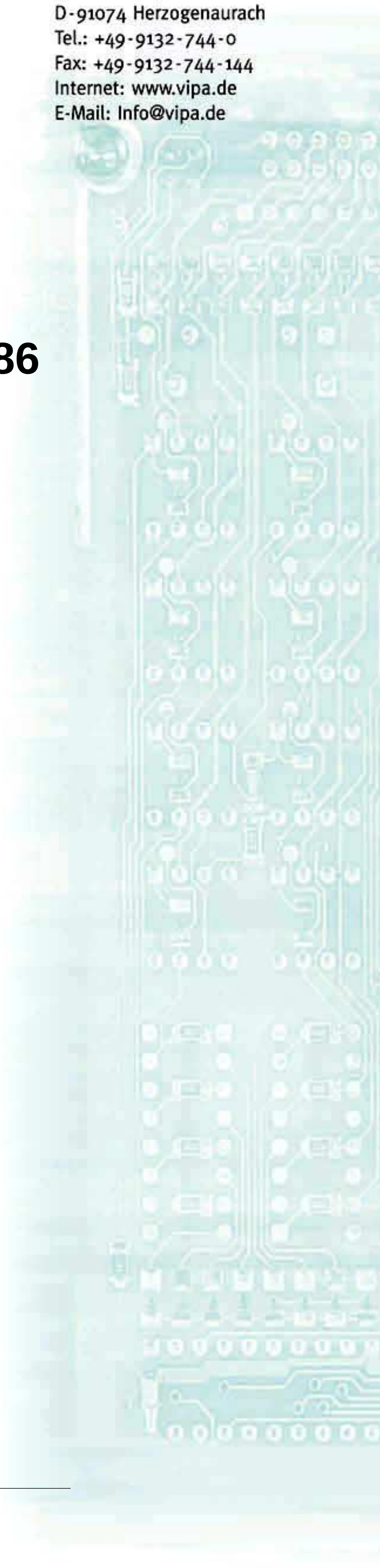
Handbücher/Manuals

VIPA
art of automation

VIPA
Gesellschaft für Visualisierung
und Prozessautomatisierung mbH

Ohmstraße 4
D-91074 Herzogenaurach
Tel.: +49-9132-744-0
Fax: +49-9132-744-144
Internet: www.vipa.de
E-Mail: Info@vipa.de



Manual

VIPA PC-CPU 486

Ord.-No.: VIPA HB111E

Rev. 02/32

Subject to change to cater for technical progress

The information contained in this manual is supplied without warranties. Information is subject to change without notice.

© Copyright 2002 VIPA, Gesellschaft für Visualisierung and Prozessautomatisierung mbH
Ohmstraße 4, D-91074 Herzogenaurach,
Tel.: +49 (91 32) 744 -0
Fax.: +49 (91 32) 744-144
EMail: info@vipa.de
http://www.vipa.de

Hotline: +49 (91 32) 744-114

Disclaimer of liability

The contents of this manual was carefully examined to ensure that it conforms with the described hardware and software.
However, discrepancies can not be avoided. The specifications in this manual are examined regularly and corrections will be included in subsequent editions.
We gratefully accept suggestions for improvement.

Trademarks

VIPA®	is a registered trademark of VIPA Gesellschaft für Visualisierung and Prozessautomatisierung mbH
STEP®5	is a registered trademark of Siemens AG.
STEP®7	is a registered trademark of Siemens AG.

Any other trademarks referred to in the text are the trademarks of the respective owner and we acknowledge their registration.

About this Manual

This manual describes the handling of the PC-CPU 486 and PC-CPU 486DPM with Profibus-DP master from VIPA for the S7-400 from Siemens.

The modules contain a STPC, a plug-in location for COM modules from Hilscher and optional a DP master from VIPA.

The module is pluggable to all S7-400 module rails from Siemens that have plug-in locations with combined P- and K-Bus connection.

Outline

Chapter 1: principles Check the cruise

This introduction includes recommendations on the handling of electrostatically sensitive modules. A short overview shows you the module and the according deployment.

The chapter closes with the block diagram.

Chapter 2: Hardware description

This chapter describes the different versions of the PC-CPU and their special features.

In addition to the description of the components we will describe the deployment of the storage mediums. The chapter closes with the technical data.

Chapter 3: BIOS / Register

This chapter contains the general structure of the system. The chapter starts with the description of the BIOS setup, followed by the address assignment.

The chapter closes with a description of the registers.

Chapter 4: Deployment of thePC-CPU

This chapter describes the deployment of the PC-CPU. Here you will find information about communication possibilities with the periphery, the CPU and via Profibus.

The chapter closes with the guidelines for Profibus networks.

Content

User considerations

Objective and contents

This manual describes the PC-CPU 486 for the S7-400 from Siemens. It contains the construction, project implementation and technical data.

Target audience

The manual is targeted at users who have a background in automation technology and PLC-programming.

Structure of the manual

This manual consists of 8 chapters. Every chapter provides the description of one specific topic.

Guide to the document

This manual provides the following guides:

- 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.

ℹ

Note!
Supplementary information and useful tips.

Safety information

Application specifications

The PC-CPU 486 is constructed and manufactured for:

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

Danger!
This device is not certified for applications in

- 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 Principles

Outline

This introducing chapter contains hints about the usage of electrostatically sensitive modules and an overview over their deployment.

The following text describes:

- Safety Information for the user
- Block diagram
- Overview PC-CPU 486

Content

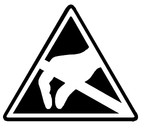
Order Data

Type	Order number	Description
PC-CPU486	VIPA 486-1BA00	32MB RAM, 8MB Flash
PC-CPU486DPM	VIPA 486-1BM00	32MB RAM, 8MB Flash, DP Master

Safety Information for the User

Handling of electrostatically sensitive modules

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



Electrostatically sensitive equipment!

The symbol at the left 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 electrostatically sensitive equipment.

Causes

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

Behavior of damaged modules

Modules that have been damaged by electrostatic discharge are usually not detected immediately. The respective failure may become apparent after a period of operation.

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

Precautions

Only the consistent implementation of protective devices and meticulous attention to the applicable rules and regulations for handling the respective equipment is able to prevent failures of electrostatically sensitive modules.

Shipping of modules

Please ship the modules exclusively in the original packing material. Additionally you may cover the shipping-ready modules with a conductive package. Conductive packages are anti-static foils or metallized plastic boxes.

Measurements and alterations on electrostatically sensitive modules

When you are executing measurements on electrostatically sensitive modules you should take the following precautions:

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

You should only use soldering irons with grounded tips when making modifications on electrostatically sensitive modules.

Attention!

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

Overview PC-CPU 486

General

The PC-CPU 486 is a complete compact PC-AT with the power of a 486-DX processor. On the internal 8MByte Flash-ROM the operating system MS-DOS 6.22 has been preinstalled.

Beside the plug for keyboard, monitor resp. TFT display, the PC-CPU 486 includes an optional Profibus master. The project engineering of the DP master takes place via WinNCS from VIPA, the access via C-functions.

Further on, the PC has an internal plug-in location for Hilscher COM modules.

For the external storage of data, the PC has an CompactFlash®-slot Type II at the front side. Here you may plug in CompactFlash® cards resp. hard discs like e.g. the IBM Microdrive with a storage capacity up to 1GByte.

Device variants

At this time the PC-CPU 486 is available in the following 2 variants:

- PC-CPU 486 (VIPA 486-1BA00)

  with 32MB RAM, 8MB Flash-ROM (DiskOnChip®),
  Plug-in locations for: Type II CompactFlash® and Hilscher COM module
  Connections: Mouse, keyboard, DVI jack, interface for communication and diagnosis at deployment of a Hilscher COM module
- PC-CPU 486 DPM (VIPA 486-1BM00)

  Like PC-CPU 486 additionally with integrated Profibus-DP master.

If nothing else is mentioned, the information in this manual refers to the basic version PC-CPU 486.

Deployment under Profibus

There are the following possibilities for deploying the PC-CPU 486 under Profibus:

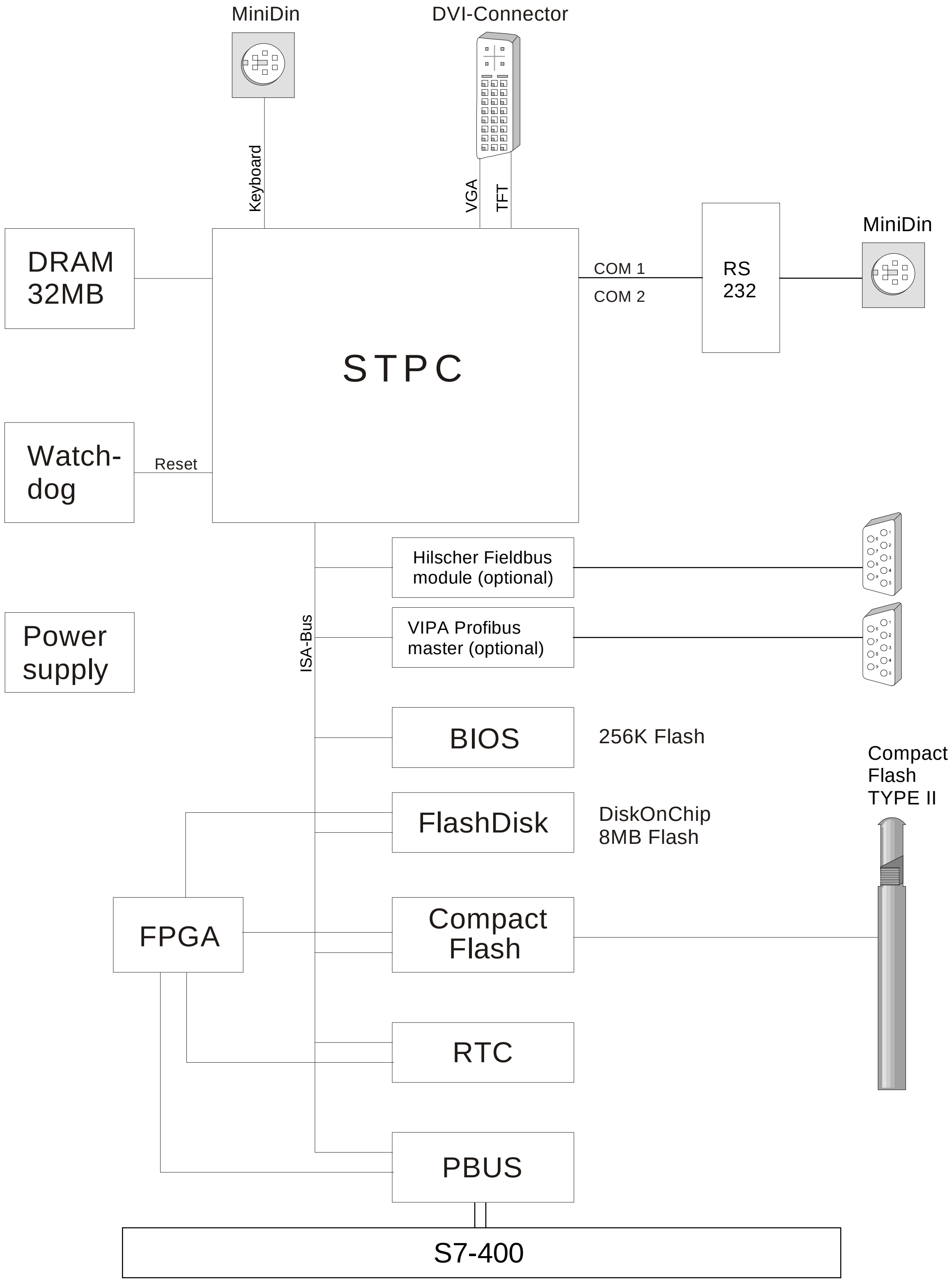
- At deployment of the PC-CPU 486DPM, the communication takes place via a Dualport-RAM. The project engineering is by means of WinNCS from VIPA and the access at PC side via C-functions.
  The DP master allows you to connect up to 125 Profibus-DP slaves to the PC. The master communicates with the slaves and provides the data for the PC via the Dualport-RAM. During this operation there may occur a maximum of 1024Byte data for in- and output.
- You deploy the PC-CPU 486 with a Profibus plug-in card at the COM module plug-in location. Then access and the project engineering takes place via the software added to the card.

Environmental conditions

- Operating temperature: 0... +55°C
- Storage temperature: -20... +65°C
- Relative humidity: 95% short time, 55% average, without condensation
- fan-less operation

Block diagram

The following block diagram shows the principle construction of the PC.



