VIPA System 300S

SPEED7 - SM | Manual

HB140E_SM | Rev. 09/45

November 2009

VIPA art of automation

VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH
Ohmstraße 4, D-91074 Herzogenaurach,Germany
Tel.: +49 (91 32) 744 -0
Fax.: +49 9132 744 1864
EMail: info@vipa.de
http://www.vipa.de

Note

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information. This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions of the following directives:

- 2004/108/EC Electromagnetic Compatibility Directive
- 2006/95/EC Low Voltage Directive

Conformity is indicated by the CE marking affixed to the product.

Conformity Information

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

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax:+49 9132 744 1204
EMail: documentation@vipa.de

Technical support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telephone: +49 9132 744 1150/1180 (Hotline)
EMail: support@vipa.de

Contents

About this manual

This manual describes the System 300S SPEED7 **s**ignal **m**odules (SM) for SPEED-Bus. Here you may find besides of a product overview a detailed description of the single modules. You'll receive information about the connection and the deployment of the System 300S SM modules.

Overview

Chapter 1: Principles

This chapter introduces the System 300 from VIPA as central res. decentral automation system.

Chapter 2: Hardware description

The SPEED7 CPU is available in different variants. This chapter describes the hardware of the different versions.

Chapter 3: Digital I/O modules FAST - SPEED-Bus

This chapter describes the fast digital I/O modules for the SPEED-Bus.

Chapter 4: Analog I/O modules FAST - SPEED-Bus

This chapter describes the fast analog I/O modules for the SPEED-Bus.

Objective and contents

This manual describes the **s**ignal **m**odules (SM) for SPEED-Bus of the System 300S from VIPA. It contains a description of the construction, project engineering and usage.

Target audience

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

Structure of the manual

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

Guide to the document

The following guides are available in the manual:

- an overall table of contents at the beginning of the manual
- an overview of the topics for every chapter
- an index at the end of the manual.

Availability

The manual is available in:

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

Icons Headings

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

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

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

Note!
Supplementary information and useful tips.

Safety information

Applications conforming with specifications

The SPEED7 signal modules are constructed and produced for:

- for the deployment with VIPA SPEED-Bus
- communication and process control
- general control and automation applications
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

Danger!

This device is not certified for applications in

- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

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

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

Disposal

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

Chapter 1 Basics

Overview

This Basics contain hints for the usage and information about a SPEED7 system from VIPA.

General information about the System 300S like dimensions and environment conditions will also be found.

Content

Safety Information for Users

Handling of electrostatic sensitive modules

VIPA modules make use of highly integrated components in MOS-Technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges.

The following symbol is attached to modules that can be destroyed by electrostatic discharges.

The Symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment.

It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable.

Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load.

Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatic sensitive modules.

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

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

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

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

Attention!

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

General description of the System 300

The System 300

The System 300 is a modular automation system for middle and high performance needs, which may be used, either centralized or decentralized. The single modules are directly clipped to the profile rail and are connected together with the help of bus clips at the backside.

The CPUs of the System 300 are instruction set compatible to S7-300 from Siemens.

System 300V
System 300S

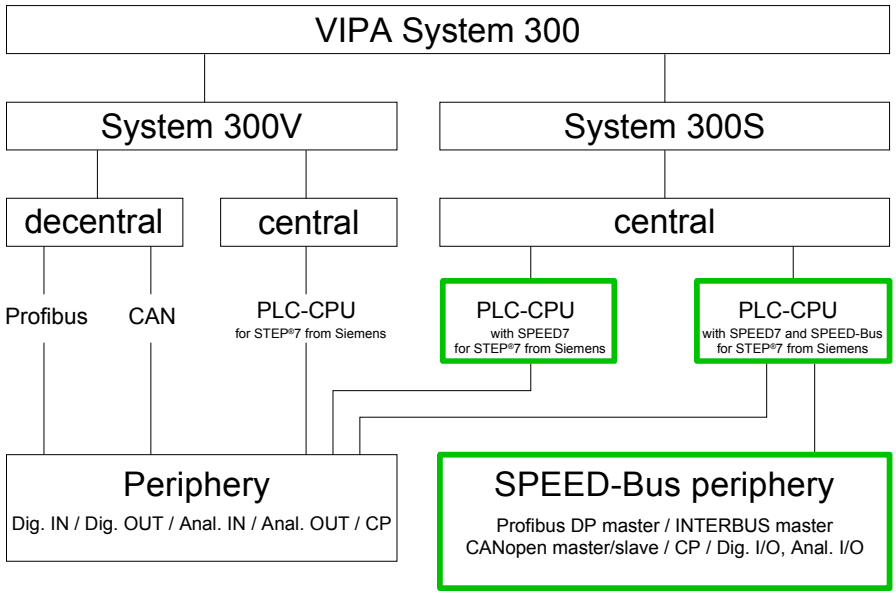
VIPA differentiates between System 300V and System 300S.

- System 300V

  The System 300V allows you to resolve automation tasks centralized and decentralized. The single modules of the System 300V from VIPA are similar in construction to Siemens. Due to the compatible backplane bus, the modules from VIPA and Siemens can be mixed.

- System 300S

  The System 300S extends the central area with high-speed CPUs that have the integrated SPEED7 chip. Additionally some CPU's have got a parallel SPEED-Bus that allows the modular connection of fast peripheral modules like IOs or bus master.

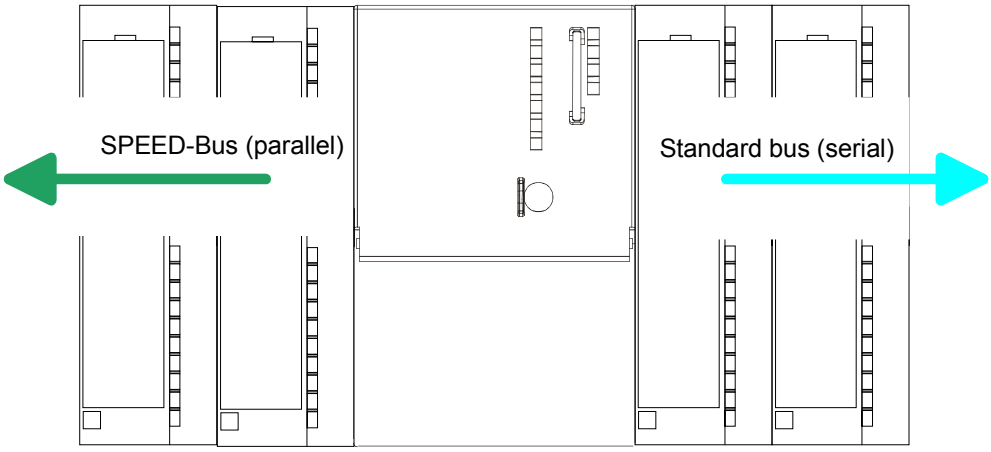


System 300S

Overview

The CPUs 31xS are based 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.

Except of the basic variant, all SPEED7-CPUs are provided with a parallel SPEED-Bus that enables the additional connection of up to 10 modules from the SPEED-Bus periphery. While the standard peripheral modules are plugged-in at the right side of the CPU, the SPEED bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.



CPU 31xS

The System 300S series consists of a number of CPUs. These are programmed in STEP®7 from Siemens. For this you may use WinPLC7 from VIPA or the Siemens SIMATIC manager.

CPUs with integrated Ethernet interfaces or additional serial interfaces simplify the integration of the CPU into an existing network or the connection of additional peripheral equipment.

The user application is stored in the battery buffered RAM or on an additionally pluggable MMC storage module.

Due to the automatic address allocation, the deployment of the CPUs 31xS allows to address 32 peripheral modules.

Additionally some SPEED7-CPUs have got a parallel SPEED-Bus that allows the modular connection of fast peripheral modules like IOs or bus master.

SPEED-Bus

The SPEED-Bus is a 32Bit parallel bus developed from VIPA with a maximum data rate of 40MByte/s. Via the SPEED-Bus you may connect up to 10 SPEED-Bus modules to your CPU 31xS.

In opposite to the "standard" backplane bus where the modules are plugged-in at the right side of the CPU by means of single bus connectors, the modules at the SPEED-Bus are plugged-in at the left side of the CPU via a special SPEED-Bus rail.

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

SPEED-Bus peripheral modules

The SPEED-Bus peripheral modules may exclusively plugged at the SPEED-Bus slots at the left side of the CPU. The following SPEED-Bus modules are in preparation:

- Fast field bus modules like Profibus DP, Interbus, CANopen master and CANopen slave
- Fast CP 343 (CP 343 Communication processor for Ethernet)
- Fast CP 341 with double RS 422/485 interface
- Fast digital input-/output modules (Fast Digital IN/OUT)

Memory management

Every CPU 31xS has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

Starting with CPU firmware 3.0.0 there is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

Integrated Profibus DP master

The CPUs of the System 300S series with SPEED-Bus have an integrated Profibus DP master. Via the DP master with a data range of 1kByte for in- and output you may address up to 125 DP slaves.

The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens.

Integrated Ethernet PG/OP channel

Every CPU 31xS has an Ethernet interface for PG/OP communication. Via the "PLC" functions you may directly access the Ethernet PG/OP channel and program res. remote control your CPU. A max. of 4 PG/OP connections is available.

You may also access the CPU with a visualization software via these connections.

Operation Security

- Wiring by means of spring pressure connections (CageClamps) at the front connector
- Core cross-section 0.08...2.5mm^2
- Total isolation of the wiring at module change
- Potential separation of all modules to the backplane bus
- ESD/Burst acc. IEC 61000-4-2/IEC 61000-4-4 (up to level 3)
- Shock resistance acc. IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)

Environmental conditions

- Operating temperature: 0 ... +60°C
- Storage temperature: -25 ... +70°C
- Relative humidity: 5 ... 95% without condensation
- Ventilation by means of a fan is not required

Dimensions/ Weight

- Available lengths of the profile rail in mm: 160, 482, 530, 830 and 2000
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 40x125x120
  2tier width: (HxWxD) in mm: 80x125x120

Compatibility

Modules and CPUs of the System 300 from VIPA and Siemens may be used at the "Standard" bus as a mixed configuration.

The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens.

The SPEED7 CPUs from VIPA are instruction compatible to the programming language STEP®7 from Siemens and may be programmed via WinPLC7 from VIPA or via the Siemens SIMATIC manager.

Here the instruction set of the S7-400 from Siemens is used.

Note!

Please do always use the **CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)** from Siemens of the hardware catalog to project a SPEED7-CPU with SPEED-Bus from VIPA. For the project engineering, a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens is required!

Integrated power supply

Every CPU res. bus coupler comes with an integrated power supply. The power supply has to be supplied with DC 24V. By means of the supply voltage, the bus coupler electronic is supplied as well as the connected modules via backplane bus. Please regard that the integrated power supply may supply the backplane bus the backplane bus (SPEED-Bus and Standard-Bus) depending on the CPU with a sum with max. 5A.

The power supply is protected against inverse polarity and overcurrent.

Every SPEED-Bus rail has a plug-in option for an external power supply. This allows you to raise the maximum current at the backplane bus for 5.5A.

Chapter 2 Assembly and installation guidelines

Overview

In this chapter you will find all information, required for the installation and the cabling of a process control with the components of the System 300.

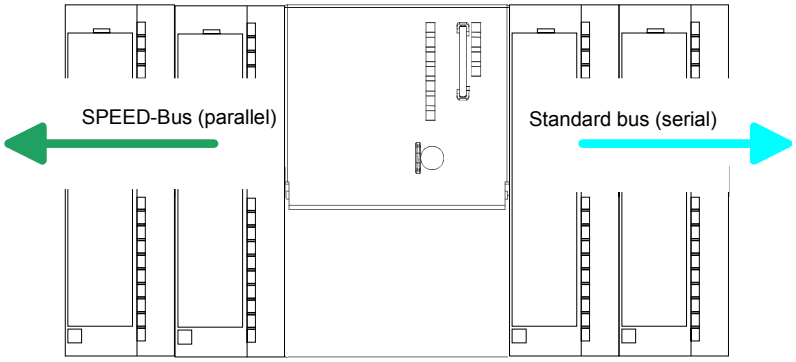
Content

Overview

General

While the standard peripheral modules are plugged-in at the right side of the CPU, the SPEED-Bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.

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



Serial Standard bus

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

The backplane bus coupler is included in the delivery of the peripheral modules.

Parallel SPEED-Bus

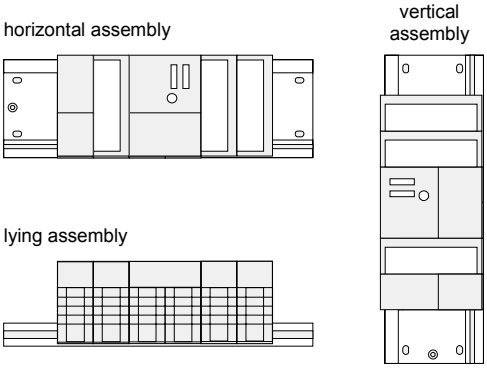
With SPEED-Bus the bus connection happens via a SPEED-Bus rail integrated in the profile rail at the left side of the CPU. Due to the parallel SPEED-Bus not all slots must be occupied in sequence.

SLOT 1 for additional power supply

At SLOT 1 DCDC) you may plug either a SPEED-Bus module or an additional power supply.

Assembly possibilities

You may assemble the System 300 horizontally, vertically or lying.



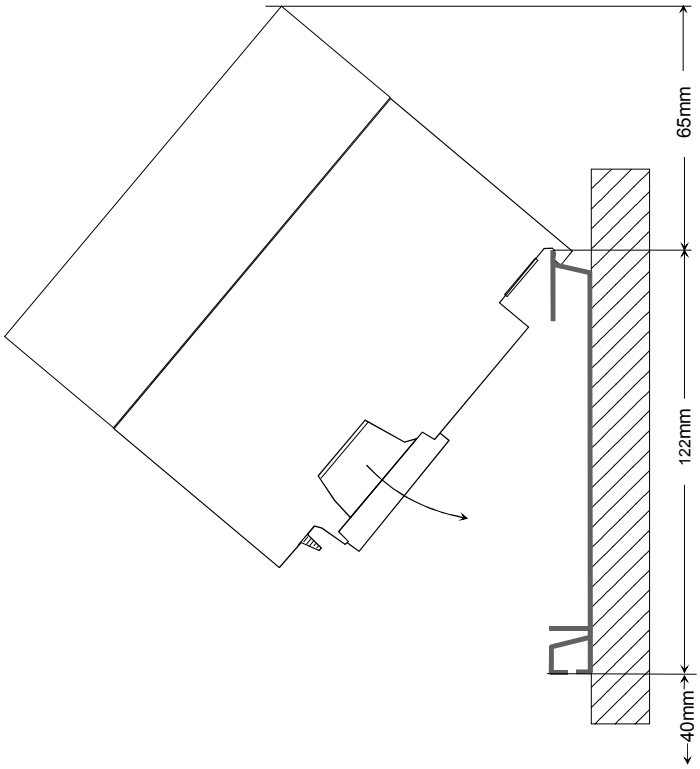
Please regard the allowed environment temperatures:

- horizontal assembly: from 0 to 60°C
- vertical assembly: from 0 to 40°C
- lying assembly: from 0 to 40°C

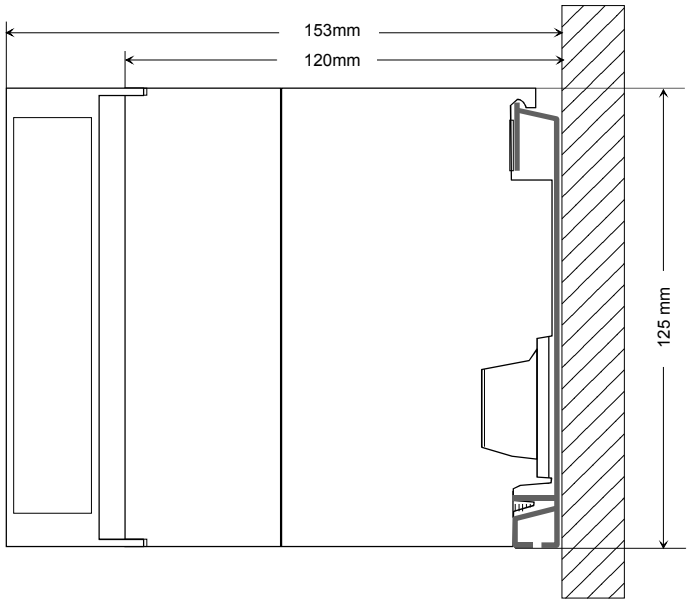
Installation dimensions

Dimensions Basic enclosure 1tier width (WxHxD) in mm: 40 x 125 x 120

Dimensions



Installation dimensions

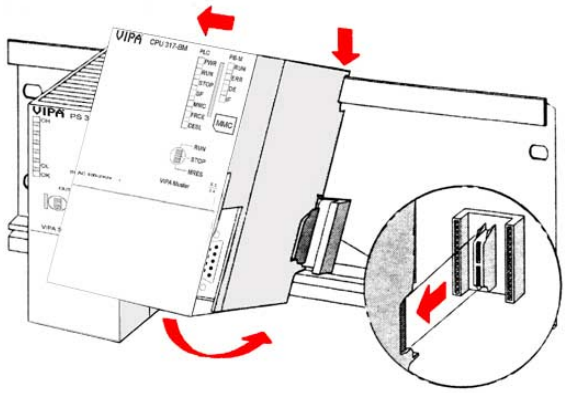


Installation Standard-Bus

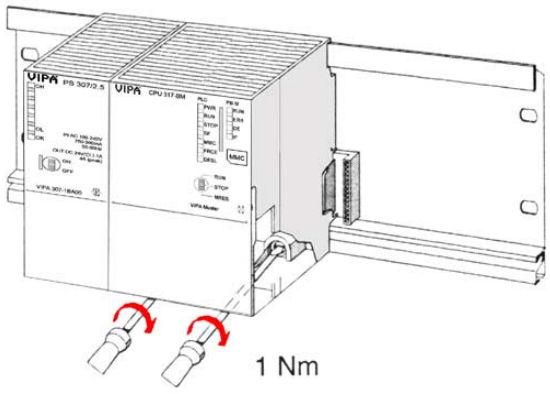
Approach

If you do not deploy SPEED-Bus modules, the assembly at the standard bus happens at the right side of the CPU with the following approach:

- 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.
- If the background is a grounded metal or device plate, please look for a low-impedance connection between profile rail and background.
- Connect the profile rail with the protected earth conductor. For this purpose there is a bolt with M6-thread.
- The minimum cross-section of the cable to the protected earth conductor has to be 10mm^2.



- Stick the power supply to the profile rail and pull it to the left side up to 5mm to the grounding bolt of the profile rail.
- Take a bus coupler and click it at the CPU from behind like shown in the picture.
- Stick the CPU to the profile rail right from the power supply and pull it to the power supply.



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

Danger!

