

System 300S⁺

CPU-SC | 314-6CG23 | Manual

HB140 | CPU-SC | 314-6CG23 | en | 26-10

SPEED7 CPU 314SC/DPM



YASKAWA Europe GmbH
Philipp-Reis-Str. 6
65795 Hattersheim
Germany
Tel.: +49 6196 569-300
Fax: +49 6196 569-398
Email: info@yaskawa.eu
Internet: www.yaskawa.eu.com

Table of contents

| | | |
|----------|---|-----------|
| 1 | General | 7 |
| 1.1 | About this manual | 7 |
| 1.2 | Copyright © YASKAWA Europe GmbH | 8 |
| 1.3 | Safety instructions | 9 |
| 2 | Basics | 12 |
| 2.1 | Safety notes for the user | 12 |
| 2.2 | Operating structure of a CPU | 13 |
| 2.2.1 | General | 13 |
| 2.2.2 | Applications | 13 |
| 2.2.3 | Operands | 13 |
| 2.3 | CPU 314-6CG23 | 15 |
| 2.4 | General data | 16 |
| 2.4.1 | Use in difficult operating conditions | 17 |
| 3 | Assembly and installation guidelines | 18 |
| 3.1 | Installation dimensions | 18 |
| 3.2 | Assembly standard bus | 18 |
| 3.3 | Cabling | 21 |
| 3.4 | Industrial security and installation guidelines | 23 |
| 3.4.1 | Industrial security in information technology | 23 |
| 3.4.2 | Installation guidelines | 25 |
| 4 | Hardware description | 28 |
| 4.1 | Properties | 28 |
| 4.2 | Structure | 29 |
| 4.2.1 | General | 29 |
| 4.2.2 | Interfaces | 30 |
| 4.2.3 | In-/Output area CPU 314-6CG23 | 32 |
| 4.2.4 | Memory management | 39 |
| 4.2.5 | Slot for storage media | 39 |
| 4.2.6 | Battery backup for clock and RAM | 39 |
| 4.2.7 | Operating mode switch | 39 |
| 4.2.8 | LEDs | 40 |
| 4.3 | Technical data | 42 |
| 5 | Deployment CPU 314-6CG23 | 51 |
| 5.1 | Assembly | 51 |
| 5.2 | Start-up behavior | 51 |
| 5.3 | Addressing | 52 |
| 5.3.1 | Overview | 52 |
| 5.3.2 | Addressing Backplane bus I/O devices | 52 |
| 5.4 | Address assignment | 54 |
| 5.5 | Hardware configuration - CPU | 55 |

| | | |
|----------|--|-----------|
| 5.6 | Hardware configuration - I/O modules. | 55 |
| 5.7 | Hardware configuration - Ethernet PG/OP channel. | 56 |
| 5.8 | Hardware configuration - Communication. | 57 |
| 5.9 | CPU parametrization. | 58 |
| 5.9.1 | Parametrization via Siemens CPU. | 58 |
| 5.9.2 | CPU parameters. | 58 |
| 5.10 | Setting product specific CPU parameters. | 61 |
| 5.10.1 | Proceeding. | 61 |
| 5.10.2 | Function RS485. | 62 |
| 5.11 | Project transfer. | 64 |
| 5.11.1 | Transfer via MPI/PROFIBUS. | 64 |
| 5.11.2 | Transfer via Ethernet. | 66 |
| 5.11.3 | Transfer via memory card. | 66 |
| 5.12 | Accessing the web server. | 67 |
| 5.13 | Operating modes. | 73 |
| 5.13.1 | Overview. | 73 |
| 5.13.2 | Function security. | 75 |
| 5.14 | Overall reset. | 76 |
| 5.15 | Firmware update. | 77 |
| 5.16 | Reset to factory settings. | 80 |
| 5.17 | Deployment storage media - MMC, MCC. | 81 |
| 5.18 | Extended know-how protection. | 83 |
| 5.19 | CMD - auto commands. | 84 |
| 5.20 | Diagnostic entries. | 85 |
| 5.21 | Control and monitoring of variables with test functions. | 86 |
| 6 | Deployment I/O periphery. | 88 |
| 6.1 | Overview. | 88 |
| 6.2 | In-/Output area CPU 314-6CG23. | 88 |
| 6.3 | Address assignment. | 95 |
| 6.4 | Analog part. | 96 |
| 6.4.1 | Access to the I/O part. | 96 |
| 6.4.2 | Analog value representation | 98 |
| 6.4.3 | Analog part - Wiring. | 100 |
| 6.4.4 | Analog part - Measurement principle. | 102 |
| 6.4.5 | Analog part - Parameterization. | 102 |
| 6.5 | Digital part. | 105 |
| 6.5.1 | Access to the I/O part. | 108 |
| 6.5.2 | Digital part - Parameterization. | 109 |
| 6.6 | Counter. | 110 |
| 6.6.1 | Counter - Fast introduction. | 110 |
| 6.6.2 | SFB 47 - COUNT - Counter controlling. | 114 |

| | | |
|----------|--|------------|
| 6.6.3 | Counter - Functions. | 119 |
| 6.6.4 | Counter - Additional functions. | 127 |
| 6.7 | Frequency measurement. | 134 |
| 6.7.1 | Overview. | 134 |
| 6.7.2 | Inputs for the frequency measurement. | 134 |
| 6.7.3 | Parameterization. | 136 |
| 6.7.4 | SFB 48 - FREQUENC - Frequency measurement. | 137 |
| 6.8 | Pulse width modulation - PWM. | 140 |
| 6.8.1 | Overview. | 140 |
| 6.8.2 | Parameterization. | 142 |
| 6.8.3 | SFB 49 - PULSE - Pulse width modulation. | 143 |
| 6.9 | Diagnostic and interrupt. | 151 |
| 6.9.1 | Process interrupt. | 151 |
| 6.9.2 | Diagnostic interrupt. | 153 |
| 7 | Deployment PtP communication. | 161 |
| 7.1 | Fast introduction. | 161 |
| 7.2 | Principle of the data transfer. | 162 |
| 7.3 | Deployment of RS485 interface for PtP. | 162 |
| 7.4 | Parametrization. | 164 |
| 7.4.1 | FC/SFC 216 - SER_CFG - Parametrization PtP. | 164 |
| 7.5 | Communication. | 165 |
| 7.5.1 | FC/SFC 217 - SER_SND - Send to PtP. | 165 |
| 7.5.2 | FC/SFC 218 - SER_RCV - Receive from PtP. | 165 |
| 7.6 | Protocols and procedures | 165 |
| 7.7 | Modbus - Function codes | 169 |
| 7.8 | Modbus - Example communication. | 172 |
| 8 | Deployment PROFIBUS communication. | 175 |
| 8.1 | Overview. | 175 |
| 8.2 | Fast introduction. | 175 |
| 8.3 | Hardware configuration - CPU. | 177 |
| 8.4 | Deployment as PROFIBUS DP master. | 177 |
| 8.5 | Deployment as PROFIBUS DP slave. | 178 |
| 8.6 | PROFIBUS installation guidelines. | 180 |
| 8.7 | Commissioning and Start-up behavior. | 183 |
| 9 | WinPLC7. | 185 |
| 9.1 | System conception. | 185 |
| 9.2 | Installation. | 185 |
| 9.3 | Example project engineering. | 187 |
| 9.3.1 | Job definition. | 187 |
| 9.3.2 | Project engineering. | 187 |
| 9.3.3 | Test the PLC program in the <i>Simulator</i> | 193 |
| 9.3.4 | Transfer PLC program to CPU and its execution. | 195 |

| | | |
|-----------|--|------------|
| 10 | Configuration with TIA Portal. | 196 |
| 10.1 | TIA Portal - Work environment | 196 |
| 10.1.1 | General | 196 |
| 10.1.2 | Work environment of the TIA Portal | 196 |
| 10.2 | TIA Portal - Hardware configuration - CPU | 197 |
| 10.3 | TIA Portal - Hardware configuration - I/O modules | 199 |
| 10.4 | TIA Portal - Hardware configuration - Ethernet PG/OP channel | 200 |
| 10.5 | TIA Portal - Include Controls Library | 202 |
| 10.6 | TIA Portal - Project transfer | 203 |
| 11 | Appendix. | 205 |
| | Content | 207 |
| A | System specific event IDs | 209 |
| B | Integrated blocks | 248 |
| C | SSL partial list | 251 |

1 General

1.1 About this manual

Objective and contents

This manual describes the SPEED7 CPU-SC 314-6CG23 of the System 300S.

- It describes the structure, configuration and application.
- The manual is targeted at users with good basic knowledge in automation technology.
- The manual does not replace sufficient basic knowledge of automation technology or sufficient familiarity with the specific product.
- The manual consists of chapters. Each chapter describes a completed topic.
- For guidance, the manual provides:
 - An overall table of contents at the beginning of the manual
 - References with page numbers

Validity of the documentation

| Product | Order number | as of version: | | |
|-----------|--------------|----------------|----------------|----------------|
| 314SC/DPM | 314-6CG23 | CPU-HW: 01 | CPU-FW: V3.7.3 | DPM-FW: V3.3.6 |

Documentation

In the context of the use of the pertinent Yaskawa product, the manual is to be made accessible to the pertinent qualified personnel in:

- Project engineering
- Installation department
- Commissioning
- Operation

Icons and 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.

1.2 Copyright © YASKAWA Europe GmbH

All rights reserved

This document contains protected information of Yaskawa and may not be disclosed or used outside of an agreement made in advance with Yaskawa and only in accordance with that agreement.

This document is protected by copyright laws. Reproduction, distribution, or modification of this document or excerpts thereof is not permitted without the written consent of Yaskawa and the owner of this document, except in accordance with applicable agreements, contracts or licenses.

For permission to reproduce or distribute, please contact: YASKAWA Europe GmbH, European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany

Tel.: +49 6196 569 300

Fax.: +49 6196 569 398

E-mail: info@yaskawa.eu

Internet: www.yaskawa.eu.com

Download Center

By entering the product order number in the '*Download Center*' at www.yaskawa.eu.com, the pertinent manuals, data sheets, declarations of conformity, certificates and other helpful information for your product can be found.

Trademarks

VIPA, SLIO, System 300S and SPEED7 are registered trademarks of YASKAWA Europe GmbH.

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

PROFINET and PROFIBUS are registered trademarks of PROFIBUS and PROFINET International (PI).

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

All Microsoft Windows, Office and Server products mentioned are registered trademarks of Microsoft Inc., USA.

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

General terms of use

Every effort was made by Yaskawa to ensure that the information contained in this document was complete and correct at the time of publication. Nevertheless, the information contained therein is only owed by Yaskawa as it is available at Yaskawa. Correctness is not assured by Yaskawa, the right to change the information contained herein is always reserved by Yaskawa. There is no obligation to inform the customer of any changes. The customer is requested to actively keep this documentation up to date. The use of the products covered by these instructions, together with the associated documentation, is always at the customer's own risk, in accordance with the applicable guidelines and standards. This documentation describes the hardware and software components and functions of the product. It is possible that units are described which the customer does not have. The exact scope of delivery is described in the respective purchase contract.

Terminology '*Master*' and '*Slave*' as technical terms

The terms '*Master*' and '*Slave*' were established in many technical areas decades ago and are used in a clearly defined way, which is independent of their historical meaning.

In this manual, the terms '*Master*' and '*Slave*' are used in a purely functional and non-judgemental way. These terms are well established in numerous international regulations, standards and documentations. Switching to alternative terms could lead to misunderstandings and ambiguities in application.

We continue to use the terms '*Master*' and '*Slave*', but are aware of their origins and are committed to reflective, non-discriminatory communication.

Document support

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

Email: Documentation.HER@yaskawa.eu

Technical support

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

YASKAWA Europe GmbH,
European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany
Tel.: +49 6196 569 500 (hotline)
Email: support@yaskawa.eu

1.3 Safety instructions

General safety instructions**DANGER****Danger to life due to non-compliance with safety instructions**

Non-compliance with the safety instructions in the manual can result in serious injury or death. The manufacturer is not responsible for any injuries or damage to the equipment.

**CAUTION****Before commissioning and operating the components described in this manual, it is essential to note the following:**

- Modifications to the automation system must only be done in a voltage-free state!
- Connection and modification only by trained electricians
- National regulations and guidelines in the respective country of use must be observed and complied with (installation, protective measures, EMC, etc.)

Intended use

- It is the customer's responsibility to comply with all pertinent standards, codes, or regulations applicable to the use of the product, including those that apply when the Yaskawa product is used in combination with other products.
- The customer must confirm that the Yaskawa product is suitable for the customer's plant, machinery and equipment.
- If the Yaskawa product is used in a manner not specified by this manual, the protection provided by the Yaskawa product may be impaired and the use may result in material or immaterial damage.
- Contact Yaskawa to determine whether use is permitted in the following applications. If the use in the respective application is permissible, the Yaskawa product is to be used by considering additional risk assessments and specifications, and safety measures are to be provided to minimise the dangers in the event of a fault. Special caution is required and protective measures must be taken in the case of:
 - Outdoor use, use with possible chemical contamination or electrical interference, or use under conditions or in environments which are not described in product catalogs or manuals.
 - Nuclear control systems, combustion systems, railway systems, aviation systems, automotive systems, medical devices, amusement machines and equipment that is specifically regulated by industry or government.
 - Systems, machines and devices that can pose a risk to life or property.
 - Systems that require a high degree of reliability, such as gas, water or electricity supply systems or systems that operate 24 hours a day.
 - Other systems that require a similarly high level of security.
- Never use the Yaskawa product in an application where failure of the product could cause serious danger to life, limb, health or property without first ensuring that the system is designed to provide the required level of safety with risk warnings and redundancy to avoid the realisation of such dangers and that the Yaskawa product is properly designed and installed.
- The connection examples and other application examples described in the product catalogs and manuals of Yaskawa are for reference purposes. Check the functionality and safety of the devices and systems actually to be used before using the Yaskawa product.
- To avoid accidental harm to third parties, read and understand all prohibitions on use and precautions, and operate the Yaskawa product correctly.

Field of application

- The Yaskawa product is not suited for use in life-support machines or systems.
- Please contact your Yaskawa representative or Yaskawa distributor if considering the use of the Yaskawa product for special purposes, such as machines or systems used in passenger cars, in medical, aircraft and aerospace applications, for power supply of networks, for electrical power distribution or for underwater applications.

**DANGER**

The device is not permitted for use

- in explosive environments (EX zone)

The system is designed and manufactured for proper use and use in accordance with the user manual and is designed for:

- Communication and process control
- general control and automation tasks
- for industrial use
- operation within the environmental conditions specified in the technical data
- installation in a cabinet

**DANGER**

If this Yaskawa product is used in applications where failure of the device can result in the loss of human life, a serious accident or physical injury, you must install appropriate safety devices.

- Death or serious injury can result if you do not install the safety devices properly.

Disclaimer

(1) The contractual and legal liability of Yaskawa and the legal representatives and vicarious agents of Yaskawa for compensation and reimbursement of expenses in relation to the content of this documentation is excluded or limited as follows:

a) For slightly negligent breaches of *Essential Contractual Duties* arising from the contractual obligation, for Yaskawa the amount of liability is limited to the foreseeable damage typical for the contract. '*Essential Contractual Duties*' are those duties that characterise the performance of the contract and on which the Yaskawa customer may reasonably rely.

(b) In each case, Yaskawa is not liable for (i) the slightly negligent breach of duties arising from the duties that are not *Essential Contractual Duties*, as well as (ii) force majeure, i.e. external events that have no operational connection and cannot be averted even by exercising the utmost care that can reasonably be expected.

(2) The aforementioned limitation of liability does not apply (i) in cases of mandatory statutory liability (in particular under the product liability law), (ii) if and to the extent that Yaskawa has assumed a guarantee or same as guaranteed procurement risk according to § 276 BGB, (iii) for culpably caused injuries to life, limb and/or health, also by representatives or vicarious agents, as well as (iv) in case of delay in the event of a fixed completion date.

(3) A reversal of the burden of proof is not associated with the provisions above.

Disposal

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

2 Basics

2.1 Safety notes for the user

Handling of electrostatic sensitive modules

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



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

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

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

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

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



CAUTION

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

2.2 Operating structure of a CPU

2.2.1 General

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

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

Cyclic processing

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

Timer processing

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

Alarm controlled processing

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

Priority based processing

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

2.2.2 Applications

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

- System routine
- User application

System routine

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

User application

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

2.2.3 Operands

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

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

Process image and periphery

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

- individual Bits
- Bytes
- Words
- Double words

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

- Bytes
- Words
- Blocks

Bit Memory

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

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

Timers and counters

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

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

Data Blocks

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

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

2.3 CPU 314-6CG23

Overview

The CPU 314-6CG23 bases upon the SPEED7 technology. This supports the CPU at programming and communication by means of co-processors that causes a power improvement for highest needs.

- The CPU is programmed in STEP®7 from Siemens. For this you may use the SIMATIC Manager or TIA Portal from Siemens. Here the instruction set of the S7-400 from Siemens is used.
- Modules and CPUs of the System 300S from Yaskawa and Siemens may be used at the bus as a mixed configuration.
- The user application is stored in the battery buffered RAM or on an additionally pluggable MMC storage module.
- The CPU 314-6CG23 is configured as CPU 314C-2DP (6ES7 314-6CG03-0AB0 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 2MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 512kbyte
 - There is the possibility to extend the work memory to its maximum printed capacity 2MB by means of a memory extension card.

Integrated PROFIBUS DP master/slave respectively PtP functionality

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

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

Integrated Ethernet PG/OP channel

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

Operation Security

- Wiring by means of spring pressure connections (CageClamps) at the front connector
- Core cross-section 0.08...2.5mm²
- Total isolation of the wiring at module change
- Potential separation of all modules to the backplane bus

Dimensions/ Weight

Dimensions of the basic enclosure:

- 3tier width: (WxHxD) in mm: 120x125x120

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.

General data

2.4 General data

Conformity and approval

Conformity

| | | |
|----|------------|-----------------------|
| CE | 2014/35/EU | Low-voltage directive |
| | 2014/30/EU | EMC directive |

Approval

| | | |
|--------|------------|---|
| UL | | Refer to Technical data |
| others | | |
| RoHS | 2011/65/EU | Restriction of the use of certain hazardous substances in electrical and electronic equipment |

Protection of persons and device protection

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

Environmental conditions to EN 61131-2

Climatic

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

Mounting conditions

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

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

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

2.4.1 Use in difficult operating conditions



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

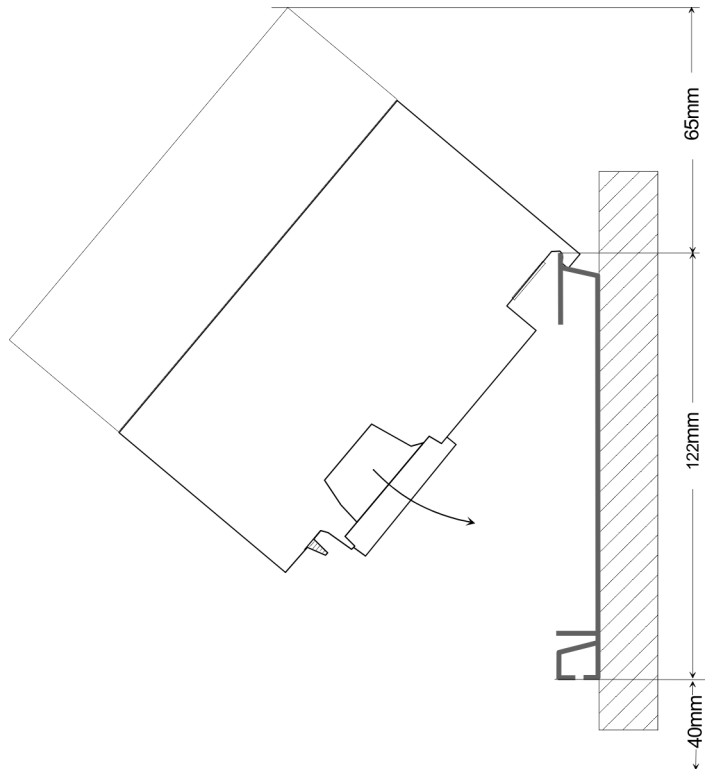
- dust generation
- chemically active substances (corrosive vapors or gases)
- strong electric or magnetic fields

3 Assembly and installation guidelines

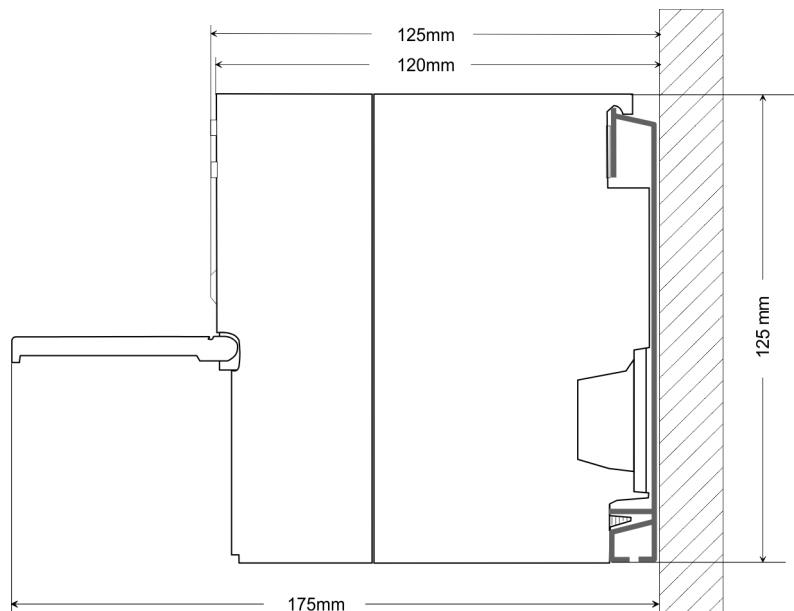
3.1 Installation dimensions

Dimensions Basic enclosure 3tier width (WxHxD) in mm: 120 x 125 x 120

Dimensions



Installation dimensions



3.2 Assembly standard bus

General

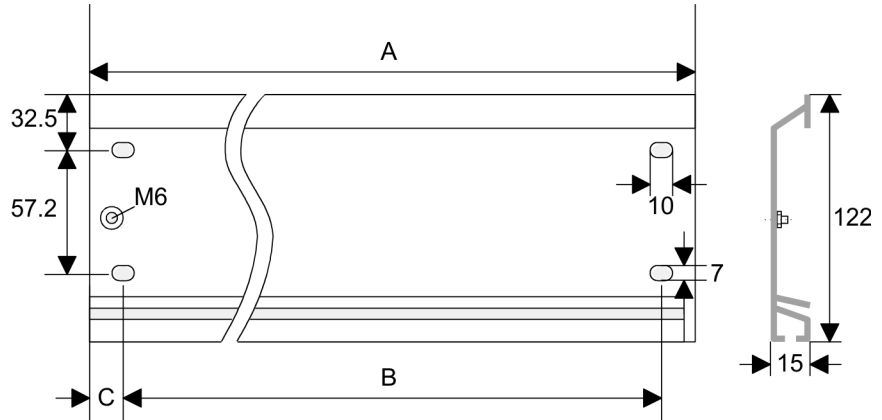
The single modules are directly installed on a profile rail and connected via the backplane bus connector. Before installing the modules you have to clip the backplane bus connector to the module from the backside. The backplane bus connector is delivered together with the peripheral modules.

Profile rail

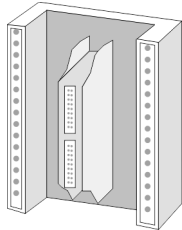
| Order number | A | B | C |
|--------------|------|---------------------|-----|
| 390-1AB60 | 160 | 140 | 10 |
| 390-1AE80 | 482 | 466 | 8.3 |
| 390-1AF30 | 530 | 500 | 15 |
| 390-1AJ30 | 830 | 800 | 15 |
| 390-9BC00* | 2000 | Drillings only left | 15 |

*) Unit pack: 10 pieces

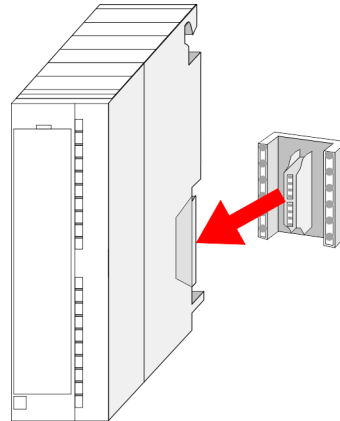
Measures in mm



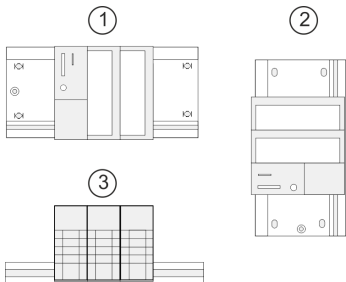
Bus connector



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

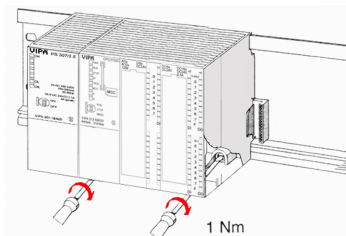
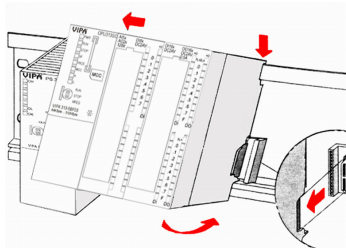
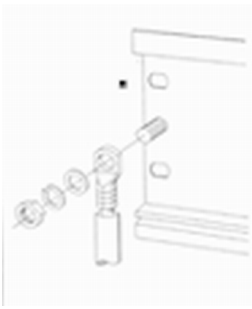


Please regard the allowed environment temperatures:

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

Assembly standard bus

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 10mm².
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



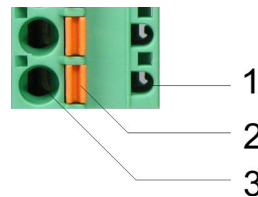
CAUTION

- 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!

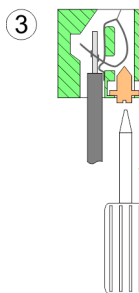
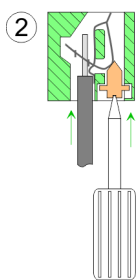
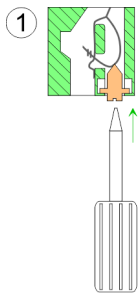
CageClamp technology (green)

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 0.08mm^2 to 2.5mm^2 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 suitable 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 0.08mm^2 to 2.5mm^2
3. → By removing the screwdriver the wire is connected safely with the plug connector via a spring.

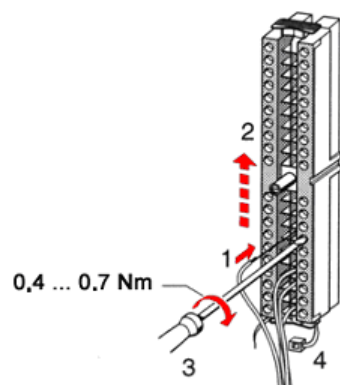
Cabling

Front connector I/O
periphery

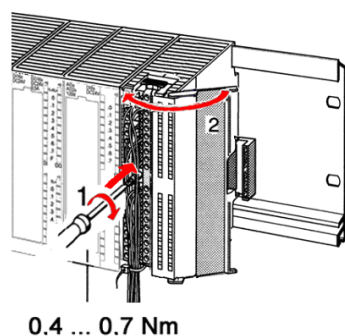


The 40pole front connector with the order number 392-1AM00 is in the delivery.

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 Industrial security and installation guidelines

3.4.1 Industrial security in information technology

Latest version

This chapter can also be found as a guide '*Industrial IT Security*' in the '*Download Center*' of www.yaskawa.eu.com

Hazards

The topic of data security and access protection has become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Hazards can arise from:

- Internal manipulation such as technical errors, operating and program errors and deliberate program or data manipulation.
- External manipulation such as software viruses, worms and trojans.
- Human carelessness such as password phishing.

Precautions

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

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

Further Information

You can find more information about the measures on the following websites:

- Federal Office for Information Technology → www.bsi.bund.de
- Cybersecurity & Infrastructure Security Agency → us-cert.cisa.gov
- VDI / VDE Society for Measurement and Automation Technology → www.vdi.de

3.4.1.1 Protection of hardware and applications

Precautions

- Do not integrate any components or systems into public networks.
 - Use VPN "Virtual Private Networks" for use in public networks. This allows you to control and filter the data traffic accordingly.
- Always keep your system up-to-date.
 - Always use the latest firmware version for all devices.
 - Update your user software regularly.
- Protect your systems with a firewall.
 - The firewall protects your infrastructure internally and externally.
 - This allows you to segment your network and isolate entire areas.
- Secure access to your plants via user accounts.
 - If possible, use a central user management system.
 - Create a user account for each user for whom authorization is essential.
 - Always keep user accounts up-to-date and deactivate unused user accounts.
- Secure access to your plants via secure passwords.
 - Change the password of a standard login after the first start.
 - Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
 - Change the passwords according to the rules and guidelines that apply to your application.
- Deactivate inactive communication ports respectively protocols.
 - Only the communication ports that are used for communication should be activated.
 - Only the communication protocols that are used for communication should be activated.
- Consider possible defence strategies when planning and securing the system.
 - The isolation of components alone is not sufficient for comprehensive protection. An overall concept is to be drawn up here, which also provides defensive measures in the event of a cyber attack.
 - Periodically carry out threat assessments. Among others, a comparison is made here between the protective measures taken and those required.
- Limit the use of external storage media.
 - Via external storage media such as USB memory sticks or SD memory cards, malware can get directly into a system while bypassing a firewall.
 - External storage media or their slots must be protected against unauthorized physical access, e.g. by using a lockable control cabinet.
 - Make sure that only authorized persons have access.
 - When disposing of storage media, make sure that they are safely destroyed.
- Use secure access paths such as HTTPS or VPN for remote access to your plant.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.

3.4.1.2 Protection of PC-based software

Precautions

Since PC-based software is used for programming, configuration and monitoring, it can also be used to manipulate entire systems or individual components. Particular caution is required here!

- Use user accounts on your PC systems.
 - If possible, use a central user management system.
 - Create a user account for each user for whom authorization is essential.
 - Always keep user accounts up-to-date and deactivate unused user accounts.
- Protect your PC systems with secure passwords.
 - Change the password of a standard login after the first start.
 - Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
 - Change the passwords according to the rules and guidelines that apply to your application.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.
- Protect your PC systems by security software.
 - Install virus scanners on your PC systems to identify viruses, trojans and other malware.
 - Install software that can detect phishing attacks and actively prevent them.
- Always keep your software up-to-date.
 - Update your operating system regularly.
 - Update your software regularly.
- Make regular backups and store the media at a safe place.
- Regularly restart your PC systems. Only boot from storage media that are protected against manipulation.
- Use encryption systems on your storage media.
- Perform security assessments regularly to reduce the risk of manipulation.
- Use only data and software from approved sources.
- Uninstall software which is not used.
- Disable unused services.
- Activate a password-protected screen lock on your PC systems.
- Always lock your PC systems as soon as you leave your PC workstation.
- Do not click any links that come from unknown sources. If necessary ask, e.g. on e-mails.
- Use secure access paths such as HTTPS or VPN for remote access to your PC system.

3.4.2 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 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 ground 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.
 - Connect all inactive metal extensive and impedance-low.
 - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
 - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
 - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
 - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).
- Proof the correct fixing of the lead isolation.
 - Data lines must be shielded.
 - Analog lines must be shielded. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
 - Cables for frequency inverters, servo and stepper motors must be shielded.
 - Lay the line isolation extensively on an isolation/protected ground conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
 - Make sure that the isolation/protected ground 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 ground 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 ground 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 ground 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.



CAUTION

Please regard at installation!

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

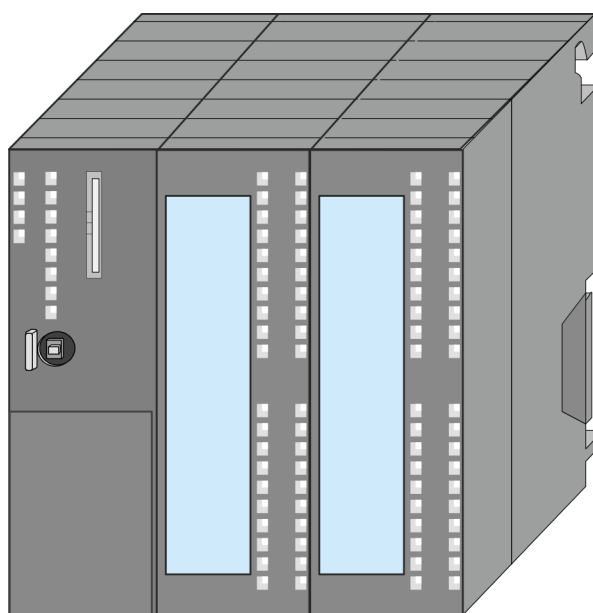
Remedy: Potential compensation line

4 Hardware description

4.1 Properties

CPU 314-6CG23

- SPEED7 technology integrated
- 512kbyte work memory integrated (256kbyte code, 256kbyte data)
- Work memory expandable to max. 2MB (1MB code, 1MB data)
- 2MB load memory
- PROFIBUS DP master integrated supported DP-V0, DP-V1
- 1. interface configurable for MPI, PROFIBUS DP master- or PtP-communication
- 2. interface configurable for PROFIBUS DP master- or PtP-communication
- RS485 interface for PtP communication
- Ethernet PG/OP interface integrated
- Slot for external memory cards (lockable)
- Status LEDs for operating state and diagnostics
- Real-time clock battery buffered
- Digital I/Os: DI 24xDC24V, DO 16xDC24V, 0.5A, DIO 8xDC24V
- Analog I/Os: AI 4x12Bit / AO 2x12Bit / AI 1xPt100
- 4 channels for counter, frequency measurement and pulse width modulation
- I/O address range digital/analog 1024byte
- 512 timer / 512 counter / 8192 flag byte



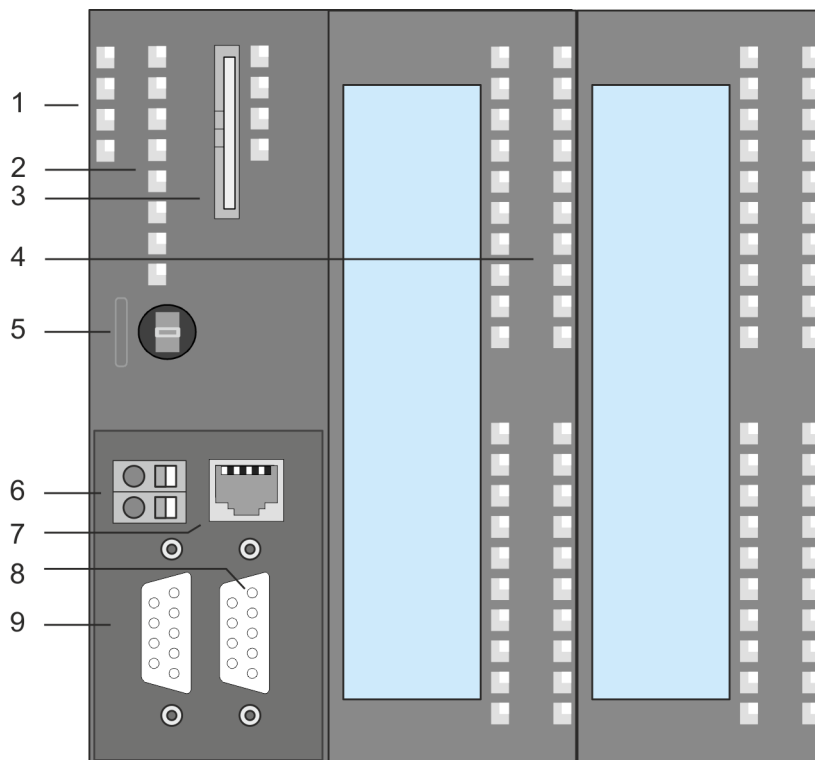
Ordering data

| Type | Order no. | Description |
|-----------|-----------|--|
| 314SC/DPM | 314-6CG23 | MPI/PtP/PB-DP master interface, card slot, Real-time clock, Ethernet interface for PG/OP, PtP/PB-DPM interface, DI 24xDC24V / DO 16xDC 24V, 0.5A / DIO 8x DC24V, 0.5A, AI 4x12Bit / AO 2x12Bit / AI 1xPt100, 4 channels technological function |

4.2 Structure

4.2.1 General

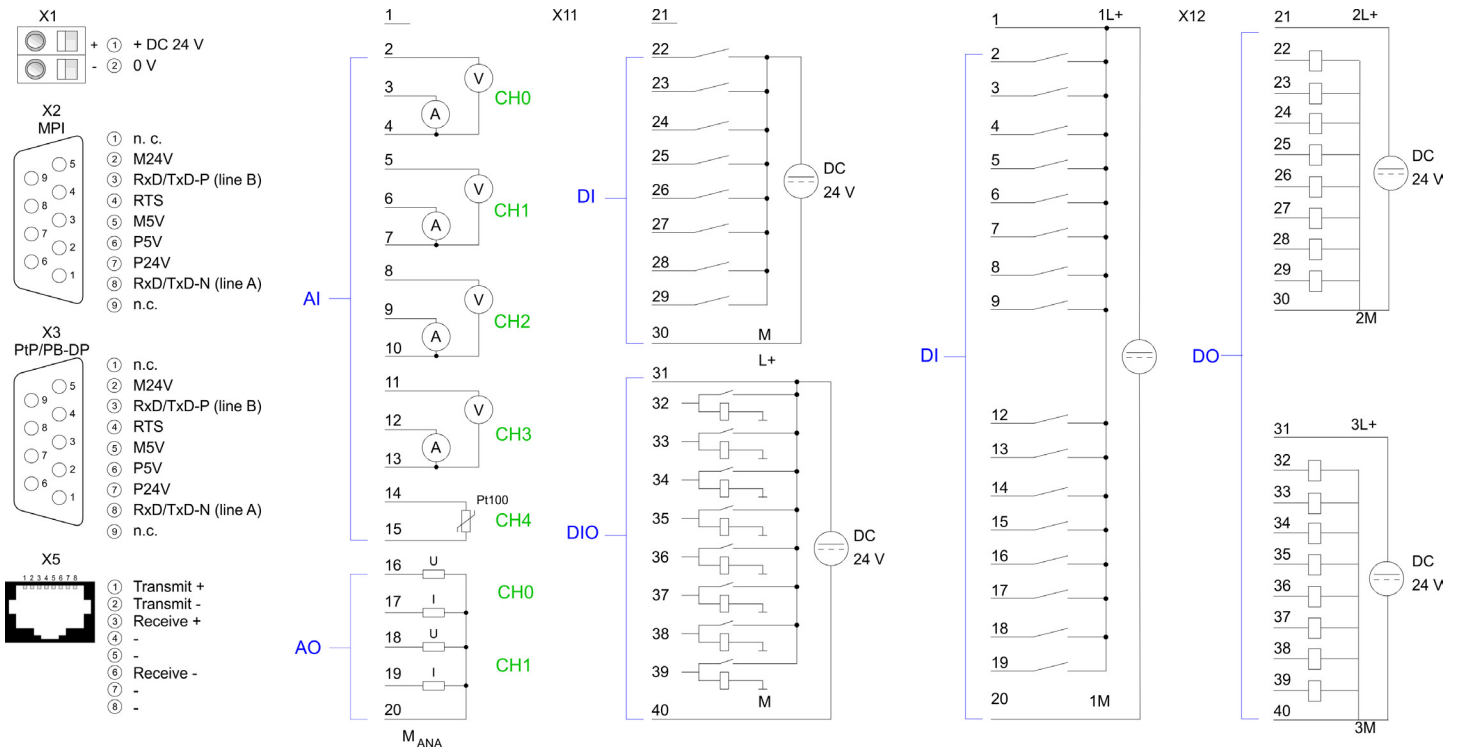
CPU 314-6CG23



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

The components 6 - 9 are under the front flap!