Chapter 2 Hardware description

Outline

The PC is available in different variants described in this chapter.

Here you will also find information about the usage of the storage mediums DiskOnChip® and CompactFlash®.

The chapter closes with the technical data.

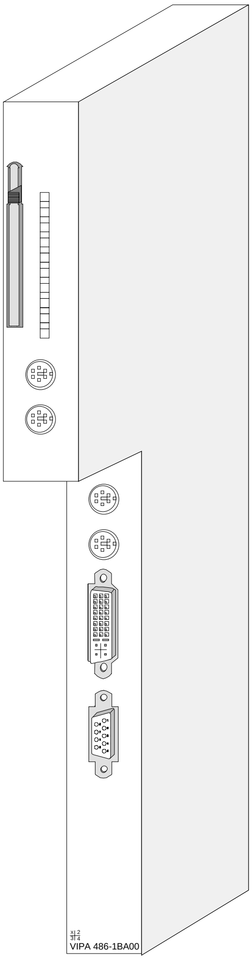
The following text describes:

- Structure
- Components of the PC
- Deployment of storage mediums
- Technical data

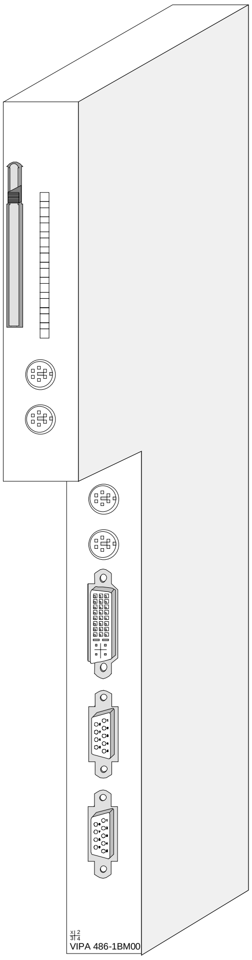
Content

Topic **Page**

System overview



VIPA 486-1BA00



VIPA 486-1BM00

Order data

Type	Order number	Description
PC-CPU 486	VIPA 486-1BA00	32MB RAM, 8MB Flash
PC-CPU 486 DPM	VIPA 486-1BM00	32MB RAM, 8MB Flash, DP Master
CompactFlash®	VIPA 950-1KS00	CompactFlash® Typ II, 32 MByte

General

Application area

The PC-CPU 486 is meant for the deployment in the S7-400 from Siemens and is included to your system via the STEP®7 manager from Siemens. For periphery and CPU communication an ANSI-C library is in the consignment.

DiskOnChip® for configuration data

For storing the operating system the PC provides an bootable DiskOnChip® disk drive (DOC). The Flash-based disk drive has a size of 8MB. The deployment of the disk drive and the booting sequence is to fix at the BIOS.

CompactFlash®

For the external storage of projects with greater data amount there is a plug-in location at the frontside for CompactFlash® storage modules Type II. Only the PC has access rights to the storage modules.

CompactFlash® storage modules are available at VIPA.

Plug-in location for Hilscher COM modules

For the flexible connection to fieldbus systems the PC provides an internal plug-in location for COM modules from the Hilscher company.

At the PC you will also find fieldbus, diagnosis jacks and LEDs, that are automatically controlled by the COM module.

DP master in the PC-CPU 486DPM

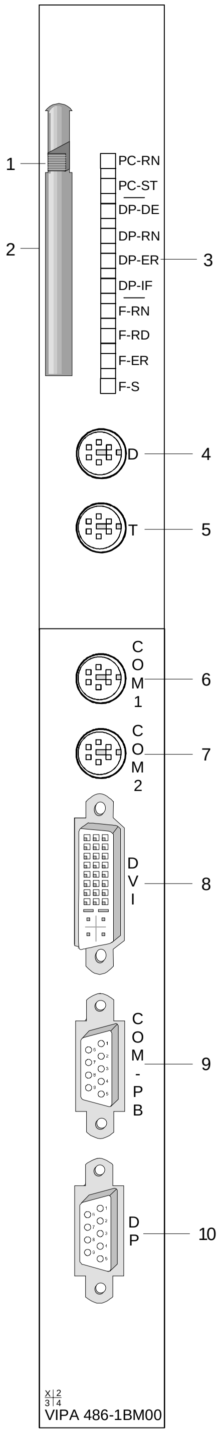
At deployment of a PC-CPU 486DPM is a DP master integrated. The communication between PC and DP master happens via Dualport-RAM. You configure the DP master via WinNCS from VIPA and access it from the PC via C-functions.

Properties

- PC-AT compatible STPC ATLAS 133MHz with 32MB main memory
- 8MB DiskOnChip®, CompactFlash®plug-in location, both bootable
- 2 RS232 interfaces and MiniDin jack for keyboard
- DVI jack (Digital Visual Interface) for TFT-LCD via PANEL LINK®
- integrated Watchdog timer
- Project engineering of the optional DP master via WinNCS from VIPA
- Standard page frame interface (4 page frames)
- CPU function, AIO, DIO via ANSI-C library

Structure

Structure PC-CPU 486DPM



[1] Exit key for CompactFlash®

[2] Plug-in location for CompactFlash®

[3] LEDs status monitor

[4] RS232 diagnosis interface for COM module

[5] Keyboard

[6] COM 1 (RS232)

[7] COM 2 (RS232)

[8] DVI interface

[9] Jack for the optional COM module

[10] RS485 jack for Profibus DP master (only VIPA 486-1BM00)

Components

PC-CPU 486

LEDs

At the frontside of the PC there are different LEDs serving the program status monitoring. The usage and the according colors of these diagnostic LEDs are to find in the following table.

Pos.	Label	Color	Description
PC			
1	PC-RN	green	PC is in RUN
2	PC-ST	red	PC is in STOP
Profibus DP master			
3	DP-DE	yellow	DE (Data exchange) shows communication via Profibus.
4	DP-RN	green	If only DP-RN is on, the DP master is in RUN. The slaves are called and the outputs are 0 ("clear" state). With DP-RN+DP-DE on, the master is in "operate" state. It changes data with the slaves.
5	DP-ER	red	Flashes at slave failure (Error)
6	DP-IF	red	Profibus DP master is signalizing an initialization error at wrong parameterization.
COM module			
7	F-RN	green	COM module is in RUN
8	F-RD	green	COM module shows READY
9	F-ER	red	COM module shows error
10	F-S	yellow	COM module shows status

Plug-in location for CompactFlash®

At this plug-in location you may insert a CompactFlash® memory card Type I+II. The PC includes the card as hard disk in the system. The deployment of a CompactFlash® card has to be set in the BIOS.

Together with a CompactFlash® adapter the memory card gets compatible to the "big" PCMCIA Type II format. Thus you may exchange data with all PCs via the PCMCIA slot.

Note!

The memory card may only be plugged in resp. out when the Pchas been shut down!

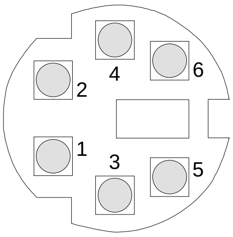
Jacks and plugs

Diagnosis

Diagnostic interface (RS232) for COM module.

This jack is used for diagnostic purposes at deployment of a COM module. More information about this diagnostic interface is to find in the documentation of Hilscher.

MiniDin jack

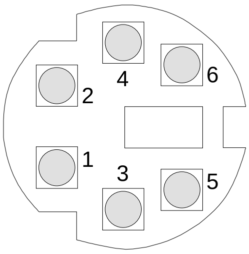


Pin	Assignment
1	DTR
2	RXD
3	GND
4	CTS
5	RTS
6	TXD

Keyboard

Please connect your keyboard here!

MiniDin jack

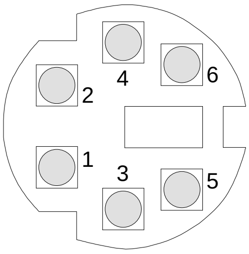


Pin	Assignment
1	+ KB-Data (I/O)
2	reserved
3	GND
4	+5V
5	+ KB-Clock (I/O)
6	reserved

Serial interface COM 1 / COM 2

The slot of the serial interface (RS232) is called as COM 1 / COM 2 and is defined for data transfers over maximum 15m at up to 38,4kBaud. The communication takes place via data, message and control lines.

MiniDin jack

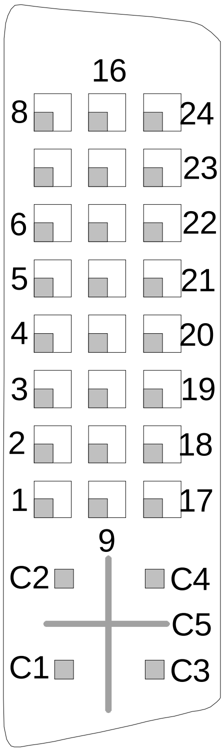


Pin	Assignment
1	n.c.
2	RXD
3	GND
4	CTS
5	RTS
6	TXD

DVI interface

The DVI jack (Digital Visual Interface) provides the connection of analoge and digital displays or monitors with a max. resolution of 1280 x 1024 Pixel.

The interface has the following pin assignment:



Pin	Signal
C1	Analog Red
C2	Analog Green
C3	Analog Blue
C4	Analog Horizontal Sync
C5	Analog RGB Return
1	T.M.D.S Data2-
2	T.M.D.S Data2+
3	T.M.D.S Data2/4 Shield
4	T.M.D.S Data4-
5	T.M.D.S Data4-
6	DDC Clock
7	DDC Data
8	Analog Vertical Sync
9	T.M.D.S Data1-
10	T.M.D.S Data1+
11	T.M.D.S Data1/3 Shield
12	T.M.D.S Data3-
13	T.M.D.S Data3+
14	+5V Power
15	Ground (return for +5V, HSync and VSync)
16	Hot Plug Detect
17	T.M.D.S Data0-
18	T.M.D.S Data0+
19	T.M.D.S Data0/5 Shield
20	T.M.D.S Data5-
21	T.M.D.S Data5+
22	T.M.D.S Clock Shield
23	T.M.D.S Clock+
24	T.M.D.S Clock-

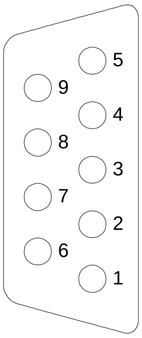
COM-PB

This slot receives the signals of the optional COM module.

The pin assignment of the interface is shown in the following picture:

More detailed information for the deployment of the COM module is to find in the documentation from Hilscher.

9pin jack



Pin	Assignment
1	Screen
2	n.c.
3	L2P
4	CNTR-P
5	GND
6	5V
7	n.c.
8	L2N
9	n.c.

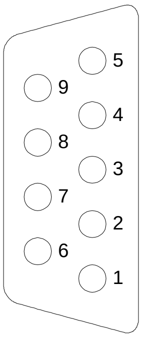
The module plugged in here is a Profibus-FMS master module from Hilscher.

RS485 interface

Via this interface you include the DP master of the PC-CPU 486DPM into your Profibus system.

The pin assignment of the interface is shown in the following picture:

9pin jack:



Pin	Assignment
1	GND
2	MODE-PIN
3	RxD/TxD-P
4	CNTR-P
5	GND
6	5V (max. 70mA)
7	n.c.
8	RxD/TxD-N
9	n.c.

Deployment of the storage mediums

Outline

The PC has an included hard disk on basis of Flash-ROM with a size of 8MB DiskOnChip® and a CompactFlash®-Slot Type II.

Via a CompactFlash® adapter the CompactFlash® card gets compatible to the "big" PCMCIA Type II format. This enables the data exchange with PCs with PCMCIA slot.

The assignment of the hard disks takes place via the BIOS-SETUP program.

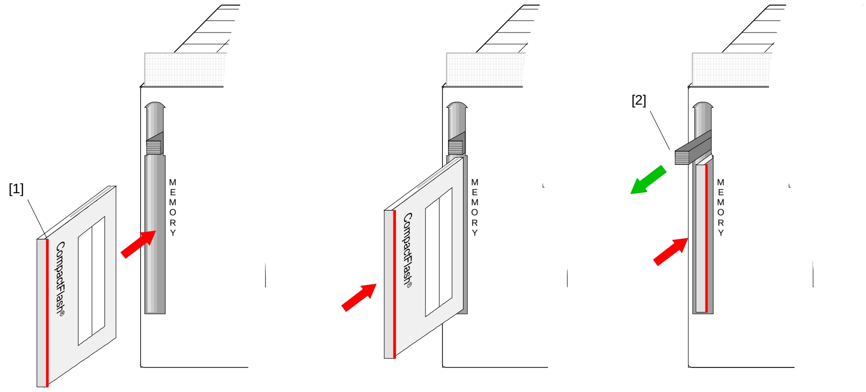
Here you have different possibilities to control the boot behavior of the PC.

