DPKONFIG

Technical Manual

Date:	19.12.1996
Drafted by:	R. Christophersen
Revision:	02.02.1999
ID of Document:	942.5042.53

Preface

The user requires comprehensive information in order to make full use of the technical abilities of this product. This manual is intended for hardware developers, project engineers, and programmers who plan on using this product.

The present publication has been designed to provide you with a complete range of structured information. The following pages contain information which should make it easier for you to use this manual. You will be provided with information on how this manual is structured.

Despite all efforts from our side some problems may not be covered by this manual. This is mainly due to the broad range of possible software and VMEbus applications. Please do not hesitate to contact DMS if you experience any problems.

Contents

- How to install the software (Chapter 1)
  This chapter summarizes all data necessary for installing of this product.

- Detailed description of the software (Chapter 2)
  Chapter 3 contains the detailed descriptions of the product incl. board depending software like drivers, decriptors and how to program it.

- Appendix (Chapter 3)

The last pages of this manual contain a number of forms (e.g. 'Reader Comments'). Please fill in those forms if you would like to propose any changes, corrections or supplements. We suggest that you mail the sheet back to us. Your suggestions will help us to improve the quality of future editions of this manual.

Printing Conventions

In order to improve its readability this document is structured like a menu.

- The first pages of this document contain a chart of contents.
- Pages, illustrations, and tables are numbered consecutively.
- Abbreviations have been used for a number of terms. You will find a list of abbreviations in the Appendix Section.
- Footnotes are indicated by text printed in superscript such as numbers (e.g. "[1]"), or stars ("*"). Generally, the comments related to a footnote can be found at the bottom of the page. Bullets (•) and dashes are used in most cases (like for example in this list) to indicate an enumeration.
- Cross references are used as follows: "(see Chapter 3.3.2)" refers to Section 3.2 of chapter 3.
- All dimensions given in drawings, sketches, and tables are in millimeters (mm).
- Value ranges are given in the following form: 17 .. 21 means a range from 17 to 21.
- Hexadecimals are indicated by a leading „$„.
- You will find that outlined boxes like the one below often contain very important information:

Warning

Please take the time to read through the section 'Information on Operational Safety' which is located at the end of this introduction. It contains definitions for certain terms like '**Warning', 'Danger', 'Caution', 'Note'**

Information on Operational Safety

This manual contains all information necessary for the designated use of the products described in it. It is written for the use by qualified personnel. The term 'qualified' as used in this manual or as printed on the product itself refers to persons

- that are working in the field of research and development
- or that are members of the project staff and are familiar with all safety regulations used in automation technology.

Safety Precautions

The following safety precautions will help to prevent personal injuries as well as the protect the product and other devices connected to it.

Safety precautions and warnings that may threaten the life or health of the operator or maintenance personnel or that may result in damage to property will be indicated by signal terms defined below. The terms used in this manual or on product labels have the following meaning:

Danger

means that death, major injuries, or major damage will occur if the appropriate safety precautions are not obeyed.

Warning

means that death, major injuries or major damage may occur if the appropriate safety precautions are not obeyed.

Caution

means that minor injuries or damages may occur if the appropriate safety precautions are not obeyed.

Note

this section contains useful hints about the product, product handling or the part of the manual which it refers to explicitly.

Normal use of the product

Warning

- The device/system/product must not be used in cases other than those mentioned in the catalogue or the manual and may only be used in conjunction with third party components that are either recommended or approved by DMS.
- Professional transport, storage, erection/installation as well as careful operation and servicing are a prerequisite for the trouble free and safe operation of the device.

Validity of this Manual

This manual is valid for:

Product: DPKONFIG Vers. 4.2.5 or higher

Table of Contents

Preface 3
Contents 3
Printing Conventions 4
Validity of this Manual 6
Table of Contents 7
Tabeles 9
Illustrations and Diagrams 9

1 DPKONFIG Overview 10
1 .1 General information 10
1 .2 Supported hardware 10
1 .3 Software licence 11
1 .4 The DPKONFIG Package 11
1 .5 Order Codes 11
1 .6 Safety precautions 11

2 Software installation 12
2 .1 System requirements 12
2 .2 Installing of DPKONFIG 12
2 .3 DPKONFIG directorys 14

3 DPKONFIG 15
3 .1 Working with DPKONFIG 15
3 .2 Main Window (DPKONFIG) 15
3 .2 .1 DPKONFIG menu 16
3 .2 .2 DPKONFIG function keys 17
3 .3 Message Field 17
3 .4 Masters configuration (DP Master Class1 Setup) 18
3 .5 Configuration of Slaves (DP Slave Editor) 19
3 .5 .1 DP-Slave editor function keys 21
3 .6 IEC1131 Variable Editor 21
3 .7 Diagnostic data 22
3 .8 DPR Debug 25
3 .9 PROFIBUS Configuration Parameters 26
3 .9 .1 Bus-/ Master configuration parameters 26
3 .9 .2 Slave configuration parameters 28
3 .9 .3 Structure of the DPKONFIG configuration file: 30
3 .10 DPKONFIG with different hardware configurations 32
3 .10 .1 DPKONFIG with VME-PROFI 32
3 .10 .1 .1 VME-PROFI - DPKONFIG with RS232 connection 33
3 .10 .1 .2 VME-PROFI - DPKONFIG with TCP/IP connection 34
3 .10 .2 PCU - DPKONFIG with RS232 connection 35
3 .10 .3 IPE - DPKONFIG with TCP/IP connection 36
3 .10 .4 PCI-40 / PCP-DP - DPKONFIG with dualport RAM connection 37
3 .10 .5 PMC-PROFI - DPKONFIG with RS232 connection 37

4 Appendix Section 38
4 .1 List of Abbreviations 38
4 .2 References 38

4 .3 Reader Comments 39

Tabeles

Table 1: Supported hardware 10

Illustrations and Diagrams

Figure 1: DPKONFIG main window 10
Figure 2: DPKONFIG Setup 12
Figure 3: Window of the PROFIBUS-DP Configurator 15
Figure 4: Master Klasse1-Setup window 18
Figure 5: Setup of a new Slave 19
Figure 6: Example for a configured Slave 20
Figure 7: VarEdit window 21
Figure 8: Slave with Diagnostic data (Output overload) 22
Figure 9: Diagnostic window 24
Figure 10: DPR Debug 25

1 DPKONFIG Overview

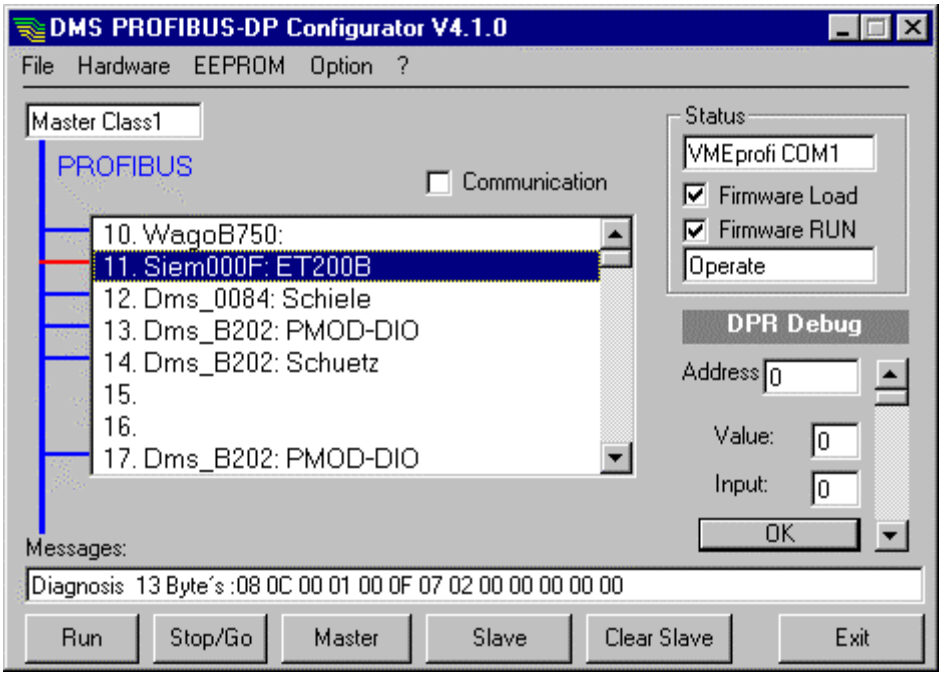


Figure 1: DPKONFIG main window

1 .1 General information

DPKONFIG is a Windows95/98/NT configuration tool for DMS DP-Master1 boards. With this tool the configuration of the DP-Slaves can generated by reading the GSD-files of the Slaves. The Slaves configuration can use up to 64 Cfg-Bytes and up to 64 User-bytes. For the test of the PROFIBUS system, the live- and diag-bits are shown online, the diagnostic data of the Slaves can be red and the I/O-data within the DP-Master's dualport RAM can be accessed.

1 .2 Supported hardware

DPKONFIG can be used with different DMS DP-Master1 interface boards. The connection between the Windows PC (DPKONFIG) and the target hardware are done via the serial interface, via a TCP/IP network or via the dualport RAM.

Hardware	connection
VME-PROFI VME-PROFI3 VME-PROF-A	RS232 and TCP/IP
IPE	TCP/IP
PCU	RS232 and TCP/IP
PCI-40	DP-RAM
PCP-DP	DP-RAM
PMC-PROFI	RS232 and DP-RAM

Table 1: Supported hardware

1 .3 Software licence

DPKONFIG is sold as company license and can be used for all company's systems, which are used with DMS hardware.

1 .4 The DPKONFIG Package

The DPKONFIG package includes:

- a Disc set (PC1.44 format)[1] or CD-ROM
- a software Licence
- technical Manual

[1] DPKONFIG can also be included in the DOAS CD-ROM Software package.