4.2.2 Interfaces



X1: Power supply

The CPU has an integrated power supply:

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

X2: MPI/PtP/PB-DP interface

9pin SubD jack:

Per default, the RS485 interface X2 of the CPU is used for MPI.

The functionality of this interface may be configured at the virtual SPEED-Bus by means of the parameter "Function RS485 ..." of the hardware configuration.

i Please note that by de-activating the MPI functionality, Siemens CP and FM modules, which require the K bus, can no longer be used!

■ MPI functionality

- 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.

- 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).
- PROFIBUS functionality
 - Using the PROFIBUS functionality the integrated PROFIBUS DP master is connected to PROFIBUS via RS485 interface.
 - At master operation there is access to up to 124 DP slaves.
 - For this the project engineering happens in the hardware configurator from Siemens.
 - Please regard there may be a delimitation of the maximum number of configurable DP slaves by the use of the Siemens SIMATIC manager.



You can only use one functionality with MPI, PtP respectively PROFIBUS.

X3: PtP/PB-DP interface

9pin SubD jack:

Per default, the RS485 interface X3 of the CPU is used for the PROFIBUS DP master.

The functionality of this interface may be configured at the virtual SPEED-Bus by means of the parameter "Function RS485 X3" of the hardware configuration.

- PROFIBUS functionality
 - Using the PROFIBUS functionality the integrated PROFIBUS DP master is connected to PROFIBUS via RS485 interface.
 - At master operation there is access to up to 124 DP slaves.
 - For this the project engineering happens in the hardware configurator from Siemens.
 - Please regard there may be a delimitation of the maximum number of configurable DP slaves by the use of the Siemens SIMATIC manager.
- 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).



You can only use one interface with PtP respectively PROFIBUS functionality.

Structure > In-/Output area CPU 314-6CG23

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 314-6CG23

Overview

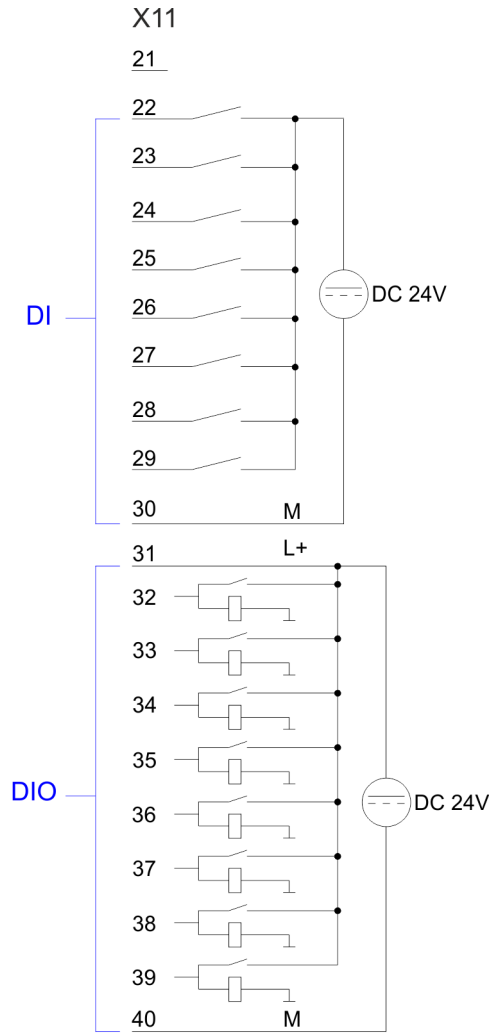
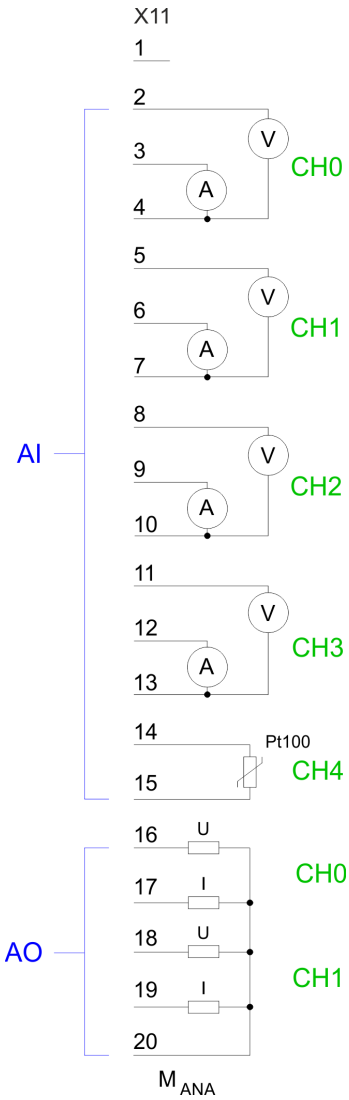
The CPU 314-6CG23 has the following analog and digital in- and output ranges integrated in one casing:

- Analog Input: 4xU/Ix12Bit, 1xPt100
- Analog Output: 2xU/Ix12Bit
- Digital Input: 24xDC 24V
- Digital Output: 16xDC 24V, 0.5A
- Digital In-/Output: 8xDC 24V, 0.5A
- Technological functions: 4 channels

The analog channels of the module are isolated to the back plane via DC/DC transducer and optocouplers.

Via the parameterization you may assign alarm properties to the digital inputs of X12.

Additionally the digital inputs of X12 are parameterizable as counter.



CAUTION

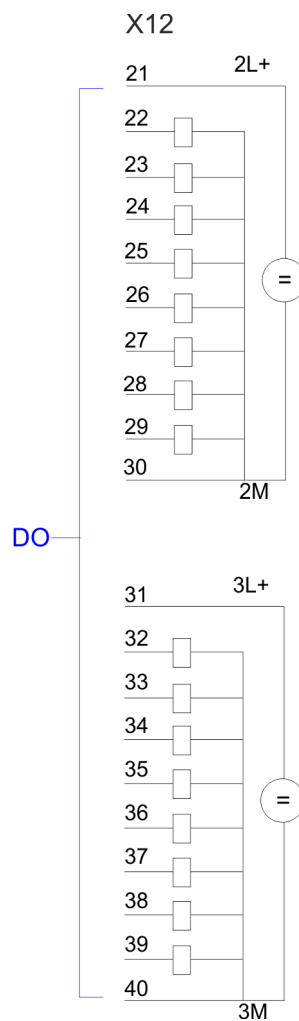
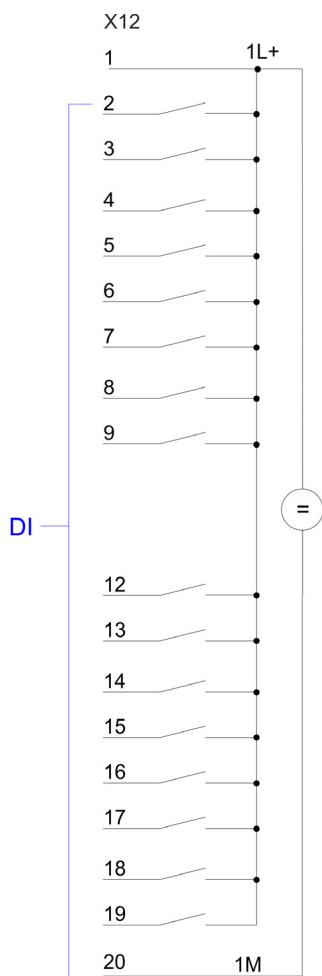
Temporarily not used analog inputs with activated channel must be connected to the concerning ground. To avoid measuring errors, you should connect only one measuring type per channel.



CAUTION

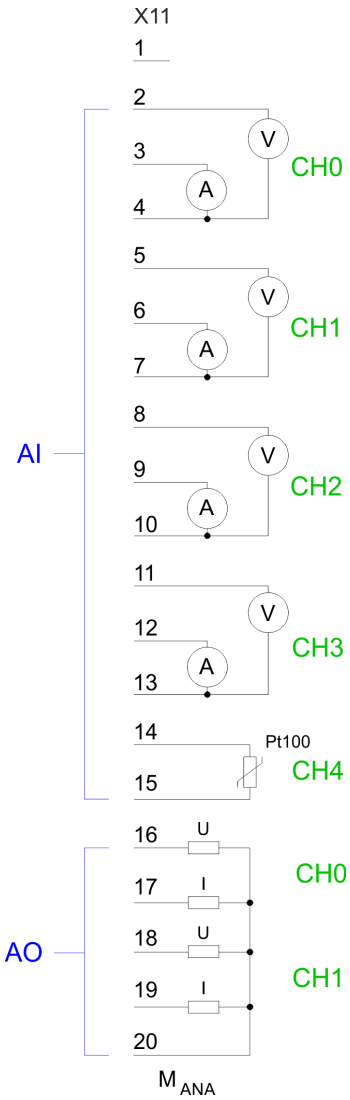
Please regard that the voltage at an output channel is always \leq the supply voltage connected to L+. Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal. A thus connected output remains active even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

Structure > In-/Output area CPU 314-6CG23



CAUTION

Please regard that the voltage at an output channel is always \leq the supply voltage connected to L+.



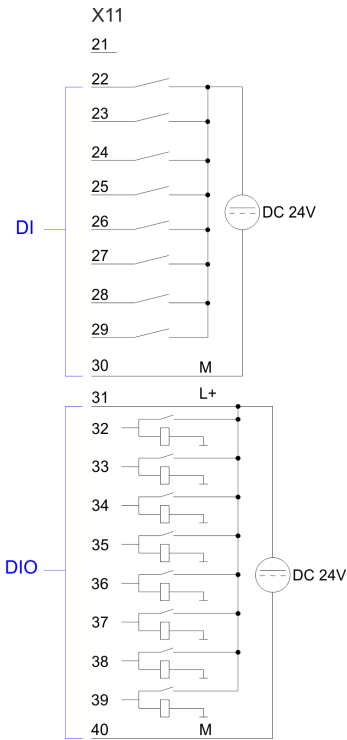
Pin assignment X11: AIO

| Pin | Assignment |
|-----|-------------------------------|
| 1 | not connected |
| 2 | Voltage measurement channel 0 |
| 3 | Current measurement channel 0 |
| 4 | Ground channel 0 |
| 5 | Voltage measurement channel 1 |
| 6 | Current measurement channel 1 |
| 7 | Ground channel 1 |
| 8 | Voltage measurement channel 2 |
| 9 | Current measurement channel 2 |
| 10 | Ground channel 2 |
| 11 | Voltage measurement channel 3 |
| 12 | Current measurement channel 3 |
| 13 | Ground channel 3 |
| 14 | Pt 100 channel 4 |
| 15 | Pt 100 channel 4 |
| 16 | Voltage output channel 0 |
| 17 | Current output channel 0 |
| 18 | Voltage output channel 1 |
| 19 | Current output channel 1 |
| 20 | Ground AO channel 0 and 1 |

Status indicator X11: AIO

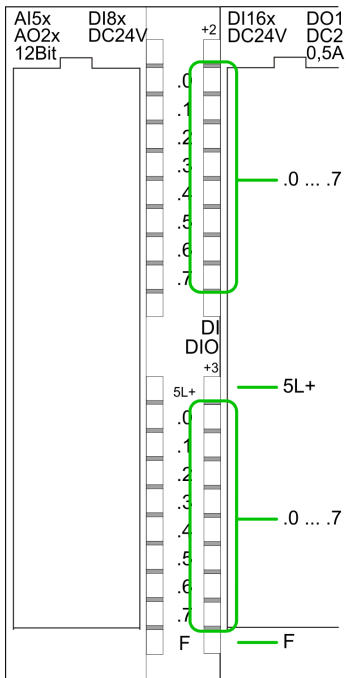
From the analog part no LEDs are controlled.

Structure > In-/Output area CPU 314-6CG23



Pin assignment X11: DI

| Pin | Assignment |
|-----|-------------------------|
| 21 | not connected |
| 22 | I+2.0 |
| 23 | I+2.1 |
| 24 | I+2.2 |
| 25 | I+2.3 |
| 26 | I+2.4 |
| 27 | I+2.5 |
| 28 | I+2.6 |
| 29 | I+2.7 |
| 30 | Ground 1M DI |
| 31 | L+ Power supply +DC 24V |
| 32 | I/O+3.0 |
| 33 | I/O+3.1 |
| 34 | I/O+3.2 |
| 35 | I/O+3.3 |
| 36 | I/O+3.4 |
| 37 | I/O+3.5 |
| 38 | I/O+3.6 |
| 39 | I/O+3.7 |
| 40 | Ground DIO |

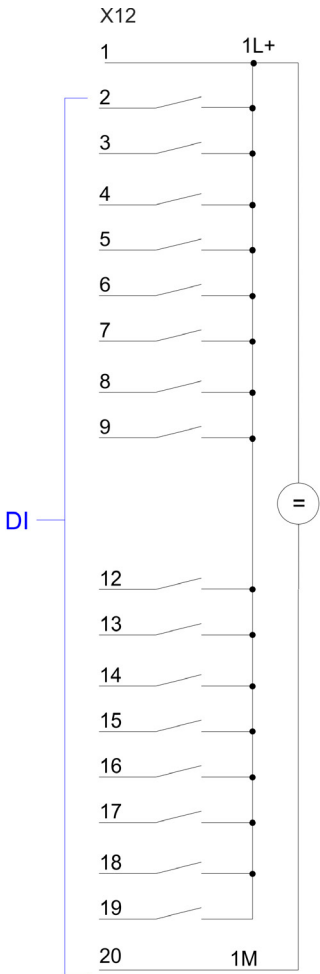


Status indication X11: DI

- .07
 - LEDs (green)
 - I+2.0 ... I+2.7
 - Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated

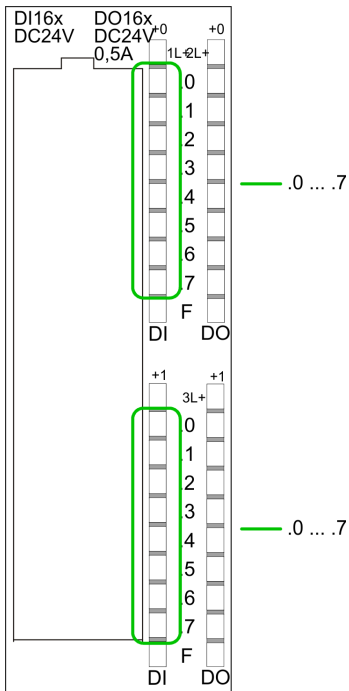
DIO

- 5L+
 - LEDs (green)
 - Supply voltage available for DIO
- .0... .7
 - LEDs (green)
 - I/Q+3.0 ... I/Q+3.7
 - On at active input resp. output
- F
 - LED (red)
 - Overload or short circuit error



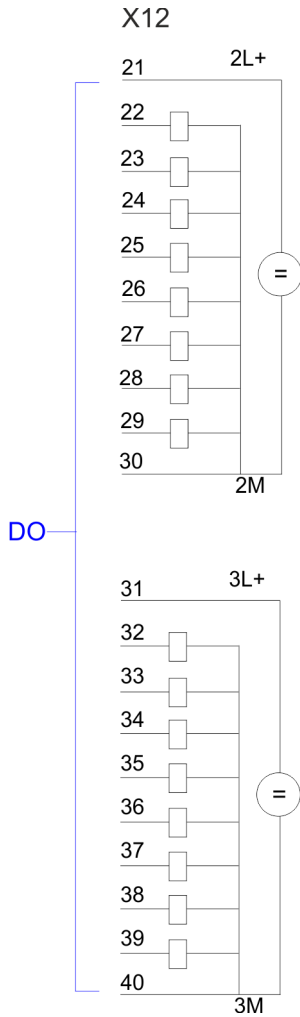
Pin assignment X12: DI

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



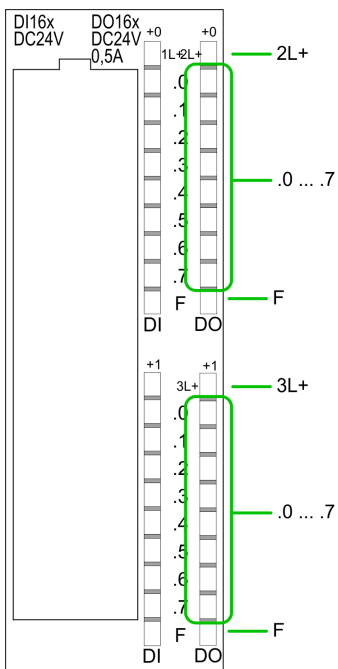
Status indication X12: DI

- .07
 - 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 X12: 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 / Channel 2 Output |
| 25 | Q+0.3 / Channel 3 Output |
| 26 | Q+0.4 |
| 27 | Q+0.5 |
| 28 | Q+0.6 |
| 29 | Q+0.7 |
| 30 | Ground 2M DO |
| 31 | 3L+ Power supply +DC 24V |
| 32 | Q+1.0 |
| 33 | Q+1.1 |
| 34 | Q+1.2 |
| 35 | Q+1.3 |
| 36 | Q+1.4 |
| 37 | Q+1.5 |
| 38 | Q+1.6 |
| 39 | Q+1.7 |
| 40 | Ground 3M DO |



Status indication X12: DO

- 2L+, 3L+
 - LED (green)
 - Supply voltage available for DO
- .07
 - LEDs (green)
 - Q+0.0 ... Q+0.7
 - Q+1.0 ... Q+1.7
 - 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 2MB
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 512kbyte
 - There is the possibility to extend the work memory to its maximum printed capacity 2MB by means of a memory extension card.

4.2.5 Slot for storage media

At this slot the following storage media can be plugged:

- SD respectively MCC (**M**ultimedia **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. → [‘Deployment storage media - MMC, MCC’...page 81](#)
 - To activate the corresponding card is to be installed and an *Overall reset* is to be established. → [‘Overall reset’...page 76](#)

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
































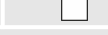















































RUN
STOP
MR

- 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.










Structure > LEDs

4.2.8 LEDs

LEDs CPU

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

Ethernet PG/OP channel





































| L/A (Link/Activity)  green | S (Speed)  green | Meaning |
|---|---|--|
|  | X | The Ethernet PG/OP channel is physically connected to Ethernet. |
|  | X | There is no physical connection. |
|  flickers | X | Shows Ethernet activity. |
|  |  | The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 100Mbit. |
|  |  | The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 10Mbit. |

not relevant: X







LEDs PROFIBUS/PtP interface X3

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

Master operation

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

Slave operation

| RN (RUN)  green | ER (ERR)  red | DE  green | IF  red | Meaning |
|--|--|---|---|--|
|  |  |  |  | Slave has no configuration respectively PtP is active. |
|  2Hz |  |  |  | Slave is without master. |
|  2Hz |  |  2Hz |  | Alternate blinking at configuration faults. |
|  |  |  |  | Slave exchanges data with the master. |

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. | 314-6CG23 |
|---|----------------------|
| Type | CPU 314SC DPM |
| SPEED-Bus | - |
| Technical data power supply | |
| Power supply (rated value) | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ |
| Current consumption (no-load operation) | 350 mA |
| Current consumption (rated value) | 1 A |
| Inrush current | 11 A |
| I ² t | 0.7 A ² s |
| Max. current drain at backplane bus | 3 A |
| Max. current drain load supply | - |
| Power loss | 14 W |
| Technical data digital inputs | |
| Number of inputs | 24 |
| Cable length, shielded | 1000 m |
| Cable length, unshielded | 600 m |
| Rated load voltage | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ |
| Current consumption from load voltage L+ (without load) | 70 mA |
| Rated value | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V |
| Input voltage hysteresis | - |
| Signal logic input | Sinking input |
| 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 | 24 |
| Number of simultaneously utilizable inputs vertical configuration | 24 |
| Input characteristic curve | IEC 61131-2, type 1 |
| Initial data size | 3 Byte |
| Technical data digital outputs | |
| Number of outputs | 16 |
| Cable length, shielded | 1000 m |

| | |
|---|----------------------------------|
| Order no. | 314-6CG23 |
| 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 |
| Signal logic output | Sourcing output |
| Output current, permitted range to 40°C | 5 mA to 0.6 A |
| Output current, permitted range to 60°C | 5 mA to 0.6 A |
| Output current at signal "0" max. (residual current) | 0.5 mA |
| Output delay of "0" to "1" | 100 µs |
| Output delay of "1" to "0" | 100 µs |
| Minimum load current | - |
| Lamp load | 5 W |
| Parallel switching of outputs for redundant control of a load | possible |
| Parallel switching of outputs for increased power | not possible |
| Actuation of digital input | ✓ |
| Switching frequency with resistive load | max. 2.5 kHz |
| Switching frequency with inductive load | max. 0.5 Hz |
| Switching frequency on lamp load | max. 2.5 kHz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic |
| Trigger level | 1 A |
| Number of operating cycle of relay outputs | - |
| Switching capacity of contacts | - |
| Output data size | 2 Byte |
| Technical data analog inputs | |
| Number of inputs | 5 |
| Cable length, shielded | 200 m |
| Rated load voltage | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ |
| Current consumption from load voltage L+ (without load) | - |
| Voltage inputs | ✓ |
| Min. input resistance (voltage range) | 100 kΩ |
| Input voltage ranges | -10 V ... +10 V 0 V ... +10 V |
| Operational limit of voltage ranges | +/-0.3% |
| Operational limit of voltage ranges with SFU | - |
| Basic error limit voltage ranges | +/-0.2% |
| Basic error limit voltage ranges with SFU | - |
| Destruction limit voltage | max. 30V |
| Current inputs | ✓ |
| Max. input resistance (current range) | 100 Ω |

Technical data

| | |
|---|--|
| Order no. | 314-6CG23 |
| Input current ranges | -20 mA ... +20 mA 0 mA ... +20 mA +4 mA ... +20 mA |
| Operational limit of current ranges | +/-0.3% |
| Operational limit of current ranges with SFU | - |
| Basic error limit current ranges | +/-0.2% |
| Radical error limit current ranges with SFU | - |
| Destruction limit current inputs (electrical current) | max. 50mA |
| Destruction limit current inputs (voltage) | max. 30V |
| Resistance inputs | ✓ |
| Resistance ranges | 0 ... 600 Ohm |
| Operational limit of resistor ranges | +/-0.4% |
| Operational limit of resistor ranges with SFU | - |
| Basic error limit | +/-0.2% |
| Basic error limit with SFU | - |
| Destruction limit resistance inputs | max. 15V |
| Resistance thermometer inputs | - |
| Resistance thermometer ranges | Pt100 |
| Operational limit of resistance thermometer ranges | +/-0.6% |
| Operational limit of resistance thermometer ranges with SFU | - |
| Basic error limit thermoresistor ranges | +/-0.4% |
| Basic error limit thermoresistor ranges with SFU | - |
| Destruction limit resistance thermometer inputs | max. 15V |
| Thermocouple inputs | - |
| Thermocouple ranges | - |
| Operational limit of thermocouple ranges | - |
| Operational limit of thermocouple ranges with SFU | - |
| Basic error limit thermocouple ranges | - |
| Basic error limit thermocouple ranges with SFU | - |
| Destruction limit thermocouple inputs | - |
| Programmable temperature compensation | - |
| External temperature compensation | - |
| Internal temperature compensation | - |
| Technical unit of temperature measurement | °C, °F, K |
| Resolution in bit | 12 |
| Measurement principle | successive approximation |
| Basic conversion time | 0.5 ms |
| Noise suppression for frequency | 80 dB |
| Initial data size | 10 Byte |
| Technical data analog outputs | |
| Number of outputs | 2 |
| Cable length, shielded | 200 m |
| Rated load voltage | - |
| Reverse polarity protection of rated load voltage | - |
| Current consumption from load voltage L+ (without load) | - |
| Voltage output short-circuit protection | ✓ |

| | |
|--|--|
| Order no. | 314-6CG23 |
| Voltage outputs | ✓ |
| Min. load resistance (voltage range) | 1 kΩ |
| Max. capacitive load (current range) | 1 μF |
| Max. inductive load (current range) | 25 mA |
| Output voltage ranges | -10 V ... +10 V 0 V ... +10 V |
| Operational limit of voltage ranges | +/-0.2% |
| Basic error limit voltage ranges with SFU | +/-0.1% |
| Destruction limit against external applied voltage | max. 16V (30V / 10s) |
| Current outputs | ✓ |
| Max. in load resistance (current range) | 500 Ω |
| Max. inductive load (current range) | 10 mH |
| Typ. open circuit voltage current output | 15 V |
| Output current ranges | -20 mA ... +20 mA 0 mA ... +20 mA +4 mA ... +20 mA |
| Operational limit of current ranges | +/-0.3% |
| Radical error limit current ranges with SFU | +/-0.2% |
| Destruction limit against external applied voltage | max. 16V (30V / 10s) |
| Settling time for ohmic load | 0.5 ms |
| Settling time for capacitive load | 0.5 ms |
| Settling time for inductive load | 0.5 ms |
| Resolution in bit | 12 |
| Conversion time | 1 ms |
| Substitute value can be applied | no |
| Output data size | 4 Byte |
| Technical data counters | |
| Number of counters | 4 |
| Counter width | 32 Bit |
| Maximum input frequency | 60 kHz |
| Maximum count frequency | 60 kHz |
| Mode incremental encoder | ✓ |
| Mode pulse / direction | ✓ |
| Mode pulse | ✓ |
| Mode frequency counter | ✓ |
| Mode period measurement | ✓ |
| Gate input available | ✓ |
| Latch input available | ✓ |
| Reset input available | - |
| Counter output available | ✓ |
| Load and working memory | |
| Load memory, integrated | 2 MB |
| Load memory, maximum | 2 MB |
| Work memory, integrated | 512 KB |
| Work memory, maximal | 2 MB |
| Memory divided in 50% program / 50% data | ✓ |
| Memory card slot | SD/MMC-Card with max. 2 GB |

Technical data

| | |
|--|------------------------|
| Order no. | 314-6CG23 |
| Hardware configuration | |
| Racks, max. | 4 |
| Modules per rack, max. | 8 |
| Number of integrated DP master | 1 |
| Number of DP master via CP | 4 |
| Operable function modules | 8 |
| Operable communication modules PtP | 8 |
| Operable communication modules LAN | 8 |
| Status information, alarms, diagnostics | |
| Status display | yes |
| Interrupts | yes |
| Process alarm | yes |
| 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 |
| Max. potential difference between inputs (U _{cm}) | - |
| Max. potential difference between Mana and Mintern (U _{iso}) | - |
| Max. potential difference between inputs and Mana (U _{cm}) | - |
| Max. potential difference between inputs and Mintern (U _{iso}) | - |
| Max. potential difference between Mintern and outputs | - |
| Insulation tested with | DC 500 V |
| Command processing times | |
| Bit instructions, min. | 0.01 µs |
| Word instruction, min. | 0.01 µs |
| Double integer arithmetic, min. | 0.01 µs |
| Floating-point arithmetic, min. | 0.06 µs |
| Timers/Counters and their retentive characteristics | |
| Number of S7 counters | 512 |
| S7 counter remanence | adjustable 0 up to 256 |
| S7 counter remanence adjustable | C0 .. C7 |
| Number of S7 times | 512 |
| S7 times remanence | adjustable 0 up to 256 |
| S7 times remanence adjustable | not retentive |
| Data range and retentive characteristic | |
| Number of flags | 8192 Byte |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 |
| Number of data blocks | 4095 |
| Max. data blocks size | 64 KB |

| | |
|---|------------------|
| Order no. | 314-6CG23 |
| 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 |
| 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 | 7856 |
| Digital outputs | 7904 |
| Digital inputs central | 979 |
| Digital outputs central | 986 |
| Integrated digital inputs | 24 |
| | 32 |
| Integrated digital outputs | 16 |
| | 24 |
| Analog inputs | 494 |
| Analog outputs | 495 |
| Analog inputs, central | 253 |
| Analog outputs, central | 250 |
| Integrated analog inputs | 5 |
| Integrated analog outputs | 2 |
| Communication functions | |
| PG/OP channel | ✓ |
| Global data communication | ✓ |
| Number of GD circuits, max. | 4 |
| Size of GD packets, max. | 22 Byte |
| S7 basic communication | ✓ |
| 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 | |

Technical data

| | |
|---------------------------------------|-----------------------------------|
| Order no. | 314-6CG23 |
| PWM channels | 4 |
| PWM time basis | 0.1 ms / 1 ms |
| 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 | ✓ |
| MP ² I (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 |
| Functionality X3 | |
| Type | X3 |
| Type of interface | RS485 |
| Connector | Sub-D, 9-pin, female |
| Electrically isolated | ✓ |
| MPI | - |
| MP ² I (MPI/RS232) | - |
| DP master | yes |
| DP slave | yes |
| 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 | - |
| Transmission speed, min. | 19.2 kbit/s |
| Transmission speed, max. | 187.5 kbit/s |
| Functionality PROFIBUS master | |
| Number of connections, max. | 32 |
| PG/OP channel | ✓ |
| Routing | ✓ |
| S7 basic communication | ✓ |
| S7 communication | ✓ |
| S7 communication as server | ✓ |
| S7 communication as client | - |

| Order no. | 314-6CG23 |
|---|----------------------|
| Activation/deactivation of DP slaves | ✓ |
| Direct data exchange (slave-to-slave communication) | - |
| DPV1 | ✓ |
| Transmission speed, min. | 9.6 kbit/s |
| Transmission speed, max. | 12 Mbit/s |
| Number of DP slaves, max. | 32 |
| Address range inputs, max. | 1 KB |
| Address range outputs, max. | 1 KB |
| User data inputs per slave, max. | 244 Byte |
| User data outputs per slave, max. | 244 Byte |
| Functionality PROFIBUS slave | |
| Number of connections, max. | 32 |
| PG/OP channel | ✓ |
| Routing | ✓ |
| S7 communication | ✓ |
| S7 communication as server | ✓ |
| S7 communication as client | - |
| Direct data exchange (slave-to-slave communication) | - |
| DPV1 | ✓ |
| Transmission speed, min. | 9.6 kbit/s |
| Transmission speed, max. | 12 Mbit/s |
| Automatic detection of transmission speed | - |
| Transfer memory inputs, max. | 244 Byte |
| Transfer memory outputs, max. | 244 Byte |
| Address areas, max. | 32 |
| User data per address area, max. | 32 Byte |
| Functionality RJ45 interfaces | |
| Type | X5 |
| Type of interface | Ethernet 10/100 MBit |
| Connector | RJ45 |
| Electrically isolated | ✓ |
| PG/OP channel | ✓ |
| Number of connections, max. | 4 |
| Productive connections | - |
| 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 | ✓ |

Technical data

| | |
|---------------------------------|--------------------------|
| Order no. | 314-6CG23 |
| RK512 protocol | - |
| USS master protocol | ✓ |
| Modbus master protocol | ✓ |
| Modbus slave protocol | - |
| Special protocols | - |
| Housing | |
| Material | PPE |
| Mounting | Rail System 300 |
| Mechanical data | |
| Dimensions (WxHxD) | 120 mm x 125 mm x 120 mm |
| Net weight | 610 g |
| Weight including accessories | - |
| Gross weight | - |
| Environmental conditions | |
| Operating temperature | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C |
| Certifications | |
| UL certification | yes |
| KC certification | - |
| UKCA certification | - |
| ChinaRoHS certification | yes |

5 Deployment CPU 314-6CG23

5.1 Assembly



Information about assembly and cabling: → ['Assembly and installation guidelines'...page 18](#)

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

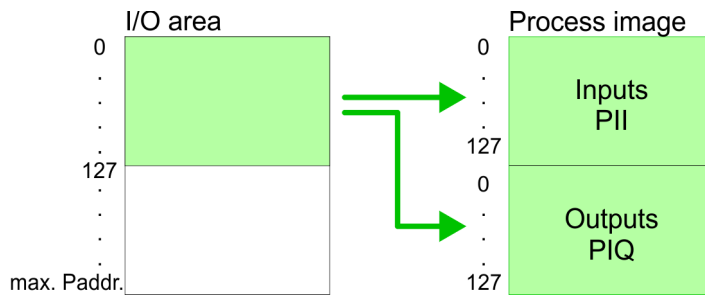
To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. At the start-up of the CPU, this assigns automatically peripheral addresses for digital in-/output modules starting with 0 and ascending depending on the slot location. If no hardware project 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 314-6CG23 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

At deployment of the CPU 314-6CG23 you may control up to 31 modules at the bus. Here the maximum of 8 modules per row may be parameterized.

For the project engineering of more than 8 modules line interface connections are to be used. For this you set in the hardware configurator the module IM 360 from the hardware catalog to slot 3 of your 1. profile rail. Now you may extend your system with up to 3 profile rails by starting each with an IM 361 from Siemens at slot 3.

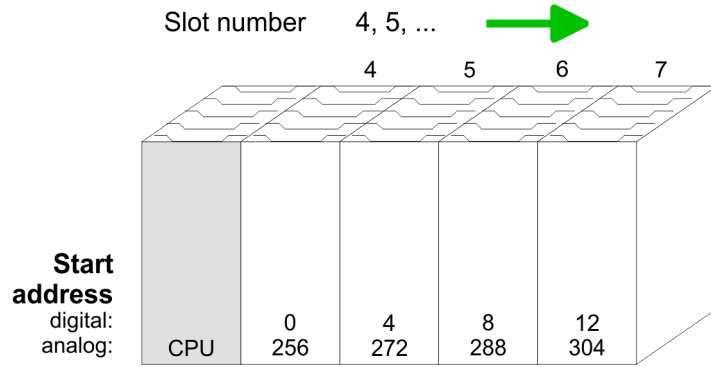
Define addresses by hardware configuration

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

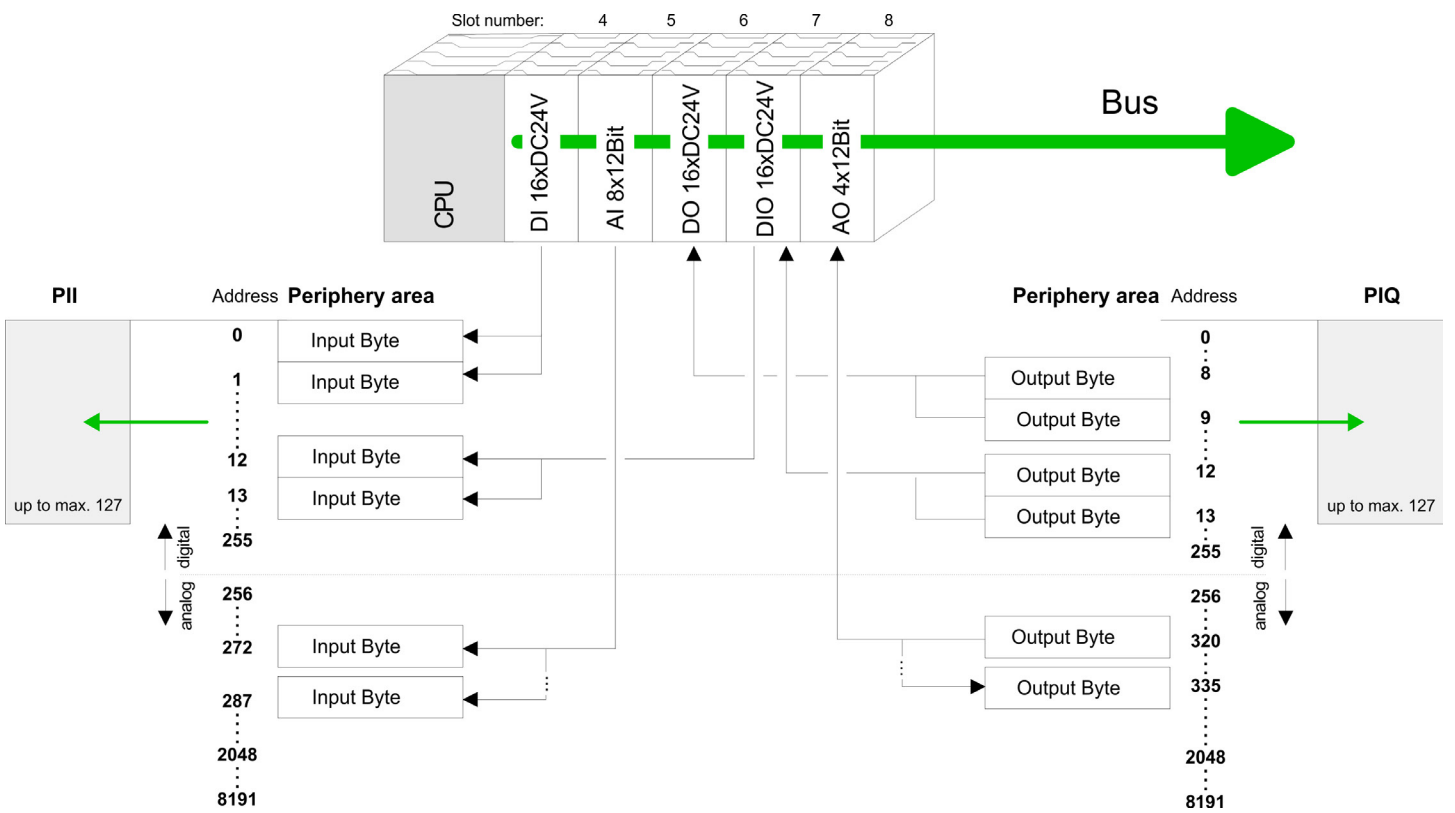
If you do not like to use a hardware configuration, an automatic addressing comes into force. At the automatic address allocation DIOs occupy depending on the slot location always 4byte and AIOs, FMs, CPs always 16byte at the bus. Depending on the slot location the start address from where on the according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

- DIOs: Start address = $4 \times (\text{slot} - 4)$
- AIOs, FMs, CPs: Start address = $16 \times (\text{slot} - 4) + 256$



Example for automatic address allocation

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



Address assignment

5.4 Address assignment

Input range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--|
| DI24/DO16 | 124 | Byte | Digital Input I+0.0 ... I+0.7 (X12) |
| | 125 | Byte | Digital Input I+1.0 ... I+1.7 (X12) |
| | 126 | Byte | Digital Input I+2.0 ... I+2.7 (X11) |
| | 127 | Byte | Digital Input I+3.0 ... I+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Input Channel 0 |
| | 754 | Word | Analog Input Channel 1 |
| | 756 | Word | Analog Input Channel 2 |
| | 758 | Word | Analog Input Channel 3 |
| | 760 | Word | Analog Input Channel 4 |
| Counter | 768 | DInt | Channel 0: Count value / Frequency value |
| | 772 | DInt | Channel 1: Count value / Frequency value |
| | 776 | DInt | Channel 2: Count value / Frequency value |
| | 780 | DInt | Channel 3: Count value / Frequency value |

Output range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--------------------------------------|
| DI24/DO16 | 124 | Byte | Digital Output Q+0.0 ... Q+0.7 (X12) |
| | 125 | Byte | Digital Output Q+1.0 ... Q+1.7 (X12) |
| | 126 | Byte | reserved |
| | 127 | Byte | Digital Output Q+3.0 ... Q+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Output Channel 0 |
| | 754 | Word | Analog Output Channel 1 |
| Counter | 768 | DWord | reserved |
| | 772 | DWord | reserved |
| | 776 | DWord | reserved |
| | 780 | DWord | reserved |



Please regard during hardware configuration due to the additional DIO channels the CPU 314-6CG23 occupies 1 more byte in the IO area (default address 127), each. It is not automatically detected if this byte was already assigned to a module in the hardware configuration.