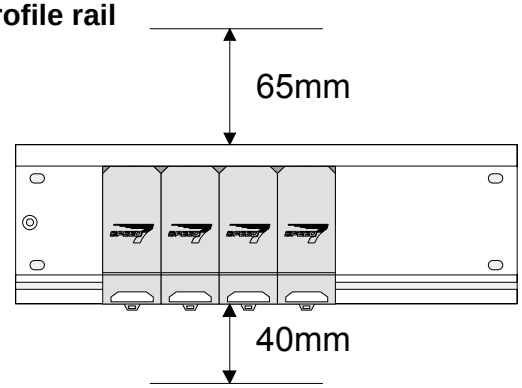
- Before installing or overhauling the system, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

Assembly SPEED-Bus

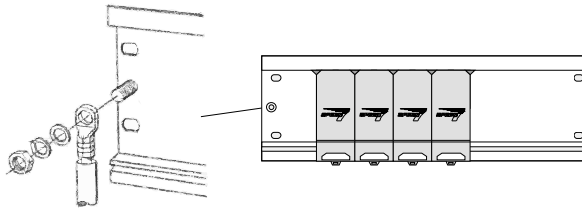
Pre-manufactured SPEED-Bus profile rail

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

Installation of the profile rail



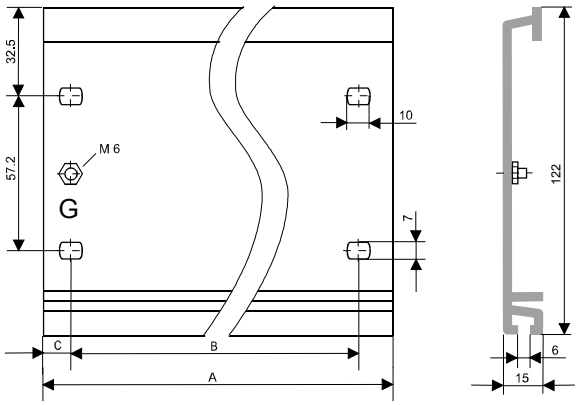
- Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail.
- Please look for a low-impedance connection between profile rail and background



- Connect the profile rail with the protected earth conductor.

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

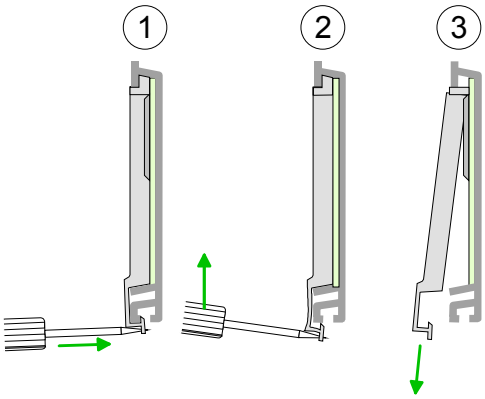
Profile rail



Order number	SPEED-Bus slots	A	B	C
VIPA 390-1AB60	-	160mm	140mm	10mm
VIPA 390-1AE80	-	482mm	466mm	8,3mm
VIPA 390-1AF30	-	530mm	500mm	15mm
VIPA 390-1AJ30	-	830mm	800mm	15mm
VIPA 390-9BC00*	-	2000mm	-	15mm
VIPA 391-1AF10	2	530mm	500mm	15mm
VIPA 391-1AF30	6	530mm	500mm	15mm
VIPA 391-1AF50	10	530mm	500mm	15mm

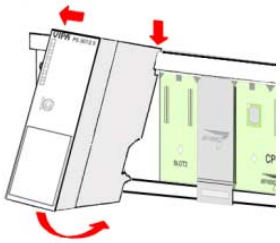
* Unit pack 10 pieces

Installation SPEED-Bus-Module

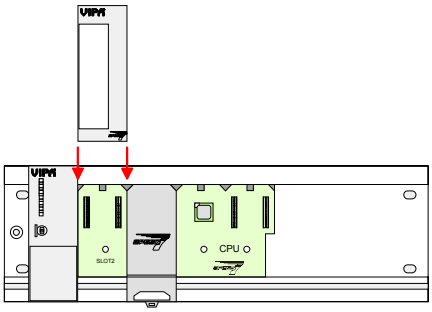


- Dismantle the according protection flaps of the SPEED-Bus plug-in locations with a screw driver (open and pull down).

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

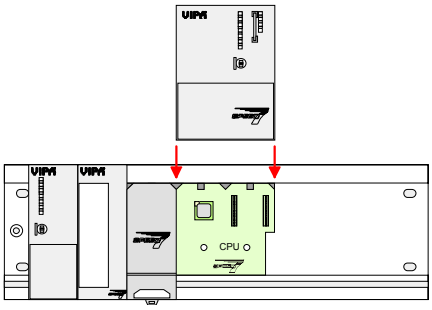


- At deployment of a DC 24V power supply, install it at the shown position at the profile rail at the left side of the SPEED-Bus and push it to the left to the isolation bolt of the profile rail.
- Fix the power supply by screwing.



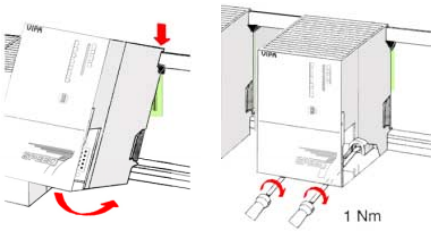
- To connect the SPEED-Bus modules, plug it between the triangular positioning helps to a plug-in location marked with "SLOT ..." and pull it down.
- Only the "SLOT1 DCDC" allows you to plug-in either a SPEED-Bus module or an additional power supply.
- Fix the modules by screwing.

Installation CPU without Standard-Bus-Modules

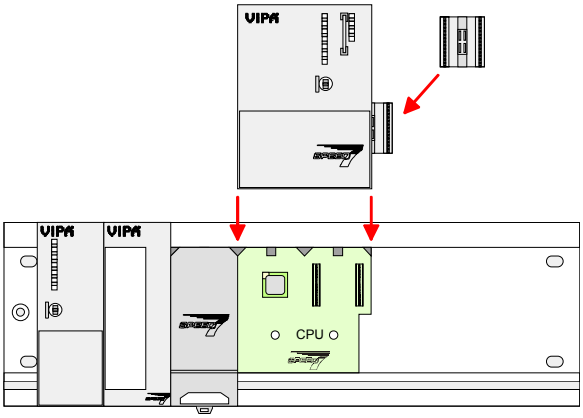


- To deploy the SPEED7-CPU exclusively at the SPEED-Bus, plug it between the triangular positioning helps to the plug-in location marked with "CPU SPEED7" and pull it down.
- Fix the CPU by screwing.

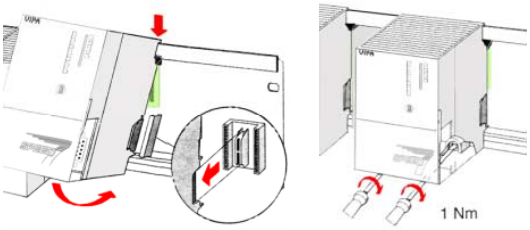
Please regard that not all CPU 31xS may be deployed at the SPEED-Bus!



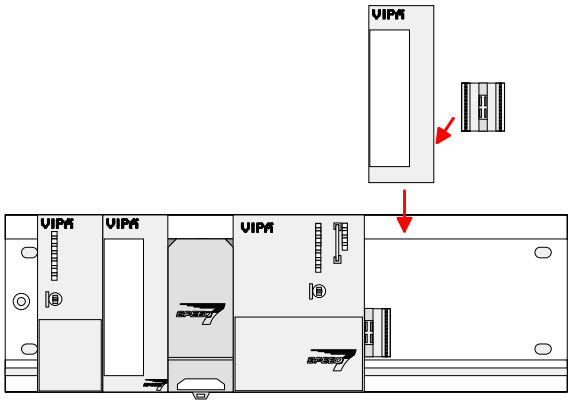
Installation CPU with Standard-Bus-Modules



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



Installation Standard-Bus-Modules



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

Danger!

- Before installing or overhauling the System, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

Cabling

Overview

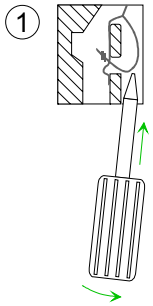
The power supplies and CPUs are exclusively delivered with CageClamp contacts. For the signal modules the front connectors are available from VIPA with screw contacts. In the following all connecting types of the power supplies, CPUs and input/output modules are described.

Danger!

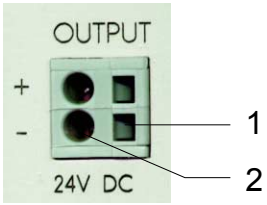
- Before installation or overhauling, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

CageClamp technology (gray)

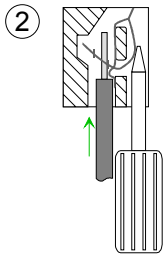
For the cabling of power supplies, bus couplers and parts of the CPU, gray connectors with CageClamp technology are used.



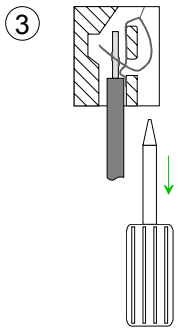
You may connect wires with a cross-section of 0.08mm^2 to 2.5mm^2. You can use flexible wires without end case as well as stiff wires.



[1] Rectangular opening for screwdriver
[2] Round opening for wires



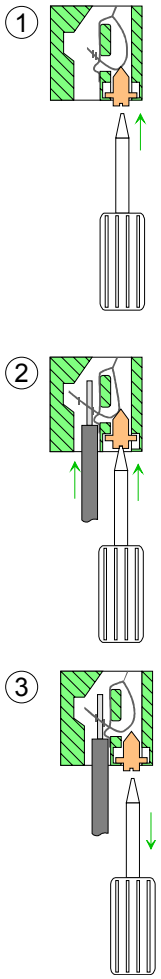
The picture on the left side shows the cabling step by step from top view.



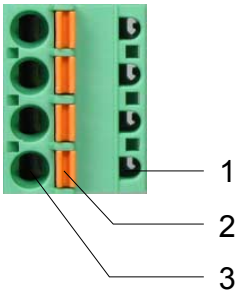
- To conduct a wire you plug a fitting screwdriver obliquely into the rectangular opening like shown in the picture.
- To open the contact spring you have to push the screwdriver in the opposite direction and hold it.
- Insert the insulation striped wire into the round opening. You may use wires with a cross-section from 0.08mm^2 to 2.5mm^2.
- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

CageClamp technology (green)

For the cabling of e.g. the power supply of a CPU, green plugs with CageClamp technology are deployed.



Here also you may connect wires with a cross-section of 0.08mm^2 to 2.5mm^2. 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.

- For cabling you push the locking vertical to the inside with a suiting screwdriver and hold the screwdriver in this position.
- Insert the insulation striped wire into the round opening. You may use wires with a cross-section from 0.08mm^2 to 2.5mm^2.
- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

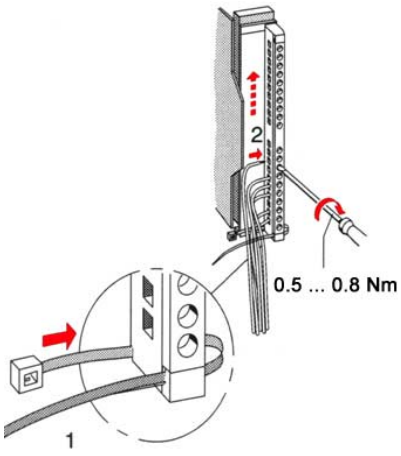
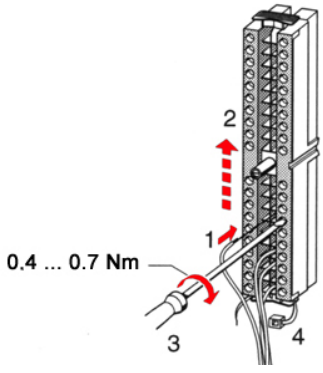
Note!

In opposite to the gray connection clamp from above, the green connection clamp is realized as plug that can be clipped off carefully even if it is still cabled.

Front connectors of the in-/output modules

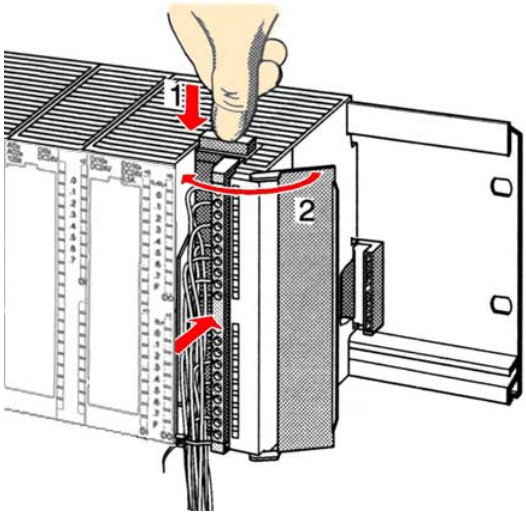
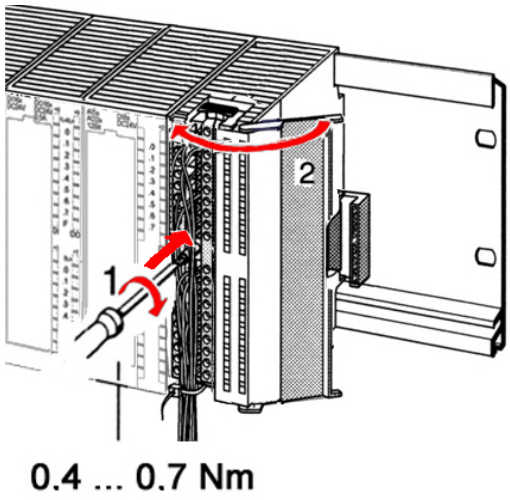
In the following the cabling of the three variants of the front-facing connector is shown:

For the I/O modules the following plugs are available at VIPA:

<table>
<tr><th>20pole screw connection
VIPA 392-1AJ00</th><th>40pole screw connection
VIPA 392-1AM00</th></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td colspan="2">Open the front flap of your I/O module.</td></tr>
<tr><td colspan="2">Bring the front connector in cabling position.
Here fore 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.</td></tr>
<tr><td colspan="2">Strip the insulation of your wires. If needed, use core end cases.</td></tr>
<tr><td>Thread the included cable binder into the front connector.</td><td></td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td colspan="2">Bolt also the connection screws of not cabled screw clamps.</td></tr>
<tr><td> 
</td><td>Put the included cable binder around the cable bundle and the front connector.
 
</td></tr>
<tr><td colspan="2">Fix the cable binder for the cable bundle.</td></tr>
</table>

continued ...

... continue

<table>
<tr><th>20pole screw connection
VIPA 392-1AJ00</th><th>40pole screw connection
VIPA 392-1AM00</th></tr>
<tr><td>Push the release key at the front connector on the upper side of the module and at the same time push the front connector into the module until it locks.
 
</td><td>Bolt the fixing screw of the front connector.
 
</td></tr>
<tr><td colspan="2">Now the front connector is electrically connected with your module.</td></tr>
<tr><td colspan="2">Close the front flap.</td></tr>
<tr><td colspan="2">Fill out the labeling strip to mark the single channels and push the strip into the front flap.</td></tr>
</table>

Installation Guidelines

General

The installation guidelines contain information about the interference free deployment of the System 300. There is the description of the ways, interference may occur in your control, how you can make sure the electromagnetic digestibility (EMC), and how you manage the isolation.

What means EMC?

Electromagnetic digestibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interferenced res. without interferencing the environment.

All System 300 components are developed for the deployment in hard industrial environments and fulfill 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:

- Fields
- I/O signal conductors
- Bus system
- Current supply
- Protected earth conductor

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

One differs:

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

Basic rules for EMC

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

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminum parts. Aluminum 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 res. data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).
- Proof the correct fixing of the lead isolation.
  - Data lines must be laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided lying of the isolation may be favorable.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metalized plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Wire all inductivities with erase links that are not addressed by the modules.
  - For lightening cabinets you should prefer incandescent lamps and avoid luminescent lamps.
- 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 is a protection and functionality activity.
  - Connect installation parts and cabinets with the System 300 in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If potential differences between installation parts and cabinets occur, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

Electrical, magnetic 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. Hereby you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve a 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 res. µA) are transferred
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metalized plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to the System 300 module and **don't** lay it on there again!

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

Chapter 3 Digital I/O modules FAST - SPEED-Bus

Overview

Content of this chapter is the structure and the functionality of the fast digital SPEED-Bus modules from VIPA.

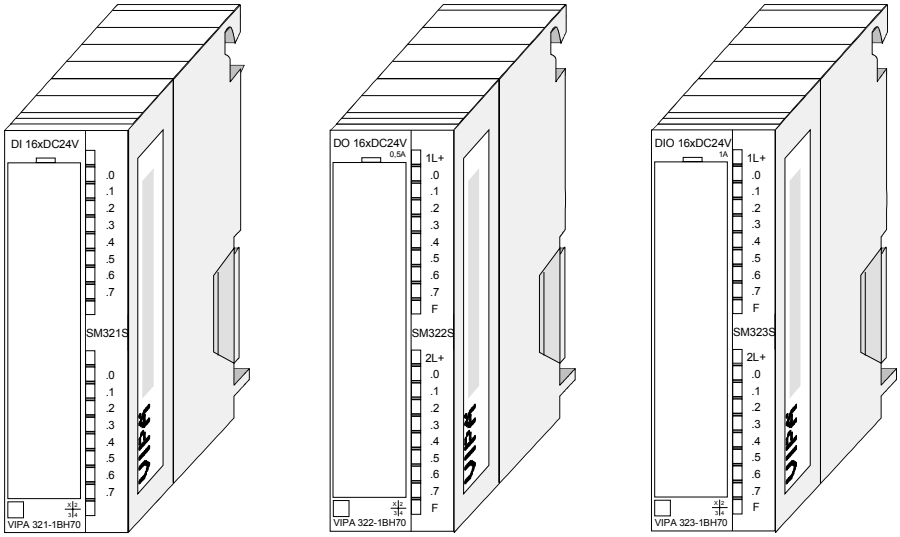
The modules can only be used at SPEED-Bus slots at the left side of the CPU.

Content

System overview

I/O modules for SPEED-Bus

- SM 321S Fast Digital Input DI 16xDC 24V for SPEED-Bus
- SM 322S Fast Digital Output DO 16xDC 24V 0.5A for SPEED-Bus
- SM 323S Fast Digital Input/Output DIO 16xDC 24V 0.5A for SPEED-Bus



Order data I/O modules

Type	Order No.	Page
DI 16xDC 24V	VIPA 321-1BH70	3-8
DO 16xDC 24V 0.5A	VIPA 322-1BH70	3-19
DIO 16xDC 24V 0.5A	VIPA 323-1BH70	3-21

Security hint for DIO modules

Attention!

Please regard that the voltage applied to an output channel must be ≤ the voltage supply applied to L+.

Due to the parallel connection of in- and output channel per group, a set output channel may be supplied via an applied input signal. Thus, a set output remains active even at power-off of the voltage supply with the applied input signal. Non-observance may cause module demolition.

Addressing at SPEED-Bus

Overview

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU.

With no hardware configuration present, the CPU assigns automatically peripheral I/O addresses during boot procedure depending on the plug-in location amongst others also for plugged modules at the SPEED-Bus.

Maximal pluggable modules

In the hardware configurator from Siemens up to 8 modules per row may be parameterized. At deployment of SPEED7 CPUs up to 32 modules at the standard bus and 10 further modules at the SPEED-Bus may be controlled. CPs and DP masters that are additionally virtual configured at the standard bus are taken into the sum of 32 modules at the standard bus.

