Protocol manual

PROFIBUS-DP slave

Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 78
D-65795 Hattersheim
Germany

Tel. +49 (0) 6190 / 9907 - 0
Fax. +49 (0) 6190 / 9907 - 50

Vertieb: +49 (0) 6190 / 9907 - 0
Hotline und Support: +49 (0) 6190 / 9907 - 99

e-mail: hilscher@hilscher.com
Homepage: http://www.hilscher.com

Index	Date	Version	Chapter	Revision
1	13.06.96	0.990	all	Translation of the german manual spelling correction of english translation
	21.06.96	0.991	4.2 5.1.1	Table 'Setup' enlarged with parameter 'Starting mode' Adding states on table 'SPC3'
2	01.10.96	1.000	4.3	Sequence of values of module type revised, hint of max. data length added
			6	Error codes added
3	07.10.96	1.000	5.1.1	INPUT Length and OUTPUT Length set to 241 bytes, reference to hardwaremanual concerning the sum of I/O

Although this protocol implementation has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this protocol implementation for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this protocol implementation or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifiactions at any time in as far as this contributes to technical progress. The version of the manual supplied with the protocol implementation applies.

1 Introduction

This description documents the implementation of the PROFIBUS-DP protocol in slave mode on our equipment. It should be noted that the scope of functions described may be limited for some equipment or applications. For protocol conversions in particular, a reduced scope of functions is as a rule used!

2 PROFIBUS-DP fundamentals

2.1 The requirements

The connection between the decentralized process sequence and the centralised control system established by the communications system takes place on the lowest hierchical level via the field or process bus.
On this level, the primary requirement is that for a simple protocol sequence and short data transfer times in communications. This guarantees the shortest possible system reaction time to dynamic peripheral conditions.

Together with classical I/O data exchange, acyclical transmission of parameter, diagnostic and configuration data must be possible without decisively impeding the real time capabilities of the bus. Only in this way can a good diagnostic concept be fulfilled and operational reliability be ensured.

2.2 Characteristics

The main function of the PROFIBUS-DP is the cyclical transmission of process data from the control system to the peripherals and in the reverse direction. Access is by the master-slave principle. A master controls the assigned slave devices on the bus in polling operation. Data exchange is initiated by a polling message and terminated by an acknowledgment message from the slave addressed. Each slave therefore only becomes active when required by the master. Simultaneous bus access is thus avoided.

The hybrid access method of PROFIBUS permits combined operation of several bus masters and even mixed operation of PROFIBUS-DP and PROFIBUS-FMS within a single bus section. This however depends on the correct configuration of the bus system and unequivocal assignment of the slave devices to the masters.
The PROFIBUS-DP distinguishes between two types of masters. The class 1 master performs cyclical transmission of user data and provides the application data. The class 1 master can be addressed by a class 2 master for certain functions. Direct access to slaves is not permitted. The functions are limited to support services such as reading of diagnostic informatoin from slaves. A class 2 master is therefore also regarded as a programming or diagnostic device.

2.3 Scope of master-slave functions

The scope of functions of a PROFIBUS-DP master of class 1 comprises the automatic cyclical transmission of the input and output data. This requires successful parameterization and configuration of the slaves.
Within the parameterization phase, each slave is parameterized with the slave specific bus parameters and the control times. The configuration phase then follows. In this phase, the master has the slaves compare the stored specified slave configuration data with the actual configuration data. Only when both phases have been successfully completed is a slave included in the transfer of user data. One exception in the master-slave functions is a global function command with which all slaves assigned to a master can be addressed within one bus access. During parameterization, the slaves can be assigned to one of eight possible groups, and therefore the command can also be used relative to groups. This multicast command can be used in PROFIBUS-DP for example to control the synchronization of the outputs and inputs or to reset the module outputs to a secure condition.

2.4 Protection functions

The PROFIBUS-DP is equipped with numerous protection functions. These ensure secure communications particularly in the harsh environment of the decentralized peripherals, not only in fault-free operation, but also with external interference or the failure of individual stations.

