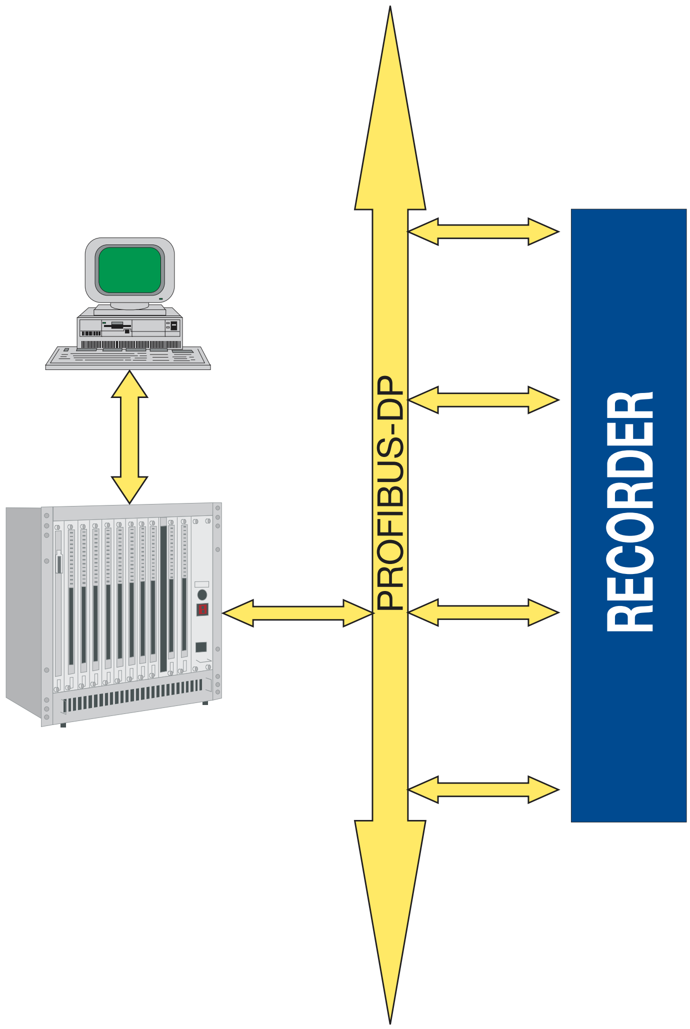



# PROFIBUS-DP

B 95.5010.2.3

Interface Description

10.01/00395223

# Contents

# Contents

## 1.1 Preface

Please read this manual before starting up the interface. Keep the manual in a place which is accessible to all users at all times.

Please assist us to improve this manual.

Your suggestions will be welcome.

If any difficulties should arise during commissioning, you are asked not to carry out any unauthorized manipulations. You could endanger your rights under the instrument warranty!

Please contact the nearest subsidiary or the main factory in such a case.

When returning chassis, modules or components, the regulations of EN 100 015 “Protection of electrostatically sensitive components” must be observed. Use only the appropriate **ESD** packaging for transport.

Please note that we cannot accept any liability for damage caused by ESD (electrostatic discharge).

## 1.2 Typographical conventions

### 1.2.1 Warnings

The signs for **Danger** and **Caution** are used in this manual under the following conditions:

**Danger**

This symbol is used where there may be **danger to personnel** if the instructions are disregarded or not followed accurately!

**Caution**

This symbol is used where there may be **damage to equipment or data** if the instructions are disregarded or not followed accurately!

**Caution**

This symbol is used if precautions must be taken when handling **electrostatically sensitive components.**

### 1.2.2 Note signs

**Note**

This symbol is used to draw your **special attention** to a remark.

**Reference**

This symbol refers to **additional information** in other manuals, chapters or sections.

abc[1]

**Footnote**

Footnotes are notes which **refer to certain points** in the text.
Footnotes consist of two parts:

Marking in the text and the footnote text.

The marking in the text is arranged as continuous superscript numbers.

**Handling instructions**

*

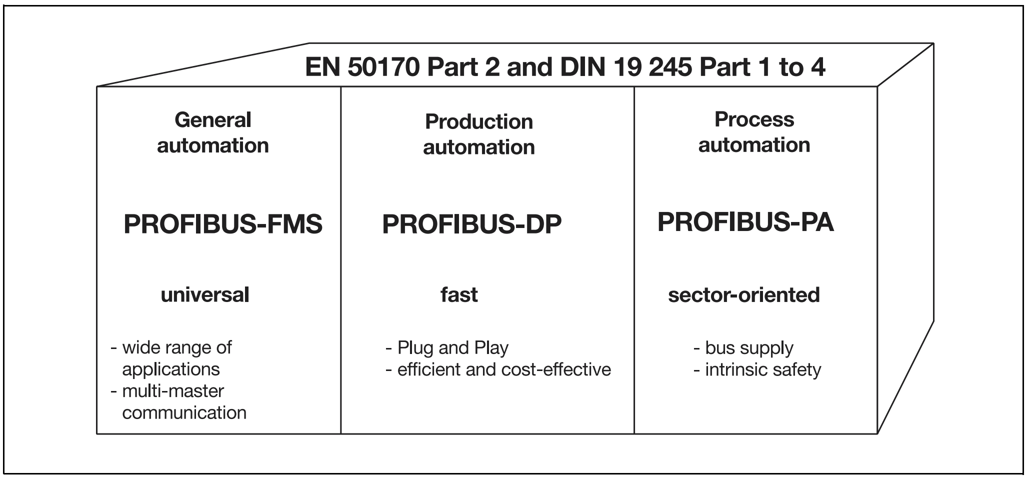
This symbol marks the description of a **required action**.

The individual steps are indicated by an asterisk, e. g.

* Press the [▲] key
* Confirm with [ENTER]

PROFIBUS is a manufacturer-independent, open fieldbus standard for a wide range of applications in manufacturing, process and building automation. Manufacturer independence and openness are ensured by the international standard EN 50 170.
Using PROFIBUS, devices from different manufacturers can communicate without any special interface adjustments. PROFIBUS can be employed for both high-speed time-critical data transmission and extensive, complex communication tasks. The PROFIBUS family consists of three versions.

## 2.1 Profibus types




**The PROFIBUS family**

**PROFIBUS-DP**

This PROFIBUS version, which is optimized for high speed and low connection costs, has been especially designed for communication between automation control systems (PLC) and distributed field devices (typical access time: < 10 msec). PROFIBUS-DP can be used to replace conventional, parallel signal transmission with 24 V or 0/4—20 mA.

DPV0: cyclic data transfer:
--> is supported by the recorder.

DPV1: cyclic and acyclic data transfer:
--> is not supported by the recorder.

DPV2: slave-to-slave communication takes place in addition to cyclic and acyclic data transfer:
--> is not supported by the recorder.

**PROFIBUS-PA**

PROFIBUS-PA has been specifically designed for process engineering. It permits the linking of sensors and actuators to a common bus cable, even in hazardous areas. PROFIBUS-PA enables the data communication and energy supply for devices in two-wire technology according to the international IEC 1158-2 standard.

**PROFIBUS-FMS**

This is the universal solution for communication tasks at cell level (typical access time: approx. 100 msec). The powerful FMS services open up a wide range of applications and provide a high degree of flexibility. FMS is also suitable for extensive communication tasks.