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

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 via a virtual Profibus system by including the SPEEDBUS.GSD may be used. For this, click on the properties of the according module and set the wanted address.

Automatic addressing

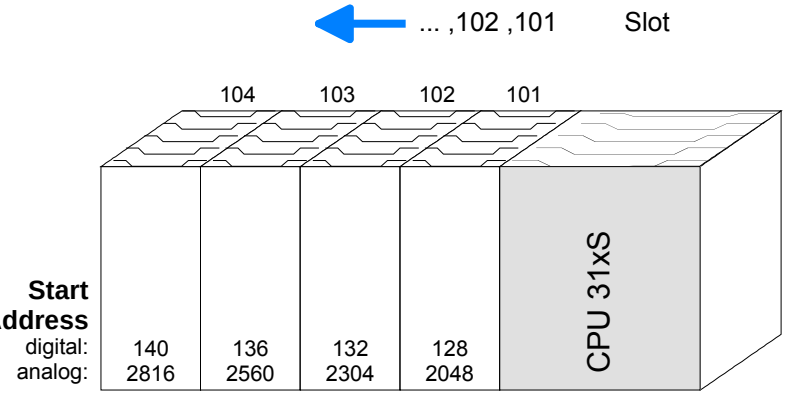
If you do not like to use a hardware configuration, an automatic addressing comes into force.

At the automatic address allocation DIOs are mapped depending on the slot location with a distance of 4byte and AIOs, FMs, CPs with a distance of 256byte.

Depending on the slot location the start address from where on the according module is stored in the address range is calculated with the following formulas:

DIOs: Start address = 4·(slot -101)+128

AIOs, FMs, CPs: Start address = 256·(slot -101)+2048



Project engineering

Overview

Every module at the SPEED-Bus including the CPU has to be configured as single "VIPA_SPEEDbus"-DP slave at a virtual DP master (342-5DA02 V5.0 from Siemens). For this you have to include the GSD speedbus.gsd.

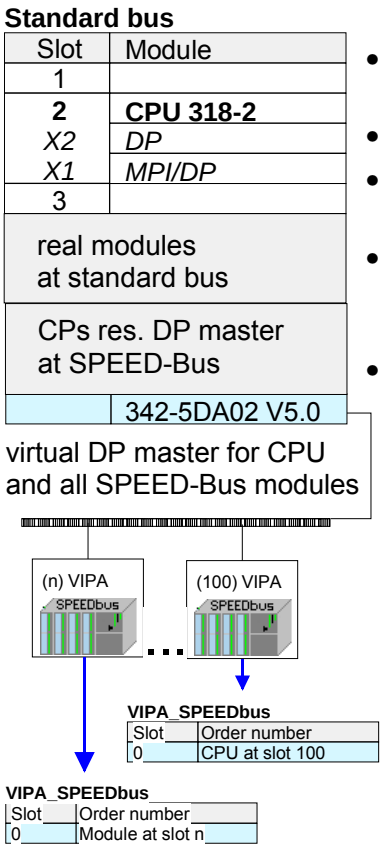
Every "VIPA_SPEEDbus"-DP slave has exactly one slot for the project engineering where you must place the according SPEED-Bus module. The assignment of a SPEED-Bus slave to a SPEED-Bus slot number takes place via the Profibus address starting with 100.

Fast introduction

For the employment of the I/O modules at the SPEED-Bus the inclusion via the GSD-file from VIPA in the hardware catalog is required.

To be compatible with the Siemens SIMATIC manager, you have to execute the following steps:

- Start the hardware configurator from Siemens and include the speedbus.gsd for SPEED7 from VIPA.
- Configure CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) from Siemens.
- Starting with slot 4, place the System 300 modules in the plugged sequence.
- Project engineering and connection of the SPEED-Bus-CPs res. –DP master at the standard bus as virtual CP 343-1 (343-1EX11) res. CP 342-5 (342-5DA02 V5.0)
- For the SPEED-Bus you always include, connect and parameterize to the *operating mode* DP master the DP master CP 342-5 (342-5DA02 V5.0) as last module. To this master system you assign every SPEED-Bus module as VIPA_SPEEDbus slave. Here the Profibus address corresponds to the slot number beginning with 100 for the CPU. Place at slot 0 of every slave the assigned module and alter the parameters if needed.



Preconditions

The hardware configurator is part of the Siemens SIMATIC manager. It serves for project engineering. The modules that may be configured here are listed in the hardware catalog.

For the employment of the System 300S modules at the SPEED-Bus you have to include the System 300S modules into the hardware catalog via the GSD-file speedbus.gsd from VIPA.

Note!

For the project engineering, a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens is required!

Include the SPEED7-GSD-file

- Browse to www.vipa.de > Service > Download > GSD- and EDS-Files > Profibus and select the file *Cx000023_Vxxx*.
- Extract the file to your work directory. The SPEEDbus.gsd is stored in the directory System_300S.
- Start the hardware configurator from Siemens.
- Close all projects.
- Select **Options** > *Install new GSD-file.*
- Change to the directory System_300S and select the **SPEEDBUS.GSD**.

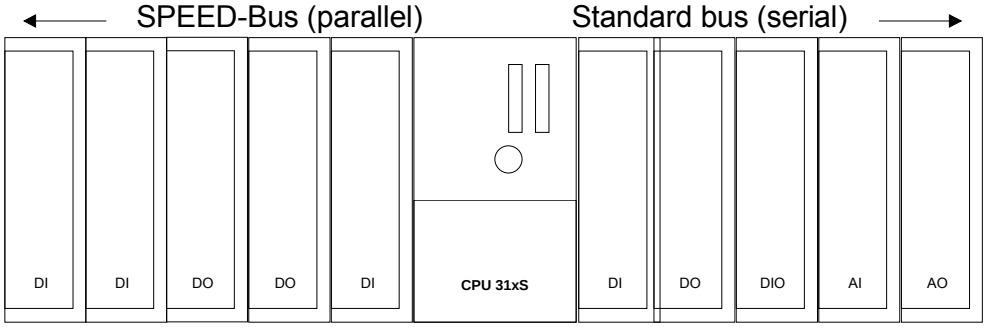
The modules of the System 300S from VIPA are now included in the hardware catalog under:
Profibus-DP / Additional field devices / I/O / VIPA_SPEEDbus.

Steps of project engineering

The following text describes the approach of the project engineering in the hardware configurator from Siemens at an abstract sample.

The project engineering is separated into following parts:

- Project engineering of the modules at the standard bus
- Project engineering of the SPEED-Bus modules in a virtual master system (speedbus.gsd required)



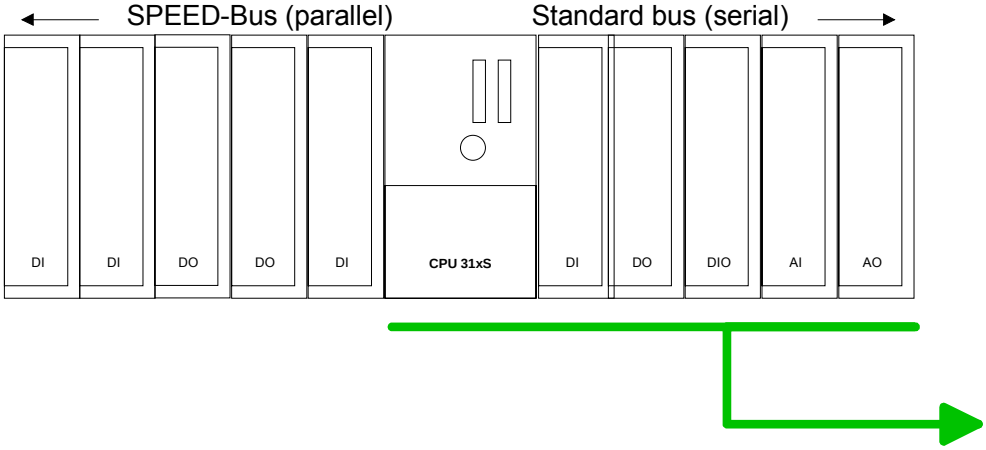
Preconditions

For the employment of the System 300S modules at the SPEED-Bus you have to include the System 300S modules into the hardware catalog via the GSD-file speedbus.gsd from VIPA.

Project engineering of the modules at the standard bus

The modules at the right side of the CPU at the standard bus are configured with the following approach:

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)
- Include your System 300V modules at the standard bus in the plugged sequence starting with slot 4.
- Parameterize the CPU res. the modules where appropriate. The parameter window opens by a double click on the according module.
- To extend the bus you may use the IM 360 from Siemens where you can connect up to 3 further extension racks via the IM 361. Bus extensions are always placed at slot 3.
- Save your project.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO

Note!

To extend the bus you may use the IM 360 from Siemens where you can connect up to 3 further extension racks via the IM 361. Bus extensions are always placed at slot 3.

Project engineering of all SPEED-Bus modules in a virtual master system

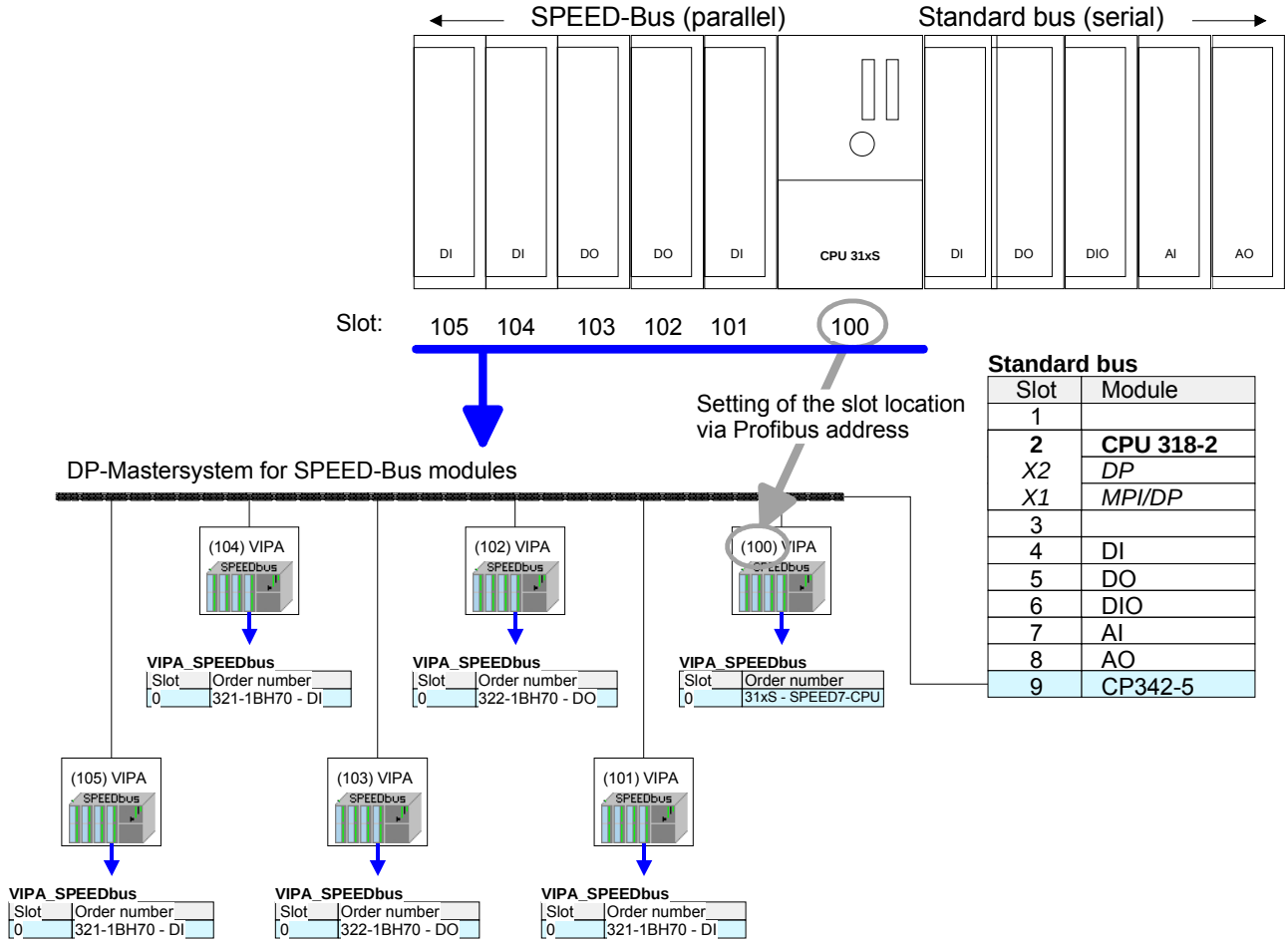
The slot assignment of the SPEED-Bus modules and the parameterization of the in-/output periphery happens via a virtual Profibus DP master system. For this, place as last module a DP master (342-5DA02 V5.0) with master system.

For the employment of the System 300S modules at the SPEED-Bus the inclusion of the System 300S modules into the hardware catalog via the GSD-file speedbus.gsd from VIPA is required.

After the installation of the speedbus.gsd you may locate under *Profibus DP* / *Additional field devices* / *I/O* / *VIPA_SPEEDbus* the DP slave system vipa_speedbus.

Now include for the CPU and every module at the SPEED-Bus a slave system "vipa_speedbus".

Set as Profibus address the slot no. (100...110) of the module and place the according module from the hardware catalog of VIPA_speedbus to slot 0 of the slave system.



The according module is to be taken over from the HW Catalog of vipa_speedbus to slot 0.

321-1BH70 - DI 16xDC 24V

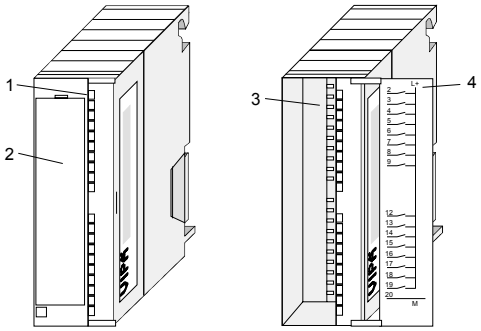
Order data DI 16xDC 24V VIPA 321-1BH70

Description The fast digital input module collects the binary control signals from the process level and transmits them galvanically separated to the superordinated bus system. It has 16 channels and their status is monitored via LEDs.

Properties

- 16 fast input channels, isolated to SPEED-Bus
- Extended parameterization possibility
- Nominal input voltage DC 24V
- Useable for switches and approximate switches
- Status monitoring of the channels via LEDs

Construction



[1] LEDs

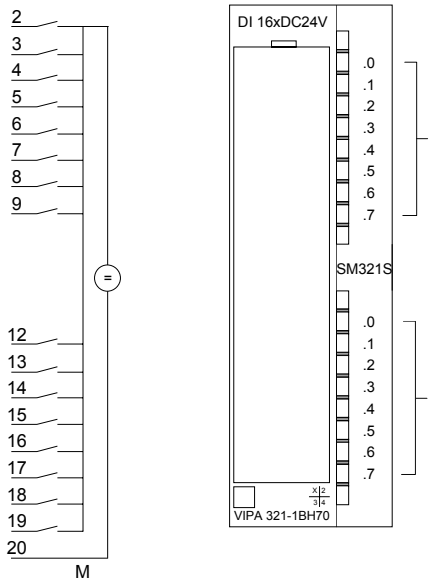
[2] flap with labeling strip

[3] contact bar

[4] flap opened with inner label

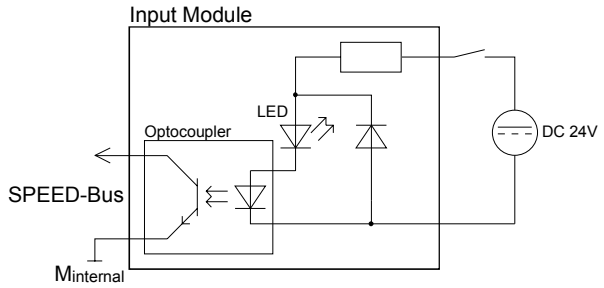
Pin assignment
Circuit diagram
Status monitor

Pin	Assignment
1	not used
2	Input I+0.0
.	.
.	.
.	.
9	Input I+0.7
10	not used
11	not used
12	Input I+1.0
.	.
.	.
.	.
19	Input I+1.7
20	Ground



LED	Description
.0 7	LEDs (green) I+0.0 to I+1.7 from ca. 15V on, the signal is recognized as "1" and the according LED is addressed

Schematic diagram



Parameterization

Depending on the project engineering, 2byte res. 48byte in the input range of the CPU where occupied by the module. You have the following opportunities for the project engineering: occupies the module. For project engineering there are the following possibilities:

- Project engineering as *321-1BH70 DI16xDC24V*

  Range in PII: 2byte

  Parameter: none

- Project engineering as *DI16 Alarm/ETS*

  Range in PII: 48byte

  Parameter: 66byte (edge selection, time stamp, filter)

Project engineering as 321-1BH70 DI16xDC24V

At this project engineering you have no parameterization options and your module occupies 2bytes in the input address range of the CPU.

If no hardware project engineering is present, this operation mode is used as default. Here the module has a delay time of 6.12µs.

Project engineering as DI16 Alarm/ETS

If the module is defined as "DI16 Alarm/ETS" in the hardware configuration, it occupies 48byte of the input range of the CPU and can be parameterized with 66byte of data.

The following parameters are at your disposal:

- Diagnostic interrupt (global for all inputs)
- Edge selection (switchable process interrupt per channel)
- Time stamp (activation of a time stamp function per channel)
- Filter value (for grubby input signals per channel)

Structure of the parameter data

During the parameterization, a parameter area of 66byte is transferred in the record sets 7Fh, 80h and 81h. Using the SFCs 55, 56 and 57 you may alter parameters and transfer them to the module during runtime.

The record sets have the following structure:

Record set 7Fh
Diagnostic interrupt

This record set activates res. de-activates the diagnostic function. A diagnostic interrupt occurs when during a process interrupt execution another process interrupt is initialized for the same event.

The record set has the following structure:

Word	
0	*Bit 15 ... 0: Diagnostic interrupt* 0000h = de-activated 0001h = activated
2	*Bit 15 ... 0: reserved*

Record set 80h
Edge selection

Via this record set you may activate a process interrupt for I+0.0 ... I+1.7 and define for which edge of the input signal a process interrupt is thrown.

The record set has the following structure:

Byte	Bit 7 ... 0
0	*Bit 1 ... 0: Edge selection I+0.0* 00b = de-activated 01b = Process interrupt at ascending edge 10b = Process interrupt at descending edge 11b = Process interrupt at both edges *Bit 7 ... 2: reserved*
...	...
15	*Bit 1 ... 0: Edge selection I+1.7* 00b = de-activated 01b = Process interrupt at ascending edge 10b = Process interrupt at descending edge 11b = Process interrupt both edges *Bit 7 ... 2: reserved*

continued ...

... continue record set 80h

Record set 80h
ETS time stamp
(Byte 16 ... 31)

Every SPEED-Bus module carries along a timer with a resolution of 1µs. The timer starts at boot-up of the CPU. Thus gives you a time base with an accuracy of ±1µs at the SPEED-Bus. By parameterization of the ETS function (**E**dge **T**ime**s**tamp) for an input, the current time value is entered in the process image of the module at according edge. Thus allows you to compare times of different input channels via your user application.