CompactFlash® insert/eject

Every CompactFlash® storage module has an eject edge [1]. Turn the module so that the eject edge shows to the right.

Push the storage module into the PC without too much pressure until it clicks and the eject lever [2] appears.

To remove the CompactFlash® card push the eject lever.

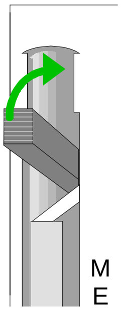


Note!

The CompactFlash® storage module may only be inserted or removed when shut down!

You have to set the storage module in the BIOS setup!

Unwanted eject protection



For protecting from unwanted ejects of the CompactFlash® card you may fold the eject lever into the upper side of the case.

For a wanted eject you either fold the eject lever back or you operate it with a screwdriver.

Technical data

PC-CPU 486

General	VIPA 486-1BA00
CPU Memory Storage medium Watchdog Profibus functionality	STPC ATLAS 133MHz 32MB work memory 8MB DiskOnChip® CompactFlash® slot Type II integrated and to trigger via registers via optional COM module from Hilscher
Electrical data	
Voltage supply Current consumption Dissipation power Status monitors Connections / interfaces Buffer clock and CMOS	DC 5V via backplane bus max. 1.3A typ. 6.5W via LEDs at the front side for the single components MiniDin Diagnosis for COM module (RS232) MiniDin Keyboard MiniDin COM 1 (RS232) MiniDin COM 2 (RS232) 9pin Fieldbus connection for COM module DVI Interface for monitor/TFT CompactFlash® slot, Type II Lithium-Akku, 30 days buffer
Dimensions and Weight	
Dimensions (WxHxD) in mm Weight	25x290x210 ca. 750g

Environmental conditions

- Operating temperature: 0 ... +55°C
- Storage temperature: -20 ... +65°C
- Relative humidity: short-timed 95% (average 55%), without condensation
- fan-less operation

PC-CPU 486DPM

General	VIPA 486-1BM00
CPU	STPC ATLAS 133MHz
Memory	32MB work memory
Storage medium	8MB DiskOnChip®
	CompactFlash® slot Type II
Watchdog	integrated and to trigger via registers
Profibus functionality	DP master integrates more fieldbus functions via optional COM module from Hilscher
Electrical data	
Voltage supply	DC 5V via backplane bus
Current consumption	max. 1.3A
Dissipation power	typ. 6.5W
Status monitors	via LEDs at the front side for the single components
Connections / interfaces	MiniDin Diagnosis for COM module (RS232)
	MiniDin Keyboard
	MiniDin COM 1 (RS232)
	MiniDin COM 2 (RS232)
	9pin Fieldbus connection for COM module
	9pin Profibus connection for DP master
	DVI Interface for monitor/TFT
	CompactFlash® slot, Type II
Buffer clock and CMOS	Lithium-Akku, 30 days buffer
Dimensions and Weight	
Dimensions (WxHxD) in mm	25x290x210
Weight	ca. 750g

Environmental conditions

- Operating temperature: 0 ... +55°C
- Storage temperature: -20 ... +65°C
- Relative humidity: short-timed 95% (average 55%), without condensation
- fan-less operation

Chapter 3 BIOS / Register

Outline

Content of this chapter is the description of the BIOS setup and the registers. Via the BIOS setup you may configure the hardware of your PC and adjust it to your conditions.

By accessing the registers you may directly influence your PC.

The following text describes:

- BIOS setup
- Register assignment

Content

Topic **Page**

BIOS setup handling

General

At the start-up of the system, the BIOS version is displayed on the connected monitor. Then the BIOS executes a test of the system components and the memory. At the end of the test the system attempts to boot. Between the start-up and the boot procedure you may access the BIOS setup routine by operating the **[Entf]/[Del]** key.

An appropriate message is displayed on the screen.

Via the setup menu you may configure the hardware of your PC.

BIOS menu

After hitting the [Entf]/[Del] key, the following menu appears:

```
+-------------------------------------------------------------------------------+
|                       System Bios Setup - Utility v4.2                        |
|             (C) 1999 General Software, Inc. All rights reserved               |
+-------------------------------------------------------------------------------+
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                          Basic CMOS Configuration                             |
|                             VIPA Configuration                                |
|                         Read Setup values from CMOS                           |
|                     Reset CMOS/Flash to factory defaults                      |
|                         Write to CMOS/Flash and Exit                          |
|                       Exit without changing CMOS/Flash                        |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
|                          <Esc> to continue (no save)                          |
|                                                                               |
+-------------------------------------------------------------------------------+
```

With the help of the standard cursor keys you may select the single menu options. You activate the selected sub-menu with [Enter].

Via [ESC] you leave the setup without saving your entries.

Control keys in BIOS

Every dialog box that is accessible via the main menu is controlled by means of the following keys:

[ESC] key

With the [ESC] key the dialog window is closed and you return to the main menu. The altered parameters are stored but not written to the CMOS.

Cursor keys

With the Cursor keys you choose the parameter you want to modify.

[PU]/[PD]

With the keys [PgUp] and [PgDn] or [Bild↑] and [Bild↓] or at the numeric block [+] and [-] you may alter the value of a parameter.

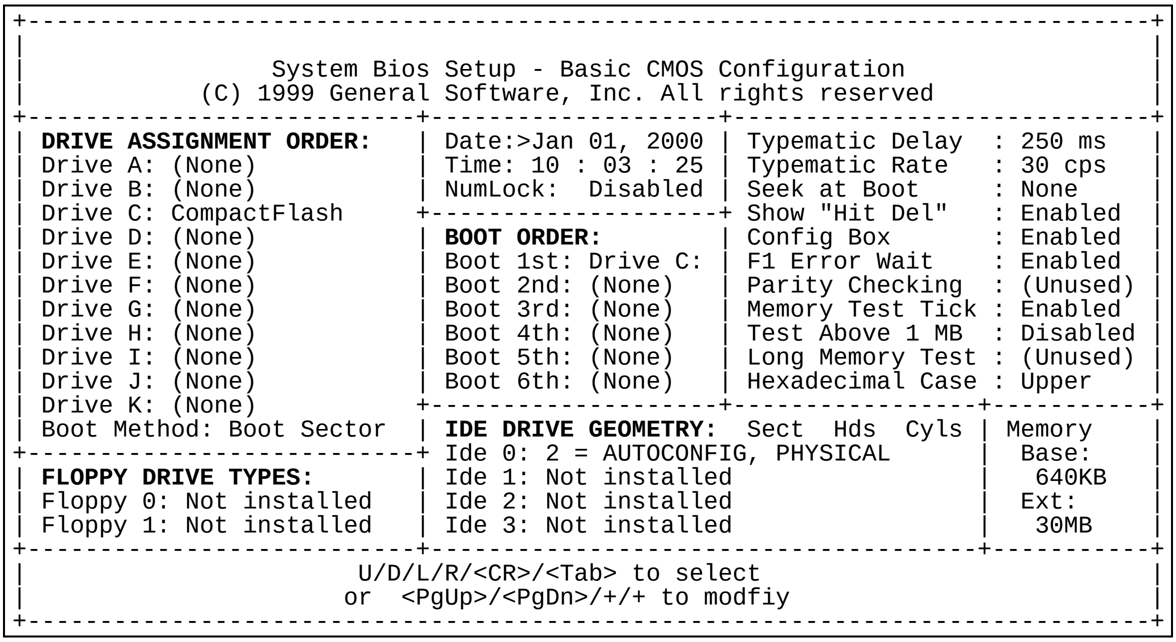
Note!

Please regard, that at the setup level there has not been loaded a driver for the German keyboard yet. Modified setup values are only valid and written into CMOS by confirming your changes explicitly with "Y". To obtain Y, you have to push the [Z] key.

Basic CMOS Configuration

This sub-menu provides access to the main settings for your system.

The menu is divided into a number of logical units. You may navigate through these by means of the cursor keys.



```
+------------------------------------------------------------------------------+
|                                                                              |
|                System Bios Setup - Basic CMOS Configuration                  |
|           (C) 1999 General Software, Inc. All rights reserved                |
+---------------------------+--------------------+-----------------------------+
| DRIVE ASSIGNMENT ORDER:   | Date:>Jan 01, 2000 | Typematic Delay  : 250 ms   |
| Drive A: (None)           | Time: 10 : 03 : 25 | Typematic Rate   : 30 cps   |
| Drive B: (None)           | NumLock:  Disabled | Seek at Boot     : None     |
| Drive C: CompactFlash     +--------------------+ Show "Hit Del"   : Enabled  |
| Drive D: (None)           | BOOT ORDER:        | Config Box       : Enabled  |
| Drive E: (None)           | Boot 1st: Drive C: | F1 Error Wait    : Enabled  |
| Drive F: (None)           | Boot 2nd: (None)   | Parity Checking  : (Unused) |
| Drive G: (None)           | Boot 3rd: (None)   | Memory Test Tick : Enabled  |
| Drive H: (None)           | Boot 4th: (None)   | Test Above 1 MB  : Disabled |
| Drive I: (None)           | Boot 5th: (None)   | Long Memory Test : (Unused) |
| Drive J: (None)           | Boot 6th: (None)   | Hexadecimal Case : Upper    |
| Drive K: (None)           +--------------------+-----------------+-----------+
| Boot Method: Boot Sector  | IDE DRIVE GEOMETRY:  Sect  Hds  Cyls | Memory    |
+---------------------------+ Ide 0: 2 = AUTOCONFIG, PHYSICAL      |  Base:    |
| FLOPPY DRIVE TYPES:       | Ide 1: Not installed                 |   640KB   |
| Floppy 0: Not installed   | Ide 2: Not installed                 |  Ext:     |
| Floppy 1: Not installed   | Ide 3: Not installed                 |   30MB    |
+---------------------------+--------------------------------------+-----------+
|                        U/D/L/R/<CR>/<Tab> to select                          |
|                       or  <PgUp>/<PgDn>/+/+ to modfiy                        |
+------------------------------------------------------------------------------+
```

Drive Assignment Order and IDE Drive Geometry

This section is used to assign logical drive names to the physical drives.

The VIPA BIOS only provides support for drive "C".

The following settings are valid for "C":

- "None": if you did **not** insert a CompactFlash® card
- "CompactFlash": if you have installed a CompactFlash® card.

Note!

All the other drives must be assigned to type "None". Any other setting can cause malfunction of the PC.

Please also note that the settings for "Ide 0" in the section *IDE Drive Geometry* must be changed at the same time as the parameters for drive "C". The following settings are valid:

CompactFlash®	DRIVE ASSIGNMENT ORDER \ Drive C:	IDE DRIVE GEOMETRY \ Ide0:
Installed	CompactFlash®	2 = AUTOCONFIG, PHYSICAL
Not installed	None	Not installed

The integrated DiskOnChip® drive (DOC) is configured by means of the "VIPA Configuration" menu.

Floppy drive types

Here you would normally define the settings for the floppy disk drive. Since the PC does not support floppy disk drives you have to select the type "Not installed"! Otherwise you will encounter long-term delays between the system test and the start-up of the system.

Date, Time, NumLock

Here you enter the current date and time.

The "NumLock" parameter defines the status of the [NumLock] key after the system has booted.

Note!

If the real-time clock has stopped it has to be assumed that the backup battery for the CMOS memory is discharged or defective.

Please contact the VIPA Hotline if this battery has still not accepted a charge after one day.

Your CMOS settings are safe even if the battery is empty. You only need to adjust the clock and the date.

Boot order

This parameter is preset to "Drive C" and it defines the boot sequence.

IDE Drive Geometry

This section defines the geometry settings of the IDE drives.

If you are using a CompactFlash® card you have to enter "autoconfig, physical" for this parameter. If you did not install a CompactFlash® card the parameter must be set to "None".

Typematic Rate/Delay

This parameter defines the keyboard interface and specifies the repetition rate for the characters. You should not change this setting.

Seek at Boot

This option specifies which drives should receive a "SEEK" command before booting the system. The default is "NONE" to ensure that the boot procedure is as quick as possible.