## 2.2 RS485 transmission technology

Transmission takes place according to the RS485 standard. It covers all areas in which a high transmission speed and simple, cost-effective installation are required. A shielded twisted copper cable with one conductor pair is used.

The bus structure permits addition and removal of stations or step-by-step commissioning of the system without affecting the other stations. Later expansions have no influence on the stations which are already in operation.

Transmission speeds between 9.6 kbit/sec and 12 Mbit/sec are available. One uniform transmission speed is selected for all devices on the bus when the system is commissioned.

| | |
|---|---|
| Network topology | linear bus, active bus termination at both ends, stub cables are only permissible for baud rates ≤1.5 Mbit/sec. |
| Medium | shielded twisted-pair cable |
| Number of stations | 32 stations in each segment without repeater (line amplifier). With repeaters, this can be expanded to 126. |
| Connector | preferably 9-pin sub-D connector |

**Basic features of the RS485 transmission technology**

**Installation tips**

All devices are connected in a bus structure (line). Up to 32 stations (master or slaves) can be linked up in one segment.
The bus is terminated by an active bus terminator at the start and end of each segment. Both bus terminators must always be powered, to ensure fault-free operation.
If there are more than 32 users, repeaters must be used to link up the individual bus segments.

**Cable length**

The maximum cable length depends on the transmission speed. The cable length specified can be extended by using repeaters. It is recommended not to connect more than 3 repeaters in series.

| Baud rate (kbit/s) | 9.6 | 19.2 | 93.75 | 187.5 | 500 | 1500 | 12000 |
|---|---|---|---|---|---|---|---|
| Range/segment | 1200 m | 1200 m | 1200 m | 1000 m | 400 m | 200 m | 100 m |

**Range based on transmission speed**

**Cable data**

These cable length specifications refer to the cable type described below:

Characteristic impedance: 135 — 165 Ω
Capacitance per unit length: < 30 pf/m
Loop resistance: 110 Ω/km
Core dia.: 0.64 mm
Core cross-section: > 0.34 $mm^2$

It is preferable to use a 9-pin sub-D connector for PROFIBUS networks incorporating RS485 transmission technology. The pin assignment at the connector and the wiring are shown at the end of this chapter.

PROFIBUS cables and connectors are supplied by several manufacturers. Please refer to the PROFIBUS product catalog (www.profibus.com) for types and addresses of suppliers.

When connecting up the devices, make sure that the data lines are not reversed. It is absolutely essential to use shielded data lines. The braided shield and the screen foil underneath (if present) should be connected to the protective earth on both sides, and with good conductivity. Furthermore, the data lines should be routed separately from all high-voltage cables, as far as this is possible.

As a suitable cable we recommend the following type from Siemens:

**Simatic Net Profibus 6XV1**
**Order No. 830-0AH10**
*** (UL) CMX 75 °C (Shielded) AWG 22 ***

# 2 Profibus description

**Data rate**

For installation, the use of stub cables must be avoided for data rates above 1.5 Mbit/sec.

☞ For important tips on installation, please refer to the Installation Guidelines PROFIBUS-DP, Order No. 2.111 by the PNO (Profibus User Organization).

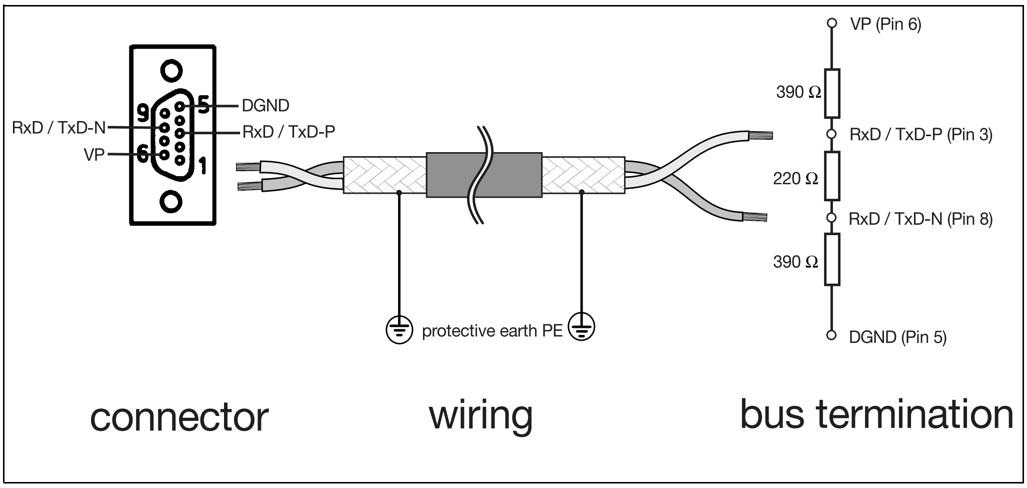
Address:
Profibus Nutzerorganisation e.V.
Haid- u. Neu-Straße 7
D-76131 Karlsruhe, Germany

Internet: www.profibus.com

**Recommendation:**

**Please follow the installation recommendations made by the PNO, especially for the simultaneous use of frequency inverters.**

**Wiring and bus termination**



## 2.3 PROFIBUS-DP

PROFIBUS-DP is designed for high-speed data exchange at the field level. The central control devices, PLC/PC for instance, communicate through a fast serial connection with distributed field devices such as I/O, paperless recorders and controllers. Data exchange with these distributed devices is mainly cyclic. The communication functions required for this are defined by the basic PROFIBUS-DP functions in accordance with EN 50 170.

**Basic functions**

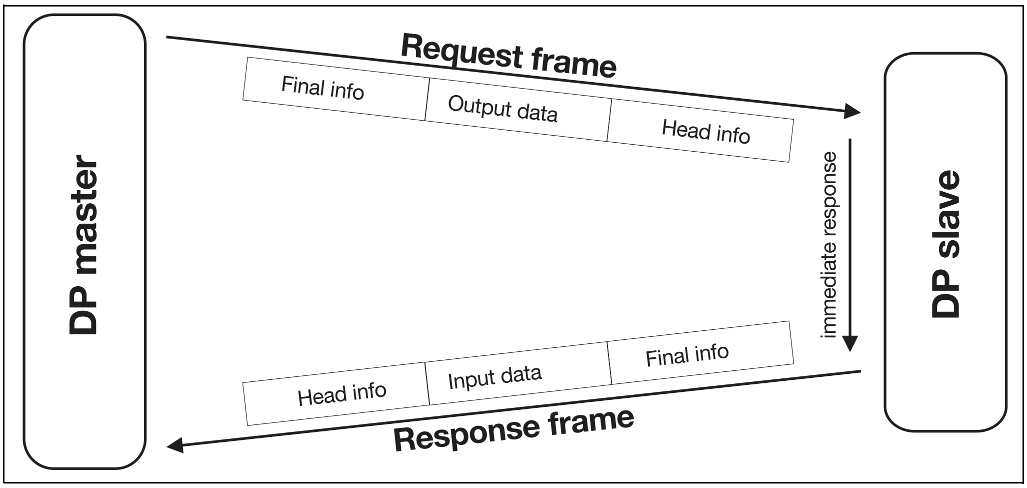
The central controller (master) reads the input information cyclically from the slaves and writes the output information cyclically to the slaves. The bus cycle time must be shorter than the program cycle time of the central PLC. In addition to cyclic user data transmission, PROFIBUS-DP provides powerful functions for diagnostics and commissioning.