Via the parameter *Time stamp* you may activate the ETS system and define the edge of the input signal that initiates the process image entry of a time stamp. You have the following options:

- No time stamp
- Time stamp at ascending edge
- Time stamp at descending edge
- Time stamp at both edges

The allocation in the process image is illustrated at the following page.

Note!

The stored times correspond the point in time when the signal has already passed the input filter of the module. To calculate the real time at the clamp, you have to subtract the delay time of 1µs and the parameterized delay time defined under *Filter*.

Byte	Bit 7 ... Bit 0
16	Bit 1,0: Time stamp channel 0 (I+0.0) 00 = no time stamp 01 = Time stamp at ascending edge 10 = Time stamp at descending edge 11 = Time stamp at both edges
...	...
31	Bit 1,0: Time stamp channel 15 (I+1.7) 00 = no time stamp 01 = Time stamp at ascending edge 10 = Time stamp at descending edge 11 = Time stamp at both edges

Record set 81h
Input filter

This record set allows you to preset an input filter in steps of 5.12µs steps for I+0.0 ... I+1.7. By preceding a filter you define how long an input signal must be present before it is recognized as "1" signal. With the help of filters you may e.g. filter signal peaks at a blurred input signal.

The entry happens as a factor of 5.12µs and is within the range 1 ... 31250 i.e. 5.12µs ... 160ms. The record set has the following structure:

Word	
0	Input filter I+0.0 in 5.12µs
...	...
30	Input filter I+1.7 in 5.12µs

48bytes in the process image

The module occupies 48byte in the input address range of the CPU that have the following meaning:

Byte	Bit 7 ... Bit 0
0	State of the channels (1 = set, 0 = not set) Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
1	State of the channels (1 = set, 0 = not set) Bit 8: Status I+1.0 ... Bit 15: Status I+1.7
2 ... 3	reserved
4	1. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) it is set back. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
5	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
6 ... 7	reserved
8	Edge lost (1 = edge lost, 0 = no edge lost) Here is noted if an edge change has been lost, i.e. if there has been more than one edge change since the last read access. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
9	Edge lost Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
10...11	reserved

continued ...

... continue

Byte	+3	+2	+1	+0
	The following bytes contain the values of the µs ticker for a channel at the time of an edge change. Only the lower 16bit are taken over. An overflow after 65ms has to be accordingly processed in the user application.			
12	Time stamp I+0.1		Time stamp I+0.0	
16	Time stamp I+0.3		Time stamp I+0.2	
20	Time stamp I+0.5		Time stamp I+0.4	
24	Time stamp I+0.7		Time stamp I+0.6	
28	Time stamp I+1.1		Time stamp I+1.0	
32	Time stamp I+1.3		Time stamp I+1.2	
36	Time stamp I+1.5		Time stamp I+1.4	
40	Time stamp I+1.7		Time stamp I+1.6	

Byte	Bit 7 ... Bit 0
44	2. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) the register is not reset. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
45	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
46...47	reserved

For guarantee of consistency of a µs ticker entry to the *1. edge evaluation* (FA1) the *2. edge evaluation* (FA2) serves for.

The consistency is ensured only if the appropriate bit of the *FA2* is "0". Since the last read access if more than one edge change took place, the corresponding bit of *edge lost* (FV) is set. Here the µs ticker entry contains the time of the last edge.

Example:

	Byte	+1								+0							
FA1	4	0	1	1	0	0	0	1	0	0	1	0	1	1	1	1	0
FV	8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
...																	
FA2	44	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0

The consistent µs ticker entries can be determined by logical bit operations: *FA1* AND NOT *FA2*

Result bit operation:	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0

Process interrupt

Via the edge selection you may activate a process interrupt in your parameterization and define on which edge of the input signal a process interrupt should be initialized.

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 the OB 40

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

Local byte	Bit 7 ... Bit 0
8	Bit 0: Edge at I+0.0 Bit 1: Edge at I+0.1 Bit 2: Edge at I+0.2 Bit 3: Edge at I+0.3 Bit 4: Edge at I+0.4 Bit 5: Edge at I+0.5 Bit 6: Edge at I+0.6 Bit 7: Edge at I+0.7
9	Bit 0: Edge at I+1.0 Bit 1: Edge at I+1.1 Bit 2: Edge at I+1.2 Bit 3: Edge at I+1.3 Bit 4: Edge at I+1.4 Bit 5: Edge at I+1.5 Bit 6: Edge at I+1.6 Bit 7: Edge at I+1.7
10	00h (fix)
11	00h (fix)

Diagnostic interrupt

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the module.

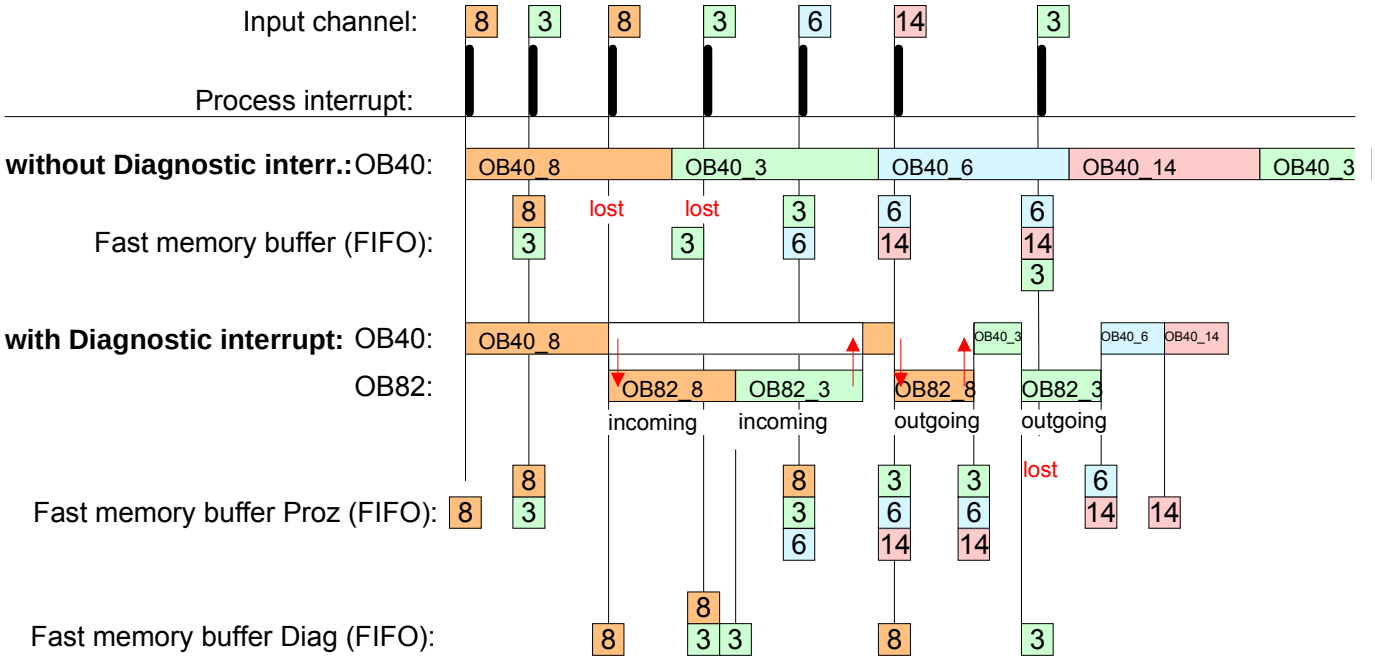
A diagnostic interrupt occurs when during a process interrupt execution in OB 40 another process interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent process interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing$_{incoming}$. 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 ... Bit 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	Bit 5 ... 0: 0 (fix) Bit 6: Process interrupt lost Bit 7: 0 (fix)

Record set 0 Diagnostic$_{outgoing}$

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

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

Diagnostic Record set 1

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Byte	Bit 7 ... Bit 0
0 ... 3	Contents record set 0 (see page before)
4	Bit 6 ... 0: channel type (here 70h) 70h: Digital input Bit 7: More channel types present 0: no 1: yes
5	Number of diagnostic bits the module puts out per channel (here 08h)
6	Number of channels of a module (here 04h)
7	Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) Bit 3: Error in channel group 3 (I+1.4 ... I+I.7) Bit 7 ... 4:0 (fix)
8	Diagnostic interrupt due to "process interrupt lost" at... Bit 0: ... input I+0.0 Bit 1: 0 (fix) Bit 2: ... input I+0.1 Bit 3: 0 (fix) Bit 4: ... input I+0.2 Bit 5: 0 (fix) Bit 6: ... input I+0.3 Bit 7: 0 (fix)
9	Diagnostic interrupt due to "process interrupt lost" at... Bit 0: ... input I+0.4 Bit 1: 0 (fix) Bit 2: ... input I+0.5 Bit 3: 0 (fix) Bit 4: ... input I+0.6 Bit 5: 0 (fix) Bit 6: ... input I+0.7 Bit 7: 0 (fix)
10	Diagnostic interrupt due to "process interrupt lost" at... Bit 0: ... input I+1.0 Bit 1: 0 (fix) Bit 2: ... input I+1.1 Bit 3: 0 (fix) Bit 4: ... input I+1.2 Bit 5: 0 (fix) Bit 6: ... input I+1.3 Bit 7: 0 (fix)
11	Diagnostic interrupt due to "process interrupt lost" at... Bit 0: ... input I+1.4 Bit 1: 0 (fix) Bit 2: ... input I+1.5 Bit 3: 0 (fix) Bit 4: ... input I+1.6 Bit 5: 0 (fix) Bit 6: ... input I+1.7 Bit 7: 0 (fix)
12 ... 15	00h (fix)

Technical Data

Module name	VIPA 321-1BH70 DI16xDC24V	VIPA 321-1BH70 DI16 Alarm/ETS
Dimensions and weight		
Dimensions WxHxD	40x125x120mm	
Weight	200g	
Data for specific module		
Number of inputs	16	
Programming specifications		
Input data	2byte	48byte
Input filter time delay	6.12µs	1µs + param. filter value
Parameter data	-	66byte
Voltages, Currents, Potentials		
Isolation - between channels and backplane bus	yes	
Isolation tested with	DC 500V	
Current consumption - from the SPEED-Bus	390mA	
Power dissipation of the module	3.5W	
Status, Interrupts, Diagnostic		
Status display	green LED per channel	
Data for selecting a sensor		
Input voltage - Rated value - for Signal "1" - for Signal "0"	DC 24V 15 ... 28.8V 0 ... 5V	
Input current - for Signal "1"	7mA	
Connection of 2-wire-BEROs - Permitted bias current	possible 1.5mA	

322-1BH70 - DO 16xDC 24V 0.5A

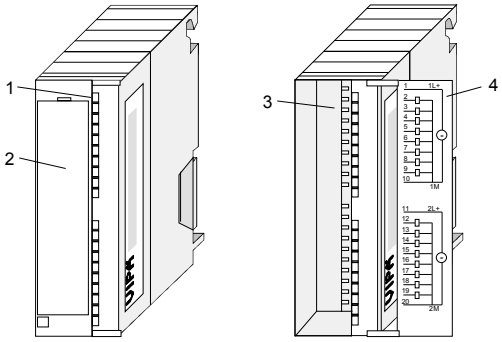
Order data DO 16xDC 24V 0.5A VIPA 322-1BH70

Description The digital fast output module collects the binary control signals from the superordinated bus system and transmits them galvanically separated to the process level. The module has to be provided with 24V via the front slot. It has 16 channels and their status is monitored via LEDs.

Properties

- 16 fast output channels, isolated to SPEED-Bus
- Supply voltage DC 24V
- Output voltage 0.5A
- Useable for magnetic valve and DC contactor
- Status monitoring of the channels via LED

Construction



[1] LEDs

[2] flap with labeling strip

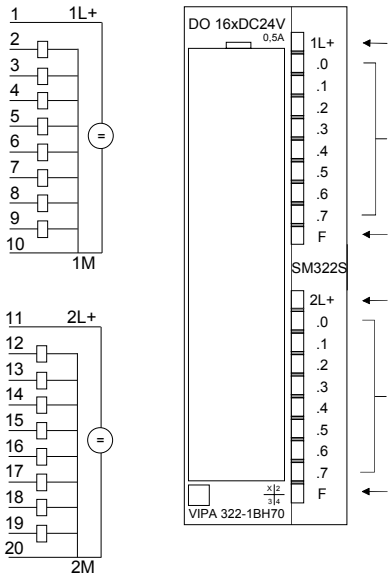
[3] contact bar

[4] flap opened with inner label

Pin assignment
Circuit diagram
Status monitor

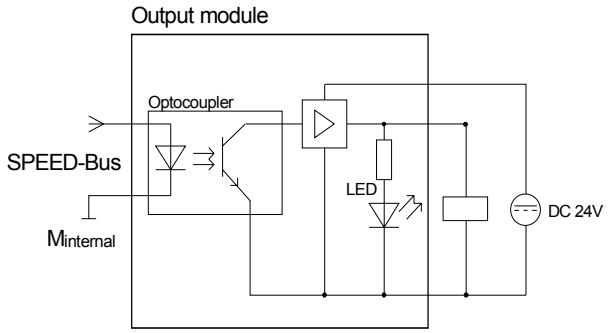
Pin	Assignment
1	Supply voltage DC 24V
2	Output Q+0.0
.	.
.	.
9	Output Q+0.7
10	Ground 1
11	Supply voltage DC 24V
12	Output Q+1.0
.	.
.	.
19	Output Q+1.7
20	Ground 2

Circuit diagram



LED	Description
1L+, 2L+	LED (green) supply voltage is on
.0 7	LEDs (green) Q+0.0 to Q+1.7 As soon as an output is active, the according LED is addressed
F	LED (red) Error when overload or short circuits

Schematic diagram



Technical Data

Module name	VIPA 322-1BH70
Dimensions and weight	
Dimensions WxHxD	40x125x120mm
Weight	200g
Data for specific module	
Number of outputs	16
Programming specifications	
Output data	2byte
Voltages, Currents, Potentials	
Rated load voltage L+	DC 24V
Isolation - between channels and backplane bus - between the channel groups in groups of	 yes yes 8
Insulation tested with	DC 500V
Current consumption - from the load voltage L+ (without load) - from the SPEED-Bus	 30mA 390mA
Power dissipation of the module	4W
Status, Interrupts, Diagnostics	
Status display	green LED per channel
Diagnostic functions - Group error display - Supply voltage display	 red F-LED per group green LED per group
Data for selecting an actuator	
Output voltage - at signal "1"	 DC 24V
Output current - at signal "1" Rated value Permitted range	 0.5A 0.7A
Lamp load max.	5W
Switch rate max. - for resistive load - for inductive load (IEC 947-5-1, DC 13) - for lamp load	 100kHz 0.5Hz 10Hz
Limit (internal) of the inductive circuit interruption voltage	typ. L+ (-52V)

323-1BH70 - DIO 16xDC 24V 0.5A

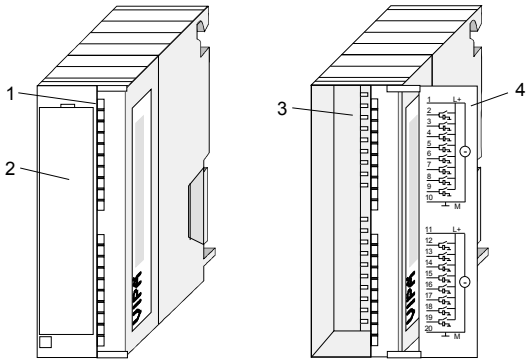
Order Data DIO 16xDC 24V 0.5A VIPA 323-1BH70

Description The module has 16 channels that can be used either as inputs or outputs. Every channel supports a diagnostic function, i.e. as soon as an output is active, the according input is set on "1". If there is a short circuit at the load, the according input is set on "0" and the error can be recognized by analyzing the input.

Properties

- 16 fast input and output channels, isolated to SPEED-Bus
- Extended parameterization possibility
- Nominal input voltage DC 24V
- Output current 0.5A
- Useable for switches, approximate switches, magnetic valve
- Activity monitoring of the channels via LED

Construction



[1] LEDs

[2] flap with labeling strip

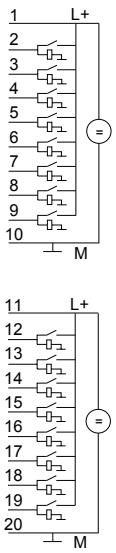
[3] contact bar

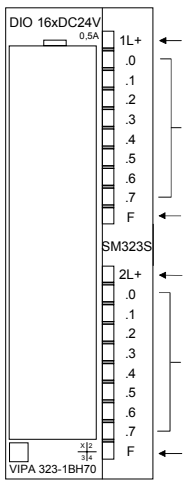
[4] flap opened with inner label

Pin assignment
Circuit diagram
Status monitor

Pin	Assignment
1	Supply voltage +DC 24V
2	I/Q+0.0
...	...
9	I/Q+0.7
10	Ground
11	Supply voltage +DC 24V
12	I/Q+1.0
...	...
19	I/Q+1.7
20	Ground

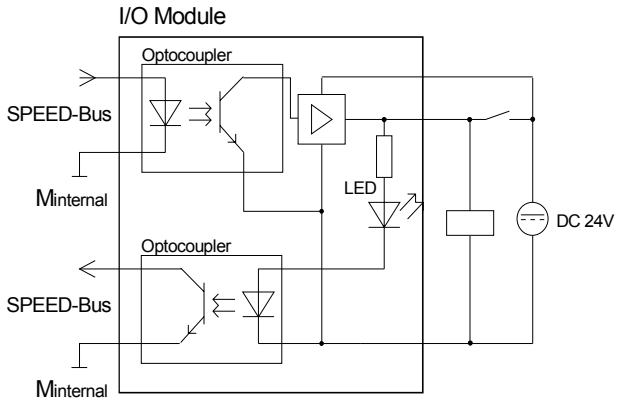
Circuit diagram





LED	Description
1L+, 2L+	LED (green) supply voltage is on
.0 7	LED (green) per byte As soon as an input signal "1" or an active output is recognized, the according LED is addressed
F	LED (red) error at overload or short circuit

Schematic diagram



Due to the parallel connection of in- and output channel per group, a set output can also be provided via a present input signal even if the power supply is shut down and thus it remains active.

Nonobservance may cause destruction of the module!

Parameterization

Dependent on project engineering 2byte each in the in and output range respectively 48byte in the input and 2byte in the output range of the CPU where occupied by the module. For project engineering there are the following possibilities:

- Project engineering as *323-1BH70 DIO16xDC24V*

  Range in PII: 2byte
  Range in PIQ: 2byte
  Parameter: none

- Project engineering as *DIO16 Alarm/ETS*

  Range in PII: 48byte
  Range in PIQ: 2byte
  Parameter: 66byte (edge selection, time stamp, filter)

Project engineering as 323-1BH70 DIO16xDC24V

At this project engineering you have no parameterization options and your module occupies each 2byte in the input and output address range of the CPU.

If no hardware project engineering is present, this operation mode is used as default. Here the module has for the inputs a delay time of 6.12µs.

Project engineering as DIO16 Alarm/ETS

If the module is defined as "DIO16 Alarm/ETS" in the hardware configuration, it occupies 48byte of the input range of the CPU and 2byte in the output range and can be parameterized with 66byte of data.