Show "Hit Del"

If this option is active the system will display a message to press [Del] to access the setup menu while the system is booting.

Config Box

This parameter specifies whether the configuration settings should be displayed on screen when the system boots.

F1 Error Wait

If you activate this menu item, the boot process is stopped when an error is detected. You then may decide what action to take. If you press [F1] the system will continue booting. Press [Del] to gain access to the setup menu.

Parity Checking

This menu item is not used.

Memory Test Tick

Turn this option on to sound a test click via the PCs speaker when memory is being tested.

Test Above 1MB

This parameter determines whether the memory above the 1MB limit will be tested by the memory test or not.

Long Memory Test

This menu item is not used.

Hexadecimal Case

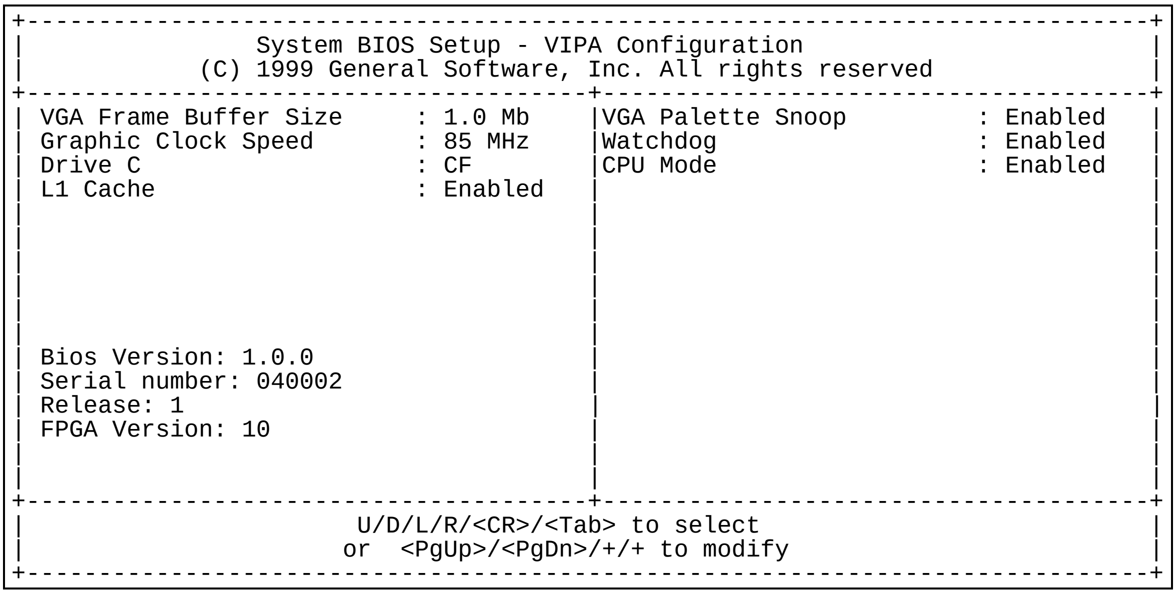
This menu item determines the display format that the BIOS uses for hexadecimal numbers. You can either choose "UPPER" (capitals) or "LOWER" (lower case).

Memory Base / Ext.

This menu item displays the memory configuration below 1MB (Base Memory) and above 1MB (Extended memory). These parameters are purely for information purposes and they cannot be changed.

VIPA Configuration

This sub-menu is used to define the board-specific settings. You may navigate through the menu by means of the cursor keys.



```
+-----------------------------------------------------------------------------+
|                   System BIOS Setup - VIPA Configuration                    |
|               (C) 1999 General Software, Inc. All rights reserved           |
+--------------------------------------+--------------------------------------+
| VGA Frame Buffer Size     : 1.0 Mb   |VGA Palette Snoop         : Enabled   |
| Graphic Clock Speed       : 85 MHz   |Watchdog                  : Enabled   |
| Drive C                   : CF       |CPU Mode                  : Enabled   |
| L1 Cache                  : Enabled  |                                      |
|                                      |                                      |
|                                      |                                      |
|                                      |                                      |
|                                      |                                      |
|                                      |                                      |
|                                      |                                      |
| Bios Version: 1.0.0                  |                                      |
| Serial number: 040002                |                                      |
| Release: 1                           |                                      |
| FPGA Version: 10                     |                                      |
|                                      |                                      |
|                                      |                                      |
+--------------------------------------+--------------------------------------+
|                        U/D/L/R/<CR>/<Tab> to select                         |
|                       or  <PgUp>/<PgDn>/+/+ to modify                       |
+-----------------------------------------------------------------------------+
```

VGA Frame Buffer Size

This parameter determines the amount of memory that is used as graphic memory for the internal graphic controller.

The respective memory area is no longer available as main memory for the processor.

Graphic Clock Speed

This option determines the clock speed used for the internal graphic controller. This option should always be set to "85MHz".

Drive C

This parameters determines the drive from which the operating system is booted:

"CF" CompactFlash® (C:) followed by the internal drive (D:)

"DOC" DiskOnChip® - internal drive (C:) followed by CompactFlash® (D:)

In every case the other drive will be accessible as drive D:.

L1 Cache

This option enables and disables the L1 cache of the processor.

VGA Palette Snoop

This menu item determines whether access to the VGA palette can only occur within the STPC (disabled) or whether it should also be routed to the external PCI bus (enabled).

Watchdog

This option is used to enable or disable a Watchdog that issues an automatic reset after a certain time has elapsed (30s).

If you deactivate the Watchdog in your application program you can ensure that your system has booted without errors. Otherwise the PC is re-booted when the watchdog has expired.

You may define the watchdog time and the watchdog properties in the system register.

CPU Mode

This parameter determines the physical properties of the P bus to the S7-400 bus from Siemens. You may select from:

enable: Operation as stand-alone CPU in the PLC
(allows the direct communication with the peripheral area without PLC-CPU)

disable: Operation as CP in the PLC
(allows page frame communication with a Siemens-CPU)

Version data

The left hand section of the "VIPA Configuration" menu displays hardware specific parameters:

BIOS-Version: The version level of the BIOS

Serial No.: Every PC is provided with a unique serial no. This serial no. is identical to the serial no. located on the enclosure.

Revision level: identical to the revision level located on the enclosure.

FPGA-Version: The version of the FPGA that controls the V-Bus access.

Note!

You should include this information when you request information from the service department of VIPA GmbH, to allow us to help you more effectively.

Note!

The following menu items display a query that is to be answered with "Yes" or "No".

Please note that the keyboard uses the US layout for the BIOS setup, i.e. that you have to press the [Z] key on German keyboards to obtain the letter "Y".

Read Setup values from CMOS

This option loads the most recent settings that were saved to CMOS memory.

Reset CMOS/Flash to factory defaults

This option loads the factory default values into CMOS memory.

If you wish to write these values into the CMOS memory you have to use the menu item "Write to CMOS and Exit" to quit from this function.

Write to CMOS/Flash and Exit

This menu item saves the modified settings to CMOS-RAM.

When the settings have been saved the system is re-started automatically which will reload the modified configuration settings.

Note!

This process can require a few seconds since the data is also saved in the EEPROM.

During this time you may **NOT** turn the system off or issue a manual reset!

Exit without changing CMOS/Flash

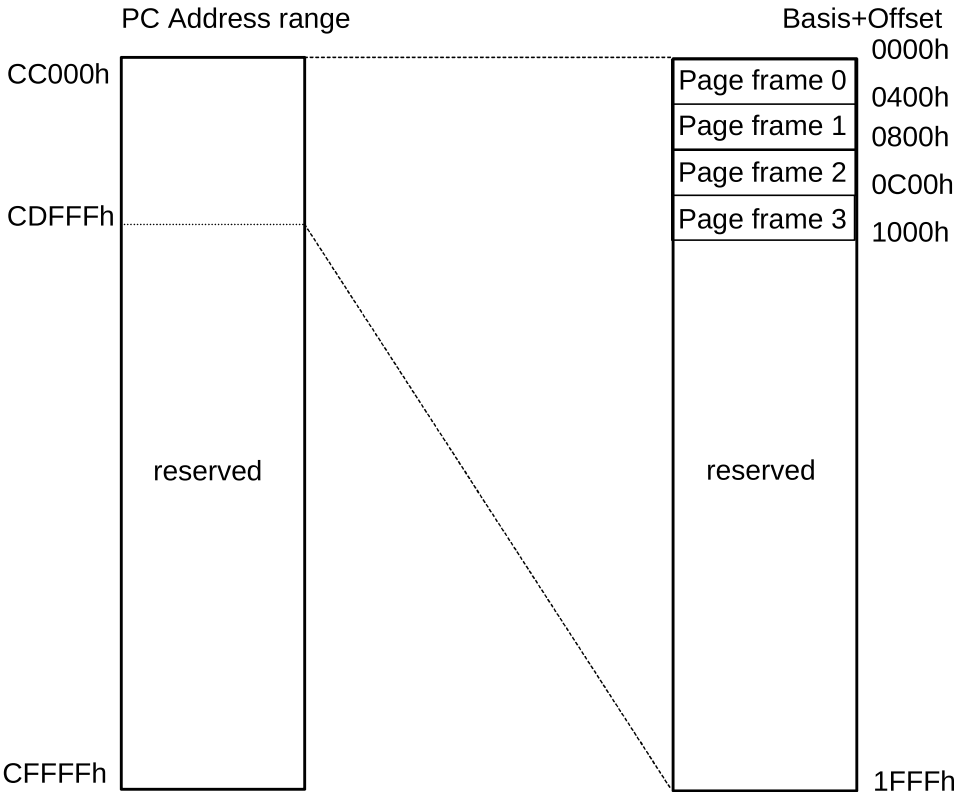
When you select this menu item, you quit from the setup menu without saving and activating any settings to CMOS memory or to the EEPROM.

Address assignment at the PC

As host controller a STPC is deployed. The address range of the RAM has a size of 8KByte and is stored from CC000h to CDFFFh in the memory area.

The interface administers the addresses SA0 to SA12.

The registers are stored in the I/O area from 298h to 29Fh.



Register description

Address range

The following addresses are occupied by VIPA:

CPU side

270h - 277h	**Watchdog**
280h - 28Fh	**Control register I**
280h - 281h	reserved
282h	LED Control
	Bit 0 = "0" RUN LED off, "1" RUN LED on
	Bit 1 = "0" STOP LED off, "1" STOP LED on
283h - 284h	reserved
285h	EEPROM Port
286h - 28Dh	reserved
28Eh	Version
28Fh	Device ID = 88h
290h - 297h	**Control register II**
290h	Watchdog timer value
291h	reserved
292h	C165 Control register
293h	Enable register for Watchdog timer value
294h - 295h	reserved
296h	CPU Control
297h	CPU Mode
298h - 29Fh	**Register for page frame communication**
298h	Operation mode (read/write)
299h	Status register (control)
29Ah	3FD - Interrupt register (IRQ Req/Ack)
29Bh	3FF - Interrupt register (IRQ Req/Ack)
29Ch	Page frame base register (read/write)
29Dh	Page frame select register (read only)
29Eh	Revision ID register (read only)
29Fh	Device ID register (read only)

Watchdog

(I/O address range **270h-277h**)

The Watchdog is turned off when the system has started or has been reset and it can be enabled under software control.

The Watchdog register is controlled via I/O address 270h and the following parameters:

Watchdog on	load 40h after address 270h
Watchdog off	load 50h after address 270h
Watchdog trigger	load 60h after address 270h and then load 70h after address 270h

The Watchdog must be triggered when the power has been turned on. The triggering time is programmable:

Trigger time

Parameter for triggering time I/O-register **290h** R/W (enabled via **293h**).

The trigger time is a multiple of 147ms and has to be entered in the register 290h Watchdog time. As soon as you load the register 290h with 00h, the Watchdog timer is deactivated.

Please regard, that you have to enable write access for the register Watchdog time.

The following sequences to the register 293h are activating resp. deactivating the write access to the registers 290h and 292h.

Enable write access:	03h, 06h, 03h, 00h
Disable write access:	03h, 06h, 03h, 01h

Note!

After reset the write access to **290h** and **292h** is inhibited.

Serial number

The serial number can be found in the registers **271h** and **272h**.

EEPROM

More detailed information about the I/O port **285h** is available at VIPA.

C165 Control register

The C165 control register I/O port **292h** R/W (release via **293h**) are reserved for download purposes.