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 Yaskawa manual "Operation list SPEED7" at "Differences between SPEED7 and 300V programming".

Proceeding

| Slot | Module |
|------|----------------------|
| 1 | |
| 2 | CPU 314C-2 DP |
| X2 | DP |
| 2.2 | DI24/DO16 |
| 2.3 | AI5/AO2 |
| 2.4 | Count |
| 2.5 | Position |
| 3 | |

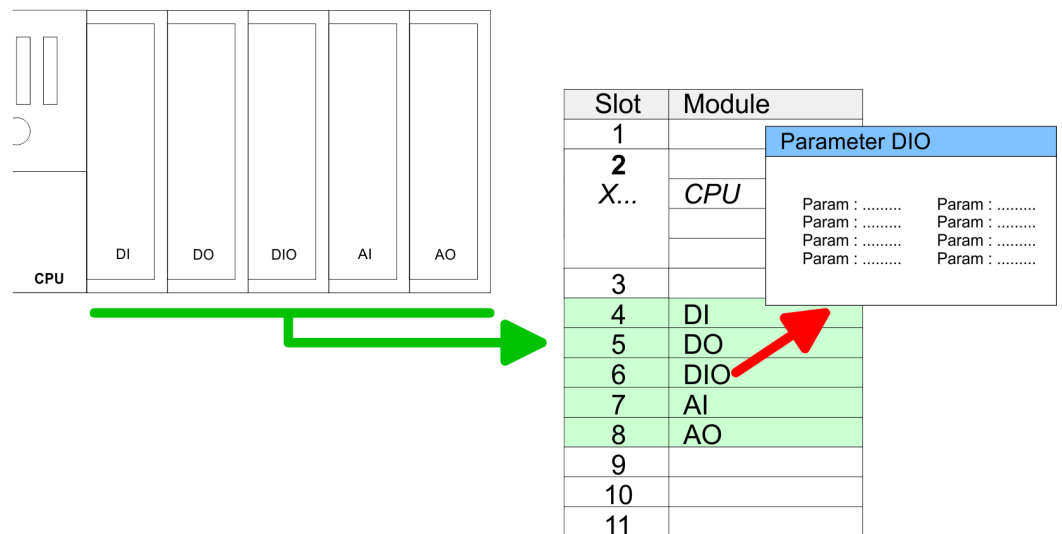
To be compatible with the Siemens SIMATIC Manager the following steps should be executed:

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 314C-2DP (6ES7 314-6CG03-0AB0 V2.6).
4. → The integrated PROFIBUS DP master is to be configured and connected via the sub module X2 (DP).

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.

Bus extension with IM 360 and IM 361

Maximally 31 modules may be addressed by the CPU 314-6CG23, but per row maximally 8 modules are supported. For project engineering the IM 360 of the hardware catalog are to be used as a bus extension during project engineering. Here 3 further extension racks can be connected via the IM 361. Bus extensions are always placed at slot 3.

5.7 Hardware configuration - Ethernet PG/OP channel

Overview

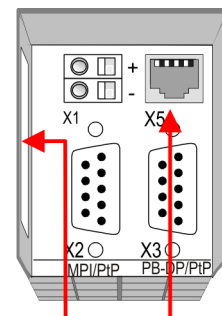
The CPU 314-6CG23 has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU. The PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc. With the first 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.

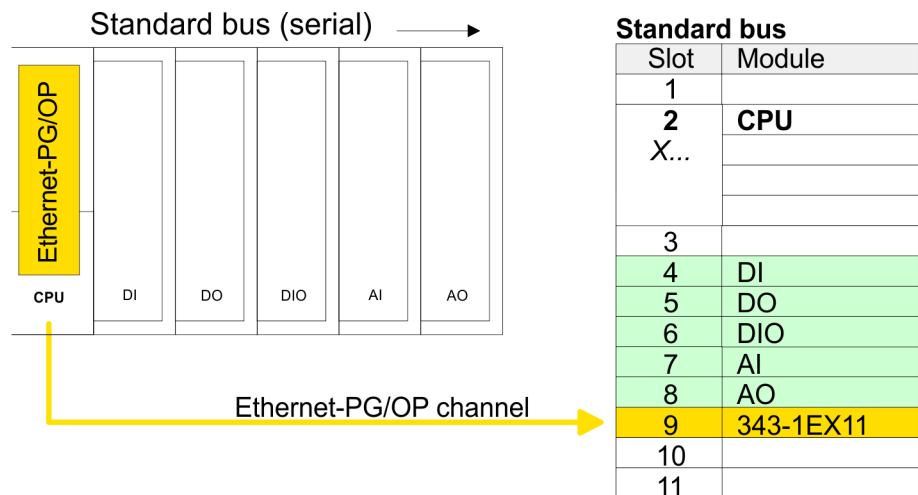
Assign IP address parameters

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 314C-2DP (6ES7 314-6CG03-0AB0 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 Hardware configuration - Communication**

The hardware configuration is described at the following pages:

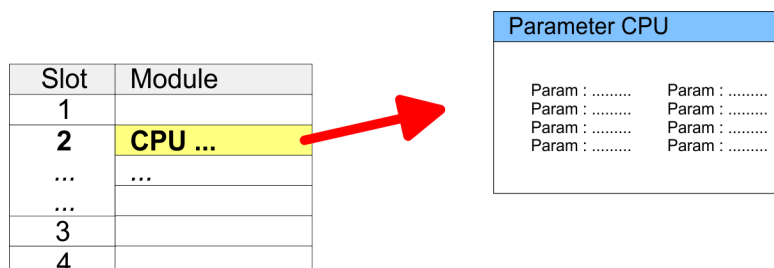
- ➔ ['Deployment as PROFIBUS DP master'...page 177](#)
- ➔ ['Deployment as PROFIBUS DP slave'...page 178](#)
- ➔ ['Deployment of RS485 interface for PtP'...page 162](#)

5.9 CPU parametrization

5.9.1 Parametrization via Siemens CPU

Parameterization via Siemens CPU 314C-2DP

Since the CPU 314-6CG23 is to be configured as Siemens CPU 314C-2DP (6ES7 314-6CG03-0AB0 V2.6) in the Siemens hardware configurator, the parameters of the CPU 314-6CG23 may be set with "Object properties" of the CPU 314C-2DP during hardware configuration. Via a double-click on the CPU 314C-2DP the parameter window of the CPU may be accessed. Using the registers you get access to every parameter of the CPU.



5.9.2 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 314-6CG03 is CPU 314C-2DP.
- 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 or if another type of module is inserted there instead, such as during an initial system start-up.
- **Monitoring time for ready message by modules [100ms]:**
This operation specifies the maximum time for the ready message of every configured module after PowerON. Here connected PROFIBUS DP slaves are also considered until they are parameterized. If the modules do not send a ready message to the CPU by the time the monitoring time has expired, the actual configuration becomes unequal to the preset configuration.
- **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 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

5.10 Setting product specific CPU parameters

5.10.1 Proceeding

Overview

Except of the product specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 314C-2DP from Siemens. With installing of the SPEEDBUS.GSD the product specific parameters may be set during hardware configuration. Here the following parameters may be accessed:

- Function RS485 (PtP, Synchronization DP master and CPU)
- Token Watch

Requirements

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








Installation of the SPEEDBUS.GSD

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

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

The GSD files may be found at www.yaskawa.eu.com in the '*Download Center*'.

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

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

Hardware configuration

The embedding of the CPU 314-6CG23 happens by means of a virtual PROFIBUS master system with the following approach:

Setting product specific CPU parameters > Function RS485

Standard bus

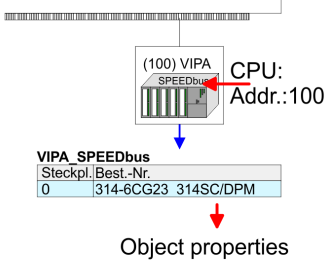
| Slot | Module |
|------|---------------------|
| 1 | |
| 2 | CPU 314C-2DP |
| X2 | |
| 2.2 | |
| 2.3 | |
| 2.4 | |
| 2.5 | |
| 3 | |

Modules at the bus

343-1EX11
Ethernet PG/OP

always as last module
342-5DA02 V5.0

virtual DP-Master for CPU



1. → Perform a hardware configuration for the CPU. → [‘Hardware configuration - CPU’...page 55](#)
2. → Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".
3. → Connect the slave system "VIPA_SPEEDbus". After installing the SPEEDBUS.GSD this may be found in the hardware catalog at Profibus-DP / Additional field devices / I/O / VIPA / VIPA_SPEEDBUS.
4. → For the slave system set the PROFIBUS address 100.
5. → Configure at slot 0 the CPU 314-6CG23 of the hardware catalog from VIPA_SPEEDbus.
6. → By double clicking the placed CPU 314-6CG23 the properties dialog of the CPU may be opened.

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

5.10.2 Function RS485

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

| | |
|----------------------------|---|
| Deactivated | Deactivates the RS485 interface. |
| PtP | With this operating mode the PROFIBUS DP master is deactivated and the RS485 interface acts as an interface for serial point-to-point communication. Here data may be exchanged between two stations by means of protocols. |
| PROFIBUS DP async | PROFIBUS DP master operation asynchronous to CPU cycle The RS485 interface is preset at default to PROFIBUS DP async. Here CPU cycle and cycles of every PROFIBUS DP master run independently. |
| PROFIBUS DP syncIn | The CPU is waiting for DP master input data. |
| PROFIBUS DP syncOut | The DP master system is waiting for CPU output data. |
| PROFIBUS DP syncInOut | CPU and DP master system are waiting on each other and form thereby a cycle. |
| Default: PROFIBUS DP async | |

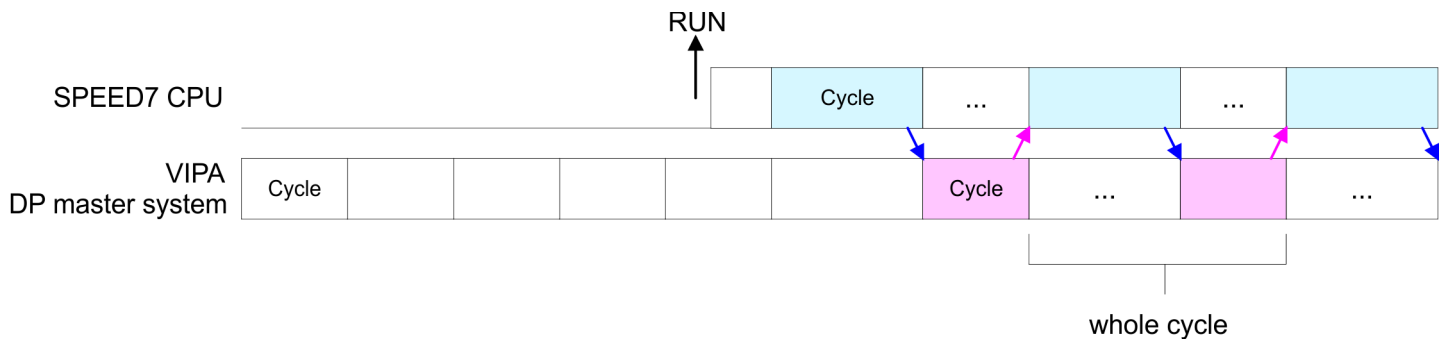
5.10.2.1 Synchronization between master system and CPU

Overview

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

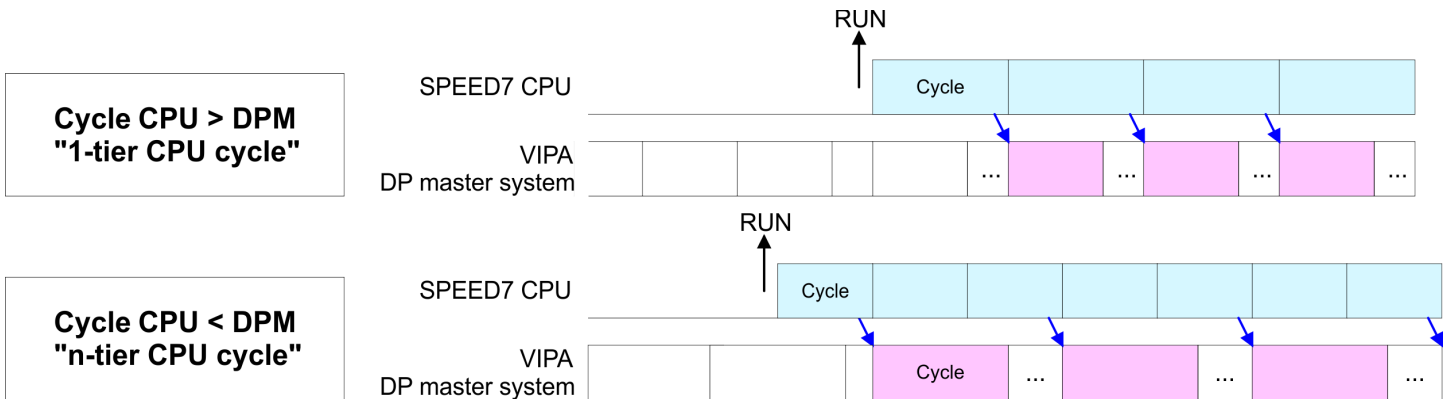
PROFIBUS DP SyncInOut

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



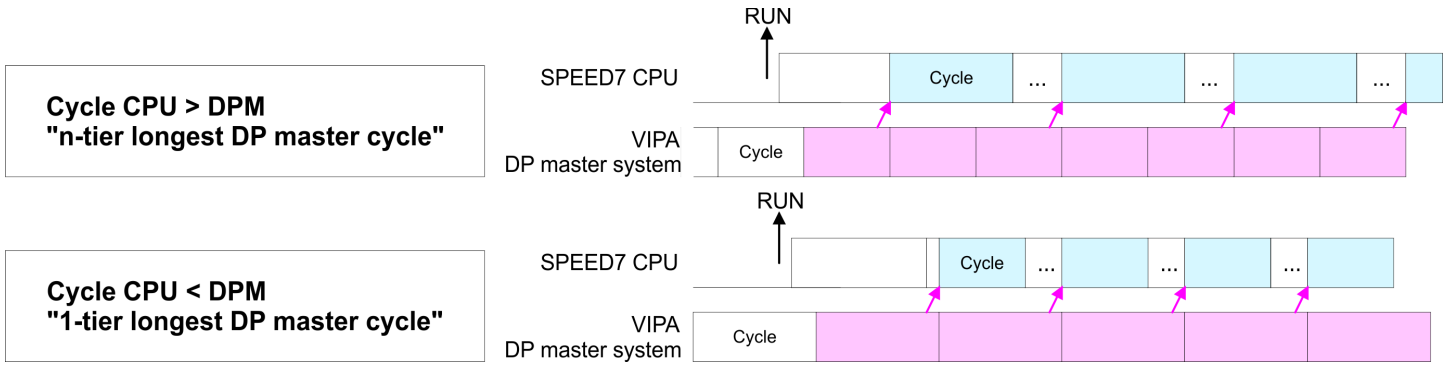
PROFIBUS DP SyncOut

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



PROFIBUS-DP SyncIn

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



5.11 Project transfer

Overview

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

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

5.11.1 Transfer via MPI/PROFIBUS

General

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

- X2: MPI interface
- X3: PROFIBUS interface

Net structure

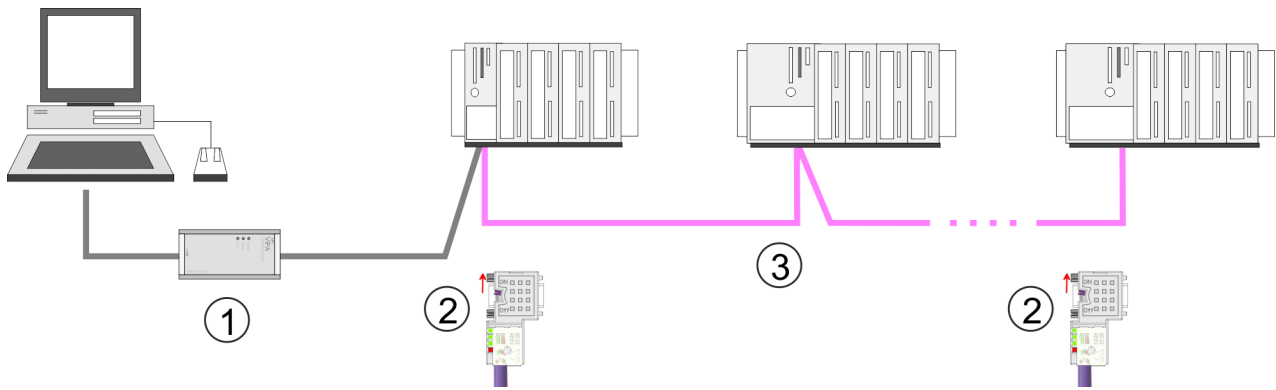
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

MPI programming cable

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

Terminating resistor

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



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

Proceeding transfer via MPI interface

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

Proceeding transfer via PROFIBUS interface

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



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

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

5.11.2 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".

→ ['Hardware configuration - Ethernet PG/OP channel'...page 56](#)

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.

5.11.3 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.

5.12 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. → ['Hardware configuration - Ethernet PG/OP channel'...page 56](#)

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

| Name | Value |
|---------------|-----------|
| Ordering Info | 314-6CG23 |
| Serial | 26164 |
| Version | 01V00 |
| HW Revision | 01 |
| Software | 3.5.9.14 |

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 | RS485 function X2 |
| X3 (RS485/COM2) | DPM-async | RS485 function X3 |
| X5 | PG/OP Ethernet Port | |
| Card Information | | |
| Type | SD | |
| Product S/N | 6BC34010 | |
| Size | 493617152 bytes | |
| Free | 492355584 bytes | |
| Active Feature Set Information | | |

Accessing the web server

| | | | |
|----------------------------------|--|--|--|
| Status | Memory Extension present | | |
| Memory Usage | | | |
| LoadMem | 0 / 1048576 Bytes | CPU: Information to memory configuration | |
| WorkMemCode | 0 / 278528 Bytes | | |
| WorkMemData | 0 / 278528 Bytes | | |
| PG/OP Network Information | | | |
| Device Name | VIPA 314-6CG23 CPU | Ethernet PG/OP channel: Address information | |
| IP Address | 172.16.129.210 | | |
| 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 Name, firmware version, package | |
| PRODUCT | VIPA 314-6CG23 V3.7.3 Px000278.pkg | | |
| HARDWARE | V0.1.0.0 5679H-V20 HX000027.110 | CPU: Information for the support | |
| Bx000227 | V6.6.29.255 | | |
| Ax000086 | V1.2.1.0 | | |
| Ax000056 | V0.2.2.0 | | |
| fx000007.wld | V1.1.8.0 | | |
| ARM Processor Load | | | |
| Last Value | 0% | | |
| Maximum load | 41% | | |

Data

Currently nothing is displayed here.

Parameter

Currently nothing is displayed here.

IP

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

Info - Overview

DP master

Slot100 (VIPA 31x-xxxx CPU)
System: (SPEED-Bus)
• Slot 201 (VIPA 342-1DA70)
Slot 206 (VIPA)
System: (VBUS/KBUS)
...

Info Data

Device (VIPA 342-1DA70) information

| Name | Value |
|---------------|----------------|
| Ordering Info | VIPA 342-1DA70 |
| Version | V3.3.0 |

[Expert View ...]

Info - Expert View

| Internal Information | Slot 201 | VIPA 342-1DA70 |
|------------------------------------|--|--|
| Module Type | 0xCB2C0010 | |
| Module Firmware Information | | |
| PRODUCT | VIPA 342-1DA70 | Name, firmware-version, package |
| | V3.3.6 | |
| | Px000181.pkg | |
| BB000218 | V5.3.0.0 | Information for support |
| AB000068 | V4.1.7.0 | |
| Runtime Information | | |
| Cycle Time | cur = 0us, min = 65535000us, max = 0us, avg = 0us, cnt = 0 | CPU cycle time: min = minimal cur = current max = maximal |

Info - Overview

CPU component: Digitale I/O

Slot100 (VIPA 31x-xxxx CPU)
System: (SPEED-Bus)
...
• Slot 202 (VIPA DI24/DO24/DIO8)
...
System: (VBUS/KBUS)
...

Info Data

Device (VIPA DI24/DO24/DIO8) information

| Name | Value |
|---------------|---------------------|
| Ordering Info | VIPA DI24/DO24/DIO8 |
| Version | V3.6.22 |

[Expert View ...]

Info - Expert View

| Internal Information | Slot 202 | Slot 202 |
|------------------------------------|---------------------|-------------------------|
| Module Type | 0x4FD30000 | Information for support |
| Module Firmware Information | | |
| PRODUCT | VIPA DI24/DO24/DIO8 | Name, firmware version |
| | V3.6.22.0 | |

Data - Input data (4Byte)

| Offset | Width | Value (dec) | Value (hex) |
|--------|-------|-------------|-------------|
| 124 | 1 | 0 | 00 |

Accessing the web server

| Offset | Width | Value (dec) | Value (hex) |
|--------|-------|-------------|-------------|
| 125 | 1 | 0 | 00 |
| 126 | 1 | 0 | 00 |
| 127 | 1 | 0 | 00 |

Data - Output data (4Byte)

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

Info - Overview

CPU component: Analog I/O

Slot100 (VIPA 31x-xxxx CPU)
System: (SPEED-Bus)

- Slot 203 (VIPA AI5/AO2)
System: (VBUS/KBUS)

...

Info Data

Device (VIPA AI5/AO2) information

| Name | Value |
|---------------|--------------|
| Ordering Info | VIPA AI5/AO2 |
| Version | V1.1.2 |

[Expert View ...]

Info - Expert View

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

Data - AI5 (10byte)

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

Data - AO2 (4byte)

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

Info - Overview

CPU component: counter

Slot100 (VIPA 31x-xxxx CPU)
System: (SPEED-Bus)

• Slot 204 (VIPA 4 COUNTERS)
System: (VBUS/KBUS)

...

Info Data

Device (VIPA 4 COUNTERS) information

| Name | Value |
|---------------|-----------------|
| Ordering Info | VIPA 4 COUNTERS |
| Version | V3.6.22 |

[Expert View ...]

Info - Expert View

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

Data - Input data (16byte)

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

Accessing the web server

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

Slot100 (... 31x-xxxx CPU)
System: (SPEED-Bus)
...
System: (VBUS/KBUS)
R0/Slot4 (Digital In/Out 16)
• R0/Slot5 (Analog Input 8)
R0/Slot6 (Analog Output 4)

Info
Data

Digital In/Out 16 - information

| Name | Value |
|---------------|-------------------|
| Ordering Info | Digital In/Out 16 |

[Expert View ...]

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 |

5.13 Operating modes

5.13.1 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.

5.13.2 Function security

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

| Event | concerns | Effect |
|-------------------------|-------------------------|--|
| RUN → STOP | general | BASP (Befehls-Ausgabe-Sperre, i.e. command output disable) is set. |
| | central digital outputs | The outputs are disabled. |
| | central analog outputs | The outputs are disabled. <ul style="list-style-type: none"> ■ Voltage outputs issue 0V ■ Current outputs 0...20mA issue 0mA ■ Current outputs 4...20mA issue 4mA If configured also substitute values may be issued. |
| | decentral outputs | Same behavior as the central digital/analog outputs. |
| STOP → RUN res. PowerON | decentral inputs | The inputs are cyclically be read by the decentralized station and the recent values are put at disposal. |
| | 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. |
| RUN | decentral inputs | The inputs are once be read by the decentralized station and the recent values are put at disposal. |
| | 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

5.14 Overall reset

Overview

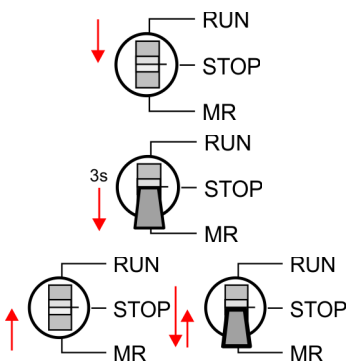
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

- 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 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.

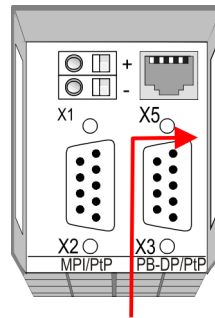
Reset to factory setting

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!
 ➔ *'Reset to factory settings'...page 80*

5.15 Firmware update

Overview

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



Firmware package and version

Latest firmware at
www.yaskawa.eu.com

The latest firmware versions are to be found at www.yaskawa.eu.com in the 'Download Center'. For example the following files are necessary for the firmware update of the CPU 314-6CG23 and its components with hardware release 01:

- 314-6CG23, Hardware release 01: Px000278.pkg
- PROFIBUS DP master: Px000181.pkg
- Analog part: Px000073.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 our 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. → 'Accessing the web server'...page 67

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 Yaskawa CPU 314-6CG23 to the CPU 314C-2DP from Siemens:

| Description: | CPU 314C-2DP | System identification: | SIMATIC 300 | | | | | | | | | |
|-----------------------|--|------------------------|-------------|-----------------------|-----------|---------|----------------|----------|---|--------------------|----------|--------|
| Name: | CPU 314C-2DP | | | | | | | | | | | |
| Version: | <table border="1"> <thead> <tr> <th>Order No./Description</th> <th>Component</th> <th>Version</th> </tr> </thead> <tbody> <tr> <td>6ES7 314-6CG03</td> <td>Hardware</td> <td>1</td> </tr> <tr> <td>... 314-6CG23-0100</td> <td>Firmware</td> <td>V3.6.0</td> </tr> </tbody> </table> | | | Order No./Description | Component | Version | 6ES7 314-6CG03 | Hardware | 1 | ... 314-6CG23-0100 | Firmware | V3.6.0 |
| Order No./Description | Component | Version | | | | | | | | | | |
| 6ES7 314-6CG03 | Hardware | 1 | | | | | | | | | | |
| ... 314-6CG23-0100 | Firmware | V3.6.0 | | | | | | | | | | |

1
2 3
4

- 1 Order number (314-6CG23)
- 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 Yaskawa 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

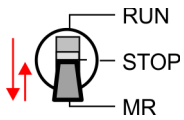
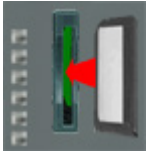
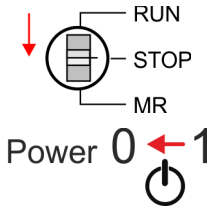
- Go to www.yaskawa.eu.com
- Navigate in the 'Download Center' to your CPU and download the zip file to your PC.
- Extract the zip file and copy the extracted pkg files to your memory card.



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 ➔ 'Reset to factory settings'...page 80.

Transfer firmware from memory card into CPU



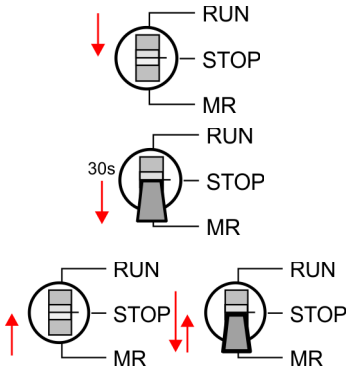
1. Switch the operating mode switch of your CPU in position STOP.
2. Turn off the power supply.
3. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card.
4. Turn on the power supply.
 - ➔ After a short boot-up time, the alternate 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.
 - ➔ ['Reset to factory settings'...page 80](#)

Reset to factory settings

5.16 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`. → 'CMD - auto commands'...page 84



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.

5.17 Deployment storage media - MMC, MCC

Overview

At this slot the following storage media can be plugged:

- SD respectively MMC (**M**ultimedia **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. → [‘Overall reset’...page 76](#)

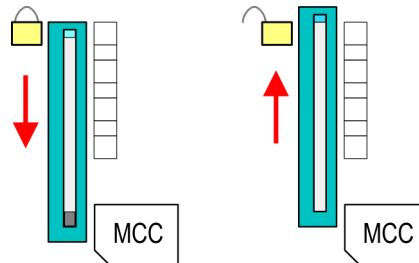


To avoid malfunctions, you should use memory cards of Yaskawa. These correspond to the industrial standard. A list of the currently available memory cards can be found at www.yaskawa.eu.com

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

MMC

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



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



CAUTION

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



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

MCC

- The MCC is a MMC with the possibility to unlock additional work memory.
- By plugging the MCC into the MCC slot and then an overall reset the according memory expansion is released. There may only one memory expansion be activated at one time.
- On the MCC there is the file memory.key. This file may not be altered or deleted.
- You may use the MCC also as "normal" MMC for storing your project.
- If the memory expansion on the MCC exceeds the maximum extendible memory range of the CPU, the maximum possible memory of the CPU is automatically used.
- You may determine the recent memory extension and the remaining time after pulling the MCC via the integrated web page. → [‘Accessing the web server’...page 67](#)
- 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. → [‘Firmware update’...page 77](#)

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.

5.18 Extended know-how protection

Overview



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

Besides the "standard" Know-how protection the CPUs 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 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*. ➔ '*Overall reset*'...[page 76](#)
 - ➔ 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.

5.19 CMD - auto commands

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

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

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

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

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

| Command | Description | Diagnostics entry |
|-----------------|--|-------------------|
| CMD_START | In the first line <i>CMD_START</i> is to be located. | 0xE801 |
| | There is a diagnostic entry if <i>CMD_START</i> 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: <i>SAVE_PROJECT</i> 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 ↪ 'Reset to factory settings'...page 80 you get the default setting. | 0xE814 |
| CMD_END | In the last line <i>CMD_END</i> is to be located. | 0xE802 |

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

Example 1

| | |
|------------------------|--|
| CMD_START | Marks the start of the command sequence (0xE801) |
| LOAD_PROJECT proj.wld | Execute an overall reset and load "proj.wld" (0xE805) |
| WAIT1SECOND | Wait ca. 1s (0xE803) |
| 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 <i>CMD_END</i> 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 <i>CMD_END</i> is not evaluated. |



The parameters IP address, subnet mask and gateway may be received from the system administrator.

Enter the IP address if there is no gateway used.

5.20 Diagnostic entries

Accessing diagnostic data → [‘System specific event IDs’...page 209](#)

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostics buffer, the 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. → [‘CMD - auto commands’...page 84](#)
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

5.21 Control and monitoring of variables with test functions

Overview

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

'Debug → Monitor'

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



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

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

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

'PLC → Monitor/Modify Variables'

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

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

- Forcing variables
 - You can pre-set individual variables of a user program with fixed values so that they can not be changed or overwritten by the user program of the CPU.
 - By pre-setting of variables with fixed values, you can set certain situations for your user program and thus test the programmed functions.

**CAUTION**

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

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

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

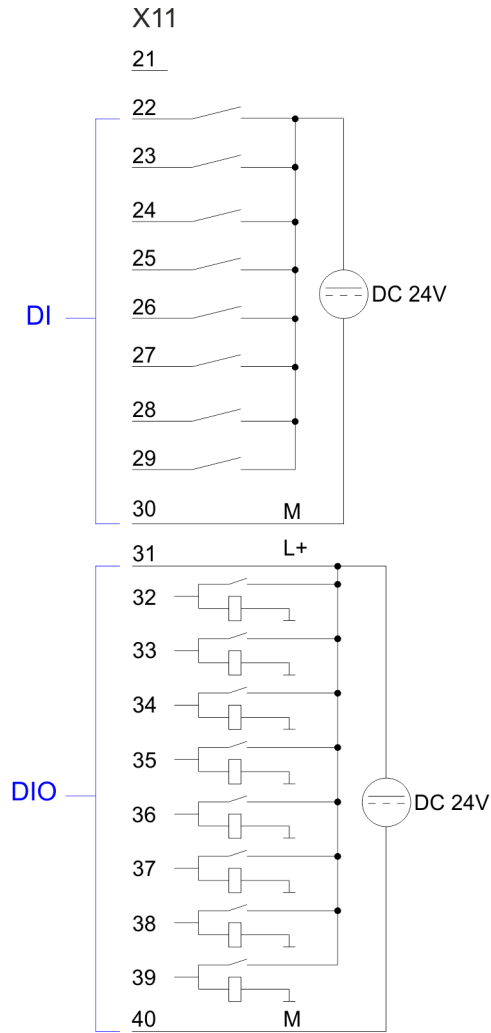
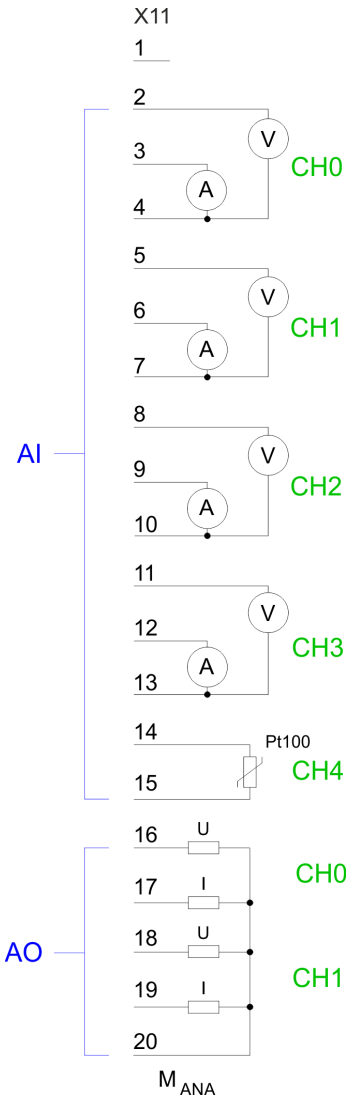
6 Deployment I/O periphery

6.1 Overview

| | |
|--------------------------------|---|
| Hardware | At the 314-6CG23 the connectors for digital in-/output and technological functions are integrated to a 3-tier casing. |
| Project engineering | The project engineering takes place in the Siemens SIMATIC manager as CPU 314C-2DP from Siemens (6ES7 314-6CG03-0AB0 V2.6). Here the CPU 314-6CG23 is parameterized by the "Properties" dialog of the CPU 314C-2DP. For parameterization of the digital I/O periphery and the technological functions the corresponding submodule of the CPU 314C-2DP may be used. |
| I/O periphery | The integrated I/Os of the 314-6CG23 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 | <p>Up to 4 channels may be parameterized as technological function. The parameterization of the appropriate channel is made in the hardware configurator by the <i>count</i> submodule of the CPU 314C-2DP. There are the following technological functions:</p> <ul style="list-style-type: none">■ Continuous count■ Single count■ Periodic count■ Frequency measurement■ Pulse width modulation (PWM) <p>The controlling of the corresponding counter mode happens by means of the SFB COUNT (SFB 47) of the user program.</p> |

6.2 In-/Output area CPU 314-6CG23

| | |
|-----------------|---|
| Overview | <p>The CPU 314-6CG23 has the following analog and digital in- and output ranges integrated in one casing:</p> <ul style="list-style-type: none">■ Analog Input: 4xU/Ix12Bit, 1xPt100■ Analog Output: 2xU/Ix12Bit■ Digital Input: 24xDC 24V■ Digital Output: 16xDC 24V, 0.5A■ Digital In-/Output: 8xDC 24V, 0.5A■ Technological functions: 4 channels <p>The analog channels of the module are isolated to the back plane via DC/DC transducer and optocouplers.</p> <p>Via the parameterization you may assign alarm properties to the digital inputs of X12. Additionally the digital inputs of X12 are parameterizable as counter.</p> |
|-----------------|---|



CAUTION

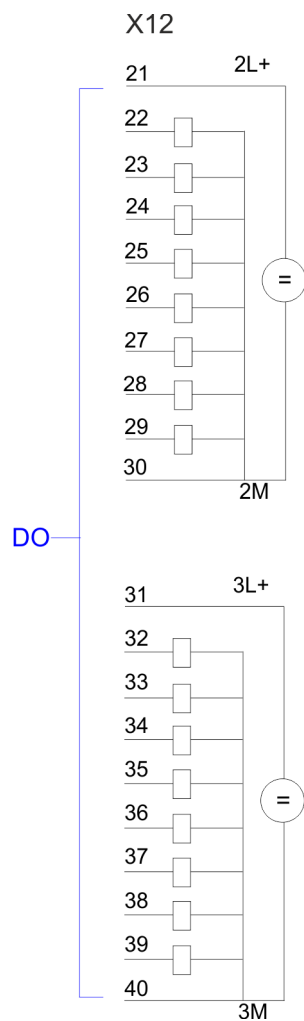
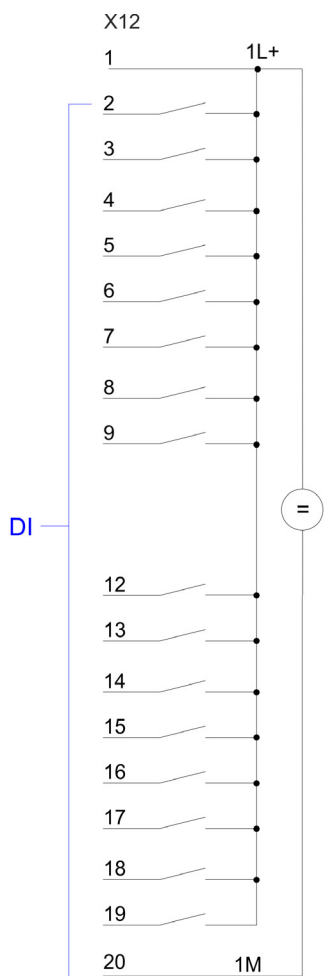
Temporarily not used analog inputs with activated channel must be connected to the concerning ground. To avoid measuring errors, you should connect only one measuring type per channel.



CAUTION

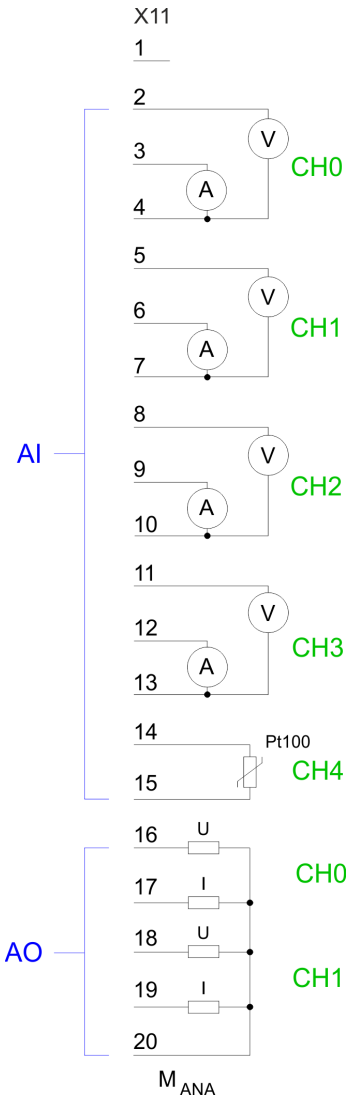
Please regard that the voltage at an output channel is always \leq the supply voltage connected to L+. Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal. A thus connected output remains active even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

In-/Output area CPU 314-6CG23



CAUTION

Please regard that the voltage at an output channel is always \leq the supply voltage connected to L+.



