit 4: Channel information present ■ Bit 7 ... 5: 0 (fix)
2	■ Bit 7 ... 0: 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 7 ... 4: 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)

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

Diagnostic record set 1 at *counter function*

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

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 125*
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
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+1.1) ■ Bit 3: reserved ■ Bit 4: Error in channel group 4 (counter 0) ■ Bit 5: Error in channel group 5 (counter 1) ■ Bit 6: reserved ■ Bit 7: reserved
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 7 ... 4: reserved
11	■ Bit 7 ... 0: reserved

Byte	Bit 7...0
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)
14	reserved
15	reserved

Diagnostic Record set 1 at *frequency measurement*

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

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 125*
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
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+1.1) ■ Bit 3: reserved ■ Bit 4: Error in channel group 4 (Frequency meter 0) ■ Bit 5: Error in channel group 5 (Frequency meter 1) ■ Bit 6: reserved ■ Bit 7: reserved

Byte	Bit 7...0
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 7 ... 4: reserved
11	■ Bit 7 ... 0: reserved
12	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 0 (End of integration time) ■ Bit 7 ... 1: 0 (fix)
13	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 1 (End of integration time) ■ Bit 7 ... 1: 0 (fix)
14	reserved
15	reserved

Diagnostic record set 1 at *pulse width modulation*

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 125*
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
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+1.1) ■ Bit 7 ... 3: reserved
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 7 ... 4: reserved
11 ... 15	■ Bit 7... 0: reserved

7 Deployment PtP communication

7.1 Fast introduction

General

The CPU has a RS485 interface X3, which is fix set to PtP communication (**p**oint **t**o **p**oint).

- PtP functionality
  - Using the PtP functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.

Protocols

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

Parametrization

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

Communication

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

Overview FCs/SFCs for serial communication

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

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

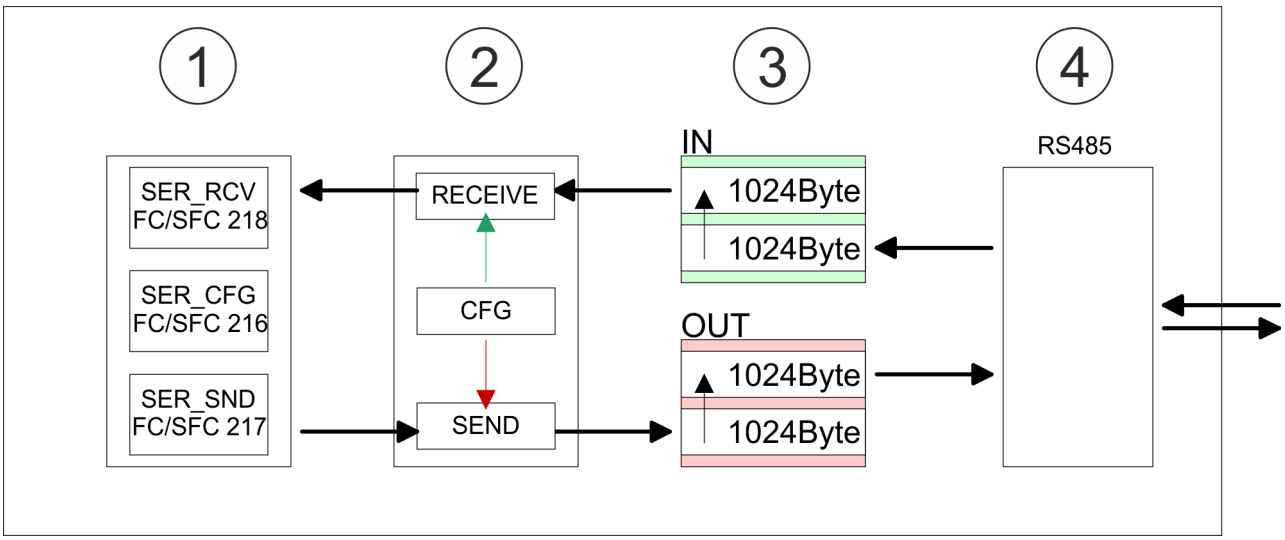
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

7.2 Principle of the data transfer

RS485 PtP communication

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

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



1 Program
2 Protocol
3 FIFO buffer
4 Interface

7.3 Deployment of RS485 interface for PtP

Overview

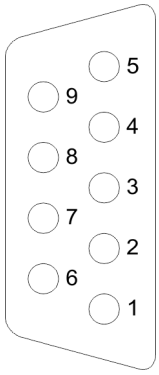
The RS485 interface from the CPU is fix set to PtP communication. Parameterization and communication happens by means of SFCs.

Properties RS485

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

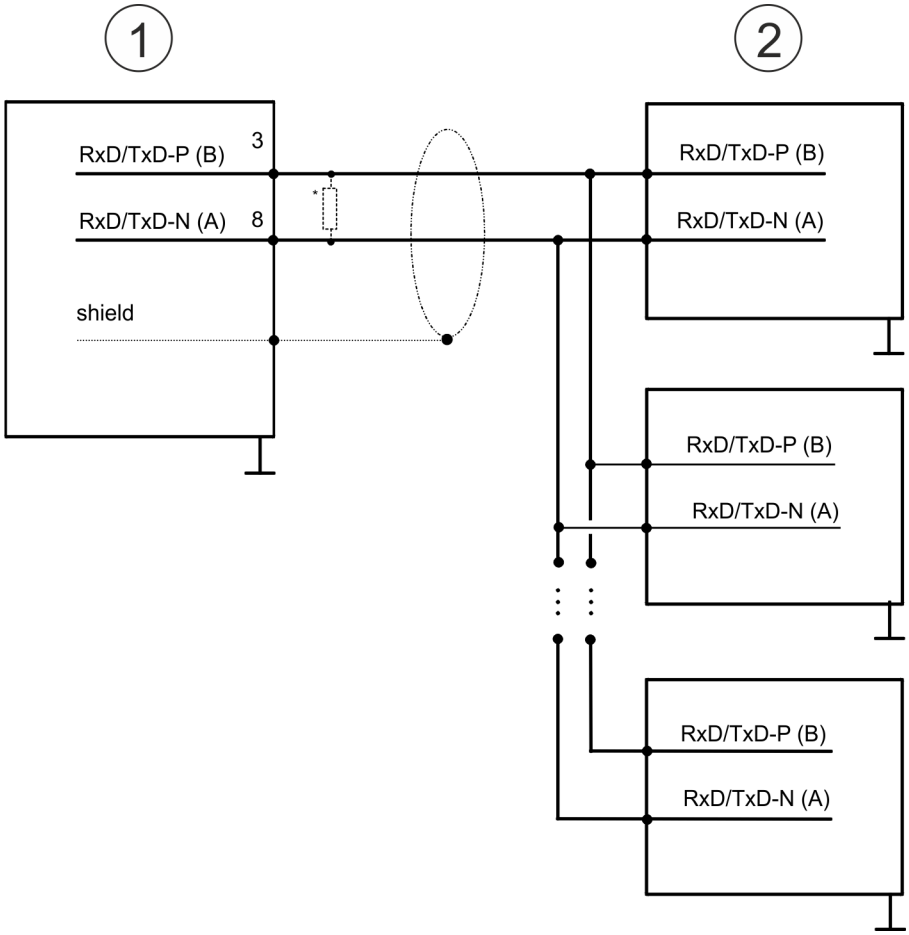
RS485

9pin SubD jack



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

Connection



1 RS485 interface
2 Periphery

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

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 "SPEED7 Operation List" from VIPA.

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 parametrized receive data block. If a telegram is spread over various cycles, the data is overwritten. There is no reception acknowledgement. The communication procedure has to be controlled by the concerning user application. For this you can use the FB 1 - Receive_ASCII.

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

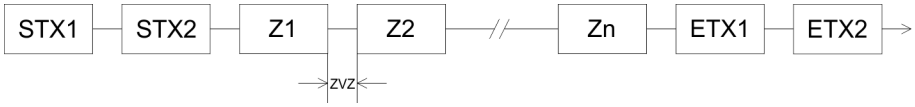
STX/ETX

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

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

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

Message structure:



3964

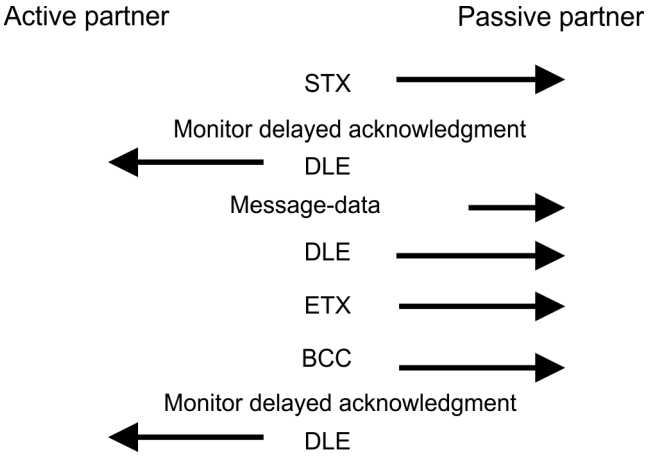
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

The procedure employs the following control characters:

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

You may transfer a maximum of 255byte per message.

Procedure



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

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

USS

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

The following features characterize the USS protocol:

- Multi point connection
- Master slave access procedure
- Single master system

- Max. 32 participants
- Simple and secure telegram frame

It is essential:

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

The telegrams for send and receive have the following structure:

Master slave telegram

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

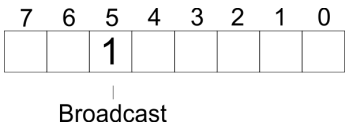
Slave master telegram

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

with

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

Broadcast with set bit 5 in ADR byte



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

Modbus

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

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

Telegram structure

Start sign	Slave address	Function Code	Data	Flow control	End sign

Broadcast with slave address = 0

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

ASCII, RTU mode

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

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

Supported Modbus protocols

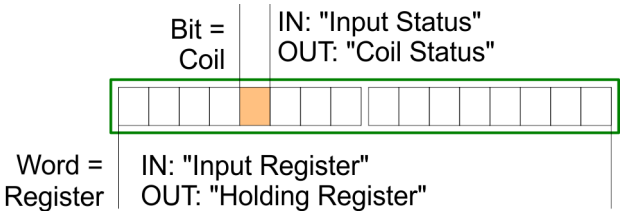
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

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.