1 .5 Order Codes

DPKONFIG	PROFIBUS DP configurator for Windows95/98/NT
Hardware with PROFIBUS DP-Master1 interface:	
VME-PROFI	VMEbus PROFIBUS-Interface with 6 HE frontplate, 3 Mbit/s incl. PROFIBUS-DP-Master Software
VME-PROFI3	VMEbus PROFIBUS-Interface 3 HE, 3 Mbit/s incl. PROFIBUS-DP-Master Software
IPE	VMEbus with 68040 CPU, 64 MHz, 8 MB RAM, IEC-1491 and PROFIBUS DP-Master interface 3 Mbit/sec, RS232, 20 mA, 2 Memorycards
VME-PROF-S	VMEbus 6HE PROFIBUS-DP Slave Interface 12Mbit/s
VME-PROFS3	VMEbus 3HE PROFIBUS-DP Slave Interface 12Mbit/s
PMC-PROFI	PMC Mezzanine Module with PROFIBUS DP-Master interface, 3 Mbit/s incl. PROFIBUS-DP-Master Software
PCI-40	PC ISA-Slotcard with 68040 CPU, 64 MHz, 8 MB RAM, IEC-1491 and PROFIBUS interface 3 Mbit/sec
PCP-DP	PC ISA-Slotcard with PROFIBUS interface 3 Mbit/sec
PCU	PROFIBUS Control Unit with graphic display 320*240 pixel, 68331 CPU, 512 kB RAM, 2 MB Flash, Ethernet-Controller, OS9-Extended, TCP/IP, DP-Master Firmware
PST	PROFIBUS DP Slave Terminal, display with 2 lines a 24 characters
Accessories:	
PROFI-STV	PROFIBUS Bus connector (9 pol. DSUB) with switchable terminating resistors
PROF-KABEL	PROFIBUS cable

1 .6 Safety precautions

Caution

- DPKONFIG can be used to set DP-Slave outputs. Never set outputs with a running machine attached to the outputs!

2 Software installation

2 .1 System requirements

Windows95/98 or WindowsNT 4.0 must be installed. The DPKONFIG software will be installed on disk c:.

2 .2 Installing of DPKONFIG

Before installing DPKONFIG view the readme-file on the last DPKONFIG disk. This file will include the last informations about DPKONFIG and the installation of this package.

DPKONFIG consists of 5 Disks. Disk 1 – 5 contains the complete DPKONFIG Vers. 4.1 installation software.

1. Insert Disk 1 into drive a: and start **`setup.exe`**..
2. For the DPKONFIG / DPM1 setup click the Start-Button.

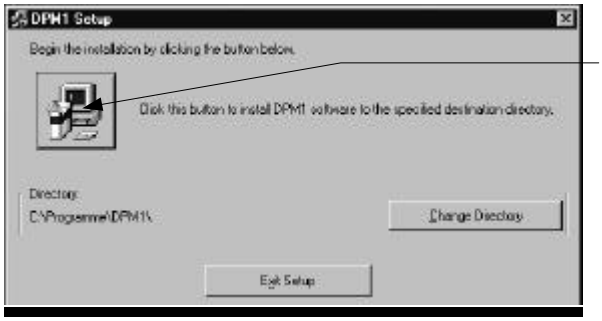


Figure 2: DPKONFIG Setup

The installation will install the software on disk into the sub directory „c:\DMS“.

> **Note:**
> Do not change the directory!

See also Note:A

3. Additional disks are requested by the setup program. When the installation are finished, the message „setup completed“ appears.
   See also Note:B

4. This step is only necessary, if a PCI-40 should be used in this system:
   The serial-drivers for the PCI-40 must be registered.
   for Windows95/98: Open directory **`c:\DMS\driver`** with the explorer and double-click the file **`inst95.bat.`**
   for WindowsNT: Open directory **`c:\DMS\driver`** with the explorer and double-click the file **`instNT.bat.`**
5. Shutdown Windows and reboot the system.
6. The Software installation (DPKONFIG Vers. 4.1) are finished.

To test the installation, start dpkonfig through the start-menü/programms. Now the

DPKONFIG window will be opened.

Note A

On some new installed systems some system files are missing or not actual. In this case the following messages appears:

"Setup kann nicht fortgesetzt werden, da einige Systemdateien
auf Ihrem System nicht mehr aktuell sind. Klicken Sie auf 'OK',
wenn setup diese Dateien jetzt jetzt für Sie aktualisieren soll.
Sie müssen windows neu starten, bevor Sie das Setup wieder
Ausführen können. Klicken Sie 'Abbrechen', um Setup zu Beenden,
ohne Systemdateien zu aktualisieren"

Click 'OK' to update these system files.

It is necessary to reboot the windows system to run Setup again.
Therefore the next message appears, after the update of these files:

"Möchten Sie Windows jetzt neu starten? Wenn Sie 'Nein' wählen,
können Sie Setup erst wieder ausführen, nachdem das System zu
einem späteren Zeitpunkt neu gestartet wurde."

Select 'Yes' / 'Ja' to reboot your Windows system and repeat step 1

Note B

On systems with an English windows version the registration of one *.ocx file could cause the following message at step 3:

"Dpkonfig v4.0-Setup
Ein Fehler ist beim Regristieren der Datei
'C:\WINDOWS\SYSTEM\DBGRID32.OCX aufgetreten
Abort Retry Ignore "

Select 'Ignore'. Than the installation will be continued.

2 .3 DPKONFIG directorys

The DPKONFIG-files are installed in the directory c:\dms. Also additionally files, which are only required, if a PCI-40 or PCP-DP is used together with DPKONFIG, are installed. The following table shows the structure of the directories and the installed files.

Directory	Files	reqired by		
		DPKONFIG	PCI-40	PCP-DP
c:\dms\cmds	Dpkonfi.exe	X		
	Pcicons.exe		X	
	Winrbf.exe		X	
c:\dms\downloads	dpm1.lo		X	X
	Os9pci.lo		X	
c:\dms\driver	Dmspci.dll	X	X	X
	Dmsadr.*		X	
	inst*.bat		X	
	Pci40con.*		X	
	Seriala.*		X	
c:\dms\gsd	*.*	X		
c:\dms\help	*.*	X		
c:\dms\os9	*.*		X	
c:\dms\setup	*.*	X		

3 DPKONFIG

3 .1 Working with DPKONFIG

To start DPKONFIG use start-menu/programms/dpkonfig or double-click the file c:\dms\cmds\dpkonfi.exe. Now the DPKONFIG window appears.

3 .2 Main Window (DPKONFIG)

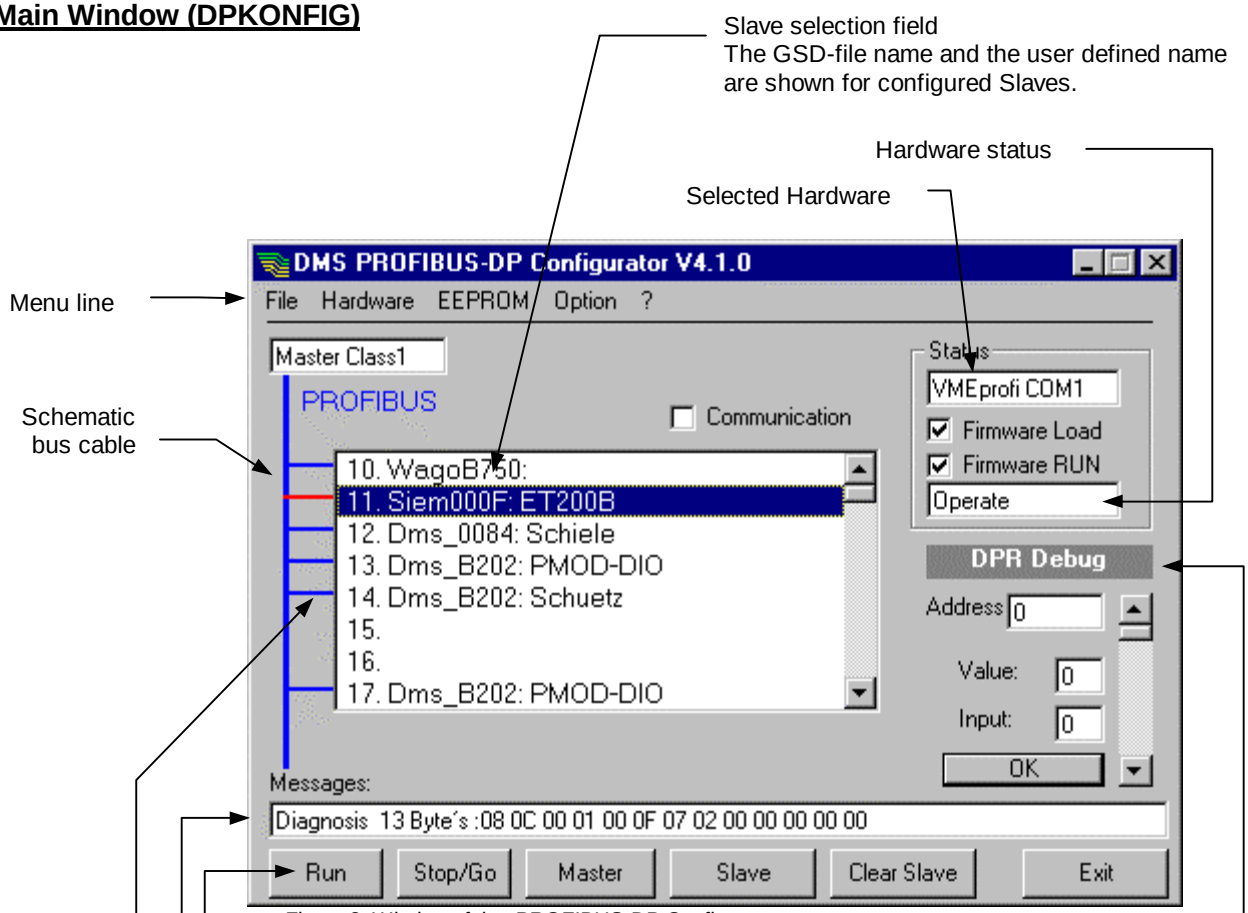


Figure 3: Window of the PROFIBUS-DP Configurator

3 .2 .1 DPKONFIG menu

File File menu

- **New** create new configuration
- **Load** load a file
- **Store** store the configuration
- **Print** prints the Master and Slave configuration
- **Exit** close DPKONFIG

Hardware Hardware menu

- **no Hardware** no hardware connected (--> default setting), the DPKONFIG is only used in offline-mode to create configuration files
- **PCI-40 /PCP-DP** a PCI-40/PCI-40S/PCP-DP is installed in the PC and controlled through the dualport-RAM-interface
- **PMC-PROFI** a PMC-PROFI are controlled through the *COM1* or *COM2* interface. The PMC-PROFI must rum the DP-Master Firmware Vers. 5.9 or higher. A PMC-PROFI is installed in the PC and controlled through the dualport-RAM-interface
- **VMEprofi** a VME-PROFI are controlled through the *COM1* or *COM2* interface. The VME-PROFI must rum the DP-Master Firmware Vers. 4.0 or higher and the option „v“ (= remote controlled) must be set. *TCP/IP s*elects the target for the TCP/IP connection.
- **PCU** a PCU are controlled through the *COM1* or *COM1* interface. *TCP/IP s*elects the target for the TCP/IP connection.
- **IPE** a IPE are controlled through the *TCP/IP* interface.

