VIPA System 300S

SPEED7 - CP | 342-1IA70 | Manual

HB140E_CP | RE_342-1IA70 | Rev. 09/46

November 2009

VIPA® art of automation

For permission to reproduce or distribute, please contact:
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 CP 342S-IBS of the System 300S from VIPA. Here you may find besides of a product overview a detailed description of the modules.

Overview

Chapter 1: Basics

This Basics contain hints for the usage and information about the project engineering of a SPEED7 system from VIPA. General information about the System 300S like dimensions and environment conditions will also be found.

Chapter 2: Assembly and installation guidelines 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.

Chapter 3: Hardware description

Here the hardware components of the CP 342S-IBS are more described. The technical data are to be found at the end of the chapter.

Chapter 4: Deployment

Content of this chapter is the functionality of the CP 342S-IBS for SPEED-Bus from VIPA. The module may only be used at a SPEED-Bus slot at the left side of the CPU.

Objective and contents

The manual describes the CP 341S-IBS from VIPA. It contains a description of the construction, project implementation and usage.

This manual is part of the documentation package with order number HB140E_CP and relevant for:

Product	Order number	as of state:	
		CP HW	CP FW
CP 342S-IBS	VIPA 341-1IA70	01	V102

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 CP is 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 the project engineering of a SPEED7 system from VIPA.

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

Content

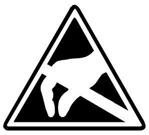
Topic **Page**

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

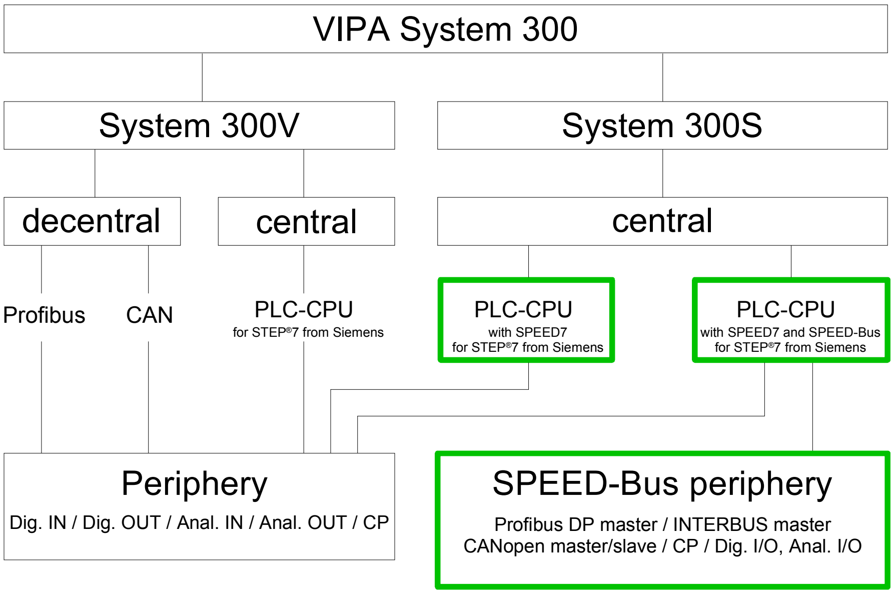
The System 300 is a modular automation system for middle and high performance needs, which you can use either centralized or decentralized. The single modules are directly clipped to the profile rail and are connected together with the help of bus clips at the backside.

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.

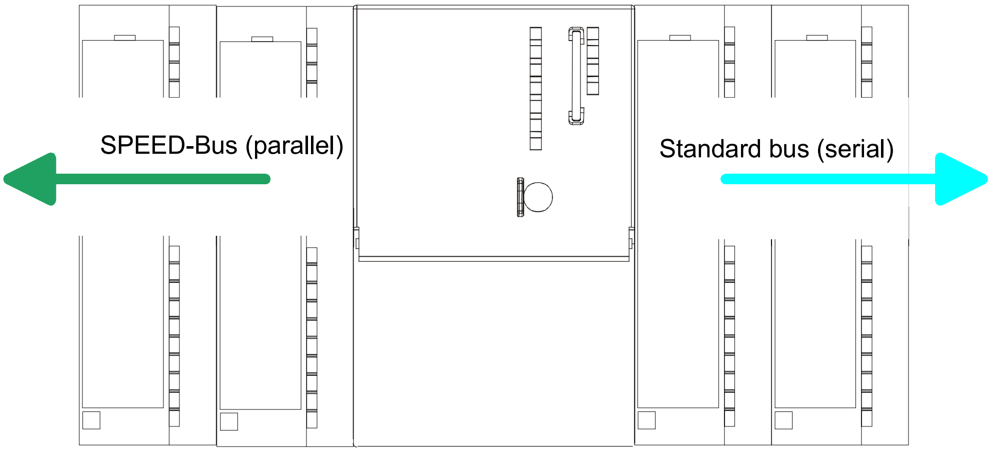


System 300S

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 fieldbus 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 124 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 2 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.

Hints for the project engineering

Overview

For the project engineering of a SPEED7 system please follow this approach:

- Project engineering of the SPEED7-CPU and the internal DP master (if existing) as CPU 318-2DP (318-2AJ00-0AB00)
- Project engineering of the real plugged modules at the standard bus
- Project engineering of the internal Ethernet PG/OP channel after the real plugged modules as virtual CP 343-1 (Setting of IP address, subnet mask and gateway for online project engineering)
- Project engineering of an internal CP343 (if existing) as 2. CP 343-1
- Project engineering and connection of the SPEED-Bus-CPs res. -DP master as CP 343-1 (343-1EX11) res. CP 342-5 (342-5DA02 V5.0)
- Project engineering of all SPEED-Bus modules as single DP slaves in a virtual DP master module (speedbus.gsd required)

Note!

Please do always use the **CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)** from Siemens in the hardware catalog to configure a CPU 31xS from VIPA.

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

Requirements

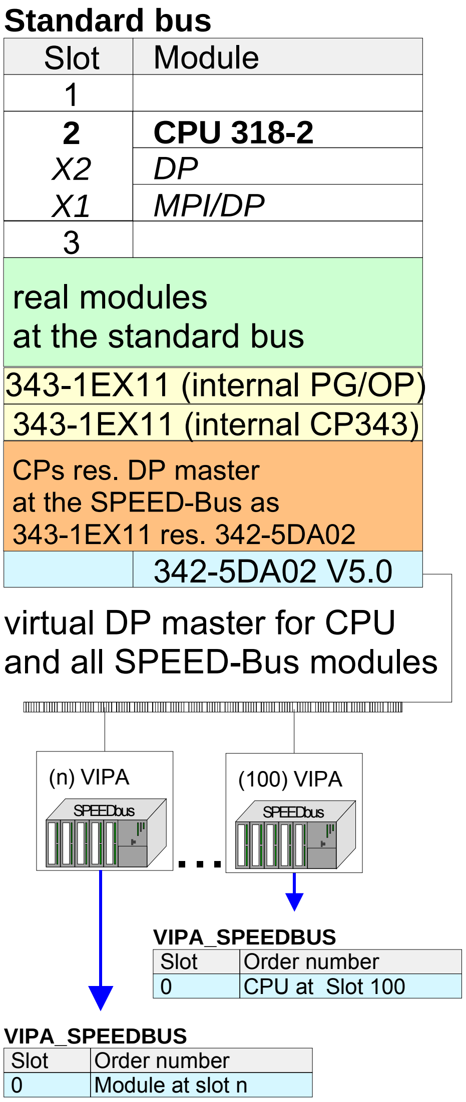
The hardware configurator is part of the Siemens SIMATIC manager. It serves the project engineering. Please look at the hardware catalog for the modules that may be configured.

For the deployment 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 necessary.

Approach

The project engineering of the SPEED7-CPU has the following components:

To be compatible with the Siemens SIMATIC manager, the following steps are required:



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
real modules at the standard bus	
343-1EX11 (internal PG/OP)	
343-1EX11 (internal CP343)	
CPs res. DP master at the SPEED-Bus as 343-1EX11 res. 342-5DA02	
	342-5DA02 V5.0

virtual DP master for CPU and all SPEED-Bus modules

VIPA_SPEEDBUS

Slot	Order number
0	CPU at Slot 100

VIPA_SPEEDBUS

Slot	Order number
0	Module at slot n

- *Preparation*
  Start the hardware configurator from Siemens and include the speedbus.gsd for the SPEED-Bus from VIPA.
- *Project engineering of the CPU*
  Project a CPU 318-2DP (318-2AJ00-0AB00 V3.0). If your SPEED7-CPU contains a DP master, you may now connect it with Profibus and configure your DP slaves.
- *Project engineering of the real plugged modules at the standard bus*
  Set the modules that are at the right side of the CPU at the standard bus starting with slot 4.
- *Project engineering of the integrated CPs*
  For the internal Ethernet PG/OP channel you have to set a CP 343-1 (343-1EX11) as 1. module at the real plugged modules. If your SPEED7-CPU has additionally an integrated CP 343, this is also configured as CP 343-1 but always below the former placed CP 343-1.
- *Project engineering of the SPEED-Bus-CPs and -DP master*
  Plug and connect all CPs as 343-1EX11 and DP master as 342-5DA02 V5.0 at the SPEED-Bus below the former configured internal CPU components.
  Please regard that the sequence within a function group (CP res. DP master) corresponds the sequence at the SPEED-Bus from right to left.
- *Project engineering of the CPU and 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 a DP master (342-5DA02 V5.0) with master system as last module. The Profibus address must be <100!
  Now include the slave "vipa_speedbus" for the CPU and every module at the SPEED-Bus. After the installation of the speedbus.gsd you may find this under Profibus-DP / Additional field devices / I/O / VIPA_SPEEDbus. Set the slot number of the module (100...110) as Profibus address and plug the according module at slot 0 of the slave system.

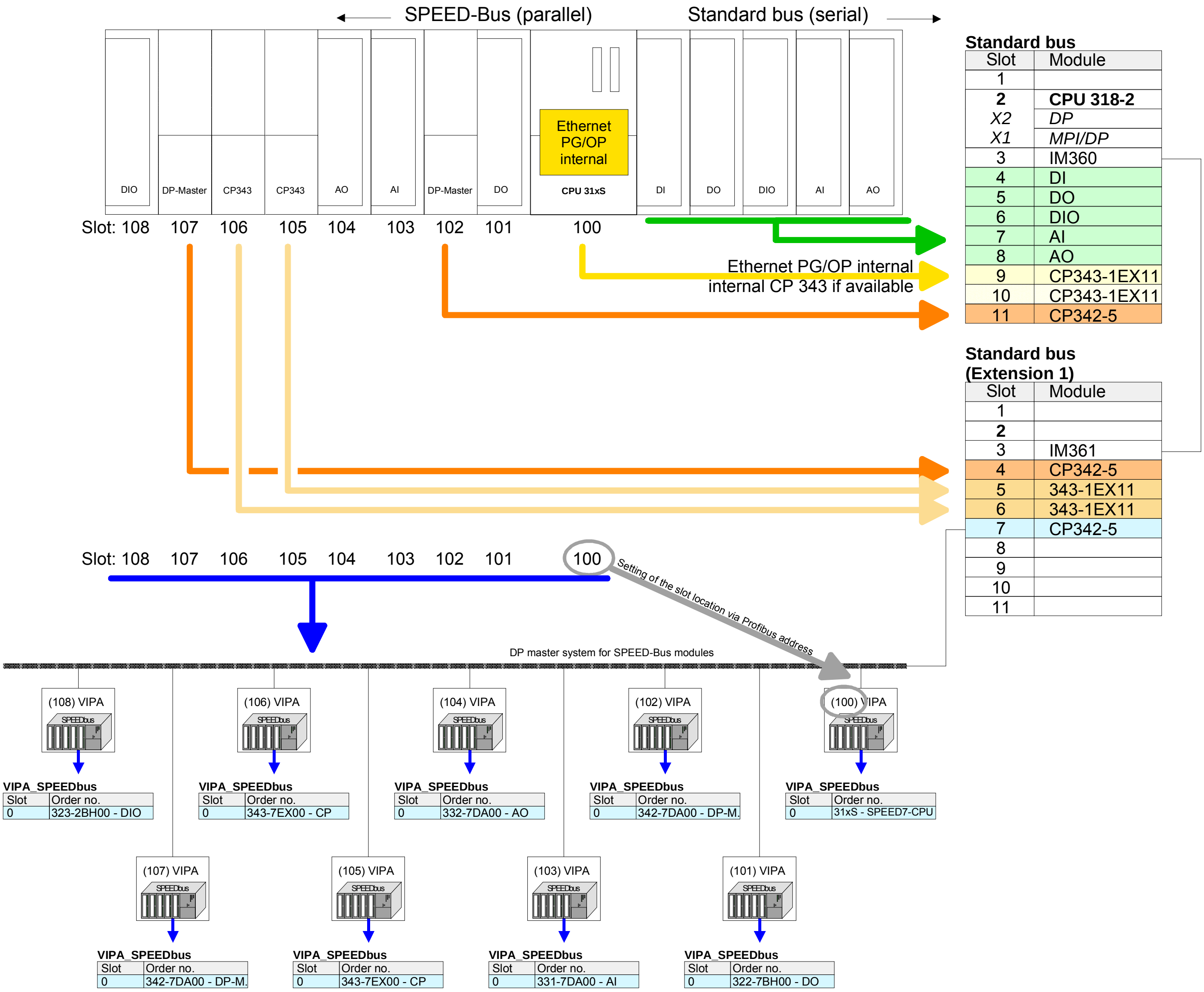
Bus extension with IM 360 and IM 361

To extend the bus you may use the IM 360 from Siemens, where 3 further extensions racks can be connected via the IM 361. Bus extensions must be placed at slot 3.

More detailed information is to be found in the chapter "Deployment CPU 31xS" at "Addressing".

Summary

The following illustration summarizes all project engineering steps:



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	IM360
4	DI
5	DO
6	DIO
7	AI
8	AO
9	CP343-1EX11
10	CP343-1EX11
11	CP342-5

Standard bus (Extension 1)

Slot	Module
1	
2	
3	IM361
4	CP342-5
5	343-1EX11
6	343-1EX11
7	CP342-5
8	
9	
10	
11	

VIPA_SPEEDbus

Slot	Order no.
0	323-2BH00 - DIO

VIPA_SPEEDbus

Slot	Order no.
0	343-7EX00 - CP

VIPA_SPEEDbus

Slot	Order no.
0	332-7DA00 - AO

VIPA_SPEEDbus

Slot	Order no.
0	342-7DA00 - DP-M.