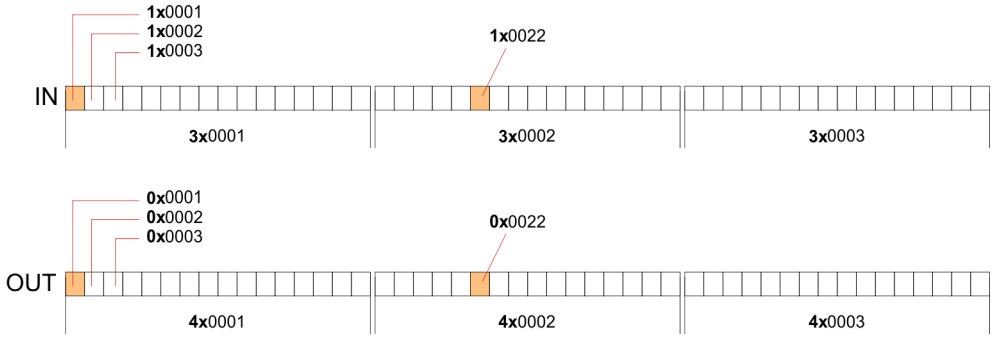
For the CPs from VIPA is not differentiating digital and analog data, the following assignment is valid:

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.

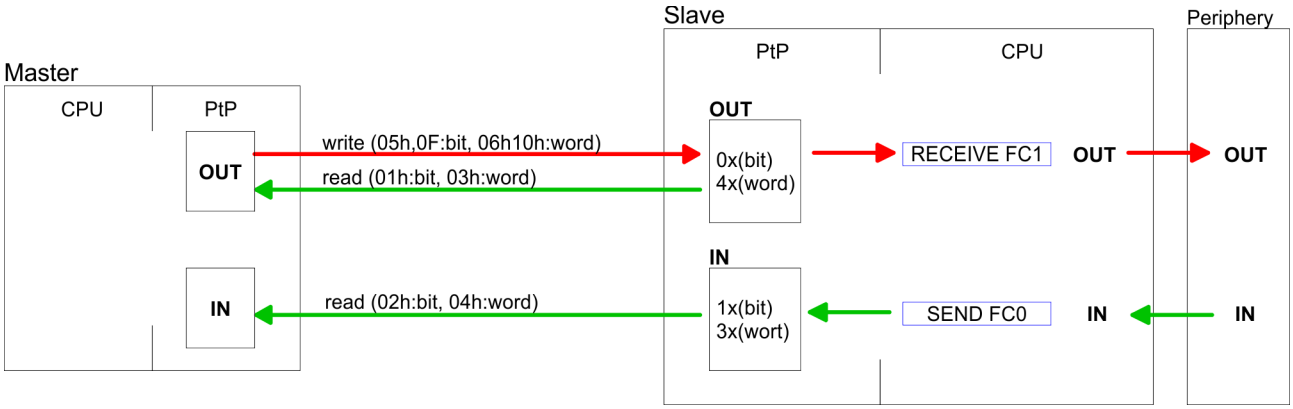
Overview

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
05h	Write 1 bit	Write 1 bit to master output area 0x
06h	Write 1 word	Write 1 word to master output area 4x
0Fh	Write n bits	Write n bits to master output area 0x
10h	Write n words	Write n words to master output area 4x

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

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



Respond of the slave

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

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

Byte sequence in a word

1 word	
High-byte	Low-byte

Check sum CRC, RTU, LRC

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

Read n bits 01h, 02h

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

Command telegram

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

Respond telegram

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

Read n words 03h, 04h

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

Command telegram

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

Respond telegram

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

Write 1 bit 05h

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

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

"Status bit" = 0000h → Bit = 0

"Status bit" = FF00h → Bit = 1

Command telegram

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

Respond telegram

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

Write 1 word 06h

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

Command telegram

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

Respond telegram

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

Write n bits 0Fh

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

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

Command telegram

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

Respond telegram

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

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

Command telegram

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

Respond telegram

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

7.8 Modbus - Example communication

Overview

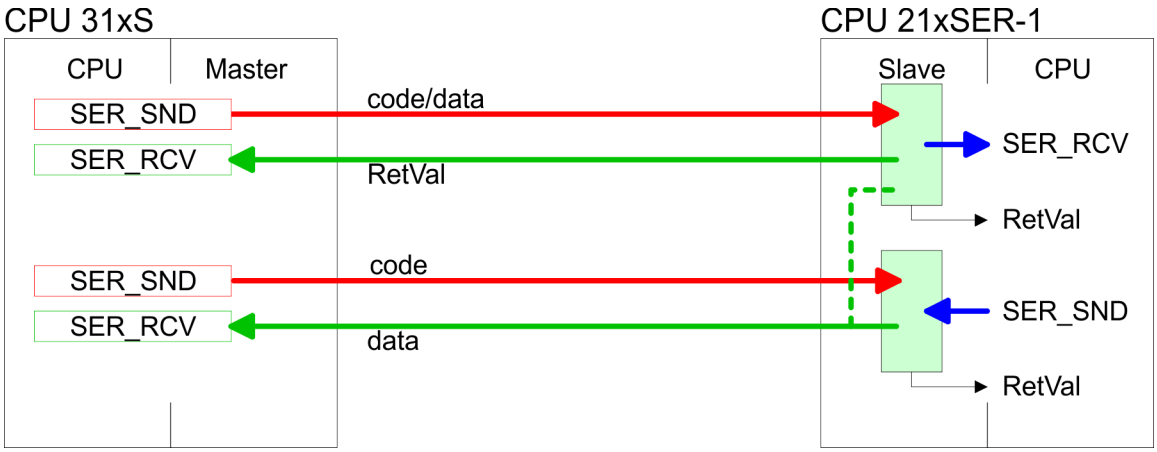
The example establishes a communication between a master and a slave via Modbus. The following combination options are shown:

- 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:




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

8 WinPLC7

8.1 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

8.2 Installation

Precondition

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

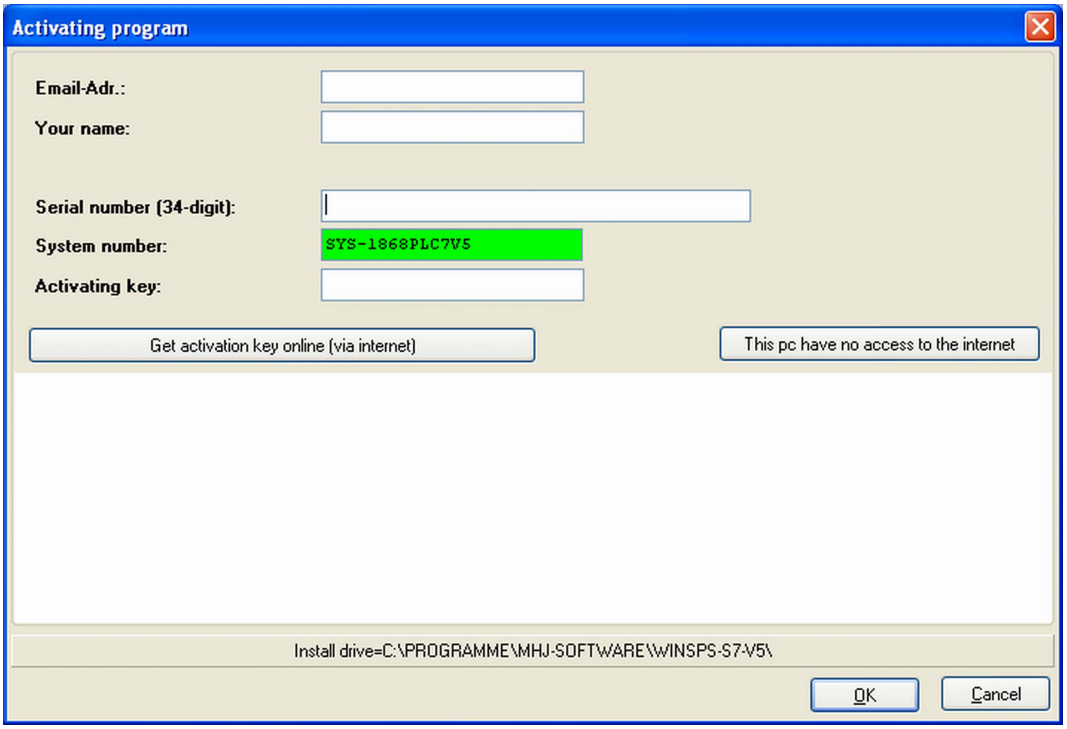
Installation WinPLC7 Demo

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.

8.3 Example project engineering

8.3.1 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

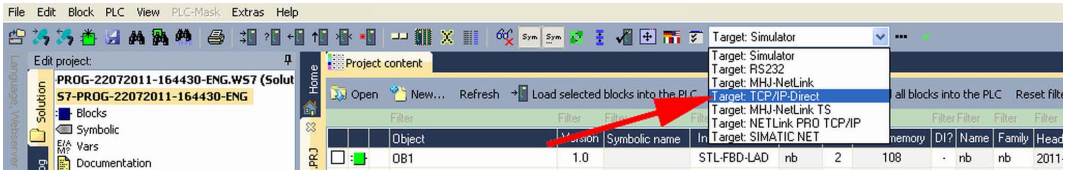
- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.
- One SPEED7 CPU and one digital output module are installed and cabled.
- The Ethernet PG/OP channel of the CPU is connected to your Ethernet network. Your CPU may be connected to your PC with an Ethernet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

8.3.2 Project engineering

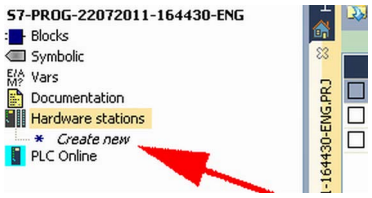
1. Start WinPLC7 ("Profi" version)
2. Create and open a new project with [Create a new solution].

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.

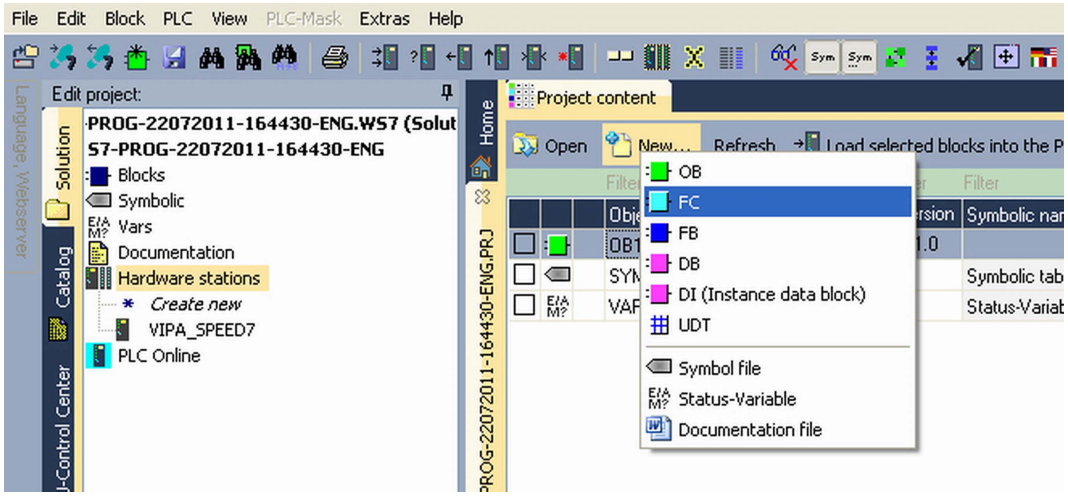
The hardware configuration is finished, now and the CPU may always be accessed by the IP parameters as well by means of WinPLC7.