EEPROM menu to store/load configuration data into the onboard EEPROM or a local file-system. This menu is only valid for boards, who provides an onboard EEPROM (for example VME-PROFI with serial connection) or which can use a local file system (for example IPE / VME-PROFI with TCP/IP connection) to store the configuration data

- **Burn** store the configuration into the EEPROM / local autoload.dp1 file
- **Read** load a configuration from EEPROM / local autoload.dp1 file
- **Start** Restarts the board with configuration data from EEPROM / local autoload.dp1 file
  (Note: this internal data could be different from the shown configuration)
- **Stop** Stop the DP-Master

Option

- **English** sets English language
- **German** sets German language

? The Help-file according to the set language are called.

Note:

If the hardware setting are changed, restart the DP-Master software of the selected board by clicking the function key „RUN“.

3 .2 .2 DPKONFIG function keys

Run Restart the DP-Master Software with the actual configuration

Stop/go by clicking this key, the DP-Master software are stopped / continued

Master a sub-window for the configuration of the Master-parameter are opened

Slave a sub-window for the configuration of the selected Slave are opened

Clear Slave the configuration of the selected Slave are cleared

Exit stops DPKONFIG

3 .3 Message Field

This field is used to display

- DPKONFIG status messages like
  „Slave 17 selected“
  „VME-PROFI at COM1 found“

- Diagnostic data of selected and running Slaves like
  „Diagnosis 13 Byte´s :09 0C 00 01 00 0F 07 02 00 00 00 00 00“

- Status messages of the DP-Master firmware like
  „Message : E7->0“
  The status codes are described in the service interface chapter of the DP-Master board manual.

3 .4 Masters configuration (DP Master Class1 Setup)

To configure the Master parameters click the function key „Master“. The „Master Klasse1-Setup“ window appears.

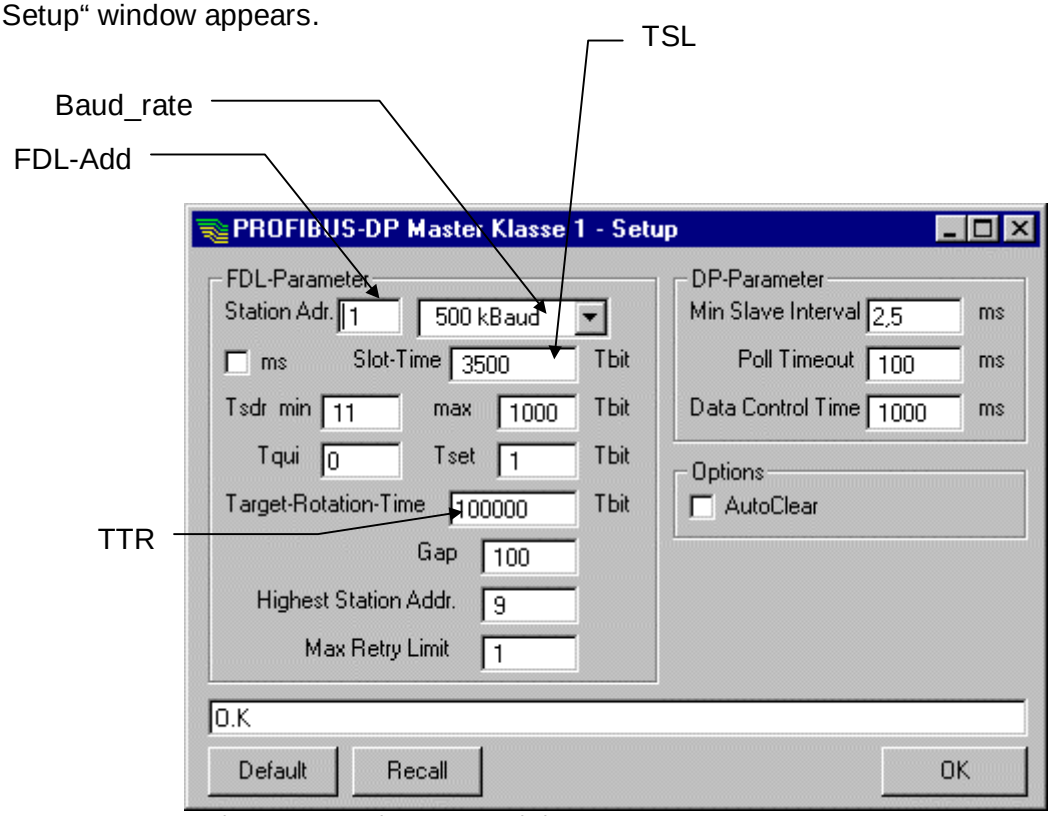


Figure 4: Master Klasse1-Setup window

The actual bus-parameters are shown. These parameters can be changed by editing the data fields. See chapter 3 .9 .1 for more information’s about these parameters.

To set the baudrate depending default values click function key „Default“.
With function key „Recall“ you can get the values back, which were set, when the window was opened.

3 .5 Configuration of Slaves (DP Slave Editor)

To configure a new Slave, select the Slave-address in the Slave-selection field (for example 25) and then click the function key „Slave“. An empty „DP Slave Editor“ window appears.
All necessary data (address, ID-number, User-Bytes, configuration-bytes, DP-RAM offset of the input and output data) must be set within this window.

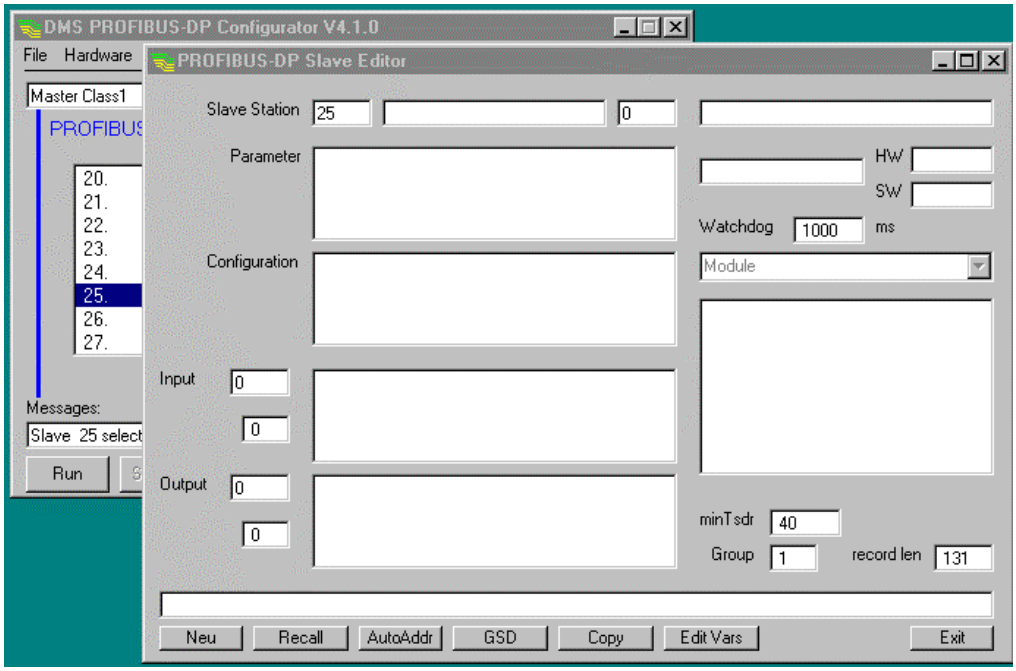


Figure 5: Setup of a new Slave

This necessary data can easily be inserted by selecting the GSD-file of the Slave. To do this, click the function-key „GSD“ and chose the correct GSD-file. (The filename consists of 4 character for the manufacturer following by the 4 character ID-number).

Note:
The GSD-file must be in the directory **c:\dms\gsd** !

The following data are automatically inserted from the GSD-file:

- ID-Number
- manufacturer name
- Device type
- HW and SW revision
- User-parameter
- Configuration-bytes (only for compact stations)

If a modular station are selected, a module selection field are activated. Now the installed modules must be selected. The configuration bytes for each selected module are added to the configuration-data field.
Which modules are necessary (installed modules, dummy module) and the order of this modules depends of the physical hardware and the type of Slave. Please check the Slave's documentation or comments in the GSD-files.

Now the offset of the input and output data in the dualport-RAM of the board must be set. This offset can be set manually or automatically by clicking the „AutoAdr“ function key. The data must be within the 4 kB I/O-range (offset $0000 to $0fff).

Additionally a name can be edit in the field next to the station address (max. 20 characters). This name will appear later in the Slave-selection field of the main-window.

The following figure shows the window of a complete configured slave.