| |
|---|
| **Transmission technology:**<br>• RS485 twisted pair<br>• Baud rates from 9.6 kbit/sec up to 12 Mbit/sec |
| **Bus access:**<br>• Master and slave devices, max. 126 users on one bus |
| **Communication:**<br>• Peer-to-peer (user data communication)<br>• Cyclic master-slave user data communication |
| **Operating states:**<br>• Operate: Cyclic transmission of input and output data<br>• Clear: Inputs are read, outputs remain in secure state<br>• Stop: Only master-master data transfer is possible |
| **Synchronization:**<br>• Sync mode: not supported by the recorder<br>• Freeze mode: not supported by the recorder |
| **Functionality:**<br>• Cyclic user data transfer between DP master and DP slave(s)<br>• Dynamic activation or deactivation of individual DP slaves<br>• Checking the configuration of the DP slaves<br>• Address assignment for the DP slaves via the bus<br>• Configuration of the DP master via the bus<br>• maximum of 246 bytes input/output data for each DP slave |
| **Protective functions:**<br>• Address monitoring for the DP slaves<br>• Access protection for inputs/outputs of the DP slaves<br>• Monitoring of user data communication with adjustable monitoring timer in the master |
| **Device types:**<br>• DP master Class 2, e. g. programming/project design devices<br>• DP master Class 1, e. g. central automation devices such as PLC, PC…<br>• DP slave e. g. devices with binary or analog inputs/outputs, controllers, recorders... |

**Cyclic data transmission**

The data transmission between the master and the DP slaves is carried out by the master in a defined, recurring order. When configuring the bus system, the user defines the assignment of a DP slave to the master. It is also defined which DP slaves are to be included in, or excluded from, the cyclic user data communication.

Data transmission between the master and the DP slaves is divided into three phases: parameterization, configuration and data transfer. Before a DP slave enters the data transfer phase, the master checks in the parameterization and configuration phase whether the planned configuration matches the actual device configuration. In the course of this check, the device type, format and length information, as well as the number of inputs and outputs must agree. These checks provide the user with reliable protection against parameterization errors. In addition to the user data transfer, which is performed automatically by the master, new parameterization data can be sent to the DP slaves at the request of the user.




**User data transmission in PROFIBUS-DP**

## 3.1 GSD files

Device base data (GSD) enable open project design.

PROFIBUS devices have different features. They differ with respect to the available functionality (e. g. number of I/O signals, diagnostic messages) or possible bus parameters, such as baud rate and time monitoring. These parameters vary individually for each device type and manufacturer. In order to obtain simple Plug & Play configuration for PROFIBUS, the characteristic device features are defined in an electronic device data sheet (Device Data Base File, GSD file). The standardized GSD files expand open communication up to the operator level. By means of the project design tool, which is based on the GSD files, devices from different manufacturers can be integrated into a bus system, simply and user-friendly. The GSD files provide a clear and comprehensive description of a device type in a precisely defined format. GSD files are prepared according to the application. The defined data format permits the project design system to simply read in the GSD files of any PROFIBUS-DP device and automatically use this information when configuring the bus system. Already during the project design phase, the project design system can automatically perform checks for input errors and the consistency of data entered in relation to the entire system.

The GSD files are divided into three sections.

- **General specifications**
  This section contains information on manufacturer and device names, hardware and software release states, baud rates supported and the possible time intervals for monitoring times.
- **DP master-related specifications**
  This section contains all the parameters related to DP-master devices only, such as the maximum number of DP slaves that can be connected, or upload and download options. This section is not available for slave devices.
- **DP-slave related specifications**
  This section contains all slave-related specifications, such as the number and type of the I/O channels, specification of diagnostic texts and information on the consistency of I/O data.

The GSD format is designed for flexibility. It contains lists, such as the baud rates supported by the device, as well as the possibility of describing the modules available in a modular device.

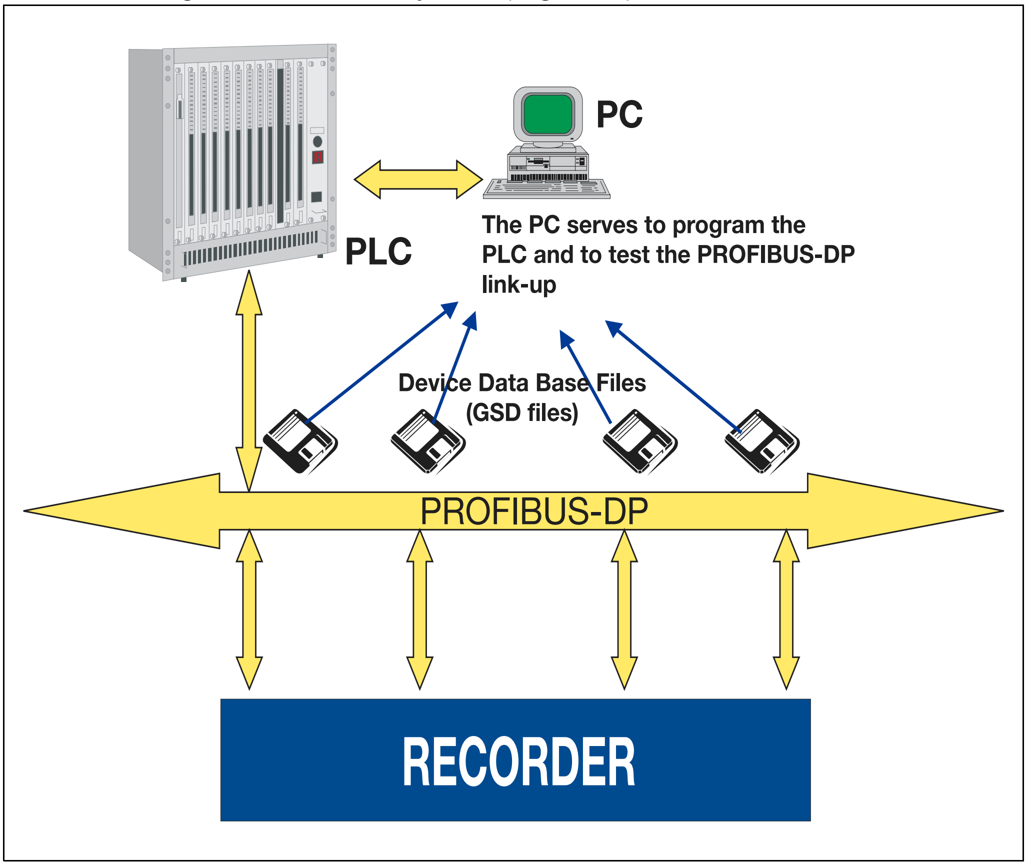
## 3.2 Configuration procedure

**Plug & Play**

To simplify the configuration of the PROFIBUS system, the master (PLC) is configured using the PROFIBUS configurator and the GSD files, or in the PLC through the hardware configurator.

**Configuration steps:**

- Create GSD file by using the GSD generator
- Load GSD files of the PROFIBUS slaves into the PROFIBUS network configuration software
- Perform configuration
- Load configuration into the system (e.g. PLC)




**The GSD file**

The characteristic device features of a PROFIBUS slave are specified by the manufacturer, clearly and comprehensively in a precisely defined format, in the GSD file (Device Data Base File).