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.

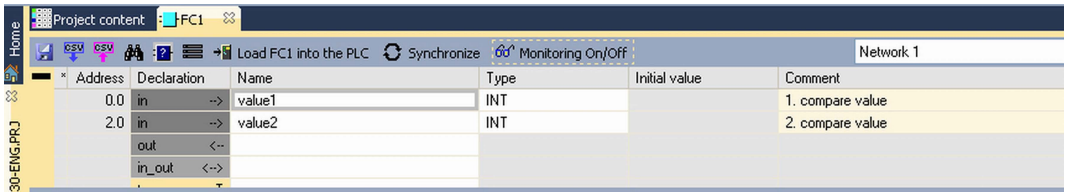
3. Here enter "1. compare value" and press the *[Return]* key.

   ⇨ 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].

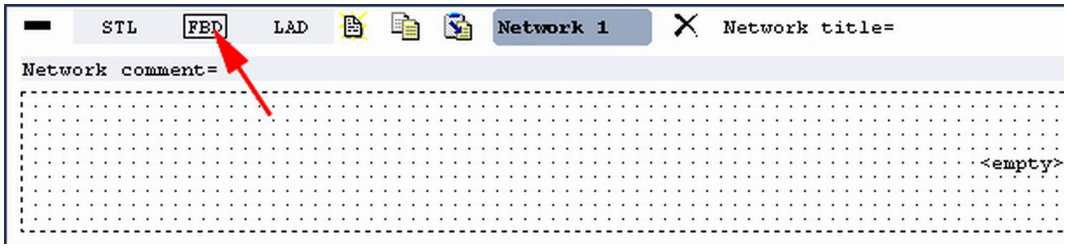
   ⇨ The parameter table shows the following entries, now:



Enter the program

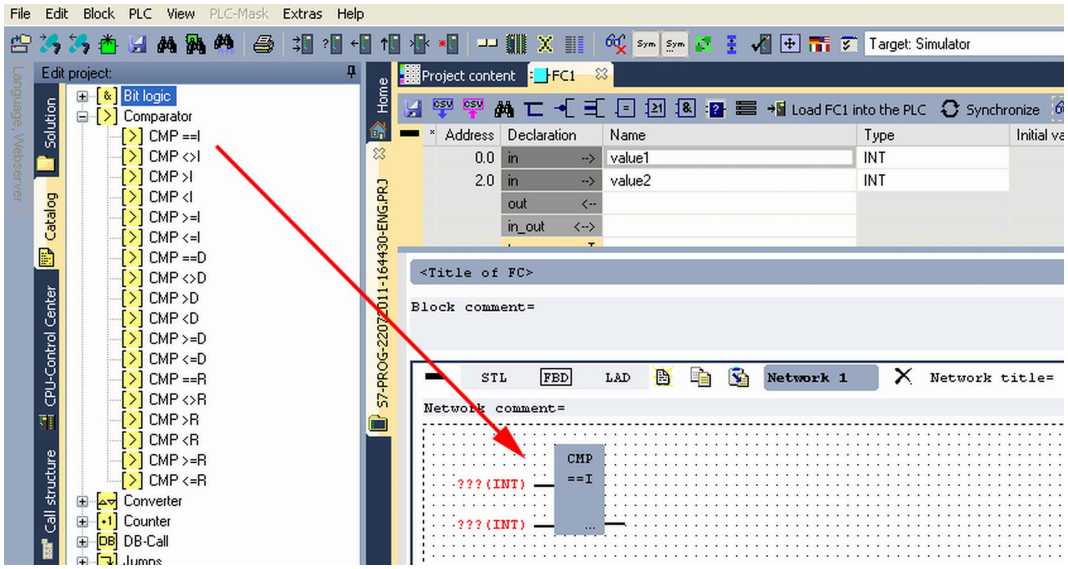
As requested in the job definition, the corresponding output is activated depending on the comparison of *value1* and *value2*. For each comparison operation a separate network is to be created.

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 "#".

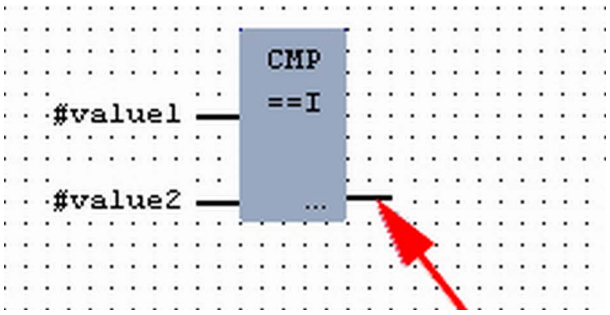
5. Type in "#" and press the *[Return]* key.

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

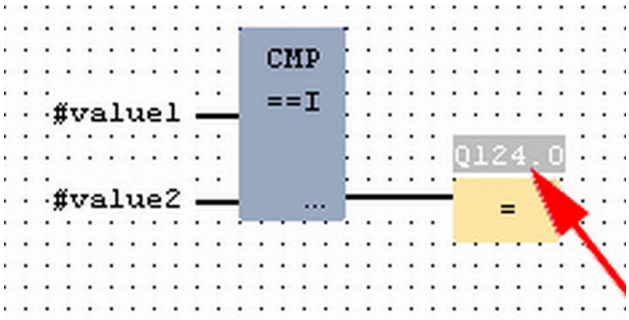
7. Proceed in the same way with the parameter *value2*.

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

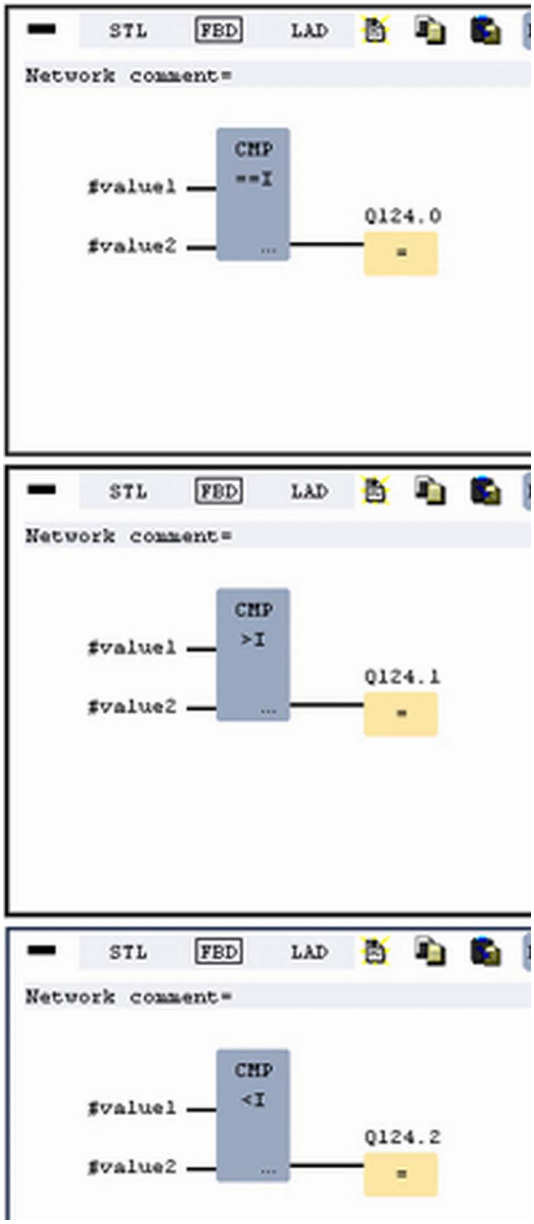
For further comparisons the operations "CMP>I" at Q 124.1 and "CMP<I" at Q 124.2 are necessary. Create a network for both operations with the following proceeding:

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:



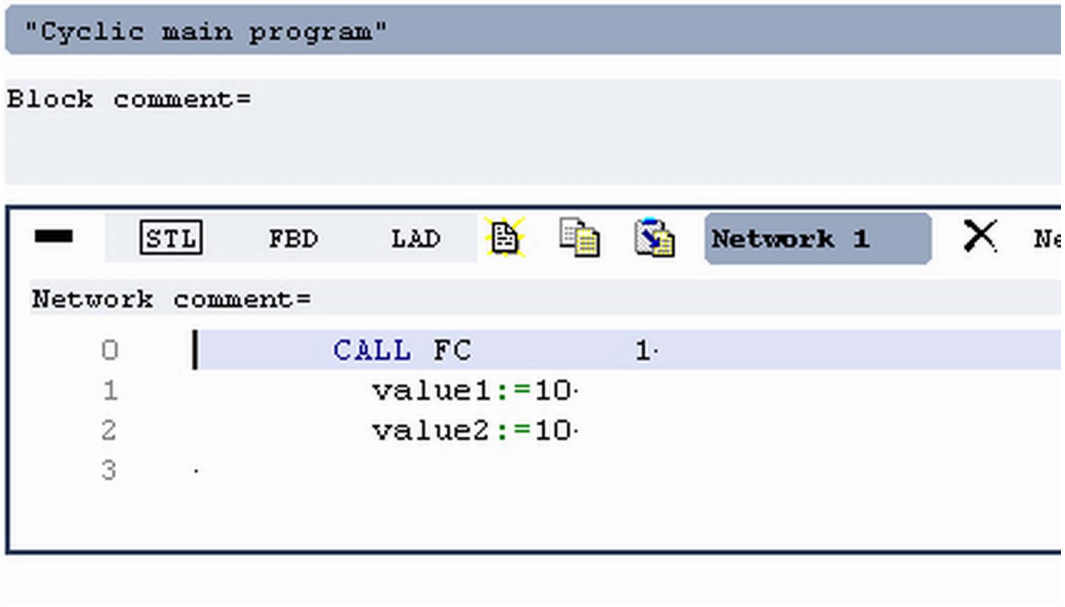
Creating the block OB 1

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:



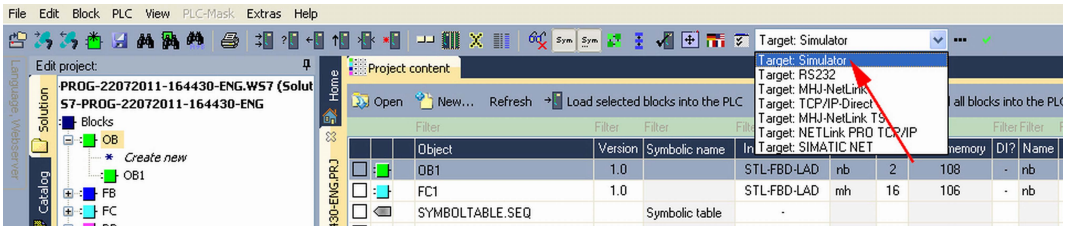
5. Save the OB 1 with respectively press *[Strg]+[S]*.