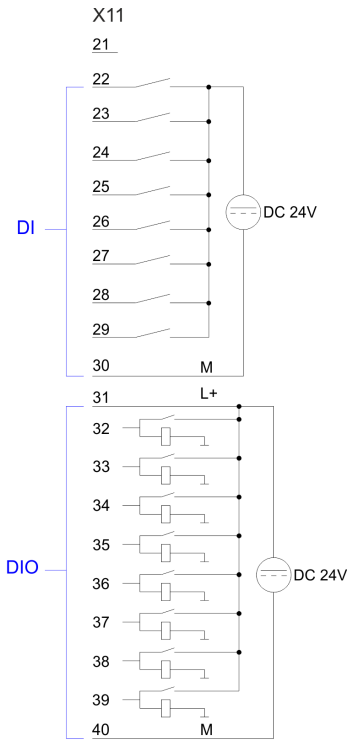
Pin assignment X11: AIO

| Pin | Assignment |
|-----|-------------------------------|
| 1 | not connected |
| 2 | Voltage measurement channel 0 |
| 3 | Current measurement channel 0 |
| 4 | Ground channel 0 |
| 5 | Voltage measurement channel 1 |
| 6 | Current measurement channel 1 |
| 7 | Ground channel 1 |
| 8 | Voltage measurement channel 2 |
| 9 | Current measurement channel 2 |
| 10 | Ground channel 2 |
| 11 | Voltage measurement channel 3 |
| 12 | Current measurement channel 3 |
| 13 | Ground channel 3 |
| 14 | Pt 100 channel 4 |
| 15 | Pt 100 channel 4 |
| 16 | Voltage output channel 0 |
| 17 | Current output channel 0 |
| 18 | Voltage output channel 1 |
| 19 | Current output channel 1 |
| 20 | Ground AO channel 0 and 1 |

Status indicator X11: AIO

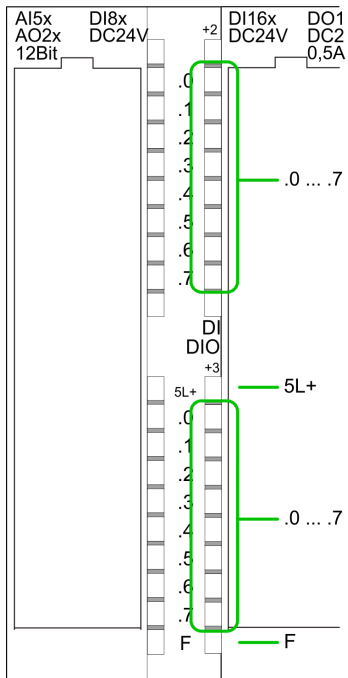
From the analog part no LEDs are controlled.

In-/Output area CPU 314-6CG23



Pin assignment X11: DI

| Pin | Assignment |
|-----|-------------------------|
| 21 | not connected |
| 22 | I+2.0 |
| 23 | I+2.1 |
| 24 | I+2.2 |
| 25 | I+2.3 |
| 26 | I+2.4 |
| 27 | I+2.5 |
| 28 | I+2.6 |
| 29 | I+2.7 |
| 30 | Ground 1M DI |
| 31 | L+ Power supply +DC 24V |
| 32 | I/O+3.0 |
| 33 | I/O+3.1 |
| 34 | I/O+3.2 |
| 35 | I/O+3.3 |
| 36 | I/O+3.4 |
| 37 | I/O+3.5 |
| 38 | I/O+3.6 |
| 39 | I/O+3.7 |
| 40 | Ground DIO |

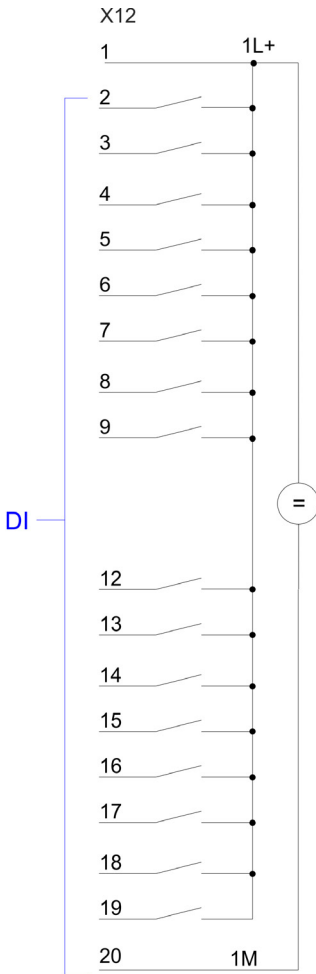


Status indication X11: DI

- .07
 - LEDs (green)
 - I+2.0 ... I+2.7
 - Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated

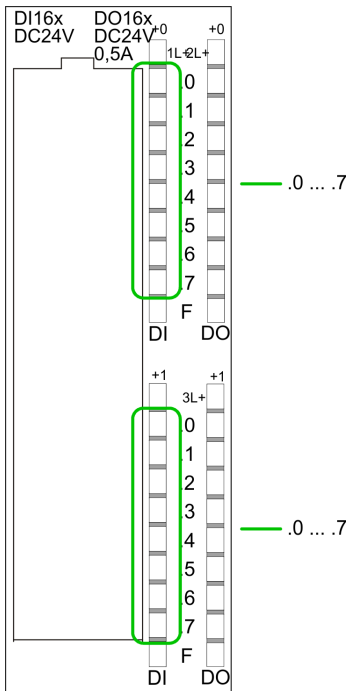
DIO

- 5L+
 - LEDs (green)
 - Supply voltage available for DIO
- .0... .7
 - LEDs (green)
 - I/Q+3.0 ... I/Q+3.7
 - On at active input resp. output
- F
 - LED (red)
 - Overload or short circuit error



Pin assignment X12: DI

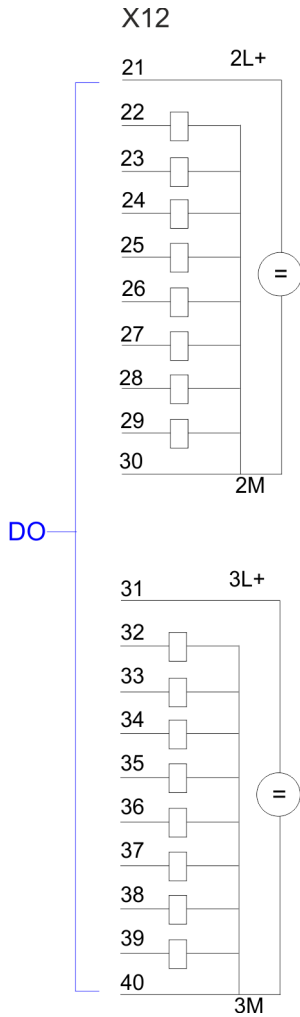
| Pin | Assignment |
|-----|---|
| 1 | 1L+ Power supply +DC 24V |
| 2 | I+0.0 / Channel 0 (A) / Pulse |
| 3 | I+0.1 / Channel 0 (B) / Direction / Frequency measurement |
| 4 | I+0.2 / Channel 0 HW gate |
| 5 | I+0.3 / Channel 1 (A) / Pulse |
| 6 | I+0.4 / Channel 1 (B) / Direction / Frequency measurement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



Status indication X12: DI

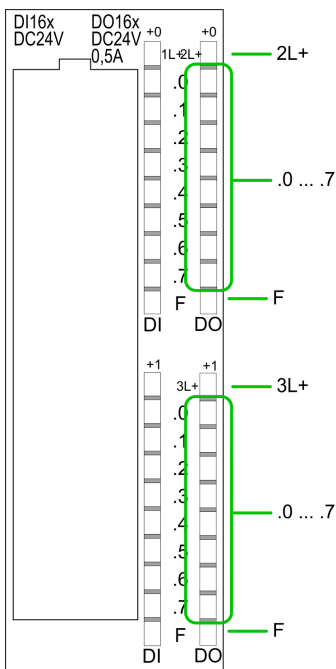
- .07
 - LEDs (green)
 - I+0.0 ... I+0.7
 - I+1.0 ... I+1.7
- Starting with ca. 15V the signal "1" at the input is recognized and the according LED is activated

In-/Output area CPU 314-6CG23



Pin assignment X12: 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 / Channel 2 Output |
| 25 | Q+0.3 / Channel 3 Output |
| 26 | Q+0.4 |
| 27 | Q+0.5 |
| 28 | Q+0.6 |
| 29 | Q+0.7 |
| 30 | Ground 2M DO |
| 31 | 3L+ Power supply +DC 24V |
| 32 | Q+1.0 |
| 33 | Q+1.1 |
| 34 | Q+1.2 |
| 35 | Q+1.3 |
| 36 | Q+1.4 |
| 37 | Q+1.5 |
| 38 | Q+1.6 |
| 39 | Q+1.7 |
| 40 | Ground 3M DO |



Status indication X12: DO

- 2L+, 3L+
 - LED (green)
 - Supply voltage available for DO
- .07
 - LEDs (green)
 - Q+0.0 ... Q+0.7
 - Q+1.0 ... Q+1.7
 - 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 |
|------------|-----------------|--------|--|
| DI24/DO16 | 124 | Byte | Digital Input I+0.0 ... I+0.7 (X12) |
| | 125 | Byte | Digital Input I+1.0 ... I+1.7 (X12) |
| | 126 | Byte | Digital Input I+2.0 ... I+2.7 (X11) |
| | 127 | Byte | Digital Input I+3.0 ... I+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Input Channel 0 |
| | 754 | Word | Analog Input Channel 1 |
| | 756 | Word | Analog Input Channel 2 |
| | 758 | Word | Analog Input Channel 3 |
| | 760 | Word | Analog Input Channel 4 |
| Counter | 768 | DInt | Channel 0: Count value / Frequency value |
| | 772 | DInt | Channel 1: Count value / Frequency value |
| | 776 | DInt | Channel 2: Count value / Frequency value |
| | 780 | DInt | Channel 3: Count value / Frequency value |

Output range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--------------------------------------|
| DI24/DO16 | 124 | Byte | Digital Output Q+0.0 ... Q+0.7 (X12) |
| | 125 | Byte | Digital Output Q+1.0 ... Q+1.7 (X12) |
| | 126 | Byte | reserved |
| | 127 | Byte | Digital Output Q+3.0 ... Q+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Output Channel 0 |
| | 754 | Word | Analog Output Channel 1 |
| Counter | 768 | DWord | reserved |
| | 772 | DWord | reserved |
| | 776 | DWord | reserved |
| | 780 | DWord | reserved |



Please regard during hardware configuration due to the additional DIO channels the CPU 314-6CG23 occupies 1 more byte in the IO area (default address 127), each. It is not automatically detected if this byte was already assigned to a module in the hardware configuration.

6.4 Analog part

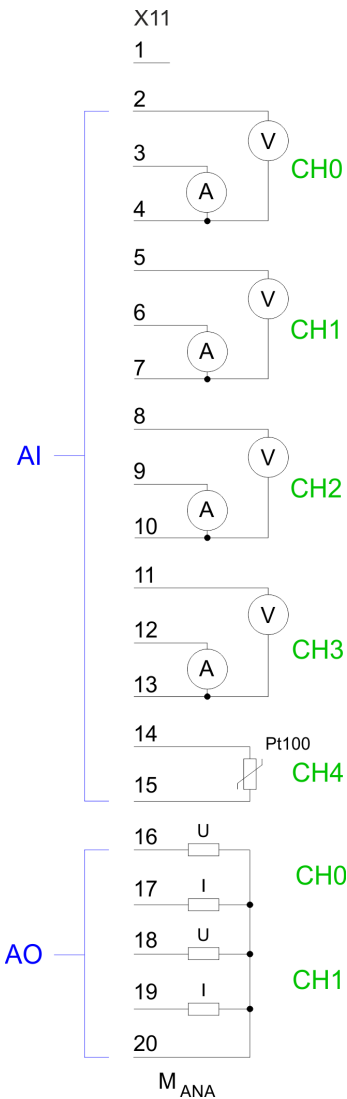
314-6CG23

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



CAUTION

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



Pin assignment X11: AIO

| Pin | Assignment |
|-----|-------------------------------|
| 1 | not connected |
| 2 | Voltage measurement channel 0 |
| 3 | Current measurement channel 0 |
| 4 | Ground channel 0 |
| 5 | Voltage measurement channel 1 |
| 6 | Current measurement channel 1 |
| 7 | Ground channel 1 |
| 8 | Voltage measurement channel 2 |
| 9 | Current measurement channel 2 |
| 10 | Ground channel 2 |
| 11 | Voltage measurement channel 3 |
| 12 | Current measurement channel 3 |
| 13 | Ground channel 3 |
| 14 | Pt 100 channel 4 |
| 15 | Pt 100 channel 4 |
| 16 | Voltage output channel 0 |
| 17 | Current output channel 0 |
| 18 | Voltage output channel 1 |
| 19 | Current output channel 1 |
| 20 | Ground AO channel 0 and 1 |

Status indicator X11: AIO

From the analog part no LEDs are controlled.

6.4.1 Access to the I/O part

The CPU 314-6CG23 creates in its peripheral area an area for input respectively output data for the modules. 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 |
|------------|-----------------|--------|--|
| DI24/DO16 | 124 | Byte | Digital Input I+0.0 ... I+0.7 (X12) |
| | 125 | Byte | Digital Input I+1.0 ... I+1.7 (X12) |
| | 126 | Byte | Digital Input I+2.0 ... I+2.7 (X11) |
| | 127 | Byte | Digital Input I+3.0 ... I+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Input Channel 0 |
| | 754 | Word | Analog Input Channel 1 |
| | 756 | Word | Analog Input Channel 2 |
| | 758 | Word | Analog Input Channel 3 |
| | 760 | Word | Analog Input Channel 4 |
| Counter | 768 | DInt | Channel 0: Count value / Frequency value |
| | 772 | DInt | Channel 1: Count value / Frequency value |
| | 776 | DInt | Channel 2: Count value / Frequency value |
| | 780 | DInt | Channel 3: Count value / Frequency value |

Output range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--------------------------------------|
| DI24/DO16 | 124 | Byte | Digital Output Q+0.0 ... Q+0.7 (X12) |
| | 125 | Byte | Digital Output Q+1.0 ... Q+1.7 (X12) |
| | 126 | Byte | reserved |
| | 127 | Byte | Digital Output Q+3.0 ... Q+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Output Channel 0 |
| | 754 | Word | Analog Output Channel 1 |
| Counter | 768 | DWord | reserved |
| | 772 | DWord | reserved |
| | 776 | DWord | reserved |
| | 780 | DWord | reserved |



Please regard during hardware configuration due to the additional DIO channels the CPU 314-6CG23 occupies 1 more byte in the IO area (default address 127), each. It is not automatically detected if this byte was already assigned to a module in the hardware configuration.

6.4.2 Analog value representation

Representation of analog values Analog values are exclusively processed in a binary format. For this the analog module transforms every process signal into a digital value and transfers this as word.

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

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

Sign bit (SG)

Here it is essential:

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

Behavior at error

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

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

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

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

Voltage measurement

±10V

| Meas. range | Voltage (U) | Decimal (D) | Hex | Range | Formulas |
|-------------|-------------|-------------|-------|------------------|--------------------------------|
| ±10V | > 11.76 | 32767 | 7FFFh | overflow | $D = 27648 \cdot \frac{U}{10}$ |
| | 11.76V | 32511 | 7EFFh | overdrive range | |
| | 10V | 27648 | 6C00h | nominal range | |
| | 5V | 13824 | 3600h | | $U = D \cdot \frac{10}{27648}$ |
| | 0V | 0 | 0000h | | |
| | -5V | -13824 | CA00h | | |
| | -10V | -27648 | 9400h | | |
| | -11.76V | -32512 | 8100h | underdrive range | |
| | < -11.76V | -32767 | 7FFFh | underflow | |

Voltage measurement

0 ... 10V

| Meas. range | Voltage (U) | Decimal (D) | Hex | Range | Formulas |
|-------------|-------------|-------------|-------|------------------|---|
| 0 ... 10V | > 11.76 | 32767 | 7FFFh | overflow | $D = 27648 \cdot \frac{U}{10}$ $U = D \cdot \frac{10}{27648}$ |
| | 11.76V | 32511 | 7EFFh | overdrive range | |
| | 10V | 27648 | 6C00h | nominal range | |
| | 5V | 13824 | 3600h | | |
| | 0V | 0 | 0000h | | |
| | -1.76V | -4864 | ED00h | underdrive range | |
| | < -1.76V | -32768 | 8000h | underflow | |

Current measurement

+/-20mA

| Meas. range | Current (I) | Decimal (D) | Hex | Range | Formulas |
|-------------|-------------|-------------|-----------|------------------|---|
| +/-20mA | > 23.52mA | 32767 | 7FFFh | overflow | $D = 27648 \cdot \frac{I}{20}$ $I = D \cdot \frac{20}{27648}$ |
| | 23.52mA | 32511 | 7EFFh | overdrive range | |
| | 20mA | 27648 | 6C00h | nominal range | |
| | 10mA | 13824 | 3600h | | |
| | 0mA | 0 | 0000h | | |
| | -20mA | -27648 | 9400h | | |
| | -23.52mA | -32512 | 8100h | underdrive range | |
| < -23.52mA | -32768 | 8000h | underflow | | |

Current measurement

0 ... 20mA

| Meas. range | Current (I) | Decimal (D) | Hex | Range | Formulas |
|-------------|-------------|-------------|-------|------------------|---|
| 0 ... 20mA | > 23.52mA | 32767 | 7FFFh | overflow | $D = 27648 \cdot \frac{I}{20}$ $I = D \cdot \frac{20}{27648}$ |
| | 23.52mA | 32511 | 7EFFh | overdrive range | |
| | 20mA | 27648 | 6C00h | nominal range | |
| | 10mA | 13824 | 3600h | | |
| | 0mA | 0 | 0000h | | |
| | -3.52mA | -4864 | ED00h | underdrive range | |
| | < -23.52mA | -32768 | 8000h | underflow | |

Current measurement

4 ... 20mA

| Output range | Current (I) | Decimal (D) | Hex | Range | Formulas |
|--------------|-------------|-------------|-------|------------------|---|
| 4 ... 20mA | > 22.81mA | 32767 | 7FFFh | overflow | $I = D \times \frac{16}{27648} + 4$ $D = 27648 \times \frac{I-4}{16}$ |
| | 22.81mA | 32511 | 7EFFh | overdrive range | |
| | 20mA | 27648 | 6C00h | nominal range | |
| | 12mA | 13824 | 3600h | | |
| | 4mA | 0 | 0000h | | |
| | 1.185mA | -4864 | ED00h | underdrive range | |
| | < 1.185mA | -32768 | 8000h | underflow | |

Resistance

0 ... 600Ω

| 0 ... 600Ω | Resistance (R) | Decimal (D) | Hex | Range | Formulas |
|---|----------------|-------------|-------|------------------|---|
| 0 ... 600Ω | > 705.53Ω | 32767 | 7FFFh | overflow | $D = 27648 \times \frac{R}{600}$ $R = D \times \frac{600}{27648}$ |
| | 705.53Ω | 32511 | 7EFFh | overdrive range | |
| | 600Ω | 27648 | 6C00h | nominal range | |
| | 0Ω | 0 | 0000h | | |
| (negative values are physically not possible) | | | | underdrive range | |

Resistance thermometer
Pt100 Standard

With Pt 100 the temperature is directly shown with the adjusted unit. Here applies: 1 Digit = 0.1 temperature unit

| Pt100 | Dec (D) | Hex | Pt100 | Dec (D) | Hex | Pt100 | Dec(D) | Hex | Range |
|----------------------|---------|-------|----------------------|---------|-------|--------------------|--------|-------|------------------|
| in °C (1digit=0.1°C) | | | in °F (1digit=0.1°F) | | | in K (1digit=0.1K) | | | |
| >1000.0 | 32767 | 7FFFh | >1832.0 | 32767 | 7FFFh | >1273.2 | 32767 | 7FFFh | overflow |
| 1000.0 | 10000 | 2710h | 1832.0 | 18320 | 4790h | 1273.2 | 12732 | 31BCh | overdrive range |
| 850.0 | 8500 | 2134h | 1562.0 | 15620 | 3D04h | 1123.2 | 11232 | 2BE0h | nominal range |
| -200.0 | -2000 | F830h | -328.0 | -3280 | F330h | 73.2 | 732 | 2DCh | |
| -243.0 | -2430 | F682h | -405.4 | -4054 | F02Ah | 30.2 | 302 | 12Eh | underdrive range |
| <-243.0 | -32768 | 8000h | <-405.4 | -32768 | 8000h | <30.2 | 32768 | 8000h | underflow |

6.4.3 Analog part - Wiring

Cables for analog signals

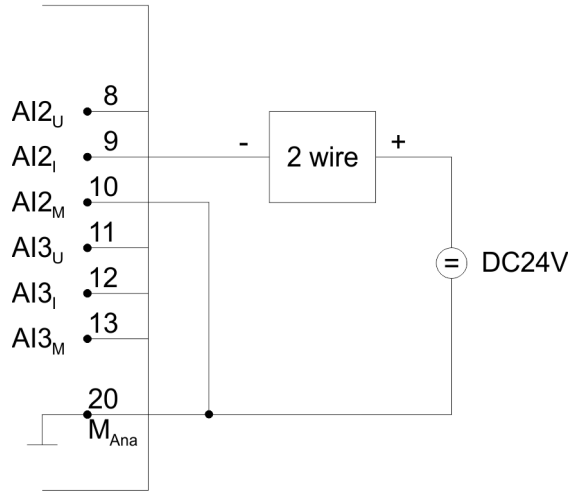
For analog signals you should use isolated cables to reduce interference. The cable screening should be grounded at both ends. If there are differences in the potential between the cable ends, there may occur a potential compensating current that could disturb the analog signals. In this case you should ground the cable screening only at one end.

Wiring of the analog current / voltage inputs

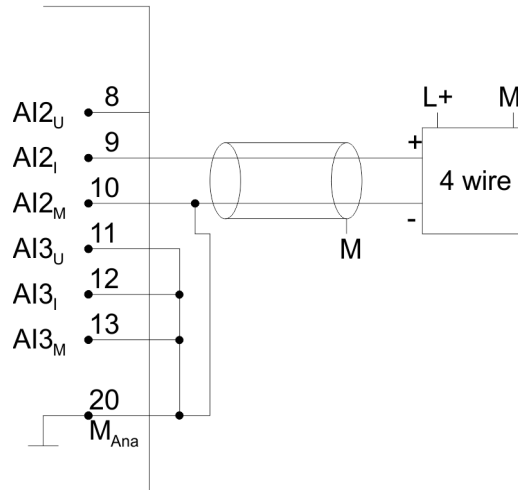
2-/4-wire measuring transducers may be connected to the analog part. Please regard that the transducers are to be supplied externally. Using 2-wire transducers an external power supply should be looped in. Please install short circuits at non-used inputs by connecting the positive contact with the channel ground. Bridging channel ground and M_{ANA} is recommended.

The connection of 2- and 4-wire measurement transducers is shown in the following figures at the example of a current measurement:

2 wire

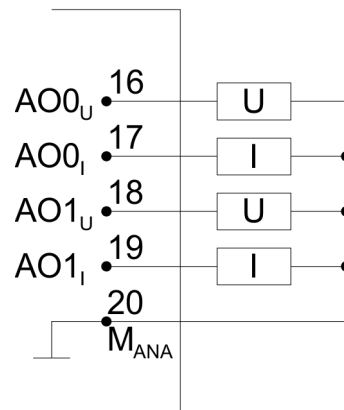


4 wire



Wiring of the analog outputs

Loads and actors may be supplied with voltage or current by the analog part. Please take always care of the correct polarity when connecting actuators! Please leave the output pins of not used channels disconnected and configure the *output type* of the channel to "deactivated".



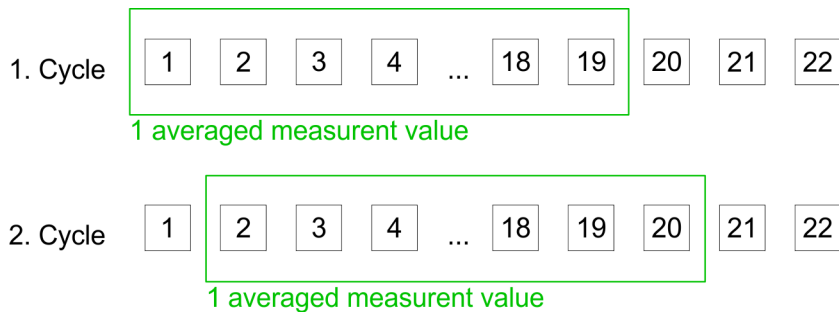
6.4.4 Analog part - Measurement principle

Overview The measurement principle of actual value encoding is used. With a sampling rate of approx. 1.05kHz each millisecond a new value is available in the periphery input word. This value may be read by the user program. The "previous" value is read again if your access time is shorter than 1ms.

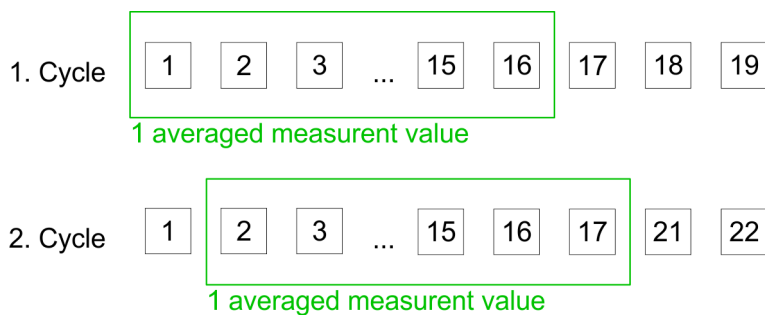
Hardware filter An integrated 400Hz low-pass filter attenuates analog input signals of channel 0 to 3. Input signals with a frequency of over 400Hz are not allowed.

Software filter Each analog current / voltage input has a configurable software filter for the input signals. The configured interference frequency (50/60Hz) may be filtered by this software filter. The selected interference suppression also determines the integration time. At an interference suppression of 50Hz the software filter forms the average based on the last 19 measurements and saves the result as a measurement value. The interference frequency 50Hz or 60Hz may be suppressed according to the parameter set. A setting of 400Hz will not suppress interference.

1 from 19 at 50Hz interference frequency



1 from 16 at 60Hz interference frequency



6.4.5 Analog part - Parameterization

Parameter data 13Byte of parameter data are available for the configuration. Parameters of the analog part may be set by means of the A/5/AO2 submodule of the Siemens CPU during hardware configuration.

By using the record set 1 of the SFC 55 "WR_PARM" you may alter the parameterization in the module during runtime. The time needed until the new parameterization is valid can last up to 2ms. During this time, the measuring value output is 7FFFh.

The following table shows the structure of the parameter data:

Record set 1

| Byte | Bit 7 ... Bit 0 | Default | |
|--------|---|---|-----|
| 0 | Temperature unit <ul style="list-style-type: none"> ■ Bit 2, 0: reserved ■ Bit 4, 3: Unit 00b = Celsius 01b = Fahrenheit 10b = Kelvin ■ Bit 7...5: reserved | 00h | |
| 1 | Integration time, Interference frequency suppression <ul style="list-style-type: none"> 00b = 2.5ms, 400Hz¹ 01b = 16.6ms, 60Hz 10b = 20.0ms, 50Hz ■ Bit 1, 0: Analog Input Channel 0 ■ Bit 3, 2: Analog Input Channel 1 ■ Bit 5, 4: Analog Input Channel 2 ■ Bit 7, 6: Analog Input Channel 3 | AAh | |
| 2 | reserved | | |
| 3 | Measurement Analog Input Channel 0 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Measuring range ■ Bit 7 ... 4: Measuring type | Measurement range: 0h: deactivated 2h: Current 0...20mA | 19h |
| 4 | Measurement Analog Input Channel 1 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Measuring range ■ Bit 7 ... 4: Measuring type | 3h: Current 4...20mA 4h: Current +/-20mA | 19h |
| 5 | Measurement Analog Input Channel 2 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Measuring range ■ Bit 7 ... 4: Measuring type | 8h: Voltage 0...10V 9h: Voltage +/-10V Measuring type: | 19h |
| 6 | Measurement Analog Input Channel 3 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Measuring range ■ Bit 7 ... 4: Measuring type | 0h: deactivated 1h: U Voltage 2h: I Current 3h: I Current | 19h |
| 7 | Measurement Analog Input Channel 4 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Measuring range ■ Bit 7 ... 4: Measuring type | Measurement range only channel 4: 0h: deactivated 2h: 600Ohm 6h: Pt100 Measurement type only channel 4: 0h: deactivated 6h: Resistor 15h: Resistance thermometer | 62h |
| 8...10 | reserved | | |
| 11 | Analog Output Channel 0 <ul style="list-style-type: none"> ■ Bit 3 ... 0: Output range ■ Bit 7 ... 4: Output type | Output range: 0h: deactivated 2h: Current 0...20mA 3h: Current 4...20mA 4h: Current +/-20mA | 19h |

Analog part > Analog part - Parameterization

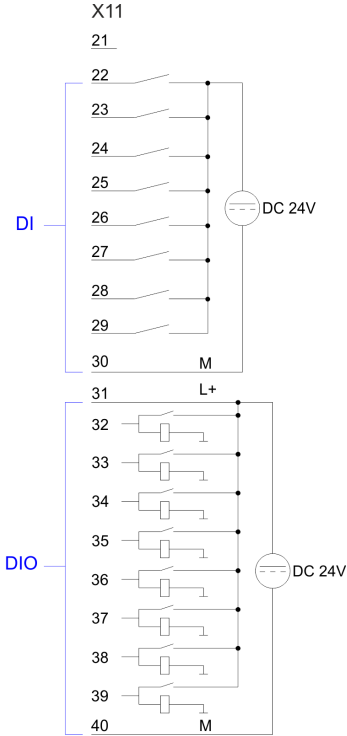
| Byte | Bit 7 ... Bit 0 | Default |
|------|--|--|
| 12 | Analog Output Channel 1 <ul style="list-style-type: none">■ Bit 3 ... 0: Output range■ Bit 7 ... 4: Output type | 8h: Voltage 0...10V 9h: Voltage +/-10V Output type: 0h: deactivated 1h: U Voltage 3h: I Current |

1) Here the Yaskawa CPU uses internally the value 1.25ms respectively 800Hz.

6.5 Digital part

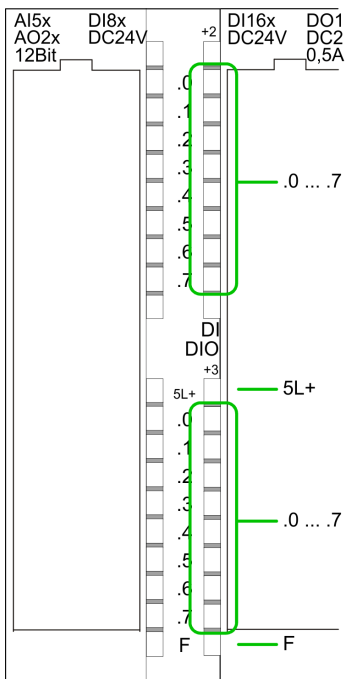
314-6CG23

The digital part consists of 24 input, 16 output, 8 in-/output channels and 4 channels for technological functions. Each of these digital input- respectively output channels shows its state via a LED. By means of the parameterization you may assign interrupt properties to the inputs I+0.0 ... I+1.7



Pin assignment X11: DI

| Pin | Assignment |
|-----|-------------------------|
| 21 | not connected |
| 22 | I+2.0 |
| 23 | I+2.1 |
| 24 | I+2.2 |
| 25 | I+2.3 |
| 26 | I+2.4 |
| 27 | I+2.5 |
| 28 | I+2.6 |
| 29 | I+2.7 |
| 30 | Ground 1M DI |
| 31 | L+ Power supply +DC 24V |
| 32 | I/O+3.0 |
| 33 | I/O+3.1 |
| 34 | I/O+3.2 |
| 35 | I/O+3.3 |
| 36 | I/O+3.4 |
| 37 | I/O+3.5 |
| 38 | I/O+3.6 |
| 39 | I/O+3.7 |
| 40 | Ground DIO |



Status indication X11: DI

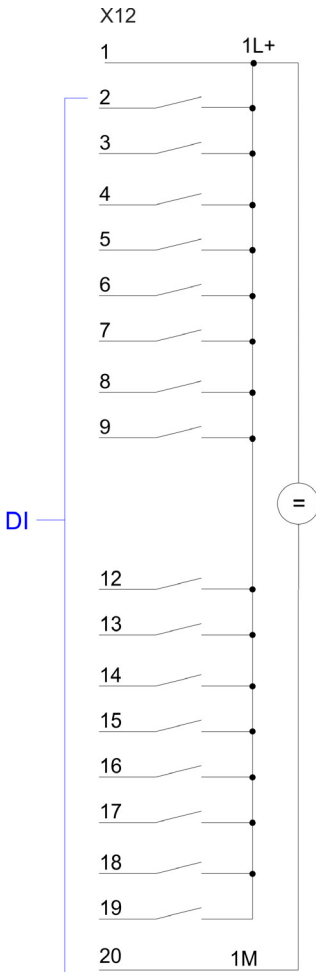
- .07
 - LEDs (green)
 - I+2.0 ... I+2.7

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

DIO

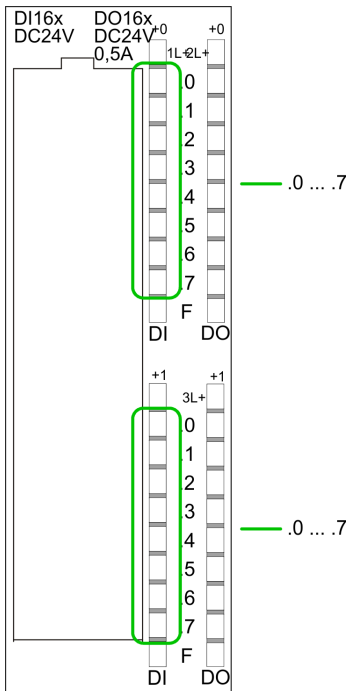
- 5L+
 - LEDs (green)
 - Supply voltage available for DIO
- .0... .7
 - LEDs (green)
 - I/Q+3.0 ... I/Q+3.7
 - On at active input resp. output
- F
 - LED (red)
 - Overload or short circuit error

Digital part



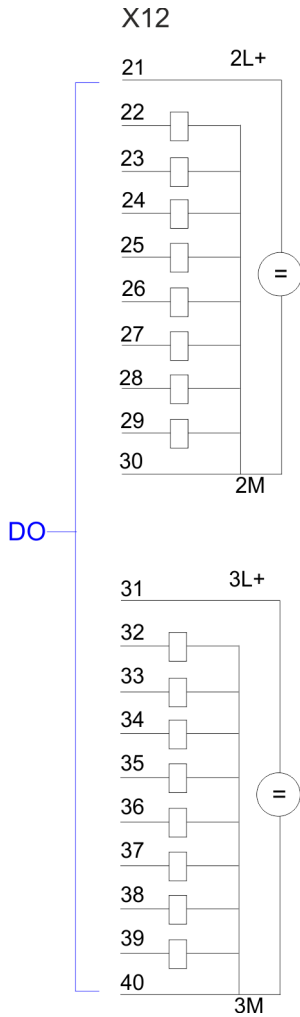
Pin assignment X12: DI

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



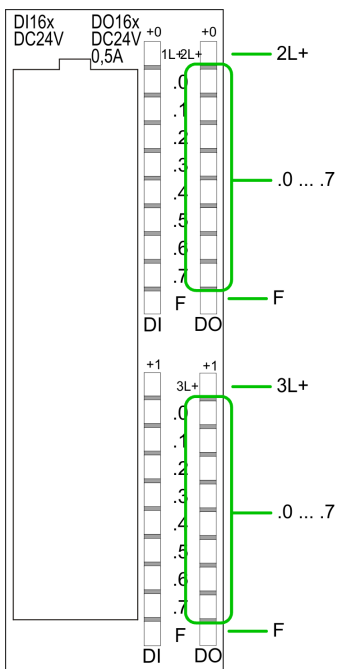
Status indication X12: DI

- .07
 - 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 X12: 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 / Channel 2 Output |
| 25 | Q+0.3 / Channel 3 Output |
| 26 | Q+0.4 |
| 27 | Q+0.5 |
| 28 | Q+0.6 |
| 29 | Q+0.7 |
| 30 | Ground 2M DO |
| 31 | 3L+ Power supply +DC 24V |
| 32 | Q+1.0 |
| 33 | Q+1.1 |
| 34 | Q+1.2 |
| 35 | Q+1.3 |
| 36 | Q+1.4 |
| 37 | Q+1.5 |
| 38 | Q+1.6 |
| 39 | Q+1.7 |
| 40 | Ground 3M DO |



Status indication X12: DO

- 2L+, 3L+
 - LED (green)
 - Supply voltage available for DO
- .07
 - LEDs (green)
 - Q+0.0 ... Q+0.7
 - Q+1.0 ... Q+1.7
 - The according LED is on at active output
- F
 - LED (red)
 - Overload or short circuit error

6.5.1 Access to the I/O part

The CPU 314-6CG23 creates in its peripheral area an area for input respectively output data for the modules. Without a hardware configuration the in the following specified default addresses are used.

6.5.1.1 Address assignment

Input range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--|
| DI24/DO16 | 124 | Byte | Digital Input I+0.0 ... I+0.7 (X12) |
| | 125 | Byte | Digital Input I+1.0 ... I+1.7 (X12) |
| | 126 | Byte | Digital Input I+2.0 ... I+2.7 (X11) |
| | 127 | Byte | Digital Input I+3.0 ... I+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Input Channel 0 |
| | 754 | Word | Analog Input Channel 1 |
| | 756 | Word | Analog Input Channel 2 |
| | 758 | Word | Analog Input Channel 3 |
| | 760 | Word | Analog Input Channel 4 |
| Counter | 768 | DInt | Channel 0: Count value / Frequency value |
| | 772 | DInt | Channel 1: Count value / Frequency value |
| | 776 | DInt | Channel 2: Count value / Frequency value |
| | 780 | DInt | Channel 3: Count value / Frequency value |

Output range

| Sub module | Default address | Access | Assignment |
|------------|-----------------|--------|--------------------------------------|
| DI24/DO16 | 124 | Byte | Digital Output Q+0.0 ... Q+0.7 (X12) |
| | 125 | Byte | Digital Output Q+1.0 ... Q+1.7 (X12) |
| | 126 | Byte | reserved |
| | 127 | Byte | Digital Output Q+3.0 ... Q+3.7 (X11) |
| AI5/AO2 | 752 | Word | Analog Output Channel 0 |
| | 754 | Word | Analog Output Channel 1 |
| Counter | 768 | DWord | reserved |
| | 772 | DWord | reserved |
| | 776 | DWord | reserved |
| | 780 | DWord | reserved |



Please regard during hardware configuration due to the additional DIO channels the CPU 314-6CG23 occupies 1 more byte in the IO area (default address 127), each. It is not automatically detected if this byte was already assigned to a module in the hardware configuration.

6.5.2 Digital part - Parameterization

| | |
|-----------------------|--|
| Parameter data | Parameters of the digital part may be set by means of the <i>DI24/DO16</i> submodule of the CPU 314C-2DP 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 <i>Comment</i> 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 | <p>Here there are the following adjustment possibilities:</p> <ul style="list-style-type: none">■ Hardware interrupt<ul style="list-style-type: none">– A hardware interrupt may be optionally triggered on the rising or falling edge of an input. There are no hardware interrupts available at input I+2.0 ... I+2.7 of the CPU 314-6CG23. 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<ul style="list-style-type: none">– 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. |

6.6 Counter

6.6.1 Counter - Fast introduction

Overview

The CPU 314-6CG23 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 314C-2DP from Siemens. Now the following technological functions at 4 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 314-6CG23 the maximum frequency amounts to 60kHz.