**The PROFIBUS configurator / hardware configurator (PLC)**

This software can read in the GSD files for PROFIBUS-DP devices of any manufacturer and integrate them for the configuration of the bus system.

Already in the project design phase, the PROFIBUS configurator automatically checks the files that have been entered for errors in system consistency.

The result of the configuration is read into the master (PLC).

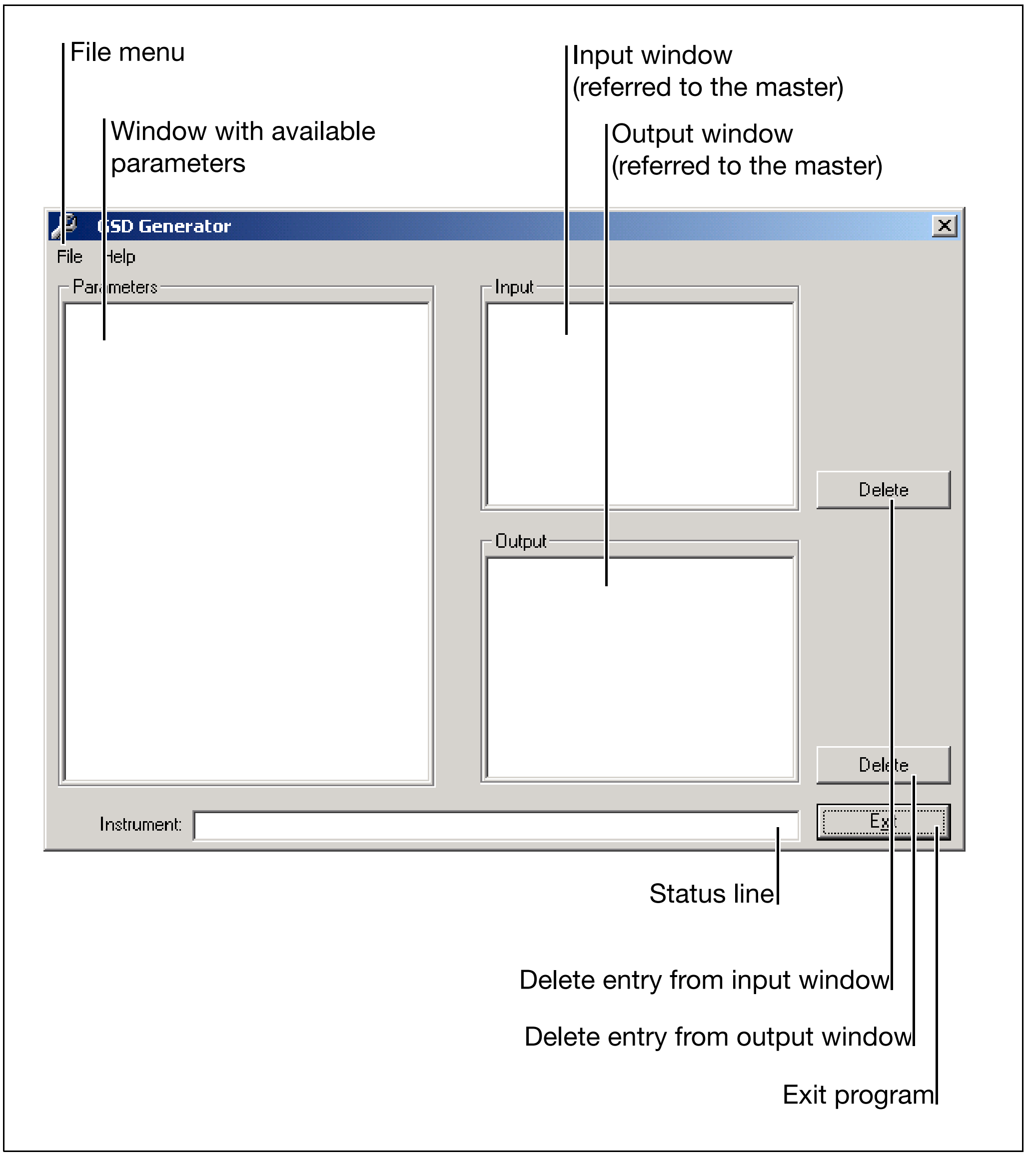
## 3.3 The GSD generator

### 3.3.1 General

GSD files for recorders with a PROFIBUS interface are generated by the user with the aid of the GSD generator.
The recorders with a PROFIBUS interface can send or receive a large variety of variables (parameters). Since, however, in most applications, only a portion of these variables will be sent via PROFIBUS, the GSD generator makes a selection of these variables.

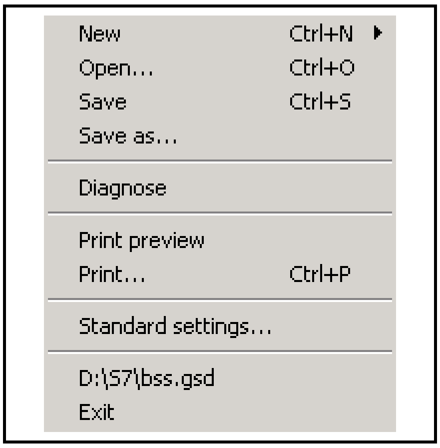
After selection of the device, all available variables are shown in the “Parameters“ window. Only after these have been copied to the “Input” or “Output” window will they later be contained in the GSD file for processing or preprocessing by the master (PLC).

### 3.3.2 Operation



**File menu**

The file menu can be called up using the Alt-D combination or the left mouse button. It provides the following options:




| | |
|---|---|
| **New** | After calling up the function which creates a new GSD file, the available devices are selected. After selection of the required device, all available parameters are shown in the parameter window. |
| **Open** | This function opens an existing GSD file. |
| ***Save/ Save as*** | This function is available for saving the generated or altered GSD file. |
| ***Diagnosis*** | Using this function, the GSD file can be tested in conjunction with a Profibus-DP adapter from B+W. |
| ***Print preview*** | shows the preview of a report[1] that can be printed. |
| **Print** | prints a report[1]. |
| **Standard settings** | The language to be used at the next restart of the program can be selected here. |
| **Exit** | exits the program. |

1. The report contains additions information for the PLC programmer (e.g. data type of the selected parameters).

⇨ Chapter 3.3.3 “Example report”