The following parameters are at your disposal:

- Diagnostic interrupt (global for all inputs)
- Edge selection (switchable process interrupt per channel)
- Time stamp (activation of a time stamp function per channel)
- Filter value (for grubby input signals per channel)

Structure of the parameter data

During the parameterization, a parameter area of 66byte is transferred in the record sets 7Fh, 80h and 81h. Using the SFCs 55, 56 and 57 you may alter parameters and transfer them to the module during runtime.

The record sets have the following structure:

Record set 7Fh
Diagnostic interrupt

This record set activates res. de-activates the diagnostic function. A diagnostic interrupt occurs when during a process interrupt execution another process interrupt is initialized for the same event.

The record set has the following structure:

Word	
0	*Diagnostic interrupt* 0000h = de-activated 0001h = activated
2	*reserved*

Record set 80h
Edge selection
(Byte 0 ... 15)

Via this record set you may activate a process interrupt for I+0.0 ... I+1.7 and define for which edge of the input signal a process interrupt is thrown.

The record set has the following structure:

Byte	Bit 7 ... 0
0	*Bit 1 ... 0: Edge selection I+0.0* 00b = de-activated 01b = Process interrupt at ascending edge 10b = Process interrupt at descending edge 11b = Process interrupt at both edges *Bit 7 ... 2: reserved*
...	...
15	*Bit 1 ... 0: Edge selection I+1.7* 00b = de-activated 01b = Process interrupt at ascending edge 10b = Process interrupt at descending edge 11b = Process interrupt both edges *Bit 7 ... 2: reserved*

continued ...

... continue record set 80h

Record set 80h
ETS Time stamp
(Byte 16 ... 31)

Every SPEED-Bus module carries along a timer with a resolution of 1µs. The timer starts at boot-up of the CPU. Thus gives you a time base with an accuracy of ±1µs at the SPEED-Bus. By parameterization of the ETS function (**E**dge **T**ime**s**tamp) for an input, the current time value is entered in the process image of the module at according edge. Thus allows you to compare times of different input channels via your user application.

Via the parameter *Time stamp* you may activate the ETS system and define the edge of the input signal that initiates the process image entry of a time stamp. You have the following options:

- No time stamp
- Time stamp at ascending edge
- Time stamp at descending edge
- Time stamp at both edges

The allocation in the process image is illustrated at the following page.

Note!

The stored times correspond the point in time when the signal has already passed the input filter of the module. To calculate the real time at the clamp, you have to subtract the delay time of 1µs and the parameterized delay time defined under *Filter*.

Byte	Bit 7 ... Bit 0
16	Bit 1,0: Time stamp channel 0 (I+0.0) 00 = no time stamp 01 = Time stamp at ascending edge 10 = Time stamp at descending edge 11 = Time stamp at both edges
...	...
31	Bit 1,0: Time stamp channel 15 (I+1.7) 00 = no time stamp 01 = Time stamp at ascending edge 10 = Time stamp at descending edge 11 = Time stamp at both edges

Record set 81h
Input filter

This record set allows you to preset an input filter in steps of 5.12µs steps for I+0.0 ... I+1.7. By preceding a filter you define how long an input signal must be present before it is recognized as "1" signal. With the help of filters you may e.g. filter signal peaks at a blurred input signal.

The entry happens as a factor of 5.12µs and is within the range 1 ... 31250 i.e. 5.12µs ... 160ms. The record set has the following structure:

Word	
0	Input filter I+0.0 in 5.12µs
...	...
30	Input filter I+1.7 in 5.12µs

48bytes in the process image

The module occupies 48byte in the input address range of the CPU that have the following meaning:

Input address range

Byte	Bit 7 ... Bit 0
0	State of the channels (1 = set, 0 = not set) Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
1	State of the channels (1 = set, 0 = not set) Bit 8: Status I+1.0 ... Bit 15: Status I+1.7
2 ... 3	reserved
4	1. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) it is set back. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
5	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
6 ... 7	reserved
8	Edge lost (1=edge lost, 0=no edge lost) Here is noted if an edge change has been lost, i.e. if there has been more than one edge change since the last read access. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
9	Edge lost Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
10...11	reserved

continued ...

... continue

Byte	+3	+2	+1	+0
	The following bytes contain the values of the µs ticker for a channel at the time of an edge change. Only the lower 16Bit are taken over. An overflow after 65ms has to be accordingly processed in the user application.			
12	Time stamp I+0.1		Time stamp I+0.0	
16	Time stamp I+0.3		Time stamp I+0.2	
20	Time stamp I+0.5		Time stamp I+0.4	
24	Time stamp I+0.7		Time stamp I+0.6	
28	Time stamp I+1.1		Time stamp I+1.0	
32	Time stamp I+1.3		Time stamp I+1.2	
36	Time stamp I+1.5		Time stamp I+1.4	
40	Time stamp I+1.7		Time stamp I+1.6	

Byte	Bit 7 ... Bit 0
44	2. edge evaluation (1=edge detected, 0=no edge detected) Here the last presence of an edge since the last read access to the register is stored. After a read access to this register (in the module) the register is not reset. Bit 0: Status I+0.0 ... Bit 7: Status I+0.7
45	Edge evaluation Bit 0: Status I+1.0 ... Bit 7: Status I+1.7
46...47	reserved

For guarantee of consistency of a µs ticker entry to the *1. edge evaluation* (FA1) the *2. edge evaluation* (FA2) serves for.

The consistency is ensured only if the appropriate bit of the *FA2* is "0". Since the last read access if more than one edge change took place, the corresponding bit of *edge lost* (FV) is set. Here the µs ticker entry contains the time of the last edge.

Example:

	Byte	+1								+0							
FA1	4	0	1	1	0	0	0	1	0	0	1	0	1	1	1	1	0
FV	8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
...																	
FA2	44	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0

The consistent µs ticker entries can be determined by logical bit operations: *FA1* AND NOT *FA2*

Result bit operation:	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0

Output address range

Byte	Bit 7 ... Bit 0
0	Control output channel (1 = set, 0 = set back) Bit 0: Status Q+0.0 ... Bit 7: Status Q+0.7
1	Control output channel (1 = set, 0 = set back) Bit 0: Status Q+1.0 ... Bit 7: Status Q+1.7

Process interrupt

Via the edge selection you may activate a process interrupt in your parameterization and define on which edge of the input signal a process interrupt should be initialized.

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 the OB 40

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

Local byte	Bit 7 ... Bit 0
8	Bit 0: Edge at I+0.0 ... Bit 7: Edge at I+0.7
9	Bit 0: Edge at I+1.0 ... Bit 7: Edge at I+1.7
10	00h (fix)
11	00h (fix)

Diagnostic interrupt

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the module.

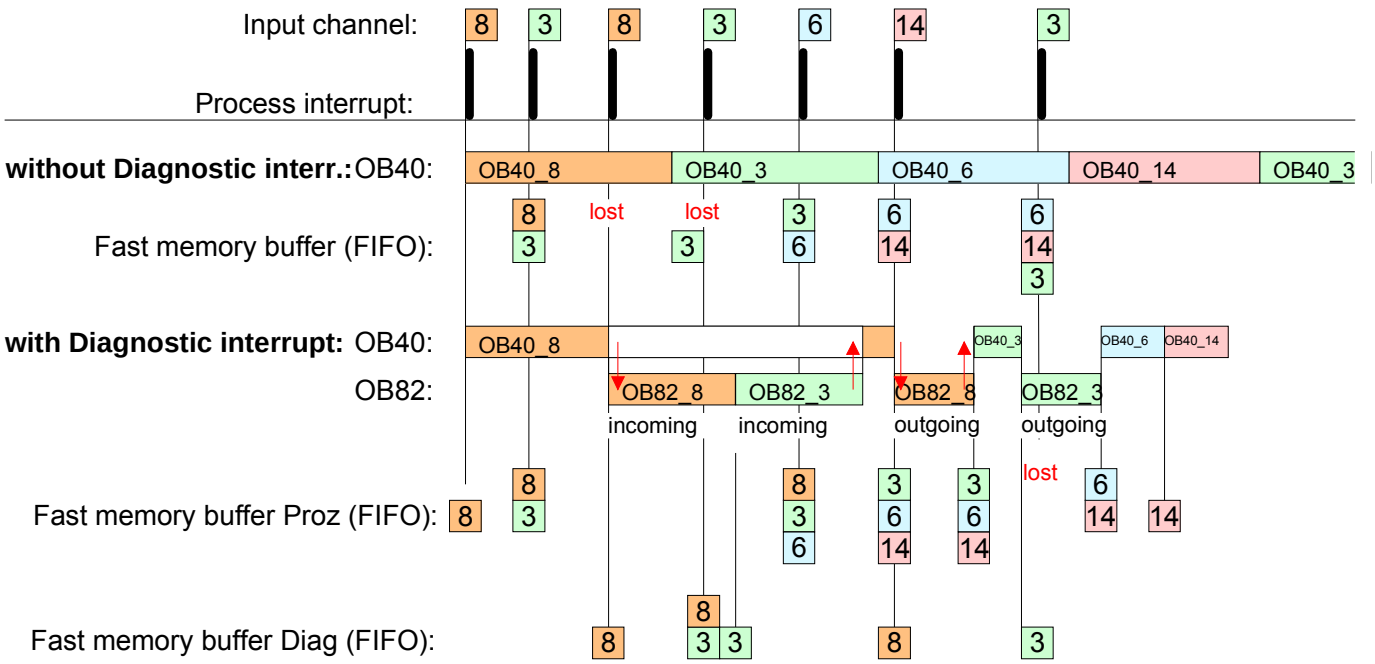
A diagnostic interrupt occurs when during a process interrupt execution in OB 40 another process interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent process interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing$_{incoming}$. 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 ... Bit 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	Bit 5 ... 0: 0 (fix) Bit 6: Process interrupt lost Bit 7: 0 (fix)

Record set 0 Diagnostic$_{outgoing}$

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

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

Diagnostic Record set 1

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Byte	Bit 7 ... Bit 0
0 ... 3	Contents record set 0 (see page before)
4	Bit 6 ... 0: channel type (here 70h) 70h: Digital input Bit 7: More channel types present 0: no 1: yes
5	Number of diagnostic bits the module puts out per channel (here 08h)
6	Number of channels of a module (here 04h)
7	Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) Bit 3: Error in channel group 3 (I+1.4 ... I+I.7) Bit 7 ... 4: 0 (fix)
8	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+0.0 Bit 1: 0 (fix) Bit 2: ... input I+0.1 Bit 3: 0 (fix) Bit 4: ... input I+0.2 Bit 5: 0 (fix) Bit 6: ... input I+0.3 Bit 7: 0 (fix)
9	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+0.4 Bit 1: 0 (fix) Bit 2: ... input I+0.5 Bit 3: 0 (fix) Bit 4: ... input I+0.6 Bit 5: 0 (fix) Bit 6: ... input I+0.7 Bit 7: 0 (fix)
10	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+1.0 Bit 1: 0 (fix) Bit 2: ... input I+1.1 Bit 3: 0 (fix) Bit 4: ... input I+1.2 Bit 5: 0 (fix) Bit 6: ... input I+1.3 Bit 7: 0 (fix)
11	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+1.4 Bit 1: 0 (fix) Bit 2: ... input I+1.5 Bit 3: 0 (fix) Bit 4: ... input I+1.6 Bit 5: 0 (fix) Bit 6: ... input I+1.7 Bit 7: 0 (fix)
12 ... 15	00h (fix)

Technical Data

Module name	VIPA 323-1BH70 DIO16xDC24V	VIPA 323-1BH70 DIO16 Alarm/ETS
Dimensions and Weight		
Dimensions WxHxD	40x125x120mm	
Weight	210g	
Data for specific module		
Number of Inputs	16	
Number of Outputs	16	
Programming specifications		
Input data	2byte	48byte
Input filter time delay	6.12µs	1µs + param. filter value
Output data	2byte	2byte
Parameter data	-	66byte
Voltages, Currents, Potentials		
Rated load voltage L+	DC 24V	
Total current of the outputs (per group)	10A	
Isolation - between channels and backplane bus	yes	
Isolation tested with	DC 500V	
Current consumption - from the SPEED-Bus	390mA	
Power dissipation of the module	4W	
Status, Interrupts, Diagnostics		
Status display	green LED per channel	
Diagnostic functions - Group error display - Supply voltage display	red F-LED per group green LED per group	
Data for selecting a sensor		
Input voltage - Rated value - for signal "1" - for signal "0"	DC 24V 15 ... 28.8V 0 ... 5V	
Input current - for signal "1"	7mA	
Connection of 2-wire-BEROs - Permitted bias current	possible 1.5mA	
Data for selecting an actuator		
Output voltage - at signal "1"	DC 24V	
Output current - at signal "1" Rated value	0.5A (max. 0.7A)	
Lamp load max.	5W per group	
Limit (internal) of the inductive circuit interruption voltage	typ. L+ (-52V)	

Chapter 4 Analog I/O modules FAST - SPEED-Bus

Overview

Contents of this chapter are the structure and the functionality of the fast analog modules for VIPA SPEED-Bus.

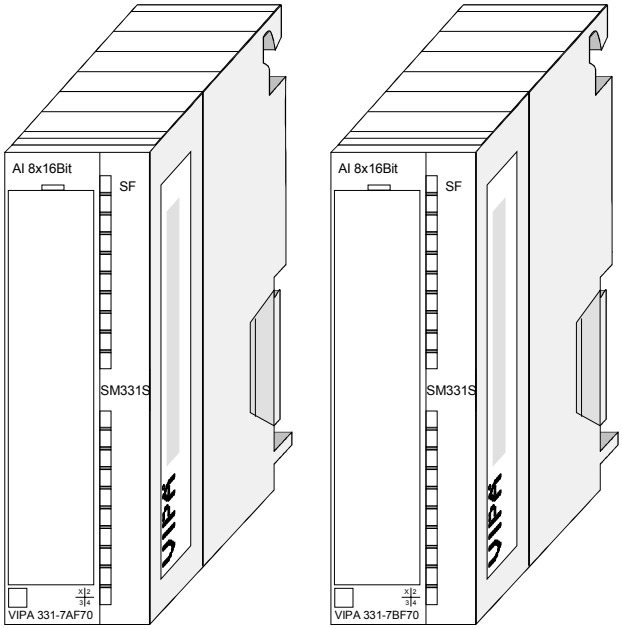
The modules may only be used at SPEED-Bus slots at the left side of the CPU.

Content

System overview

Analog input module for SPEED-Bus

- SM 331S Analog Input FAST AI 8x16Bit, ±20mA for SPEED-Bus
- SM 331S Analog Input FAST AI 8x16Bit, ±10V for SPEED-Bus



Order data analog input modules

Type	Order No.	Page
AI 8x16Bit, ±20mA	VIPA 331-7AF70	4-29
AI 8x16Bit, ±10V	VIPA 331-7BF70	4-29

Security hint

Attention!

Please regard that the modules described here do not have hardware precautions against wrong wiring. The modules are fix preset to one measuring range.

For example, the modules may get a defect if you connect a voltage at current measuring module.

General

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.

Connecting sensors

Depending on the module the following sensors may be connected to the analog input modules:

- Current sensor ±20mA
- Voltage sensor ±10V

Note!

Please take care of the correct polarity when installing the sensors! Please install short circuits at non-used inputs by connecting the positive contact with the channel ground of the according channel.

Parameterization

The modules may be parameterized by hardware configuration respectively at run time by means of SFCs.

Diagnostic functions

The modules have diagnostics capability. The following errors can release a diagnostic:

- Error in parameterization
- Process interrupt lost
- Measuring range over-/underflow
- External power supply is missing

Process interrupts

The following events can be defined by parameterization to release a process interrupt:

- Limit overflow
- Limit underflow
- End of cycle as soon as measuring value conversion of every channel has finished.

At a process interrupt 4bytes of process interrupt data are transferred.

The process interrupts are deactivated when using oscilloscope- or FIFO functions.

Analog value representation

Numeric representation in Siemens S7 format

The analog values are only processed by the CPU in binary representation. Hereby the process signals are transformed into digital format in the analog module and passed on to the CPU as word variable.

The digitized analog value is the same for input and output values at the same nominal range.

The analog value is represented as two's-complement

	Analog value															
	High byte								Low byte							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
15bit + SG	SG	Measuring value														

Sign bit (SG)

The algebraic sign bit is represented by Bit 15. Here it is essential:

Bit 15 = "0" $\rightarrow$ positive value

Bit 15 = "1" $\rightarrow$ 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.

Digital/analog conversion

In the following there are the measuring ranges listed. The support depends on the analog module.

The here listed formulas allow you to transform an evaluated measuring value (digital value) to a value assigned to the measuring range and vice versa.

+/- 10V

Voltage	Decimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	+13824	3600
+10V	+27648	6C00

Formulas for calculation:

$$Value = 27648 \cdot \frac{U}{10}, \quad U = Value \cdot \frac{10}{27648}$$

U: voltage, Value: Decimal value

+/- 20mA

Current	Decimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formulas for calculation:

$$Value = 27648 \cdot \frac{I}{20}, \quad I = Value \cdot \frac{20}{27648}$$

I: current, Value: Decimal value

Operating modes

Mode

There are the following modes at the analog input modules to be set by means of a hardware configuration at the Siemens SIMATIC Manager.

Standard mode

At the standard mode the analog values of the 8 input channels were cyclically read up to 25µs, converted to 16bit digital values and transferred to the CPU via SPEED-Bus.

Only for cycle times ≥ 200µs an end of cycle interrupt may be activated. This is generated as soon as there are new measuring values available.

Oscilloscope mode

With the oscilloscope mode the digitized input values were buffered in the memory of the module. There is space for a total of 65536 measuring values.

At this mode hardware interrupts are not supported.

Recording may be started manually or automatically, whereas there is reacted at a rising respectively falling edge of the measuring signal. As soon as the memory of the module is full the recording ends automatically.

FIFO mode

If FIFO mode is activated the input values of channel CH0 to CH7 are stored at a buffer. There is space for 8190 values each channel. These may cyclically be read in packets.

At overflow the memory contents is overwritten from the beginning and an error is reported.

SFC 193

The activation of the oscilloscope-/FIFO function as well as the readout of the stored data happens by means of the VIPA specific SFC 193.

Parameter

There are a lot of parameters to adapt these functions to your requirements. The parameters may be set by GSD file respectively at run time by SFC 58.

Addressing at SPEED-Bus

Overview

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU.

With no hardware configuration present, the CPU assigns automatically peripheral I/O addresses during boot procedure depending on the plug-in location amongst others also for plugged modules at the SPEED-Bus.

Maximal pluggable modules

In the hardware configurator from Siemens up to 8 modules per row may be parameterized. At deployment of SPEED7 CPUs up to 32 modules at the standard bus and 10 further modules at the SPEED-Bus may be controlled. CPs and DP masters that are additionally virtual configured at the standard bus are taken into the sum of 32 modules at the standard bus.

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

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 via a virtual Profibus system by including the SPEEDBUS.GSD may be used. For this, click on the properties of the according module and set the wanted address.