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

Pin assignment

→ ['Pin assignment X12: DI' ...page 106](#)

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 314C-2DP. 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 314C-2DP.
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 been 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 operation 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 \geq comparison value: Output is set as long as counter value \geq comparison value.
- Count value \leq comparison value: Output is set as long as counter value \leq 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.

6.6.1.1 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 CPU 314-6CG23 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 | <ul style="list-style-type: none"> ■ <i>None</i>: No restriction of the counting range ■ <i>Up</i>: 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. ■ <i>Down</i>: 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. | <ul style="list-style-type: none"> ■ None ■ Up ■ Down (not with continuous count) | <ul style="list-style-type: none"> ■ None |
| End value/ Start value | <i>End value</i> , with up-count as default. <i>Start value</i> , with down-count as default. | 2...2147483647 ($2^{31}-1$) | 2147483647 ($2^{31}-1$) |

| Parameters | Description | Range of values | Default |
|--|--|--|-----------------|
| Gate function | <ul style="list-style-type: none"> ■ <i>Cancel count</i>: The count starts when the gate opens and resumes at the load value when the gate opens again. ■ <i>Stop count</i>: The count is interrupted when the gate closes and resumed at the last actual value when the gate opens again. | <ul style="list-style-type: none"> ■ Abort the count operation ■ Interrupt the count operation | Cancel count |
| Comparison value | <p>The count value is compared with the comparison value. see also the parameter "Characteristics of the output":</p> <ul style="list-style-type: none"> ■ No main direction of count ■ Up-count as default ■ Down-count as default | <p>-2^{31} to $+2^{31}-1$</p> <p>-2^{31} to End value</p> <p>1 to $+2^{31}-1$</p> | 0 |
| Hysteresis | <p>A hysteresis is used to eliminate frequent output jitter if the count value lies within the range of the comparison value.</p> <p>0 and 1 means: Hysteresis switched off</p> | 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. | 60, 30, 10, 5, 2, 1kHz | 60kHz |
| max. frequency: Latch | You can set the maximum frequency of the latch signal in fixed steps. | 60, 30, 10, 5, 2, 1kHz | 10kHz |
| Signal evaluation | <p>The count and direction signals are connected to the input.</p> <p>A rotary transducer is connected to the input (single, dual or quadruple evaluation).</p> | <ul style="list-style-type: none"> ■ 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. | <ul style="list-style-type: none"> ■ activated ■ deactivated | deactivated |
| Count direction inverted | In the activated state the "direction" input signal is inverted. | <ul style="list-style-type: none"> ■ activated ■ deactivated | deactivated |
| Characteristics of the output | The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter. | <ul style="list-style-type: none"> ■ No comparison ■ $\text{Count} \geq \text{comparison value}$ ■ $\text{Count} \leq \text{comparison value}$ ■ 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. | <ul style="list-style-type: none"> ■ 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. | <ul style="list-style-type: none"> ■ activated ■ deactivated | deactivated |

Counter > SFB 47 - COUNT - Counter controlling

| Parameters | Description | Range of values | Default |
|--|---|--|-------------|
| 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". | <ul style="list-style-type: none"> ■ 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). | <ul style="list-style-type: none"> ■ activated ■ deactivated | deactivated |
| Hardware interrupt: Underflow | In the activated state a hardware interrupt is generated in the event of an underflow (under-shooting the lower count limit). | <ul style="list-style-type: none"> ■ activated ■ deactivated | deactivated |

6.6.2 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 <i>STS_C_DN</i> 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 <i>STS_C_UP</i> 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 ¹ Status bit <i>STS_CMP</i> indicates that the comparison condition of the comparator is or was reached. <i>STS_CMP</i> also indicates that the output was set. (<i>STS_DO</i> = TRUE). |
| RES04 | BOOL | 26.4 | FALSE | reserved |
| STS_OFLW | BOOL | 26.5 | FALSE | Overflow status ¹ |
| STS_UFLW | BOOL | 26.6 | FALSE | Underflow status ¹ |
| STS_ZP | BOOL | 26.7 | FALSE | Status of the zero mark ¹ 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: <i>STS_CMP</i> , <i>STS_OFLW</i> , <i>STS_ZP</i> . The SFB must be twice called to reset the status bit. |

1) 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) ¹ |
| JOB_ID | WORD | 6.0 | 0 | Job ID: 00h Job without function 01h Writes the <i>count value</i> 02h Writes the <i>load value</i> 04h Writes the <i>comparison value</i> 08h Writes the <i>hysteresis</i> 10h Writes the <i>pulse duration</i> 20h Writes the <i>end value</i> 82h Reads the <i>load value</i> 84h Reads the <i>comparison value</i> 88h Reads the <i>hysteresis</i> 90h Reads the <i>pulse duration</i> A0h Reads the <i>end value</i> |
| JOB_VAL | DINT | 8.0 | 0 | Value for write jobs |

1) 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 <i>Comparison value</i> too low 0122h <i>Comparison value</i> too high 0131h <i>Hysteresis</i> too low 0132h <i>Hysteresis</i> too high 0141h <i>Pulse duration</i> too low 0142h <i>Pulse duration</i> too high 0151h <i>Load value</i> too low 0152h <i>Load value</i> too high 0161h <i>Count value</i> too low 0162h <i>Count value</i> too high 01FFh Invalid <i>job ID</i> |

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.

Counter > SFB 47 - COUNT - Counter controlling

Permitted value range for JOB_VAL**Continuous count:**

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

Single/periodic count, no main count direction:

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

Single/periodic count, main count direction up:

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

Single/periodic count, main count direction down:

| Job | Valid range |
|--|----------------------------------|
| Writing <i>counter</i> directly | 2 ... +2147483647 ($2^{31}-1$) |
| Writing the <i>load value</i> | 2 ... +2147483647 ($2^{31}-1$) |
| Writing <i>comparison value</i> | 1 ... +2147483647 ($2^{31}-1$) |
| Writing <i>hysteresis</i> | 0 ... 255 |
| Writing <i>pulse duration</i> ¹ | 0 ... 510ms |

¹) 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.

6.6.3 Counter - Functions

6.6.3.1 Overview

Counter functions

You can count forwards and backward 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 respectively end value via the parametrization. For every counter additional parametrizable functions are available such as gate function, comparison, hysteresis and hardware interrupt.

Main counting direction

Via the parametrization you have the opportunity to define a main counting direction for every counter. If "none" is selected, the entire counting range is available:

| Limits | Valid range of values |
|-------------------|---------------------------------|
| 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 starts from 0 respectively *load value*, counts in positive direction up to the parametrized end value -1 and then jumps back to *load value* with the next following encoder pulse.
- Main counting direction backward
 - Lower restriction of the count range.
 - The counter starts from parametrized start respectively *load value*, counts in negative direction up to the parametrized end value +1 and then jumps back to start value with the next following encoder pulse.

Gate function abort/interrupt

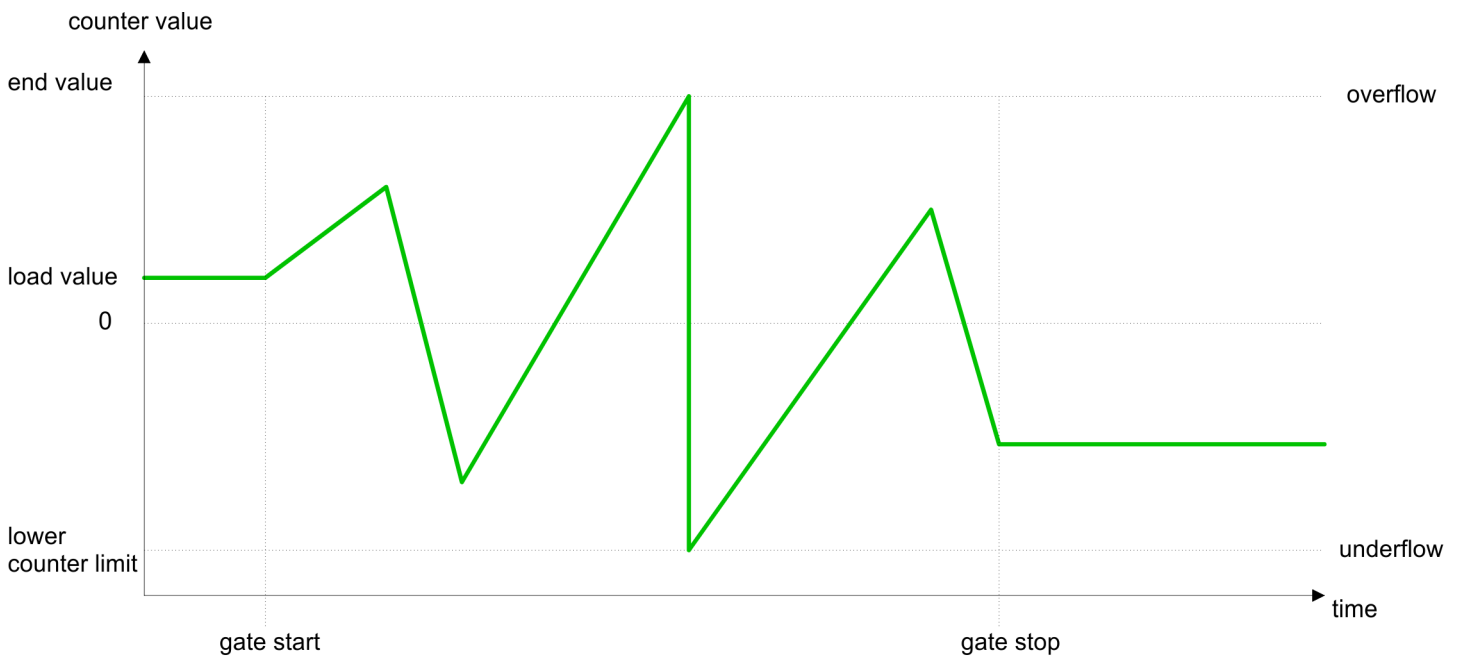
If the HW gate is enabled, the gate functions only affect the HW gate. Opening and closing the software gate has an aborting or interrupting effect:

- Abort count process
 - The count process starts after closing and reopening the gate, beginning with the *load value*.
- Interrupt count process
 - The count process continuous after closing and reopening the gate, beginning with the last recent counter value.

6.6.3.2 Count continuously

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

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



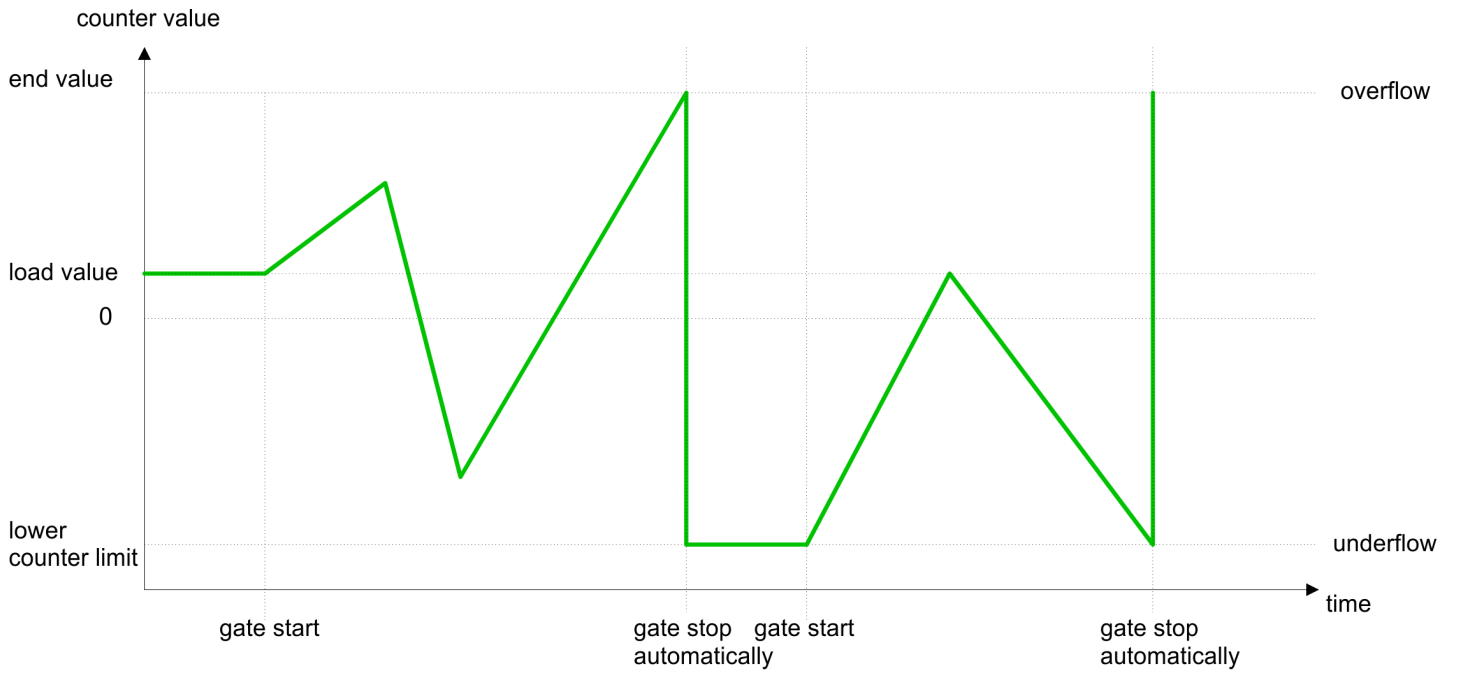
6.6.3.3 Count once

6.6.3.3.1 No main counting direction

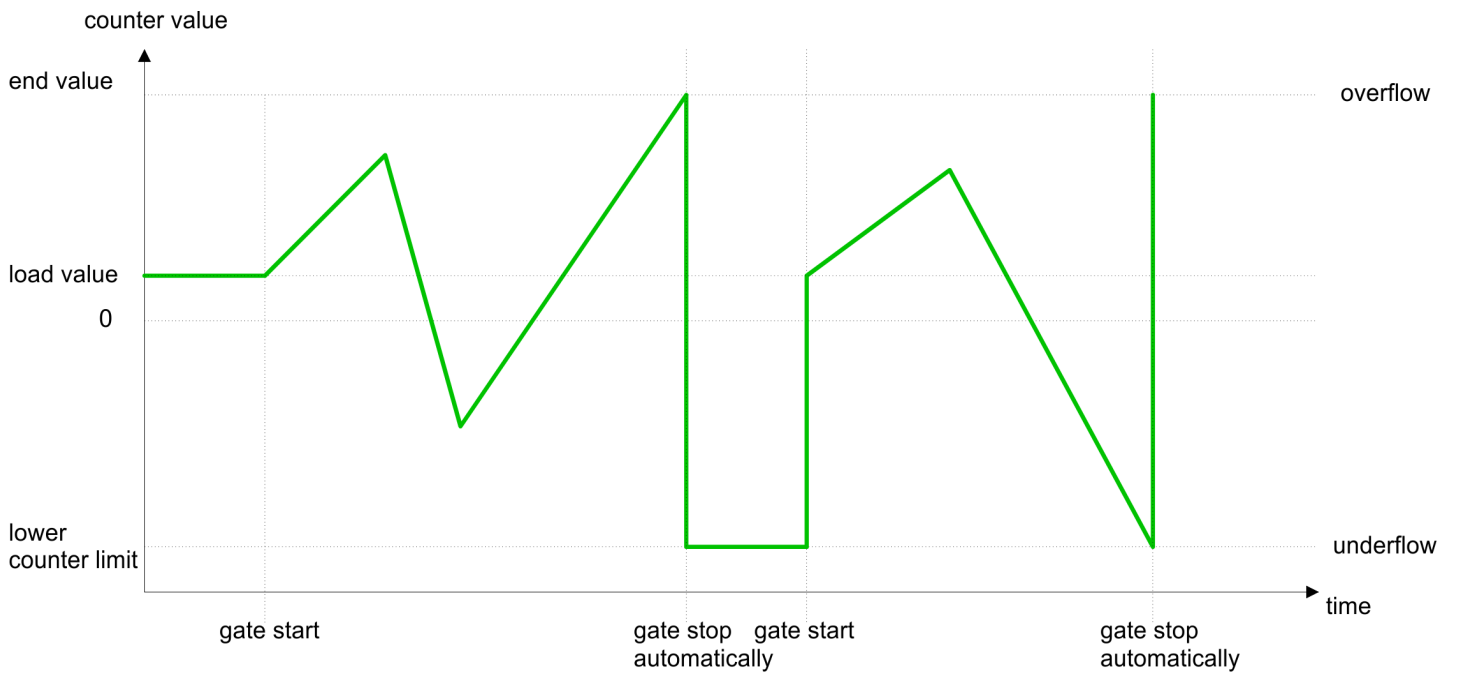
- The counter counts once starting with *load value*.
- It is counted forward or backward.
- The counter limits are fix set to maximum range.
- At over- or underflow at the count limits, the counter jumps to the according other count limit and the gate is automatically closed.
- To restart the count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

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

Interrupting gate control



Aborting gate control



6.6.3.3.2 Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the *end value* -1 in positive direction, it jumps to the *load value* at the next count pulse and the internal gate is automatically closed. If enabled additionally a hardware interrupt is triggered.
- To restart the count process, you have to generate an edge 0-1 at the gate. Then the counter counts again from the *load value*.
- You may exceed the lower count limit.

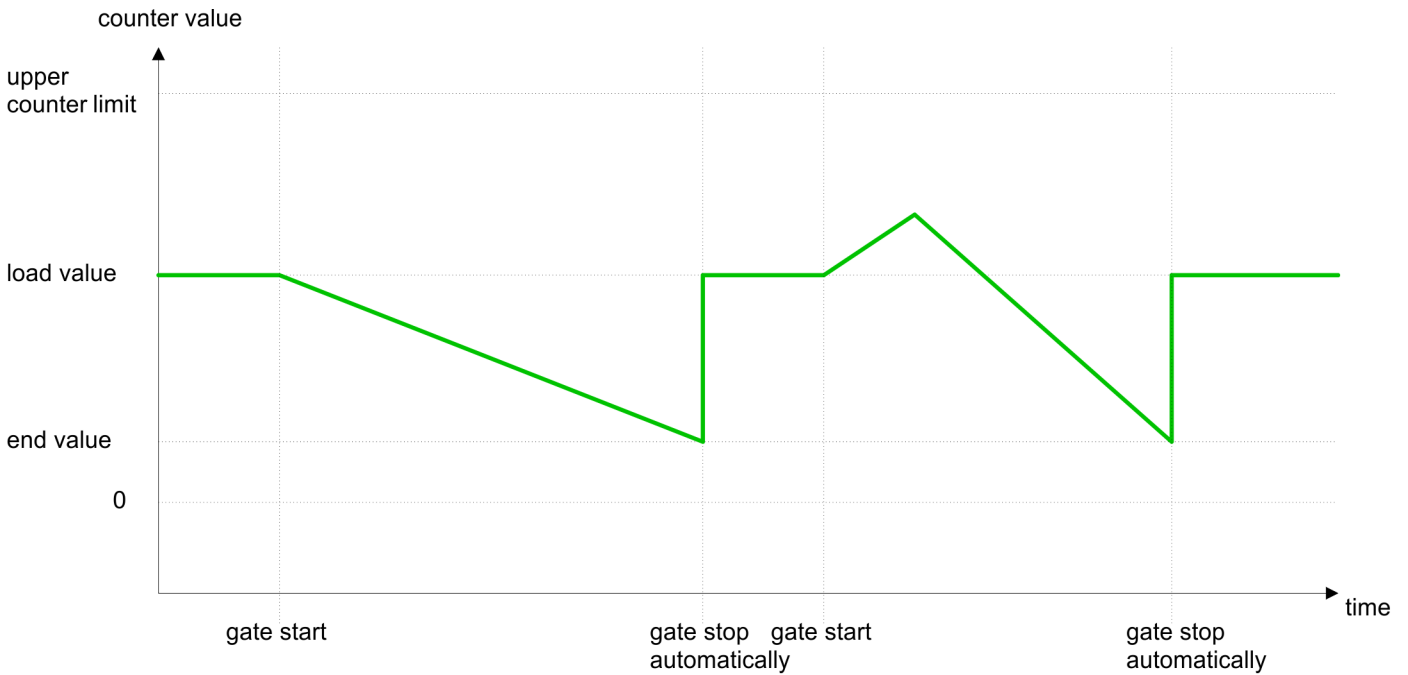
| Limits | Valid range of values |
|-------------------|---|
| End value | -2 147 483 647 ($-2^{31} + 1$) up to +2 147 483 647 ($2^{31} - 1$) |
| Lower count limit | -2 147 483 648 (-2^{31}) |



6.6.3.3.3 Main counting direction backward

- The counter counts backward starting with the *load value*.
- When the counter reaches the end value +1 in negative direction, it jumps to the *load value* with the next count and the internal gate is automatically closed. If enabled additionally a hardware interrupt is triggered.
- To restart the count process, you have to generate an edge 0-1 at the gate. Then the counter counts again from the *load value*.
- You may exceed the upper count limit.

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

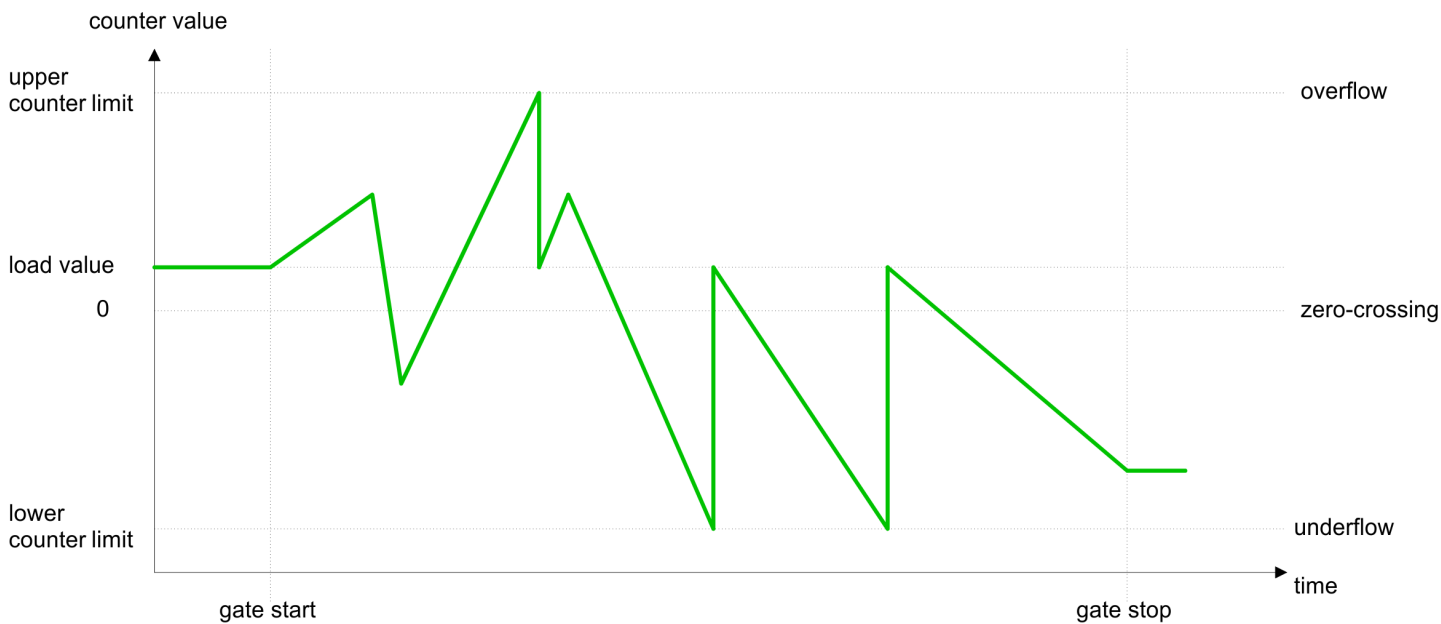


6.6.3.4 Count periodically

6.6.3.4.1 No main counting direction

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

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



6.6.3.4.2 Main counting direction forward

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

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



6.6.3.4.3 Main counting direction backward

- The counter counts backward 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 continues counting. If enabled additionally a hardware interrupt is triggered.
- You may exceed the upper count limit.

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



6.6.4 Counter - Additional functions

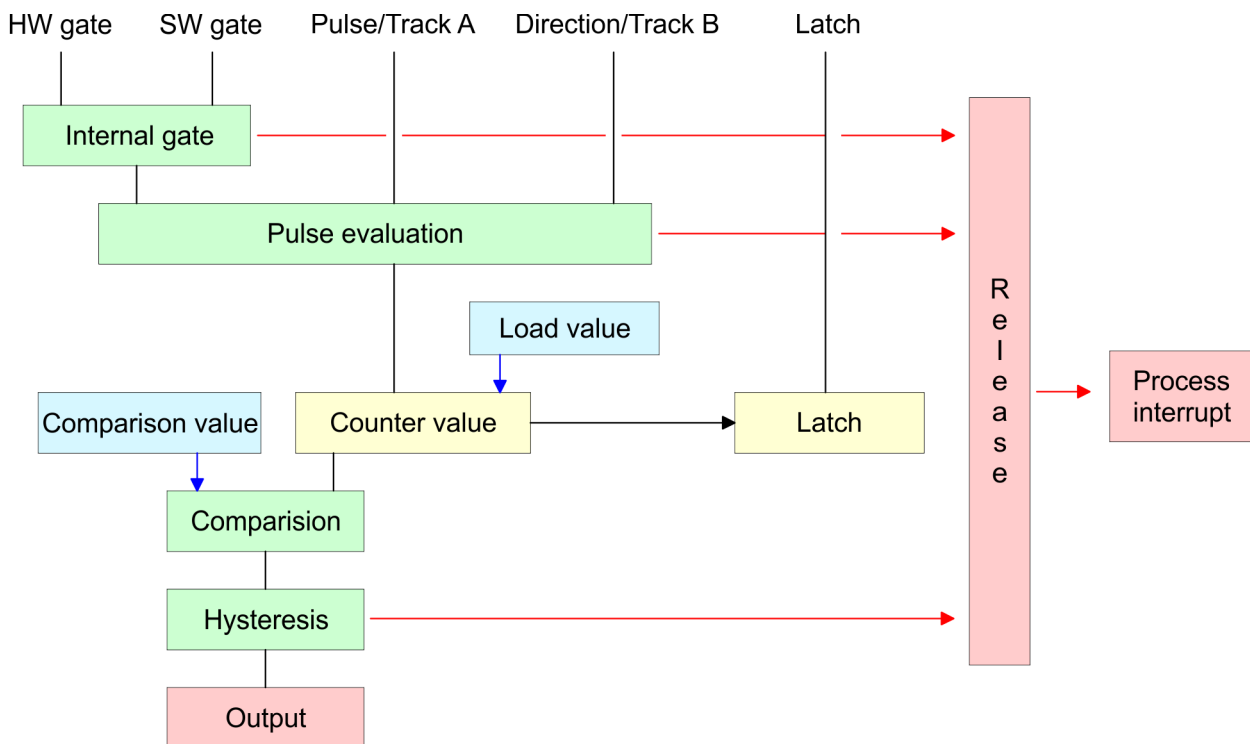
Overview

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

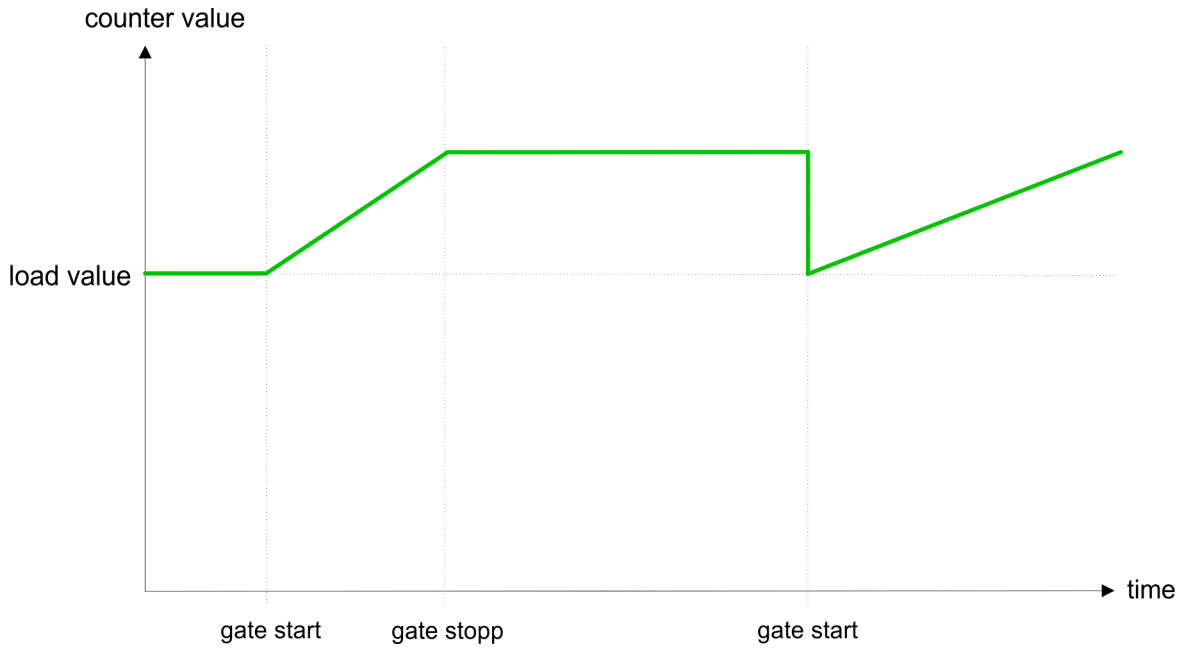
The counter is controlled by the internal gate (i gate). The i gate is the result of logic operation of hardware gate (HW gate) and software gate (SW gate), where the HW gate evaluation may be deactivated by the parameterization.

| | | |
|----------|---------------------|--|
| 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 |

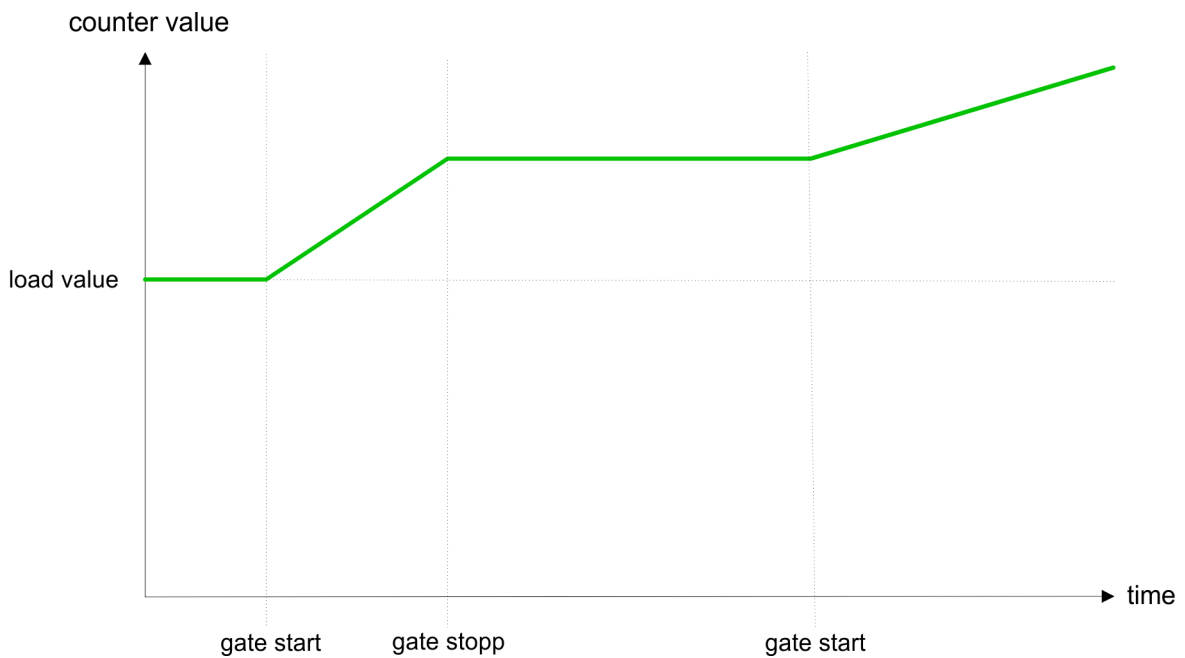
Gate function cancel and stop

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

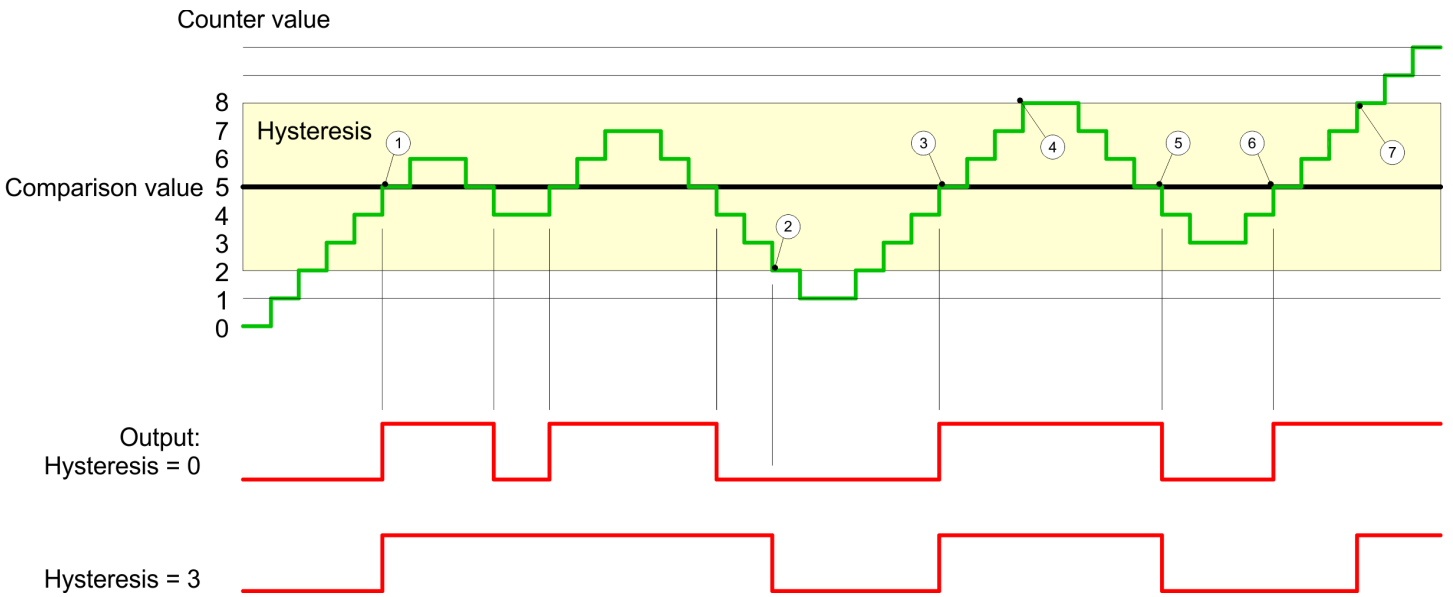
You pre-define the behavior of the counter output via the parameterization:

- 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 \geq comparison value respectively Count \leq comparison value
The output remains set as long as the counter value is higher or equal comparison value respectively lower or equal comparison value. 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.