8.3.3 Test the PLC program in the *Simulator*

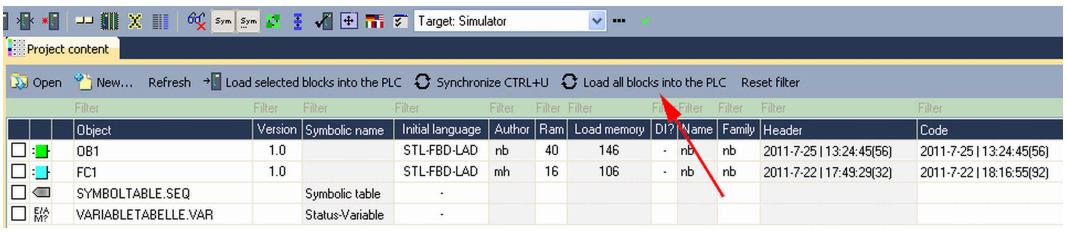
Proceeding

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.

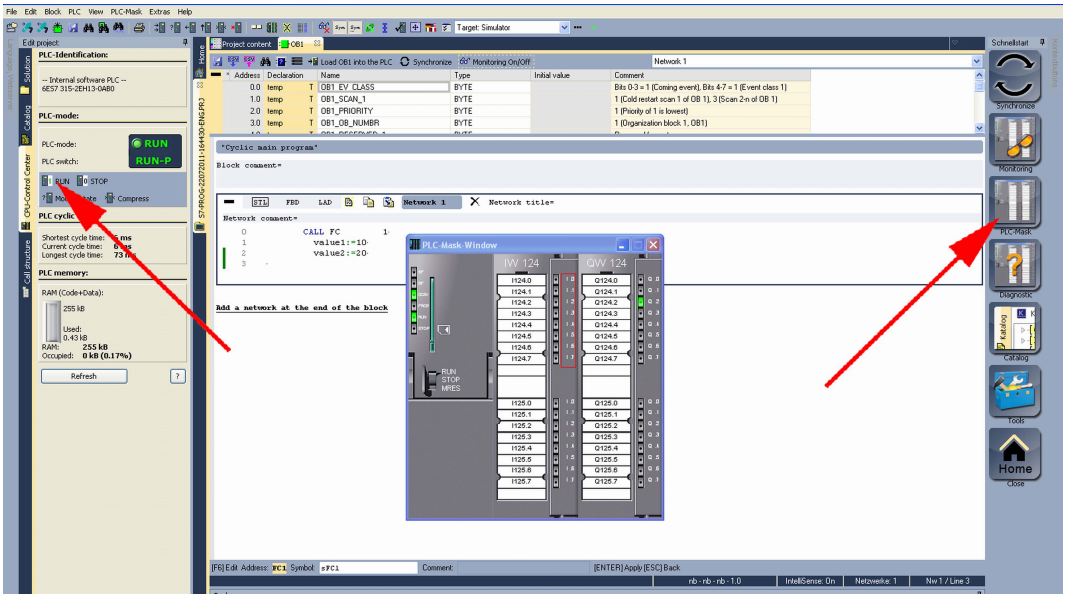
5. Double click to the process image and enter at *'Line 2'* the address PQB 124. Confirm your input with [OK]. A value marked by red color corresponds to a logical "1".

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.

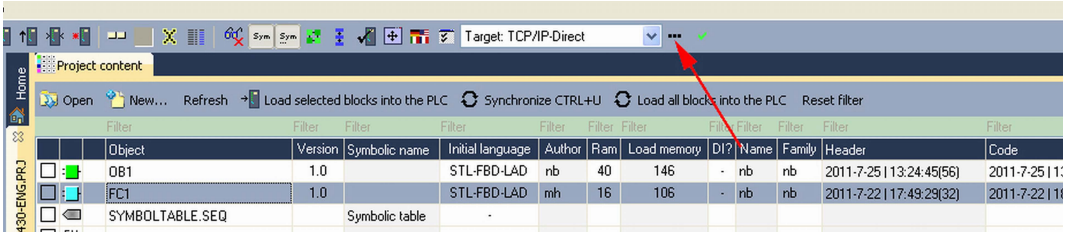
1. Open the *PLC mask* with *'view ➔ PLC mask'*.
   ⇨ A CPU is graphically displayed.
2. Double-click to the output module, open its properties dialog and enter the Module address 124.
3. Switch the operating mode switch to RUN by means of the mouse.
   ⇨ Your program is executed and displayed in the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



8.3.4 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'*.

9 Configuration with TIA Portal

9.1 TIA Portal - Work environment

9.1.1 General

General

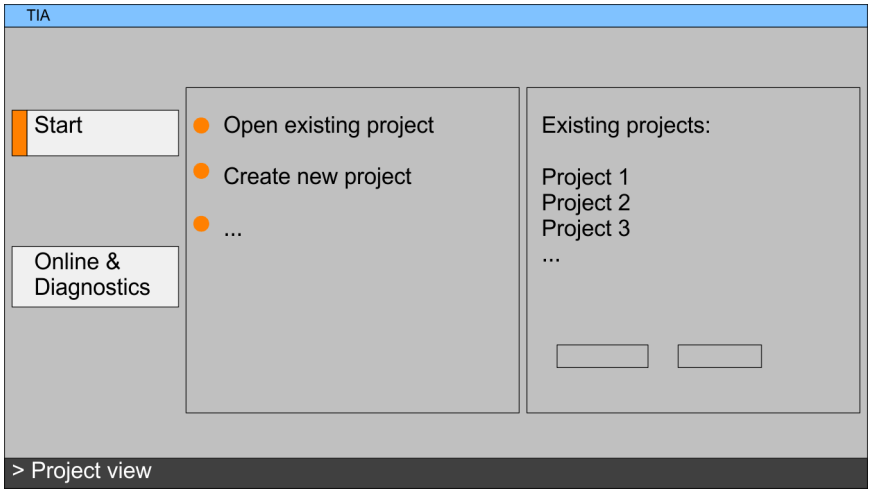
In this chapter the project engineering of the VIPA CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a VIPA CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means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

9.1.2 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

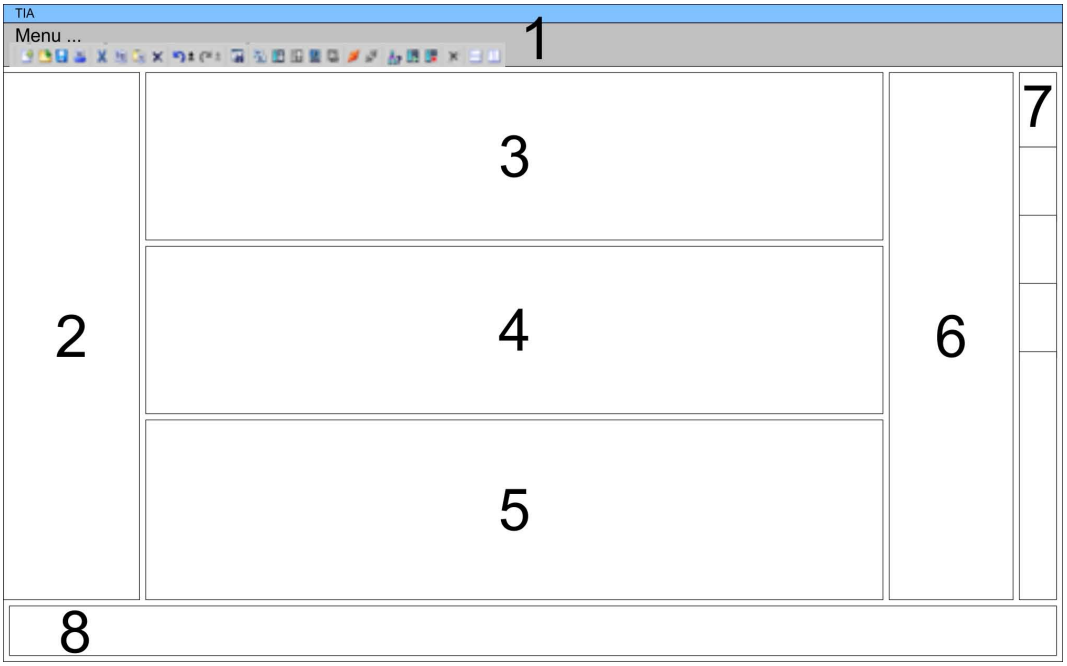
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

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

9.2 TIA Portal - Hardware configuration - CPU

Configuration Siemens CPU

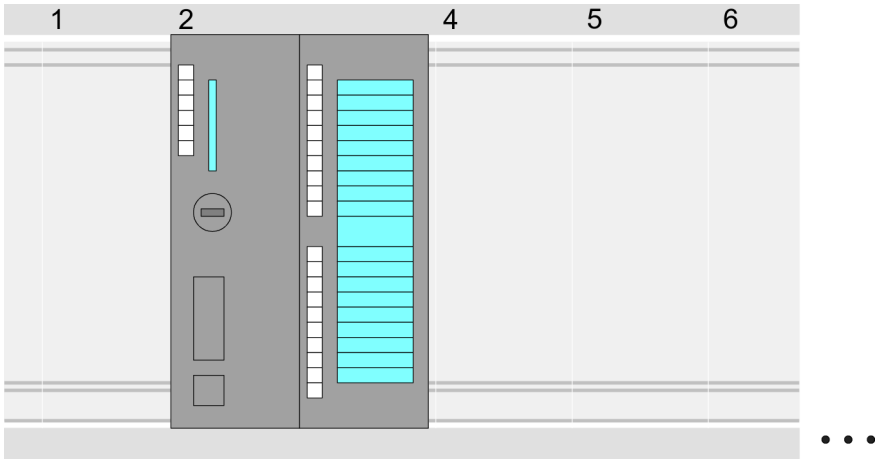
With the Siemens TIA Portal, the CPU 312-5BE23 from VIPA is to be configured as CPU 312C (6ES7 312-5BE03-0AB0/V2.6) from Siemens.

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 312C (6ES7 312-5BE03-0AB0/V2.6)

   ⇨ The CPU is inserted with a profile rail.

Project area:



Device overview:

Module	...	Slot	...	Type	...
PLC ...		2		CPU 312C	
MPI interface...		2 0		MPI interface	
DI10/DO6...		2 X2		DI10/DO6	
Counter...		2 4		Counter	
...					