Automatic addressing

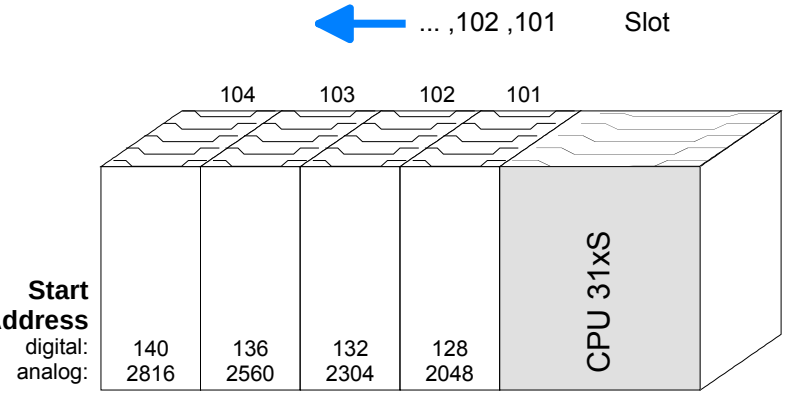
If you do not like to use a hardware configuration, an automatic addressing comes into force.

At the automatic address allocation DIOs are mapped depending on the slot location with a distance of 4byte and AIOs, FMs, CPs with a distance of 256byte.

Depending on the slot location the start address from where on the according module is stored in the address range is calculated with the following formulas:

DIOs: Start address = 4·(slot -101)+128

AIOs, FMs, CPs: Start address = 256·(slot -101)+2048



Project engineering

Overview

Every module at the SPEED-Bus including the CPU has to be configured as single "VIPA_SPEEDbus" DP slave at a virtual DP master (342-5DA02 V5.0 from Siemens). For this you have to include the GSD speedbus.gse.

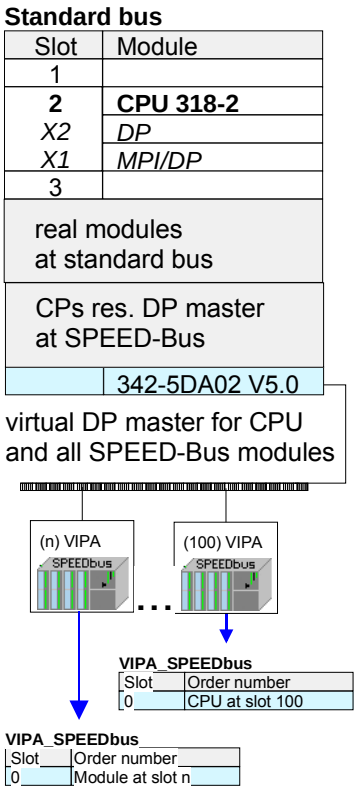
Every "VIPA_SPEEDbus" DP slave has exactly one slot for the project engineering where you must place the according SPEED-Bus module. The assignment of a SPEED-Bus slave to a SPEED-Bus slot number takes place via the Profibus address starting with 100.

Fast introduction

For the employment of the I/O modules at the SPEED-Bus the inclusion via the GSD-file from VIPA in the hardware catalog is required.

To be compatible with the Siemens SIMATIC manager, you have to execute the following steps:

- Start the hardware configurator from Siemens and include the speedbus.gse for SPEED7 from VIPA.
- Configure CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) from Siemens.
- Starting with slot 4, place the System 300 modules in the plugged sequence.
- Project engineering and connection of the SPEED-Bus-CPs res. DP master at the standard bus as virtual CP 343-1 (343-1EX11) res. CP 342-5 (342-5DA02 V5.0)
- For the SPEED-Bus you always include, connect and parameterize to the *operating mode* DP master the DP master CP 342-5 (342-5DA02 V5.0) as last module. To this master system you assign every SPEED-Bus module as VIPA_SPEEDbus slave. Here the Profibus address corresponds to the slot number beginning with 100 for the CPU. Place at slot 0 of every slave the assigned module and alter the parameters if needed.



Preconditions

The hardware configurator is part of the Siemens SIMATIC manager. It serves for project engineering. The modules that may be configured here are listed in the hardware catalog.

For the employment of the System 300S modules at the SPEED-Bus you have to include the System 300S modules into the hardware catalog via the GSD-file speedbus.gse from VIPA.

Note!

For the project engineering, a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens is required!

Include the SPEED7-GSD-file

- Browse to www.vipa.de > Service > Download > GSD- and EDS-Files > Profibus and select the file *Cx000023_Vxxx*.
- Extract the file to your work directory. The speedbus.gse is stored in the directory System_300S.
- Start the hardware configurator from Siemens.
- Close all projects.
- Select **Options** > *Install new GSD-file.*
- Change to the directory System_300S and select the **SPEEDBUS.GSE**.

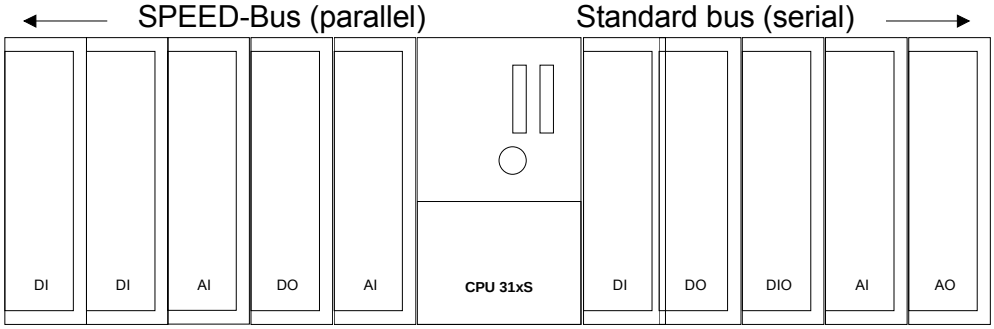
The modules of the System 300S from VIPA are now included in the hardware catalog at: *Profibus-DP / Additional field devices / I/O / VIPA_SPEEDbus*.

Steps of project engineering

The following text describes the approach of the project engineering in the hardware configurator from Siemens at an abstract sample.

The project engineering is separated into following parts:

- Project engineering of the modules at the standard bus
- Project engineering of the SPEED-Bus modules in a virtual master system (speedbus.gse required)



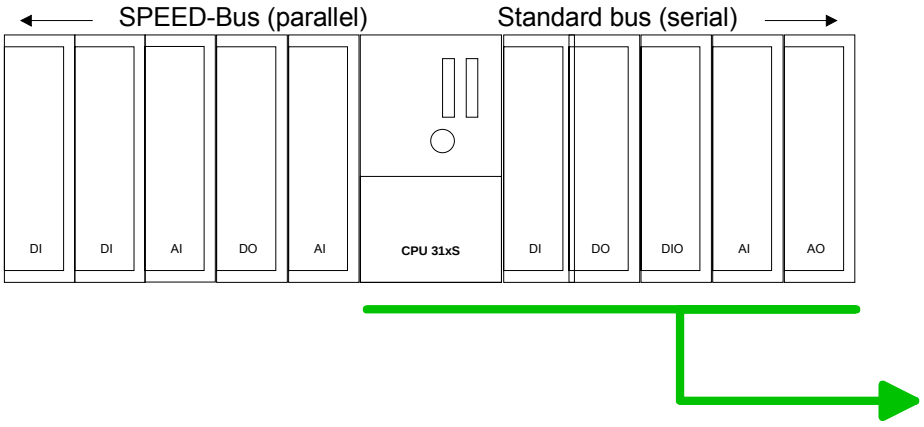
Preconditions

For the employment of the System 300S modules at the SPEED-Bus you have to include the System 300S modules into the hardware catalog via the GSD-file speedbus.gse from VIPA.

Project engineering of the modules at the standard bus

The modules at the right side of the CPU at the standard bus are configured with the following approach:

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)
- Include your System 300V modules at the standard bus in the plugged sequence starting with slot 4.
- Parameterize the CPU res. the modules where appropriate. The parameter window opens by a double click on the according module.
- To extend the bus you may use the IM 360 from Siemens where you can connect up to 3 further extension racks via the IM 361. Bus extensions are always placed at slot 3.
- Save your project.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO

Note!

To extend the bus you may use the IM 360 from Siemens where you can connect up to 3 further extension racks via the IM 361. Bus extensions are always placed at slot 3.

Project engineering of all SPEED-Bus modules in a virtual master system

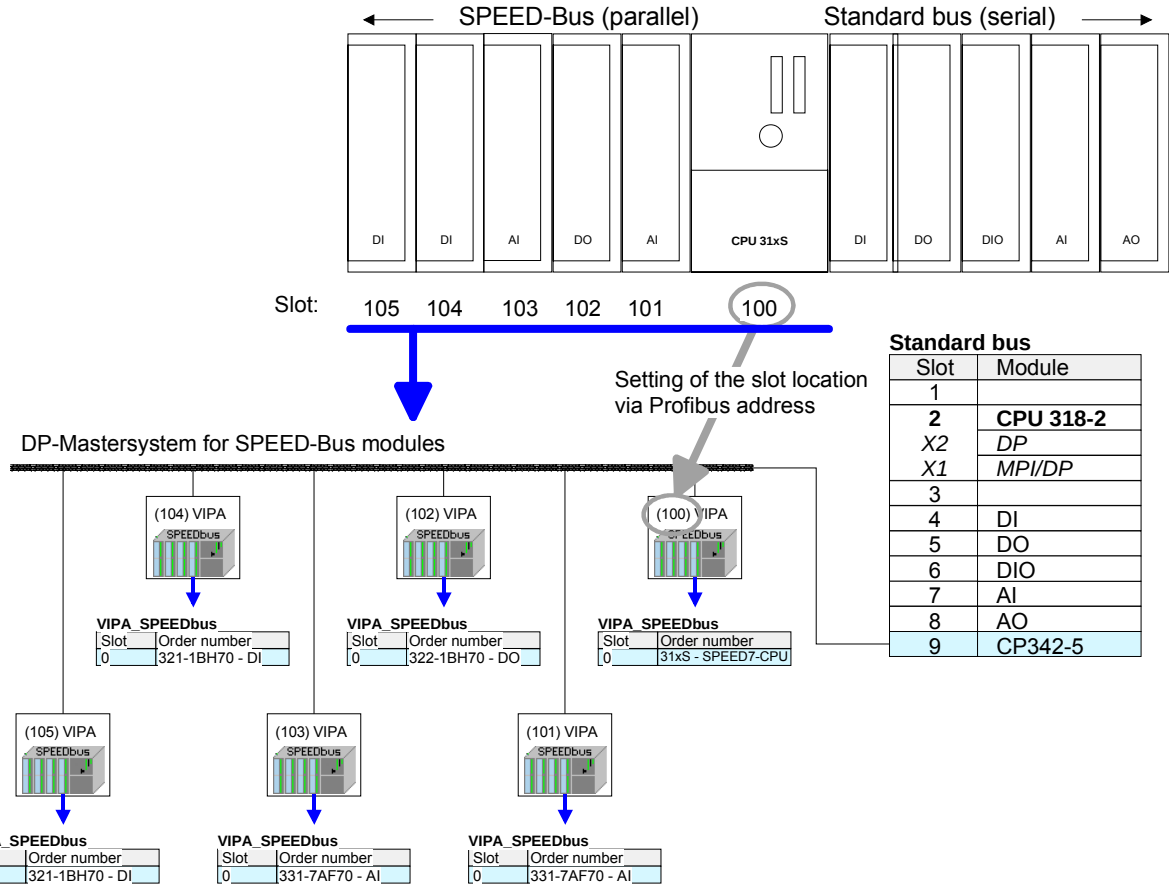
The slot assignment of the SPEED-Bus modules and the parameterization of the in-/output periphery happens via a virtual Profibus DP master system. For this, place as last module a DP master (342-5DA02 V5.0) with master system.

For the employment of the System 300S modules at the SPEED-Bus the inclusion of the System 300S modules into the hardware catalog via the GSD-file speedbus.gse from VIPA is required.

After the installation of the speedbus.gse you may locate under *Profibus DP* / *Additional field devices* / *I/O* / *VIPA_SPEEDbus* the DP slave system vipa_speedbus.

Now include for the CPU and every module at the SPEED-Bus a slave system "vipa_speedbus".

Set as Profibus address the slot no. (100...110) of the module and place the according module from the hardware catalog of VIPA_speedbus to slot 0 of the slave system.



The according module is to be taken over from the HW Catalog of vipa_speedbus to slot 0.

Parameterization

Overview

After Power ON the diagnostics function of every channel is deactivated. For parameterization the parameter data of the module are transferred by the Siemens SIMATIC manager to the CPU.

There is also the possibility to change parameters during run time by means of SFCs.

Place module

- Start the hardware configurator and install speedbus.gse for SPEED7 from VIPA.
- Configure CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) from Siemens.
- Include your System 300V modules at the standard bus in the plugged sequence starting with slot 4.
- For the SPEED-Bus you always include, connect and parameterize to the *operating mode* DP master the DP master CP 342-5 (342-5DA02 V5.0) as last module.
- To this master system you assign every SPEED-Bus module as VIPA_SPEEDbus slave. Here the Profibus address corresponds to the slot number beginning with 100 for the CPU.
- Place at slot 0 of every slave the assigned module and alter the parameters if needed. In this way also the project engineering of the analog modules takes place.

Parameterize the module

Via double click on the wanted module in the hardware configurator the corresponding parameter dialog is opened.

You may alter the following parameters there:

- Start address of the data of the module stored in the CPU
- Enable interrupt / Mode
  (end of cycle, oscilloscope-/FIFO mode, diagnostics, limit)
- Limit (upper/lower)
- Oscilloscope parameter (channel, pre-trigger, level, condition)
- Cycle time (scan time at oscilloscope-/FIFO mode)

Save and transfer project

- Save and compile your project.
- Set your CPU to STOP.
- Transfer your project into the CPU.

As soon as you switch the CPU into RUN, the parameters are transmitted to the analog input module.

More detailed information about the parameters may be found at the following pages.

Structure of the parameter bytes

The parameterization happens during hardware configuration. Here the following parameter data are transferred:

Length in Byte	Record set	Description
4	A0h	Limit upper/lower channel 0
4	A1h	Limit upper/lower channel 1
...	...	...
4	A7h	Limit upper/lower channel 7
2	A8h	Cycle time / (sampling time at oscilloscope-/FIFO mode)
2	7Fh	Interrupt enable / Operating mode
5	BEh	Oscilloscope (Parameter for oscilloscope mode)

Using the SFCs 55, 56, 57 and 58 every parameter of the module may be transferred to the module during run time.

Here the favored parameters are transferred as record set by the user program by means of SFCs.

By this parameters may be transferred, which are not supported by the Siemens SIMATIC manager.

Record set A0...A7h
Limit upper/lower

Upper and *lower limits* may be set for the corresponding channel by record set A0h...A7h. As soon as your measured value leaves the work area defined by the limit values, a limit value interrupt is released, if activated.

The record set has the following structure:

Word		Default	
		Byte 0	Byte 1
0	Limit upper	7FFFh	
2	Limit lower	8000h	

Record set A8h
Cycle time / Sampling time

With this record set a factor may be set, which sets the cycle time multiplied by 100µs, this is independent of the number of activated channels. The cycle time of 25µs is set by 0.

During hardware configuration the cycle time may be directly chosen.

Is oscilloscope respectively FIFO mode activated this time represents the sampling time the read values are stored.

Range of values: 0 ... 600

The record set has the following structure:

Word		Default	
		Byte 0	Byte 1
0	Cycle time / sampling time	0001h	

As soon as this record set is transferred during recording at oscilloscope or FIFO operation the recording is stopped.

Record set 7Fh
Interrupt enable / Operating mode

Here the interrupt behavior and the operating mode of the module may be adjusted. Is the diagnostic interrupt deactivated during run-time and a diagnostic interrupt is just pending, there may no $diagnostic_{going}$ be generated to reset the SF-LED. Please do not execute a diagnostic interrupt deactivation during run time!

As soon as this record set is transferred during recording at oscilloscope or FIFO operation the recording is stopped.

The record set has the following structure:

Byte	Bit 7 ... Bit 0	Default
0	Interrupt enable / Operating mode Bit 0: reserved Bit 5 ... 1: Operating mode 0 0000: without end of cycle interrupt 0 0010: with end of cycle interrupt 0 0100: Oscilloscope: Channel 0 0 1000: Oscilloscope: Channels 0 ... 1 0 1100: Oscilloscope: Channels 0 ... 3 1 0000: Oscilloscope: Channels 0 ... 7 0 0001: FIFO mode Bit 6: Diagnostic interrupt enable Bit 7: reserved	00h
1	Limit interrupt enable Bit 0: Channel 0 ... Bit 7: Channel 7	00h

with/without end of cycle interrupt

Setting *with* or *without end of cycle interrupt* the module may be used in standard operating mode. Here the 8 channels are read synchronously and allocated as 16bit value.

Setting *with end of cycle interrupt* an end of cycle interrupt is generated as soon as new measuring values are available. Please note that end of cycle monitoring is only available starting from a module cycle time of 200µs.

Oscilloscope operating mode

In the oscilloscope mode the fragmentation of the memory is configured by number of channels to be recorded. The memory has a total space for 65536 measuring values. For memory fragmentation see the following table:

Byte 0, Bit 5 ... 1	Operating mode	Channel	Number of words	Values each channel
0 0100	Oscilloscope: Channel 0	CH0	1 x 64 k	65.536
0 1000	Oscilloscope: Ch. 0 ... 1	CH0, CH1	2 x 32 k	32.768
0 1100	Oscilloscope: Ch. 0 ... 3	CH0 … CH3	4 x 16 k	16.384
1 0000	Oscilloscope: Ch. 0 ... 7	CH0 … CH7	8 x 8 k	8.192

FIFO operating mode

During FIFO operation all of the 8 channels are recorded and stored at a buffer. These values may be read as packets by means of the user program. At overflow the memory contents is overwritten from the beginning and an error is reported by *RETVAL*.

The buffer offers place for 8190 values per channel.

Diagnostic interrupt enable

With activated diagnostic interrupt, in the case of an error and after error correction a diagnostic interrupt is released to the CPU.

With a diagnostic interrupt the CPU interrupts its user program and jumps to OB 82. There detailed diagnostic information can be requested by means of the SFC 51 respectively SFC 59. The diagnostics data are consistent during OB 82 operation.

Limit interrupt enable

A work area may be defined by the parameters *limit upper/lower*.
If your measuring signal leaves this work area and the limit interrupt is enabled, then the module releases a process interrupt of the corresponding channel.

Here the CPU interrupts its user program and jumps to OB 40. There it may be reacted accordingly to the process interrupt. With leaving the OB 40 the process interrupt is acknowledged at the corresponding module.

Please note that at oscilloscope-/FIFO operating mode the process interrupts are not supported.

Record set BEh
Oscilloscope

The parameters of the oscilloscope operation may be set with this record set.

As soon as this record set is transferred during recording at oscilloscope or FIFO operation the recording is stopped.

The record set has the following structure:

Byte	Bit 7 ... 0	Default
0	Bit 2 ... 0: Trigger channel 000: CH0 ... 111: CH7 Bit 7 ... 3: reserved	00h
1	Bit 6 ... 0: Pre-trigger (%) 00h: 0% Pre-trigger ... 64h: 100% Pre-trigger Bit 7: reserved	00h
2, 3	Bit 15 … 0: Trigger level 8100h: -32512 (decimal) ... 0000h: 0 ... 7EFFh: 32511 (decimal)	00h
4	Bit 1 ... 0: Trigger condition 00: rising edge (automatic start) 01: falling edge (automatic start) 10: manual start Bit 7 ... 2: reserved	00h

Trigger channel

With this parameter a channel may be defined to be triggered, this means the recording is to be started on its rising or falling edge. At manual operation this setting will be ignored.

Pre-trigger (%)

Here a number per cent may be set as pre-trigger. On this way values may also be picked out, which were buffered before the trigger event occurred. At manual operation this setting will be ignored.