VIPA_SPEEDbus

Slot	Order no.
0	31xS - SPEED7-CPU

VIPA_SPEEDbus

Slot	Order no.
0	342-7DA00 - DP-M.

VIPA_SPEEDbus

Slot	Order no.
0	343-7EX00 - CP

VIPA_SPEEDbus

Slot	Order no.
0	331-7DA00 - AI

VIPA_SPEEDbus

Slot	Order no.
0	322-7BH00 - DO

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

Note!

The sequence of the DPM- and CP function groups is insignificant. You only have to take care to regard the sequence within a function group (DP1, DP2... res. CP1, CP2 ...).

Hint, valid for all SPEED-Bus modules!

The SPEED-Bus always requires the Siemens DP master CP 342-5 (342-5DA02 V5.0) as last module to be included, connected and parameterized to the *operation mode* DP master. Every SPEED-Bus module has to be connected as VIPA_SPEEDbus slave into this master system.

By setting the SPEED-Bus slot number via the Profibus address and by including the according SPEED-Bus module at slot 0, the SIMATIC manager receives information about the modules at the SPEED-Bus.

Additionally the following configurations are required depending on the module.

Project engineering of the DP master at the SPEED-Bus

The hardware configuration and Profibus project engineering happens in the SIMATIC manager from Siemens. You have to parameterize a virtual CP 342-5 (342-5DA02 V5.0) for every SPEED-Bus-DP master at the standard bus following the real modules and connect it with the depending DP slaves.

Project engineering CP 343 at the SPEED-Bus

SPEED-Bus-CPs have to be configured in the Siemens SIMATIC manager at the standard bus behind the real modules as virtual CP 343 (343-1EX11) and are then connected with the according Ethernet components. For the connection, the Siemens project engineering tool NetPro is required.

Project engineering of the CAN master at the SPEED-Bus

The project engineering of the CANopen master at the SPEED-Bus happens in WinCoCT (**Win**dows **C**AN**o**pen **C**onfiguration **T**ool) from VIPA.

You export your project from WinCoCT as wld-file. This wld-file can be imported into the hardware configurator from Siemens.

An additional inclusion at the standard bus is not necessary.

Project engineering of the Interbus master at the SPEED-Bus

The project engineering of the IBS master system takes place in your CPU user application using the VIPA FCs.

An additional inclusion at the standard bus is not necessary.

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.

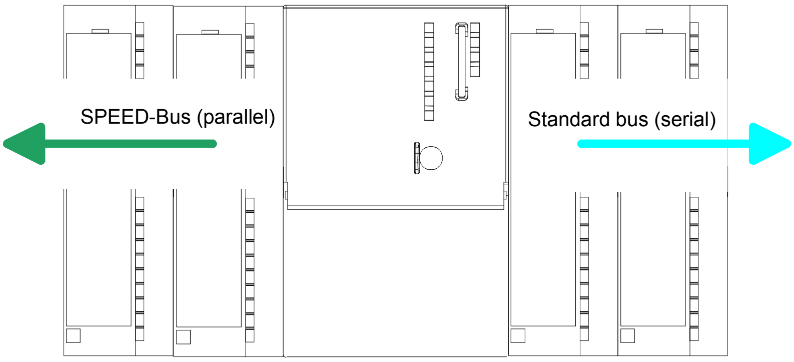
Content

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

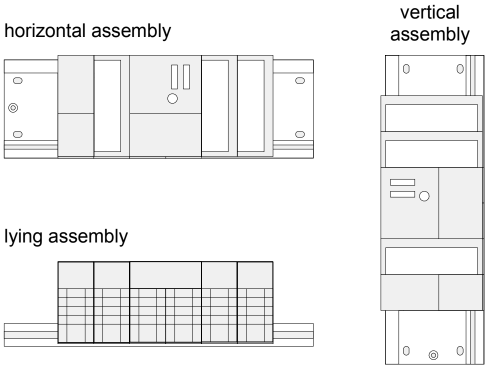
With SPEED-Bus the bus connection happens via a SPEED-Bus rail integrated in the profile rail at the left side of the CPU. Due to the parallel SPEED-Bus not all slots must be occupied in sequence.

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:

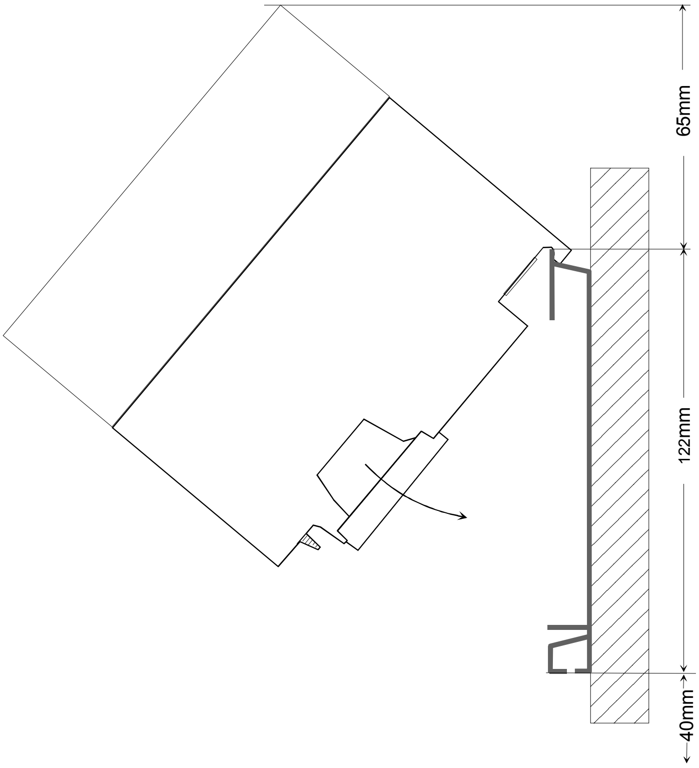
• horizontal assembly:	from 0 to 60°C
• vertical assembly:	from 0 to 40°C
• lying assembly:	from 0 to 40°C

Installation dimensions

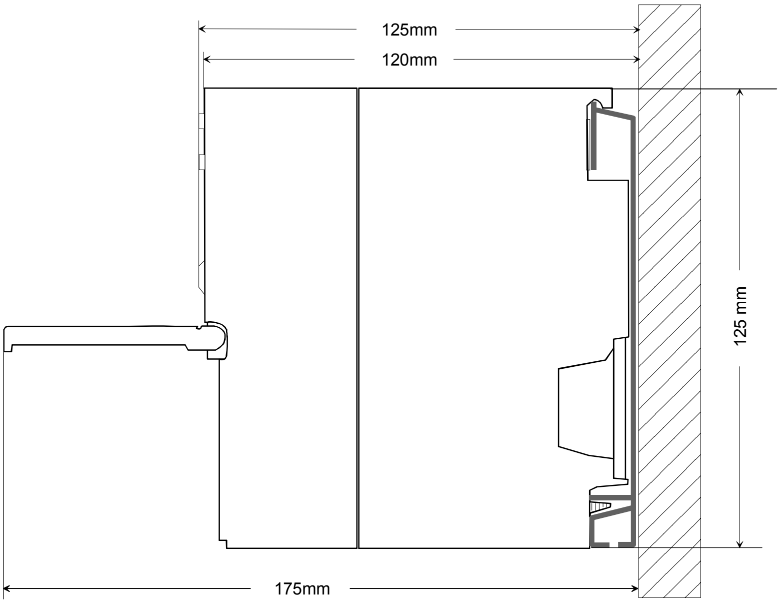
Dimensions Basic enclosure

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

Dimensions



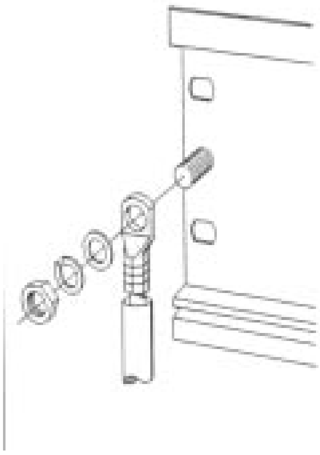
Installation dimensions



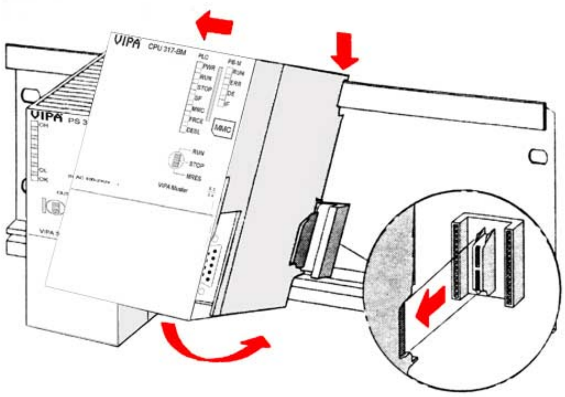
Installation Standard-Bus

Approach

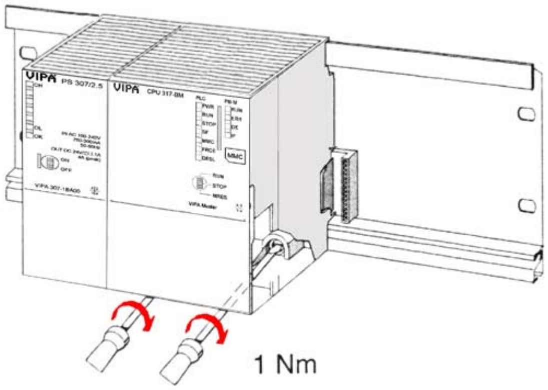
If you do not deploy SPEED-Bus modules, the assembly at the standard bus happens at the right side of the CPU with the following approach:



- 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.

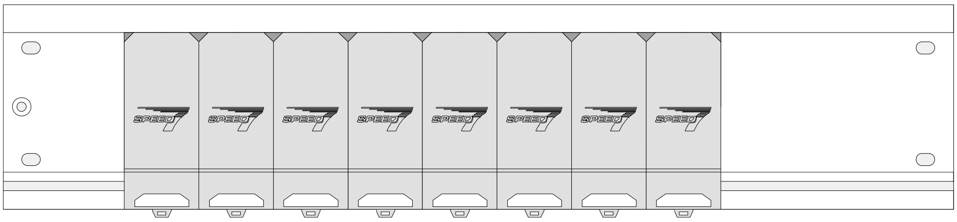
Danger!

- Before installing or overhauling the System 300, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

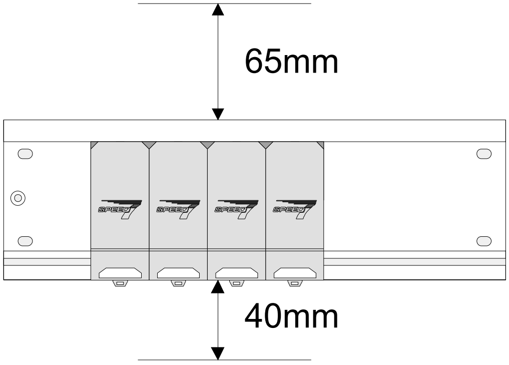
Assembly SPEED-Bus

Pre-manufactured SPEED-Bus profile rail

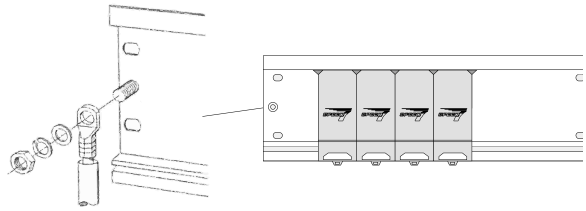
For the deployment of SPEED-Bus modules, a pre-manufactured SPEED-Bus rail is required. This is available mounted on a profile rail with 2, 6 or 10 extension plug-in locations.



Installation of the profile rail



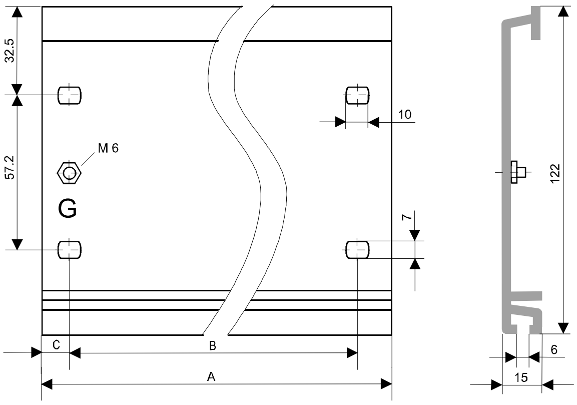
- Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail.
- Please look for a low-impedance connection between profile rail and background



- 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.

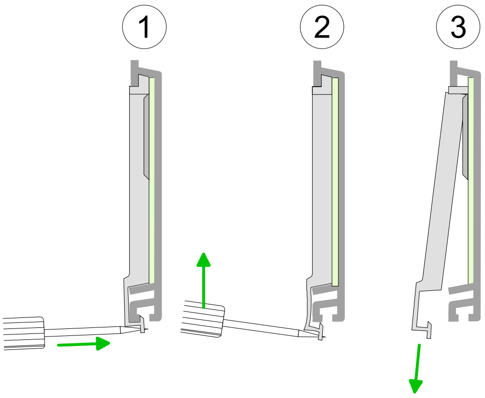
Profile rail



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

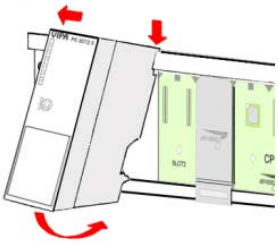
* Unit pack 10 pieces

Installation SPEED-Bus-Module

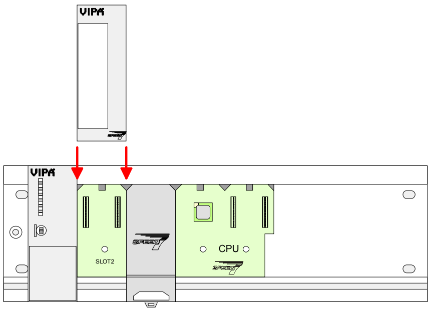


- Dismantle the according protection flaps of the SPEED-Bus plug-in locations with a screw driver (open and pull down).