- Incorrect parameterization is avoided by excluding stations with incorrect parameters from user data operations.
- The failure of stations is registered by the master and displayed to the user by group diagnostic.
- The failure of the transmission line is detected by the slaves with a timed watchdog function, and leads to shutdown of the outputs.
- EMC interference is almost completely filtered out by the particularly interference free transmission process to RS485 with the differential signal.

Data transfer errors are detected by the CRC test and lead to repetition of the message.

2.5 Commissioning

Before a PROFIBUS-DP system can be put in operation, all the connected stations including the master system must be given unequivocal bus addresses. Only in this way can unequivocal addressing on the bus take place. As an option, the addresses can also be assigned via the bus.

The physical system settings are made using the master parameter set. This contains the master bus address and, for example, the baud rate, the timeout settings and the number of transmission retries. Together with the master parameter set, a slave data set has to be stored for each slave to be activated. Each data set contains the parameterization and configuration data of the slave and the address vectors for the logical storage of the I/O data.

When the parameter sets are available, the master system begins to start up the slaves in succession on instruction by the user or automatically. The first diagnostic cycles indicate which slaves are present on the bus. Only those slaves which have sent a correct feedback message in the diagnostic cycle are then parameterised with the relevant data stored in the master in subsequent parameterization cycles. When this has been performed correctly, configuration cycles follow with the comparison of the specified configuration data in the master and the actual configuration data in the slaves. After the last diagnostic cycle, each slave which has not detected an error in the comparison is ready for operation. Each of these slaves is then automatically included in user data transfer by the master.

For diagnostic, the master maintains a diagnostic buffer for each slave, which can be read by the user. For simplified diagnostic, a group diagnostic field is maintained at the same time, showing bit by bit whether a slave has diagnostic data ready or not.

3 Configuration of the assembly

The standard requires every manufacturer of PROFIBUS-DP slaves and PROFIBUS-DP masters to compile equipment data sheets detailing the characteristics and capabilities of the PROFIBUS-DP devices. In addition, all these data are to be grouped together in a master file for the device and provided to the user.

The objective of this master device file is to facilitate uniform and simple configuration by means of input programs independent of particular manufacturers and based on stipulated structures and codes within the file.

The format of the master device file is clearly stipulated by the PROFIBUS-DP standard. It is maintained as an ASCII file and can therefore by generated or edited with any conventional text editor.

4 Configuration

This chapter provides a general description of the PROFIBUS DP slave configuration of our devices. Various configuration parameters can be adapted for specific projects or specific devices. It is therefore possible for certain parameters not to be available on your device.

The data of the configuration is kept in a data base. This data base can be edited with the project planning and diagnostic program ComPro. Please find in the manual 'Project planning and diagnostic program ComPro' more information concerning the working method of the program ComPro.

The following tables describe the PROFIBUS-DP slave configuration data which can be edited by the user.

4.1 Table PROJECT

General information can be stored in the PROJECT table as project documentation. Moreover this table contains informations - which is not mentioned here - that you will need when you contact our Hotline.

Parameter	Data type	Contents
Project	40 characters	Entry of ASCII text as required
Date	8 characters	
Project engineer	40 characters	

Table PROJECT

4.2 Table SETUP

In this table the parameters for general settings of the PROFIBUS-DP slave will be configured.

Parameter	Data type	Contents	
Bus address	numerical	0..**2**..126	Busaddress of the slave
Starting mode	toggle		Startbehaviour after power on / coldstart:
		standard	Starting with flashparameter
		configuration by application	Waiting for initialisation of the application

Table SETUP

- **Bus address**

Parameter to configurate the own bus address. It's not allowed to choose a bus address more than once in a PROFIBUS system!

- **Starting mode**

This parameter defines the starting behaviour of the device after power on / coldstart. The value 'standard' initializes the device with the parameters of the data base, stored in the flash-eprom. This database can be edit and downloaded by ComPro. Choosing the value 'configuration by application' means, the device will wait after power on / coldstart for the initializing of the application. The initialization parameters will be given from the application via dual-ported memory.