In the following there is a formula to calculate the number n of values in the buffer, which were buffered before the trigger event occurred. This value depends on the parameters pre-trigger and the number of channels to be buffered.

$$n = \underbrace{\frac{8192 \times \text{Pre-trigger}}{100}}_{\text{Integer}} \times \frac{8}{\text{Number channels}}$$

Trigger level

At this parameter a threshold may be set, which when exceeded/fallen below generates a trigger event.
At manual operation this setting will be ignored.

Trigger condition

Here the start condition for recording may be set. To start the recording there is basically differentiated between an automatic operation with a triggered edge and a manual operation.

SFC 193 for controlling

The oscilloscope/FIFO operation is controlled by means of the SFC 193. This is described at the following pages.

SFC 193 - Oscilloscope-/FIFO function

Description

The SFC 193 serves for controlling the oscilloscope-/FIFO function.
It allows to start the recording and to read the buffered data. Depending upon the parameterization there are the following possibilities:

Oscilloscope operation

- Depending on the trigger condition at edge evaluation the monitoring of the configured channel may be started respectively at manual operation the recording may be started.
- The recorded measuring values may be accessed by the SFC 193 as soon as the buffer is full.

FIFO operation

- Start the recording
- Read the puffer at any time

Note!

The SFC may only be called from on level of priority e.g. only from OB 1 or OB 35.

The module is to be parameterized before.

For starting and reading in each case the SCF 193 is to be called. The differentiation of both variants takes place in the parameter *MODE.*

Parameter

Parameter	Declaration	Data type	Function depending on MODE
REQ	IN	BOOL	Execute function (start/read)
LADR	IN	WORD	Base address of the module
MODE	IN	WORD	Mode (start/read)
CHANNEL	IN	BYTE	Channel to be read
OFFSET	IN	DWORD	Address offset for reading (not FIFO operation)
RECORD	IN	ANY	Memory for the read data
RETVAL	OUT	WORD	Return value (0=OK)
BUSY	OUT	BOOL	Function is busy
TIMESTAMP	OUT	DWORD	Time stamp (only at edge evaluation)
LEN	INOUT	DWORD	Number of values to be handled per channel

REQ

Depending on the set MODE when the bit is set the recording respectively the reading may be started.

Depending on the trigger condition at edge evaluation the monitoring of the configured channel may be started respectively at manual operation the recording may be started.

The data are read from the module, if "read" is set at *MODE.*

LADR

Logical basic address of the module

MODE

The SFC 193 may be called with 3 different modes. The corresponding mode may be set by the parameter *MODE*. The configured mode is executed by setting *REQ*.

The following values are supported:

01h: Starts recording respectively edge monitoring depending upon the parameterization.

00h: Read data within several cycles until BUSY = 0.

80h: Read data with one access.

CHANNEL

Here the channel is specified to be read. With each call one channel may be read. This parameter is irrelevant at start calls with *MODE* = 01h.

OFFSET

Offset specifies an address offset for the reading process. By this you get access to sub-ranges of the recorded data. The value for the maximum offset depends on the number of values, which were recorded per channel.

OFFSET is not supported in FIFO operation. It will be ignored.

RECORD

Here an area for the read values to be stored at may be defined.

In FIFO operation every value of the selected channel may be read, which were stored up to the time of start reading. Please regard that the buffer has a sufficient size for the data to be buffered, otherwise an error is reported.

BUSY

BUSY = 1 indicates that the function just processed. BUSY = 0 indicates that the function is finished.

TIMESTAMP

There is an internal clock with a resolution of 1µs running in every SPEED-Bus module. The returned value corresponds to the time at the SPEED-Bus module, on which the trigger event occurred.

TIMESTAMP is only available at the edge triggered oscilloscope operation. It is valid as long as the job is running (*RETVAL* = 7xxxh) and bit 4 of byte 0 is set respectively the job has been finished without an error (*RETVAL* = 0000h).

LEN

The length parameter realized as IN/OUT is variably interpreted depending on the selected mode at the function call.

Mode: start (MODE: = 01h)

At *MODE* = 01h this parameter may only be used at the manual oscilloscope start. Here the requested number of values per channel to be buffered may be assigned. In this mode there is no value reported by *LEN*.

Mode: read (MODE: = 00h or 80h)

At *MODE* = 00h respectively 80h the number of values to be read may be set. This parameter is ignored in FIFO operation. The number of the read values is returned by LEN.

RETVAL (Return value)

In addition to the module specific error codes listed here, there general SFC error information may be returned as well.
More may be found at the operation list.

RETVAL	Description depending on the BUSY-Bit	BUSY
Byte		
0	Bit 1, 0:	
	00: Call with REQ: = 0 (idle, waiting for REQ = 1)	0
	01: First call with REQ: = 1	1
	10: Subsequent call with REQ: = 1	1
	11: Oscilloscope is just recording	1
	Bit 2: REQ: = 1, but recording was not yet started. (MODE: = 00h or MODE: = 80h)	0
	Bit 3: reserved	-
	Bit 4: Trigger event occurred and recording is just running.	1
	Bit 5: Waiting for trigger event	1
	Bit 7…6: reserved	-
1	Bit 0: reserved	-
	Bit 1: The number of recorded values exceeds the target area defined by RECORD (in words)	0
	Bit 2: The number of the recorded values exceeds the area defined by *LEN* and *OFFSET*.	0
	Bit 3: Buffer overflow in FIFO operation.	0
	Bit 7...4:	
	0000: Job finished without an error	0
	0111: Job still running	1
	1000: Job finished with error (see following table)	0

Job finished without an error

RETVAL	Description depending on the BUSY-Bit	BUSY
0000h	Job was finished without an error.	0

Job finished with error

RETVAL	Description depending on the BUSY-Bit	BUSY
8002h:	Oscilloscope-/FIFO function is not configured.	0
8003h:	An internal error occurred - please contact VIPA.	0
8005h:	The selected channel may not be read - wrong channel number.	0
8007h:	The value at *OFFSET* exceeds the number of recorded values.	0
8090h:	There is no SPEED-Bus module with this address available.	0
80D2h:	LADR exceeds the peripheral address area.	0

Example for the oscilloscope function

Job definition

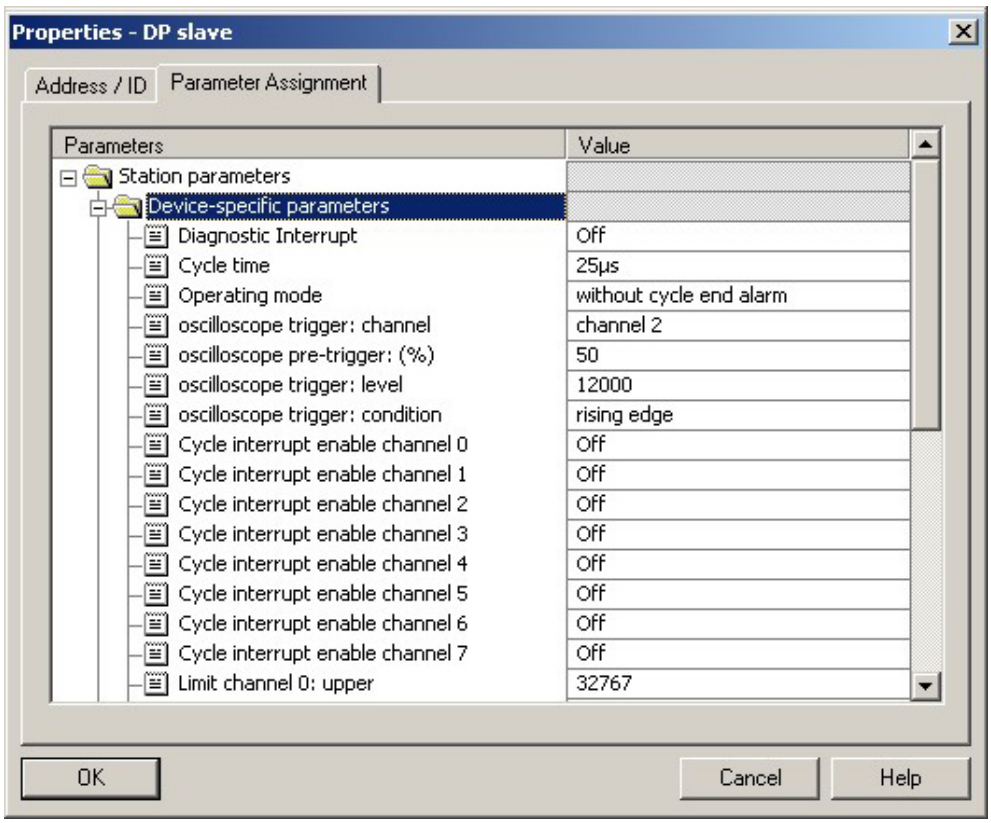
At this example 4 channels were recorded with 25µs sampling time whereas channel 2 is monitored. As soon as the decimal value 12000 is exceeded by the input signal, a trigger event is generated. 50% of the buffer should contain the last values before the trigger event occurred (pre-trigger) and 50% the values after the event.

Parameterization

The parameterization happens by a hardware configuration of the Siemens SIMATIC manager. Here the integration of the VIPA GSD file speedbus.gse is necessary. More may be found above at "Project engineering".

Parameterize the following module parameters after configuring the system:

Cycle time:	25µs
Operating mode:	Oscilloscope Channels 0 … 3 (16384 values per channel)
Oscilloscope trigger channel:	2
Oscilloscope pre-trigger (%):	50
Oscilloscope trigger level:	12000
Oscilloscope trigger condition:	rising edge



User program

The SFC 193 calls for starting the oscilloscope recording and for reading the data are implemented in the OB 1 of the user program.

For the simplified representation and for controlling the parameters are handled in a variable table.

```
CALL  SFC  193      //  start oscilloscope function (1. SFC call)
 REQ      :=M99.0          //  bit to start recording
 LADR     :=W#16#64        //  module address at SPEED-Bus
 MODE     :=W#16#1         //  mode: start
 CHANNEL  :=B#16#0         //  not used
 OFFSET   :=DW#16#0        //  not used
 RECORD   :=DB10           //  not used
 RETVAL   :=MW1110         //  return value
 BUSY     :=M112.0         //  busy bit
 TIMESTAMP:=MD100          //  not used
 LEN      :=MD114          //  length parameter for recording
                           //  (only at "manual start")
U     M     99.0    //  request bit set by 1. call?
S     M     98.1    //    yes:   set request bit for 2. call
                    //
R     M     99.0    //           Reset request bit for 1. call
                    //
CALL  SFC  193      //  read data (2. SFC call)
 REQ      :=M98.1          //  bit for reading the data
 LADR     :=W#16#64        //  module address at SPEED-Bus
 MODE     :=W#16#80        //  mode: read (complete, 1 access)
 CHANNEL  :=MB148          //  channel to be read
 OFFSET   :=MD150          //  address offset for reading
 RECORD   :=DB10           //  data block for the read values
 RETVAL   :=MW110          //  return value
 BUSY     :=M112.0         //  busy bit
 TIMESTAMP:=MD104          //  timestamp at trigger event
 LEN      :=MD114          //  length parameter for reading
U     M     98.1    //  request bit set and
U     M    112.0    //  busy bit set?
BEB                 //    yes:   reading is not yet finished
                    //           finish block
U     M     98.1    //  request bit is set and
UN    M    112.0    //  busy bit is not set?
SPBN  end           //    no:    jump to label end

L     MW   110      //    yes:   load return value and
T     MW   160      //           transfer to flag
end:  NOP  0
U     M     98.1
R     M     98.1    //  reset request bit from 2. call
```

Process

The recording at oscilloscope operation is started by setting flag 99.0. From this moment on the configured monitoring of channel 2 on the rising edge and the threshold of 12000 begins. With the configured operation mode *Oscilloscope channels 0...3* these channels are recorded, 16384 values each channel.

Exceeding the configured threshold 12000 a trigger event is released. With the pre-trigger of 50% 8192 values per channel were finally recorded, then the recording is finished and the *BUSY* bit is reset.

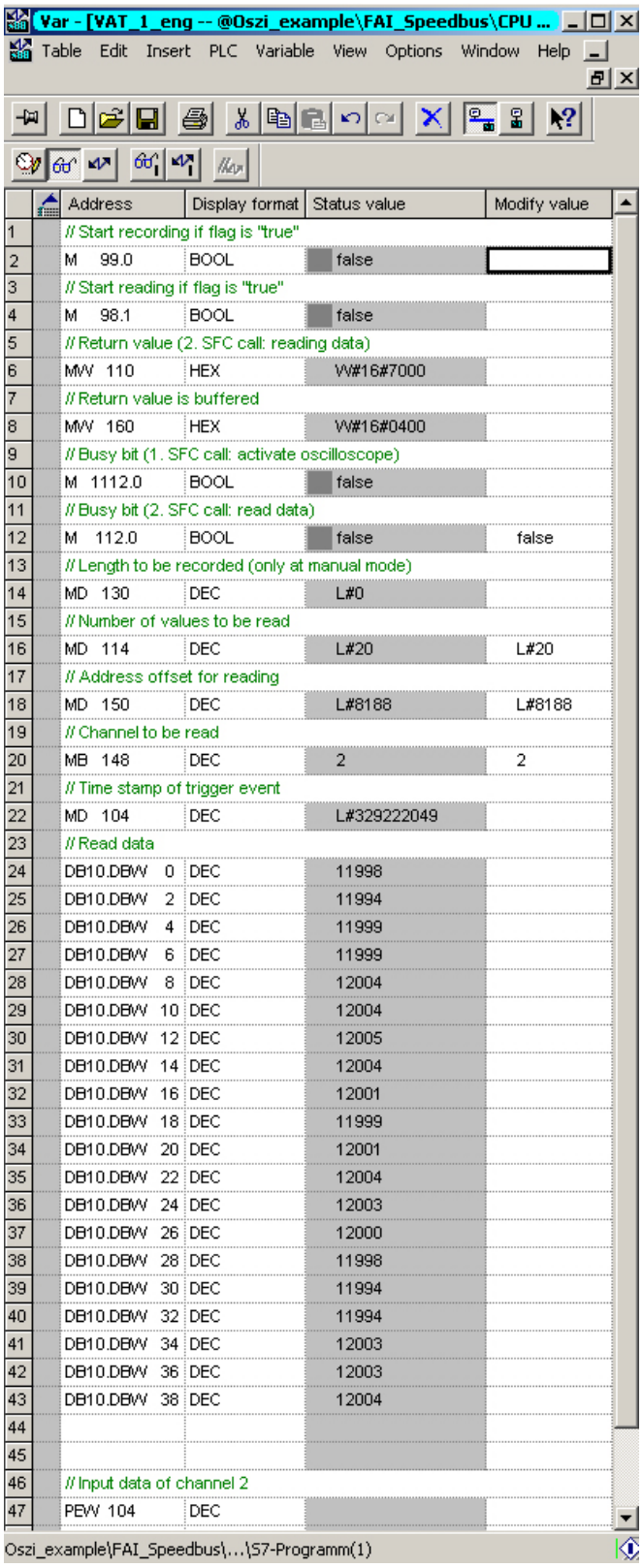
Now the data may be read. With the configured pre-trigger of 50% the 8193. value is the value, which released the event.

Further reading accesses with e.g. other address offsets or to read values of the other channels may be executed by setting flag 98.1. The oscilloscope recording may be started again by setting flag 99.0.

Variable table

The output of the values 8189 ... 8208 is generated by the address offset of 8188 and the length of 20.

The event was released by the 8193. value (DB10.DBW 8 = 12004), because it has exceeded the configured threshold of 12000.



Example for the FIFO function

Job definition

At this example the recorded values of channel 0 were read and the minimum and maximum input value is evaluated.

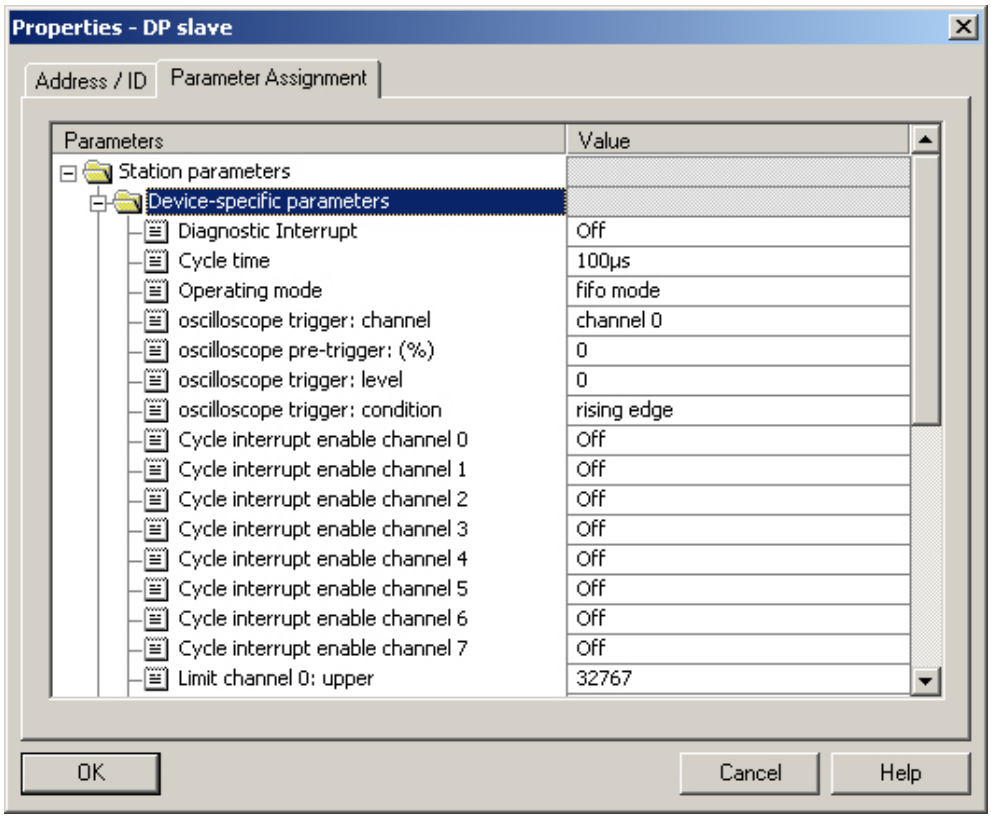
Parameterization

The parameterization happens by a hardware configuration of the Siemens SIMATIC manager. Here the integration of the VIPA GSD file speedbus.gse is necessary. More may be found above at "Project engineering".

Parameterize the following module parameters after configuring the system:

Cycle time:	100µs
Operating mode:	FIFO

The oscilloscope parameters (channel, pre-trigger, level, condition) are not necessary for FIFO operation and were ignored.