### 3.3.3 Example report

I/O report

Device: RECORDER

Length of inputs (byte): 7
Length of outputs (byte): 4

Inputs

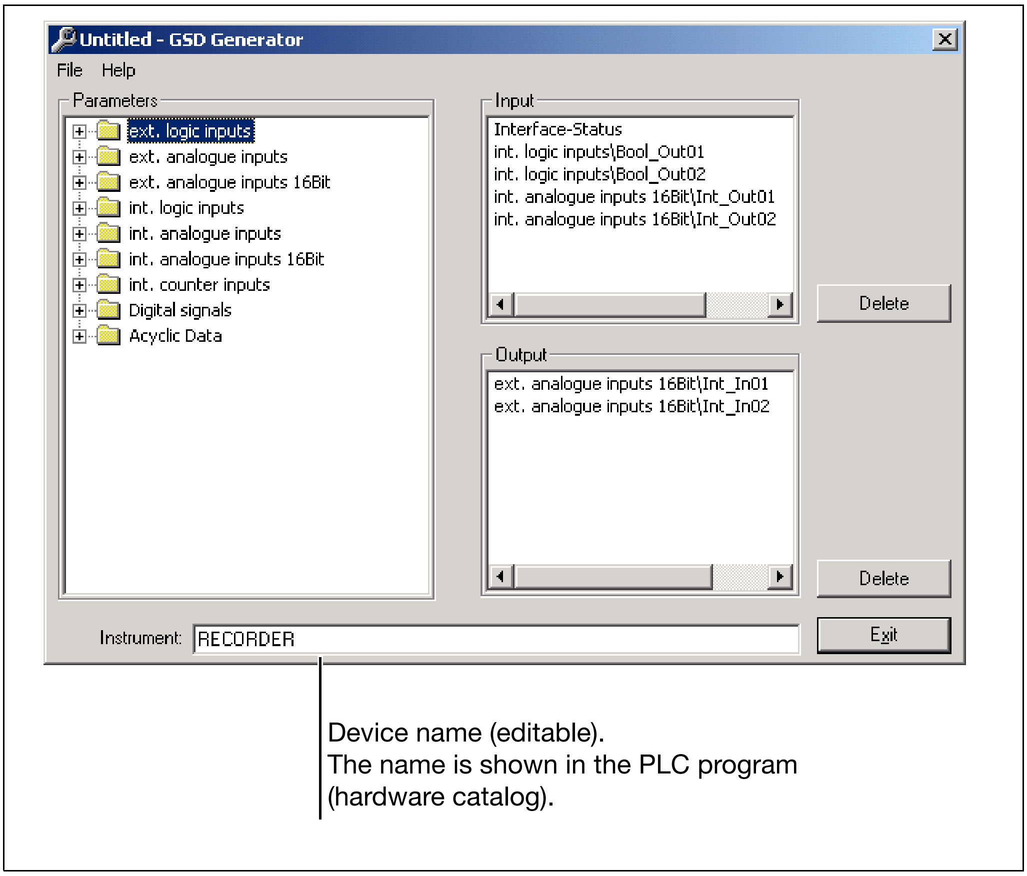
| Byte | Description | Type |
|---|---|---|
| [ 0] | Interface status | BYTE |
| [ 1] | int. logic inputs\Bool_Out01 | BOOLEAN |
| [ 2] | int. logic inputs\Bool_Out02 | BOOLEAN |
| [ 3] | int. analogue inputs 16Bit\Int_Out01 | INTEGER |
| [ 5] | int. analogue inputs 16Bit\Int_Out02 | INTEGER |

Outputs

| Byte | Description | Type |
|---|---|---|
| [ 0] | ext. analogue inputs 16Bit\Int_In01 | INTEGER |
| [ 2] | ext. analogue inputs 16Bit\Int_In02 | INTEGER |

**Select parameter**

If an existing file has been opened, or a new one created, all available parameters are shown in the parameter window.




A click with the left mouse button on the "+" (int. analogue inputs 16Bit) symbol or "-" (int. analogue inputs 16Bit) will extend the parameter list or reduce it.

Click on the parameter with the left mouse button, and, keeping it pressed, copy it to the input or output window by Drag & Drop.

**Remove parameter**

Parameters are deleted from the input or output window using the corresponding Delete button.

The parameter "Interface status" will automatically appear in the input window and cannot be deleted. It is used for diagnosis of the internal data transmission in the device and can be requested by the PLC:

| | |
|---|---|
| 0 | : internal communication in device is OK |
| unequal 0 | : faulty communication in device |

## 3.4 Connection example

The example below shows the path for the connection of a recorder to a S7 from Siemens.

### 3.4.1 RECORDER

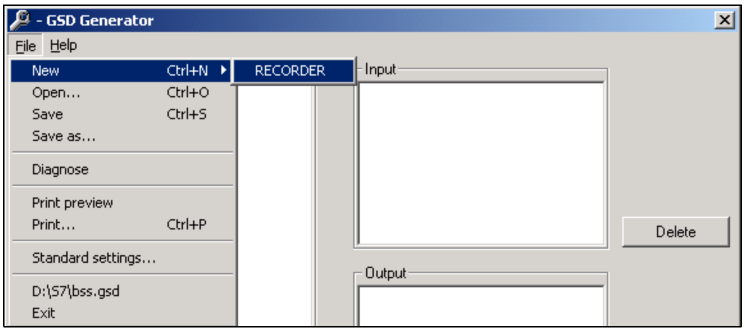
✱ Connect the device to the PLC.

✱ Set the device address.
The device (instrument) address can be selected via the instrument keys or through the setup program.

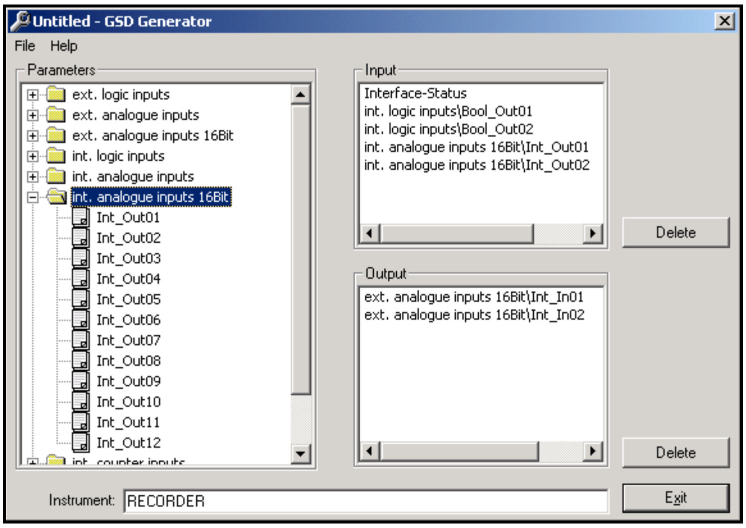
### 3.4.2 GSD generator

✱ Start up the GSD generator (Example: *Start* ➔ *Programs* ➔ *OEM devices* ➔ *Profibus* ➔ *GSD generator*).

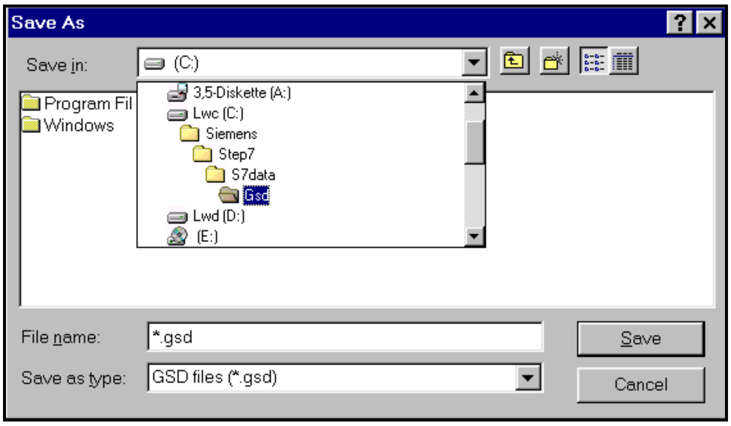
✱ Select the recorder.