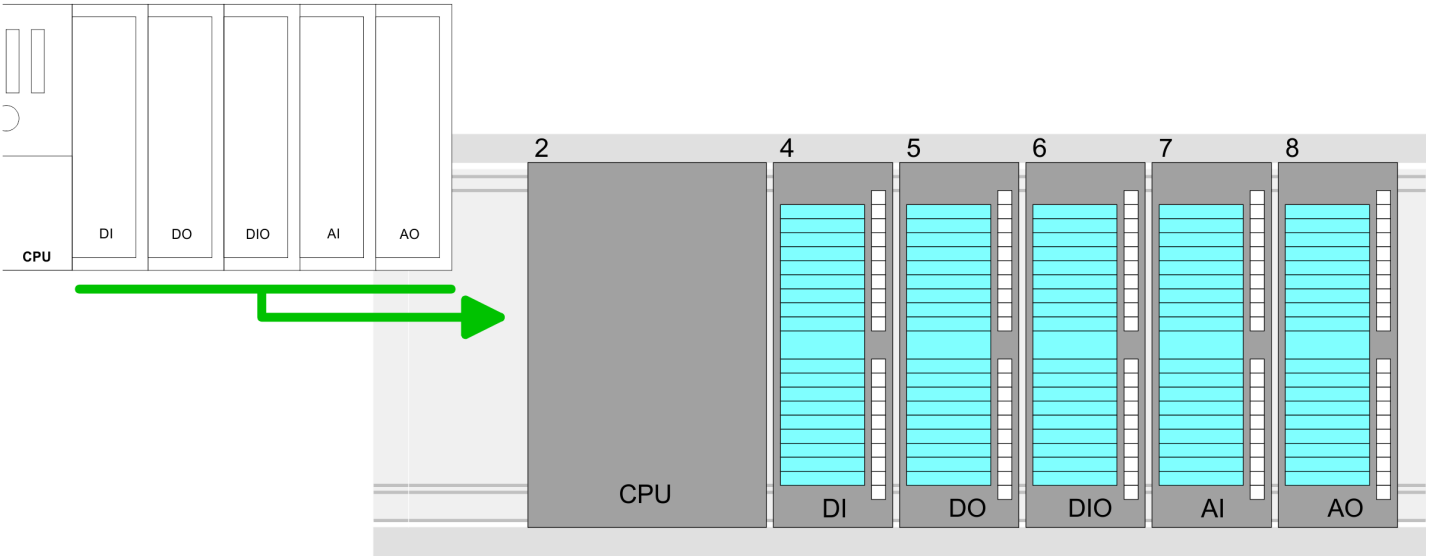
Setting standard CPU parameters

Since the CPU 312-5BE23 from VIPA is configured as Siemens CPU 312C, so the parametrization 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 'CPU parametrization' on page 50*

9.3 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.

9.4 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.

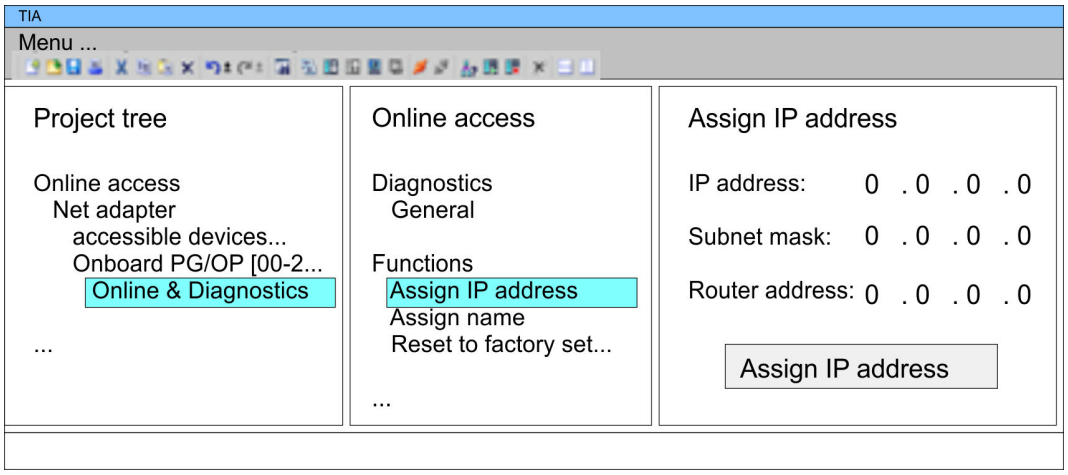
Assign IP address parameters

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].

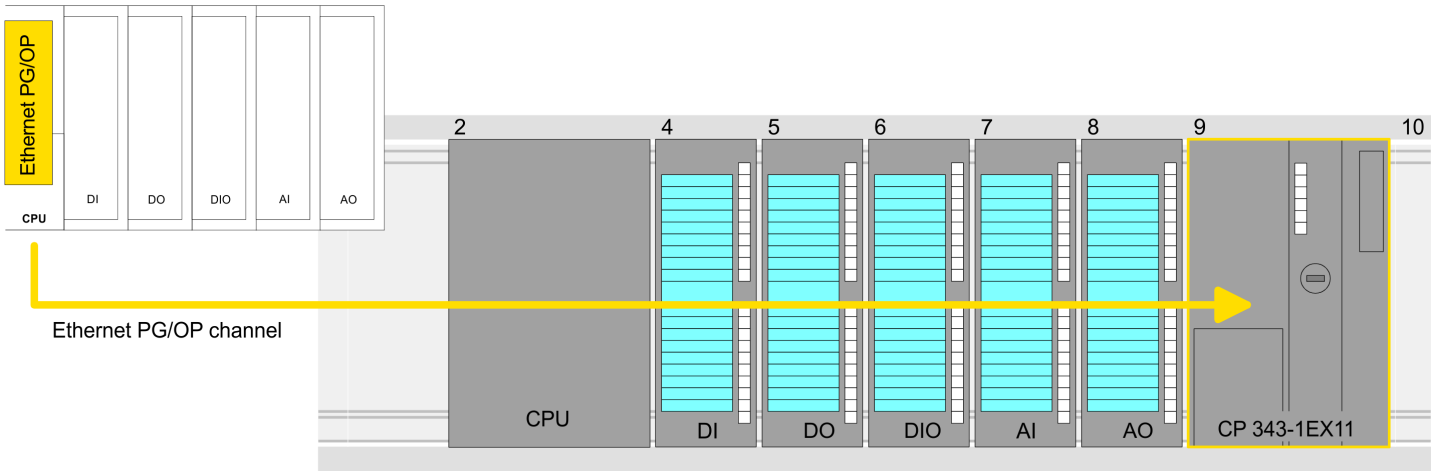
   ⇨ Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.



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 312C (6ES7 312-5BE03-0AB0/V2.6).
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	

9.5 TIA Portal - VIPA-Include library

Overview

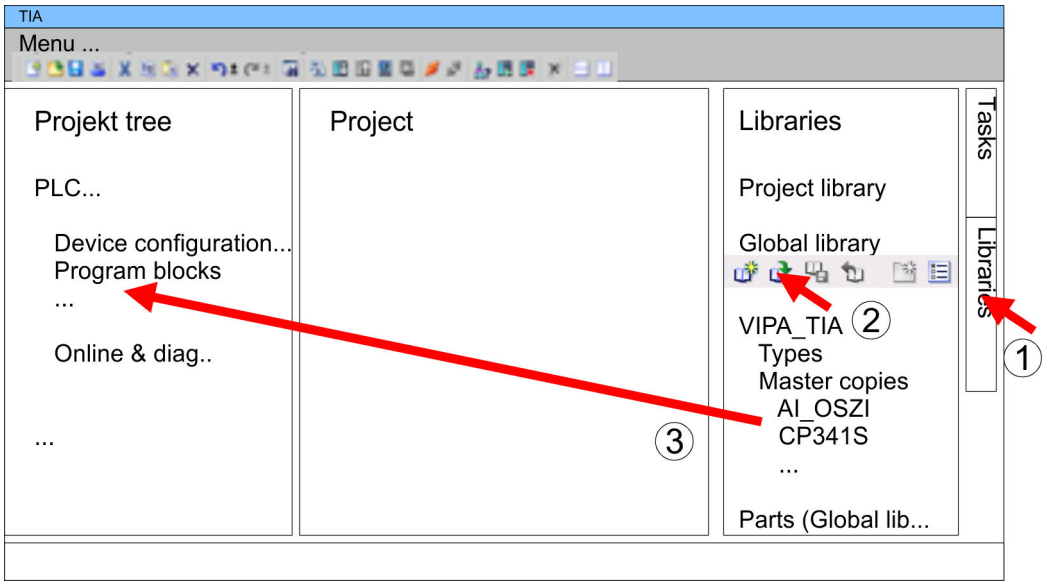
- The VIPA specific blocks can be found in the "Service" area of www.vipa.com as library download file at *Downloads* > *VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use VIPA specific blocks you have to import them into your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

Unzip ...TIA_Vxx.zip

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

Open library and transfer blocks into the project

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



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

9.6 TIA Portal - Project transfer

Overview

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

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

Transfer via MPI

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

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

Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- 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 5.17 'CMD - auto commands' on page 73*

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 5.18 'Diagnostic entries' on page 74*

Appendix

Content

A System specific event IDs

Event IDs Chapter 5.18 'Diagnostic entries' on page 74

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

Event ID	Description
0xE005	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE007	Configured input/output bytes do not fit in the periphery area
0xE008	Internal error - Please contact the hotline!
0xE009	Error on accessing the standard backplane bus
0xE010	Non-defined component recognised at the standard backplane bus
	ZINFO2: Slot
	ZINFO3: Type identifier
0xE011	Master project engineering at slave CPU not possible or wrong slave configuration
0xE012	Error at configuration standard backplane bus
0xE013	Error at shift register access to standard backplane bus digital modules
0xE014	Error in Check_Sys
0xE015	Error in access to master
	ZINFO2: Slot of the master
	ZINFO2: Page frame master
0xE016	Maximum block size exceeded in master transfer
	ZINFO1: Periphery address
	ZINFO2: Slot
0xE017	Error in access to integrated slave
0xE018	Error in mapping the master periphery
0xE019	Error on standard backplane bus system detection
0xE01A	Error at detection of the operating mode (8/9 bit)
0xE01B	Error: Maximum number of plug-in components exceeded
0xE020	Error: Interrupt information undefined
	ZINFO2: Slot
	ZINFO3: Not user relevant
	DatID: Interrupt type
0xE030	Error of the standard backplane bus
0xE033	Internal error - Please contact the hotline!
0xE0B0	SPEED7 is not stoppable
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE0C0	Not enough memory space in the working memory for code block (block too large)
0xE0CB	Error on SSL access