User program

The SFC 193 calls for starting the FIFO operation and for reading the data are implemented in the OB 35. The OB 35 is to be parameterized that it is cyclically called for operation every 10ms. For the simplified representation and for controlling the parameters are handled in a variable table.

```
        UN    M     20.0    //  start bit set?
        SPB   go            //     no:   do not start FIFO function
        L     0             //     yes:  initialize limits
                            //           and start FIFO function
        L     32767
        T     MW    46      //           initialize minimum value
        L     -32768
        T     MW    48      //           initialize maximum value
        CALL  SFC  193      //  start FIFO function (1. SFC call)
         REQ      :=TRUE           //  bit starting the FIFO function
         LADR     :=W#16#64        //  module address at SPEED-Bus
         MODE     :=W#16#1         //  mode: start
         CHANNEL  :=B#16#0         //  not used
         OFFSET   :=DW#16#0        //  not used
         RECORD   :=DB1            //  not used
         RETVAL   :=MW22           //  return value
         BUSY     :=M20.1          //  busy bit
         TIMESTAMP:=MD24           //  not used
         LEN      :=MD28           //  not used
        U     M     20.0          //  reset start bit
        R     M     20.0
        S     M     20.2    //  start reading if FIFO active
go:     UN    M     20.2    //  bit set for reading?
        BEB                 //     no:   finish block
        CALL  SFC  193      //  read data (2. SFC call)
         REQ      :=TRUE           //  bit for reading the data
         LADR     :=W#16#64        //  module address at SPEED-Bus
         MODE     :=W#16#80        //  mode: read (complete, 1 access)
         CHANNEL  :=MB21           //  channel to be read
         OFFSET   :=DW#16#0        //  not used
         RECORD   :=DB1            //  data block for the read values
         RETVAL   :=MW32           //  return value
         BUSY     :=M20.3          //  busy bit
         TIMESTAMP:=MD34           //  not used
         LEN      :=MD38           //  length parameter for reading

        L     MD    38      //  load length parameter
        L     0
        ==D                 //  check if values were read
        BEB                 //     no:   finish block
                            //  every value in the buffer was read
                            //  and stored in DB 1
        L     P#0.0         //  set pointer to the 1. value of DB 1
        T     MD    42      //  store pointer in flag
        AUF   DB     1      //  open DB 1

        // Check for new minimum:
loop:   L     DBW [MD  42]  //  load input value from DB
        L     MW    46      //  load previous minimum value
        >=I                 //  is the input value exceeding
                            //  the previous minimum?
        SPB   max           //     yes:  check for maximum
        TAK                 //     no:   exchange accu1 and accu2 -
                            //           then the measuring value is in
                            //           accu1 again
        T     MW    46      //           store new minimum in flag
```

continued ...

... continue

```
        //Check for a new maximum:
max:    L     DBW [MD    42] //  load input value from DB
        L     MW    48       //  load previous maximum
        <=I                  //  is the input value less the
                             //  previous maximum?
        SPB   ex             //     yes: next value
        TAK                  //     no:  exchange accu1 and accu2 -
                             //          then the measuring value is in
                             //          in accu1 again
        T     MW    48       //          store new maximum in flag

ex:     NOP   0
        //Set pointer to the next value in DB 1:
        L     MD    42       //  load pointer from flag
        L     P#2.0          //  2bytes because the input values were
                             //  stored as words in the DB 1
        +D                   //  increment pointer
        T     MD    42       //  store pointer in flag

        //Compare pointer with length of read data:
        SRD   4
        L     MD    38       //  number of read values
        <D                   //  Does the pointer point to a valid field
                             //  in the DB 1?
        SPB   loop           //     yes: check next value
```

Process

The recording at FIFO operation is started by setting flag 20.0. From this moment on the whole buffered input values of channel 0 were every 10ms cyclically read and stored in the data block.

The evaluation for minimum and maximum is executed in a loop. Here the number of read values and so the number of necessary loop operations is represented by the parameter *LEN.*

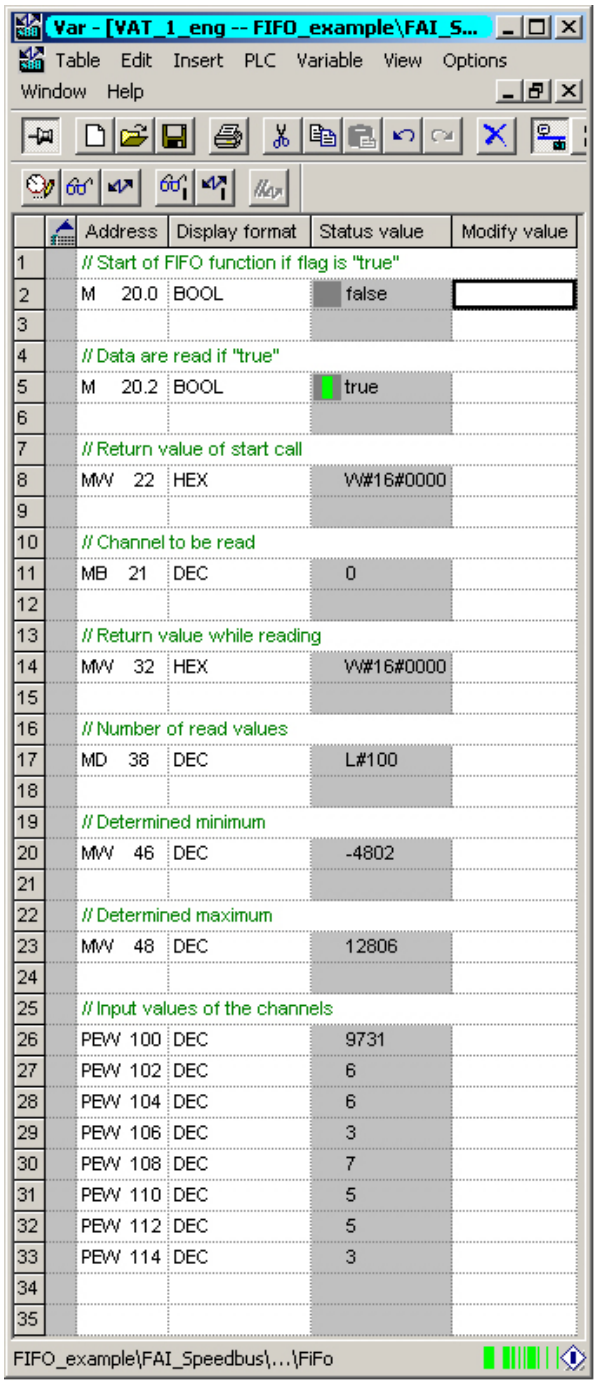
After evaluation of the whole read data the OB 35 is finished.

Variable table

The recording at FIFO operation is started by setting flag 20.0. The cyclic read access is indicated by flag 20.2.

The channel to be read may be defined by flag 21.

At a cyclic read access every 10ms and a sample time of 100µs about 100 values may be read from the buffer. The number of read values is reported in flag 38. The minimum respectively maximum value may be found in the flag word 46 respectively 48.



	Address	Display format	Status value	Modify value
1	// Start of FIFO function if flag is "true"			
2	M 20.0	BOOL	false	
3				
4	// Data are read if "true"			
5	M 20.2	BOOL	true	
6				
7	// Return value of start call			
8	MW 22	HEX	W#16#0000	
9				
10	// Channel to be read			
11	MB 21	DEC	0	
12				
13	// Return value while reading			
14	MW 32	HEX	W#16#0000	
15				
16	// Number of read values			
17	MD 38	DEC	L#100	
18				
19	// Determined minimum			
20	MW 46	DEC	-4802	
21				
22	// Determined maximum			
23	MW 48	DEC	12806	
24				
25	// Input values of the channels			
26	PEW 100	DEC	9731	
27	PEW 102	DEC	6	
28	PEW 104	DEC	6	
29	PEW 106	DEC	3	
30	PEW 108	DEC	7	
31	PEW 110	DEC	5	
32	PEW 112	DEC	5	
33	PEW 114	DEC	3	
34				
35				

Diagnostics

Overview

A diagnostic is an error message to a superordinated system (CPU). If enabled by parameterization the following events can release a diagnostic interrupt:

- Error in parameterization
- Process interrupt lost
- Measuring range over-/underflow
- External power supply is missing

At a diagnostic interrupt the CPU interrupts the user application and jumps to the OB 82. Within this OB you can accordingly react to the requested diagnostics information of the module.

In the case of an error $diagnostic_{comming}$ and with correction $diagnostic_{going}$ is released.

Error indication via measuring value and LEDs

The module sends the measuring value 7FFFh at overflow, when recognizing a parameterization error or power supply is missing and 8000h at underflow.

The group error LED (SF) indicates an error, if the diagnostics interrupt is activated.

Evaluating the diagnostics

At a diagnostics event the CPU interrupts the user program and jumps into the OB 82. This OB allows you via according programming to request detailed diagnostic information with record set 0 and 1 by means of the SFCs 51 and 59 and react to it.

After processing of the OB 82, the processing of the user application is continued. The diagnostic data are consistent until leaving the OB 82.

As soon as you have enabled the diagnostic interrupt, *record set 0* is transferred to the superordinated system in cause of an error. The *record set 0* has a fixed content and a length of 4byte. The content of *record set 0* may be monitored in plain text in the diagnosis window of the CPU.

For extended diagnostics during run time, you may also evaluate the *record set 1* of 16byte length via SFCs 51 and 59.

Record set 0 and 1 have the following structure:

Diagnostics record set 0

Record set 0 (Byte 0 to 3):

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Error in module Bit 1: Internal error Bit 2: External error Bit 3: Channel error Bit 4: External power supply is missing Bit 6, 5: reserved Bit 7: Wrong parameters in module	00h
1	Bit 3 ... 0: Module class 0101 Analog module Bit 4: Channel information present Bit 7 ... 5: reserved	15h
2	reserved	00h
3	Bit 5 ... 0: reserved Bit 6: Process interrupt lost Bit 7: reserved	00h

Diagnostics record set 1

Byte 0 to 15:

The record set 1 contains the 4byte of record set 0 and additionally 12byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Record set 1 (Byte 0 to 15):

Byte	Bit 7 ... Bit 0	Default
0 ... 3	Content record set 0 (see previous page)	-
4	Bit 6 ... 0: Channel type 70h: Digital input 71h: Analog input 72h: Digital output 73h: Analog output 74h: Analog in-/output Bit 7: More channel types present 0: no 1: yes	71h
5	Bit 7 ... 0: Number of diagnostic bits, that the module throws per channel	08h
6	Bit 7 ... 0: Number of similar channels of a module	08h
7	Bit 0: Channel error Channel 0 ... Bit 7: Channel error Channel 7	00h
8	Bit 0: Project engineering/Parameterization error Channel 0 Bit 5 ... 1: reserved Bit 6: Underflow Channel 0 Bit 7: Overflow Channel 0	00h
...	...	
15	Bit 0: Project engineering/Parameterization error Channel 7 Bit 5 ... 1: reserved Bit 6: Underflow Channel 7 Bit 7: Overflow Channel 7	00h

Process interrupts

When a process interrupt occurs, the CPU interrupts the user application and jumps to OB 40. Within the OB 40 there is the possibility to get the basic address of the module, which released the process interrupt by means of the local word 6.

At the operation mode oscilloscope-/FIFO the process interrupts are deactivated.

Activator

The following releases for a process interrupt may be defined during parameterization:

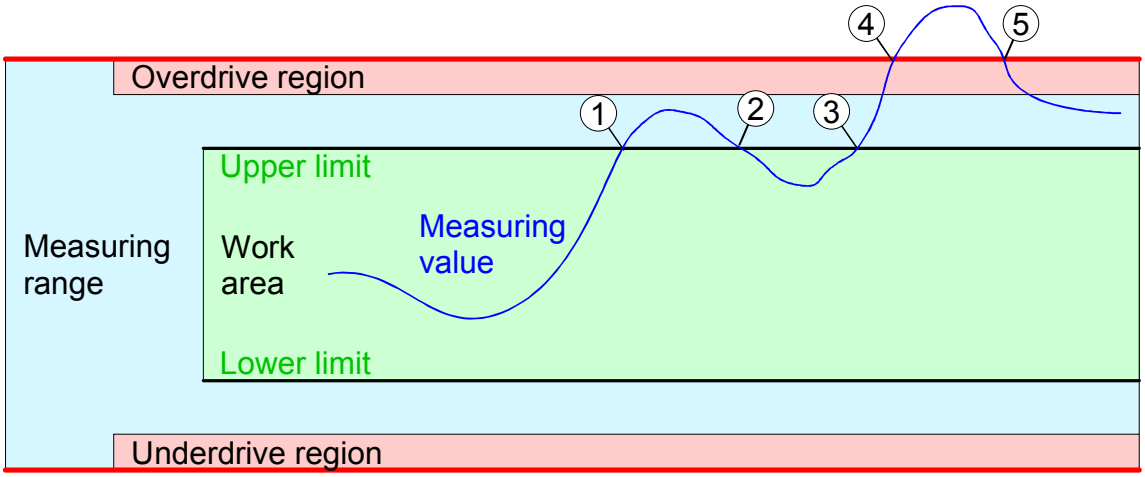
- Limit overflow
- Limit underflow
- End of cycle as soon as measuring value conversion of every channel has finished.

Interrupt data

The interrupt data of the module may be accessed by local double word 8. The local double word 8 has the following structure:

Local double word 8	Bit 7...0
Byte 0	Upper limit overflow Bit 0: Channel 0 . . Bit 7: Channel 7
Byte 1	Lower limit underflow Bit 0: Channel 0 . . Bit 7: Channel 7
Byte 2	Event end of cycle Bit 2...0: reserved Bit 3: End of cycle reached Bit 7...4: reserved
Byte 3	reserved

In the following illustration the interrupt behavior during limit exceedance is graphically represented:



(1) Process interrupt limit exceedance
(2) no reaction
(3) Process interrupt limit exceedance
(4) Diagnostic interrupt$_{comming}$ overflow channel
(5) Diagnostic interrupt$_{going}$ overflow channel

331-7xF70 - AI 8x16Bit

Order data

AI 8x16Bit, ±20mA	VIPA 331-7AF70
AI 8x16Bit, ±10V	VIPA 331-7BF70

Description

The analog input modules transform analog signals from the process into digital signals for the internal processing. The modules are preset to one measuring range.

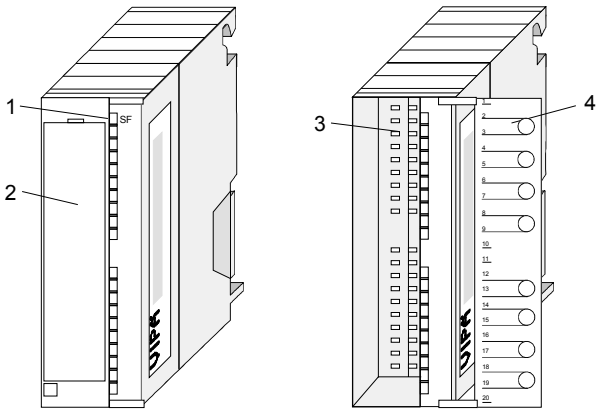
Properties

- 8 inputs
- Oscilloscope-/FIFO-Function parameterizable
- The 8 inputs are read synchronous
- Measuring value resolution 15Bit + sign
- Suitable for sensors
  ±20mA (VIPA 331-7AF70)
  ±10V (VIPA 331-7BF70)
- Parameterizable diagnostic and process interrupt
- Isolated to the backplane bus and between the channels

Default settings

After Power ON the diagnostics function of every channel is deactivated.

Structure



[1] LEDs

[2] flap with labeling strip

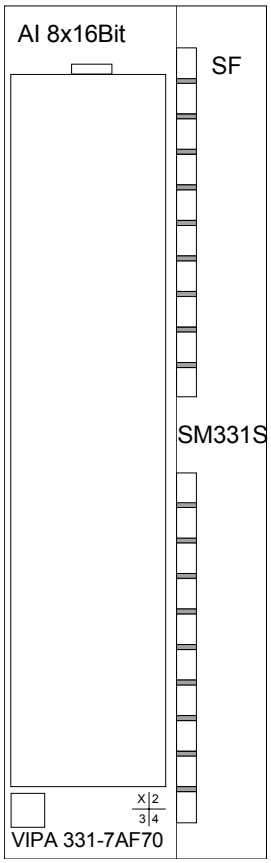
[3] contact bar

[4] flap opened with inner label

Pin assignment status monitor

Depending on the module current or voltage sensors may be connected.

Pin	Assignment	Connection
1	Power supply DC 24V	1 — L+
2	+ Channel 0	2
3	Ground Channel 0	3
4	+ Channel 1	4
5	Ground Channel 1	5
6	+ Channel 2	6
7	Ground Channel 2	7
8	+ Channel 3	8
9	Ground Channel 3	9
10	n.c.	10
11	n.c.	11
12	+ Channel 4	12
13	Ground Channel 4	13
14	+ Channel 5	14
15	Ground Channel 5	15
16	+ Channel 6	16
17	Ground Channel 6	17
18	+ Channel 7	18
19	Ground Channel 7	19
20	Power supply Ground	20 — M



LED	Description
SF	LED (red) Group error, ON as soon as a diagnostic entry is present

Technical data

Module name	VIPA 331-7AF70	VIPA 331-7BF70
Dimensions and Weight		
Dimensions (WxHxD in mm)	40x125x117	
Weight	approx. 185g	
Data for specific module		
Number of inputs	8	
Length of cable - shielded	200m	
Programming specifications		
Input data	16byte (1word per channel)	
Parameter data	41byte (in 11 Data records)	
Diagnostic data	16byte	
Process interrupt data	4byte	
Voltages, Currents, Potentials		
Power supply Reverse polarity protection	DC 24V yes	
Isolation - between channels and backplane bus - between channels	 yes yes	
Permitted potential difference - between channels (U_{CM}) - between channels and $M_{INTERNAL}$ (U_{ISO})	 DC 30V DC 75V / AC 60V	
Isolation tested with	DC 500V	
Current consumption - from the backplane bus (5V) - from power supply L+ (DC 24V)	 530mA 62mA	
Power dissipation of the module	4.0W	
Analog value generation		
Measuring principle	Successive approximation	
Integration time/conversion time/resolution (per channel) - parameterizable - Basic conversion time - Resolution (incl. over range) in Bit - Basic execution time of the module (all channels enabled)	 no (cycle time parameterizable) 25µs 15bit + sign 25µs	

continued ...

... continue

Suppression of interference, Limits of error	VIPA 331-7AF70	VIPA 331-7BF70
Noise suppression for f=nx (f1±1%) (f1= Interference frequency, n=1,2,...) - Common-mode interference (U_{CM}<?V)	(U_{CM}<20V) >80dB	
Crosstalk between the inputs	>50dB	
Operational limit (in the entire temperature range with reference to the input range) - Voltage input ±10V - Current input ±20mA	±0.6% ±0.6%	
Basic error (Operational limit at 25°C referred to the input range) - Voltage input ±10V - Current input ±20mA	±0.4% ±0.4%	
Temperature error (reference to the input range)	±0.0025%/K	
Linearity error (with reference to the input range)	±0.02%	
Repeatability (in steady state at 25°C, reference to the input range)	±0.05%	
Status, Interrupts, Diagnostics		
Interrupts - Process interrupt when limit has been exceeded - Process interrupt at end of cycle - Diagnostic interrupt	parameterizable parameterizable parameterizable	
Diagnostic functions - Group error display - Diagnostic information read-out	red LED (SF) possible	
Data for selecting a sensor		
Input range - Current ±20mA - Voltage ±10V	Input resistance 100Ω -	Input resistance - 120kΩ
Maximum input current for current input (destruction limit)	max. 40mA	-
Maximum input voltage for voltage input (destruction limit)	-	max. 30V
Connection of the sensor - for measuring voltage - for measuring current 2-wire transmitter	- possible	possible -

Appendix

A Index

M.Stich