✱ Select the variables that are transmitted to the master.

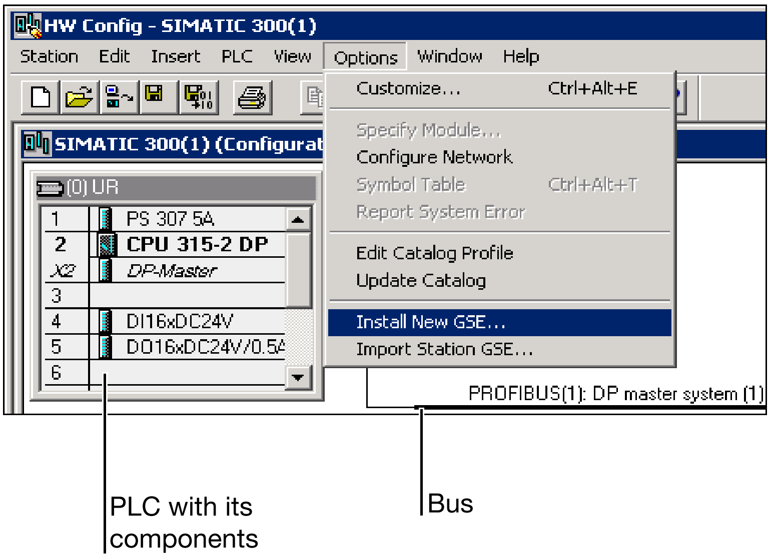


* Save the GSD file in any folder.



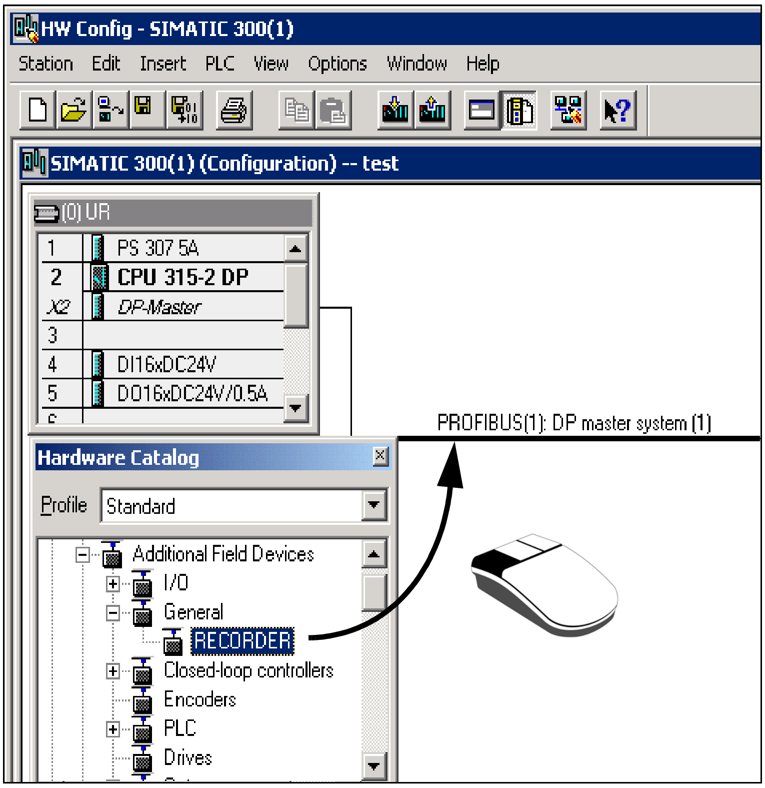

## 3.4.3 PLC configuration

* Start the PLC software.
* Call up the hardware configuration and execute menu command “Install new GSE”.



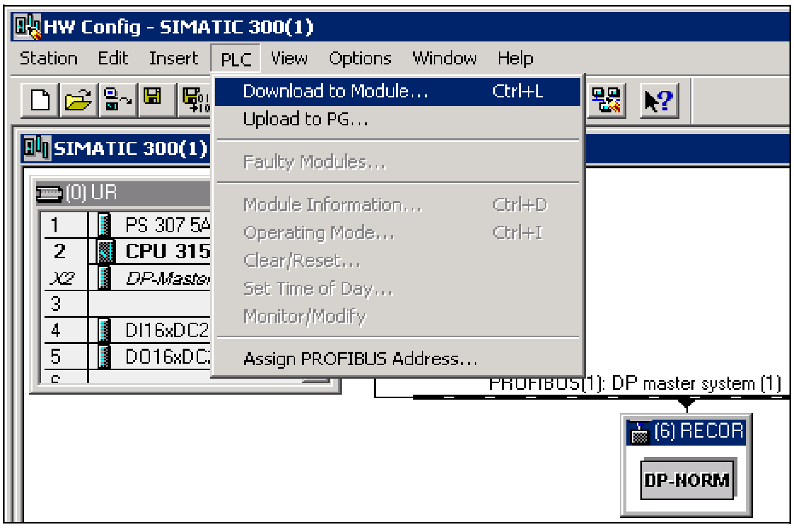

The new GSD file is read in and processed, and the recorder is inserted in the hardware catalog.

✱ Open the hardware catalog and place the new device in the working area.




The recorder is placed on the bus using the left mouse button. After releasing the mouse button, the recorder address has to be assigned.
The baud rate is determined automatically.

✱ Finally, you have to load the configuration into the PLC (*PLC ➔ Download to module*)



By using the paperless recorder, you can visualize and record up to 36 analog channels of a PLC.

This chapter describes the connection to the PROFIBUS-DP for the paperless recorder.

☞ All the devices described can be used exclusively as DP slaves.

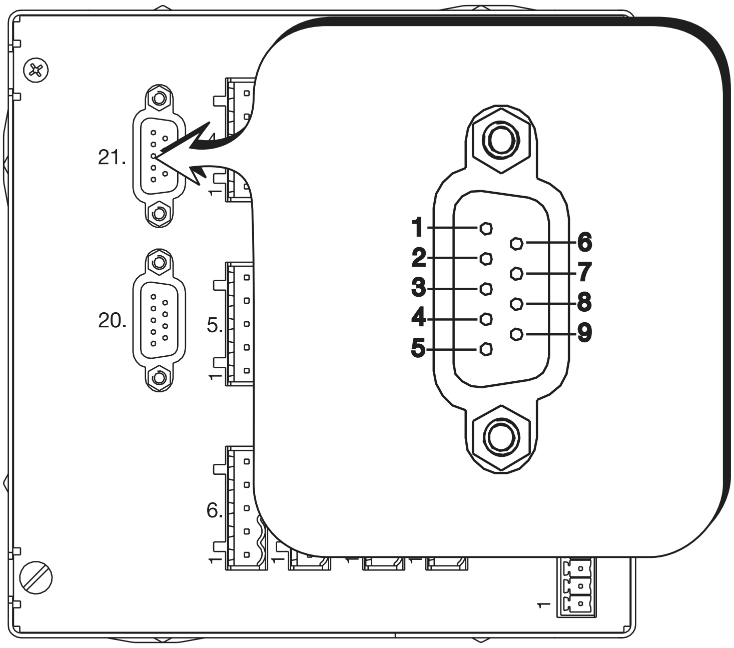
## 4.1 System requirements

The following requirments have to be met when connecting a paperless recorder to the PROFIBUS-DP:

- Fit the PROFIBUS-DP interface to the paperless recorder
- Program version from 100.03.02
  The program version can be requested from the paperless recorder menus via *Device (Instrument) info* ➔ *Version number*.