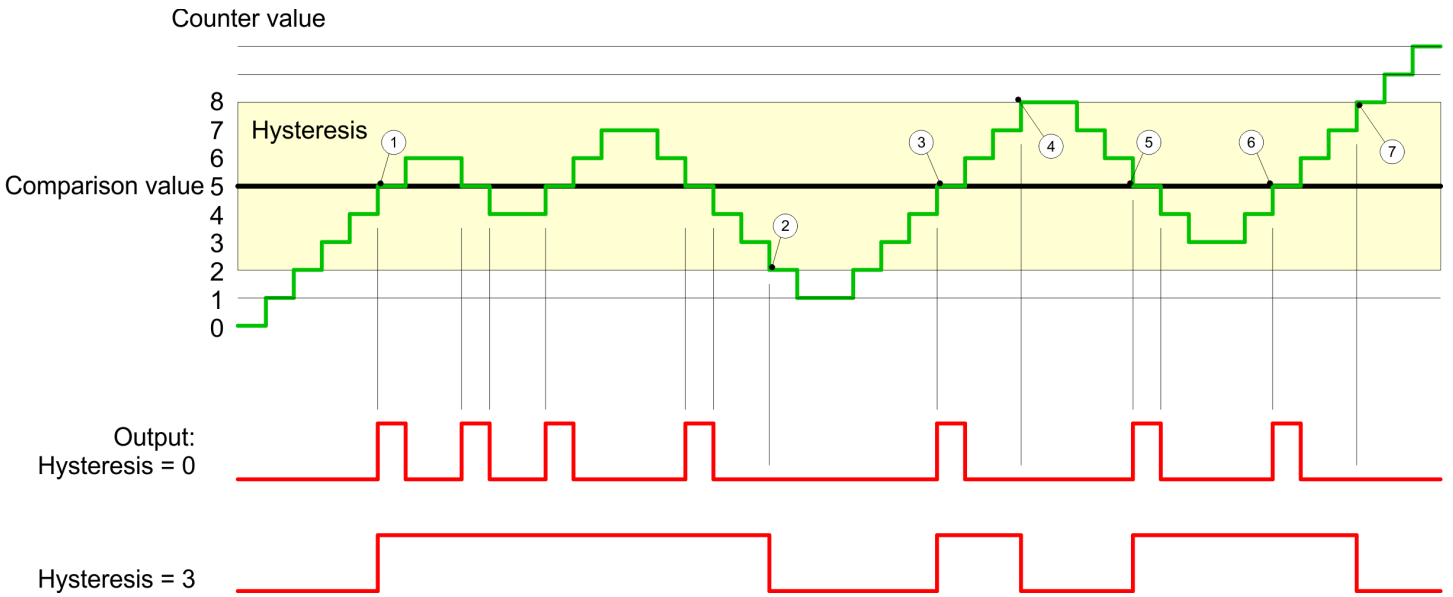
The following pictures illustrate the output behavior for *hysteresis* 0 and *hysteresis* 3 for the according conditions:

Effect at counter value \geq comparison value

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

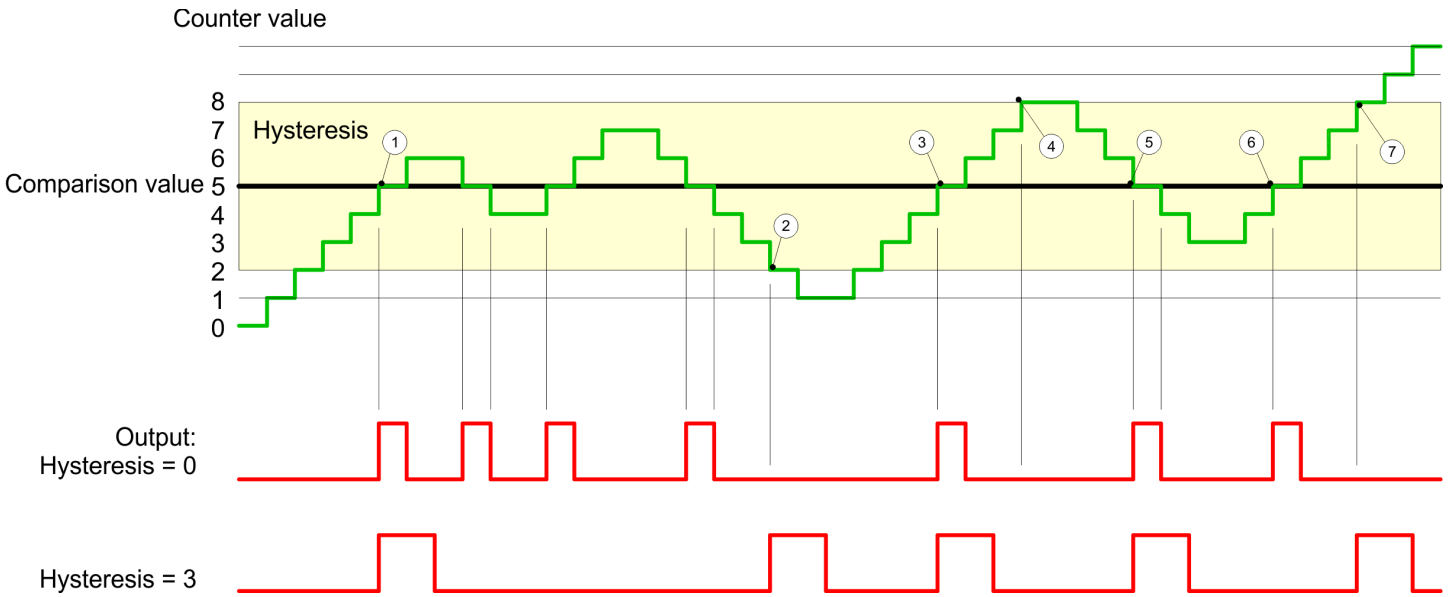
With reaching the comparison condition the *hysteresis* gets active. At active *hysteresis* the comparison result remains unchanged until the counter value leaves the set *hysteresis* range. After leaving the *hysteresis* range a new *hysteresis* is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration Zero



- 1 - Counter value = comparison value → 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

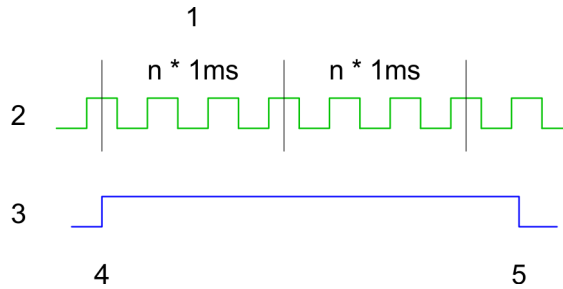
- 1 - Counter value = comparison value → pulse of the parameterized pulse duration is put out, the hysteresis is activated and the counting direction stored
- 2 - Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized pulse duration is put out, the hysteresis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

6.7 Frequency measurement

6.7.1 Overview

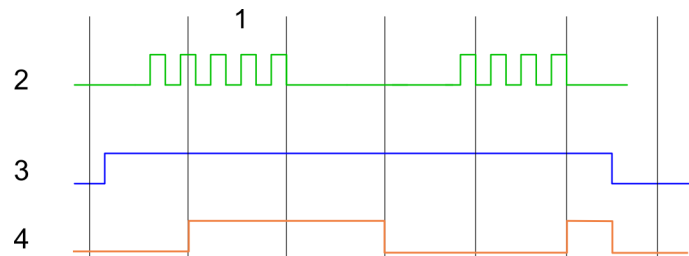
In this operating mode the CPU counts the incoming pulses during a specified integration time and outputs them as frequency value. You can set a value for the integration time between 10ms and 10000ms, in steps of 1 ms. You can set the integration time in the parameter assignment screen forms or you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

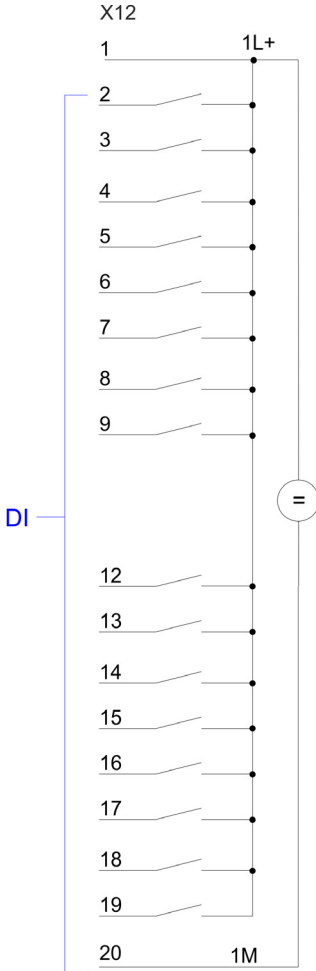


During frequency measurement the count function at the same channel is deactivated.

6.7.2 Inputs for the frequency measurement

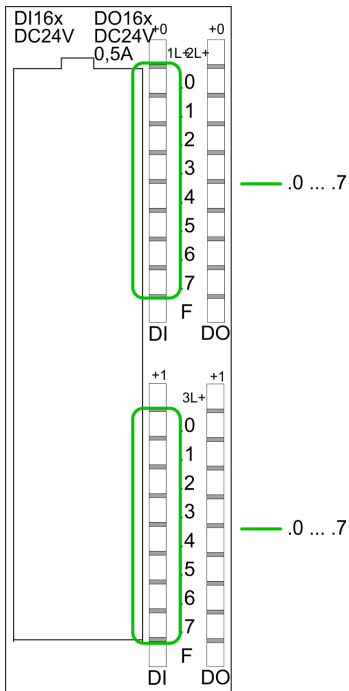
For frequency measurement, connect your signal to be measured at input B.

- Channel 0: Pin 3
- Channel 1: Pin 6
- Channel 2: Pin 9
- Channel 3: Pin 14



Pin assignment X12: 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 / Frequency measurement |
| 4 | I+0.2 / Channel 0 HW gate |
| 5 | I+0.3 / Channel 1 (A) / Pulse |
| 6 | I+0.4 / Channel 1 (B) / Direction / Frequency measurement |
| 7 | I+0.5 / Channel 1 HW gate |
| 8 | I+0.6 / Channel 2 (A) / Pulse |
| 9 | I+0.7 / Channel 2 (B) / Direction / Frequency measurement |
| 10 | not used |
| 11 | not used |
| 12 | I+1.0 / Channel 2 HW gate |
| 13 | I+1.1 / Channel 3 (A) / Pulse |
| 14 | I+1.2 / Channel 3 (B) / Direction / Frequency measurement |
| 15 | I+1.3 / Channel 3 HW gate |
| 16 | I+1.4 / Channel 0 Latch |
| 17 | I+1.5 / Channel 1 Latch |
| 18 | I+1.6 / Channel 2 Latch |
| 19 | I+1.7 / Channel 3 Latch |
| 20 | Ground 1M DI |



Status indication X12: DI

- .07
 - 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

6.7.3 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 314C-2DP (6ES7 314-6CG03-0AB0 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.

6.7.4 SFB 48 - FREQUENC - Frequency measurement

Description

The SFB 48 is a specially developed block for compact CPUs for frequency 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 been 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 |
| 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

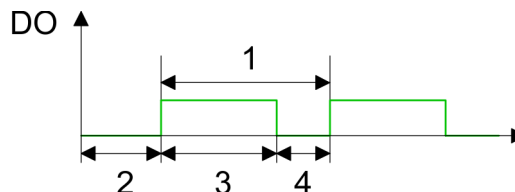
close (deactivate): In the user program by resetting *SW_GATE* of SFB 48

6.8 Pulse width modulation - PWM

6.8.1 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



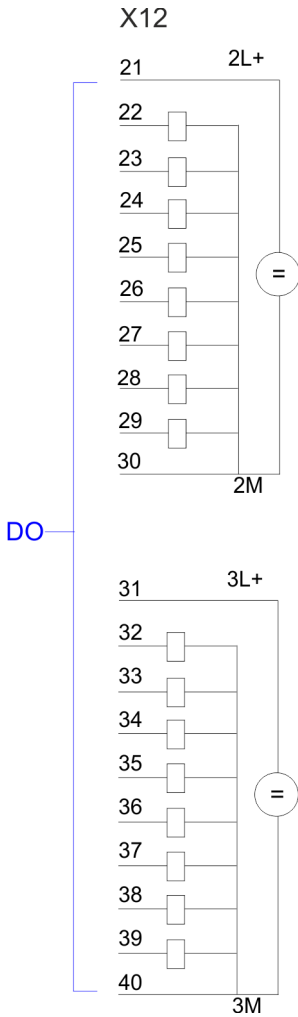
During pulse width modulation the count function at the same channel is deactivated.

PWM Outputs

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

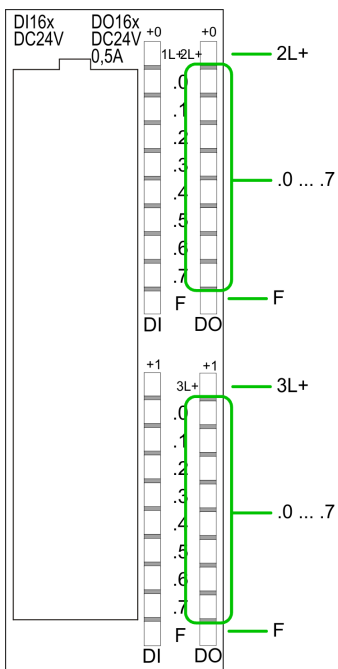
- Channel 0: Pin 22
- Channel 1: Pin 23
- Channel 2: Pin 24
- Channel 3: Pin 25

Connect the common ground to pin 30.



Pin assignment X12: 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 / Channel 2 Output |
| 25 | Q+0.3 / Channel 3 Output |
| 26 | Q+0.4 |
| 27 | Q+0.5 |
| 28 | Q+0.6 |
| 29 | Q+0.7 |
| 30 | Ground 2M DO |
| 31 | 3L+ Power supply +DC 24V |
| 32 | Q+1.0 |
| 33 | Q+1.1 |
| 34 | Q+1.2 |
| 35 | Q+1.3 |
| 36 | Q+1.4 |
| 37 | Q+1.5 |
| 38 | Q+1.6 |
| 39 | Q+1.7 |
| 40 | Ground 3M DO |



Status indication X12: DO

- 2L+, 3L+
 - LED (green)
 - Supply voltage available for DO
- .07
 - LEDs (green)
 - Q+0.0 ... Q+0.7
 - Q+1.0 ... Q+1.7
 - The according LED is on at active output
- F
 - LED (red)
 - Overload or short circuit error

6.8.2 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 314C-2DP (6ES7 314-6CG03-0AB0 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.

6.8.2.1 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:

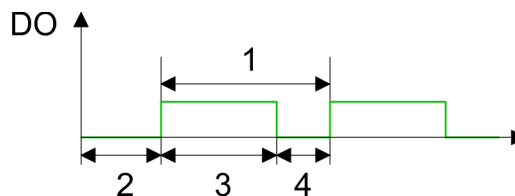
- None: There is no interrupt triggered.
- Process: The counter component triggers a hardware interrupt.
- Diagnostics and Process: With the CPU the diagnostic interrupt of the digital in-/output periphery is only supported in connection with "hardware interrupt lost".



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

6.8.3 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**ulse **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 |
| 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'...page 145 |
| JOB_REQ | INPUT | BOOL | 8.0 | FALSE | Job trigger (edge 0-1) |
| JOB_ID | INPUT | WORD | 10.0 | 0 | Job number ↪ 'JOB_ID'...page 145 |
| 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 <ul style="list-style-type: none"> ■ 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 <ul style="list-style-type: none"> ■ 0: no error ■ 1: Error (see <code>JOB_STAT</code>) |
| JOB_STAT | OUTPUT | WORD | 18.0 | 0 | ↪ 'Return value <code>JOB_STAT</code> '...page 151 |

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
 - $Pulse\ duration = (OUTP_VAL / 1000) \times period\ duration$
- Output format: S7 analog value
 - $Pulse\ duration = (OUTP_VAL / 27648) \times period\ duration$
 - 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

Pulse width modulation - PWM > SFB 49 - PULSE - Pulse width modulation

- 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 (edge 0-1)
- ➔ ■ 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. → To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.

3. → 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 (edge 0-1)
- ➔ ■ 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`

4. → To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.



5. → 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 (edge 0-1)
- ➔ ■ 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`

6. → To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.



7. → Call the SFB 49:

- `SW_EN` = TRUE (edge 0-1)
- `JOB_REQ` = 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`

8.  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.
9.  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!

10.  By resetting of `SW_EN` (`SW_EN = FALSE`) the output is immediately stopped.
11.  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` (edge 0-1)
 - ➔ ■ 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.  To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ = FALSE`.
3.  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` (edge 0-1)
 - ➔ ■ 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`
4.  To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ = FALSE`.
5.  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` (edge 0-1)
 - ➔ ■ 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`
6.  To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ = FALSE`.

7. **→** 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 (edge 0-1)
 - ➔ ■ 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`
8. **→** To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.
9. **→** Call the SFB 49:
 - `SW_EN` = TRUE (edge 0-1)
 - `JOB_REQ` = 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`
10. **→** To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.
11. **→** Call the SFB 49 cyclically:
 - `SW_EN` = FALSE
 - Via `STS_EN` you get the current status of the pulse train output.
12. **→** As soon as `JOB_DONE` returns TRUE, you can transfer additional pulse train jobs by repeating the steps 1 to 6.
13. **→** By resetting of `SW_EN` (`SW_EN` = FALSE) the output is immediately stopped.
14. **→** 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` = TRUE (edge 0-1)
 - `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 (edge 0-1)
 - ➔ ■ 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.  To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.
3.  Call the SFB 49:
 - `SW_EN` = TRUE
 - `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 (edge 0-1)
 - ➔ ■ 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`
4.  To reset `JOB_REQ`, call SFB 49 again with the same parameters and `JOB_REQ` = FALSE.
5.  Call the SFB 49:
 - `SW_EN` = TRUE (edge 0-1)
 - `JOB_ID` = 0Ch: Attach 2. pulse train job to the 1. pulse train job.
 - `JOB_REQ` = TRUE (edge 0-1)
 - `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 | <i>Period duration</i> too small |
| 0412h | <i>Period duration</i> too big |
| 0421h | <i>On-delay</i> too small |
| 0422h | <i>On-delay</i> too big |
| 0431h | <i>Minimum pulse duration</i> too small |
| 0432h | <i>Minimum pulse duration</i> too big |
| 0441h | <i>Number of pulses</i> too small |
| 0442h | <i>Number of pulses</i> too big |
| 0451h | Invalid <i>time base</i> |
| 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 <i>CHANNEL</i> , 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 <i>CHANNEL</i> , which is bigger than 3. |

6.9 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

6.9.1 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 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7 |
| 9 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7 |
| 10 | <ul style="list-style-type: none"> ■ 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 | <ul style="list-style-type: none"> ■ Bit 0: Gate counter 2 open (activated) ■ Bit 1: Gate counter 2 closed ■ Bit 2: Over-/underflow/end value counter 2 ■ Bit 3: Counter 2 reached comparison value ■ Bit 4: Gate counter 3 open (activated) ■ Bit 5: Gate counter 3 closed ■ Bit 6: Over-/underflow/end value counter 3 ■ Bit 7: Counter 3 reached comparison value |

Local double word 8 of OB 40 at frequency measurement

| Local byte | Bit 7...0 |
|------------|--|
| 8 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7 |

| Local byte | Bit 7...0 |
|------------|--|
| 9 | <ul style="list-style-type: none"> ■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7 |
| 10 | <ul style="list-style-type: none"> ■ 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 | <ul style="list-style-type: none"> ■ Bit 0: End of measurement channel 2 (end of the integration time) ■ Bit 3 ... 1: reserved ■ Bit 4: End of measurement channel 3 (end of the integration time) ■ Bit 7 ... 5: reserved |

6.9.2 Diagnostic interrupt

Activation

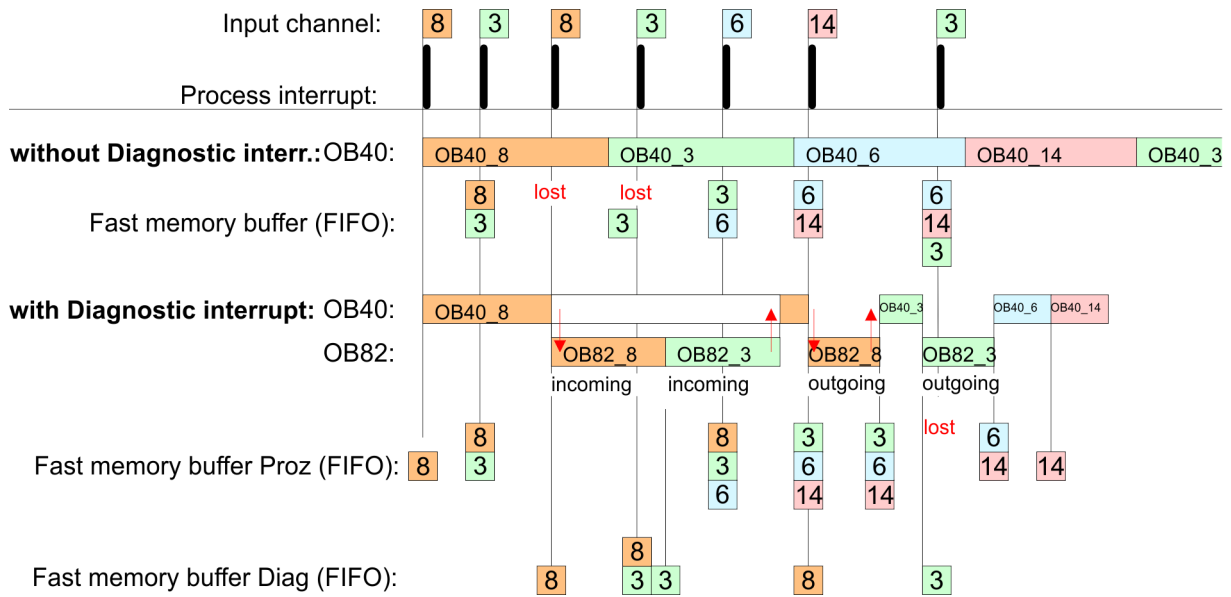


Please consider that diagnostic interrupts are enabled only if you have selected one of the technology functions (counting, frequency measurement, PWM) and set "Diagnostics+Process interrupt" in the basic parameters.

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 | <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> Bit 3 ... 0: Module class <ul style="list-style-type: none"> 0101b: Analog 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix) |
| 2 | <ul style="list-style-type: none"> Bit 7 ... 0: 0 (fix) |
| 3 | <ul style="list-style-type: none"> Bit 5 ... 0: 0 (fix) Bit 6: Process interrupt lost Bit 7: 0 (fix) |

Record set 0 Diagnostic_{outgoing}

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

| Byte | Bit 7...0 |
|------|--|
| 0 | <ul style="list-style-type: none"> ■ Bit 0: set at module failure ■ Bit 1: 0 (fix) ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 7 ... 4: 0 (fix) |
| 1 | <ul style="list-style-type: none"> ■ Bit 3 ... 0: Module class <ul style="list-style-type: none"> - 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}'...page 154 |
| 4 | <ul style="list-style-type: none"> ■ Bit 6 ... 0: Channel type (here 70h) <ul style="list-style-type: none"> - 70h: Digital input - 71h: Analog input - 72h: Digital output - 73h: Analog output - 74h: Analog input/output ■ Bit 7: More channel types present <ul style="list-style-type: none"> - 0: no - 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 08h) |
| 7 | <ul style="list-style-type: none"> ■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 2 (I+1.4 ... I+1.7) ■ Bit 4: Error in channel group 4 (counter 0) ■ Bit 5: Error in channel group 5 (counter 1) ■ Bit 6: Error in channel group 6 (counter 2) ■ Bit 7: Error in channel group 7 (counter 3) |

| Byte | Bit 7...0 |
|------|--|
| 8 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix) |
| 9 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix) |
| 10 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix) |
| 11 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix) |
| 12 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ 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) |

| Byte | Bit 7...0 |
|------|--|
| 13 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ 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 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... gate counter 2 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 2 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value counter 2 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 2 reached comparison value ■ Bit 7: 0 (fix) |
| 15 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... gate counter 3 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... gate counter 3 opened ■ Bit 3: 0 (fix) ■ Bit 4: ... over-/underflow/end value counter 3 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 3 reached comparison value ■ Bit 7: 0 (fix) |

Diagnostic Record set 1 at *frequency measurement*

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

| Byte | Bit 7...0 |
|---------|--|
| 0 ... 3 | Content record set 0 → 'Record set 0 Diagnostic_{incoming}'...page 154 |
| 4 | <ul style="list-style-type: none"> ■ Bit 6 ... 0: Channel type (here 70h) <ul style="list-style-type: none"> - 70h: Digital input - 71h: Analog input - 72h: Digital output - 73h: Analog output - 74h: Analog input/output ■ Bit 7: More channel types present <ul style="list-style-type: none"> - 0: no - 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 08h) |

| Byte | Bit 7...0 |
|------|--|
| 7 | <ul style="list-style-type: none"> ■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 2 (I+1.4 ... I+1.7) ■ Bit 4: Error in channel group 4 (Frequency meter 0) ■ Bit 5: Error in channel group 5 (Frequency meter 1) ■ Bit 6: Error in channel group 6 (Frequency meter 2) ■ Bit 7: Error in channel group 7 (Frequency meter 3) |
| 8 | <p>Diagnostic interrupt due to "process interrupt lost" at...</p> <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix) |
| 9 | <p>Diagnostic interrupt due to "process interrupt lost" at...</p> <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix) |
| 10 | <p>Diagnostic interrupt due to "process interrupt lost" at...</p> <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix) |
| 11 | <p>Diagnostic interrupt due to "process interrupt lost" at...</p> <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix) |

| Byte | Bit 7...0 |
|------|--|
| 12 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> Bit 0: End of measurement channel 0 (End of integration time) Bit 7 ... 1: 0 (fix) |
| 13 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> Bit 0: End of measurement channel 1 (End of integration time) Bit 7 ... 1: 0 (fix) |
| 14 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> Bit 0: End of measurement channel 2 (End of integration time) Bit 7 ... 1: 0 (fix) |
| 15 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> Bit 0: End of measurement channel 3 (End of integration time) Bit 7 ... 1: 0 (fix) |

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}'...page 154 |
| 4 | <ul style="list-style-type: none"> Bit 6 ... 0: Channel type (here 70h) <ul style="list-style-type: none"> 70h: Digital input 71h: Analog input 72h: Digital output 73h: Analog output 74h: Analog input/output Bit 7: More channel types present <ul style="list-style-type: none"> 0: no 1: yes |
| 5 | Number of diagnostic bits per channel (here 08h) |
| 6 | Number of channels of a module (here 08h) |
| 7 | <ul style="list-style-type: none"> Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) Bit 7 ... 4: reserved |
| 8 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> Bit 0: ... input I+0.0 Bit 1: 0 (fix) Bit 2: ... input I+0.1 Bit 3: 0 (fix) Bit 4: ... input I+0.2 Bit 5: 0 (fix) Bit 6: ... input I+0.3 Bit 7: 0 (fix) |

| Byte | Bit 7...0 |
|-----------|---|
| 9 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix) |
| 10 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix) |
| 11 | Diagnostic interrupt due to "process interrupt lost" at... <ul style="list-style-type: none"> ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix) |
| 12 ... 15 | <ul style="list-style-type: none"> ■ Bit 7... 0: reserved |

7 Deployment PtP communication

7.1 Fast introduction

| | |
|---|--|
| General | <p>One of the RS485 interfaces of the CPU 314-6CG23 may be set to PtP communication (point-to-point) by means of a hardware configuration.</p> <ul style="list-style-type: none"> ■ PtP functionality <ul style="list-style-type: none"> – 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 |



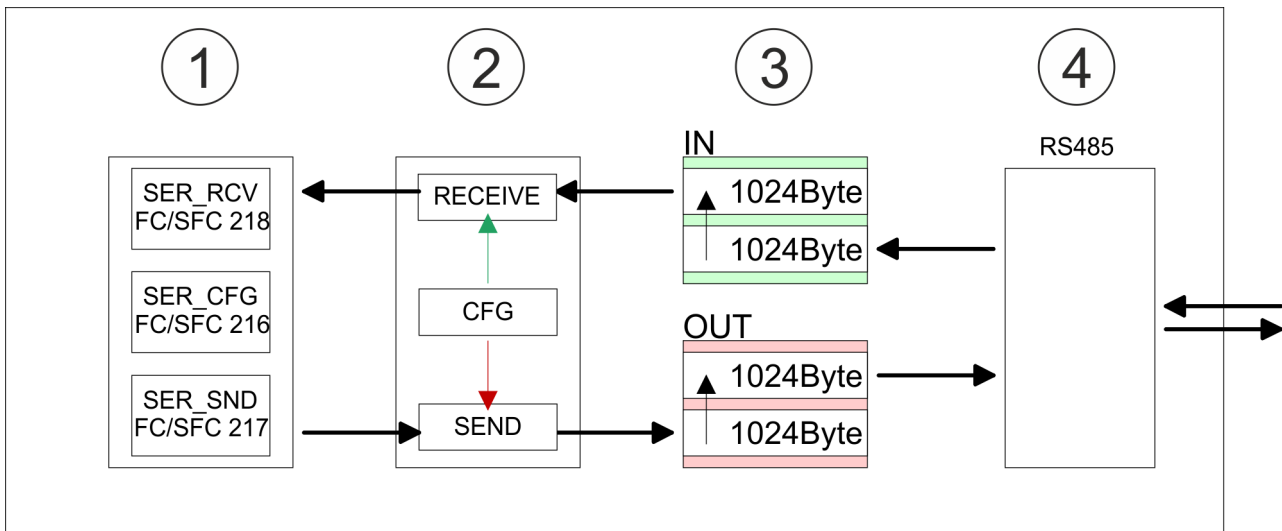
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List".

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

Switch to PtP operation

Per default, the RS485 interfaces of the CPU are used for MPI interface or PROFIBUS DP master interface. Via hardware configuration the RS485 interfaces may be switched to point-to-point (PtP) communication via the Parameter 'Function RS485 ...' of the Properties.

Requirements

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

Installation of the SPEEDBUS.GSD

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

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

The GSD files may be found at www.yaskawa.eu.com in the 'Download Center'.

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

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

Proceeding

Standard bus

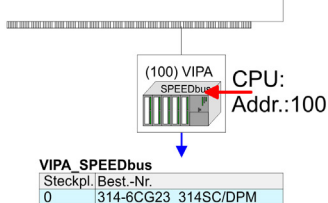
| Slot | Module |
|------|---------------------|
| 1 | |
| 2 | CPU 314C-2DP |
| X2 | |
| 2.2 | |
| 2.3 | |
| 2.4 | |
| 2.5 | |
| 3 | |

Modules at the bus

343-1EX11
Ethernet PG/OP

always as last module
342-5DA02 V5.0

virtual DP-Master for CPU



Object properties

The embedding of the CPU 314-6CG23 happens by means of a virtual PROFIBUS master system with the following approach:

1. Perform a hardware configuration for the CPU. ➔ ['Hardware configuration - CPU'...page](#)
2. Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".
3. Connect the slave system "VIPA_SPEEDbus". After installing the SPEEDBUS.GSD this may be found in the hardware catalog at PROFIBUS DP / Additional field devices / I/O / VIPA / VIPA_SPEEDBUS.
4. For the slave system set the PROFIBUS address 100.
5. Configure at slot 0 the CPU 314-6CG23 of the hardware catalog from VIPA_SPEEDbus.
6. By double clicking the placed CPU 314-6CG23 the properties dialog of the CPU may be opened.

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



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

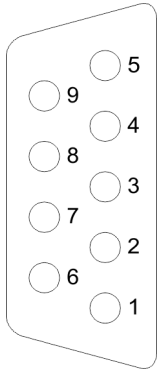
Setting PtP parameters

1. By double clicking the CPU 314-6CG23 placed in the slave system the properties dialog of the CPU may be opened.
2. Switch the Parameter 'Function RS485...' to 'PtP'.

Properties RS485

- Logical states as voltage difference between 2 twisted wires
- 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

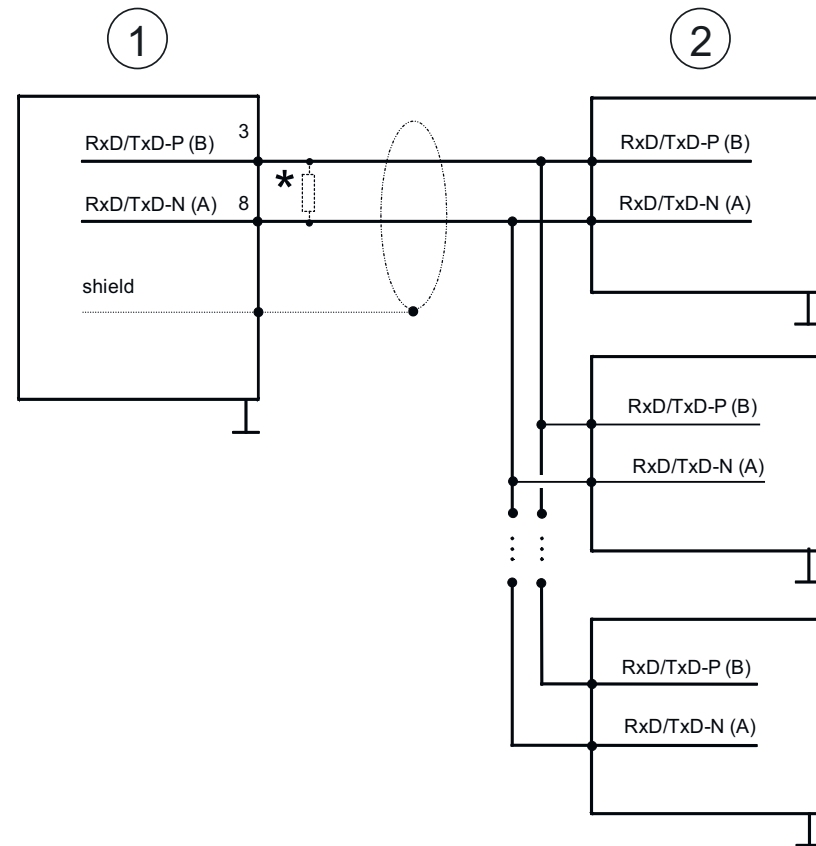
RS485



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".

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".

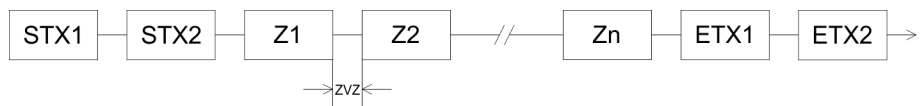
STX/ETX

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

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

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

Message structure:



3964

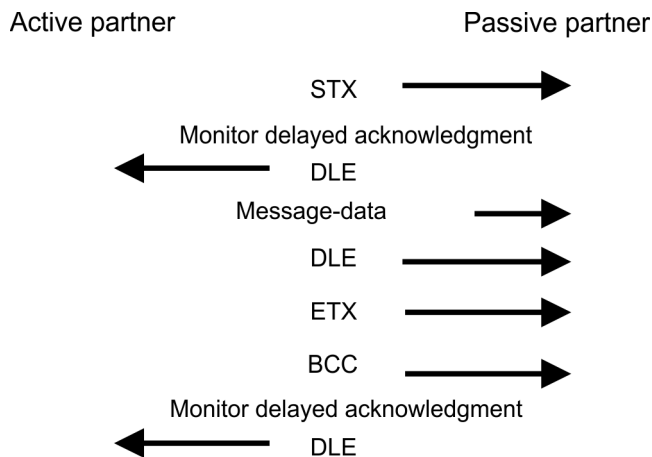
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

The procedure employs the following control characters:

- STX: **Start of Text**
- DLE: **Data Link Escape**
- ETX: **End of Text**
- BCC: **Block Check Character**
- NAK: **Negative Acknowledge**

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

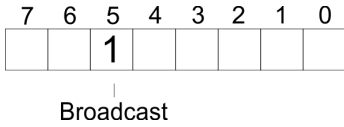
| STX | LGE | ADR | PKE | | IND | | PWE | | ZSW | | HIW | | BCC |
|-----|-----|-----|-----|---|-----|---|-----|---|-----|---|-----|---|-----|
| 02h | | | H | L | H | L | H | L | H | L | H | L | |

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

Protocols and procedures

Broadcast with set bit 5 in ADR byte



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

Modbus

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

Telegram structure

| | | | | | |
|------------|---------------|---------------|------|--------------|----------|
| Start sign | Slave address | Function Code | Data | Flow control | End sign |
|------------|---------------|---------------|------|--------------|----------|

Broadcast with slave address = 0

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

ASCII, RTU mode

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

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

Supported Modbus protocols

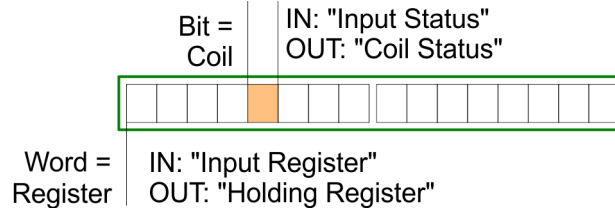
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

7.7 Modbus - Function codes

Naming convention

Modbus has some naming conventions:

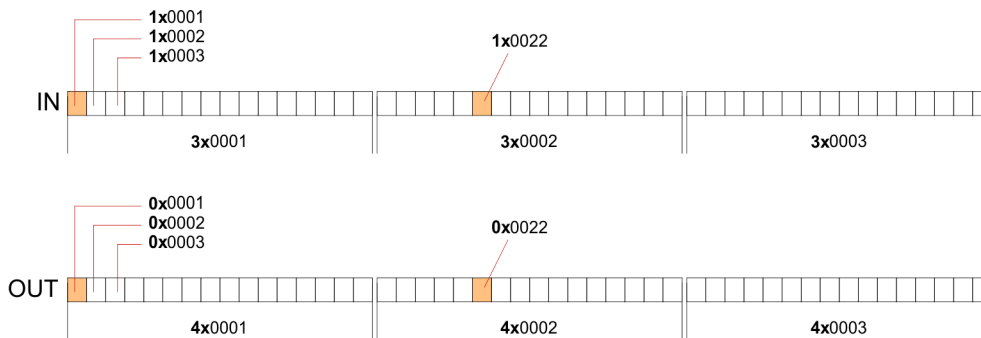


- Modbus differentiates between bit and word access; bits = "Coils" and words = "Register".
- Bit inputs are referred to as "Input-Status" and bit outputs as "Coil-Status".
- word inputs are referred to as "Input-Register" and word outputs as "Holding-Register".

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 is not differentiating digital and analog data, the following assignment is valid:

- 0x - Bit area for master output data
Access via function code 01h, 05h, 0Fh
- 1x - Bit area for master input data
Access via function code 02h
- 3x - word area for master input data
Access via function code 04h
- 4x - word area for master output data
Access via function code 03h, 06h, 10h



A description of the function codes follows below.

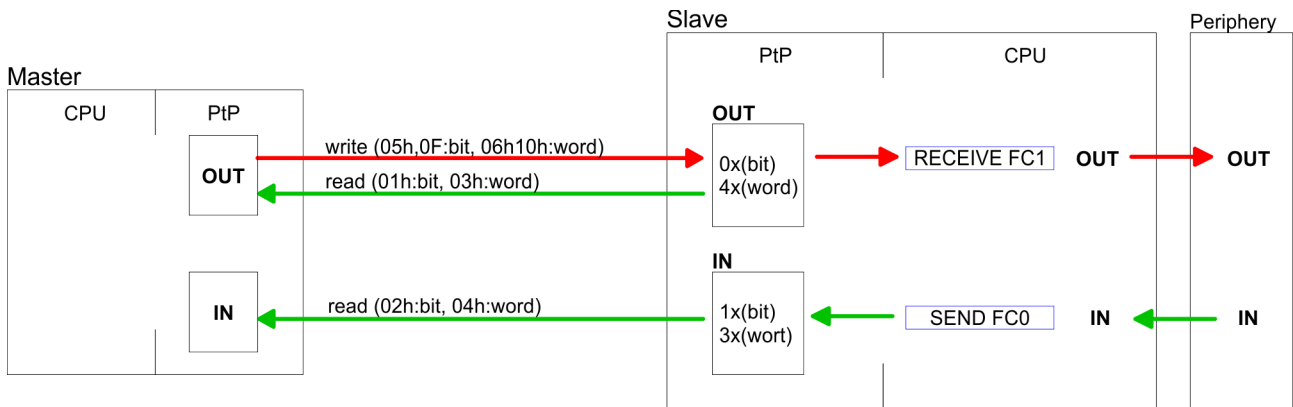
Overview

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

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

Modbus - Example communication

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:

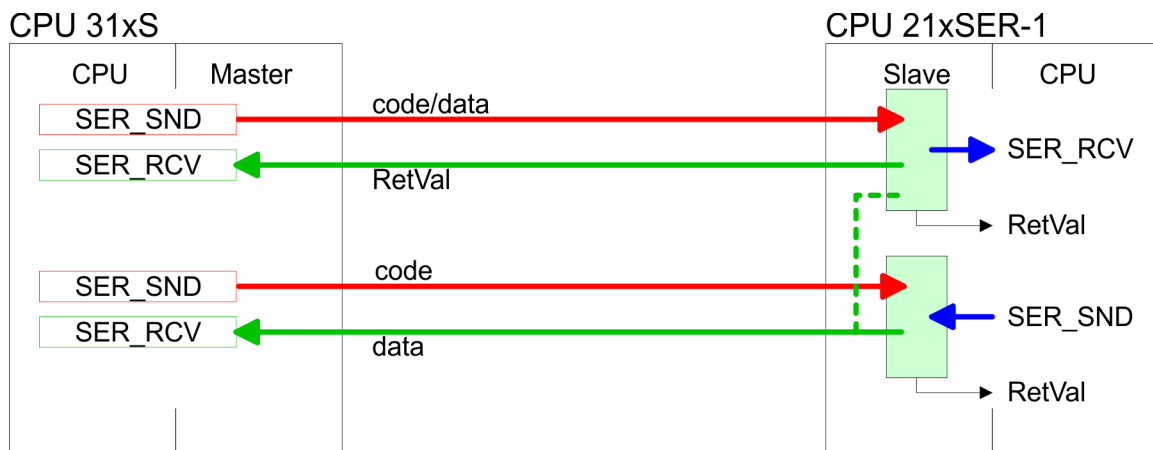
- CPU 31xS as Modbus RTU master
- CPU 21xSER-1 as Modbus RTU slave
- Siemens SIMATIC Manager and possibilities for the project transfer
- Modbus cable connection

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.