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

Event ID	Description
0xE200	Memory card writing finished (Copy Ram2Rom)
	OB: Not user relevant
	PK: Not user relevant
0xE210	Memory card reading finished (reload after memory reset)
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1 - Position 0: Not user relevant
0xE21D	Memory card reading: Error on reload (after memory reset), error in the block header
	ZINFO1: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO2: Block number
	ZINFO3: Block length
0xE21E	Memory card reading: Error in recharging (after memory reset), "Protect.wld" file too large
	OB: Not user relevant
0xE21F	Memory card reading: Error at reload (after memory reset), checksum error when reading
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Not user relevant
	ZINFO2: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB

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

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

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

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

Event ID	Description
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE404	Feature set deleted due to CRC error
0xE405	The trial time of a feature set/memory card has expired
	OB: Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00

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

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

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

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

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

Event ID	Description
	ZINFO2: Length of the address range
	ZINFO3: Size of the process image
	DatID: Address range
0xE801	CMD - auto command: CMD_START recognized and executed
0xE802	CMD - auto command: CMD_End recognized and executed
0xE803	CMD - auto command: WAIT1SECOND recognized and executed
0xE804	CMD - auto command: WEBPAGE recognized and executed
0xE805	CMD - auto command: LOAD_PROJECT recognized and executed
0xE806	CMD - auto command: SAVE_PROJECT recognized and executed
	ZINFO3: Status
	0: Error
	1: OK
	32768: Wrong password
0xE807	CMD - auto command: FACTORY_RESET recognized and executed
0xE808	Internal error - Please contact the hotline!
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE809	Internal error - Please contact the hotline!
	ZINFO3: Not user relevant
0xE80A	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80B	CMD - auto command: DIAGBUF recognized and executed
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80C	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80D	Internal error - Please contact the hotline!

Event ID	Description
0xE80E	CMD - auto command: SET_NETWORK recognized and executed
0xE80F	Internal error - Please contact the hotline!
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE810	Internal error - Please contact the hotline!
0xE811	Internal error - Please contact the hotline!
0xE812	Internal error - Please contact the hotline!
0xE813	Internal error - Please contact the hotline!
0xE814	CMD - auto command: SET_MPI_ADDRESS identified
0xE816	CMD - auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal error - Please contact the hotline!
	ZINFO3: Not user relevant
0xE820	Internal message
0xE821	Internal message
0xE822	Internal message
0xE823	Internal message
0xE824	Internal message
0xE825	Internal message
0xE826	Internal message
0xE827	Internal message
0xE828	Internal message
0xE829	Internal message
0xE82A	CMD - auto command: CPUTYPE_318 recognized and executed
	ZINFO3: Error code
0xE82B	CMD - auto command: CPUTYPE_ORIGINAL recognized and executed
	ZINFO3: Error code
0xE82C	CMD - auto command: WEBVISU_PGOP_ENABLE recognized and executed
0xE82D	CMD - auto command: WEBVISU_PGOP_DISABLE recognized and executed
0xE82E	CMD - auto command: WEBVISU_CP_ENABLE recognized and executed
0xE82F	CMD - auto command: WEBVISU_CP_DISABLE recognized and executed
0xE8FB	CMD - auto command: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty
0xE8FC	CMD - auto command: Error: Some IP parameters missing in SET_NETWORK
0xE8FE	CMD - auto command: Error: CMD_START not found
0xE8FF	CMD - auto command: Error while reading CMD file (memory card error)

Event ID	Description
0xE901	Checksum error
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	DatID: Not user relevant
0xE902	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	DatID: Not user relevant
0xE904	PG/OP: Multiple parametrization of a peripheral address
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
	DatID: 0x54 Peripheral address is input address
	DatID: 0x55 Peripheral address is output address
0xE910	PG/OP: Input peripheral address out of peripheral area
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
0xE911	PG/OP: Output peripheral address out of peripheral area
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
0xE920	Configuration error PROFINET
	ZINFO1 - Position 0: Error code
0xE980	Error when loading the WebVisu project file
0xE981	Error in the configuration of the WebVisu project
0xE982	Internal error of the WebVisu server
0xE983	Hardware configuration of the control is not loaded, WebVisu is not started
0xE984	WebVisu is blocked by the user, start of the WebVisu was prevented
0xE985	WebVisu was started
0xE986	WebVisu was stopped
0xE987	WebVisu was enabled by the user
0xE988	WebVisu was disabled by the user
0xEA00	Internal error - Please contact the hotline!
	PK: Not relevant to user
	DatID: Not user relevant
0xEA01	Internal error - Please contact the hotline!
	PK: Not user relevant

Event ID	Description
	ZINFO1: Slot
	DatID: Not user relevant
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Type identifier target
	ZINFO3: Type identifier
	DatID: Not user relevant
0xEA03	SBUS: Communication error between CPU and IO controller
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Status
	0: OK
	1: Error
	2: Empty
	3: Busy
	4: Time out
	5: Internal blocking
	6: Too many telegrams

Event ID	Description
	7: Not Connected
	8: Unknown
	DatID: Not user relevant
0xEA04	SBUS: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA05	Internal error - Please contact the hotline!
0xEA07	Internal error - Please contact the hotline!
0xEA08	SBUS: Configured input data width not the same as the connected input data width
	ZINFO1: Configured input data width
	ZINFO2: Slot
	ZINFO3: Input data width of the connected component
0xEA09	SBUS: Configured output data width not the same as the connected output data width
	ZINFO1: Configured output data width
	ZINFO2: Slot
	ZINFO3: Output data width of the plugged component
0xEA10	SBUS: Input periphery address outside the periphery area
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA11	SBUS: Output periphery address outside the periphery area
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA12	SBUS: Error in writing dataset
	ZINFO1: Slot
	ZINFO2: Dataset number
	ZINFO3: Dataset length
0xEA14	SBUS: Multiple configuration of a periphery address (diagnostic address)
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA15	Internal error - Please contact the hotline!
	ZINFO2: Slot of the master
0xEA18	SBUS: Error in mapping the master periphery
	ZINFO2: Slot of the master
0xEA19	Internal error - Please contact the hotline!

Event ID	Description
	PK: Not user relevant
	ZINFO2: HW slot
	ZINFO3: Interface type
	DatID: Not user relevant
0xEA1A	SBUS: Error in access to SBUS FPGA address table
	PK: Not user relevant
	ZINFO2: HW slot
	ZINFO3: Table
	0: Read
	1: Writing
	DatID: Not user relevant
0xEA20	Error: RS485 interface is not pre-set to PROFIBUS DP master bus a PROFIBUS DP master is configured
0xEA21	Error: Configuration RS485 interface X2/X3: PROFIBUS DP master is configured but missing
	ZINFO2: Interface X is configured incorrectly
0xEA22	Error: Configuration RS485 interface X2: Value is outside the limits
	ZINFO2: Configuration for X2
0xEA23	Error: Configuration RS485 interface X3: Value is outside the limits
	ZINFO2: Configuration for X3
0xEA24	Error: Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used
	ZINFO2: Configuration for X2
	ZINFO3: Configuration for X3
0xEA30	Internal error - Please contact the hotline!
	ZINFO1: Status
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xEA40	Internal error - Please contact the hotline!
	OB: Slot of the CP
	PK: File number
	ZINFO1: Version of the CP
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Line
0xEA41	Internal error - Please contact the hotline!
	OB: Slot of the CP
	PK: File number
	ZINFO1: Version of the CP
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant

Event ID	Description
	DatID: Line
0xEA50	PROFINET IO controller: Error in the configuration
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Slot at the device
	DatID: Not user relevant
0xEA51	PROFINET IO controller: There is no PROFINET IO controller at the configured slot
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Recognized type identifier at the configured slot
	DatID: Not user relevant
0xEA52	PROFINET IO controller: Too many configured PROFINET IO controllers
	PK: Not user relevant
	ZINFO1: Number of configured controllers
	ZINFO2: Slot of the excessively configured controller
	DatID: Not user relevant
0xEA53	PROFINET IO controller: Too many configured PROFINET IO devices
	ZINFO1: Number of configured devices
	ZINFO2: Slot
	ZINFO3: Maximum possible number of devices
0xEA54	PROFINET IO controller: Multiple configuration of a periphery address
	PK: Not user relevant
	ZINFO1: Logical address of the IO system
	ZINFO2: Rack/slot of the controller
	ZINFO3: Base address of the block which is too large
	DatID: Not user relevant
0xEA55	PROFINET IO controller: Too many slots configured
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Number of configured slots
0xEA56	PROFINET IO controller: Too many subslots configured
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Number of configured subslots
0xEA57	PROFINET IO controller: The port configuration in the virtual SLIO device has no effect.
0xEA61	Internal error - Please contact the hotline!

Event ID	Description
	OB: File number
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA62	Internal error - Please contact the hotline!
	OB: File number.
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA63	Internal error - Please contact the hotline!
	OB: File number
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA64	PROFINET IO controller/EtherCAT-CP: Error in configuration
	PK: Interface
	ZINFO1 - Bit 0: Too many devices
	ZINFO1 - Bit 1: Too many devices per second
	ZINFO1 - Bit 2: Too many input bytes per millisecond
	ZINFO1 - Bit 3: Too many output bytes per millisecond
	ZINFO1 - Bit 4: Too many input bytes per device
	ZINFO1 - Bit 5: Too many output bytes per device
	ZINFO1 - Bit 6: Too many productive connections
	ZINFO1 - Bit 7: Too many input bytes in the process image
	ZINFO1 - Bit 8: Too many output bytes in the process image
	ZINFO1 - Bit 9: Configuration not available
	ZINFO1 - Bit 10: Configuration invalid
	ZINFO1 - Bit 11: Refresh interval too small
	ZINFO1 - Bit 12: Refresh interval too large
	ZINFO1 - Bit 13: Invalid device number
	ZINFO1 - Bit 14: CPU is configured as an I device
	ZINFO1 - Bit 15: Assume IP address in another way. Is not supported for the IP address of the controller.
	ZINFO2 - Bit 0: Incompatible configuration (SDB version not supported)
	ZINFO2 - Bit 1: EtherCAT: EoE configured but not supported (Possible cause is a too short cycle time of the EtherCAT master system. When using EoE terminals, at least a cycle time of 4ms must be configured.)