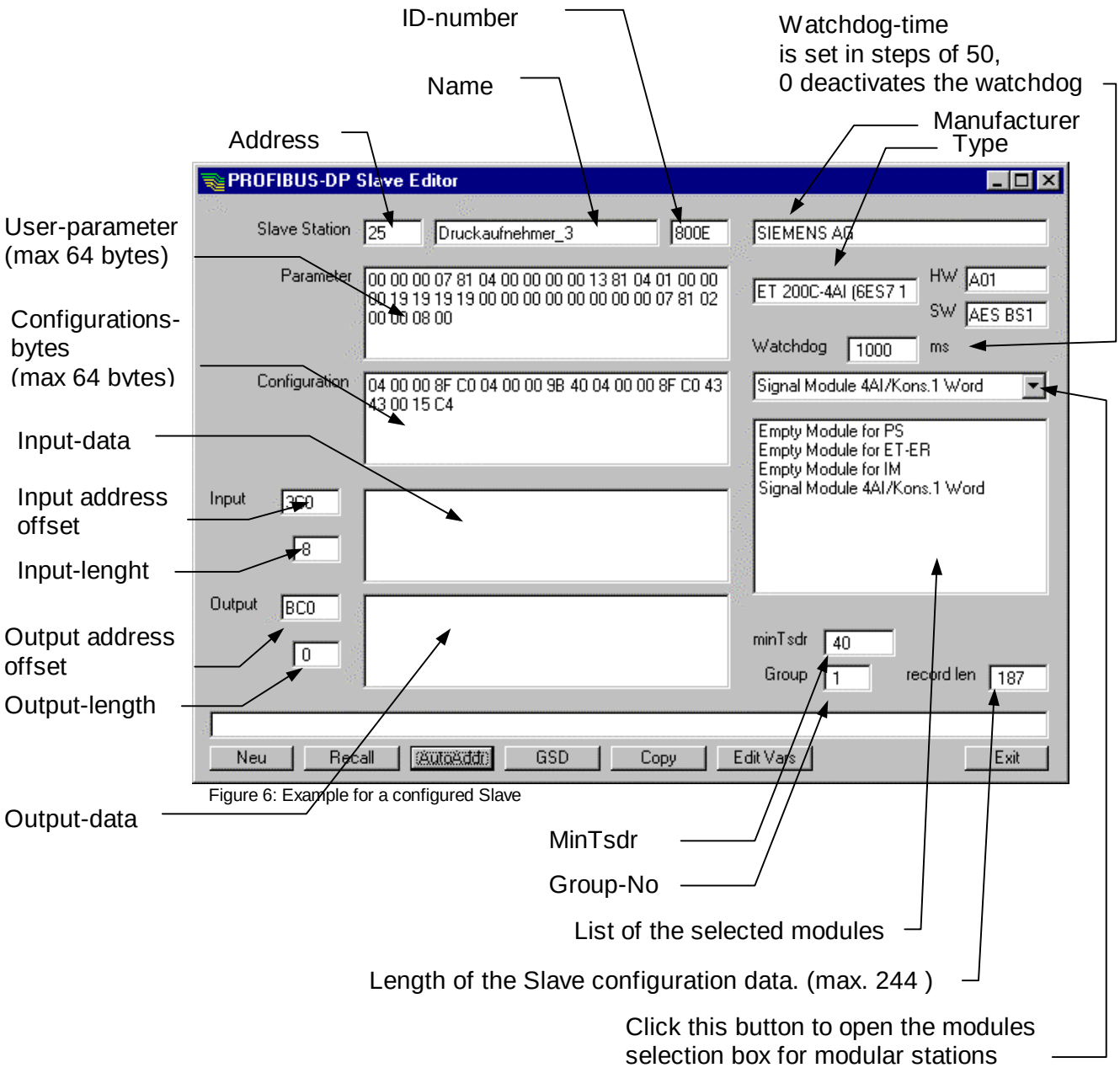


Figure 6: Example for a configured Slave

The Input and output data are only valid by DP-RAM and TCP/IP. The data are updated once when the window opens. To modify output-data use the DPR-Debug function.

More information’s about the Slave’s parameters can be found in Chapter 3 .9 .2 .

3 .5 .1 DP-Slave editor function keys

New Clears the Slave window

Recall With function key „Recall“ you can get the values back, which were set, when the window was opened.

AutoAdd The Input and Output offsets of this slave are set as following:
Input_offset = $000 +(Slave_add - 10) * $40
Output_offset = $800 +(Slave_add - 10) * $40

GSD Opens a GSD-file

Copy The configuration of this Slave can be copied to a selected slave address

VarEdit Calls the IEC1131 variable editor for MULTIPROGwt

3 .6 IEC1131 Variable Editor

The variable editor is a tool, included in DPKONFIG, to assign MULTIPROGwt[1] variables to the physical hardware, in this case to the PROFIBUS I/O’s.

For the variable editor DPKONFIG uses the MULTIPROGwt DDE interface. Type **MWT /DDE** to start MULTIPROGwt. With the VarEdit function BOOL and INT variables could be assigned to the physical PROFIBUS I/O’s. VarEdit adds or updates the required references in this two files.

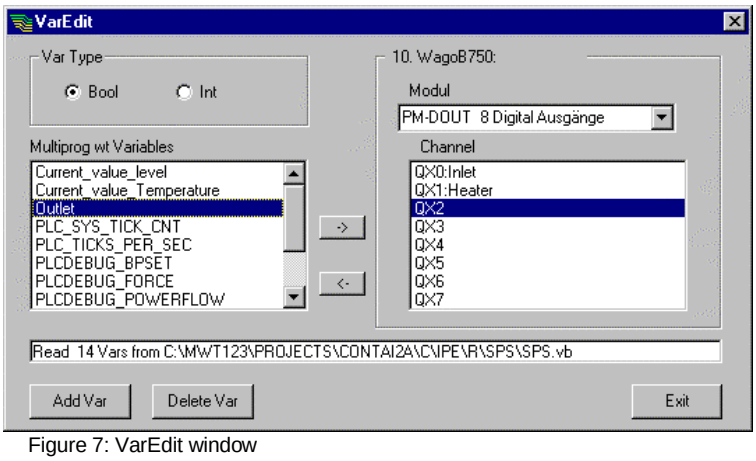


Figure 7: VarEdit window

[1] DMS has different types of CPU boards (VMEbus-CPUs, PC-ISA-Slotcards and embedded operator panels). All of these CPU-boards are using the realtime multitasking multiuser operatingsystem OS-9. For PLC applications DMS has portated the IEC 1131-3 compatible operating system Klöppner & Wiege ProConOS (PROgrammable CONtroller Operating System) for OS-9.
The programming system for this software PLC is MULTIPROGwt. This is a Windows95/98/NT program tool which supports all five IEC1131-3 programming languages.
For the assignment of PROFIBUS I/O’s, a link to DPKONFIG was integrated. DPKONFIG can be directly started from the MULTIPROGwt project tree.

3 .7 Diagnostic data

The diagnostic data of Slave stations can be viewed in the message field. The data are shown after a Slave is selected by clicking the address/name in the Slave-selection field. Diagnostic data are only available, if the status are „online/operate“.

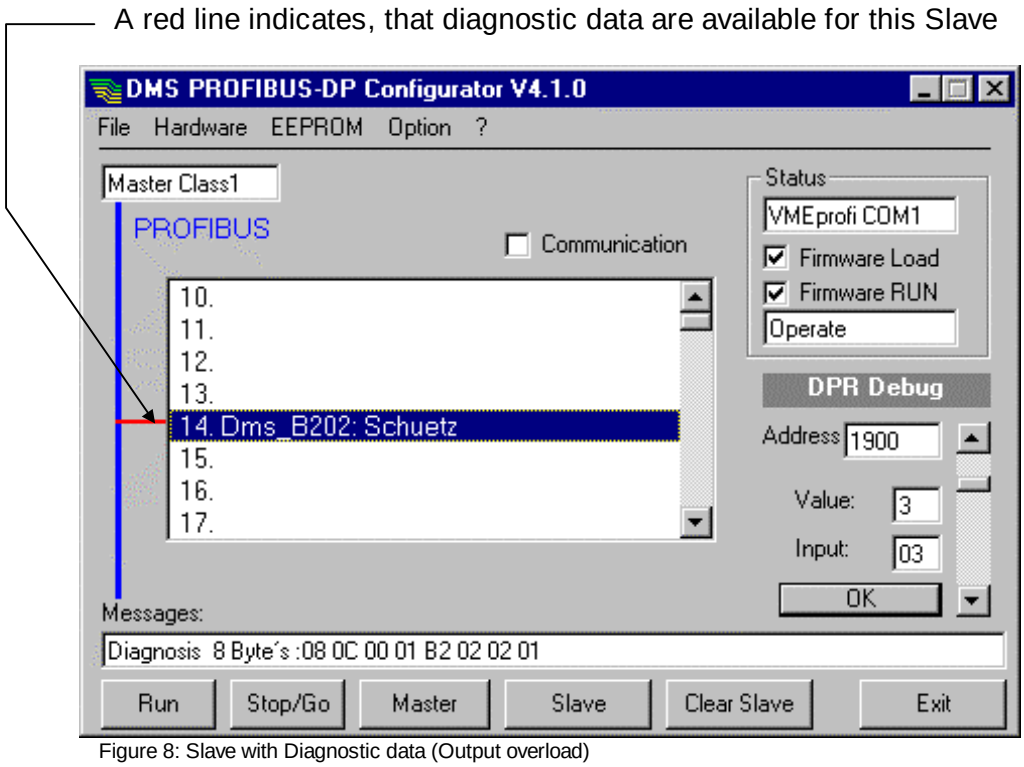
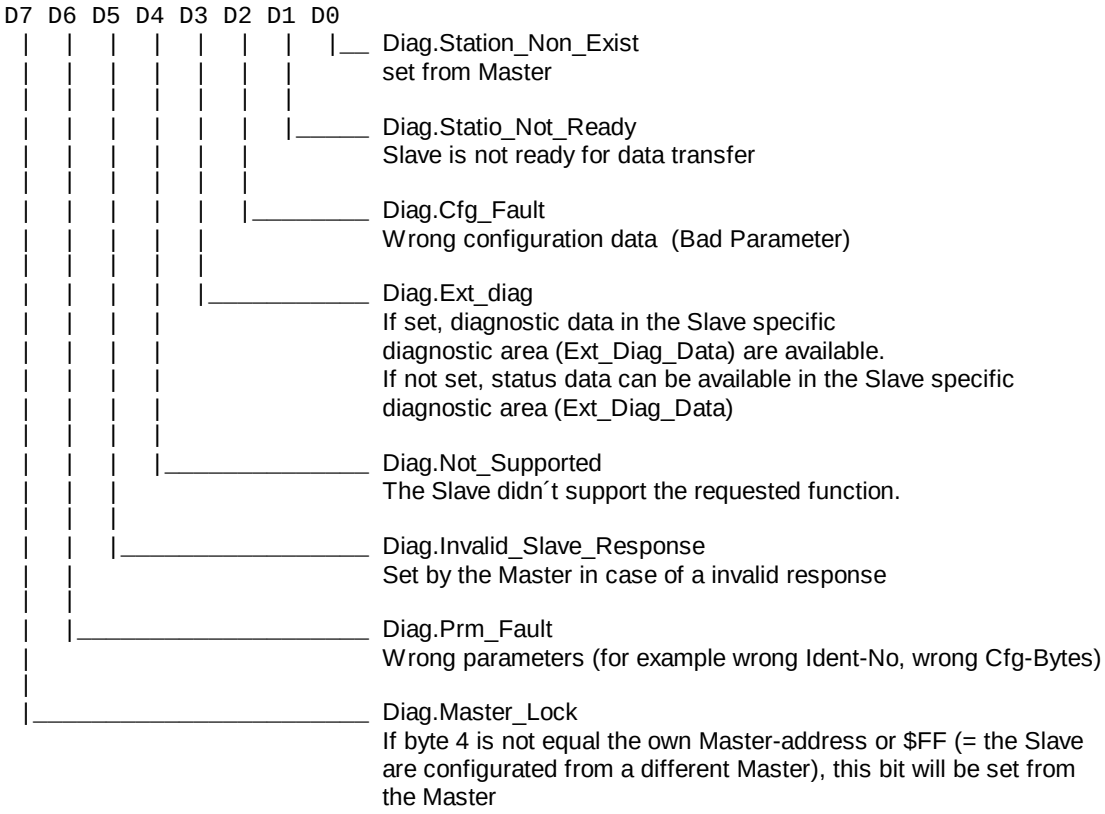