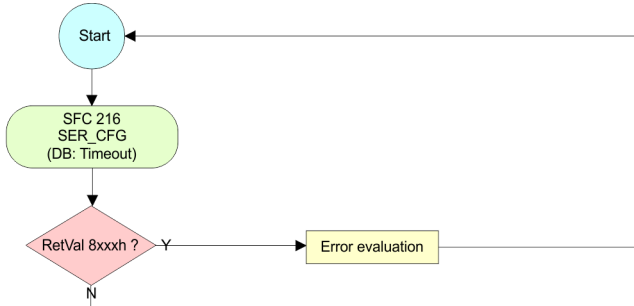
3. → Execute the project engineering of the slave! The PLC user application at the slave has the following structure:
- OB 100:
Call SFC 216 (configuration as Modbus RTU slave) with timeout setting and Modbus address in the DB and error evaluation.
 - OB 1:
Call SFC 217 (SER_SND) for data transport from 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:

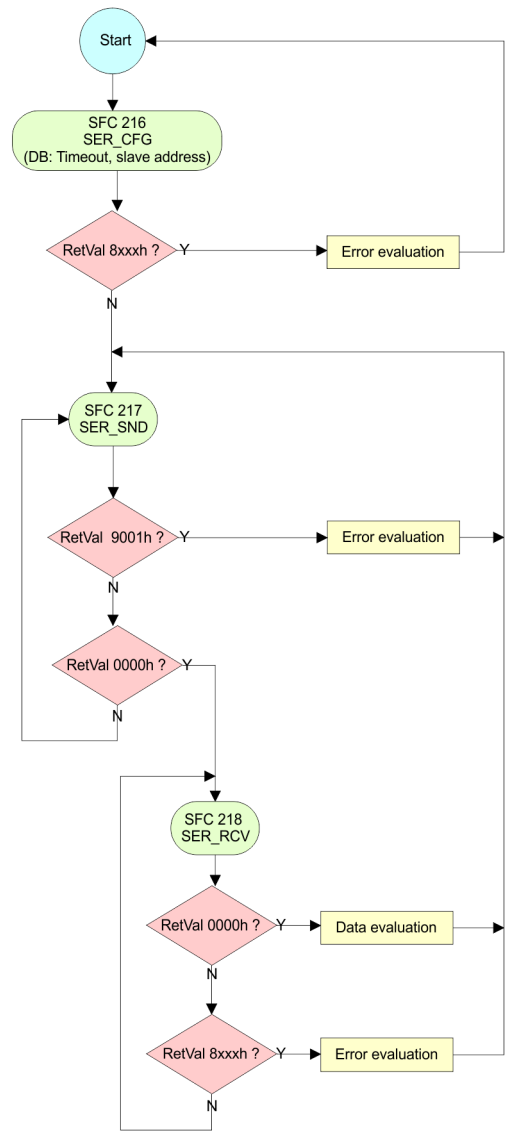
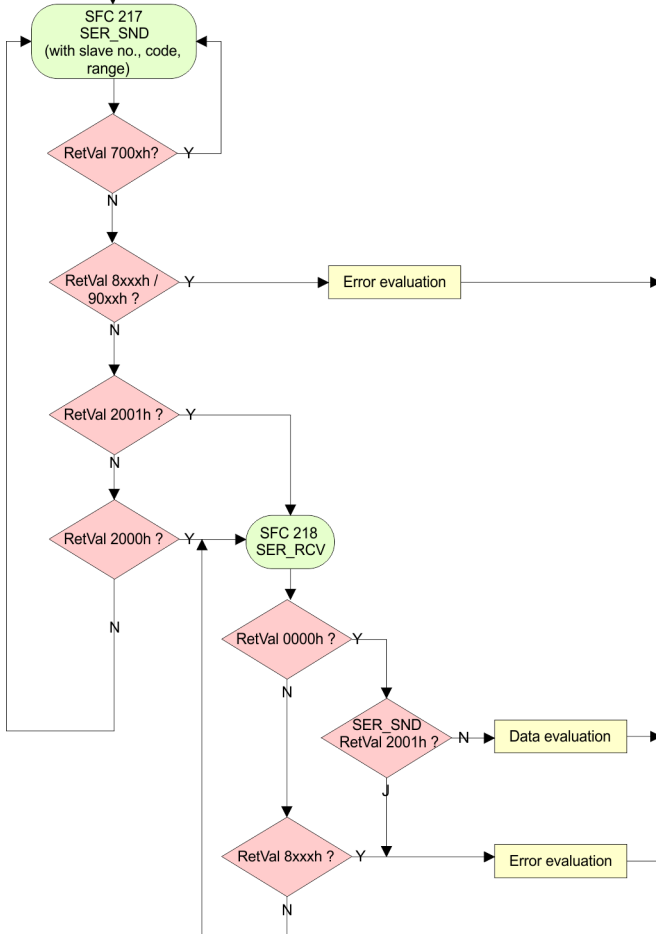


Modbus - Example communication

OB100:



OB1:



8 Deployment PROFIBUS communication

8.1 Overview

PROFIBUS DP

- PROFIBUS is an international standard applicable to an open and serial field bus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.
- PROFIBUS comprises an assortment of compatible versions. The following details refer to PROFIBUS DP.
- PROFIBUS DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. PROFIBUS DP was designed for high-speed data communication on the sensor-actuator level.
- The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slaves.

CPU with DP master

The PROFIBUS DP master is to be configured in the hardware configurator from Siemens. Therefore the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU.

After the transmission of the data to the CPU, the configuration data are internally passed on to the PROFIBUS master part.

During the start-up the DP master automatically includes his data areas into the address range of the CPU. Project engineering in the CPU is not required.

Deployment of the DP master with CPU

Via the PROFIBUS DP master PROFIBUS DP slaves may be coupled to the CPU. The DP master communicates with the DP slaves and links up its data areas with the address area of the CPU.

At every POWER ON res. overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the ER-LED is on and the OB 86 is requested. If this is not available, the CPU switches to STOP and BASP is set. As soon as the BASP signal comes from the CPU, the DP master is setting the outputs of the connected periphery to zero. The DP master remains in the operating mode RUN independent from the CPU.

DP slave operation

For the deployment in a super-ordinated master system you first have to project your slave system as Siemens CPU in slave operation mode with configured in-/output areas. Afterwards you configure your master system. Couple your slave system to your master system by dragging the CPU 31x from the hardware catalog at *Configured stations* onto the master system, choose your slave system and connect it.

8.2 Fast introduction

Overview

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

Steps of configuration

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

- **Hardware configuration - CPU**
- **Deployment as DP master or Deployment as DP slave**
- **Transfer of the complete project to CPU** → ['Project transfer'...page 64](#)



In the Siemens SIMATIC Manager the CPU 314-6CG23 from Yaskawa is to be configured as

CPU 314C-2DP (6ES7 314-6CG03-0AB0 V2.6)

The integrated PROFIBUS DP master is to be configured and connected via the sub module X2 (DP). The Ethernet PG/OP channel of the 314-6CG23 is always to be configured as 1. module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens.

8.3 Hardware configuration - CPU

Precondition

The configuration of the CPU takes place at the Siemens 'hardware configurator'. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering. The modules, which may be configured here are listed in the hardware catalog. If necessary you have to update the hardware catalog with 'Options → Update Catalog'.

For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required.



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

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

Proceeding

| Slot | Module |
|------|----------------------|
| 1 | |
| 2 | CPU 314C-2 DP |
| X2 | DP |
| 2.2 | DI24/DO16 |
| 2.3 | AI5/AO2 |
| 2.4 | Count |
| 2.5 | Position |
| 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 314C-2DP (6ES7 314-6CG03-0AB0 V2.6).
4. → The integrated PROFIBUS DP master is to be configured and connected via the sub module X2 (DP).

8.4 Deployment as PROFIBUS DP master

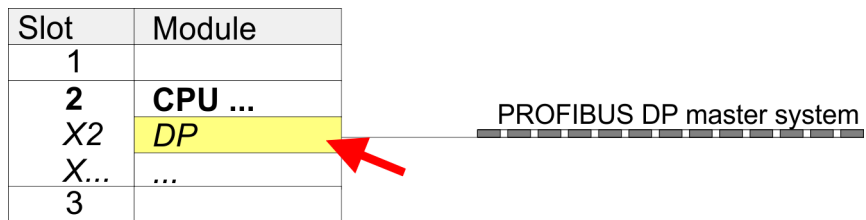
Precondition

The hardware configuration described before was established.

Proceeding

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

→ A PROFIBUS DP master system is inserted:



Now the project engineering of your PROFIBUS DP master is finished. Please link up now your DP slaves with periphery to your DP master.

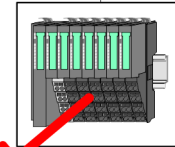
1. → For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.

Deployment as PROFIBUS DP slave

2. ➔ Assign a valid PROFIBUS address to the DP slave.
3. ➔ Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
4. ➔ If needed, parameterize the modules.
5. ➔ Save, compile and transfer your project.

| Slot | Module |
|------|----------------|
| 1 | |
| 2 | CPU ... |
| X2 | <i>DP</i> |
| X... | ... |
| 3 | |

PROFIBUS DP master system



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

8.5 Deployment as PROFIBUS DP slave

Fast introduction

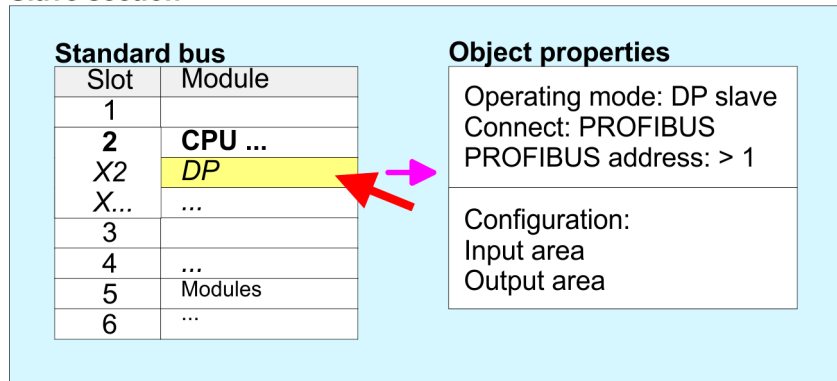
In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

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

Project engineering of the slave section

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

Slave section

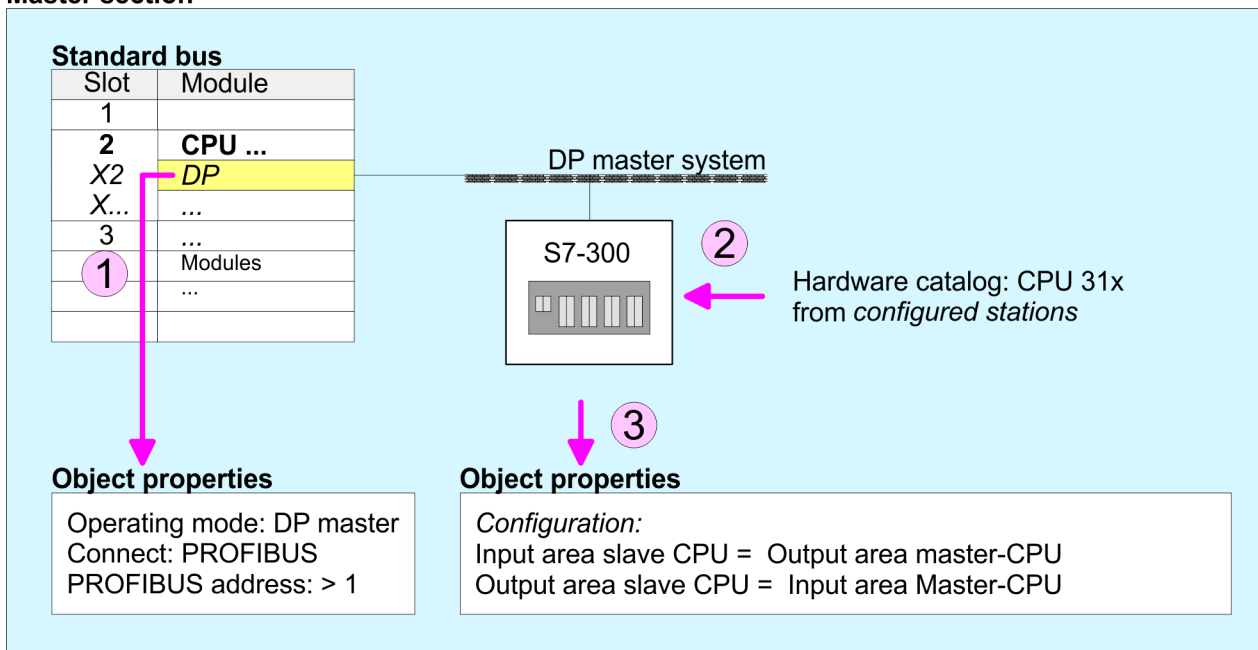


Project engineering of the master section

DP master and DP slave are in the same project

1. Insert another station and configure a CPU.
2. Designate the station as "...DP master".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at 'DP'.
5. Set Interface: type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2) and confirm with [OK].
7. Switch at Operating mode to "DP master" and confirm the dialog with [OK].
8. Connect your slave system to this master system by dragging the "CPU 31x" from the hardware catalog at *Configured stations* onto the master system and select your slave system to be coupled.
9. Open the *Configuration* at *Object properties* of your slave system.
10. Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.
11. Save, compile and transfer your project to your CPU.

Master section



DP master and DP slave are in different projects

1. Create a new project, add a station and configure a CPU.
2. Designate the station as "...DP master".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at 'DP'.
5. Set Interface: type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2) and confirm with [OK].
7. Switch at Operating mode to "DP master" and confirm the dialog with [OK].
8. For further configuration, install the GSD file from the appropriately configured Siemens slave CPU.
9. Choose via 'Additional field devices → PLC → SIMATIC' the Siemens slave CPU.
10. Connect your slave system to the master system by dragging the slave CPU via PROFIBUS onto the master system.
11. Via the slots configure the I/O area of your slave system.
12. Save, compile and transfer your project to your CPU.

8.6 PROFIBUS installation guidelines

PROFIBUS in general

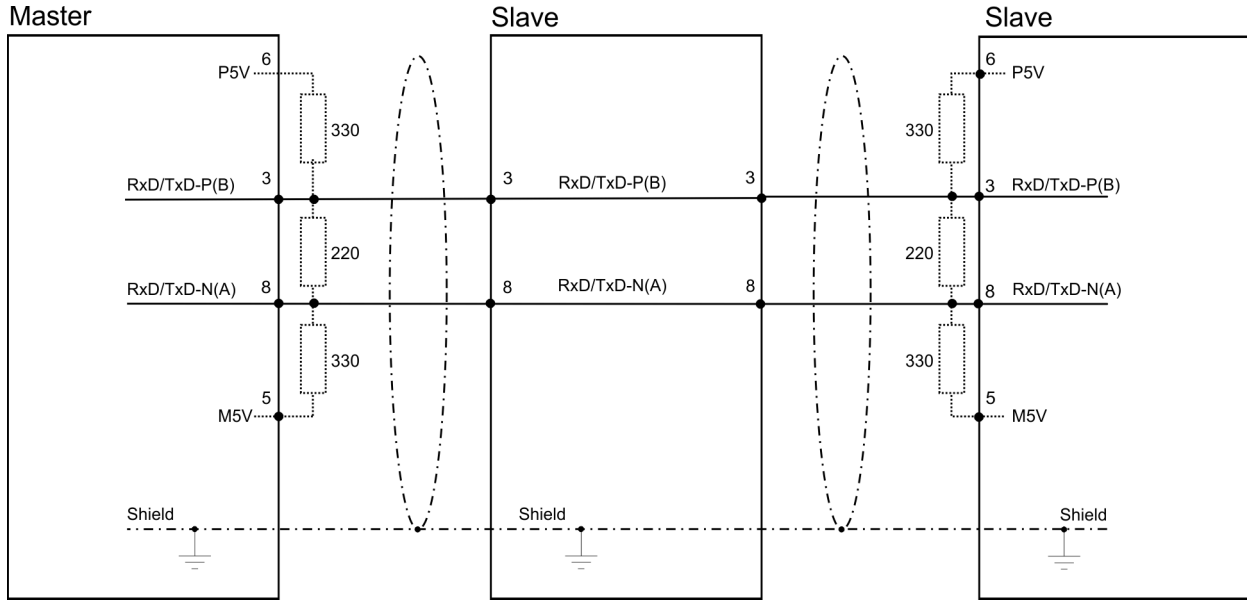
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:
 - 9.6 ... 187.5bit/s → 1000m
 - 500kbit/s → 400m
 - 1.5Mbit/s → 200m
 - 3 ... 12Mbit/s → 100m
- Max. 10 segments may be built up. The segments are connected via repeaters. Every repeater counts for one participant.
- The bus respectively a segment is to be terminated at both ends.
- All participants are communicating with the same transfer rate. The slaves adjust themselves automatically on the transfer rate.

Transfer medium

- As transfer medium PROFIBUS uses an isolated twisted-pair cable based upon the RS485 interface.
- The RS485 interface is working with voltage differences. Though it is less irritable from influences than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure.
- Max. 32 participants per segment are permitted. Within a segment the members are linear connected. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.
- PROFIBUS DP uses a transfer rate between 9.6kbit/s and 12Mbit/s, the slaves are following automatically. All participants are communicating with the same transfer rate.
- The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

Bus connection

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

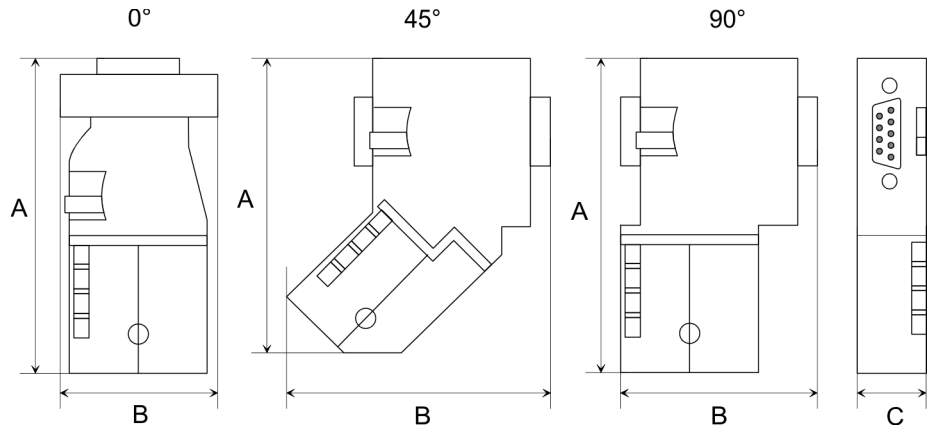


The PROFIBUS line has to be terminated with its ripple resistor. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.

EasyConn bus connector



In PROFIBUS all participants are wired parallel. For that purpose, the bus cable must be feed-through. Via the order number 972-0DP10 you may order the bus connector "EasyConn" from Yaskawa. This is a bus connector with switchable terminating resistor and integrated bus diagnostic.



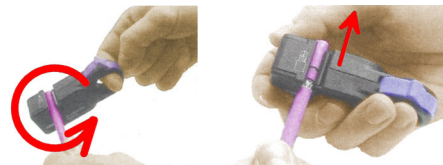
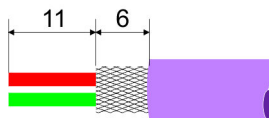
| Dimensions in mm | 0° | 45° | 90° |
|------------------|------|------|------|
| A | 64 | 61 | 66 |
| B | 34 | 53 | 40 |
| C | 15.8 | 15.8 | 15.8 |



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

Lapp cable order no: 2170222, 2170822, 2170322.

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

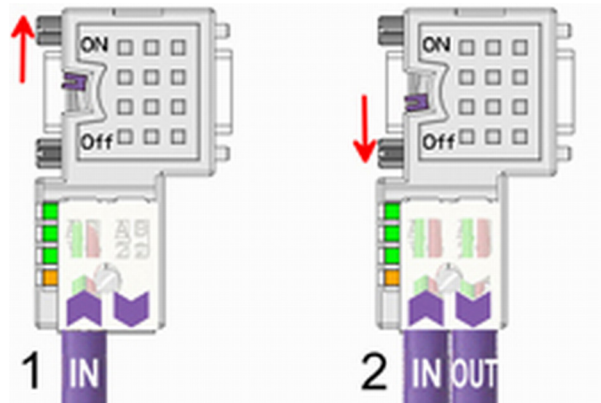


Dimensions in mm

Termination with "EasyConn"

The "EasyConn" bus connector is provided with a switch that is used to activate a terminating resistor.

Wiring



- [1] 1./last bus participant
[2] further participants

**CAUTION**

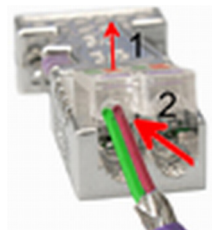
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

The tightening torque of the screws to fix the connector to a device must not exceed 0.02Nm!



A complete description of installation and deployment of the terminating resistors is delivered with the connector.

Assembly



1. Loosen the screw.
2. Lift contact-cover.
3. Insert both wires into the ducts provided (watch for the correct line colour as below!)
4. Please take care not to cause a short circuit between screen and data lines!



5. Close the contact cover.
6. Tighten screw (max. tightening torque 0.08Nm).



The green line must be connected to A, the red line to B!

8.7 Commissioning and Start-up behavior

Start-up on delivery

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

Online with bus parameter without slave project

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

Slave configuration

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

CPU state controls DP master

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

Master behavior at CPU STOP

- The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.
- DP slaves with fail safe mode were provided with output telegram length "0".
- DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.
- The input data of the DP slaves were further cyclically transferred to the input area of the CPU.

Master behavior at CPU RUN

- The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is on.
- Every connected DP slave is cyclically attended with an output telegram containing recent output data.
- The input data of the DP slaves were cyclically transferred to the input area of the CPU.

9 WinPLC7

9.1 System conception

General

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



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

Alternatives

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

System requirements

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

Source

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

There are the following sources to get WinPLC7:

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

9.2 Installation

Precondition

The project engineering of a SPEED7 CPU from Yaskawa 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.

Installation

Activation of the "Profi" version

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

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

Installation of WinPCAP for station search via Ethernet

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

9.3 Example project engineering

9.3.1 Job definition

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

Here it should apply:

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

Precondition

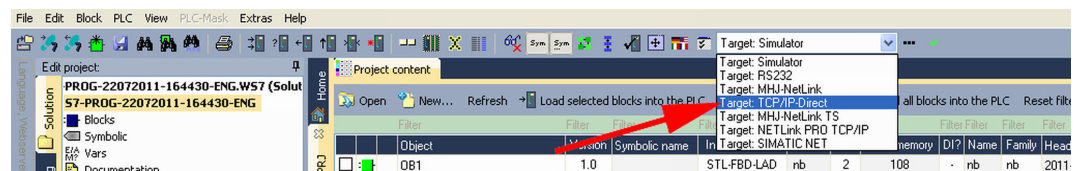
- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.
- One SPEED7 CPU and one digital output module are installed and cabled.
- The Ethernet PG/OP channel of the CPU is connected to your Ethernet network. Your CPU may be connected to your PC with an Ethernet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

9.3.2 Project engineering

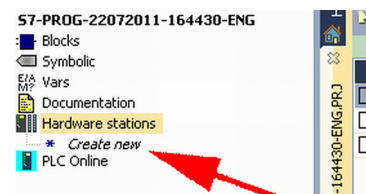
1. Start WinPLC7 ("Profi" version)
2. Create and open a new project with [Create a new solution].

Hardware configuration

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

















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




3. Enter a station name. Please consider that the name does not contain any spaces.
4. After the load animation choose in the register Select PLC-System the system "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 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.

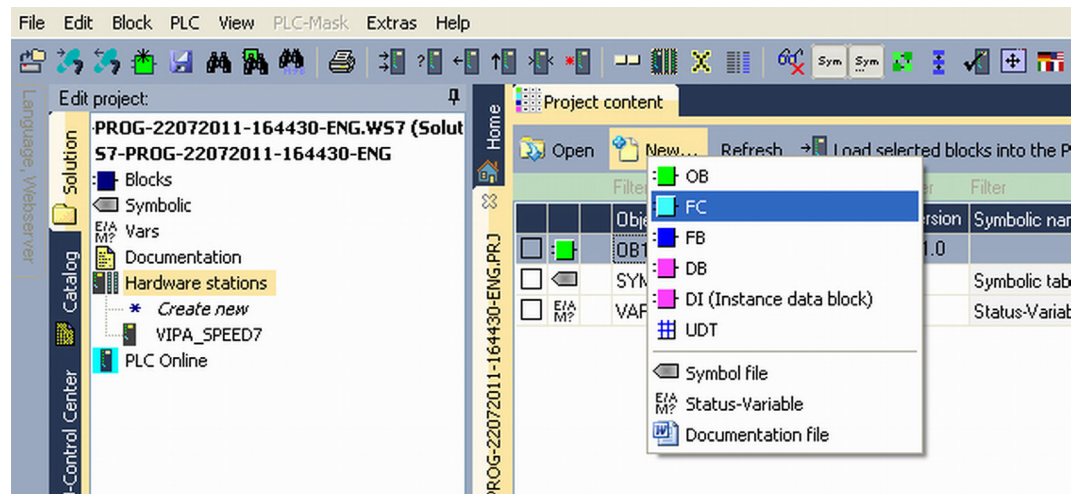


Usually the online transfer of the hardware configuration happens within the hardware configurator. With 'File → Save active station in the WinPL7 sub project' there is also the possibility to store the hardware configuration as a system file in WinPLC7 to transfer it from WinPLC7 to the CPU.

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".

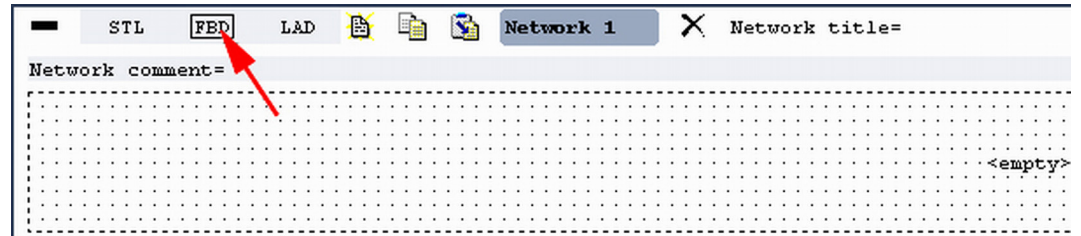
1. Select the 'in →' row at the 'parameter table' and enter at the field 'Name' "value1". Press the [Return] key.
 - ➔ The cursor jumps to the column with the data type.
2. The data type may either directly be entered or be selected from a list of available data types by pressing the [Return] key. Set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

| Address | Declaration | Name | Type | Initial value | Comment |
|---------|-------------|--------|------|---------------|------------------|
| 0.0 | in --> | value1 | INT | | 1. compare value |
| 2.0 | in --> | value2 | INT | | 2. compare value |
| | out <-- | | | | |
| | in_out <-> | | | | |

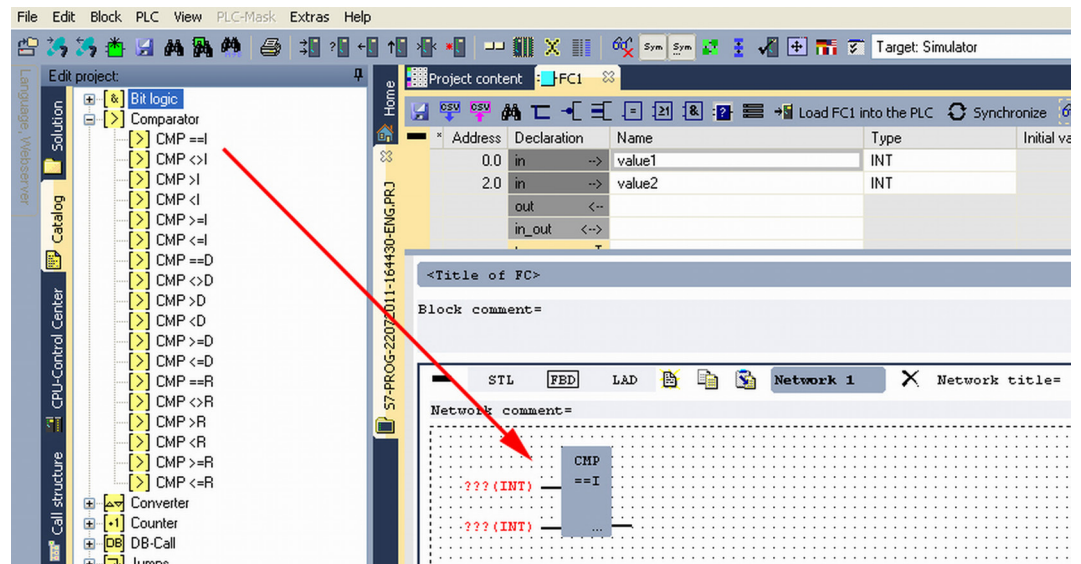
Enter the program

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

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




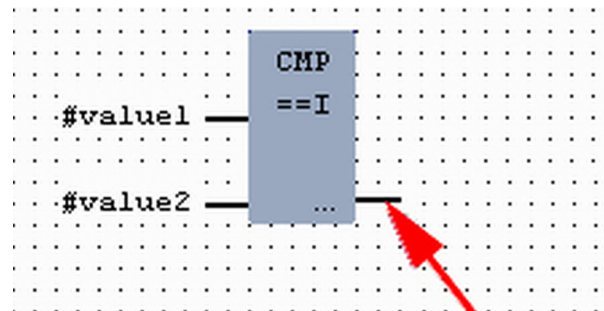
2. Click to the input field designated as "<empty>". The available operations may be added to your project by drag&drop from the *hardware catalog* or by double click at them in the *hardware catalog*.
3. Open in the *catalog* the category "Comparator" and add the operation 'CMP==I' to your network.





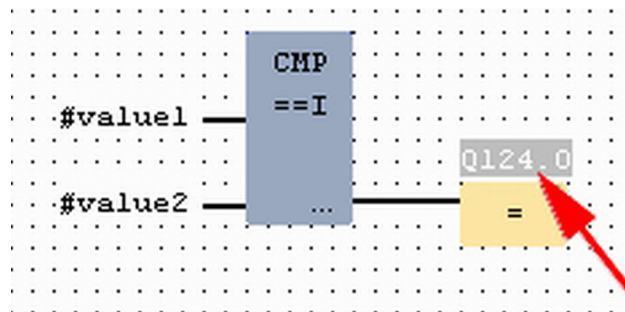
4. Click to the input left above and insert *value1*. Since these are block parameters a selection list of block parameters may be viewed by entering "#".
5. Type in "#" and press the [Return] key.
6. Choose the corresponding parameter of the list and confirm it with the [Return] key.
7. Proceed in the same way with the parameter *value2*.

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

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







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

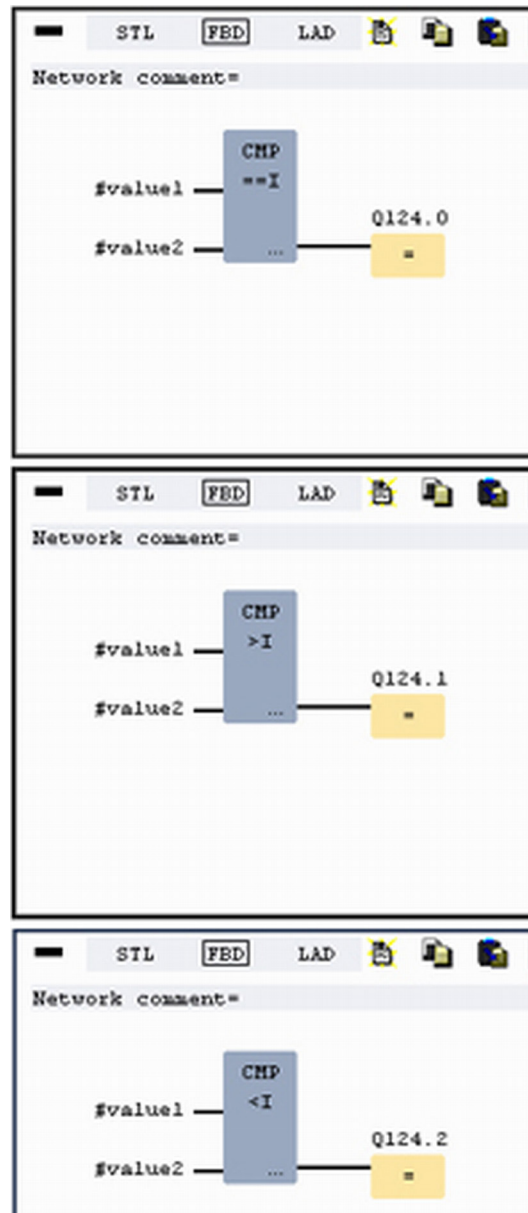


➔ Network1 is finished, now.

Adding a new network

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

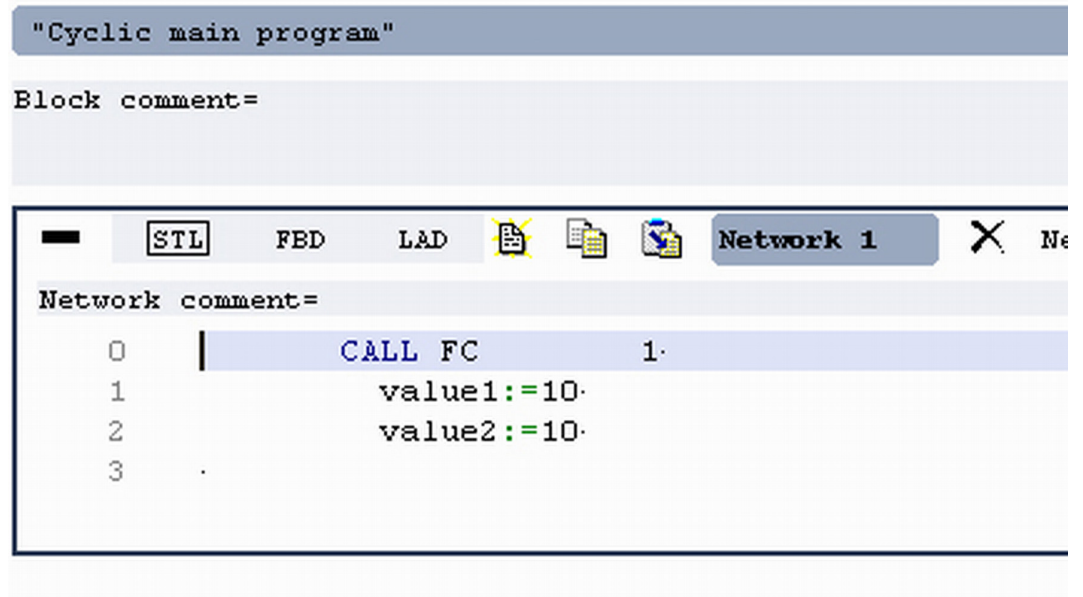
1.  Move your mouse at an arbitrary position on the editor window and press the right mouse key.
2.  Select at '*context menu* → *Insert new network*'.
 - ➔ A dialog field is opened to enter the position and number of the networks.
3.  Proceed as described for "Network 1".
4.  Save the FC 1 with '*File* → *Save content of focused window*' respectively press [Strg]+[S].
 - ➔ After you have programmed the still missing networks, the FC 1 has the following structure:



Creating the block OB 1

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

1. Go to OB 1, which was automatically created with starting the project.
2. Go to 'Project content' or to 'Solution' and open the OB 1 by a double click.
3. Change to the STL view.
4. Type in "Call FC 1" and press the [Return] key.
 - ➔ The FC parameters are automatically displayed and the following parameters are assigned:



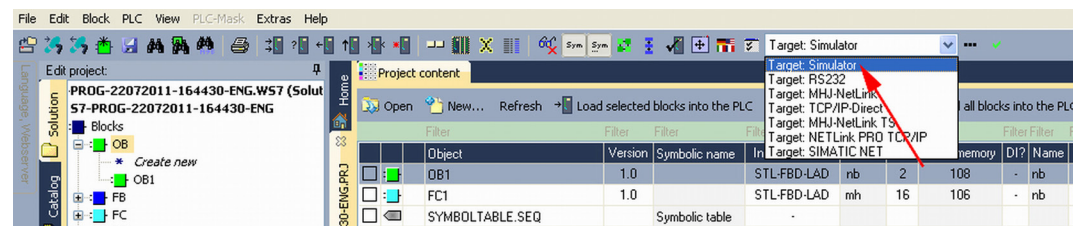
5. Save the OB 1 with respectively press [Strg]+[S].

9.3.3 Test the PLC program in the *Simulator*

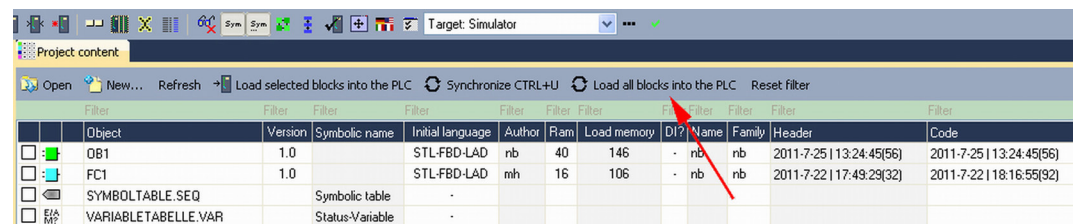
Proceeding

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

1. Here select 'Target: Simulator'.



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



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

➔ The displayed state changes from STOP to RUN.

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

➔ The various areas are displayed.

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

6. Open the OB 1.

7. Change the value of one variable, save the OB 1 and transfer it to the simulator.

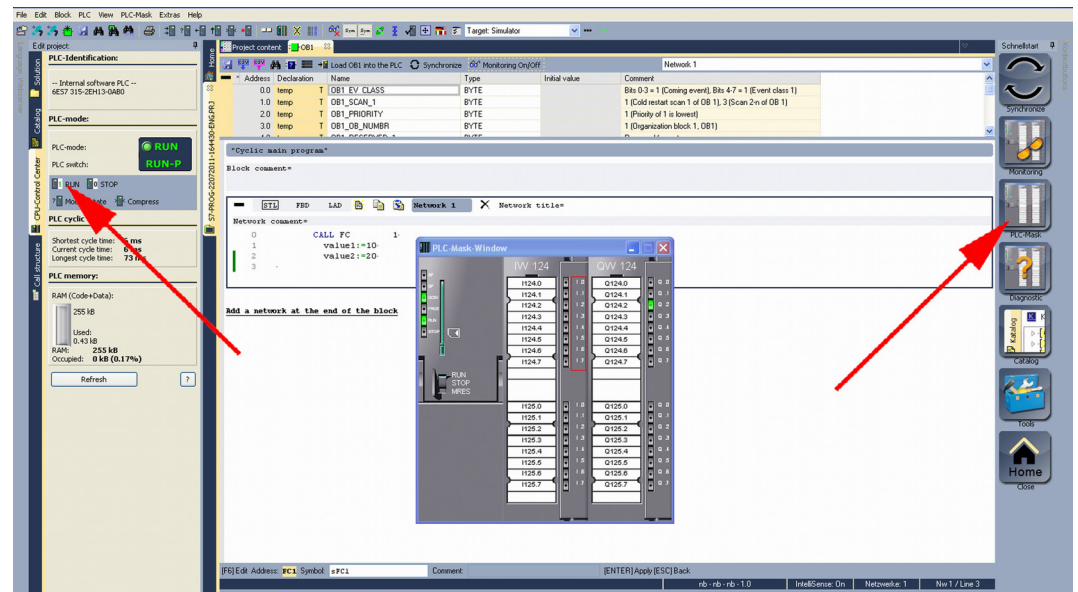
➔ According to your settings the process image changes immediately. The status of your blocks may be displayed with 'Block → Monitoring On/Off'.