CPU Control register 296h

Index:	**296h**
Reset Value:	-
Powerup Value:	BE, STP
Access:	Read/Write

Bit	Description
7	reserved
6	reserved
5	Read = 0, Write rst ZYK (STOP state)
4	Read = 0, Write rst BT (Block transfer end BE)
3	reserved
2	reserved
1	Read state of the ZYK signal at the bus, write 1 set ZYK (RUN state)
0	Read state of the signal for block transfer at bus, write 1 set BT (Block transfer start BA)

CPU Mode register 297h

Index:	**297h**
Reset Value:	-
Powerup Value:	00h
Access:	Read/Write

Bit	Description
7	Operation as CPU (0 = disable, 1 = enable)
6	reserved
5	reserved
4	reserved
3	reserved
2	reserved
1	reserved
0	reserved

This parameter determines the physical properties of the P bus to the S7-400 bus from Siemens. You may select from:

enable: Operation as stand-alone CPU in the PLC
(allows the direct communication with the peripheral area without PLC-CPU)

disable: Operation as CP in the PLC
(allows page frame communication with a Siemens-CPU)

Page frame interface to S7-400 from Siemens

Outline

For the communication to the CPU there are 4 page frames available at the CPU. The communication has to be realized at the PLC side with an according PLC application, at the PC side with C-functions. A more detailed description of the C-functions is to find in the header file and in the sources.

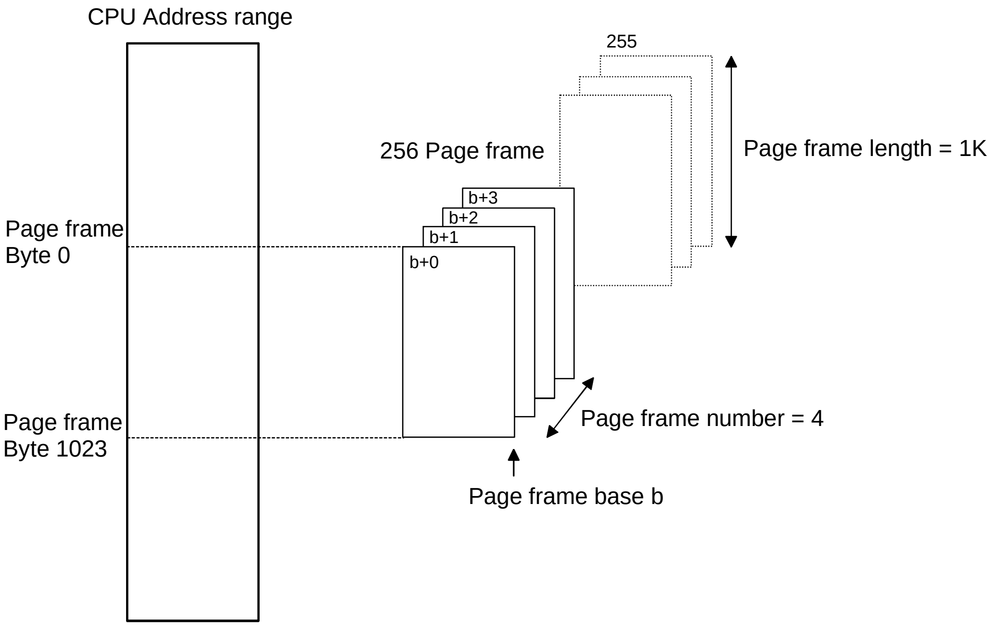
Note!

This kind of communication with a CPU is only possible when you're operating the PC as CP. The according adjustment has to take place in the BIOS in the "VIPA Configuration". Choose the *CPU Mode* "disable".

The project engineering in the hardware configurator from Siemens is as *S5 adapter* in the hardware catalog under simatic400/im-400/s5-adapter without additional parameters.

Structure

The CPU supports up to 255 page frames with a length of each 1kByte. The PC may use up to 4 page frames for the communication with the CPU. Starting with the basic page frame that you have to set, the following 3 ascending page frames are occupied by the PC:



Operation mode register 298h

Index:	**298h**
Reset Value:	-
Powerup Value:	00h
Access:	Read/Write

Bit	Description
7	reserved
6	reserved
5	reserved
4	reserved
3	reserved
2	reserved
1	3FF-IRQ, 0=disable, 1=enable
0	3FD-IRQ, 0=disable, 1=enable

The register operation mode (298h) is write-protected. This write protection is active after a RESET. With the following sequences at the register 29Fh you may enable resp. disable the write protection.

enable write protection:	03h, 06h, 03h, 00h
disable write protection:	03h, 06h, 03h, 01h

Status and interrupt register 299h

IRQ-Req/Ack	299h
Address:	**299h**
Reset Value:	00h
Powerup Value:	00h
Access:	Read/Write

Bit	Description
7	reserved
6	reserved
5	reserved
4	reserved
3	reserved
2	reserved
1	Read 1 = Write Cycle Address 3FFh pending, Write 1 clear all 3FF-IRQ Request
0	Read 1 = Write Cycle Address 3FFh pending, Write 1 clear all 3FF-IRQ Request

Interrupt register 29Ah, 29Bh

You may initiate an interrupt at the PC by means of a write access of the CPU to the offset addresses 3FDh and 3FFh.

The according entries containing more detailed information about the interrupt are stored in the registers:

At an interrupt by an access to 3FDh resp. 3FFh an entry in the register 29Ah resp. 29Bh is created. After the start-up and after a RESET the register contains 00h. This register allows read and write access and it has the following structure:

IRQ-Req/Ack, page frame 0-3, 29Ah		29Bh
Address:	**29Ah**	**29Bh**
Reset Value:	00h	00h
Powerup Value:	00h	00h
Access:	Read/Write	Read/Write

Bit	Description
7	0
6	0
5	0
4	0
3	Read 1=Write Cycle Addr. 3FDh on Page 3+Base pending, Write 1 clear IRQ Request
2	Read 1=Write Cycle Addr. 3FDh on Page 2+Base pending, Write 1 clear IRQ Request
1	Read 1=Write Cycle Addr. 3FDh on Page 1+Base pending, Write 1 clear IRQ Request
0	Read 1=Write Cycle Addr. 3FDh on Page 0+Base pending, Write 1 clear IRQ Request

Page frame base 29Ch

Address:	**29Ch**
Reset Value:	-
Powerup Value:	00h
Access:	Read/Write

Bit	Description
7:0	Page frame base address

page frame select register 29Dh

Address:	**29Dh**
Reset Value:	-
Powerup Value:	00h
Access:	Read only

Bit	Description
7:0	Page frame select register

Information register

Revision Identification Register 29Eh

Address: **29Eh**
Access: Read only

Bit	Description
7:0	Revision number

Device Identification Register 29Fh

Address: **29Fh**
Access: Read only

Bit	Description
7:0	Identification Number

Chapter 4 Deployment of the PC-CPU

Outline

This chapter describes the deployment of the PC in the S7-400 from Siemens. The description refers to the communication possibilities with periphery and CPU and those via Profibus.

The following text describes:

- Overview over the communication possibilities
- Communication with the periphery
- Communication with the CPU
- Profibus communication
- Deployment of the COM modules

Content

Topic **Page**

Overview

General

At deployment in a S7-400 system from Siemens you may insert the PC-CPU 486 into all module rails that have plug-in locations with combined P- and K-Bus plug. You may deploy the PC as stand-alone CPU or as CP together with a CPU.

Via C-functions and PLC program you have the following possibilities of communication:

- Periphery communication with I/O periphery as stand-alone CPU
- Page frame communication with a CPU as CP
- Profibus communication of the PC-CPU 486DPM
- Fieldbus communication via Hilscher COM module

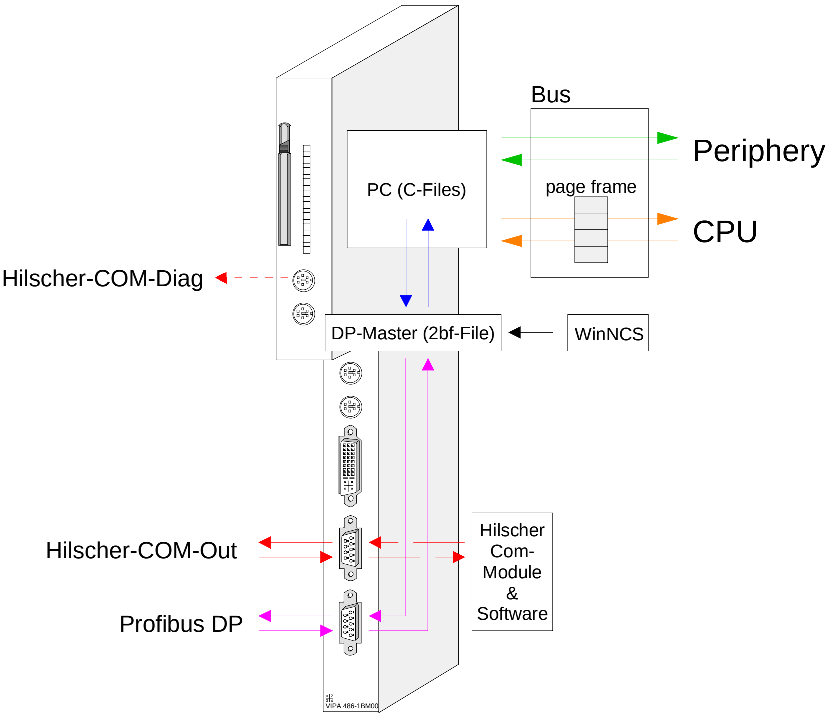
Deployment under Profibus

For the deployment of the PC-CPU 486 under Profibus you have the following possibilities:

- You deploy the variant with integrated Profibus master. Hereby the communication works via a Dualport-RAM. The project engineering of the Profibus master happens via WinNCS and the access at the PC side via C-functions.
- You deploy the PC-CPU 486 with a Profibus plug-in module at the COM module plug-in location. Hereby access and project engineering functions via the software added to the card.

Outline

Here once more all communication possibilities:



Communication via periphery

Outline

At the communication via the periphery there is a communication to the peripheral modules connected to the backplane bus. For the communication there is an extensive function library at your disposal, where more detailed information is to find in the header file and the sources.

Note!

This kind of direct communication with the peripheral area is only available if you deploy the PC as stand-alone CPU. The according adjustment has to be made in the BIOS at the "VIPA Configuration". Choose "enable" in the *CPU Mode*.

Functions of p400_api

The following functions are at this time available via the library p400_api:

```
void   p400_init                   (void); //Initializes the backplane bus, has to be called first!
BYTE   p400_scan                   (void); //Scans the backplane bus(detect module configuration) and prepares
                                          //internal structures for I/O access, has to be called second
BYTE   p400_di_read                (BYTE bSlot, BYTE far * pbBuf, WORD wLength); //Reading a digital module
BYTE   p400_do_write               (BYTE bSlot, BYTE far * pbBuf, WORD wLength); //Writing a digital module
BYTE   p400_ai_read                (BYTE bSlot, BYTE far * pbBuf, WORD wLength); //Reading an analog module
BYTE   p400_ao_write               (BYTE bSlot, BYTE far * pbBuf, WORD wLength); //Writing an analog module
BYTE   p400_write_dataset (BYTE bSlot, BYTE bDatasetNumber, BYTE far * pbDataset, BYTE bDatasetLength);
//Writing a record set (Basic function for parameterizing the modules)

BYTE   p400_parameterize_default (BYTE bSlot); //Sets a module to the default parameterization state
BYTE   p400_parameterize_S7H_6ES7_432_1HF00_0AB0_CT (BYTE bSlot); //Example for the parameterization of this analog
                                                                  //output
BYTE   p400_parameterize_S7H_6ES7_431_1KF00_0AB0_CT (BYTE bSlot); //Example for the parameterization of this analog
                                                                  //input
```

Example program p400test

The usage of the shown functions is illustrated in the example application p400test, that you may order at VIPA.

The application p400test provides the following test functions:

Key	Command	Description
[0]	Exit	Exit application
[1]	ScanSystem	Detects the module configuration by a bus scan (has to be called first)
[2]	ReadDigitalIn	Reading a digital module (dialog slot input)
[3]	WriteDigitalOut	Writing a digital module (dialog slot input and value)
[4]	Parameterize_default	Set module group to default parameterization state (dialog slot input)
[5]	Parameterize_S7H_6ES7_432_1HF00_0AB0_CT	Example parameterization of this analog output module
[6]	Parameterize_S7H_6ES7_431_1HF00_0AB0_CT	Example parameterization of this analog input module
[7]	ReadAnalogIn	Reading an analog module (dialog slot input)
[8]	WriteAnalogOut	Writing an analog module (dialog slot input and value)

Please regard that the example application executes the function p400_init automatically.