Event ID	Description
	ZINFO2 - Bit 2: DC parameter invalid
	ZINFO2 - Bit 3: I device configuration invalid (slot gap)
	ZINFO2 - Bit 4: MRP configuration invalid (client)
0xEA65	Internal error - Please contact the hotline!
	PK: Platform
	0: none
	8: CP
	9: Ethernet CP
	10: PROFINET CP
	12: EtherCAT CP
	16: CPU
	ZINFO1: ServiceID in which the error occurred
	ZINFO2: Command in which the error occurred
	1: Request
	2: Connect
	3: Error
0xEA66	PROFINET IO controller: Error in the communication stack
	OB: StackError.Service
	PK: Rack/slot
	ZINFO1: StackError.Error.Code
	ZINFO2: StackError.Error.Detail
	ZINFO3 - Position 0: StackError.Error.AdditionalDetail
	ZINFO3 - Position 8: StackError.Error.AreaCode
	DatID: StackError.DeviceRef
0xEA67	PROFINET IO controller: Error reading dataset
	OB: Rack/slot of the controller
	PK: Error type
	0: Dataset error local
	1: Dataset error stack
	2: Dataset error station
	ZINFO1: Dataset number
	ZINFO2: Dataset handle (caller)
	ZINFO3: Internal error code from PN stack
	DatID: Device
0xEA68	PROFINET IO controller: Error writing dataset
	OB: Rack/slot of the controller
	PK: Error type
	0: Dataset error local

Event ID	Description
	1: Dataset error stack
	2: Dataset error station
	ZINFO1: Dataset number
	ZINFO2: Dataset handle (caller)
	ZINFO3: Internal error code from PN stack
	DatID: Device
0xEA69	Internal error - Please contact the hotline!
	ZINFO1: Minimum version for the FPGA
	ZINFO2: Loaded FPGA version
0xEA6A	PROFINET IO controller: Service error in the communication stack
	OB: Service ID
	PK: Rack/slot
	ZINFO1: ServiceError.Code
	ZINFO2: ServiceError.Detail
	ZINFO3 - Position 0: ServiceError.AdditionalDetail
	ZINFO3 - Position 8: ServiceError.AreaCode
0xEA6B	PROFINET IO controller: Incorrect Vendor-ID
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID

Event ID	Description
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6C	PROFINET IO controller: Incorrect Device-ID
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
0xEA6D	PROFINET IO controller: No empty name
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING

Event ID	Description
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6E	PROFINET IO controller: Wait for RPC response
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6F	PROFINET IO controller: PROFINET module deviation

Event ID	Description
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA70	PROFINET IO controller: PROFINET stack configuration error
	OB: UnsupportedApiError.api
	PK: Rack/slot
	ZINFO1: UnsupportedApiError.slot
	ZINFO2: UnsupportedApiError.subslot
	DatID: UnsupportedApiError.deviceID
0xEA71	Internal error - Please contact the hotline!
	PK: Rack/slot
	ZINFO1: functionIndex
	ZINFO2: Not user relevant
0xEA72	Internal error - Please contact the hotline!
	OB: Connection number
	PK: Slot of the controller
	ZINFO1: Error cause

Event ID	Description
	129: PNIO
	207: RTA error
	218: AlarmAck
	219: IODConnectRes
	220: IODReleaseRes
	221: IOD/IOXControlRes
	222: IODReadRes
	223: IODWriteRes
	ZINFO2: ErrorDecode
	128: PNIORW: Service Read Write
	129: PNIO: Other Service or internal e.g. RPC errors
	130: Vendor specific
	ZINFO3: Error code (PN spec. V2.722 chapter 5.2.6)
	DatID: Device ID
0xEA81	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: SvnRevision
0xEA82	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: SvnRevision
0xEA83	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: SvnRevision
0xEA91	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status

Event ID	Description
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number
0xEA92	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number
0xEA93	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number
0xEA97	Internal error - Please contact the hotline!
	ZINFO3: Slot
0xEA98	Error in file reading via SBUS
	PK: Not user relevant
	ZINFO3: Slot

Event ID	Description
	DatID: Not user relevant
0xEA99	Parameter assignment job could not be executed
	PK: Not user relevant
	ZINFO1: File version on MMC/SD (if not 0)
	ZINFO2: File version of the SBUS module (if not 0)
	ZINFO3: Slot
	DatID: Not user relevant
0xEAA0	Internal error - Please contact the hotline!
	OB: Current operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: Not user relevant
	ZINFO3: Number of errors which occurred
0xEAB0	Invalid link mode
	OB: Current operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)

Event ID	Description
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: Current connection mode
	1: 10Mbit half-duplex
	2: 10Mbit full-duplex
	3: 100Mbit half-duplex
	4: 100Mbit full-duplex
	5: Connection mode undefined
	6: Auto Negotiation
0xEAC0	Internal error - Please contact the hotline!
	ZINFO1: Error code
	2: Internal error
	3: Internal error
	4: Internal error
	5: Internal error
	6: Internal error
	7: Internal error
	8: Internal error
	8: Internal error
0xEAD0	SyncUnit configuration error
	ZINFO1: Status
0xEB02	System SLIO error: Preset configuration does not match actual configuration
	ZINFO1: Bit mask slots 1-16
	ZINFO2: Bit mask slots 17-32
	ZINFO3: Bit mask slots 33-48
	DatID: Bit mask slots 49-64
0xEB03	System SLIO error: IO mapping

Event ID	Description
	PK: Not user relevant
	ZINFO1: Error type
	1: SDB parser error
	2: Configured address already used
	3: Mapping error
	ZINFO2: Slot (0=cannot be determined)
	DatID: Not user relevant
0xEB04	SLIO-Bus: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	DatID: Input
	DatID: Output
0xEB05	System SLIO error: Bus structure for isochronous process image not suitable
	PK: Not user relevant
	ZINFO2: Slot (0=cannot be determined)
	DatID: Not user relevant
0xEB06	System SLIO error: Timeout with the isochronous process image
0xEB10	System SLIO error: Bus error
	PK: Not user relevant
	ZINFO1: Error type
	96: Bus enumeration error
	128: General error
	129: Queue execution error
	130: Error interrupt
	ZINFO2: Error on bus enumeration error (ZINFO1)
	DatID: Not user relevant
0xEB11	System SLIO error: Error during bus initialization
	PK: Not user relevant
	DatID: Not user relevant
0xEB20	System SLIO error: Interrupt information undefined
0xEB21	System SLIO error: Accessing configuration data
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC02	EtherCAT: configuration warning
	ZINFO1: Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid

Event ID	Description
	3: Slot invalid
	4: Master configuration invalid
	5: Master type invalid
	6: Slave diagnostic address invalid
	7: Slave address invalid
	8: Slave module IO configuration invalid
	9: Logical address already in use
	10: Internal error
	11: IO mapping error
	12: Error
	13: Error in initialising the EtherCAT stack (is entered by the CP)
	14: Slave station number already occupied by virtual SLIO device
	ZINFO2: Station number
0xEC03	EtherCAT: Configuration error
	PK: Not user relevant
	ZINFO1: Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid
	3: Slot invalid
	4: Master configuration invalid
	5: Master type invalid
	6: Slave diagnostic address invalid
	7: Slave address invalid
	8: Slave module IO configuration invalid
	9: Logical address already in use
	10: Internal error
	11: IO mapping error
	12: Error
	13: Error in initialising the EtherCAT stack (is entered by the CP)
	14: Slave station number already occupied by virtual SLIO device
	ZINFO2: Station number
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC04	EtherCAT: Multiple configuration of a periphery address
	PK: Not user relevant
	ZINFO1: Periphery address
	ZINFO2: Slot
	DatID: Not user relevant

Event ID	Description
0xEC05	EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Not user relevant
	ZINFO1: Station address of the EtherCAT device
	ZINFO2: Error code
	1: WARNING: For the drive the DC Beckhoff mode is recommended (DC reference clock is not in Beckhoff Mode)!
	2: NOTE: For the drive the DC Hilscher mode is recommended (DC reference clock is not in Beckhoff Mode)!
	3: The station address could not be determined for checking (station address in ZINFO1 is accordingly 0)
	4: The slave information could not be determined for checking (station address in ZINFO1 is accordingly 0)
	5: The EtherCAT status of the drive could not be determined
	6: Error when sending the SDO request (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	7: Drive returns error in the SDO response (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	8: SDO time out, DC mode could not be determined (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC10	EtherCAT: Recurrence bus with all slaves
	ZINFO1 - Position 0: New status

Event ID	Description
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: Number of stations, which are not in the same state as the master
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xEC11	EtherCAT: Recurrence bus with missing slaves
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station not available
	DatID: Station available

Event ID	Description
	DatID: Input address
	DatID: Output address
0xEC12	EtherCAT: Recurrence slave
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AL status code
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xEC30	EtherCAT: Topology OK
	ZINFO2: Diagnostic address of the master
0xEC40	Bus cycle time infringement resolved
	ZINFO2: Logical address of the IO system
0xEC50	EtherCAT: Distributed clocks (DC) out of sync
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN

Event ID	Description
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2: Diagnostic address of the master
	ZINFO3: DC state change
	0: DC master out of sync
	1: DC slave stations out of sync
0xEC80	EtherCAT: Bus error resolved
	ZINFO1: Logical address of the IO system
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0xED10	EtherCAT: Breakdown bus
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station available
	DatID: Station not available
	DatID: Input address

Event ID	Description
	DatID: Output address
0xED12	EtherCAT: Breakdown slave
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state

Event ID	Description
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xED20	EtherCAT: Bus state change without calling OB86
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address

Event ID	Description
0xED21	EtherCAT: Incorrect bus status change
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Error code
	4: Cancel (master state change)
	8: Busy
	11: Invalid parameters
	14: Invalid status
	16: Time out
	DatID: Station available
	DatID: Station not available
	DatID: Output address
	DatID: Input address
0xED22	EtherCAT: Slave status change that does not generate an OB86
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp

Event ID	Description
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT

Event ID	Description
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xED23	EtherCAT: Time out while changing the master state to OP, after CPU has changed to RUN
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Master status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: EtherCAT configuration present
	0: There is no EC configuration
	1: There is an EC configuration
	ZINFO3: DC in sync
	0: Not in sync
	1: In sync
0xED30	EtherCAT: Topology deviation

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

Event ID	Description
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Slave address
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start

Event ID	Description
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Recurrence slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status
0xED61	EtherCAT: Diagnostic buffer CP: CoE emergency
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1 - Position 0: Error register
	ZINFO1 - Position 8: MEF-Byte1
	ZINFO2 - Position 0: MEF-Byte2
	ZINFO2 - Position 8: MEF-Byte3
	ZINFO3 - Position 0: MEF-Byte4
	ZINFO3 - Position 8: MEF-Byte5
	DatID: Error code
0xED62	EtherCAT: Diagnostic buffer CP: Error on SDO access
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1: Index
	ZINFO2: SDO error code (high word)
	ZINFO3: SDO error code (low word)
	DatID: Sub index