4.3 Table MODULES

Most PROFIBUS-DP slaves only have a few input and/or output bytes. The PROFIBUS-DP presents length up to 246 bytes. Our PROFIBUS-DP slave offers a flexible, modular composition. That means that parts of the input and output-areas can be seen as single modules.

The master - for example a PLC - can put the different modules from our PROFIBUS-DP slave to different locations in its I/O area. So it is possible that the PROFIBUS-DP slave simmulates some I/O modules, that earlier have been hardware components of the PLC.

Parameter	Data type	Contents	
Modulet ype	toggle	in byte in word out byte out word **in byte con** in word con **out byte con** out word con blank space	Input byte without consistence Input word without consistence Output byte without consistence Output word without consistence Input byte with consistence Input word with consistence Output byte with consistence Output word with consistence blank space
Modulel ength	toggle	1 2 3 **4** 8 12 16 20 32 64	

Table MODULES

Please notice, that the sum of I- and O-data should not exceed the maximum possible size of I/O - data supported of your hardware. Please find further details in the chapter 'technical data' of your hardware manual.

A module is defined by a module type and the module length.

- **Module type**

The modul type describes, if it is a blank space, an input module or output module of the type byte or word. A blank space can be used to reserve space for later expansions.

On byte and word modules the consistence of the whole module data can be configurated. Consistence means, that the data though the whole module belongs together and can only be evaluated together. So it is not possible to transfer only one part of the module data. In the **GSD file** of the PROFIBUS-DP slave, that tells the configutation tool for the master, which possibilities the slave has got, **only the modules with consistence** are placed. You should use only modules with consistence!

During tests we noticed, that some PROFIBUS-DP masters have problems with long consistent modules. So we gave the possibility to configure some modules as non consistent modules. The PROFIBUS-DP slave will handle the data consistent, but it is possible, that the master transfers the data to its application inconsistent.

- **Module length**

This parameter describes the length of the module in bytes or words (see module type). Only the numbers given in the table can be configurated.

Please notice, that module type and sequence of the modules, configured at the slave, has to be identical with the module type and sequence of the modules, configured at the master! Use only modules with consistence if possible!

4.4 Table DPM-CTRL

The settings of the handshake mode between the PROFIBUS-DP slave and the user program via dual-port memory have to be configurated in this table.

Parameter	Data type	Contents
Handshake of the process-data	toggle	bussychron/device controlled buffered/device controlled no consistence/uncontrolled buffered/host controlled bussynchron/host controlled

Table DPM-CTRL

Notice: Use only the mode ordered by the user program. Otherwise a faultless communication between the user program and the PROFIBUS-DP slave could not be ensured.

4.5 The table GLOBAL

General, protocol independent settings are effected in the GLOBAL table.

Parameter	Data type	Contents
Application available	toggle	yes, no

Table GLOBAL

- **Application available**

This parameter is only of significance in conjunction with a CIF board or COM board. It determines the status of the dual-port memory driver after a reset or warm start.

In the dual-port memory of the CIF there is a bit which signals to the CIF that the application on the PC is not available at the moment. The PROFIBUS-DP slave will stay in a diagnostic-mode for this time.

If the application is ready for communications, it has to signal this to the CIF via the bit in the dual-port memory.

5 Diagnostic functions

To identify and solve problems quickly during putting the PROFIBUS system into operation, the PROFIBUS-DP slave offers informations about actual states, stored in tables that could be shown by the program ComPro by choosing the menu *'online.task.status read'* and one of the following items.

5.1 SPC3CTRL taskstates

This menu offers all tables containing the states of this task, that has direct contact to the SPC3.