  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.

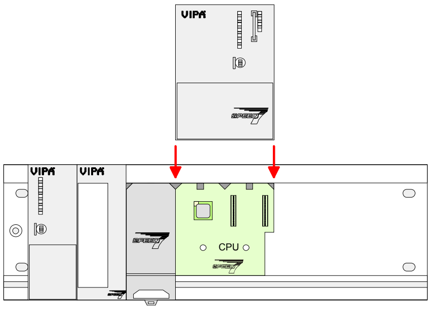


- At deployment of a DC 24V power supply, install it at the shown position at the profile rail at the left side of the SPEED-Bus and push it to the left to the isolation bolt of the profile rail.
- Fix the power supply by screwing.



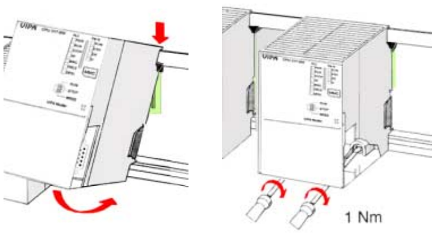
- To connect the SPEED-Bus modules, plug it between the triangular positioning helps to a plug-in location marked with "SLOT ..." and pull it down.
- Only the "SLOT1 DCDC" allows you to plug-in either a SPEED-Bus module or an additional power supply.
- Fix the modules by screwing.

Installation CPU without Standard-Bus-Modules

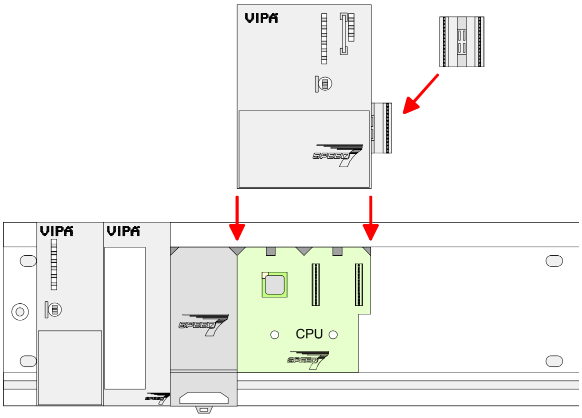


- To deploy the SPEED7-CPU exclusively at the SPEED-Bus, plug it between the triangular positioning helps to the plug-in location marked with "CPU SPEED7" and pull it down.
- Fix the CPU by screwing.

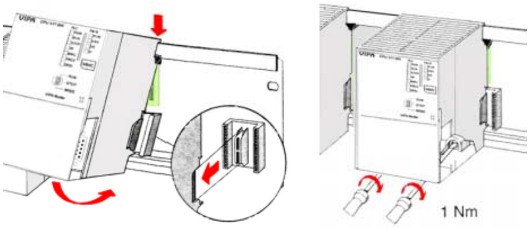
Please regard that not all CPU 31xS may be deployed at the SPEED-Bus!



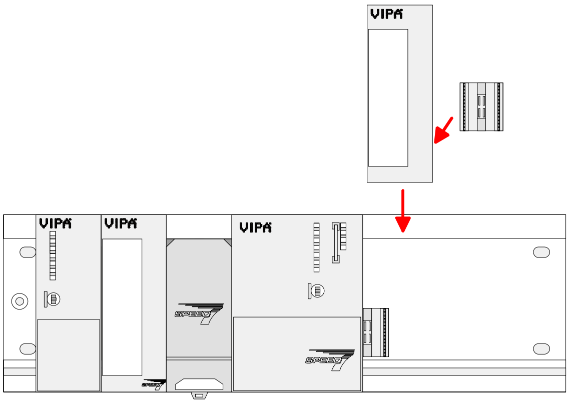
Installation CPU with Standard-Bus-Modules



- 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 300V, 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.

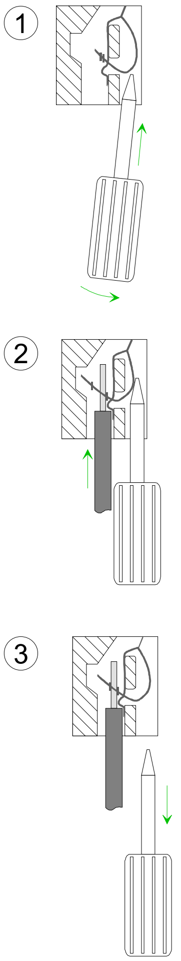
Danger!

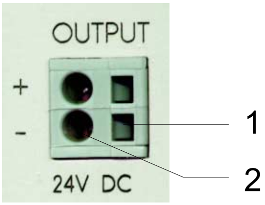
- Before installation or overhauling, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

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.





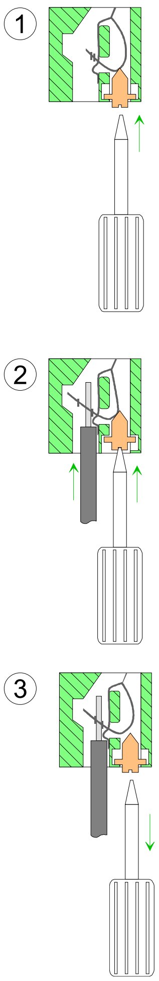
[1] Rectangular opening for screwdriver
[2] Round opening for wires

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

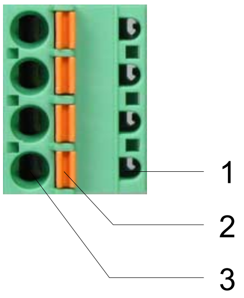
- To conduct a wire you plug a fitting screwdriver obliquely into the rectangular opening like shown in the picture.
- To open the contact spring you have to push the screwdriver in the opposite direction and hold it.
- Insert the insulation striped wire into the round opening. You may use wires with 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.

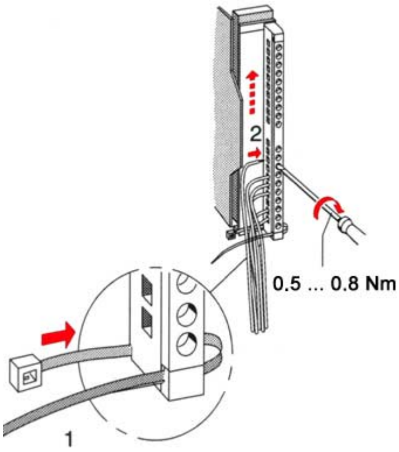
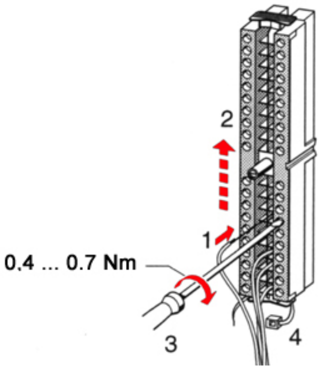
- For cabling you push the locking vertical to the inside with a suiting screwdriver and hold the screwdriver in this position.
- Insert the insulation striped wire into the round opening. You may use wires with a cross-section from 0.08mm^2 to 2.5mm^2.
- By removing the screwdriver the wire is connected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

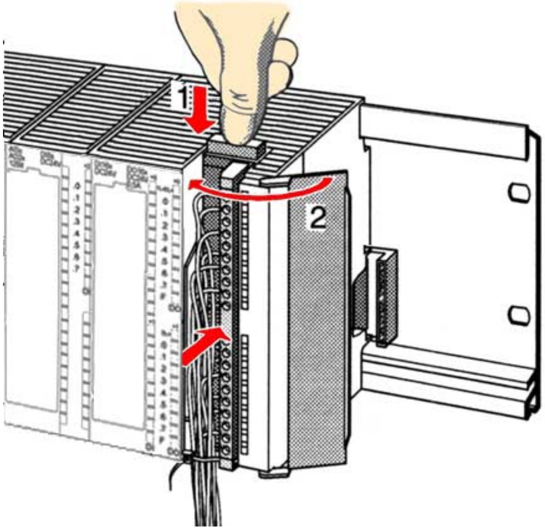
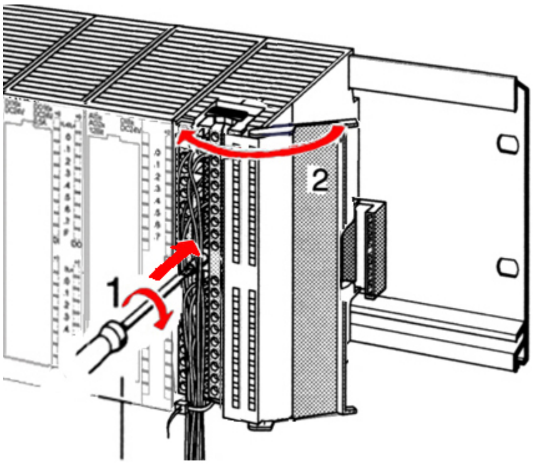
In the following the cabling of the three variants of the front-facing connector is shown:

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

20pole screw connection VIPA 392-1AJ00	**40pole screw connection** VIPA 392-1AM00
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.  	Bolt the fixing screw of the front connector.  0.4 ... 0.7 Nm
Now the front connector is electrically connected with your module.	
Close the front flap.	
Fill out the labeling strip to mark the single channels and push the strip into the front flap.	

Installation Guidelines

General

The installation guidelines contain information about the interference free deployment of System 300 systems. 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 System 300V modules.
  - For lightening cabinets you should prefer incandescent lamps and avoid luminescent lamps.
- Create an 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 300V 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 300V 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 Hardware description

Overview

Here the hardware components of the CP 342S-IBS are more described. The technical data are to be found at the end of the chapter..

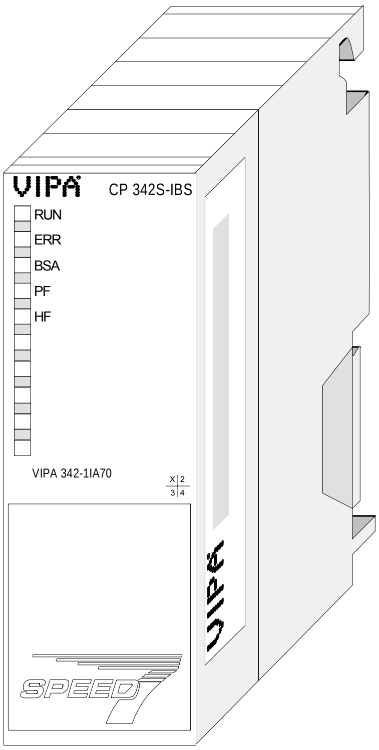
Content

Topic **Page**

Properties

General

The CP 342S-IBS in the following may only be used at the SPEED-Bus.



CP 342S-IBS

- **I**nter**b**u**s** master (IBS master) for SPEED-Bus
- Up to 512 slaves connectable
- Supports PCP communication 2.0 with band widths of 1, 2 and 4 words at 62 couplers with basic functions and 127 configurable couplers.
- Diagnostics via LEDs, RS232 interface, Mini-DIN slot and DPM

Order data

Type	Order No.	Description
CP 342S-IBS	VIPA 342-1IA70	Interbus master for SPEED-Bus

Structure

Interbus platform

As Interbus hardware platform, the Interbus master card USC4-2 from Phoenix Contact is used.

The Interbus section manages all tasks concerning network management and network diagnosis. Here the communication with the CPU happens via a **D**ual **p**ort **m**emory (DPM).

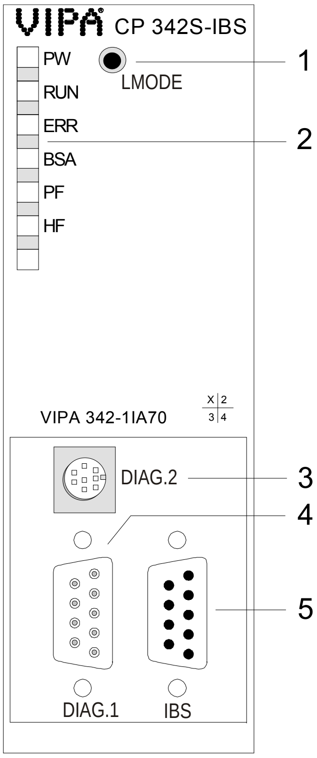
Among others, the DPM has the following interfaces for send and receive:

- SSGI (**S**tandard **Sig**nal **I**nterface) for the exchange of messages like e.g. request of services from the master
- DTA (**Da**ta) interface for the exchange of process data

Note!

Due to the fact that VIPA provides the same services for master and slave parameterization for this master, we refer at the according places to the extensive documentation of the services from Phoenix Contact.

Front view



[1] LMODE button

[2] LED status indicators

The following components are under the front flap

[3] Mini-DIN slot for diagnostics

[4] RS232 interface for project engineering and diagnostics

[5] RS422 Interbus interface

Components

LEDs

The CP 342S-IBS carries a number of LEDs that are available for diagnostic purposes on the bus and for displaying the local status. These give information according to the following pattern over the operating condition of the CP:

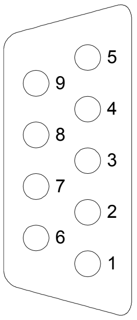
PW green	RUN green	ERR red	BSA yellow	PF yellow	HF yellow	Meaning
○	○	○	○	○	○	Module is not power supplied
●	●	○	○	○	○	Interbus is ready for data transfer
●	☼	○	○	○	○	Interbus is active, bus parameters are transferred, bus is checked.
●	☼	●	○	○	○	At least 1 slave is missing or bus error.
●	X	●	●	○	○	At least 1 segment of the subordinate bus is switched off.
●	●	●	○	●	○	Peripheral fault at a subordinate bus member
●	X	●	○	○	●	Error in CP 342S-IBS

on: ● off: ○ flashing: ☼ irrelevant: X

Jacks and plugs

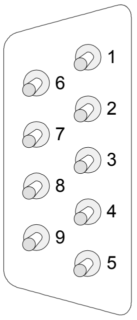
The interfaces for the inbound and the outbound bus lines are located on the front of the module.