Communication via page frame

Outline

For the communication with a CPU the PC supports 4 page frames. The communication is realized at PLC side via a PLC program and at PC side via C-functions. A more detailed description of the C-functions is to find in the header file and the sources. You may request a demo application also to this topic.

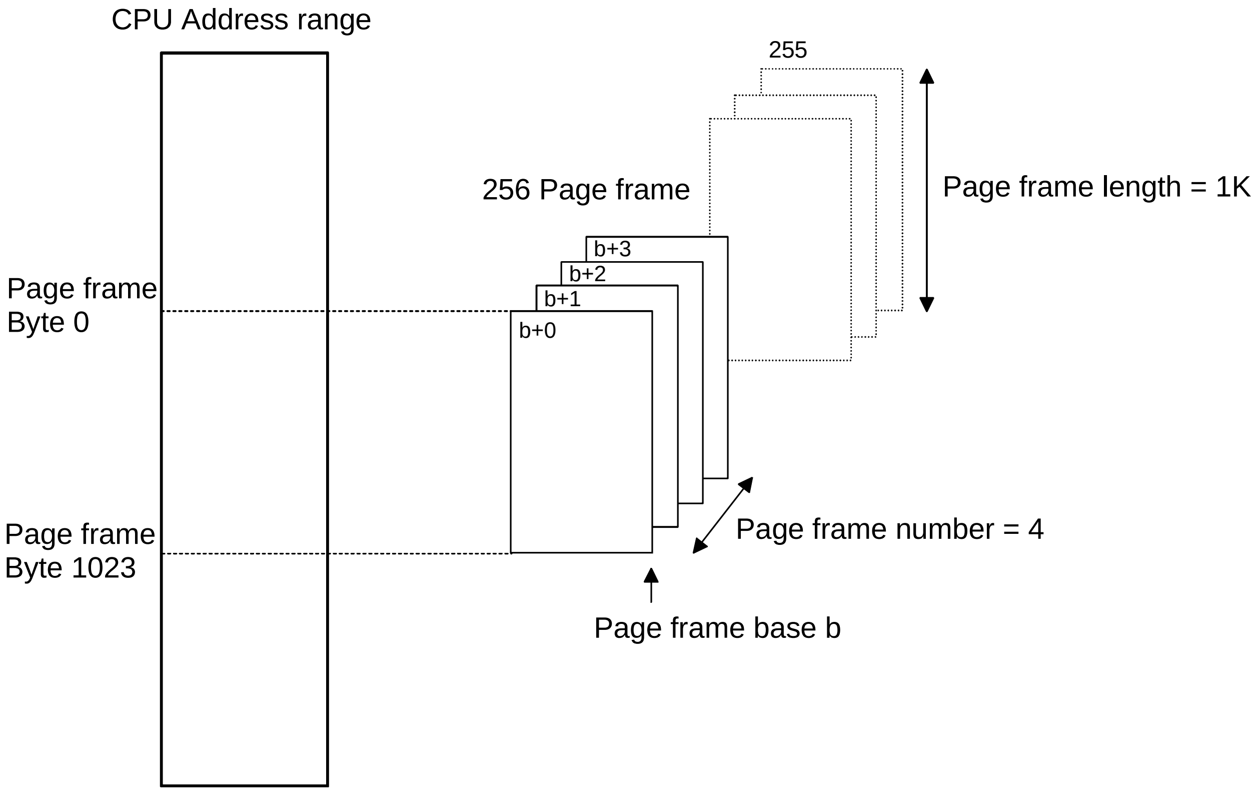
Note!

This kind of direct communication with a CPU is only possible if you deploy the PC as CP. The according adjustment has to be made in the BIOS at the "VIPA Configuration". Choose "disable" in the *CPU Mode*.

The project engineering in the hardware configurator from Siemens is as *S5 adapter* in the hardware catalogue under simatic400/im-400/s5-adapter without additional parameters.

Page frame interface

The CPU supports up to 255 page frames with a length of 1kByte each. The PC may use up to 4 page frames for the communication with the CPU. Starting with the base page frame that you have to set, the ascending 3 page frames are occupied by the PC:



Functions at the CPU

The CPU uses the following standard communication functions:

Read functions:

SFC 65514	Read Byte from page frame
SFC 65515	Read word from page frame
SFC 65516	Read double-word from page frame

Before calling the read functions you have to set the page frame number and the address of the Byte to read in AKKU1. The module expects the page frame number in AKKU1-H and the address in AKKU1-L.

The read value is returned in AKKU1.

Example: Reading a Byte from page frame 2 of Byte 30

```
L    2                 Page frame number
SLD  16                to AKKU1-H
L    30                Byte address
OD                     Create AKKU1
UC   SFC65514          Call SFC
T    MB 18             Save value
```

Write functions:

SFC 65511	Write Byte to page frame
SFC 65512	Write word to page frame
SFC 65513	Write double-word to page frame

Before calling the write functions you have to set the page frame number and the address of the Byte to write in AKKU2. The module expects the page frame number in AKKU2-H and the address in AKKU2-L.

The value to write is loaded from AKKU1.

Example: Writing a Byte to page frame 1 of Byte 50

```
L    1                 Page frame number
SLD  16                to AKKU1-H
L    50                Byte address
OD                     Create AKKU1
L    MB 20             Value to write in the MB20
UC   SFC65514          Call SFC
```

Functions at the PC

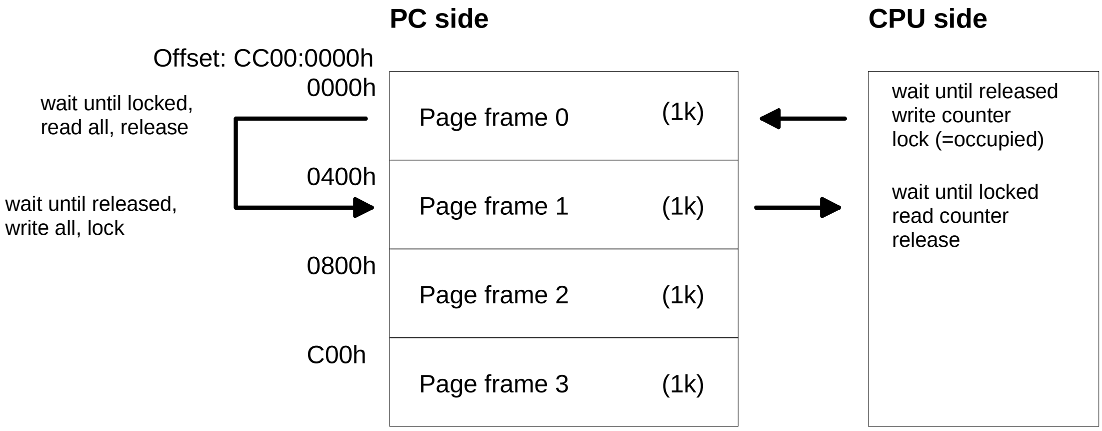
For using the page frame communication, you need a C-function user application at the PC. The user application receives the page frame data transferred via the CPU, acknowledges the transmission and transfers the data to the CPU. The programming approach is to see in the following example.

You may request a demo application from VIPA also to this topic.

Example for the page frame communication

This example outlines the communication between the PC and a CPU. For this the PC uses page frame 0 for reading and page frame 1 for writing data. A lock happens via the 1st Byte per page frame. The data read by the PC are immediately transferred to the CPU via page frame 1.

The procedure is illustrated in the following picture:



Write process CPU → PC via page frame 0

As soon as the PC released the page frame (Byte 0, Bit 0 = 0), the content of the 1st word in the user data area is ascended for 1 (MW0) and again written into the page frame.

Read process PC → CPU via page frame 1

As soon as the PC wrote data to the page frame (Byte 0, Bit 0 =1), the content of the 1st word in the user data area is copied to MW2 and the reading counter MW4 is ascended for 1.

Note!

You may request a demo application from VIPA .

Profibus communication of the PC-CPU 486DPM

General

Profibus is an open fieldbus standard for building, manufacturing and process automation. Profibus defines the technical and functional properties of a serial fieldbus system that can be used to create a network of distributed digital field-automation equipment on the lower (sensor/actuator level) to middle performance level (process level).

Master and slaves

Profibus distinguishes between active stations (masters) and passive stations (slaves).

Master equipment

Master equipment controls the data traffic on the bus. There may be also several masters at one Profibus. This is referred to as multi-master operation. The bus protocol establishes a logical token ring between the intelligent devices connected to the bus. Only the master that owns the token communicates with the slaves.

A master may send unsolicited messages if it has the bus access permission (Token).

The PC-CPU 486DPM has an integrated Profibus-DP master.

Slave equipment

Typical slave equipment holds data of peripheral equipment, sensors, drives, transducers.

These devices do not have bus access permission in accordance with the Profibus standard. They can only acknowledge messages or transfer messages to a master if requested by the respective master.

Data consistency

Data is referred to as being consistent, if it has the same logical contents. Data that belongs together is: the high- and low-byte of an analog value (word consistency) and the control and the status byte with the respective parameter word required to access the registers.

The data consistency during the interaction between the peripherals and the controller is guaranteed. That means that the data transferred via Profibus are read resp. written together.

Profibus guarantees the consistency of the required length.

Diagnosis

There is a wide range of diagnosis functions under Profibus-DP to allow a fast error localization. The diagnosis data are broadcasted by the bus system and summarized at the master.

Restrictions

- Max. 125 DP slaves at one DP master - max. 32 slaves/segment
- Max. cable length under RS485 between two stations 1200m (depending on baudrate)
- The maximum baudrate is 12MBaud
- The Profibus address may not be changed during operation

Profibus with the PC-CPU 486

For the deployment of the PC-CPU 486 under Profibus you have the following possibilities:

- You deploy the variant with the integrated Profibus master. Here the communication happens via a Dualport-RAM. The project engineering of the Profibus master takes place via VIPA WinNCS and the access from the PC via C-functions.
  The DP master allows to connect up to 125 Profibus-DP slaves to the PC. The master communicates with the slaves and provides the data via the Dualport-RAM for the PC. Here you may create a maximum of 1024Byte data for inputs and outputs.
- You deploy the PC-CPU 486 together with a Profibus plug-in module at the COM module plug-in location. Hereby access and project engineering functions via the software added to the card.

Transfer medium RS485

As transfer medium Profibus uses an isolated twisted-pair cable based upon the RS485 interface. The transfer rate is max. 12 MBaud.

The RS485 interface is working with voltage differences. Though it is less irritable from failures than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure. Your VIPA PC-CPU 486 includes a 9pin slot where you connect the Profibus master into the Profibus network.

The bus structure under RS485 allows an easy connection resp. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

Addressing

Every partner of the Profibus network has to identify itself with a certain address.

You have to assign the address to the VIPA Profibus master during the project engineering under WinNCS.

Project engineering DP master

General

With the help of the project engineering tool WinNCS from VIPA you may configure your master via the option "Profibus" as well as the according periphery of your slaves.

The module transfer functions of WinNCS allow you numerous possibilities for the data transfer to your master module.

The DP master reacts like the IM 308-C from Siemens and may also be configured under "Com Profibus" from Siemens as IM 308-C.

Include GSD in WinNCS

For the project engineering of DP slaves of the different manufacturers, the import of a GSD (Gerätestammdatei = electronic data sheet) allows you to transfer the functionality of the slave and its modules into WinNCS.

For including the GSD you close WinNCS and copy the GSD referring to your slave into the directory ...\VIPA Component Library\WinNCS\GSD\ into the folder *German* resp. *English*.

When the copy has been completed you may start WinNCS again. Now you may use the new options via the slave project window by choosing a *Family* and the *Station type*.

Project engineering under WinNCS

With the projecting tool WinNCS from VIPA you may configure your Profibus master as follows.