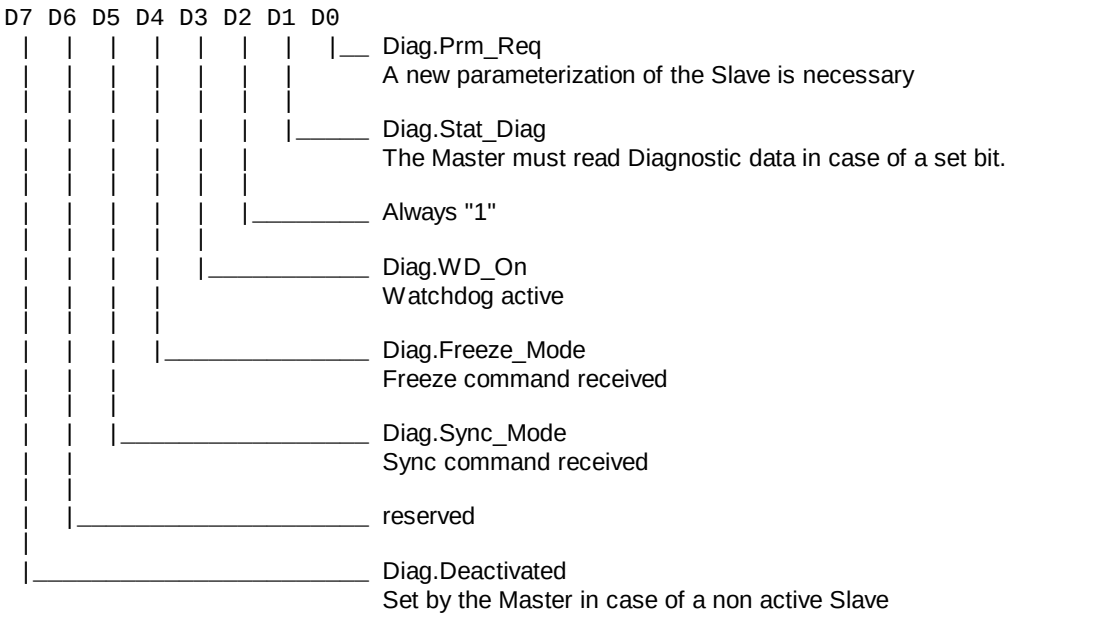
Figure 8: Slave with Diagnostic data (Output overload)

Bytes 1 - 6 are general diagnostic data and show the operating mode of the Slave.
Bytes 7 - 244 are optional external diagnostic data for module or channel status. The coding for the external data can be found in the DIN19245 part 3, the Slave documentation and the GSD-file.

Byte 1:


D7 D6 D5 D4 D3 D2 D1 D0
Diag.Station_Non_Exist
set from Master
Diag.Statio_Not_Ready
Slave is not ready for data transfer
Diag.Cfg_Fault
Wrong configuration data (Bad Parameter)
Diag.Ext_diag
If set, diagnostic data in the Slave specific
diagnostic area (Ext_Diag_Data) are available.
If not set, status data can be available in the Slave specific
diagnostic area (Ext_Diag_Data)
Diag.Not_Supported
The Slave didn´t support the requested function.
Diag.Invalid_Slave_Response
Set by the Master in case of a invalid response
Diag.Prm_Fault
Wrong parameters (for example wrong Ident-No, wrong Cfg-Bytes)
Diag.Master_Lock
If byte 4 is not equal the own Master-address or $FF (= the Slave
are configurated from a different Master), this bit will be set from
the Master

Byte 2 :


D7 D6 D5 D4 D3 D2 D1 D0
Diag.Prm_Req
A new parameterization of the Slave is necessary
Diag.Stat_Diag
The Master must read Diagnostic data in case of a set bit.
Always "1"
Diag.WD_On
Watchdog active
Diag.Freeze_Mode
Freeze command received
Diag.Sync_Mode
Sync command received
reserved
Diag.Deactivated
Set by the Master in case of a non active Slave

Byte 3 : D7 D6 D5 D4 D3 D2 D1 D0
| |__|__|__|__|__|__| reserved
|
|_____ Diag.Ext_Overflow
Overrun of Diagnostic Data

Byte 4 : D7 D6 D5 D4 D3 D2 D1 D0
|__|__|__|__|__|__|__|__| Diag.Master_Add
= <Address of Master> after parameterization
= $FF if the Slave is not parameterized

Byte 5 : D7 D6 D5 D4 D3 D2 D1 D0
|__|__|__|__|__|__|__|__| Identnummer High

Byte 6 : D7 D6 D5 D4 D3 D2 D1 D0
|__|__|__|__|__|__|__|__| Identnummer Low

Byte 7 ..32 (bzw 244) :
Extern Diagnostic Data see DIN 19245-3 or Slave dokumentation.

If a Slave is selected with the right mouse key, a diagnostic window opens. This window shows the bits of the diagnostic bytes 1 to 3. Each set bit is highlighted. The byte 4 (master address), byte 5/6 (ID-number of the Slave) and the extended diagnostic bytes 7 - x are shown directly. The contents of this window are updated every time, when another Slave is selected. The window remains active, until it's killed by the X-button.

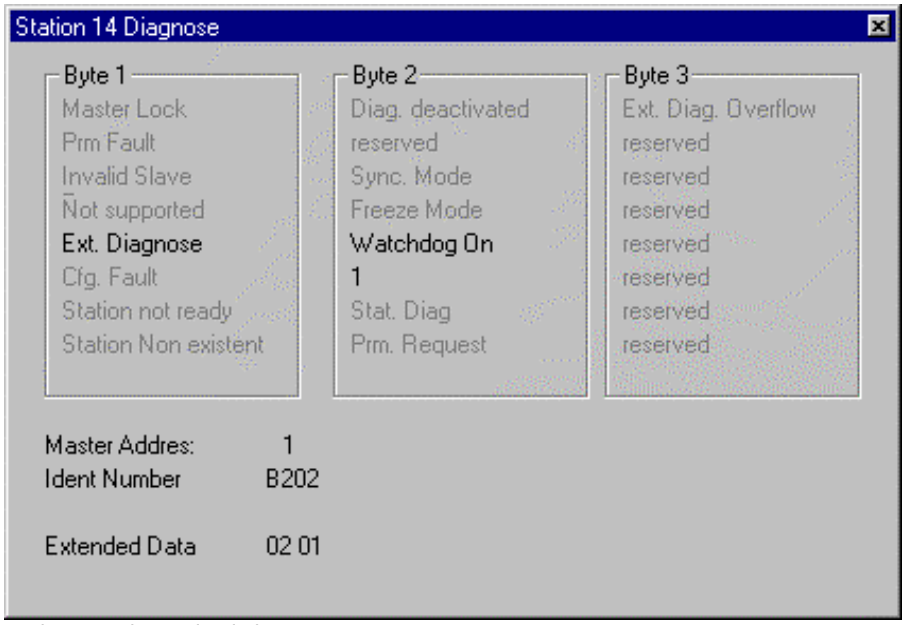
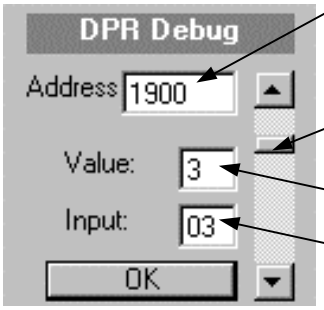


Figure 9: Diagnostic window

3 .8 DPR Debug

The data in the dualport-RAM of the DP-Master interface can be viewed and changed with the debug function.



Dualport-RAM address

Address scroll bar

Red value

New value
The new value will be set by clicking the „OK“ button

Figure 10: DPR Debug

The **PMC-PROFI** I/O-range starts at offset **$0000**.
The **VME-PROFI** I/O-range starts at offset **$1000**.
The **PCI-40, PCP-DP** and **IPE** I/O-range starts at offset **$2000**.

For example:
PCI-40: The first output byte of Slave 14 shall be changed. The output data from Slave 14 are set to offset $900. Select address $2900 ($2000 + $900), enter the new value at field „Input“ and click button „OK“. Now field „Value“ will show the new value and the outputs of Slave 14 will be set.

Note:
DPR debug can access the whole PCI-40/PCP-DP/IPE/VME-PROFI dualport RAM. It’s only allowed to change values within the I/O-range $2000 - $2FFF / $1000 - $1FFF.

3 .9 PROFIBUS Configuration Parameters

Before a PROFIBUS system could be used, the Master (DPM1) must get the information, which DP-Slaves are available for him, and how this Slaves are configurated. Therefore the PROFIBUS-address (0 .. 125), the type of Slave (-> ID-code), the configuration of the Slave (-> Cfg-Bytes), if necessary operating parameters (-> User-bytes) an the I/O-range in the DP-RAM must be told to the Master. In the parameterization and configuration phases of the PROFIBUS, the DP-Master sends this information (ID-code, Cfg-bytes, User-bytes) to the Slaves. Each DP-Slave compares its real configuration with the configuration data received from the DPM1. When verifying the configuration, the device type, format and the length of information as well as the amount of inputs/outputs have to be identical. Only if these tests are successful a DP-Slave is ready to enter the data transfer phase.