5.1.1 State table 'SPC3'

Parameter	Values	Description
Baud rate (kBaud)		Last detected baud rate.
	12000	12 MBaud
	6000	6 MBaud
	3000	3MBaud
	1500	1,5 MBaud
	500	500 kBaud
	187	187,5 kBaud
	93	93,75 kBaud
	9	9 600 Baud
	0	Not detected
Busaddress	1 ... 126	Actual configured bus address.
Ident Number	7 5 0 4	CIF 30-DPS, CIF 104-DPS
	7 5 0 3	PKV 30-DPS
	7 5 0 1	COM-DPS
Task state	xxx1	Task is during initialization. If this state stays for some seconds, the configuration parameters may be invalid.
	xx1x	Task running The initialization happened without error, Generally the task is able to run communication on the bus.
	x1xx	Diagnostic Slave-diagnostic telegrams will be sent at the moment on the bus. Reasons could be, if the user program or the DP master orders this.
	1xxx	Data exchange The data exchange mode is active. The user-data will be transfered on the bus between the master and the slave actually..
INPUT Length	0 - 241	Length of the inputs in bytes *
OUTPUT Length	0 - 241	Length of the outputs in bytes *
Error Cnt	0 - 65535	Number of errors
Last Error	0 - 255	Last error that happened

State table 'State SPC3'

**** Depending on intenal resources of the SPC3 the maximum Length of the sum of inputs and outputs is restricted. Please note the limitations in your hardware-manual.***

Baud rate
The baud rate will be detected automatically by the SPC3. If no valid telegrams have been received, the slave checks all other baud rates until valid telegrams are received. Up this time the last valid baud rate will be displayed.

Ident Number
This field shows the from PROFIBUS Trademark Organisation (PTO) given identification code of the slave.

Task state
The parameter description of 'task state' stands an 'x' for 'don't care'. If the task state shows the value 'task during initialization' during some seconds, the configuration has to be checked. A detailed error description can be seen in the trace by choosing the menu 'diagnostic trace'.

After succesful initialization the state 'task running' will be reached. The task is ready for communication. This task state stays active during the task runns faultless.

If bus communication is running, the states 'diagnostic' and 'data exchange' give more information about the kind of communication. Reasons to run diagnostic can be the one of the following:

- the user program demands diagnostic (please refer to the description of the user program)
- the DP slave demands diagnostic on itself, if internal errors happen
- the DP master can't start with data exchange because of the slave behaviour (refer to the following chapters)
- the DP master demands diagnostic, because its user program (or CPU of the PLC) or internal state does not allow to change to data exchange (please refer to the master documentation)

By reaching the state 'data exchange', the process-data will be updated between DP master and DP slave - the data transfer is running.

INPUT Length, OUTPUT Length
The calculated length of the input-data will be displayed in the field INPUT Length, those of the output-data in the field OUTPUT Length. Please note, that depending on the resources of the SPC 3, the sum of both is restricted. The limitations can be found in the chapter 'technical data' in the hardware-manual.

Error Cnt, Last Error
Error Cnt shows, how many errors happened since the last cold-/warmstart. Last Error gives more information about the last error that happened.

5.1.2 State table 'Slave Config'

This table contains the actual calculated configutation bytes, which have to be identical in value and sequence with the at the master configurated configutation bytes. The conposition of these bytes is explained in the PROFIBUS-norm EN 50170.

Parameter	Values	Description
Config Length	0 ... 48	Number of valid config bytes.
Config Byte 1	0x00..0xFF	Value of first config byte.
Config Byte 2	0x00..0xFF	Value of second config byte.
Config Byte ...	...	Value of ... config byte.
Config Byte 48	0x00..0xFF	Value of 48th config byte.

State table 'Slave Config'

The parameter 'config length' tells how many config bytes are valid and could be evaluated. If the parameter shows for example the value 6, the config bytes 1 to 6 are valid.

5.1.3 State table 'Master Config'

Some master send during initializing the string of the configuration bytes to the slave. This string can be seen in this table. If the slave does not leave the diagnostic mode, please compare the contents of this table with the contents of the 'slave config' table. They should be identical!

Parameter	Values	Description
Config Length	0 ... 48	Number of valid config bytes.
Config Byte 1	0x00..0xFF	Value of first config byte.
Config Byte 2	0x00..0xFF	Value of seccond config byte.
Config Byte ...	...	Value of ... config byte.
Config Byte 48	0x00..0xFF	Value of 48th config byte.

State table 'Master Config'

The parameter 'config length' tells how many config bytes are valid and could be evaluated. If the parameter shows for example the value 6, the config bytes 1 to 6 are valid.