9pin D-type jack for Interbus connection



Pin	Assignment
1	DOH
2	DIH
3	GND_{iso}
4	GND
5	$+5\text{V}_{\text{iso}}$ (90mA)
6	DOL
7	DIL
8	+5V (90mA)
9	reserved

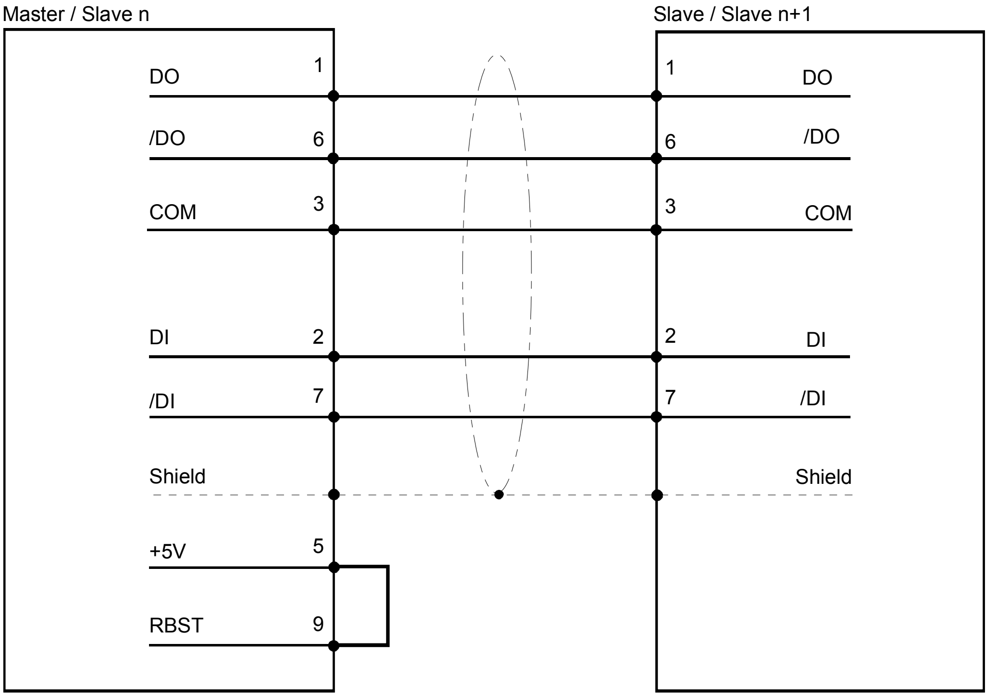
9pin D-type plug for diagnosis and project engineering.



Pin	Assignment
1	reserved
2	TxD
3	RxD
4	reserved
5	GND
6	reserved
7	RTS
8	CTS
9	reserved

Cabling with Interbus

For master-slave and slave-slave connection the same connection cable is used. Due to the ring structure and the common logic ground, the cable consists of 5 cores and has the following assignment:



Note!

Please take care that the plug for the „continuative interface" has a bridge between Pin 5 and 9, otherwise the following slaves are not recognized!

LMODE button

By pressing the LMODE button during Power ON since RUN of the IBS-Master the current Interbus configuration is read and stored at the flash memory. In these ways the IBS master may be pursued e.g., for diagnosis without configuration.

Power supply

The Interbus master gets its power supply via the SPEED-Bus. Here the current consumption is max. 600mA.

Firmware update

There is the possibility to execute a firmware update of the CP 342S-IBS among others via the SPPED7 CPU by means of a MMC.

So a firmware files may be recognized and assigned with startup, a pkg file name is reserved for each updateable component and hardware release, which begins with "px" and differs in a number with six digits.

The pkg file name may be found at a label right down the front flap of the module.

Details to the firmware update may be found in manual HB140_CPU at chapter "Deployment CPU 31xS" at "firmware update".

Technical data

Electrical data	VIPA 342-1IA70	
Power supply	via backplane bus	
Current consumption	max. 600mA	
Power dissipation	3.0W	
Isolation	≥ AC 500V	
Status indicators	via LEDs on the front	
Connections/interfaces	9pin D-type socket	Interbus connector
	9pin D-type plug	Diagnostics interface
	Mini-DIN slot	Diagnostics interface
Interbus interface		
Connection	9pin D-type socket	
Network topology	Linear with integrated return circuit	
Medium	Screened twisted pair cable	
Data transfer rate	500kBaud	
Total length	12.8km (400m between 2 stations)	
Max. no. of stations	512	
Combination with peripheral modules		
Max. no of slaves	512	
Max. no. of input bytes	32byte each station	
Max. no. of output bytes	32byte each station	
Dimensions and weight		
Dimensions (WxHxD) in mm	40x125x120	
Weight	205g	

Chapter 4 Deployment

Overview

Content of this chapter is the functionality of the CP 342S-IBS for SPEED-Bus from VIPA. The module may only be used at a SPEED-Bus slot at the left side of the CPU.

Content

Topic **Page**

Basics Interbus

General

Interbus is a pure master/slave system that has very few protocol overheads. For this reason it is well suited for applications on the sensor/actuator level. Interbus was developed by PHOENIX CONTACT, Digital Equipment and the Technical University of Lemgo during the 80s. The first system components became available in 1988. To this day the communication protocol has remained virtually unchanged. It is therefore means that it is entirely possible to connect devices of the first generation to the most recent master interfaces (generation 4).

Interbus devices are subject to the DIN standard 19258 that defines levels 1 and 2 of the protocol amongst others.

Interbus as shift register

Interbus is based upon a ring structure that operates as a cyclic shift register. Every Interbus module inserts a shift register into the ring. The number of I/O points supported by the module determines the length of this shift register. A ring-based shift register is formed due to the fact that all the devices are connected in series and that the output of the last shift register is returned to the bus master.

The length and the structure of this shift register depend on the physical construction of the entire Interbus system.

Interbus operates by means of a master-slave access method where the master also provides the link to any high-level control system. The ring-structure includes all connected devices actively in a closed communication loop.

In comparison to client-server protocols where data is only exchanged when a client receives a properly addressed command, Interbus communications is cyclic in nature and data is exchanged at constant intervals. Every data cycle addresses all devices on the bus.

Restrictions

- Max. 512 participants with 32byte I/O per station
- Up to 400m distance between 2 stations at 500kB
- Total distance up to 13km (Repeater function in every station)
- Removal res. addition of modules during runtime is not permitted
- Data consistency is secure for 1byte. To avoid inconsistencies use the *asynchronous* data exchange with consistency bit or the *interrupt controlled* synchronous pulse.

Note!

Before alterations you must disconnect the according bus coupler from voltage. Please take care to adjust the initialization in the master when changing the periphery!

Modes of operation

Interbus has two modes of operation:

- ID cycle
  An ID cycle is issued when the Interbus system is being initialized and also upon request. During the ID cycle the bus master reads the ID register of every module connected to the bus to generate the process image.
- Data cycle
  The actual transfer of data occurs during the data cycle. During the data cycle the input data from the registers of all devices is transferred to the master and the output data is transferred from the master to the devices. This is a full duplex data transfer.

ID cycle

During the ID cycle that is executed when the Interbus system is being initialized the different modules connected to the bus identify themselves with their individual functionality and the word length. When the Interbus coupler is turned on, it determines its Interbus length during the initialization phase of the bus modules and generates the respective ID code. Depending on the configuration the Interbus coupler replies with a message identifying it as an analog or a digital remote bus device with variable word length.

The Interbus ID code consists of 2byte. The MSB (byte 2) describes the length of the data words that will be transferred. The LSB (byte 1) describes the type of bus module, i.e. the type of signal and other performance criteria

Byte	Bit 7 ... Bit 0
1	Bit 1 ... Bit 0: Direction of data transfer: 00: not used 01: output 10: input 11: input/output Bit 3 ... Bit 2: terminal type Bit 7 ... Bit 4: terminal class The type and class are determined by the Interbus-Club
2	Bit 4 ... Bit 0: Data width 0 to 10 words (binary) Bit 7 ... Bit 5: reserved

Data cycle

Process data words also contain control and inspection information. This information is only transferred once at the beginning or at the end of the peripheral data of any data cycle. This is why this system is also referred to as a cumulative frame procedure.

The communication principle is independent of the type of data being transferred:

Process data that must be transferred to the periphery is stored in the output buffer of the master in the same sequence as the output stations are connected to the bus. The transfer occurs when the master shifts the "loop-back word" through the ring. Following the loop-back word, all the output data is placed on the bus.

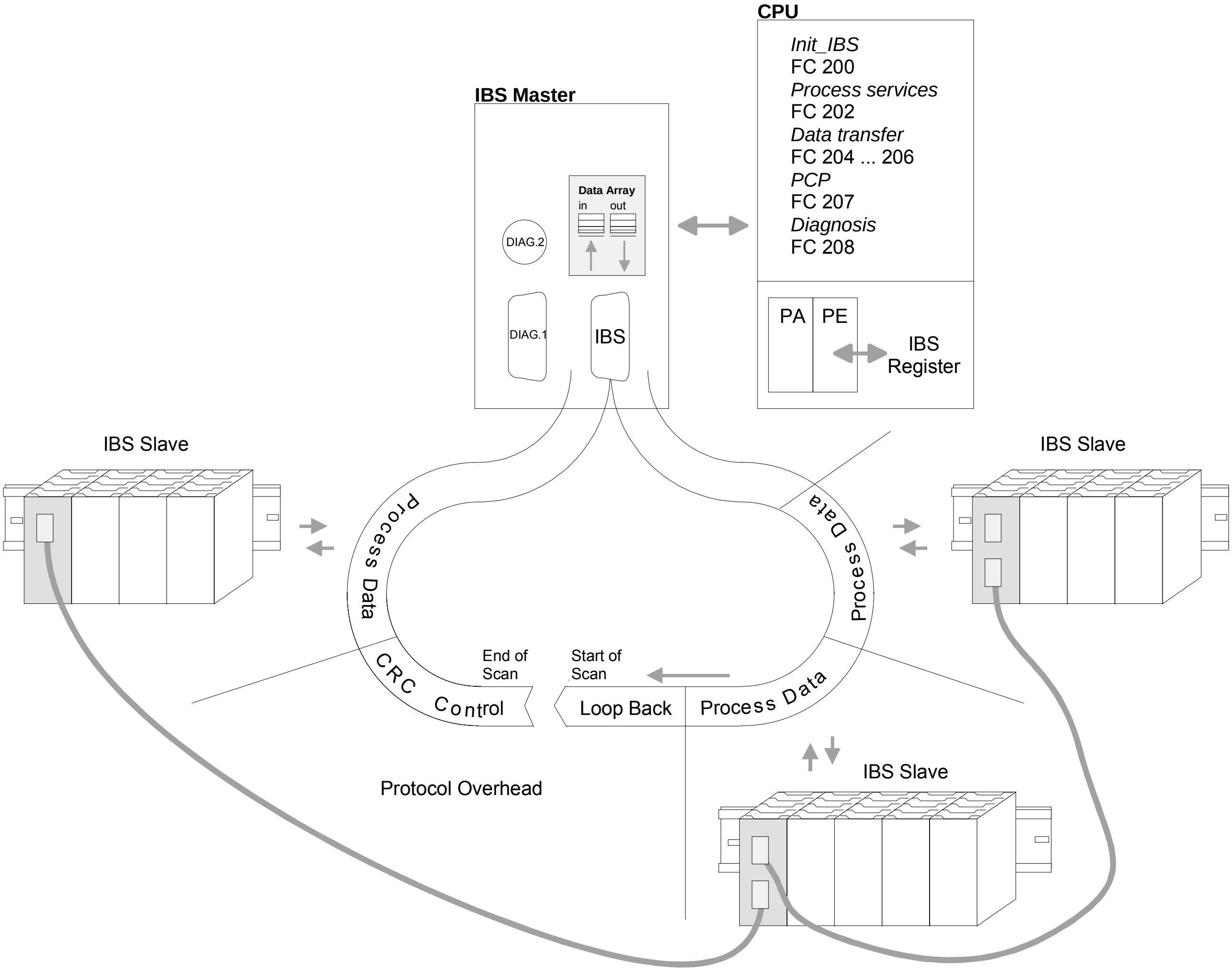
... continue data cycle

This means that the data is shifted through the shift register. The information from the process is returned as input data to the input buffer of the master at the same time as the output data is being sent.

The output data is located at the correct position in the shift registers of the different stations when the entire cumulative frame telegram has been sent and read back again. At this point, the master issues a special control command to the devices on the bus to indicate the end of the data transfer cycle.

When the data check sequence has been processed, output data for the process is transferred from the shift registers. This is stored in the devices connected to the bus and transferred to the respective periphery. At the same time, new information is read from the periphery into the shift registers of the input devices in preparation for the next input cycle.

This procedure is repeated on a cyclic basis. This means that the input and output buffers of the master are also updated cyclically. Interbus data communications is therefore full duplex in nature; i.e. both input data and output data are transferred during a single data cycle.



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.