## 4.2 Connection diagram

**Rear view of paperless recorder**




**Connector 21**

**Interface** — **PROFIBUS-DP**

**Connections**

| Sub-D | Signal | Designation |
|---|---|---|
| 3 | RxD/TxD-P | Receive/Transmit Data-P, B-cable |
| 5 | DGND | Data transmission potential |
| 6 | VP | Supply voltage-P, (P5V) |
| 8 | RxD/TxD-N | Receive/Transmit data-N, A-cable |

☞ When making the connection to the PROFIBUS-DP it is important to ensure that the connectors 20 and 21 are not swapped. Connector 20 is reserved for the serial interface. The serial interface is used to read out instrument and process data from the paperless recorder. The connection and function of the serial interface are described in the Interface Description.

## 4.3 Setting the slave address

The slave address is set via the paperless recorder or the setup program.

| Setting | Meaning |
|---|---|
| 1 — 124 | Slave address, as selected |
| 125 | The setting of the slave address can be predefined by the bus master |

The baud rate is determined automatically (max. 12 Mbps).

If a new device address is selected, the device has to be reset (switch off/on) for the new address to be accepted.

## 4.4 Diagnostic and status messages

If errors occur during communication with the device, a message appears in the header (i) and in the "Instrument info" menu.

Please check the wiring and the master (PLC). It may be necessary to restart the system.

## 4.5 Acyclic data

You can read and write different measurement and process data of the paperless recorder with "acyclic data" (from program version 100.03.03).

The acyclic data, too, can be transmitted through the cyclic data transfer.

In order to establish the communication with the paperless recorder (device), it must receive 3 info bytes and a maximum of 10 bytes of actual data.

**Arrangement of the protocol**

| Byte No. | 1 | 2 | 3 | 4 — 13 |
|---|---|---|---|---|
| contents | control byte | function | address | actual data |

# 4 Device-specific data

**Control byte**

The control byte (byte No. 1) is organized as follows:

bit 0 — 3: length of actual data (in words)

bit 4 — 5: "toggle flag"
The two bits have their state changed (toggled) with every new job that is sent to the device so that it can recognize the new command. The bits may only be transmitted after the transmit buffer has been fully prepared for the new command.
Example:

| Bit 5 | Bit 4 | |
|---|---|---|
| 0 | 0 | no job present |
| 0 | 1 | bit 4 is set, job 1 is being processed |
| 1 | 0 | bit 5 is set, job 2 is being processed |
| 0 | 1 | bit 4 is set, job 3 is being processed |
| ... | ... | ............................................................... |

bit 6 — 7: Response OK: bit 6 = 0 and bit 7 = 1
Response faulty: bit 6 = 1 and bit 7 = 0
Bits 6 and 7 are a signal to the PLC that the command has been processed by the device and the next command to the device can be generated and transmitted by the PLC.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | bits 4 and 5 are returned unchanged by the device as a “job is being processed” info |
| 0 | 0 | 1 | 0 | bits 4 and 5 are returned unchanged by the device as a “job is being processed” info |
| 1 | 0 | 0 | 1 | fault-free processing of job with bit 4 = 1 |
| 0 | 1 | 0 | 1 | processing of job with bit 4 = 1 was not fault-free |
| 0 | 0 | 1 | 0 | bits 4 and 5 are returned unchanged by the device as a “job is being processed” info |
| 1 | 0 | 1 | 0 | fault-free processing of job with bit 5 = 1 |
| 0 | 1 | 1 | 0 | processing of job with bit 5 = 1 was not fault-free |
| ... | ... | ... | ... | ....................................................... |

**Function**

03x: read
10x: write

**Address**

The addresses below can be read and written. The list corresponds to a portion of the addresses that are contained in the interface description of the paperless recorder.
The address that is defined in the protocol is calculated as follows:

**address = base address + address of variable**

Example: address for the measurement of analog input 6:

**address = 0x35 + 0x0A = 0x3F**

Base address: 0x35

| Address of variable | Access | Data type | Signal designation |
|---|---|---|---|
| 0x00 | R/O | RealReal | meas. input 1 (analog input 1) |
| 0x02 | R/O | Real | meas. input 2 (analog input 2) |
| 0x04 | R/O | Real | meas. input 3 (analog input 3) |
| 0x06 | R/O | Real | meas. input 4 (analog input 4) |
| 0x08 | R/O | Real | meas. input 5 (analog input 5) |
| 0x0A | R/O | Real | meas. input 6 (analog input 6) |
| 0x0C | R/O | Real | meas. input 7 (analog input 7) |
| 0x0E | R/O | Real | meas. input 8 (analog input 8) |
| 0x10 | R/O | Real | meas. input 9 (analog input 9) |
| 0x12 | R/O | Real | meas. input 10 (analog input 10) |
| 0x14 | R/O | Real | meas. input 11 (analog input 11) |
| 0x16 | R/O | Real | meas. input 12 (analog input 12) |
| 0x18 | R/O | Real | not used |
| 0x1A | R/O | Real | not used |
| 0x1C | R/O | Real | not used |
| 0x1E | R/O | Real | not used |
| 0x20 | R/O | Real | counter value 1 |
| 0x22 | R/O | Real | counter value 2 |
| 0x24 | R/O | Real | external counter value 1 (from external I/O modules) |
| 0x26 | R/O | Real | external counter value 2 (from external I/O modules) |
| 0x28 | R/W | Real | external analog input 1 (from external I/O modules or via MODbus) |
| 0x2A | R/W | Real | external analog input 2 |
| 0x2C | R/W | Real | external analog input 3 |

| | | | |
|---|---|---|---|
| 0x2E | R/W | Real | external analog input 4 |
| 0x30 | R/W | Real | external analog input 5 |
| 0x32 | R/W | Real | external analog input 6 |
| 0x34 | R/W | Real | external analog input 7 |
| 0x36 | R/W | Real | external analog input 8 |
| 0x38 | R/W | Real | external analog input 9 |
| 0x3A | R/W | Real | external analog input 10 |
| 0x3C | R/W | Real | external analog input 11 |
| 0x3E | R/W | Real | external analog input 12 |
| 0x40 | R/W | Real | external analog input 13 |
| 0x42 | R/W | Real | external analog input 14 |
| 0x44 | R/W | Real | external analog input 15 |
| 0x46 | R/W | Real | external analog input 16 |
| 0x48 | R/W | Real | external analog input 17 |
| 0x4A | R/W | Real | external analog input 18 |
| 0x4C | R/W | Real | external analog input 19 |
| 0x4E | R/W | Real | external analog input 20 |
| 0x50 | R/W | Real | external analog input 21 |
| 0x52 | R/W | Real | external analog input 22 |
| 0x54 | R/W | Real | external analog input 23 |
| 0x56 | R/W | Real | external analog input 24 |
| 0x58 | R/W | Real | external analog input 25 |
| 0x5A | R/W | Real | external analog input 26 |
| 0x5C | R/W | Real | external analog input 27 |
| 0x5E | R/W | Real | external analog input 28 |
| 0x60 | R/W | Real | external analog input 29 |
| 0x62 | R/W | Real | external analog input 30 |
| 0x64 | R/W | Real | external analog input 31 |
| 0x66 | R/W | Real | external analog input 32 |
| 0x68 | R/W | Real | external analog input 33 |
| 0x6A | R/W | Real | external analog input 34 |
| 0x6C | R/W | Real | external analog input 35 |
| 0x6E | R/W | Real | external analog input 36 |

Base address: 0xA6

| Address of variable | Access | Data type | Signal designation |
|---|---|---|---|
| 0x00 | R/W | char 21 | text 1 for batch report |
| 0x0B | R/W | char 21 | text 2 for batch report |
| 0x16 | R/W | char 21 | text 3 for batch report |
| 0x21 | R/W | char 21 | text 4 for batch report |

Addresses are defined as bytes in the report, addressing of data word by word.