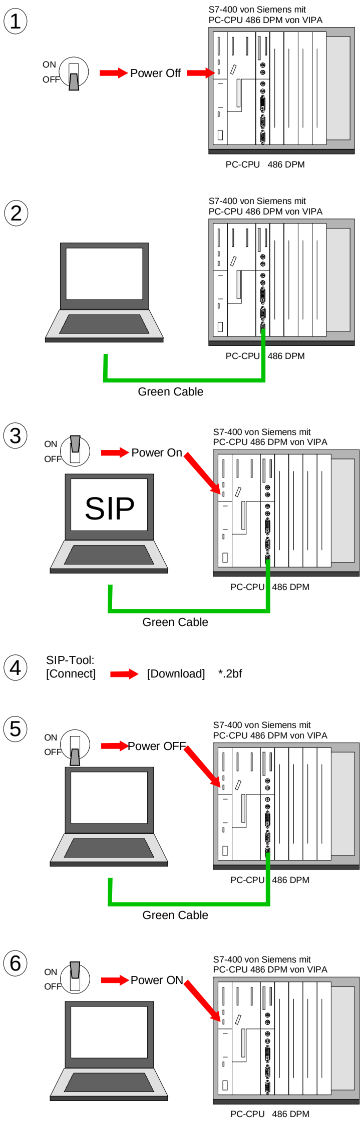
1.	For the project engineering of DP slaves of other manufacturers you include the according GSD in WinNCS.
2.	Start WinNCS and create a new project file via **File** > *create/open* under the option "Profibus".
3.	If not done yet, create a new **Profibus function group** in the network window via and click at [apply] in the parameter window.
4.	Use to create a **Profibus host/master** in the network window and add the Profibus address of your master in the parameter window.
5.	Create a **Profibus slave** via . Add the Profibus address, the *Family* and the *Station type* in the parameter window and click at [apply].
6.	Click at to configure all peripheral modules that are connected to the Profibus slave via the backplane bus. By means of [Auto] you may start the automatical address assignment for the periphery and monitor the address assignment via [MAP]. If you have an intelligent module, like e.g. a CP, you will see the adjustable parameters.

7.	After you have configured all slaves with the depending periphery, the bus parameters for the Profibus have to be calculated. Activate the function group Profibus in the network window. Click at the register "Busparameter" in the network window. Choose a baudrate and click at [calculate]. The bus parameters are calculated - [apply] them. Every time you change the assembly of your modules you have to repeat this calculation!
8.	Activate the master level in the network window and export your project into a 2bf-file.
9.	Transfer your 2bf-file into your DP master (see also "Transfer project").

Transfer project

For the data transfer from your programming PC to the DP master, you may deploy the VIPA "Green Cable". This enables you to transfer your project into the internal Flash-ROM of the DP master serial via the Profibus interface.

The "Green Cable" is available at VIPA and has the order number VIPA 950-0KB00.



Preconditions

The project engineering of your Profibus system is completed and you stored your project as 2bf-file.

Approach

- Assemble your system but leave the voltage supply off.
- Plug the "Green Cable" to the serial interface of your programming PC and to the Profibus interface of the DP master.
- Turn on the voltage supply of your PLC.
- Turn on the programming PC and start the SIP-Tool included to WinNCS. Choose the according COM interface and establish a connection via [Connect]. As soon as the connection is successful, the SIP-Tool announces an OK message in the status bar, otherwise an ERR message.
- Click at [Download], choose your 2bf-file and transfer this into the DP master.
- Close the connection and the SIP-Tool after the data transfer has been completed.
- Turn off the voltage supply of your PLC.
- Disconnect the "Green Cable" from the master.
- Connect the master with the Profibus network and turn on the voltage supply of your PLC again.

Now your master switches automatically to RUN with the created project.

Communication PC with DP master

Outline

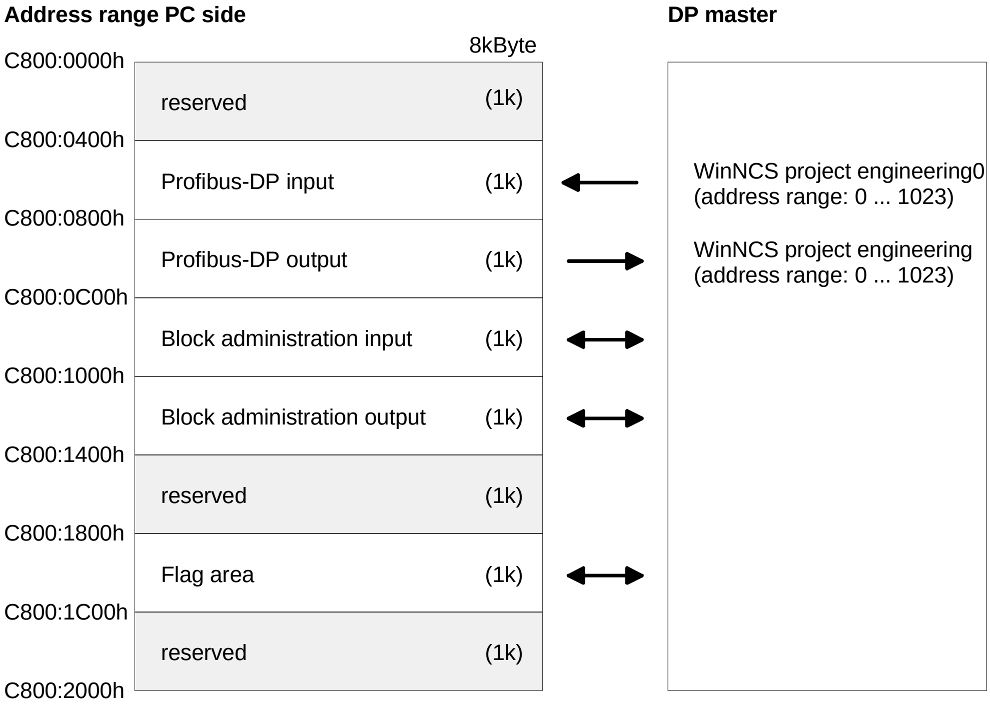
The communication between the PC and the DP master of the PC-CPU 486DPM takes place via a Dualport-RAM. The master communicates with the slaves and provides the data via the Dualport-RAM for the PC. Here you may create a maximum of 1024Byte data for inputs and outputs.

The PC accesses the Dualport-RAM via C-functions, that will be described in the following.

Structure Dualport-RAM

At the PC the address range of the Dualport-RAM is fixed. At the DP master you may access an address range from 0 ... 1023.

The Dualport-RAM has a size of 8kByte and is organized as follows:



Establishing a communication

This is the event sequence to establish a communication:

	PC		DP master
1.	Start-up		Start-up waits for `OPCODE`
2.	sets `OPCODE` to start	→	Master starts with *.2bf-configuration
3.	waits for `INFO_GUELTIG`		
4.		←	sets `INFO_GUELTIG`
5.			waits for BASP = 0
6.	sets BASP = 0	→	releases all outputs
7.	reads I/O data cyclically	←→	creates cyclically an I/O image of all configured DP slaves.

C-program

In the following you find a C-program to illustrate the access to the DP master.

```
/*****************************************************************************
*                               VIPA GmbH                                    *
*                                                                            *
* Image of the Profibus DP master in the DPRAM of the PC-CPU 486DPM.        *
*                                                                            *
* revision    date         changes                                           *
* -------------------------------------------------------------------------- *
* 1.00        15.02.2002   created                                           *
* 1.01        15.02.2002   release outputs with BASP                         *
* 3.00        03.06.2002   new structure of DPRAM. Needs firmware            *
*                          Bb000124.3xx in the master!                       *
******************************************************************************/

#define VERSION "V300"

#include <conio.h>
#include <stdio.h>
#include <dos.h>
#define BYTE unsigned char
#define WORD unsigned short

//The Dual Port-RAM of the master is addressed from the PC  with SEGMENT:OFFSET
//The Dos.h of the Borland C Compiler 4.5 shows e.g. the following definition:
//#define MK_FP(seg,ofs) ((void __seg *)(seg) + (void __near *)(ofs))


///////////////Location of the DP master in the address range of the PC/////////////////////

#define ADR_DPMASTER_SEG                  0xC800   //basic-segment address of Dual Port Ram

////////////////////////////////////////////////////////////////////////////////////////////


#define PC_CPU_486_DPM
/////////////////////////////////////////////////////////////////////////////////////////
//
// The DPRAM between CPU and DP master is a 8k memory.
// Every cell of this area may be accessed with R e a d / W r i t e .
// The access has a data width of 8 B i t  (= 1Byte).
//
// The DP master notices an 8k area.
// The LED- and VBUS- registers are only visible for the DP master.
// The areas 'PG channel' and 'page frame' may not be accessed
// because they are reserved for other devices!
// The PC notices an 8k area.
//
  #define ADR_IM208_FPGA_BASE                   0x0uL
  #define ADR_IM208_IN                          (ADR_IM208_FPGA_BASE + 0x0400uL)
  #define ADR_IM208_OUT                         (ADR_IM208_FPGA_BASE + 0x0800uL)
  #define ADR_IM208_IN_BLOCKVERWALTUNG          (ADR_IM208_FPGA_BASE + 0x0C00uL)
  #define ADR_IM208_OUT_BLOCKVERWALTUNG         (ADR_IM208_FPGA_BASE + 0x1000uL)
```

```
//Flags and Frame of  0x1800uL
#ifdef PC_CPU_486_DPM
  #define ADR_IM208_KONSISTENZ_INPUT_HANDSHAKE  (ADR_IM208_FPGA_BASE + 0x1AE0uL) //Lock input area
  #define ADR_IM208_KONSISTENZ_OUTPUT_HANDSHAKE (ADR_IM208_FPGA_BASE + 0x1AE2uL) //Lock output area
#endif

  #define ADR_IM208_MMC_IM208K_READ             (ADR_IM208_FPGA_BASE + 0x1AE4uL)
  #define ADR_IM208_MMC_IM208K_WRITE            (ADR_IM208_FPGA_BASE + 0x1AE6uL)
  #define ADR_IM208_MMC_CPU_LOCK                (ADR_IM208_FPGA_BASE + 0x1AE8uL)
  #define ADR_IM208_MMC_IM208K_LOCK             (ADR_IM208_FPGA_BASE + 0x1AEAuL)
  #define ADR_IM208_IM208K_ALARM_FLAG           (ADR_IM208_FPGA_BASE + 0x1AECuL)
  #define ADR_IM208_BASP                        (ADR_IM208_FPGA_BASE + 0x1AEEuL)
  #define ADR_IM208_IM208K_KACHEL_OK            (ADR_IM208_FPGA_BASE + 0x1AF0uL)
  #define ADR_IM208_LED_STATE                   (ADR_IM208_FPGA_BASE + 0x1AF2uL)
  #define ADR_IM208_INFO_GUELTIG                (ADR_IM208_FPGA_BASE + 0x1AF4uL)
  #define ADR_IM208_HOCHLAUFVERZOEGERUNG        (ADR_IM208_FPGA_BASE + 0x1AF6uL)
  #define ADR_IM208_FIRMWARE_VERSION            (ADR_IM208_FPGA_BASE + 0x1AF8uL) //4Byte
  #define ADR_IM208_HWCONF_RET                  (ADR_IM208_FPGA_BASE + 0x1AFCuL)
  #define ADR_IM208_HWCONF_OPC                  (ADR_IM208_FPGA_BASE + 0x1AFEuL)
  #define ADR_IM208_HWCONF_FRAME                (ADR_IM208_FPGA_BASE + 0x1B00uL) //256Byte frame length
                                                 //to  0x1C00uL
//----------------------------------------------------------------------------

  #define ADR_IM208_VBUS_TYPKENNUNG             (ADR_IM208_FPGA_BASE + 0x1F00uL) //visible only at DP master (R/W)
  #define ADR_IM208_VBUS_STECKPLATZKENNUNG      (ADR_IM208_FPGA_BASE + 0x1F02uL) //visible only at DP master (R)
  #define ADR_IM208_VBUS_ADRESSOFFSET           (ADR_IM208_FPGA_BASE + 0x1F04uL) //visible only at DP master (R)
  #define ADR_IM208_VBUS_INTERRUPT_CONFIG       (ADR_IM208_FPGA_BASE + 0x1F06uL) //visible only at DP master (R/W)
  #define ADR_IM208_VBUS_INTERRUPT_EVENT        (ADR_IM208_FPGA_BASE + 0x1F08uL) //visible only at DP master (R/W)
  #define ADR_IM208_REGISTER_LED                (ADR_IM208_FPGA_BASE + 0x1F0AuL) //visible only at DP master (W)
                                                //(steuert LEDs bei VBUS und page frame FPGAs)

  #define ADR_IM208_VBUS_PARAMETERDATEN         (ADR_IM208_FPGA_BASE + 0x1FB0uL) //visible only at DP master (R)
  #define ADR_IM208_VBUS_DIAGNOSEBEREICH        (ADR_IM208_FPGA_BASE + 0x1FD0uL) //visible only at DP master (W)
  #define ADR_IM208_VBUS_ALARMSTATUSBEREICH     (ADR_IM208_FPGA_BASE + 0x1FF0uL) //visible only at DP master (W)
                                                //to  0x2000uL
//----------------------------------------------------------------------------
/////////////////////////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////////////////
// Defines for the block administration in the DPRAM
//
// +-----------------+
// | X D B K . . . . | block administration Byte0
// +-----------------+
// | Y L L L L L L L | block administration Byte1
// +-----------------+
//
// D: Data-Bit:                  1=ocuppied with data, 0=free
// B: Block-Bit:                 1=Block (>= 2Byte), 0=only 1Byte
// K: Consistency-Bit:           1=consistent, 0=not consistent
// X: Semaphore of the CPU:      1=occupied, 0=free
// Y: Semaphore of the DP master:  1=occupied, 0=free
// LLLLLLL: Block length:        0x00..0x7F = 1...128Byte
//
// [The block administrationByte 0 has 4 free Bits. If the block length of LLLLLLL (1..128)
//  is not enough, it would be possible to write blocks between (1..2048)
//  by using this 4 Bit.]
//
// 4 cases with examples:
//
// 1. This Byte address is not occupied:
// Block administrationByte 0: x 0 x x x x x x
// The next Byte is again a block administrationByte 0 !
//
// 2. There is only 1Byte:
// Block administrationByte 0: x 1 0 x x x x x
// The next Byte is again a block administrationByte 0 !
//
// 3. There is a non-consistent block of 8Byte:
// Block administrationByte 0: x 1 1 0 x x x x
// Block administrationByte 1: x 0 0 0 0 1 1 1 (Block length: 0x07 = 8Byte)
// The following 6Byte are: 0x00
//
// 4. There is a consistent block of 128Byte:
// Block administrationByte 0: x 1 1 1 x x x x
// Block administrationByte 1: x 1 1 1 1 1 1 1 (Block length: 0x7F = 128Byte)
// The following 126Byte are: 0x00

  #define BLOCKVERWALTUNG_X   0x80
  #define BLOCKVERWALTUNG_D   0x40
  #define BLOCKVERWALTUNG_B   0x20
  #define BLOCKVERWALTUNG_K   0x10
  #define BLOCKVERWALTUNG_Y   0x80

//////////////////////OPCODE for starting the master//////////////////////////////////////////
#define HWCONF_OPC_WITHOUT  0x0003  //Master is started without hardware configuration; from Flash


//////////////////////Marker; shows master ready/////////////////////////////
#define INFO_GUELTIG        0x82    //Pattern for CPU marker for valid address information
```

```
//////////////////////Constants for consistency control of I/O data/////////////////////////
#define KONSISTENZ_UPDATE_FINISHED  0x0000
#define KONSISTENZ_UPDATE_START     0x0001
////////////////////////////////////////////////////////////////////////////////////////////


//Application
void main(void)
{
  BYTE Input;
  BYTE sVersion[5]={0,0,0,0,0};

  printf("voith.exe %s \n", VERSION);

  sVersion[0]= *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_FIRMWARE_VERSION +0 );
  sVersion[1]= *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_FIRMWARE_VERSION +1 );
  sVersion[2]= *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_FIRMWARE_VERSION +2 );
  sVersion[3]= *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_FIRMWARE_VERSION +3 );
  printf("Firware-Version DPMaster: %s\n", sVersion);


//Commands of the PC as C program:
//
//set OPCODE for start:
//
  *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_HWCONF_OPC ) = HWCONF_OPC_WITHOUT;
//
//
//waits for INFO_GUELTIG
//
 while( *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_INFO_GUELTIG ) != INFO_GUELTIG )
 {
  //wait
 }
//
//release outputs
//When setting a value > 0, the command output lock BASP is set.
//Then all inputs and outputs are set to 0.
//To enable the output values, set BASP to 0:
//
  *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_BASP ) = 0;
//
//cyclic Read / Write of I/O data
//
 while(1)
 {
  //initiate the update of the inputs
  *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_KONSISTENZ_INPUT_HANDSHAKE ) = KONSISTENZ_UPDATE_START;

  //wait for update complete
  while( *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_KONSISTENZ_INPUT_HANDSHAKE ) != KONSISTENZ_UPDATE_FINISHED )
  {
    //wait
  }
  //Now the Profibus inputs are consistent in the input area.

  /////////////////////////////////////////////////////////////////////////
  //Calculate the new output values from the consistent input data
  //and write it to the output area of the DPRAM.
  //
  //e.g. for the master an input byte at address 1 has been configured.
  //     Reading input address 1:
                    Input = *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_IN +1 );
  //
  //e.g. for the master an output byte at address 0 has been configured.
  //     Writing 0x01 to output address 0:
                    *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_OUT +0 ) = 0x01;
                    //write input of address 1 to output of address 1
                    *(BYTE far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_OUT +1 ) = Input;
  /////////////////////////////////////////////////////////////////////////

  //Initiate update of outputs
  *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_KONSISTENZ_OUTPUT_HANDSHAKE ) = KONSISTENZ_UPDATE_START;

  //wait for update complete
  while( *(WORD far *) MK_FP( ADR_DPMASTER_SEG, ADR_IM208_KONSISTENZ_OUTPUT_HANDSHAKE ) != KONSISTENZ_UPDATE_FINISHED )
  {
    //wait
  }
  //Now the output data has been put consistent to the Profibus
 }
}//end of main()
```