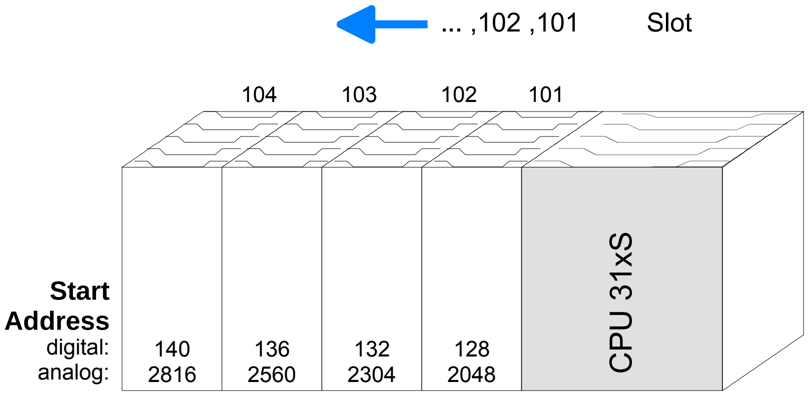
Automatic addressing

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 = $4 \cdot (\text{slot} - 101) + 128$
AIOs, FMs, CPs:	Start address = $256 \cdot (\text{slot} - 101) + 2048$



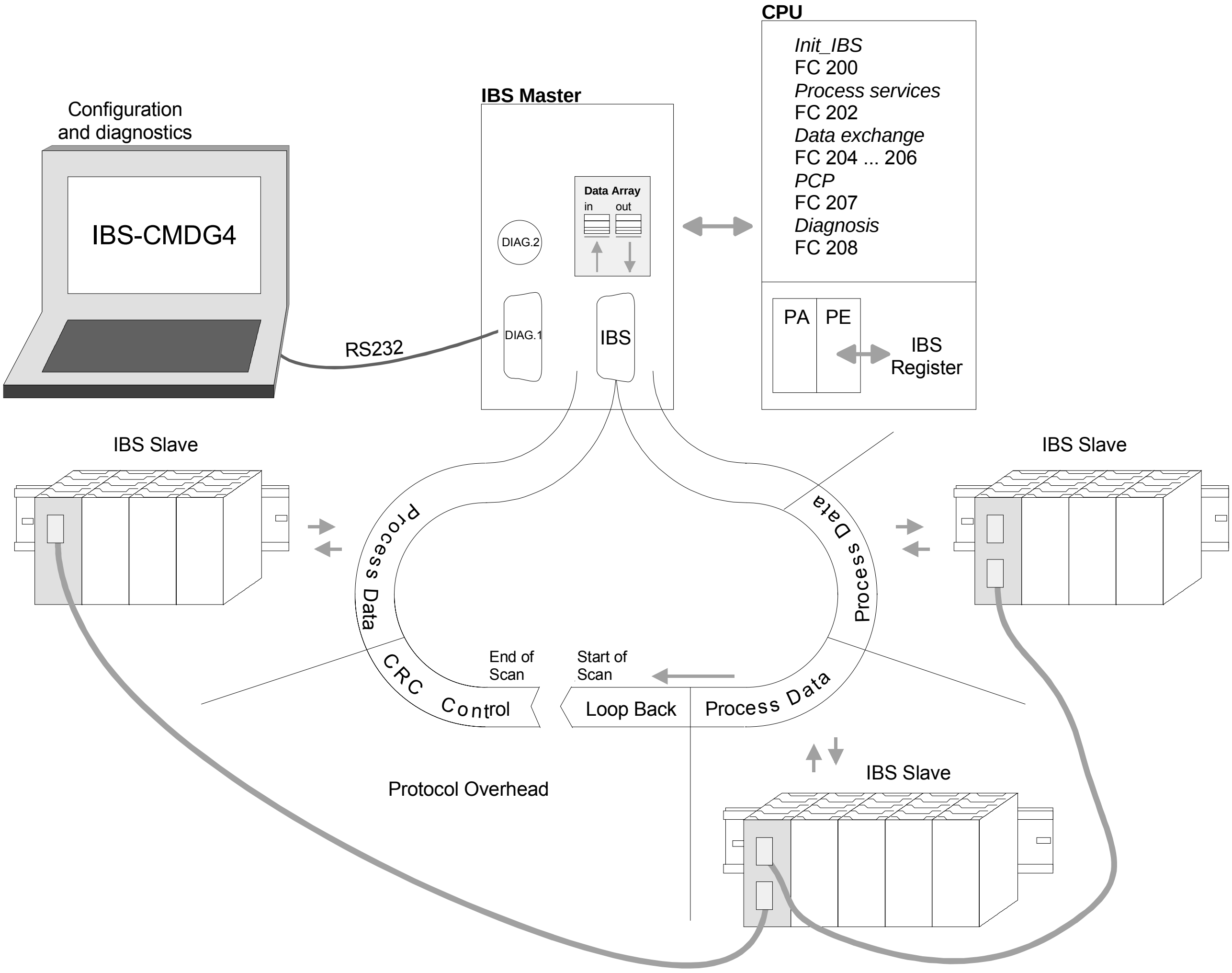
Configuration and Diagnostics

Overview

Project engineering and diagnosis at the VIPA SPEED-Bus IBS master are taking place by means of FCs that you may receive from VIPA.

Additionally you may access the IBS master via the serial diagnosis interface with the help of the project engineering and parameterization tool IBS-CMDG4 from Phoenix Contact.

After the boot sequence, the IBS master occupies in the CPU depending on the plug-in location each 34byte in the I/O address range.



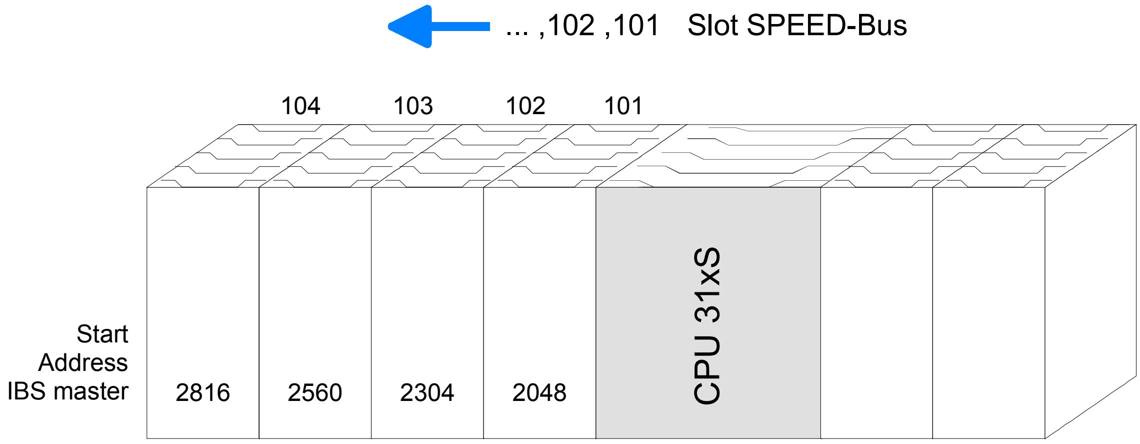
Addressing

Overview

At the start-up of the CPU, the IBS master are, if no hardware configuration is present, automatically mapped into the address range of the CPU after the following formula:

$$\textit{Start address} = 256 \cdot (\textit{Slot}-101)+2048$$

Information about the plug-in location is to be found in the following table:



Alter address

To alter automatically assigned addresses, you have to execute a hardware configuration. Here you may alter the addresses at any time via a virtual Profibus system by including the SPEEDBUS.GSD. For this, click on the properties of the according IBS master module and set the wanted address.

The approach of the hardware configuration is to be found in the according chapter of this manual.

34byte register in the CPU I/O address range

Every IBS master occupies with a register a 34byte wide memory area in the I/O address range of the CPU.

A more detailed description about the register structure is to be found at the following pages.

Register allocation

Overview

The registers have, starting with the start address LADDR, the following structure and occupy 34byte in the I/O address range of the CPU. In the following you see a description of the register elements:

Address	Assignment	Direction
LADDR	Interrupt register	CPU > Master
LADDR+1	Interrupt register	Master > CPU
LADDR+2	SSGI acknowledge	Master > CPU
LADDR+4	SSGI notification	Master > CPU
LADDR+6	SSGI result	Master > CPU
LADDR+8	SSGI status	Master > CPU
LADDR+10	SSGI start	CPU > Master
LADDR+12	reserved	-
LADDR+14	Standard function parameter register	CPU > Master
LADDR+16	Standard function start register	CPU > Master
LADDR+18	Standard function status register	Master > CPU
LADDR+20	Master diagnosis parameter register	Master > CPU
LADDR+22	Master diagnosis status register	Master > CPU
LADDR+24	reserved	-
LADDR+26	Slave diagnosis status register	Master > CPU
LADDR+28	Configuration register	Master > CPU
LADDR+30	reserved	-
LADDR+32	Status sysfail register	Master > CPU

Interrupt register CPU > Master (LADDR)

Via this register and the register "Interrupt Register Master > CPU" interrupt requests for the synchronous operating mode (FC 206 - IRQ_RW) are created.

7	6	5	4	3	2	1	0
x	x	x	x	X	x	x	x

Possible contents of the register:

APPLICATION_READY_COMMAND 0Eh

Interrupt register Master > CPU (LADDR+1)

This register serves the synchronization between CPU and IBS master during the boot sequence. Additionally it serves together with the register "Interrupt Register CPU > Master" for creation of interrupt requests for the synchronous operating mode. After Power-up-Reset and successfully finished self test, the IBS master writes the value C3h into this register.

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x

Possible contents of the register:

MASTER_READY_COMMAND	C3h
DATA_CYCLE_READY_COMMAND	10h

SSGI acknowledge
Master > CPU
(LADDR+2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	x	res.	res.	res.	res.	res.	res.	res.	res.

Bit 8: Acknowledge-Bit for the message exchange via SSGI (**St**andard **Sig**nal **I**nterface)

SSGI notification
Master > CPU
(LADDR+4)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	x	res.	res.	res.	res.	res.	res.	res.	res.

Bit 8: Notification-Bit for the message exchange via SSGI

SSGI result
Master > CPU
(LADDR+6)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	x	res.	res.	res.	res.	res.	res.	res.	x

Bit 0: Error during automatic configuration

Bit 8: Result-Bit for the message exchange via SSGI

SSGI status
Master > CPU
(LADDR+8)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	X	res.	res.	res.	res.	res.	res.	res.	x

Bit 0: 0: Automatic start-up is not executed at this time
1: Automatic start-up is executed at this time

Bit 8: Status-Bit for the message exchange via SSGI

SSGI start
CPU > Master
(LADDR+10)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	x	res.	res.	res.	res.	res.	res.	res.	x

Bit 0: 0: Automatic start-up is not executed at this time
1: Automatic start-up is executed at this time

Bit 8: Status-Bit for the message exchange via SSGI

Standard fct.-
param. register
CPU > Master
(LADDR+14)

The register is used by the CPU for transmission of parameters for the standard functions that are activated with the standard function start register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Standard fct. start register CPU > Master (LADDR+16)

With the help of this registers and the standard functions parameter register you may control the IBS master without using the SSGI. Several often used commands or command sequences may be executed with the two registers. This minimizes the efforts for service requests.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	res.	res.	res.	res.	res.	res.	res.	x	x	x	x	x	x	x

Bit 0 Start bit Start_Data_Transfer_Req
Starts the data transfer.
Precondition: IBS master is in state ACTIVE
Parameter: none

Bit 1 Start bit Alarm_Stop_Req, Activate_Configuration_Req
Interrupts the data transfer, sets the outputs of all IBS stations to "0" and activates a new configuration frame. Afterwards the IBS master is in state ACTIVE.
Parameter: Number of the configuration frame to be loaded (e.g. "1")

Bit 2 Start bit Confirm_Diagnostics_Req
This bit updates the contents of the diagnosis register and the diagnosis monitors.

Bit 3 Start bit Control_Active_Configuration_Req Off
This bit allows you to shut down INTERBUS segments.
Parameter: The segment-no. has to be stored in the higher valued byte and the position in the lower valued byte. At shut-down of a local bus participant, all stations in the according local bus are shut down. When entering a distant bus station or a bus coupler, besides of the concerning device also the continuative IBS interface is shut down and thus all further IBS stations.

Bit 4 Start bit Control_Active_Configuration_Req On
This bit re-activates IBS segments that have been shut down before.
Parameter: See Bit 3

Bit 5 Start bit Control_Active_Configuration_Req Disable
The station set as parameter is toggled in-active within the configuration frame. It may also physically not remain within the data ring and has to be bridged manually.
Parameter: The segment-no. has to be stored in the higher valued byte and the position in the lower valued byte.

Bit 6 Start bit Control_Active_Configuration_Req Enable
The station set as parameter is toggled active again within the configuration frame. It must also physically included back into the data ring.
Parameter: See Bit 5

Bit 14 Application-Busy-Bit (at bus synchronous operating mode) res. Data-Cycle-Activate-Bit (at program synchronous operating mode)
Bit 15 Cons-Activate-Bit for the consistency lock

The bits 14 and 15 serve the processing of protocols for the process data exchange between the IBS master and the CPU.

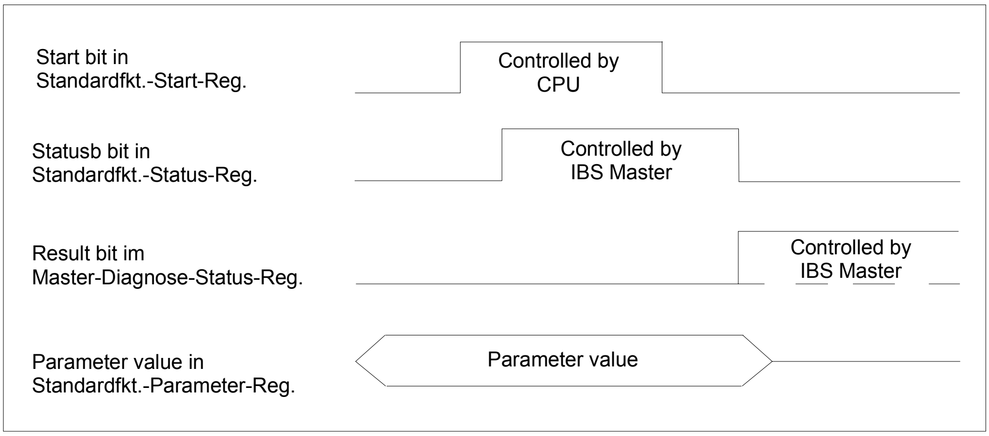
Standard fct.-status register Master > CPU (LADDR+18)

The bits 0...6 of this registers are used by the IBS master to monitor and control the processing of standard functions activated in the standard functions start register. Bit 15 serves the processing of a protocol for the process data exchange between IBS master and the CPU.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	res.	res.	res.	res.	res.	res.	res.	res.	x	x	x	x	x	x	x

Bit 0 : Status bit Start_Data_Transfer_Request
Bit 1 : Status bit Alarm_Stop_Request, Activate_Configuration_Request
Bit 2 : Status bit Confirm_Diagnostics_Request
Bit 3 : Status bit Control_Active_Configuration_Req Off
Bit 4 : Status bit Control_Active_Configuration_Req On
Bit 5 : Status bit Control_Active_Configuration_Req Disable
Bit 6 : Status bit Control_Active_Configuration_Req Enable
Bit 15: Cons-State-Bit for consistency lock

Execution of a standard function with parameter transfer



The diagram in the picture above shows the handshake mechanism at usage of the standard functions. A "0" in Bit 10 (RESULT) of the master diagnosis status register shows that the standard function has been finished successful.