Example project engineering > Test the PLC program in the *Simulator*

Visualization via PLC mask

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

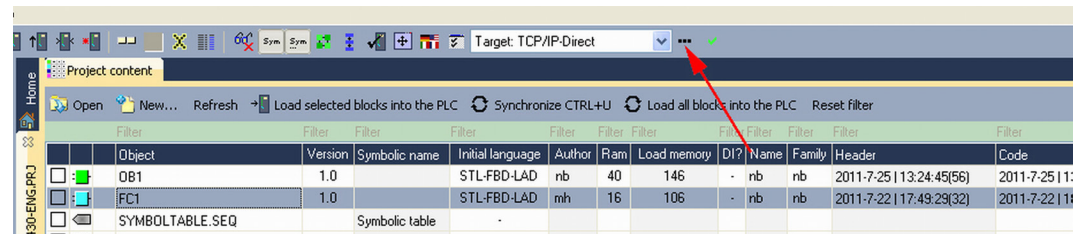
1. Open the *PLC mask* with 'view → *PLC mask*'.
 - ➔ A CPU is graphically displayed.
2. Double-click to the output module, open its properties dialog and enter the Module address 124.
3. Switch the operating mode switch to RUN by means of the mouse.
 - ➔ Your program is executed and displayed in the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



9.3.4 Transfer PLC program to CPU and its execution

Proceeding

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



4. ➤ Click at [Determining accessible nodes].
 - ➔ After a waiting time every accessible station is listed.
5. ➤ Choose your CPU, which was provided with TCP/IP address parameters during the hardware configuration and click to [Confirm].
6. ➤ Close the dialog 'Ethernet properties' with [OK].
7. ➤ Transfer your project to your CPU with 'PLC → Send all blocks'.
8. ➤ Switch your CPU to RUN state.
9. ➤ Open the OB 1 by double click.
10. ➤ Change the value of one variable, save the OB 1 and transfer it to the CPU.
 - ➔ According to your settings the process image changes immediately. The status of your blocks may be displayed with 'Block → Monitoring On/Off'.

10 Configuration with TIA Portal

10.1 TIA Portal - Work environment

10.1.1 General

General

In this chapter the project engineering of the System 300S CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a System 300S 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 PLCs may be configured and linked. For diagnostics online tools are available.

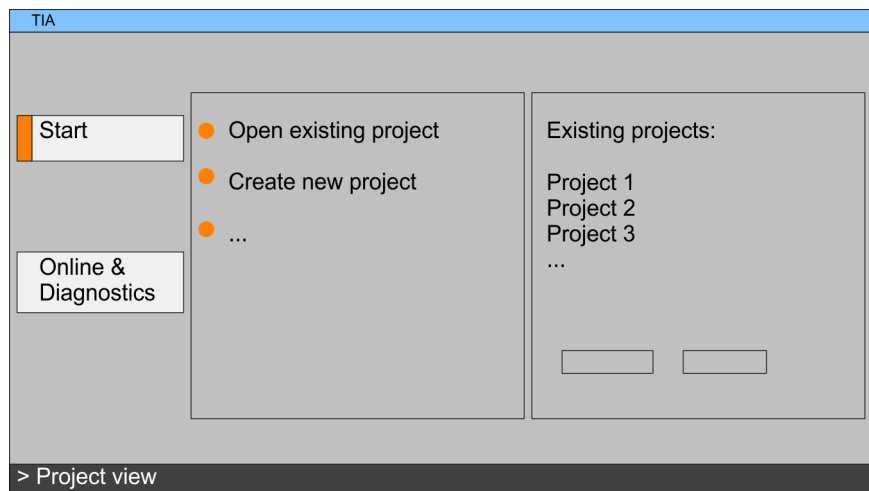


Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.

Starting the TIA Portal

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

Then the TIA Portal opens with the last settings used.



Exiting the TIA Portal

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

10.1.2 Work environment of the TIA Portal

Basically, the TIA Portal has the following 2 views. With the button on the left below you can switch between these views:

Portal view

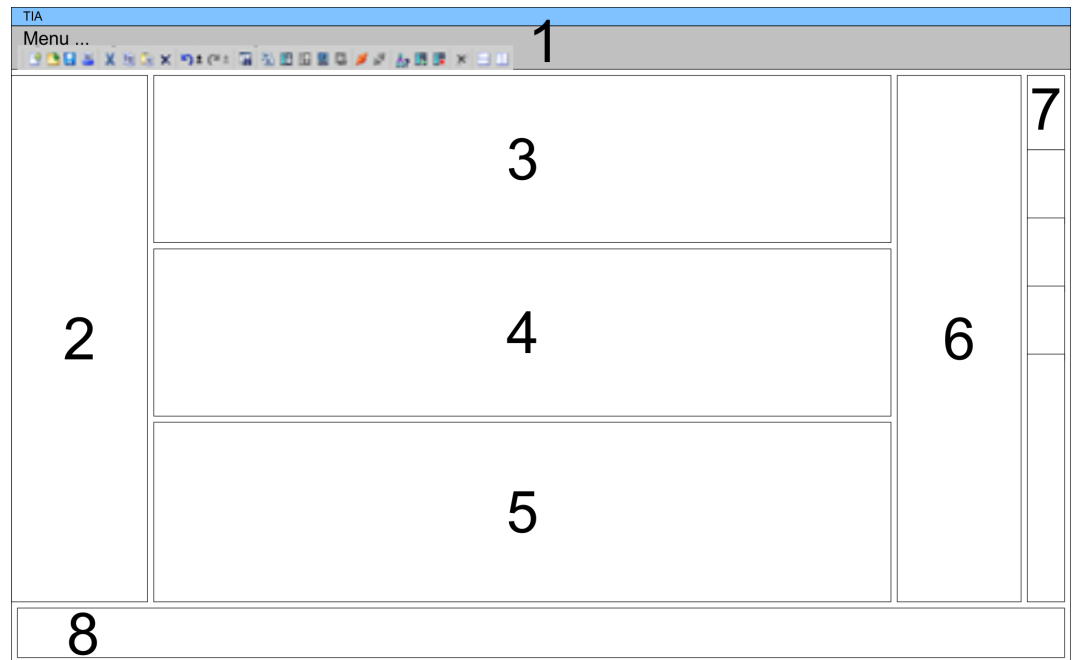
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

Project view

The *'Project view'* is a "structured" view to all constituent parts of your project.

Areas of the Project view

The Project view is divided into the following areas:



- 1 Menu bar with toolbars
- 2 Project tree with Details view
- 3 Project area
- 4 Device overview of the project respectively area for block programming
- 5 Properties dialog of a device (parameter) respectively information area
- 6 Hardware catalog and tools
- 7 "Task-Cards" to select hardware catalog, tasks and libraries
- 8 Jump to Portal or Project view

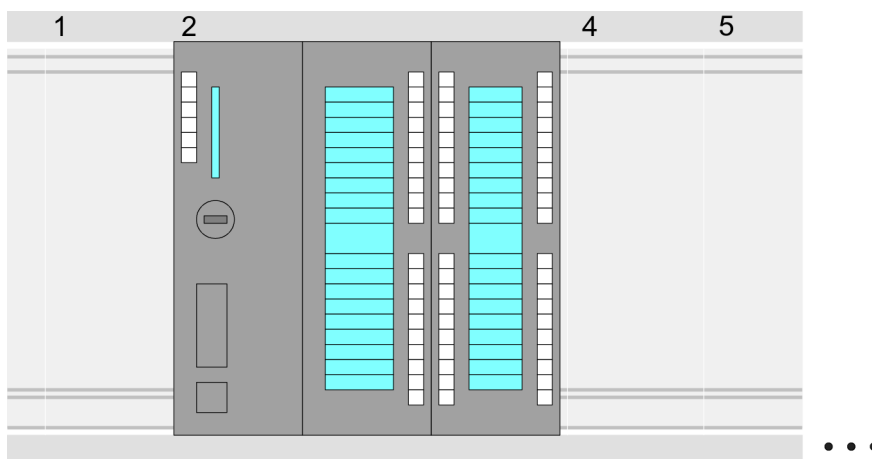
10.2 TIA Portal - Hardware configuration - CPU

Configuration Siemens CPU

With the Siemens TIA Portal, the CPU 314-6CG23 from Yaskawa is to be configured as CPU 314C-2DP (6ES7 314-6CG03-0AB0 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 314C-2DP (6ES7 314-6CG03-0AB0 V2.6)
→ The CPU is inserted with a profile rail.

Project area:



Device overview:

| Module | ... | Slot | ... | Type | ... |
|------------------|-----|------|-----|---------------|-----|
| PLC ... | | 2 | | CPU 314C-2DP | |
| MPI interface... | | 2 0 | | MPI interface | |
| DP interface... | | 2 X2 | | DP interface | |
| DI24/DO16... | | 2 2 | | DI24/DO16 | |
| AI5/AO2 | | 2 3 | | AI5/AO2 | |
| Counter... | | 2 4 | | Counter | |
| Positioning... | | 2 5 | | Positioning | |
| ... | | | | ... | |

Setting standard CPU parameters

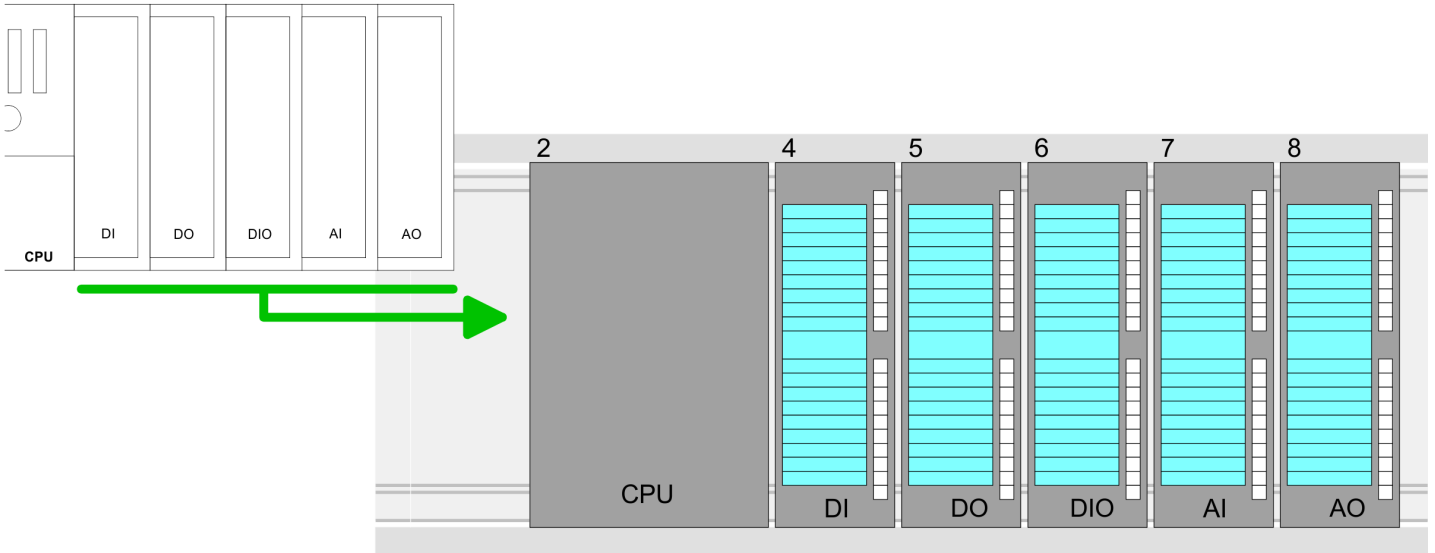
Since the CPU 314-6CG23 from Yaskawa is configured as Siemens CPU 314C-2DP, so the parametrization takes place via the Siemens CPU. For parametrization click in the *Project area* respectively in the *Device overview* at the CPU part. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings.

➔ [‘CPU parametrization’...page 58](#)

10.3 TIA Portal - Hardware configuration - I/O modules

Hardware configuration of the modules

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



Device overview

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

Parametrization

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





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

Overview

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


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

Assembly and commissioning

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








"Initialization" via Online functions

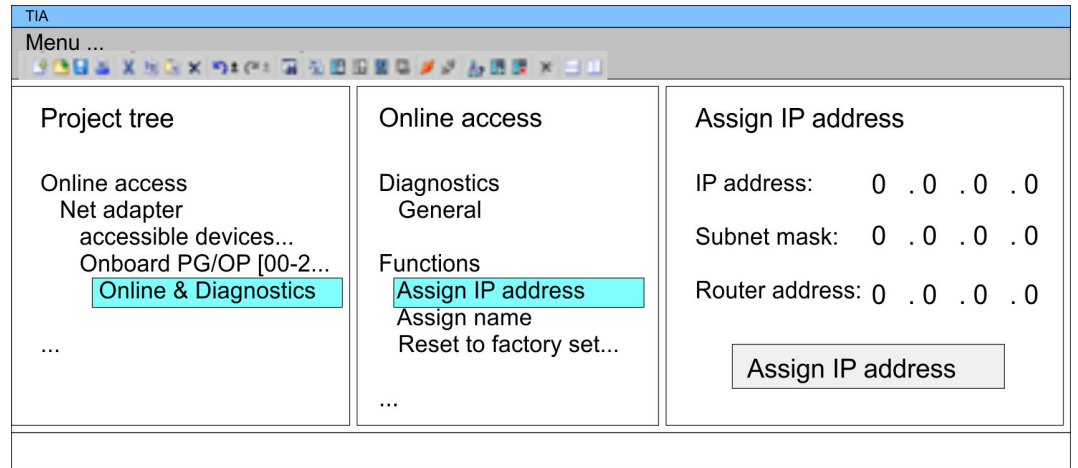
The initialization via the Online functions takes place with the following proceeding:

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

Assign IP address parameters

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

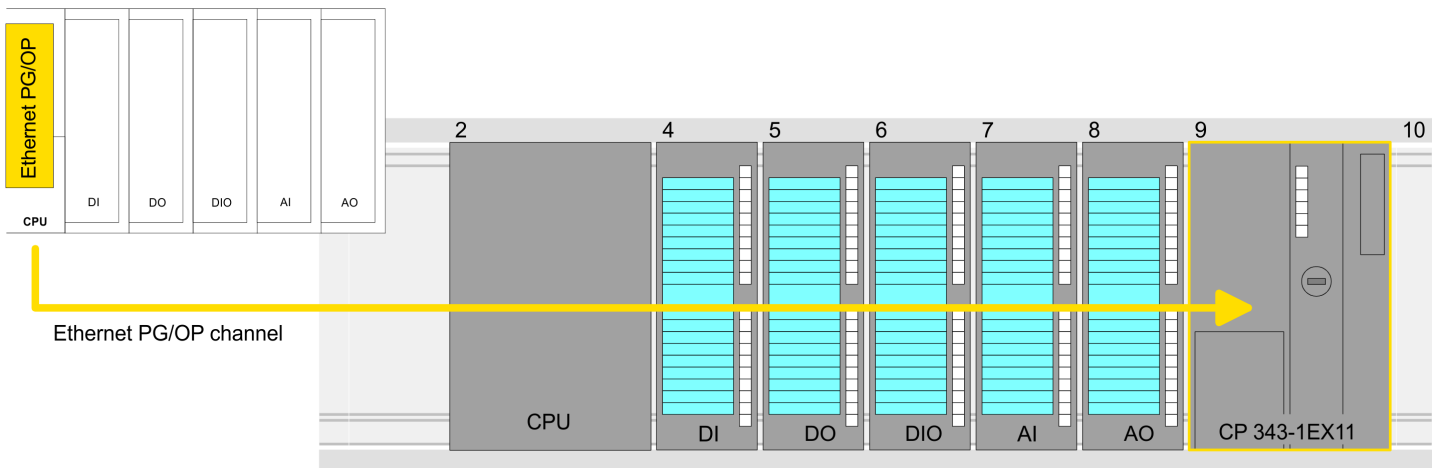
1.  Start the Siemens TIA Portal.
2.  Switch to the 'Project view'.
3.  Click in the 'Project tree' at 'Online access' and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.
4.  To get the stations and their MAC address, use the 'Accessible device'. The MAC address can be found at the 1. label beneath the front flap of the CPU.
5.  Choose from the list the module with the known MAC address (Onboard PG/OP [MAC address]) and open with "Online & Diagnostics" the diagnostics dialog in the Project area.
6.  Navigate to *Functions > Assign IP address*. Type in the IP configuration like IP address, subnet mask and gateway.
7.  Confirm with [Assign IP configuration].
 - ➔ Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.



i 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 314C-2DP (6ES7 314-6CG03-0AB0 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 ... | |
| ... | | ... | | ... | |
| DI... | | 4 | | DI... | |
| DO... | | 5 | | DO... | |

| | | | |
|--|---|----------|--|
| DIO... | 6 | DIO... | |
| AI... | 7 | AI... | |
| AO... | 8 | AO... | |
|  CP 343-1 | 9 | CP 343-1 | |
| | | | |

10.5 TIA Portal - Include Controls Library

Overview

- The product specific blocks can be found at www.yaskawa.eu.com in the 'Download Center' under 'Controls Library' as library download file.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use product specific blocks you have to import them into your project.

Execute the following steps:

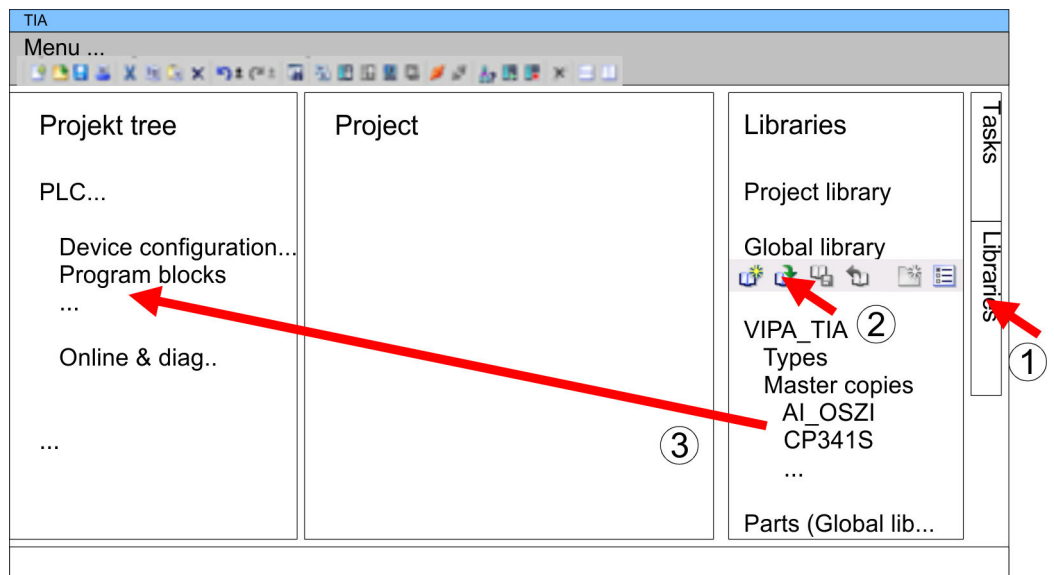
- Load an unzip the file ...TIA_Vxx.zip (note TIA Portal version)
- Open library and transfer blocks into the project

Unzip ...TIA_Vxx.zip

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

Open library and transfer blocks into the project

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



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

10.6 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 Yaskawa 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. ➔ ['CMD - auto commands'...page 84](#)

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". ➔ ['Diagnostic entries'...page 85](#)

11 Appendix

Content

| | | |
|----------|--|------------|
| A | System specific event IDs. | 209 |
| B | Integrated blocks. | 248 |
| C | SSL partial list. | 251 |

A System specific event IDs

Event IDs [↔ 'Diagnostic entries'...page 85](#)

| 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 |
| 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 |

| Event ID | Description |
|----------|---|
| 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 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 |

| Event ID | Description |
|----------|---|
| | 13: Faulty block type in the telegram |
| | 14: Inactive function |
| | 15: Wrong size in the telegram |
| | 20: Error in writing on MMC |
| | 90: Faulty buffer size |
| | 98: Unknown error |
| | 99: Internal error |
| 0xE0CD | Error at DP-V1 job management |
| | ZINFO1: Not user relevant |
| | ZINFO2: Not user relevant |
| | ZINFO3: Not user relevant |
| | DatID: Not user relevant |
| 0xE0CE | Error: Time out when sending i-slave diagnostics |
| 0xE100 | Memory card access error |
| 0xE101 | Memory card error file system |
| 0xE102 | Memory card error FAT |
| 0xE104 | Memory card error at saving |
| | ZINFO3: Not user relevant |
| 0xE200 | Memory card writing finished (Copy Ram2Rom) |
| | OB: Not user relevant |
| | PK: Not user relevant |
| 0xE210 | Memory card reading finished (reload after memory reset) |
| | OB: Not user relevant |
| | PK: Not user relevant |
| | 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 |

| Event ID | Description |
|----------|---|
| | ZINFO1: Not user relevant |
| | ZINFO2: Block type |
| | 56: OB |
| | 65: DB |
| | 66: SDB |
| | 67: FC |
| | 68: SFC |
| | 69: FB |
| | 70: SFB |
| | 97: VDB |
| | 98: VSDB |
| | 99: VFC |
| | 100: VSFC |
| | 101: VFB |
| | 102: VSFB |
| | 111: VOB |
| | ZINFO3: Block number |
| 0xE300 | Internal flash writing completed (copy Ram2Rom) |
| 0xE310 | Internal flash reading completed (recharging after battery failure) |
| 0xE400 | FSC card was plugged |
| | OB: FSC activated from this slot (PK) |
| | OB: The inserted FSC is the activated FSC |
| | OB: The inserted FSC is compatible with the CPU |
| | PK: FSC source |
| | 0: CPU |
| | 1: Card |
| | ZINFO1: FSC(CRC) |
| | 1146: 955-C000070 |
| | 1736: 955-C0NE040 |
| | 2568: FSC-C0ME040 |
| | 3450: 955-C000M30 |
| | 3903: 955-C000S30 |
| | 4361: FSC-C000M30 |
| | 4940: FSC-C000S30 |
| | 5755: 955-C0ME040 |
| | 6843: FSC-C0NE040 |
| | 8561: FSC-C000S20 |
| | 9012: FSC-C000M20 |
| | 13895: 955-C000060 |
| | 15618: 955-C000S20 |
| | 16199: 955-C000M20 |
| | 17675: FSC-C000S00 |
| | 18254: FSC-C000M00 |
| | 20046: FSC-C000040 |
| | 21053: 955-C000040 |
| | 22904: 955-C000S00 |
| | 23357: 955-C000M00 |
| | 24576: 955-C000050 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 22904: 955-C000S00 |
| | 23357: 955-C000M00 |
| | 24576: 955-C000050 |
| | 35025: 955-C00MC10 |
| | 36351: FSC-C000S40 |
| | 36794: FSC-C000M40 |
| | 37260: 955-C000S40 |
| | 37833: 955-C000M40 |
| | 38050: FSC-C00MC10 |
| | 41460: 955-C000M50 |
| | 41526: 955-C0PE040 |
| | 42655: FSC-C00MC00 |
| | 47852: 955-C00MC00 |
| | 48709: FSC-C0PE040 |
| | 50574: 955-C000M70 |
| | 52366: 955-C000030 |
| | 53501: FSC-C000030 |
| | 58048: FSC-C000020 |
| | 63411: 955-C000M60 |
| | 65203: 955-C000020 |
| | ZINFO2: FSC serial number (high word) |
| | ZINFO3: FSC serial number (low word) |
| | DatID: FeatureSet Trialtime in minutes |
| 0xE402 | A configured functionality is not activated. The configuration is accepted, but the PLC can not go to RUN. ZINFO1: Required FSC: PROFIBUS ZINFO1: Required FSC: MOTION ZINFO2: Number of released axes ZINFO3: Number of configured axes |
| 0xE403 | FSC can not be activated in this CPU OB: FCS error code PK: FSC source 0: CPU 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 |

| Event ID | Description |
|----------|---|
| | 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) |
| 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 |

| Event ID | Description |
|----------|---|
| | 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 |
| 0xE406 | The inserted feature set is corrupt PK: FSC source 0: CPU 1: Card |
| 0xE410 | A CPU feature set was activated PK: FSC source 0: CPU 1: Card ZINFO1: FSC(CRC) 1146: 955-C000070 1736: 955-C0NE040 2568: FSC-C0ME040 3450: 955-C000M30 3903: 955-C000S30 4361: FSC-C000M30 4940: FSC-C000S30 |

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |

| Event ID | Description |
|----------|---|
| | 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 |
| 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 peripheral address for Ethernet PG/OP channel ZINFO1: Peripheral address ZINFO3: 0: peripheral address is input, 1: peripheral address is output |
| 0xE605 | Too many productive connections configured ZINFO1: Interface slot ZINFO2: Number of configured connections ZINFO3: Number of admissible connections |

| Event ID | Description |
|----------|--|
| 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 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 |

| Event ID | Description |
|----------|---|
| | ZINFO2: Master system ID DatID: Not user relevant |
| 0xE780 | Internal error - Please contact the hotline! |
| 0xE781 | Address range exceeds process image limit ZINFO1: Address ZINFO2: Length of the address range ZINFO3: Size of the process image DatID: Address range |
| 0xE801 | CMD - auto command: CMD_START recognized and executed |
| 0xE802 | CMD - auto command: CMD_End recognized and executed |
| 0xE803 | CMD - auto command: WAIT1SECOND recognized and executed |
| 0xE804 | CMD - auto command: WEBPAGE recognized and executed |
| 0xE805 | CMD - auto command: LOAD_PROJECT recognized and executed |
| 0xE806 | CMD - auto command: SAVE_PROJECT recognized and executed ZINFO3: Status 0: Error 1: OK 32768: Wrong password |
| 0xE807 | CMD - auto command: FACTORY_RESET recognized and executed |
| 0xE808 | Internal 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! |
| 0xE80E | CMD - auto command: SET_NETWORK recognized and executed |
| 0xE80F | Internal error - Please contact the hotline! ZINFO3: Status 0: OK 65153: File create error |

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

| Event ID | Description |
|----------|---|
| 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 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 |

| Event ID | Description |
|----------|--|
| | 15: De-energised |
| | 253: Process image release in STOP |
| | 254: Watchdog |
| | 255: Not set |
| | PK: Not user relevant |
| | ZINFO1: Slot |
| | ZINFO2: Status |
| | 0: OK |
| | 1: Error |
| | 2: Empty |
| | 3: Busy |
| | 4: Time out |
| | 5: Internal blocking |
| | 6: Too many telegrams |
| | 7: Not Connected |
| | 8: Unknown |
| | DatID: Not user relevant |
| 0xEA04 | SBUS: Multiple configuration of a periphery address ZINFO1: Periphery address ZINFO2: Slot ZINFO3: Data width |
| 0xEA05 | Internal error - Please contact the hotline! |
| 0xEA07 | Internal error - Please contact the hotline! |
| 0xEA08 | SBUS: Configured input data width not the same as the connected input data width ZINFO1: Configured input data width ZINFO2: Slot ZINFO3: Input data width of the connected component |
| 0xEA09 | SBUS: Configured output data width not the same as the connected output data width ZINFO1: Configured output data width ZINFO2: Slot ZINFO3: Output data width of the plugged component |
| 0xEA10 | SBUS: Input periphery address outside the periphery area ZINFO1: Periphery address 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! |

| Event ID | Description |
|----------|---|
| | ZINFO2: Slot of the master |
| 0xEA18 | SBUS: Error in mapping the master periphery ZINFO2: Slot of the master |
| 0xEA19 | Internal error - Please contact the hotline! PK: Not user relevant ZINFO2: HW slot ZINFO3: Interface type DatID: Not user relevant |
| 0xEA1A | SBUS: Error in access to SBUS FPGA address table PK: Not user relevant ZINFO2: HW slot ZINFO3: Table 0: Read 1: Writing DatID: Not user relevant |
| 0xEA20 | Error: RS485 interface is not pre-set to PROFIBUS DP master bus a PROFIBUS DP master is configured |
| 0xEA21 | Error: Configuration RS485 interface X2/X3: PROFIBUS DP master is configured but missing ZINFO2: Interface X is configured incorrectly |
| 0xEA22 | Error: Configuration RS485 interface X2: Value is outside the limits ZINFO2: Configuration for X2 |
| 0xEA23 | Error: Configuration RS485 interface X3: Value is outside the limits ZINFO2: Configuration for X3 |
| 0xEA24 | Error: Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used ZINFO2: Configuration for X2 ZINFO3: Configuration for X3 |
| 0xEA30 | Internal error - Please contact the hotline! ZINFO1: Status ZINFO2: Not user relevant ZINFO3: Not user relevant |
| 0xEA40 | Internal error - Please contact the hotline! OB: Slot of the CP PK: File number ZINFO1: Version of the CP ZINFO2: Not user relevant ZINFO3: Not user relevant DatID: Line |
| 0xEA41 | Internal error - Please contact the hotline! OB: Slot of the CP PK: File number ZINFO1: Version of the CP ZINFO2: Not user relevant ZINFO3: Not user relevant DatID: Line |
| 0xEA50 | PROFINET IO controller: Error in the configuration OB: Not user relevant PK: Not user relevant ZINFO1: Rack/slot of the controller |

| Event ID | Description |
|----------|--|
| | 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! |
| | 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 |

| Event ID | Description |
|----------|---|
| | 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.) 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 |

| Event ID | Description |
|----------|---|
| | DatID: StackError.DeviceRef |
| 0xEA67 | PROFINET IO controller: Error reading dataset OB: Rack/slot of the controller PK: Error type 0: Dataset error local 1: Dataset error stack 2: Dataset error station ZINFO1: Dataset number ZINFO2: Dataset handle (caller) ZINFO3: Internal error code from PN stack DatID: Device |
| 0xEA68 | PROFINET IO controller: Error writing dataset OB: Rack/slot of the controller PK: Error type 0: Dataset error local 1: Dataset error stack 2: Dataset error station ZINFO1: Dataset number ZINFO2: Dataset handle (caller) ZINFO3: Internal error code from PN stack DatID: Device |
| 0xEA69 | Internal error - Please contact the hotline! ZINFO1: Minimum version for the FPGA ZINFO2: Loaded FPGA version |
| 0xEA6A | PROFINET IO controller: Service error in the communication stack OB: Service ID PK: Rack/slot ZINFO1: ServiceError.Code ZINFO2: ServiceError.Detail ZINFO3 - Position 0: ServiceError.AdditionalDetail ZINFO3 - Position 8: ServiceError.AreaCode |
| 0xEA6B | PROFINET IO controller: Incorrect Vendor-ID OB: Operating mode 0: Configuration in operating condition RUN 1: STOP (update) 2: STOP (memory reset) 3: STOP (auto initialization) 4: STOP (internal) 5: STARTUP (cold start) 6: STARTUP (restart/warm start) 7: STARTUP (hot restart) 9: RUN 10: HALT 11: COUPLING 12: UPDATING 13: DEFECTIVE 14: Error search mode 15: De-energised |

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

| Event ID | Description |
|----------|---|
| | 254: Watchdog |
| | 255: Not set |
| | PK: Rack/slot |
| | ZINFO1: Device ID |
| | ZINFO2: Not user relevant |
| | ZINFO3: Not user relevant |
| | DatID: Not user relevant |
| 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | 129: PNIO |
| | 207: RTA error |
| | 218: AlarmAck |
| | 219: IODConnectRes |
| | 220: IODReleaseRes |
| | 221: IOD/IOXControlRes |
| | 222: IODReadRes |
| | 223: IODWriteRes |
| | ZINFO2: ErrorDecode |
| | 128: PNIO: 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] |

| Event ID | Description |
|----------|---|
| | 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 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 |

| Event ID | Description |
|----------|---|
| 0xEA98 | Error in file reading via SBUS PK: Not user relevant ZINFO3: Slot DatID: Not user relevant |
| 0xEA99 | Parameter assignment job could not be executed PK: Not user relevant ZINFO1: File version on MMC/SD (if not 0) ZINFO2: File version of the SBUS module (if not 0) ZINFO3: Slot DatID: Not user relevant |
| 0xEAA0 | Internal error - Please contact the hotline! OB: Current operating mode 0: Configuration in operating condition RUN 1: STOP (update) 2: STOP (memory reset) 3: STOP (auto initialization) 4: STOP (internal) 5: STARTUP (cold start) 6: STARTUP (restart/warm start) 7: STARTUP (hot restart) 9: RUN 10: HALT 11: COUPLING 12: UPDATING 13: DEFECTIVE 14: Error search mode 15: De-energised 253: Process image release in STOP 254: Watchdog 255: Not set ZINFO1: Diagnostic address of the master ZINFO2: Not user relevant ZINFO3: Number of errors which occurred |
| 0xEAB0 | Invalid link mode OB: Current operating mode 0: Configuration in operating condition RUN 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 |

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |

| Event ID | Description |
|------------------------|---|
| 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 |
| | 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 | |

| Event ID | Description |
|----------|---|
| | 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 |
| 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) |

| Event ID | Description |
|----------|---|
| | ZINFO3: Not user relevant |
| | DatID: Not user relevant |
| 0xEC10 | EtherCAT: Recurrence bus with all slaves |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the station |
| | ZINFO3: Number of stations, which are not in the same state as the master |
| | DatID: Station not available |
| | DatID: Station available |
| | DatID: Input address |
| | DatID: Output address |
| 0xEC11 | EtherCAT: 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 |
| | DatID: Input address |
| | DatID: Output address |
| 0xEC12 | EtherCAT: Recurrence slave |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| | 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 |
| | DatID: Output address |
| 0xED12 | EtherCAT: Breakdown slave |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the station |
| | ZINFO3: AIStatusCode |
| | 0: No error |
| | 1: Unspecified error |
| | 17: Invalid requested status change |
| | 18: Unknown requested status |
| | 19: Bootstrap not supported |
| | 20: No valid firmware |
| | 22: Invalid mailbox configuration |
| | 22: Invalid mailbox configuration |

| Event ID | Description |
|----------|--|
| | 23: Invalid sync manager configuration |
| | 24: No valid inputs available |
| | 25: No valid outputs available |
| | 26: Synchronisation error |
| | 27: Sync manager watchdog |
| | 28: Invalid sync manager types |
| | 29: Invalid output configuration |
| | 30: Invalid input configuration |
| | 31: Invalid watchdog configuration |
| | 32: Slave station needs cold start |
| | 33: Slave station needs to be in INIT state |
| | 34: Slave station needs to be in PreOp state |
| | 35: Slave station needs to be in SafeOp state |
| | 45: Invalid output FMMU configuration |
| | 46: Invalid input FMMU configuration |
| | 48: Invalid DC Sync configuration |
| | 49: Invalid DC Latch configuration |
| | 50: PLL error |
| | 51: Invalid DC IO error |
| | 52: Invalid DC time out error |
| | 66: Error in acyclic data exchange Ethernet Over EtherCAT |
| | 67: Error in acyclic data exchange CAN Over EtherCAT |
| | 68: Error in acyclic data exchange Fileaccess Over EtherCAT |
| | 69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT |
| | 79: Error in acyclic data exchange Vendorspecific Over EtherCAT |
| | DatID: 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 |

| Event ID | Description |
|----------|--|
| | DatID: Input address |
| | DatID: Output address |
| 0xED21 | EtherCAT: Incorrect bus status change |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the master |
| | ZINFO3: Error code |
| | 4: Cancel (master state change) |
| | 8: Busy |
| | 11: Invalid parameters |
| | 14: Invalid status |
| | 16: Time out |
| | DatID: Station available |
| | DatID: Station not available |
| | DatID: Output address |
| | DatID: Input address |
| 0xED22 | EtherCAT: Slave status change that does not generate an OB86 |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO1 - Position 8: Previous status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Diagnostic address of the station |
| | ZINFO3: AIStatusCode |
| | 0: No error |
| | 1: Unspecified error |
| | 17: Invalid requested status change |

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

| Event ID | Description |
|----------|---|
| | 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 |
| | 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 |

| Event ID | Description |
|----------|---|
| | 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) |
| | 7: STARTUP (hot restart) |
| | 9: RUN |
| | 10: HALT |
| | 11: COUPLING |
| | 12: UPDATING |
| | 13: DEFECTIVE |
| | 14: Error search mode |
| | 15: De-energised |
| | 253: Process image release in STOP |
| | 254: Watchdog |
| | 255: Not set |
| | ZINFO1 - Position 0: New status |
| | 0: Undefined/Unkown |
| | 1: Init |
| | 2: PreOp |
| | 3: Bootstrap |
| | 4: SafeOp |
| | 8: Op |
| | ZINFO2: Slave address |
| | ZINFO3: AIStatusCode |
| | 0: No error |
| | 1: Unspecified error |
| | 17: Invalid requested status change |
| | 18: Unknown requested status |
| | 19: Bootstrap not supported |
| | 20: No valid firmware |
| | 22: Invalid mailbox configuration |
| | 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 |
| 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 |

| Event ID | Description |
|----------|--|
| | 0: Configuration in operating condition RUN |
| | 1: STOP (update) |
| | 2: STOP (memory reset) |
| | 3: STOP (auto initialization) |
| | 4: STOP (internal) |
| | 5: STARTUP (cold start) |
| | 6: STARTUP (restart/warm start) |
| | 7: STARTUP (hot restart) |
| | 9: RUN |
| | 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 |
| | 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 |

| Event ID | Description |
|----------|--|
| 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) |
| | 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 |

| Event ID | Description |
|----------|-------------------------|
| | 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".

| 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 55 | DP: STATUS ALARM | Status interrupt |
| OB 56 | DP: UPDATE ALARM | Update interrupt |
| OB 57 | DP: MANUFACTURE ALARM | Vendor specific interrupt |
| OB 80 | CYCL_FLT | Time error |
| OB 81 | PS_FLT | Power supply error |
| OB 82 | I/O_FLT1 | Diagnostics interrupt |
| OB 83 | I/O_FLT2 | Insert / remove module |
| OB 85 | OBNL_FLT | Priority class error |
| OB 86 | RACK_FLT | Slave failure / restart |
| OB 100 | COMPLETE RESTART | Start-up |
| OB 121 | PROG_ERR | Programming error |
| OB 122 | MOD_ERR | Periphery access error |

| SFB | Name | Description |
|--------|----------|---|
| SFB 0 | CTU | Up-counter |
| SFB 1 | CTD | Down-counter |
| SFB 2 | CTUD | Up-down counter |
| SFB 3 | TP | Create pulse |
| SFB 4 | TON | On-delay |
| SFB 5 | TOF | Create turn-off delay |
| SFB 7 | TIMEMESS | Time measurement |
| SFB 12 | BSEND | Sending data in blocks |
| SFB 13 | BRCV | Receiving data in blocks: |
| SFB 14 | GET | Remote CPU read |
| SFB 15 | PUT | Remote CPU write |
| SFB 31 | NOTIFY8P | Messages without acknowledge display (8x) |
| SFB 32 | DRUM | Realize a step-by-step switch |
| SFB 33 | ALARM | Messages with acknowledgement display |
| SFB 34 | ALARM_8 | Messages without associated values (8x) |
| SFB 35 | ALARM_8P | Messages with associated values (8x) |
| SFB 36 | NOTIFY8 | Messages without acknowledgement display |
| SFB 47 | COUNT | Control counter |
| SFB 48 | FREQUENC | Frequency measurement |
| SFB 49 | PULSE | Pulse width modulation |

| SFB | Name | Description |
|---------|----------|--|
| 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 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 | Name | Description |
|---------|----------|---|
| 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 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".

| 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 |