To configure a DP-Master system the DIN 19245 / EN50170 defines two type of data records, a Bus-/ Master-record and Slave-records.
DPKONFIG uses this records-format to store the configuration data (-> autoload.dp1 file) and to setup the DP-Master interface. The values of the parameters can be entered/changed in the „DP-Masterclass1 setup"- and the „DP-Slave editor"-windows.
These parameters and the structure of the configuration file are described in the next chapters. For more information about these parameters see also the DIN 19245 / EN50170.

3 .9 .1 Bus-/ Master configuration parameters

FDL_Add — PROFIBUS-address of the Master station
Value: 0 ... 125 (normally 1)

Baud_rate — Baudrate
Range: 0 ... 7

0 -> 9,6 kBit/s	4 -> 500 kBit/s
1 -> 19,2 kBit/s	5 -> 38,4 kBit/s
2 -> 93,7 kBit/s	6 -> 1,5 Mbit/s
3 -> 187,5 kBit/s	7 -> 3 Mbit/s

9600 Bit/s not supported by the DPM1

TSL — Slot-Time T_{SL} [t_{Bit}]
The Slot Time, is the timeout time for the DP-Master. After sending a telegram, the Master will wait for receiving a answer-, acknowledge- or token-telegram until the T_{SL} time is over. Select T_{SL} as followed:

$T_{SL} = maxT_{SDR} + T_Z$ (T_Z = additional value for the signal transmission on the bus)

Note:
Due to a bug of the DPM1, the maximal possible master telegram length T_{TEL} must be added to the T_{SL} time.

T_{TEL} = (9 + <max. max. output length of a Slave>) * 11

max. bytes	T_{TEL} = Additional value for T_{SL}
10	209
32	451
244	2783

Example: Baud rate = 500 kBit/s
$maxT_{SDR}$ = 100 (see max. MaxTsdr_500 from Slaves GSD-files)
T_Z = 50 (Additional value for delay on the bus)
Slave_a : 6 byte input, 4 byte output
Slave_b : 2 byte input, 8 byte output
Slave_c : 16 byte input, 10 byte output
T_{TEL} = 209 (telegram length for 10 byte output)
$T_{SL} = maxT_{SDR} + T_Z + T_{TEL}$
T_{SL} = 100 + 50 + 209

minTSDR

minimal Station Delay $minT_{SDR}$ [t_{Bit}]
Range: 11 .. 255
The value of the master´s $minT_{SDR}$ must = the lowst slaves´s $minT_{SDR}$. Normally the minimum value (=11) should be used).

maxTSDR

Maximal Station Delay, $maxT_{SDR}$ [t_{Bit}]
max T_{SDR} is the max. time used by the Slave before the requested data will be sent to the DP-Master. As $maxT_{SDR}$ the greatest T_{SDR} of the connected Slaves should be set. $maxT_{SDR}$ depends of the used baud rate an can be found in the Slaves GSD-file. $maxT_{SDR}$ must be greater than $minT_{SDR.}$

TQUI

Quit-Time for Modulator/Repeater [t_{Bit}]
Tqui < $minT_{SDR}$
Default value = 0

TSET

Setup-Time [t_{Bit}]
Default value = 1

TTR

Target-Rotation-Time T_{TR} [t_{Bit}]
For Mono-Master-Systems a TTR = the max. cyle time should be used. For example the default value 100 000. For Multi-Master-Systems a different (lower) values may be necessary, see DIN 19245 Part 1/2 for detailed info.

G

GAP-update factor
The DP-Master will start a GAP-Update when the time TTR * G expired.
During an GAP-update cycle, the Master checks all addresses (one address after each communication cycle) within his GAP range, whether a new Master appears, how want to be included into the token ring cycle.
The GAP factor will influence the time, until another Master will be included into the token ring. For Mono-Master-Systems the default value 100 should be used.

HSA

Highest-Station-Adress
Range: 2 .. 125
Mono-Master-System:

Select a low HSA for example 2. The HSA has only a minimal influence for the overall timing of the system.

Multi-Master-System:

Select a HSA = the highest address of another Master station.

As a Ddefault value HSA = 9 can be used for Mono- and Multi-Master-systems.

max_retry_limit — Default value = 1

Min_Slave_Intervall — Min-Slave-Interval [100 µs]

Range: 0 ... 650 (max. 65ms)

Minimal time interval between two Slave scanning cycles. MinS must be ≥ the greatest „Min_Slave_Intervall“ value from the Slaves GSD-files.

Poll_Timeout — Poll-Timeout [1 ms]

Timeuot for Master-Master communications (master1 to master2). Not used by the DPM1 firmware.

Data_Control_Time — Data-Control-Time [10 ms]

The Master will sent his operating mode, at least once per DCT-time, with a multicast-telegram to all Slaves.

Master_Class2_Name — Not used by the DPM1 firmware.

Master_User_Data — Not used by the DPM1 firmware.

Auto_Clear — System error reaction (failure of a DP-Slave off/on)

Default off

3 .9 .2 Slave configuration parameters

Sl-Flag — Slave depending flags

Default value = $80

Slave_Typ — Slave type

Default value = $00 (= DP-Slave)

Mode — Slave Mode (Lock, Unlock, Sync, Frezze, Watchdog)

Default value = $88 (= Watchdog on)

WD-factor — Watchdog_time = 10 ms * WD_Fact_1 * WD_Fact2

In the parameterization phase, the Master send this time to the Slave. The Slave will switch back to it’s Reset-state, if it is not accessed at least once per watchdog interval.

minTSDR — minimal Station Delay min T_{SDR} [t_{Bit}]

Range: 11 .. 255

MinT sets the minimal time interval, the slave must wait, before his

responding telegram are allowed. For Mono-Master-Systems (the VME-PROFI(DPM1) is the only active Master), the minimal value (=11) could be set. For Multi-Master-Systems a higher value could be neseccary.

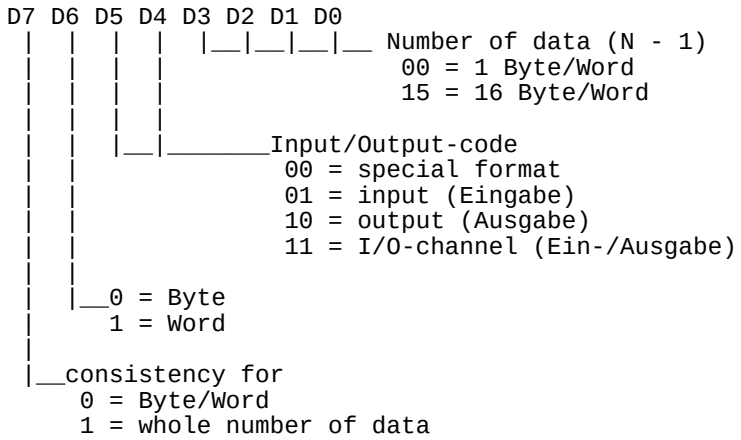
ID — Ident-number of the used Slave-type (see technical data of the slave).

Group — Group-number for broadcast messages, normally $01

User-Parameter — User-Parameter, if necessary (see technical data of the slave).
Max 64 User-bytes (with DPKONFIG) or 8 User-bytes (with the internal Configuration program) can be used. The number of user-bytes and their values must be read from the GSD-file. Normally the default user-bytes are shown in the GSD-file. For some Slaves, different bytes could be necessary (for example to set the slow / high-mode or to set the input-range of analog modules). Please check the manual of the Slaves.

Cfg_Data — I/O configuration as specified in DIN 19245 part 3. See the technical data of the Slave-station or the GSD-file.
Max 64 Cfg-bytes (with DPKONFIG) or 12(32) Cfg-bytes (with the internal Configuration program) can be used. For modular Slaves, the Cfg-byte (or bytes) for each module, corresponding to the installation, are necessary.

Meaning of the Hex-code:



Add_Tab — Address offset of the inputs and outputs within the I/=-range (DPM1io, see DP-Master board manual).

Slave_User_Data — Not used by the DPM1.
DPKONFIG uses this area to store additionally information's like the GSD-filename, the user defined Slave name and the module codes.

3 .9 .3 Structure of the DPKONFIG configuration file:

For the configuration data DPKONFIG uses a file format, similar to the DIN 19245 Bus-/Master- and the Slave-records.

	Section	Length
A	Header	8 byte
B	Headerdata for bus- / master--record	8 byte
C	Bus-/ master-record DIN 19245-3	variable
	D + E for each configurated Slave	
D	Headerdata for slave-record	8 byte
E	Slave-Parametersatz DIN 19245-3 10	variable

Byte	**Parameter name**	**Data type**	**Note**
B: Headerdata fot bus- / Master-record			
0	Function code	$24	
1 - 4	--	Unsigned32	
5 - 6	Add_Offset	Unsigned16	
7	Data_len	Unsigned8	
C: Bus- / master-record (see DIN 19245-3 10.8.1)			
8 - 9	Bus_Para_Len	Unsigned16	
10	FDL_Add	Unsigned8	Master1 address
11	Baud_rate	Unsigned8	
12 - 13	TSL	Unsigned16	
14 - 15	minTSDR	Unsigned16	
16 - 17	maxTSDR	Unsigned16	
18	TQUI	Unsigned8	
19	TSET	Unsigned8	
20 - 23	TTR	Unsigned32	
24	G	Unsigned8	
25	HSA	Unsigned8	
26	max_retry_limit	Unsigned8	
	Bp_Flag	Unsigned8	
28 - 29	Min_Slave_Intervall	Unsigned16	
30 - 31	Poll_Timeout	Unsigned16	
32 - 33	Data_Control_Time	Unsigned16	
34 - 39	Octet 1 - Octet 6 (reserviert)	Octet-String	
40 - 41	Master_User_Data_Len	Unsigned16	= $0034
42 - 73	Master_Class2_Name	Visible-String(32)	= $44 $44 $44 $20 - $20
	Master_User_Data	Octet-String	not used

Byte	Parameter name	Datd type	Note
D: Headerdata for slave-record			
0	Funktion code	$25	
1- 2	--	Unsigned16	
3	Slave_Adresse	Unsigned8	
4	--	Unsigned8	
5 - 6	Add_Offset	Unsigned16	
7	Data_len	Unsigned8	
E: Slave-record (see DIN 19245-3 10.8.2)			
8 - 9	Slave_Para_Len	Unsigned16	incl. length bytes
10	Sl-Flag	Unsigned8	
11	Slave_Typ	Unsigned8	
12 - 23	Octet 1 - Octet 12 (reserviert)	Octet-String	
24 - 25	Prm_Data_Len	Unsigned16	incl. length bytes (=Pln)
26 -	Prm_Data	Octet-String	26 : Mode 27 - 28: WD-Zeit 29 : minTSDR 30 - 31: ID 32 : Group User-Parameter
33+Pln	Cfg_Data_Len	Unsigned16	incl. length bytes (=Cln)
	Cfg_Data	Octet-String	
35+Pln +Cln	Add_Tab_Len	Unsigned	
	Add_Tab	Octet-String	Input-/ output addresses within the DPM1io area of the DP-RAMs 37+Pln+Cln: Input offset 39+Pln+Cln: Output offset
41+Pln +Cln	Slave_User_Data_Len	Unsigned16	incl. length bytes
	Slave_User_Data	Octet-String	43+Pln+Cln:File name[1-4] 47+Pln+Cln:Slave-name 67+Pln+Cln: Module codes

3 .10 DPKONFIG with different hardware configurations

DPKONFIG can operate in the offline mode (no hardware) or with target DP-Master1 hardware connected via the COM1/2 serial interface, via a TCP/IP network or with PC plug-in boards via the dualport RAM interface.