Master diag. param register Master > CPU (LADDR+20)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

This register monitors depending on the type of the error the error code or the error location. The content of the register is managed by the IBS master. Some error types cause additional entries in the *Extended master diagnosis parameter register*. The contents of the extended master diagnosis parameter register is to be found as word under address 168.0 in the work DB.

Master diag status register Master > CPU (LADDR+22)

This register contains information about the state of the IBS master. The table contains the meaning of the bits when set ("1"). The content of the registers is managed by the IBS master. In case of an error additional information is available in the master diagnosis parameter register and in the extended master diagnosis parameter register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	x	x	res.	res.	x	x	x	x	x	x	x	x	x	x	x

Bit 0 (USER)	User/Parameterization error
Bit 1 (PF)	Periphery failure
Bit 2 (BUS)	Bus failure
Bit 3 (CTRL)	Error at the IBS master
Bit 4 (DETECT)	Diagnosis routine is active
Bit 5 (RUN)	Data transfer is active
Bit 6 (ACTIVE)	Selected Interbus configuration ready for operation
Bit 7 (READY)	IBS master ready for operation
Bit 8 (BSA)	Bus segment(s) shut down
Bit 9 (BASP/SYSFAIL)	Function failure of the CPU detected; outputs at the IBS set back
Bit10 (RESULT)	Negative result of a standard function
Bit13 (WARNING)	Defined bus waiting period exceeded
Bit14 (QUALITY)	Defined error density exceeded (is set at more than 20 failures per 1 million IBS cycles)

Slave diag. status register Master > CPU (LADDR+26)

This register contains information about the state of the optional slave interface to a hierarchical super-ordinated Interbus network. The content of the register is managed by the IBS master.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	x	x	x	x	x

Bit 0 COPY
1 Data between IBS master and slave interface are exchanged. The super-ordinated Interbus network is operating.
0 No data between IBS master and slave interface are exchanged. The super-ordinated Interbus network is not operating.

Bit 1 FAIL
1 The super-ordinated IBS network has been stopped by a bus error or alarm. No data is exchanged with the slave interface anymore. The output data of the slave interface are set to "0".
0 No error in the super-ordinated Interbus network.

Bit 2 READY-TO-COPY
1 The parameterization of the slave interface has been finished successful.
0 The slave interface has not been parameterized yet.

Bit 3 POWER-ON
1 The power supply of the slave interface is on.
0 The power supply of the slave interface is off.

Bit 4 READY
1 The content of the slave diagnosis status register has been initialized.
0 The content of the slave diagnosis status register has been not yet initialized.

Configurations register Master > CPU (LADDR+28)

In this register it is monitored if the IBS master has finished a parameterization process storage initialized or from operator panel (IBS SWT CMD G4 from Phoenix Contact).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	x	res.

Bit 1 : DPM-Node-Par-Ready 1

1 IBS master is parameterized.

0 IBS master is not parameterized.

If a parameterization has been stored in the parameterization memory of the IBS master, the IBS master starts the execution of the stored instructions as soon as it reaches the state READY. Bit 1 is set by the IBS master after all instructions of the parameterization memory has been processed.

Status sysfail register Master > CPU (LADDR+32)

This register shows a function failure of the CPU that may occur.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res.	res.	res.	x	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.

Bit 12 : 1 function failure of the CPU.

0 no function failure of the CPU.

This bit is set by the IBS master when a function failure of the CPU is announced by the interrupt IRQHOSTL. In this case, all outputs of the Interbus stations are set to "0". Additionally the diagnosis-LED "HF" is on.

Project engineering

Outline

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, the inclusion of the GSD SPEEDBUS.GSD is required.

Every "VIPA_SPEEDBUS" DP slave has exactly one slot for the project engineering where the according SPEED-Bus module must be placed. The assignment of a SPEED-Bus slave to a SPEED-Bus slot no. takes place via the Profibus address starting with 100.

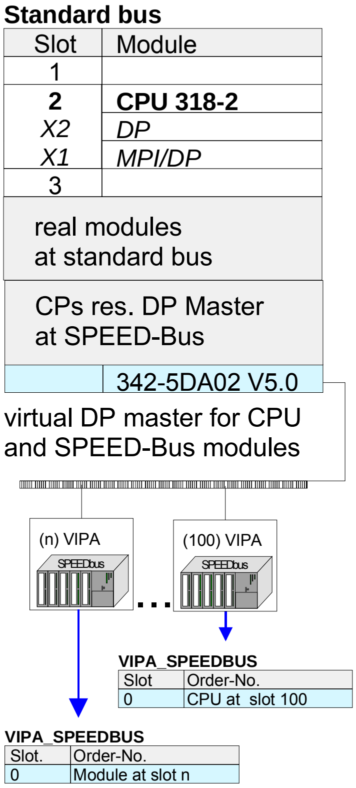
Include the SPEED7-GSD-file

- Extract the delivered vipa gsd file SPEEDBUS.GSD to your gsd directory... \siemens\step7\s7data\gsd
- 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.

Fast introduction

For the deployment of modules at the SPEED-Bus the inclusion of the System 300S modules via the GSD-file from VIPA in the hardware catalog is required.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
real modules at standard bus	
CPs res. DP Master at SPEED-Bus	
	342-5DA02 V5.0

VIPA_SPEEDBUS

Slot	Order-No.
0	CPU at slot 100

VIPA_SPEEDBUS

Slot.	Order-No.
0	Module at slot n

To be compatible with the SIMATIC Manager from Siemens, 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 DP master at the standard bus as virtual 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 IBS master. Here the Profibus address corresponds to the slot no. Beginning with 100 for the CPU. Place on slot 0 of every slave the assigned module and alter the parameters if needed.
- For Interbus communicaton:
  - Include IBS-FCs (ftp.vipa.de/support/library/ FX000002_Vxxx.ZIP)
  - Create a work DB for every IBS master
  - Build-up a PLC communication program

  More information may be found at the following pages.

Interbus configuration

Overview

The configuration and diagnosis happens by means of FCs that are provided under ftp.vipa.de.

At deployment of the FCs you have to create a work DB for every IBS master which the FCs use to communicate.

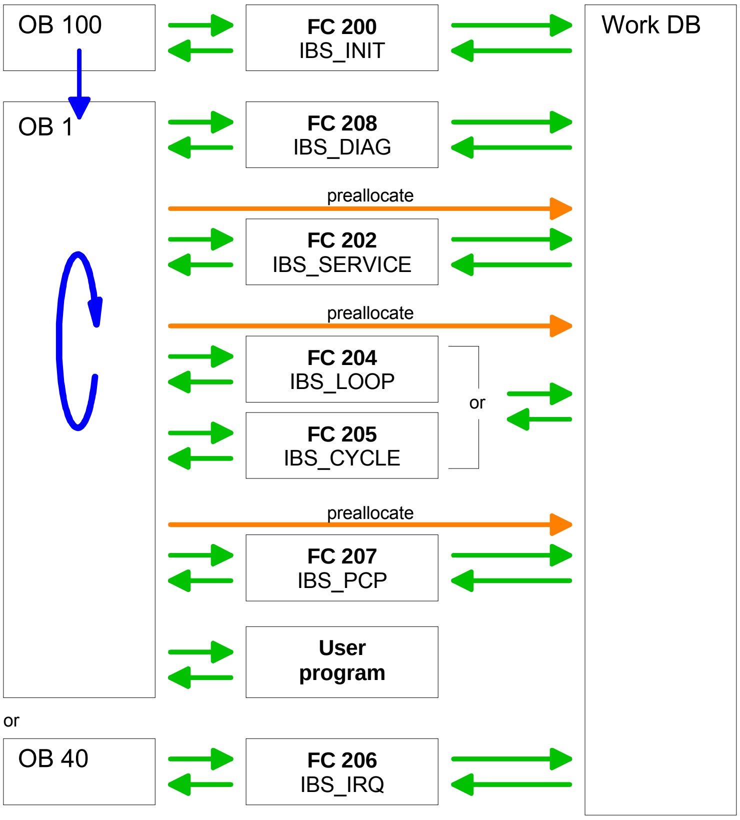
Every IBS master occupies with its register each 34byte in the in-/output address range of the CPU.

Initialization, diagnosis and data exchange between CPU and IBS master happens via "services" that are transferred by means of FCs.

For the transmission of instructions and parameters to an IBS slave, the "**P**eripherals **C**ommunication **P**rotocol" (PCP) is at your disposal where the transfer also happens with a FC.

Fast introduction

- Include IBS-FCs (ftp.vipa.de/support/library/ FX000002_Vxxx.ZIP)
- Create a work DB for every IBS master
- Build-up PLC program following this structure:



Note!

Before calling the according FCs you have to provide the work DB with parameters!

Include FCs

The VIPA specific FCs are available in a library on the ftp server from VIPA.

For the deployment of these FCs you have to import them into your project.

Approach

The following steps are required:

- Browse to ftp.vipa.de
- Load "FX000002_Vxxx.ZIP" from Support/Library
- Extract and *de-archivate* "VIPA.ZIP"
- Open library and transfer the FCs from the IBS directory to the project

De-archivate library

To de-archivate the FC library you start the SIMATIC manager from Siemens. Open a dialog window for archive selection via **File** > *De-archivate*. Choose VIPA.ZIP and click on [Open].

Set a destination directory where the blocks shall be stored. Click [OK] to start the process.

Open library and transfer FCs to project

After the extraction, start the library.

Open your project and copy the needed IBS-FCs from the library to the directory "Blocks" of your project.

Now you have access to the VIPA specific Interbus blocks via the user application.

Handling blocks for IBS communication

The deployment of the IBS master at the SPEED-Bus happens via the following handling blocks:

Block	Name	Description
FC 200	IBS_INIT	Registration and initialization of an Interbus master at the CPU
FC 202	IBS_SERVICE	Service communication between CPU and IBS master
FC 204	IBS_LOOP	Slow asynchronous data communication between CPU and IBS master (waits for master release)
FC 205	IBS_CYCLE	Fast asynchronous data communication between CPU and IBS master (waits not for master release)
FC 206	IBS_IRQ	Synchronous data communication between CPU and IBS master with synchronization via interrupt
FC 207	IBS_PCP	Peripherals Communication Protocol (PCP) communication for instructions and parameters for IBS slaves
FC 208	IBS_DIAG	Read diagnostic data from IBS master res. IBS slaves
SFC 254	RW_SBUS	Communication block, required for usage of the FCs

Requirements

For the deployment of the handling blocks every IBS master must fulfill the following conditions:

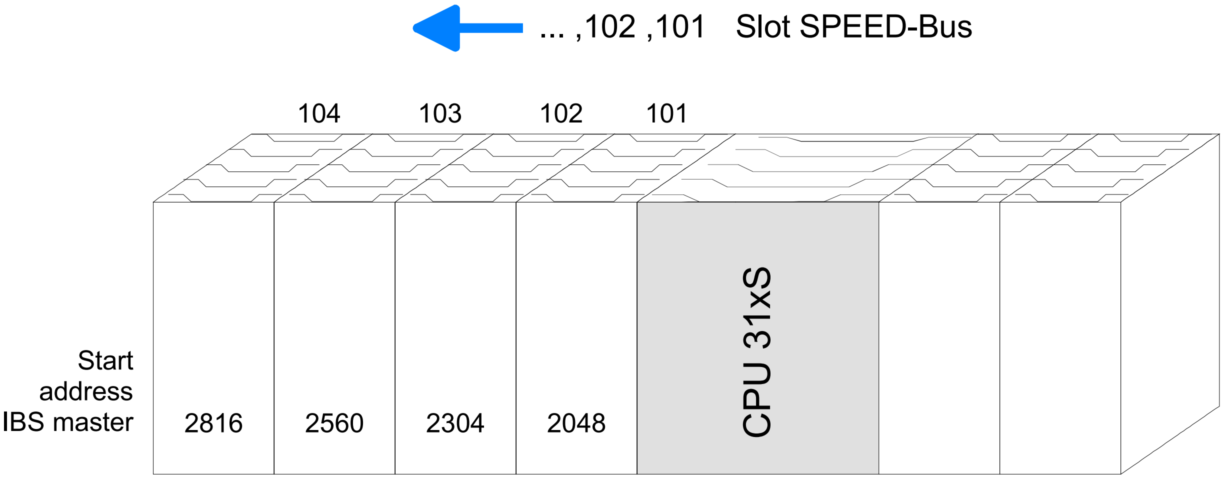
- Set start address from where on the IBS master is mapped into the address range of the CPU.
- Create work DB which the according FCs use to communicate.

Setting of the start address

The IBS masters are mapped automatically into the address range of the CPU at start-up following the formula:

Start address = 256· (Slot-101)+2048

All information is to be found in the following illustration:



Change address

The automatical addressing is used when no hardware configuration is present.

A hardware configuration allows you to alter the addresses at any time by including the SPEEDBUS.GSD. For this click on the properties of the according IBS master module and set the wanted address.

The approach of the hardware configuration of the IBS master is to be found above under "Interbus master for SPEED-Bus – Hardware configuration".

Structure of the "Work DB"

You have to create a work DB for every IBS master. You may download this DB together with a sample project in the service area of ftp.vipa.de.