6 Error codes

The following error description should help to solve problems quickly. Initialization errors can be seen in the trace (menu in ComPro: *diagnostic.trace*).

Error number	Description
52	Invalid bus address configured Valid addresses are between 1 and 126
53	Waiting for warm start The parameter 'starting mode' in the table 'setup' is configured to the value 'configuration by application' but the application did not make a warmstart to the device at all.
54	Invalid Module type configured The configured code for the module type is not defined. If this Error happens after a warmstart check also the buffer_len and contact the developer of the user program.
55	Invalid Module length The code for the parameter module length is not defined.
61	No address-switches available The configuration tells the firmware to select the busaddress from address-switches, but the hardware has no address-switches.
70	I/O-data too long The maximum size of I/O-data has been exceeded. Please check the length of all modules.
71	SPC3 initialization error The SPC3 return an error during initialization. Please contact your distributor.
210	Data base access error The access to the data base failed. Please check if data base is loaded correctly.

Initialization errors of the task SPC3CTRL

Runtime errors can happen after initializing. The SPC3CTRL task communicates to the application only per diagnostic trough the dual-port memory. Message communication to the SPC3CTRL is not possible.

Error value	Description
115	Diagnostic length too long The Length of the diagnostic data is too long.
116	No diagnostic buffer There is no free diagnostic buffer available at the moment. *This Error will be temporary.*
117	Invalid Message received The task received an invalid Message from any application. Please contact the developer of the user program.

Run time errors

7 Example applications

One part of the systempackage are example applications. By using them it is possible to transfer data through the bus.

These applications are localised in the path UTIL. One is created for DOS and one for Windows 3.xx. The application for Windows 3.xx is also available for Windows 95 and for Windows NT. If you need these, please contact our HOTLINE.

7.1 Example application 'IOView' for DOS

For the online viewing and changing of the process data an example program can be loaded of the system-package. Under MS-DOS or compatible MS-DOS-window or under Windows the DOS example application 'IOVIEW.EXE' can be started with the startaddress of the device as delivery parameter like '/A:CA00'. Example:

```
IOVIEW   /A:CA00                in case the addresse 0xCA00 (0) is jumperd
```

Be sure that the device works in 'standard, uncontrolled' operating mode to establish the communication between the program and the device, because the program supports no sychronization. After its start the program checks, if the communication with the device is possible. When it recognizes the card, the card will be initialized. Afterwards the send and receive process data contents are displayed in word size.

With the 'cursor up' or 'cursor down' key it is possible to change between the 256 different send process datawords and with the 'cursor left' and 'cursor right' key the receive process datawords can be selected. With the 'i' or the 'd' key the contents of the shown send process data can be increased or decreased.

In addition to the process data the three bases states of the card are displayed. 'RDY' reports that the device has finished its initialisation without an error. 'RUN' is reported, if non of the implemented software task has recognized an error. The condition 'COM' is activated, if the communication state 'data_exchange' has been reached.

The <*ESC*> button quit the the program.

7.2 Example application 'IOViewWi' for Windows 3.xx

Open the file manager and change to the path, where the system package is installed. Change to UTIL.

Double-click the file 'IOVIEWWI.EXE'.

After starting up the program expects a device at the address 0xCA00 (0). If your device is jumperd on another address, choose the menu 'Device.Deviceaddress' and change to your address. If there is DPM-access, the firmware-name and -version will be displayed in the area 'Firmware'.

The 'Device-State' shows, if the device is actually in the state 'Data-Exchange'. If you want to leave the Mode 'Data-Exchange' toggle the check-box 'Userprogram Online'.

In the 'SEND-Data'-box the user-data is expected, that will leave the device. The box 'RECEIVE-Data' displays the user-data, that is received from the device. It is only possible to display/edit blocks up to 50 bytes length (Length-boxes). Those blocks can start from an offset-address (Offset, given in bytes).

Errors will be displayed in information-boxes. A list of error-numbers with an explaination can be displayed with the menu 'Help.Errorlist'.