The following chapters give information's how to use and how to setup DPKONFIG for the used target hardware.

3 .10 .1 DPKONFIG with VME-PROFI

The VME-PROFI(DPM1) allows different ways to configure the PROFIBUS.

- Remote configuration through the serial interface with DPKONFIG
  The VME-PROFI(DPM1) can be configured through the serial interface with the program DPKONFIG. To enable this function, the option v = „VMEbus Parametrierung" must be set with the VME-PROFI's Control- and Configuration program (=> default setting). The configuration can easily be generated through reading of the GSD-files. The slaves configuration can use up to 64 Cfg-Bytes and up to 64 User-bytes. The configuration data are stored into the EEPROM of the VME-PROFI (remote format).
  After configuration and test, the PC is no longer necessary. At the next start of the VME-PROFI(DPM1) the configuration data automatically will be read from the EEPROM.
  ==> this is the recommended configuration mode for general usage of the VME-PROFI(DPM1)

- Remote configuration through the VMEbus
  The VME-PROFI(DPM1) can be configured through the VMEbus. To enable this function, the option v = „VMEbus Parametrierung" must be set with the VME-PROFI's Control- and Configuration program (=> default setting). The VMEbus host CPU must run dpstart (or a similar program, see example program on the VMEPROFI disk). This program reads the configuration data from a configuration file („autoload.vp1") and starts-up the PROFIBUS. The configuration file „autoload.dp1" can be generated with DPKONFIG in the offline mode (no hardware).
  ==> this configuration mode is the recommended, if the VMEbus host CPU shall keep track of the configuration of the whole system (incl. PROFIBUS).

  A special case of this VMEbus configuration can be realized by using a network connection from a PC, running DPKONFIG, to the VMEbus host CPU. On the VMEbus Host CPU a TCP/IP-domain program is running which converts the DPKONFIG commands to access the VME-PROFI over the VMEbus. The configuration file, generated from DPKONFIG can be transferred over the network and is stored on the local file system of the Host-CPU. The TCP/IP-domain program is available for OS-9 systems. On request the ANSI-C source code could be delivered.
  ==> this configuration mode is recommended for network based systems. (DPKONFIG running on a PC with network connection to the VMEbus-host CPU)

Note:
The EEPROM data format of the internal configuration program and the remote configuration (with DPKONFIG) are different. This means, this two types of configuration can not be mixed. The configuration data (EEPROM) are cleared if the configuration mode is changed (internal mode <--> remote mode).

3 .10 .1 .1 VME-PROFI - DPKONFIG with RS232 connection

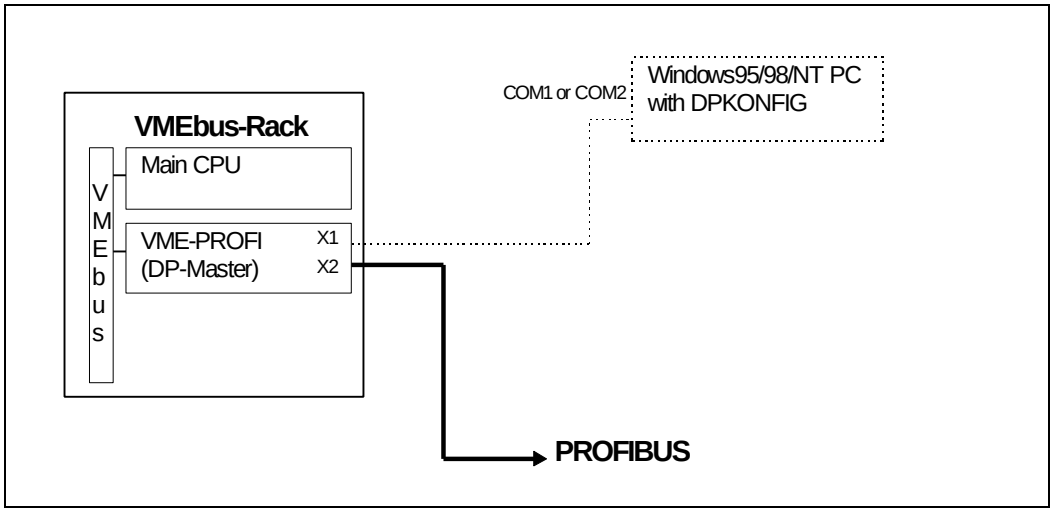


Figure 1: System configuration for DPKONFIG with RS232 connection

The COM1 or COM2 port of the Windows PC are connected with the VME-PROFI's RS232 port X1. To enable the DPKONFIG configuration mode, the option v = „VMEbus Parametrierung" must be set with the Control- and Configuration program (=> default setting).
Now the PROFIBUS can be online configured and tested. The final configuration Data are stored (via the RS232 connection) into the VME-PROFI's internal EEPROM. After configuration and test, the PC is no longer necessary. At the next start of the VME-PROFI(DPM1) the configuration data automatically will be read from the EEPROM.

Note:
Due to allow short cycle times, the DPM1 Vers. 5.x synchronized the data exchange via the RS232 interface with the PROFIBUS cycle. Therefore only one character will be send over the RS232 per PROFIBUS cycle. For the normal operation this would not be noticeable for the user. Only if long cycle times (low baud rate, high Slot-Time, many Slaves) used, the EEPROM data transfer can be very slow. In this case first stop the DPM1 and then transfer the data.

3 .10 .1 .2 VME-PROFI - DPKONFIG with TCP/IP connection

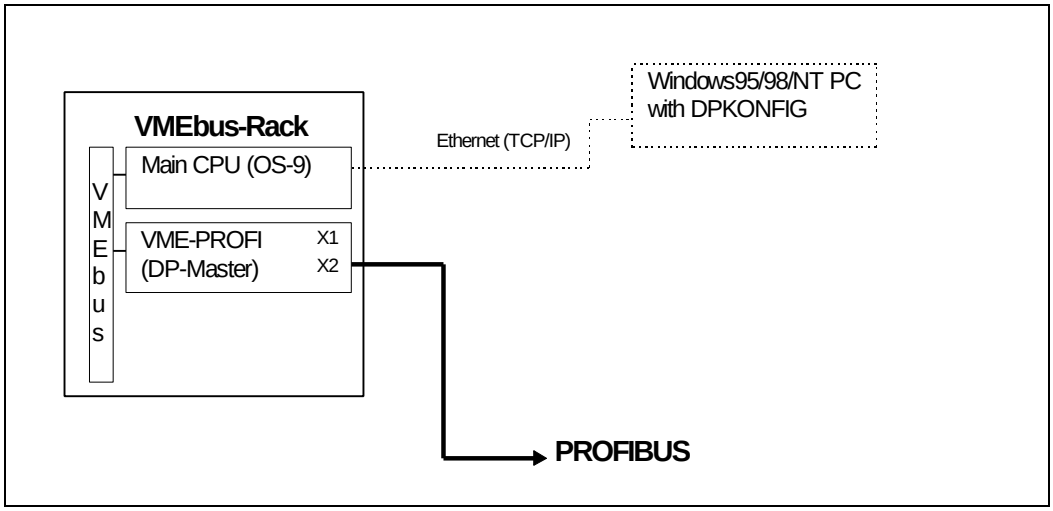


Figure 2: System configuration for DPKONFIG with TCP/IP connection

To enable the DPKONFIG configuration mode, the option v = „VMEbus Parametrierung“ must be set with the VME-PROFI Control- and Configuration program (=> default setting). For the TCP/IP connection the Main CPU within the VMEbus rack needs to run a TCP/IP domain program. This domaim program converts the DPKONFIG commands to access the VME-PROFI via the service interface on the VMEbus. For OS-9 systems a ready compiled program is included in the VMEPROFI disk. The following files are required:

profigo.tcp	start procedurefile
usrtcpd	TCP/IP domain program
usrtcpdc	TCP/IP client program
st_profi	startprogram for the VME-PROFI, simular to the dpstart example program
autoload.dp1	DPKONFIG configuration file

The start procedurefile profigo.tcp should be called within the OS-9 startup procedure.

With DPKONFIG the TCP/IP-connection are selected under the Hardware menu. Now the same online configuration and test function as with the RS232 connection are available.

With this configuration mode the configuration are not stored in the EEPROM, instead the data are stored via TCP/IP in the autoload.dp1 file on the local OS-9 file system. After configuration and test, the PC (and therefore the TCP/IP connection) is no longer necessary. At the next start of the OS-9 system the VME-PROFI(DPM1) will be configured with the data stored in the autoload.dp1 file from the profigo.tcp / st_profi program.