The following table shows the structure of the work DB. Parameters that must be set before calling the according FC are marked gray:

Addr.	Name	Type	Comment
0.0	free	BYTE	
1.0	free_1	BYTE	
2.0	Addr_INT_Host_Mas	DWORD	Address of the interrupt Host-->Master 0xFFF
6.0	Addr_INT_Mas_Host	DWORD	Address of the interrupt Master-->Host 0xFFE
10.0	Addr_SSGI_Ack	DWORD	Address SSGI acknowledge 0xFDE
14.0	Addr_SSGI_Notif	DWORD	Address SSGI notification 0xFE0
18.0	Addr_SSGI_Result	DWORD	Address SSGI result 0xFE2
22.0	Addr_SSGI_Status	DWORD	Address SSGI status 0xFE4
26.0	Addr_SSGI_Start	DWORD	Address SSGI start 0xFE6
30.0	reserved	DWORD	
34.0	Addr_Stand_Fct_Param	DWORD	Address standard function parameter 0xFEA
38.0	Addr_Stand_Fct_Start	DWORD	Address standard function start 0xFEC
42.0	Addr_Stand_Fct_Status	DWORD	Address standard function status 0xFEE
46.0	Addr_Master_Diag_Param	DWORD	Address Master diagnosis parameter 0xFF0
50.0	Addr_Master_Diag_Status	DWORD	Address Master diagnosis status 0xFF2
54.0	reserved_2	DWORD	
58.0	Addr_Slave_Diag_Status	DWORD	Address Slave diagnosis status 0xFF6
62.0	Addr_Configuration	DWORD	Address Configuration 0xFF8
66.0	reserved_3	DWORD	
70.0	Addr_Status_Sysfail	DWORD	Address status system error 0xFFC
74.0	SSGI_Ack	WORD	Register value SSGI Acknowledge
76.0	SSGI_Notif	WORD	Register value SSGI Notification
78.0	SSGI_Result	WORD	Register value SSGI result
80.0	SSGI_Status	WORD	Register value SSGI status
82.0	SSGI_Start	WORD	Register value SSGI start
84.0	reserved_4	WORD	
86.0	Stand_Fct_Param	WORD	Register value standard function parameter
88.0	Stand_Fct_Start	WORD	Register value standard function start
90.0	Stand_Fct_Status	WORD	Register value standard function status
92.0	Master_Diag_Param	WORD	Register value Master diagnosis parameter
94.0	Master_Diag_Status	WORD	Register value Master diagnosis status
96.0	reserved_5	WORD	
98.0	Slave_Diag_Status	WORD	Register value Slave diagnosis status
100.0	Configuration	WORD	Register value Configuration
102.0	reserved_6	WORD	
104.0	Status_Sysfail	WORD	Register value status system error
106.0	Step_Counter_Service	INT	Step counter for FC 202 "process services"
108.0	RET_VALSEND_Service	WORD	Return value of the SFC 254 at send command via FC 202
110.0	RET_VALRECEIVE_Service	WORD	Return value of the SFC 254 at read command via FC 202
112.0	Error_Byte_Service	BYTE	Error ID of FC 202
113.0	Number_Service_Error	BYTE	Number of the service where the error has been detected.
114.0	Return1_Function_Service	WORD	Error code 1 return value of the service
116.0	Return2_Function_Service	WORD	Error code 2 return value of the service
118.0	Number_Services	BYTE	Number of services to process for FC 202
119.0	Processed_Services	BYTE	Number of processed services
120.0	Waiting_Receipt	WORD	Interim storage of expected acknowledgement
122.0	Start_Services	BYTE	Number of service that is 1. to process
124.0	Waiting_Time	S5TIME	Waiting period for acknowledgements
126.0	Timer_No	WORD	Timer number for waiting period
128.0	Extended_Diagnosis	BYTE	Bit 0 bit memory bit when extended diagnosis requested
129.0	Additional03	BYTE	
130.0	DB_No_Write	WORD	DB_No. of output data (for FC204/FC205 if data in DB)
132.0	Start_Data_In	WORD	Address of 1st output byte (for FC 204/205)
134.0	Length_Data_In	WORD	Length of output data (for FC 204/205)

continued ...

... continue work DB

136.0	Start_Data_DPM_In	WORD	Start address of output data in DPM (for FC 204/205)
138.0	DB_No_Read	WORD	DB-No. of input data (for FC 204/205 if data in DB)
140.0	Start_Data_Out	WORD	Address of 1st input byte (for FC 204/205)
142.0	Length_Data_Out	WORD	Length of input data (for FC 204/205)
144.0	Start_Data_DPM_Out	WORD	Start address of input data in DPM (for FC 204/205)
146.0	RET_VAL_DATEN_SEND	WORD	Return value of the SFC 254 at writing data via FC 204/205
148.0	RET_VAL_DATEN_REC	WORD	Return value of the SFC 254 at reading data via FC 204/205
150.0	Error_Data_L_S	WORD	Error byte of the FC 204/FC205
152.0	Step_FC205	BYTE	Read/Write step counter I/O of FC 205
153.0	Additional013	BYTE	
154.0	Step_FC208	WORD	Step counter of FC 208
156.0	Order_Diag_316	WORD	Enter command for transmission (316 fix)
158.0	Parameter_Order_316	WORD	Number of parameters for command (316 fix)
160.0	Error_Service_FC208	BYTE	Error byte of the FC 208
161.0	Control_bit_FC208	BYTE	Control bits of the FC 208
162.0	Waiting_period_Auto	S5TIME	Waiting period at auto start after error
164.0	RET_VAL_SFC_FC208	WORD	Return value of the SFC 254 at read/write data via FC 208
166.0	Timer_FC208	WORD	Number additional timer (Number = Timer_No +1)
168.0	Extension_Diagnosis_Param	WORD	Diagnostic addition to master diagnosis parameter register
170.0	Number_Bus_Errors	INT	Error counter of all bus disruptions
172.0	Number_IBSUSC4_Errors	INT	Error counter of all IBS USC4 errors
174.0	Number_Peripherieerrors	INT	Error counter of all periphery errors
176.0	Number_User_Errors	INT	Error counter of all user errors
178.0	Order_Diag_315	WORD	Enter command for transmission (315 fix)
180.0	Parameter1_Order_315	WORD	Number of parameters for command (315 fix)
182.0	Parameter2_Order_315	WORD	Parameters for command 315
184.0	Waiting_Period_Detection	S5TIME	Waiting period for detection
186.0	Step_Counter_PCP	INT	Step counter of FC 207
188.0	RET_VALSEND_PCP	WORD	Return value of the SFC 254 at send command via FC 207
190.0	RET_VALRECEIVE_PCP	WORD	Return value of the SFC 254 at read command via FC 207
192.0	Error_Byte_PCP	BYTE	Error byte of the FC 207
193.0	Number_PCP_Errors	BYTE	Number of the PCP where error has been detected
194.0	DW_Counter_PCP	DWORD	Number of error codes returned from the PCP
198.0	Number_PCP	BYTE	Number of PCP to be processed for FC 207
199.0	Processed_PCP	BYTE	Number of already processed PCP
200.0	Waiting_Receipt_PCP	WORD	ID of expected acknowledgement
202.0	Start_PCP	BYTE	Number of 1st PCP to be processed
204.0	Error1_PCP	WORD	Error code 1 Return value of the PCP
⋮	⋮	⋮	⋮
218.0	Error8_PCP	WORD	Error code 8 Return value of the PCP
220.0	Address_SFC254	WORD	Module address for SFC 254
222.0	Additional110	BYTE	
⋮	⋮	⋮	⋮
249.0	Additional137	BYTE	
250.0	Diagnosis_Bus_Error[1]	BYTE	Entry of extended diagnosis at bus error
⋮	⋮	⋮	⋮
299.0	Diagnosis_Bus_Error[50]	BYTE	
300.0	Diagnosis_IBSUBC4_Error[1]	BYTE	Entry of extended diagnosis at IBS UBC4 error
⋮	⋮	⋮	⋮
349.0	Diagnosis_IBSUBC4_Error[50]	BYTE	
350.0	Diagnosis_Periph_Error[1]	BYTE	Entry of extended diagnosis at periphery error
⋮	⋮	⋮	⋮
399.0	Diagnosis_Periph_Error[50]	BYTE	
400.0	Diagnosis_USER_Error[1]	BYTE	Entry of extended diagnosis at user error
⋮	⋮	⋮	⋮
449.0	Diagnosis_USER_Error[50]	BYTE	

Program structure

The Interbus functions have to be called at boot of the CPU and in the cyclic program by means of conditional or absolute jumps.

You have to include the FC 200 in the boot sequence. This FC synchronizes the IBS master with the CPU and checks the structure of the connected in- and output bytes as well as the bus structure.

Via the FC 208 you may read diagnostic data of the master res. the slaves in the cyclic program. This block also sets the starting type of the IBS master after an error.

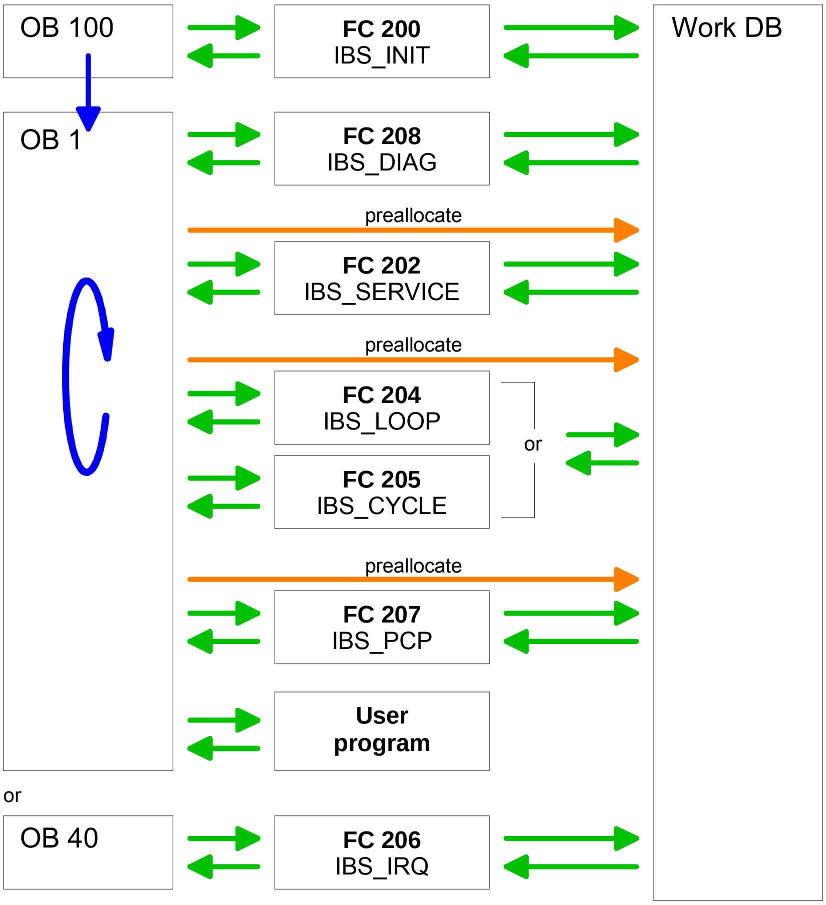
You may parameterize the IBS master via the FC 202. For this you have to transmit a DB that may contain up to 30 service instructions. Before this you must enter the number of services in the work DB under "Number_services".

By calling FC 204 or FC 205 the asynchronous data exchange between IBS master and the CPU starts. Both FCs have the same request parameters. The FC 204 waits after the data request for the data release of the IBS master and then continues the cycle process. In opposite to FC 204, the FC 205 does not wait. As long as no data release is present, it continues the cycle processing. Thus the cycle processing of the CPU is not interrupted.

You may also synchronize the data transfer by using the FC 206 instead and call this within a HW-Interrupt-OB. Here the IBS master announces new data via an interrupt. Reading of data by the CPU is also signalized via an interrupt.

User application

Your user application should have the following structure:



Function blocks

In the following you will find a more detailed description of the function blocks that are required for the Interbus communication.

FC 200 IBS_INIT

This FC synchronizes the IBS master with the CPU and checks the number of connected in- and output bytes as well as the bus structure.

Parameter

Parameter	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
LADDR	IN	INT	Logical address of IBS master
MODE	IN	INT	Mode for operation
WAIT_TIME	IN	S5TIME	Wait time for IBS Master receipt
TIMER_NO	IN	INT	Timer No. for wait time
SERVICE_DB_SEND	IN	INT	DB No. with services
SERVICE_DB_REC	IN	INT	DB No. with service receipts
NO_OF_SERVICES	IN	WORD	No. of services to be processed
READ_DIAG	IN	BOOL	Read Diagnostic data
RET_VAL	OUT	WORD	Return value of error number
FIRST_SERVICE	IN_OUT	BYTE	No of 1st service to be processed

WORK_DB

Set the work DB for the wanted master.

LADDR

Set the address (**L**ogical **Addr**ess) from where on the register of the masters is to be mapped into the CPU. At start-up of the CPU, the IBS master are mapped into the I/O address range of the CPU with the following formula if no hardware configuration is present:

Start address = 256· (Slot-101)+2048

The slot numbering at the SPEED-Bus starts with 101 at the left side of the CPU and ascends from the right to the left. For example, the 1st slot has the address 2048, the 2nd the address 2304 etc.

MODE

This parameter allows you to preset 3 modes for start-up:

0 = Calculate address only

1 = Calculate address and wait for Ready of the IBS master

2 = Calculate address, parameterize and start IBS master

3 = Calculate address and automatical start of Interbus after auto-configuration via switch

WAIT_TIME
TIMER_NO

Here you may define a waiting period with the according timer by setting *WAIT_TIME* and *TIMER-NO* that the CPU has to wait for a master acknowledgement after a service command.

Note!

Please regard at setting a timer-No. That always 2 sequential timers are used:

Timer 1: *TIMER_NO*, Timer 2: *TIMER_NO* + 1

SERVICE_DB_SEND
SERVICE_DB_REC

Enter the DB that contains the according service instructions via *SERVICE_DB_SEND*. In *SERVICE_DB_REC* the IBS master returns the receipt.

More details about the structure of the service DB are to be found on the following page under "FC 202 Process service".

NO_OF_SERVICES
FIRST_SERVICE

In *NO_OF_SERVICES* you enter the number of services that have to be processed in the service DB after the 1st service that you set in *FIRST_SERVICE*.

READ_DIAG

This parameter allows you to influence the structure of a diagnosis:

0 = Normal diagnosis

1 = Extended diagnosis

RET_VAL

In case of an error, *RET_VAL* may contain the following error messages:

1 = Waiting period for master receipt (READY) exceeded - master not ready

2 = Execution of a service to process has failed

FC 202
IBS_SERVICE

This function block allows you to transfer services to the IBS master and to react to the according acknowledgements.

For the Interbus master card USC4-1 from Phoenix Contact is deployed as Interbus hardware platform, please also refer to the extensive documentation (IBS SYS FW G4 UM) from Phoenix Contact for the description of the IBS services and IBS error messages.