Communication by deploying COM modules

Outline

The PC provides an internal plug-in location for Hilscher COM modules. By deploying pluggable bus modules you may choose between all leading fieldbus system modules.

Every module includes a detailed description. Further on, you receive a function library in C as source code, to create own device drivers as well as a projecting tool.

COM modules

The COM modules are compact plug-in modules and they include a complete fieldbus connection at very small dimensions. Here the whole protocol stack is executed.

The connection to your PC is via a plug.

The data transfer between PC and module takes place via the Dualport-RAM interface, that may be accessed by means of the included software.

Software tools

Together with your COM module the Hilscher company delivers not only a detailed documentation but also a series of software tools and driver. The following components are in the consignment:

Toolkit

This C library serves the import of the COM module into PC applications.

COM interface

The COM interface provides COM components for object-orientated programming languages like Visual® C++ or Visual® Basic.

Device driver for Windows 9x/NT/2000/XP/CE and Linux

The device driver manages the complete administration of the COM module in the according operating system.

Projecting tool SyCon®

SyCon® is a projecting tool with an unique user interface for all COM modules. As basic for the project engineering GSDs (Gerätestammdateien) or Electronic Data Sheets ESD are used. These files define the properties of the bus participants.

Via a graphical editor you build the bus structure and locate the single participants. The tool also supports detailed diagnosis and commissioning hints; for e.g. error messages are shown in plain text.

Profibus assembly guidelines

General

- A VIPA Profibus network may only be built-up in linear structure.
- Profibus exists of minimum one segment with at least one master and one slave.
- Profibus supports max. 125 participants.
- For each segment a max. of 32 participants is allowed.
- The maximum segment length depends on the baudrate:

9.6...187.5kBaud	→	1000m
500kBaud	→	400m
1.5MBaud	→	200m
3...12MBaud	→	100m

- You may build-up a max. of 10 segments. The segments are connected together with repeaters. Every repeater is taken as a participant.
- All participants are communicating with the same baudrate. The slaves are adjusting themselves to this baudrate.
- You have to terminate the bus at both ends.
- Master and slaves are combinable at will.

Assembly and link-up to Profibus

- Assemble your Profibus system with the according modules.
- Choose an address at your bus coupler that is not yet in use.
- Transfer the delivered GSD file into your configuration tool and configure your system.
- Transfer your project into the master.
- Connect the Profibus cable to the coupler and turn on the power supply.

Note!

The Profibus cable has to be terminated with its ripple resistor. Please make sure to terminate the bus at the particular last participant by activating the terminating resistor.

Transfer medium

As transfer medium Profibus uses an isolated twisted-pair cable based upon the RS485 interface.

The RS485 interface is working with voltage differences. Though it is less irritable from failures than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure.

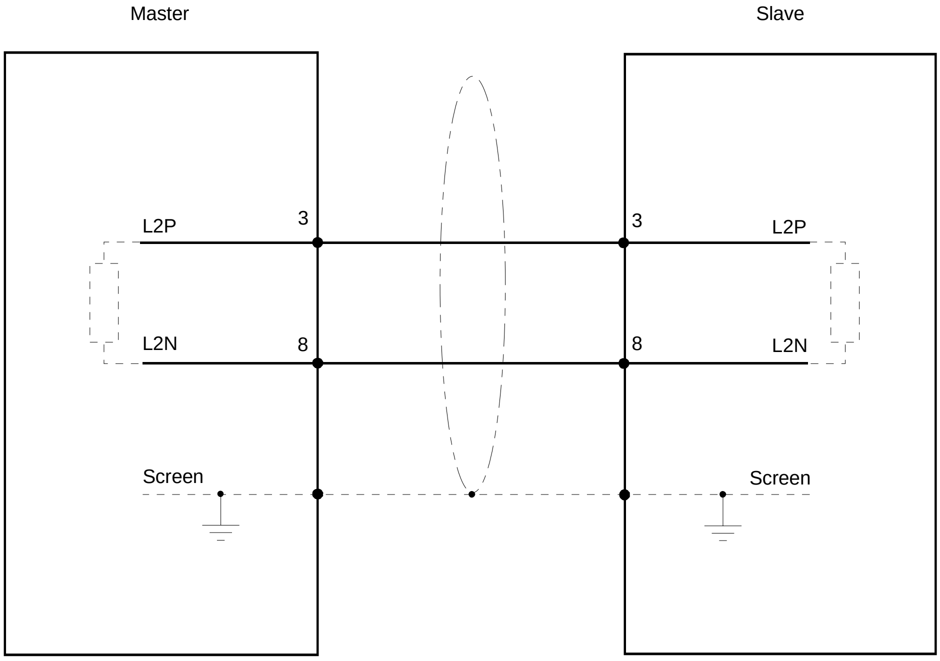
At the front side of your PCs is a 9pin slot. Via this slot you connect the coupler as master directly into your Profibus network.

For every segment a max. of 32 participants is allowed. The single segments are connected together via repeaters. The max. segment length depends on the baudrate.

Profibus uses a baudrate between 9.6kBaud and 12MBaud, the slaves are following automatically. All participants of the network are communicating with the same baudrate.

The bus structure under RS485 allows an easy connection resp. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

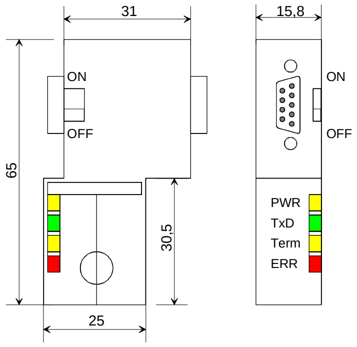
The picture shows a Profibus connection under RS485 with outlined terminating resistors:



Bus connector

In systems with more than two stations all partners are wired in parallel. For that purpose the bus cable must be connected in a continuous uninterrupted loop.

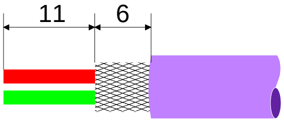
Via the order number VIPA 972-0DP10 you may order the bus connector "EasyConn". This is a bus connector with switchable terminating resistor and integrated bus diagnosis.



all in mm

To connect this connector please use the standard Profibus cable type A according to EN50170.

Under the order no. 905-6AA00 VIPA offers the "EasyStrip" deisolating tool, that makes the connection of the EasyConn much easier.



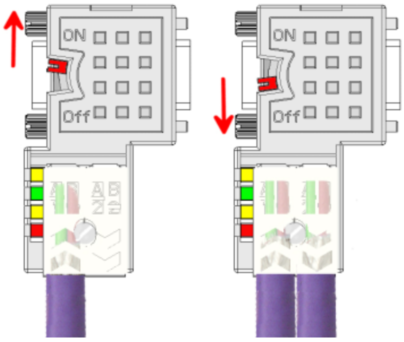
all in mm

Attention!

The bus cable has always to be terminated with the ripple resistor to avoid reflections and therefore communication problems!

Termination

The bus connector is provided with a switch that may be used to activate a terminating resistor.



Attention!

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

Note!

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

Appendix

A Index

M.Stich