3 .10 .2 PCU - DPKONFIG with RS232 connection

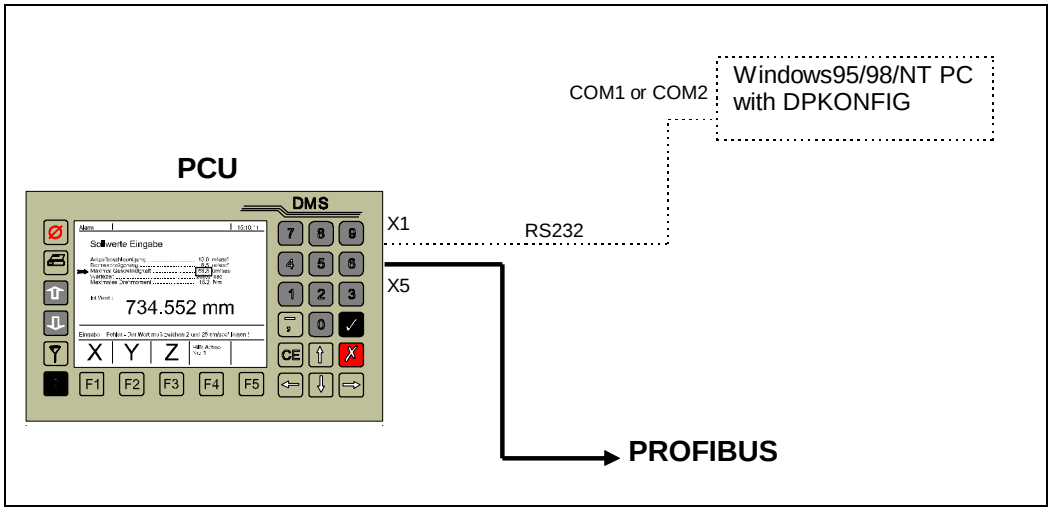


Figure 3: System configuration forPCU - DPKONFIG with RS232 connection

The PCU is a enbedded controller with operator panel and PROFIBUS DP-Master interface, running with the operating system OS-9. To configure the DP-Master interface, a RS232 connection from the PCU to the PC must exist

With this configuration mode the configuration are stored in the onboard Flash-EPROM. After configuration and test, the PC (and therefore the RS232 connection) is no longer necessary. At the next start of the PCU the DP-Master will be configured with the data stored in the Flash-EPROM.

3 .10 .3 IPE - DPKONFIG with TCP/IP connection

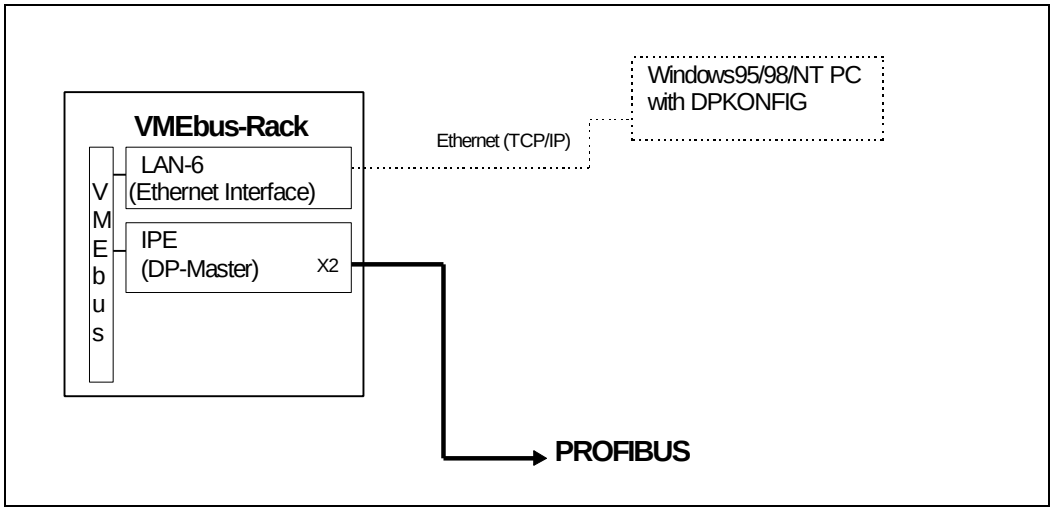


Figure 4: System configuration for IPE - DPKONFIG with TCP/IP connection

The IPE is a stand-alone VMEbus CPU board (MC68040) with onboard PROFIBUS DP-Master interface, running with the operating system OS-9. To configure the DP-Master interface, a TCP/IP connection from the IPE to the PC must exist (Ethernet or serial SLIP connection) and a TCP/IP domain program must run. This domaim program converts the DPKONFIG commands to access the DP-Master via the service interface. The following files required for this connection:

profigo.tcp	start procedurefile
usrtcpd	TCP/IP domain program
usrtcpdc	TCP/IP client program
st_profi	startprogram for the DP-Master
autoload.dp1	DPKONFIG configuration file

The start procedurefile profigo.tcp should be called within the OS-9 startup procedure.

With DPKONFIG the TCP/IP-connection are selected under the Hardware menu.

With this configuration mode the configuration are <u>not</u> stored in the EEPROM, instead the data are stored via TCP/IP in the autoload.dp1 file on the local OS-9 file system. After configuration and test, the PC (and therefore the TCP/IP connection) is no longer necessary. At the next start of the IPE the DP-Mastter will be configured with the data stored in the autoload.dp1 file from the profigo.tcp / st_profi program.

3 .10 .4 PCI-40 / PCP-DP - DPKONFIG with dualport RAM connection

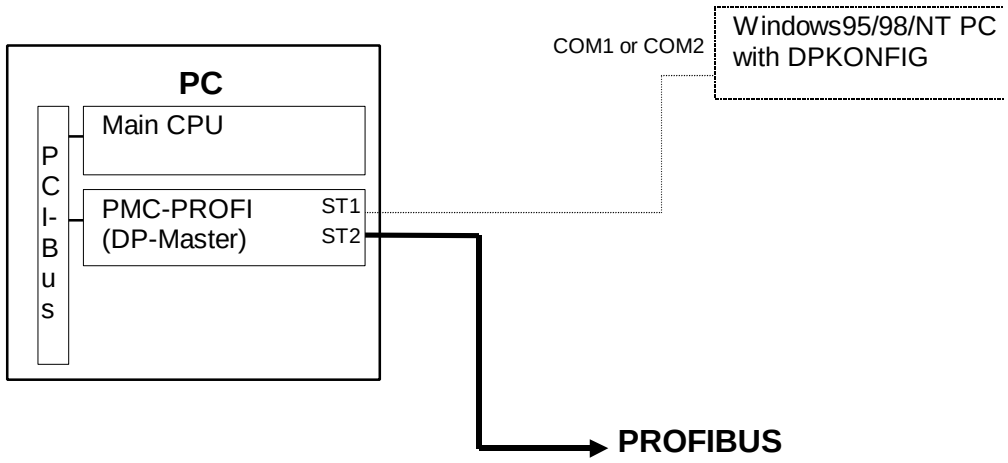
The PCI-40 is a realtime CPU board (MC68040) with onboard PROFIBUS DP-Master interface as a PC plug-in board. The PCP-DP is a partly mounted version of the PCI-40 and includes only of the DP-Master interface. Both boards use a identical dualport-RAM interface for the DP-Master.

DPKONFIG must run on that PC, which includes the PCI-40 / PCP-DP. After selection of the „PCI-40 / PCP-DP"-hardware DPKONFIG communicates via the dualport RAM interface with the board. No special setup or external connection is required.

For this boards the EEPROM-menu is not available. The data must be stored in the file „c:\dms\setup\autoload.dp1". This file must be used by the user program, to setup the PROFIBUS. See PCI-40 / PCP-DP manual for the PROFIBUS startup.

For the PCP-40 / PCP-DP the contents of the DP-RAM can be viewed in the „DP-Slave editor" window.

3 .10 .5 PMC-PROFI - DPKONFIG with RS232 connection



The COM1 or COM2 port of the Windows PC are connected with the PMC-PROFI's RS232 port ST1.
Now the PROFIBUS can be online configured and tested. The final configuration Data are stored (via the RS232 connection) into the PMC-PROFI's internal Flash Eprom. After configuration and test, the PC is no longer necessary. At the next start of the PMC-PROFI(DPM1) the configuration data automatically will be read from the Flash Eprom.

4 Appendix Section

4 .1 List of Abbreviations

ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
DMS	DMS Dorsch Mikrosystem GmbH
DPM1	DP-Master Class 1 (PROFIBUS DP controller for data exchange with DP-Slaves)
DPM2	DP-Master Class 2 (PROFIBUS DP controller for programming, configuration and diagnostics)
DPR	Dualport RAM
DPS	PROFIBUS DP Slave
EEPROM	Electrical Erasable Programmable read-only Memory
ESD	Electrostatic Discharge
GAP	Address range beginning from the own address +1 up to the next active Master
GAP-update	During an GAP-update cycle, the Master checks all addresses (one address after each communication cycle) within his GAP range, whether a new Master appears, how want to be included into the token cycle.
GSD	Geräte-Stammdaten-Datei, device database information.
ID	Identifier
IEC	International Electrotechnical Commission
IEC1131	International standard for PLC programming
IEEE	Institute of Electrical and Electronic Engineers
IPE	Industrial Processing Engine, DMS VMEbus CPU board with PROFIBUS DP-Master interface
MS-DOS	Microsoft Disk Operating System
MULTIPROGwt	IEC1131 Programming system for ProConOS
n.c.	not connected
OS-9	Multi-user and Multitasking Operating System by Microware
PCU	PROFIBUS Control Unit, DMS Operater Panel / Enbedded Controller with DP-Master interface
PLC	Programmable Logic Controler (= SPS)
PNO	PROFIBUS user organization
ProConOS	PROgrammable CONtroller Operating System The IEC1131 software PLC runtime system for DMS CPU boards
PROFIBUS	a standardized field bus, specified in DIN 19245 / pr EN50170
PROFIBUS-DP	PROFIBUS for decentralized peripheral, DIN 19245 Part 1 + 3
PROFIBUS-FMS	PROFIBUS with Fieldbus Message Specification protocol DIN 19245 Part 1 + 2
SPS	Speicherprogrammierbare Steuerung (= PLC)
VITA	VMEbus International Trade Association
VMEbus	Versa Module Eurocard Bus System
VME-PROFI(DPM1)	VME-PROFI with DP-Master Class 1 firmware

4 .2 References

[1] DMS Katalog, Dorsch Mikrosystem GmbH
[2] DIN 19245 Part 1 - 3
[3] M. Poop, The Rapid way to PROFIBUS-DP
PROFIBUS User Organisation

4 .3 Reader Comments

DMS Dorsch Mikrosystem GmbH
Abt.: Qualitätsmanagement
Holmlück 13-15

D-24972 Steinbergkirche
Germany

Sender (please fill in)

Name and Department: _____

_____

Company: _____

Address: _____

Phone/FAX: _____

Did you encounter any errors while reading through this technical manual? If so please fill in this form. Do you have any proposals on how to improve this manual? What subject would you like more information on?

Document ID: _____ Date: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____