Parameter

Name	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
SERVICE_DB_SEND	IN	INT	DB No. with services
SERVICE_DB_REC	IN	INT	DB No. with service receipts
FIRST_SERVICE	IN	BYTE	No of 1st service to be processed
START	IN_OUT	BOOL	Start bit of the function
ERROR	IN_OUT	BOOL	Error bit of the function

WORK_DB

Set the work DB for the wanted master.

SERVICE_DB_SEND
SERVICE_DB_REC

Enter the DB that contains the according service instructions via *SERVICE_DB_SEND*. In *SERVICE_DB_REC* the IBS master returns the receipt.

FIRST_SERVICE

Enter the position of the first service within the send DB.

Note!

Please regard that you have to enter the number of services that are to be transferred after *FIRST_SERVICE* in the work DB before calling the FC 202.

Structure service DB

You may enter a max. of 30 services in one DB. Up to 2 DBs, 60 services in total, may be transferred to the IBS master at every FC call.

DBB	Contents
0 ... 69	Record set 1
70 ... 139	Record set 2
:	:
2030 ... 2099	Record set 30
2100	Instruction no. 2. DB

Structure record set

DBW	Contents
0	Send length (Number of bytes to be send)
1	Code No. of service
2	Parameter count
3 ... 68	Parameter

START

By setting the start bit, the services are transferred to the IBS master and started.

ERROR

In case of an error, the start bit is set back and the error bit is set. Additionally, the number of the service that has been processed when the error occurred is entered in the DBB 113 of the work DB. The error code is displayed in DBB 112.

The following error codes may occur:

2 = Error of the master at reading data from SSGI Box

3 = Return code of the acknowledgement not valid

4 = Service could not be processed

5 = No acknowledgement within waiting period

Note!

If DBB 112 contains the error code 4, further error codes are entered into DBW 114 and 116 of the work DB.

Information about these error codes is to be found in the documentation of the services (IBS SYS FW G4 UM) from Phoenix Contact.

FC 204
IBS_LOOP
FC 205
IBS_CYCLE

The FC 204 serves the exchange of in- and output data between IBS master and CPU. This block always awaits an acknowledgement of the master after a data request and continues the cycle processing only after reception.

If this block influences the cycle processing of the CPU too much, you should use the FC 205 Asynchr_Cycle instead. In opposite to the FC 204 this does not wait for an acknowledgement but continues cycle processing after data request.

Occurring error messages are to be found after block processing in the work DB in DBW150.

Parameter

Parameter	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
RW_MODE	IN	INT	Mode of R/W (0=R/W, 1=R, 2=W)
OPERATION_ MODE	IN	INT	Operation mode (0=asynchr., 1=asynchr. with consistency)
TYP_OUT	IN	INT	Data type of IBS slave out data (0=DB, 1=MB, 2=OB)
TYP_IN	IN	INT	Data type of IBS slave in data (0=DB, 1=MB, 2=IB)
START	IN_OUT	BOOL	Start bit of the function

WORK_DB

Set the work DB for the wanted master.

RW_MODE

The following modes are available:

0 = Read input data and write output data

1 = Read input data only

2 = Write output data only

OPERATING_MODE

The transfer may happen with the following operating modes:

0 = Asynchronous data exchange without consistency lock

In this operating mode it may happen that read res. written data is not out of the same Interbus cycle and is therefore inconsistent.

1 = Asynchronous data exchange with consistency lock

Here the CPU sets a bit for read/write request. As soon as the next Interbus cycle is finished and data is ready, the IBS master sets a release bit. The CPU transfers its data and signalizes the end of data transfer by setting back the request. Now the IBS master deletes the release and continues the Interbus cycle.

TYP_OUT
TYP_IN

This parameter defines the type of the data area where the I/O data of connected IBS slaves is stored.

The following types are available:

0 = DB (data block)

1 = MB (bit memory byte)

2 = I/O range of the CPU

START

By setting the start bit, the FC is executed. The start is set back again in the block.

Error message

During the execution of the block, the following errors that are stored in DBW 150 of the work DB may occur:

1 = Data release of the master missing – read inputs

2 = Data release of the master missing – write outputs

3 = Data release of the masters is not deleted

FC 206
IBS_IRQ

At deployment of the FC 206, the data transfer of the in- and output data between CPU and IBS master is controlled via interrupts.

As soon as the IBS master has provided its data, it initializes an interrupt. The CPU transfers its data and also signalizes the end of the data transfer via an interrupt. Now the IBS master continues the Interbus cycle.

Parameter

Parameter	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
RW_MODE	IN	INT	Mode of R/W (0=R/W, 1=R, 2=W)
TYP_OUT	IN	INT	Data type of IBS slave out data (0=DB, 1=MB, 2=OB)
TYP_IN	IN	INT	Data type of IBS slave in data (0=DB, 1=MB, 2=IB)

WORK_DB

Set the work DB for the wanted master.

RW_MODE

The following modes are available:

0 = Read input data and write output data

1 = Read input data only

2 = Write output data only

TYP_OUT
TYP_IN

This parameter defines the type of the data area where the I/O data of connected IBS slaves is stored.

The following types are available:

0 = DB (data block)

1 = MB (bit memory byte)

2 = I/O range of the CPU

FC 207
IBS_PCP

This function block allows you to transfer PCP services to the IBS master and to react to the according acknowledgements. The **P**eripherals **C**ommunication **P**rotocol (PCP) serves the transmission of instructions and parameters to connected slaves and the reception of acknowledgements and data of the slaves.

Information about the services is to be found in the documentation of the services, available via our application department.

Parameter

Parameter	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
SERVICE_DB_SEND	IN	INT	DB No. with services
SERVICE_DB_REC	IN	INT	DB No. with service receipts
FIRST_SERVICE	IN	BYTE	No of 1st service to be processed
START	IN_OUT	BOOL	Start bit of the function
ERROR	IN_OUT	BOOL	Error bit of the function

WORK_DB

Set the work DB for the wanted master.

SERVICE_DB_SEND
SERVICE_DB_REC

Enter the DB that contains the according PCP service instructions via *SERVICE_DB_SEND*. In *SERVICE_DB_REC* the slaves return the receipt.

FIRST_SERVICE

Enter the position of the first PCP service within the send.

Note!

Please regard that you have to enter the number of services that are to be transferred after *FIRST_SERVICE* in the work DB before calling the FC 207.

Structure
service DB

You may enter a max. of 30 PCP services in one DB. Up to 2 DBs, 60 PCP services in total, may be transferred to the IBS master at every FC call.

DBB	Content
0 ... 69	Record set 1
70 ... 139	Record set 2
.	.
.	.
.	.
2030 ... 2099	Record set 30
2100	Sequence no. of 2. DB

Structure record set

DBW	Content
0	Send length (Number of bytes to be send)
1	Code No. of PCP service
2	Parameter count
3 ... 68	Parameter

START

By setting the start bit, the PCP services are transferred to the IBS master and started.

ERROR

In case of an error, the start bit is set back and the error bit is set. Additionally, the number of the PCP service that has been processed when the error occurred is entered in the DBB 193. The following error codes may be entered into DBB 192:

2 = Error of the master at reading data from SSGI Box

3 = Return code of the acknowledgement not valid

4 = Service could not be processed

5 = No acknowledgement within waiting period

Note!

If Error contains the error code 4, further error codes are entered into DBW 194 and 196 of the work DB.

Information about these error codes is to be found in the documentation of the error codes, available via our application department.

FC 208
IBS_DIAG

Via this function block you may read diagnostic data from the master res. slave in case of an Interbus break-down. Here you may also define the reboot operating mode of the IBS master after break-down.

Parameter

Parameter	Declaration	Type	Description
WORK_DB	IN	BLOCK_DB	IBS work DB
ACTIVATE	IN	INT	Manual error receipt
AUTO_START	IN	INT	Automatic error receipt
RUN	OUT	BYTE	IBS at status RUN
PERIPHERAL_ERROR	OUT	BOOL	Error at periphery
BUS_QUALITY	OUT	BOOL	Sporadic bus errors occurred
DETECTION	OUT	BOOL	Bus error is searched
BUSY_STATE	OUT	BOOL	Diagnostic function is busy

WORK_DB

Set the work DB for the wanted master.

ACTIVATE
AUTO_START

The *ACTIVATE* transmission parameter of the type Boolean that you may control for example via an external caliper, allows you to reboot the IBS master by setting (push button).

By setting of auto-start, the IBS master reboots automatically after error recovering. *AUTO-START* has always preference before *ACTIVATE*.

RUN

This parameter shows the status of the IBS master:

0 = IBS master is in STOP

1 = IBS master is in RUN

PERIPHERAL_ ERROR

If an periphery error occurs, the IBS master announces PF = 1. At PF = 0 no periphery error occurred.

In case of an error you will see the number of the causing slave in the work DB starting with 1.

BUS_QUALITY

This parameter displays information about the transfer quality within the Interbus. As soon as the bit is set by the IBS master, some single transmission interferences have occurred. Please check the transfer routes with according diagnosis software.

DETECTION

The parameter *DETECTION* is set by the IBS master when the internal error detection is running. When the error detection is finished, *DETECTION* is set back again.

BUSY_STATE

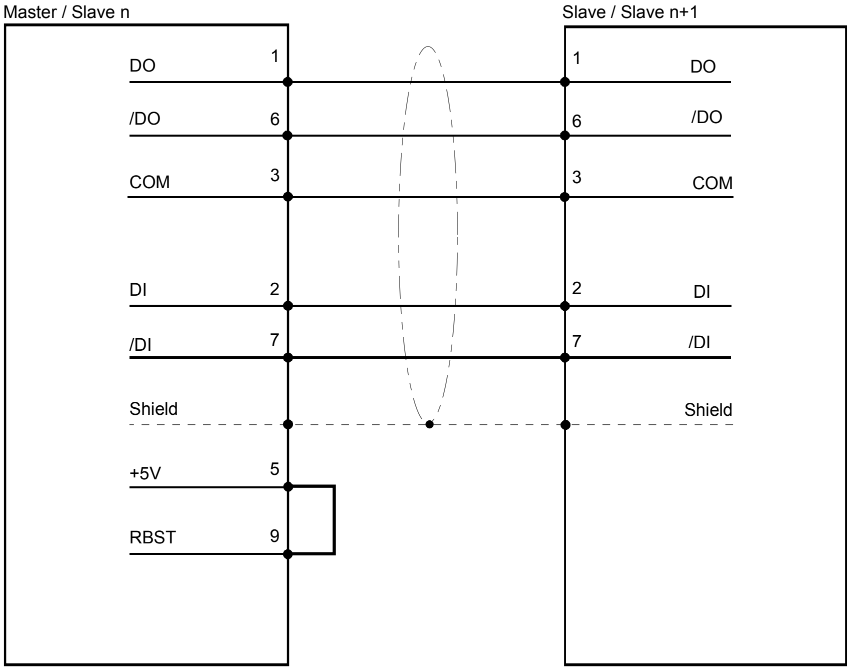
When a diagnosis is executed within the diagnosis block, *BUSY_STATE* is set. As soon as diagnosis data are available, the block sets *BUSY_STATE* back again.

Connection

Cabling with Interbus

Although Interbus is substantial build-up in line structure (only one line from the master to the last module) it is in principle a ring structure where the for- and backwards conductor are together in one lead. The ring is closed by the last participant. The most devices are closing the ring automatically when no continuative lead is connected.

For master-slave and slave-slave connection the same connection cable is used. Due to the ring structure and the common logic ground, the cable consists of 5 cores and has the following assignment:



Note!

Please take care that the plug for the „continuative interface" has a bridge between Pin 5 and 9, otherwise the following slaves are not recognized!

Potential separation

For Interbus distant bus segments cover a wide volume expansion, the single segments must be galvanically separated to avoid potential procrastination. According to the recommendation of the Interbus-Club, a galvanically separation of the incoming distant bus interface from the rest of the control is sufficient. The continuative distant bus interface is thus at the potential of the rest control and the backplane bus.

Please use metal plug casings and put the cable screen on the plug casing.

Example

Overview

The following sample shall illustrate a communication between a SPEED-Bus-IBS master and a connected IBS slave.

You may find the example at ftp.vipa.de at *support/demo_files* as "speed7_ibs_pr_1.zip". After the download you may load the zip file as project into the SIMATIC Manager from Siemens via **File** > *De-archivate*.

Properties

The sample project provides the following features:

- Appropriate for deployment at a VIPA IBS master CP 342S-IBS
- The addressing of the IBS master happens automatically (no hardware configuration). The IBS master must be at the 1st slot at the SPEED-Bus for it works with module address 2048. Otherwise you have to adjust the LADDR parameter for the FC 202 accordingly.
- For the IBS configuration is newly detected at reboot, the number of connected IBS slaves is irrelevant.
- During the hardware configuration the Ethernet PG/OP channel is assigned to the IP address 172.16.129.71. With this IP address you may access the CPU online via the Ethernet PG/OP channel. Please consider to adjust the IP address accordingly if you are working with another number circle.

Program structure

For the program has comment at according lines, here you will only find the basic structure.

OB 100
Boot/Reboot

- Presetting of services for automatic start.

Service 1303h	Stop bus
Service 0710h	Read configuration automatically
Service 0701h	Start bus communication (Master→Slave)

- Call function for initialization of the master
- Set control bit for cyclical reading
- Set control bit for automatical start after error

OB 1

Call FC 1000

OB 40

Call sample at interrupt controlled data transfer

FC 1000 Master processing

- Read diagnosis
- Detect status IBS master (RUN/STOP)
- Call the function for service processing
- Preset data in work DB (DB 120) for read and write accesses of the IBS I/Os
- Read and write IBS I/Os
- Initialize again read and write

DB 10, UDT1, DB 11

For every service, an entry of the type „service" is to be found in *DB 10*. The data type of "service" is defined under *UDT1*.

DB 11

Acknowledgement DB is the DB 11.

DB 110

If you want to transfer more than 30 services, you may set a pointer to another service DB in DB 10 (here DB 110).

This is monitored in OB 100.

SFC 254

The SFC 254 is required for the communication between CPU and IBS master. The call of the SFC 254 happens from IBS-FCs
(FC 200 ... FC 208).

SFC 1

The system block SFC 1 is automatically created when you call the FC 208 "Diagnostic".

Appendix

A Index

M.Stich