Event ID	Description
0xED63	EtherCAT: Diagnostic buffer CP: Error in the response to an INIT command
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1: Error type
	0: Not defined
	1: No response
	2: Validation error
	3: INIT command failed, requested station could not be reached
0xED70	EtherCAT: Diagnostic buffer CP: Twofold hot connect group recognised
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: EtherCAT station address
0xED80	Bus error occurred (receive time-out)
	ZINFO1: Logical address of the IO system
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0xEE00	Additional information at UNDEF_OPCODE
	OB: Not user relevant
	ZINFO1: Not user relevant

Event ID	Description
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEE01	Internal error - Please contact the hotline!
	ZINFO3: SFB number
0xEEEE	CPU was completely deleted, since after PowerON the start-up could not be finished
0xEF00	Internal error - Please contact the hotline!
	DatID: Not user relevant
0xEF01	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEF11	Internal error - Please contact the hotline!
0xEF12	Internal error - Please contact the hotline!
0xEF13	Internal error - Please contact the hotline!
0xEFFE	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEFFF	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xF9C1	Restart of the component
	OB: NCM_EVENT
	1: OVS: Component start-up request was denied
	3: Component data basis invalid
	6: IP_CONFIG: New IP address assigned by STEP7 configuration
	10: IP_CONFIG: A non-configured new IP address was assigned
	13: HW reset at P bus (for CPU memory reset)
	19: Switch actuation from STOP to RUN causes the restart of the component
	20: MGT: PG command causes the restart of the component
	21: MGT: Take-over of component data basis causes the hot restart of the component
	23: Stopping the sub-system after having loaded the already existing consistency-secured SDBs xxxx by the rack component
	25: The SIMATIC procedure has been selected for the time synchronisation of the component.
	26: Component actively established a connection
	28: The SDB xxxx loaded by the rack component is the consistency securing object (SDB type 0x3118)

Event ID	Description
	29: The component actively disconnected the system connection to the CPU
	31: Inconsistency of the component data base by loading SDB xxxx by the rack component (SDB type 0x3100)
	32: Periphery enabled by S7-CPU
	33: Periphery disabled by S7-CPU
	34: Component STOP due to switch actuation
	35: Component STOP due to invalid configuration
	36: Component STOP due to PG command
	38: SDB xxxx is not registered in the still valid consistency securing object, or it has an incorrect time stamp (SDB type 0x3107), the error is being corrected
	40: Memory reset executed
	44: Consistency of the data base achieved after loading the SDBs xxxx by the rack component (SDB type xxxx)
	45: Remanent part of the component data base is deleted by the rack component after being loaded
	70: Restore factory defaults (same as memory reset of CPU!)
	83: Network interface: automatic configuration, TP/ITP with 10 Mbit/s semi-duplex
	96: The MAC address was retrieved from the system SDB. This is the configured address.
	97: The MAC address was retrieved from the boot EPROM. This is the factory-provided address.
	100: Restart of the component
	101: Component STOP due to deletion of system SDBs
	104: PG command start was denied due to missing or inconsistent configuration
	105: Component STOP due to double IP address
	107: Start-up request by switch actuation was denied due to missing or inconsistent configuration
	PK: NCM_SERVICE
	2: Management
	3: Object management system
	6: Time synchronisation
	10: IP_CONFIG
	38: SEND/RECEIVE

B Integrated blocks

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

OB	Name	Description
OB 1	CYCL_EXC	Program Cycle
OB 10	TOD_INT0	Time-of-day Interrupt
OB 20	DEL_INT0	Time delay interrupt
OB 28	CYC_INT_250us	Cyclic interrupt
OB 29	CYC_INT_500us	Cyclic interrupt
OB 35	CYC_INT5	Cyclic interrupt
OB 40	HW_INT0	Hardware interrupt
OB 80	CYCL_FLT	Time error
OB 81	PS_FLT	Power supply error
OB 82	I/O_FLT1	Diagnostics interrupt
OB 83	I/O_FLT2	Insert / remove module
OB 85	OBNL_FLT	Priority class error
OB 86	RACK_FLT	Slave failure / restart
OB 100	COMPLETE RESTART	Start-up
OB 121	PROG_ERR	Programming error
OB 122	MOD_ERR	Periphery access error

SFB	Name	Description
SFB 0	CTU	Up-counter
SFB 1	CTD	Down-counter
SFB 2	CTUD	Up-down counter
SFB 3	TP	Create pulse
SFB 4	TON	On-delay
SFB 5	TOF	Create turn-off delay
SFB 7	TIMEMESS	Time measurement
SFB 12	BSEND	Sending data in blocks
SFB 13	BRCV	Receiving data in blocks:
SFB 14	GET	Remote CPU read
SFB 15	PUT	Remote CPU write
SFB 31	NOTIFY8P	Messages without acknowledge display (8x)
SFB 32	DRUM	Realize a step-by-step switch

SFB	Name	Description
SFB 33	ALARM	Messages with acknowledgement display
SFB 34	ALARM_8	Messages without associated values (8x)
SFB 35	ALARM_8P	Messages with associated values (8x)
SFB 36	NOTIFY8	Messages without acknowledgement display
SFB 47	COUNT	Control counter
SFB 48	FREQUENC	Frequency measurement
SFB 49	PULSE	Pulse width modulation
SFB 52	RDREC	Read record set
SFB 53	WRREC	Write record set
SFB 54	RALRM	Receiving an interrupt from a periphery module
SFB 238	EC_RWOD	Function is used internally

SFC	Name	Description
SFC 0	SET_CLK	Set system clock
SFC 1	READ_CLK	Read system clock
SFC 2	SET_RTM	Set run-time meter
SFC 3	CTRL_RTM	Control run-time meter
SFC 4	READ_RTM	Read run-time meter
SFC 5	GADR_LGC	Logical address of a channel
SFC 6	RD_SINFO	Read start information
SFC 7	DP_PRAL	Triggering a hardware interrupt on the DP master
SFC 12	D_ACT_DP	Activating and deactivating of DP slaves
SFC 13	DPNRM_DG	Read diagnostic data of a DP salve
SFC 14	DPRD_DAT	Read consistent data
SFC 15	DPWR_DAT	Write consistent data
SFC 17	ALARM_SQ	ALARM_SQ
SFC 18	ALARM_SQ	ALARM_S
SFC 19	ALARM_SC	Acknowledgement state last alarm
SFC 20	BLKMOV	Block move
SFC 21	FILL	Fill a field
SFC 22	CREAT_DB	Create a data block
SFC 23	DEL_DB	Deleting a data block
SFC 24	TEST_DB	Test data block
SFC 25	COMPRESS	Compressing the User Memory
SFC 28	SET_TINT	Set time-of-day interrupt
SFC 29	CAN_TINT	Cancel time-of-day interrupt
SFC 30	ACT_TINT	Activate time-of-day interrupt

SFC	Name	Description
SFC 31	QRY_TINT	Query time-of-day interrupt
SFC 32	SRT_DINT	Start time-delay interrupt
SFC 33	CAN_DINT	Cancel time-delay interrupt
SFC 34	QRY_DINT	Query time-delay interrupt
SFC 36	MSK_FLT	Mask synchronous errors
SFC 37	MSK_FLT	Unmask synchronous errors
SFC 38	READ_ERR	Read error register
SFC 39	DIS_IRT	Disabling interrupts
SFC 40	EN_IRT	Enabling interrupts
SFC 41	DIS_AIRT	Delaying interrupts
SFC 42	EN_AIRT	Enabling delayed interrupts
SFC 43	RE_TRIGR	Re-trigger the watchdog
SFC 44	REPL_VAL	Replace value to ACCU1
SFC 46	STP	STOP the CPU
SFC 47	WAIT	Delay the application program
SFC 49	LGC_GADR	Read the slot address
SFC 51	RDSYSST	Read system status list SSL
SFC 52	WR_USMSG	Write user entry into diagnostic buffer
SFC 53	µS_TICK	Time measurement
SFC 54	RD_DPARM	Reading predefined parameters
SFC 55	WR_PARM	Write dynamic parameter
SFC 56	WR_DPARM	Write default parameter
SFC 57	PARM_MOD	Parametrize module
SFC 58	WR_REC	Write record set
SFC 59	RD_REC	Read record set
SFC 64	TIME_TCK	Read system time tick
SFC 65	X_SEND	Sending data
SFC 66	X_RCV	Receiving data
SFC 67	X_GET	Read data
SFC 68	X_PUT	Write data
SFC 69	X_ABORT	Disconnect
SFC 70	GEO_LOG	Determining the start address of a module
SFC 71	LOG_GEO	Determining the slot belonging to a logical address
SFC 75	SET_ADDR	Set PROFIBUS MAC address
SFC 81	UBLKMOV	Copy data area without gaps
SFC 101	HTL_RTM	Handling runtime meters

SFC	Name	Description
SFC 102	RD_DPARA	Reading predefined parameters
SFC 105	READ_SI	Reading dynamic system resources
SFC 106	DEL_SI	Releasing dynamic system resources
SFC 107	ALARM_DQ	ALARM_DQ
SFC 108	ALARM_DQ	ALARM_DQ
SFC 192	CP_S_R	CP communication
SFC 193	AI_OSZI	Oscilloscope-/FIFO function
SFC 194	DP_EXCH	Data exchange with CP 342S
SFC 195	FILE_ATT	Change file attributes
SFC 208	FILE_OPN	Open file
SFC 209	FILE_CRE	Create file
SFC 210	FILE_CLO	Close file
SFC 211	FILE_RD	Read file
SFC 212	FILE_WR	Write file
SFC 213	FILE_SEK	Position pointer
SFC 214	FILE_REN	Rename file
SFC 215	FILE_DEL	Delete file
SFC 216	SER_CFG	Parametrization PtP
SFC 217	SER_SND	Send to PtP
SFC 218	SER_RCV	Receive from PtP
SFC 219	CAN_TLGR	CANopen communication
SFC 227	TD_PRM	Parameterization of a text display
SFC 253	IBS_ACC	IBS communication
SFC 254	RW_SBUS	IBS communication

C SSL partial list

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

SSL-ID	SSL partial list
xy11h	Module identification
xy12h	CPU characteristics
xy13h	User memory areas
xy14h	System areas
xy15h	Block Types
xy19h	Status of all LEDs
xy1Ch	Identification of the component
xy22h	Interrupt status
xy32h	Communication status data
xy37h	Ethernet details of the module
xy74h	Status of the LEDs
xy91h	Status information CPU
xy92h	Stations status information (DPM)
xy94h	Stations status information (DPM, PROFINET-IO and EtherCAT)
xy96h	Module status information (PROFIBUS DP, PROFINET-IO, EtherCAT)
xyA0h	Diagnostic buffer of the CPU
xyB1h	Module diagnostic information (record set 0)
xyB2h	Module diagnostic information (record set 1) via physical address
xyB3h	Module diagnostic information (record set 1) via logical address
xyB4h	Diagnostic data of a DP slave
xyE0h	Information EtherCAT master/slave
xyE1h	EtherCAT bus system