**Actual data**

A maximum of 10 bytes actual data may be defined. The number of the actual data being used (in words) is stored in bits 4 — 13.

**Command sequence**

- PLC sends job 1
  - PLC sets bit 4 in the control byte
  - PLC receives the answer “job OK” or “job faulty”
- PLC sends job 2
  - PLC resets bit 4 in the control byte and sets bit 5
  - PLC receives the answer “job OK” or “job faulty”
- PLC sends job 3
  - PLC resets bit 5 in the control byte and sets bit 4
  - PLC receives the answer “job OK” or “job faulty”
- and so on.

**Example (write):**

Text 1 for the batch report has to be written. Since a batch text may consist of up to 20 characters, it is transmitted in 2 parts with 10 characters each. Character 21 (0x) can be dispensed with.

The following bytes have to be transmitted to the recorder:

**a.) Byte 1 — 10**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x25 | 0x10 | 0xA6 | 0x54 | 0x68 | 0x75 | 0x65 | 0x72 | 0x69 | 0x6E | 0x67 | 0x65 | 0x72 |
| | | | T | h | u | e | r | i | n | g | e | r |

**b.) Byte 11 — 20**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x15 | 0x10 | 0xAB | 0x2D | 0x42 | 0x72 | 0x61 | 0x74 | 0x77 | 0x75 | 0x72 | 0x73 | 0x74 |
| | | | - | B | r | a | t | w | u | r | s | t |

When sending the first 10 characters, the start address is 0xA6. Since addressing is done word by word, the second 10 characters have to be sent with the address from 0xAB (0xA6 + 5).

## 4.6 PLC data in 16-bit format

The functions described below can only be implemented on paperless recorders from program version 100.03.05.

**Internal measurement inputs of the paperless recorder**

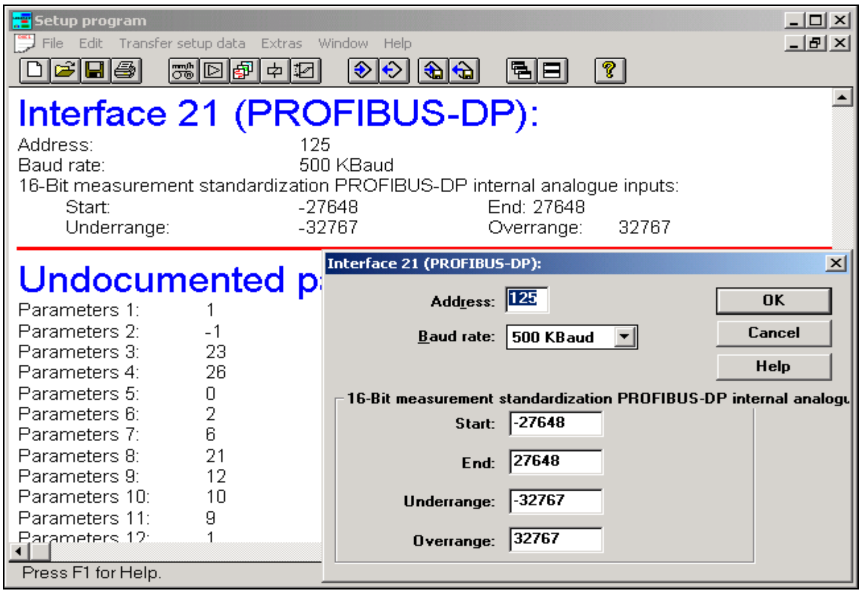
From program version 100.03.05, the internal measurement inputs (1 — 6 or 1 — 12) can not only be sent to the PROFIBUS master (PLC) in “Real” format (4 bytes), but also in “Integer” format (2 bytes).

Using the setup program for the paperless recorder, four values for measurement standardization have to be entered for **all internal channels** (not for each individually).

- range start
- range end
- value at underrange
- value at overrange

The internal measurements are converted from “Real” format to “Integer” format. In order to enable a uniform conversion, these four parameters have been inserted.

The parameters are entered in the field “Interface 21”. The dialog can be called up by a double click on the working area, or through the menu *Edit* ➔ *Interface 21 (PROFIBUS-DP)*.




**External measurement inputs of the paperless recorder**

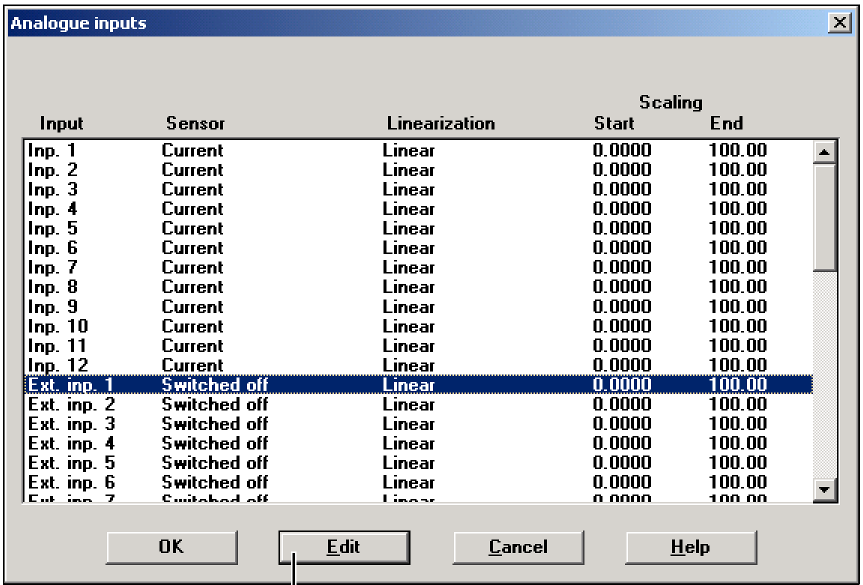
The external measurement inputs (1 — 36) can also be sent to the paperless recorder, either in “Real” format (4 bytes), or “Integer” format (2 bytes). Which data format is to be used, can also be decided with the help of the GSD generator.

With the “Integer” format, the parameters *Range start* and *Range end* are used to carry out standardization to 16 bits. This can be done through the setup program, or from the instrument keys.

The analog input of the PLC provides a 16-bit measurement in the range from -27648 to +27648; the ranges depend on the input sensor that was selected or the input card being used.

**Example**

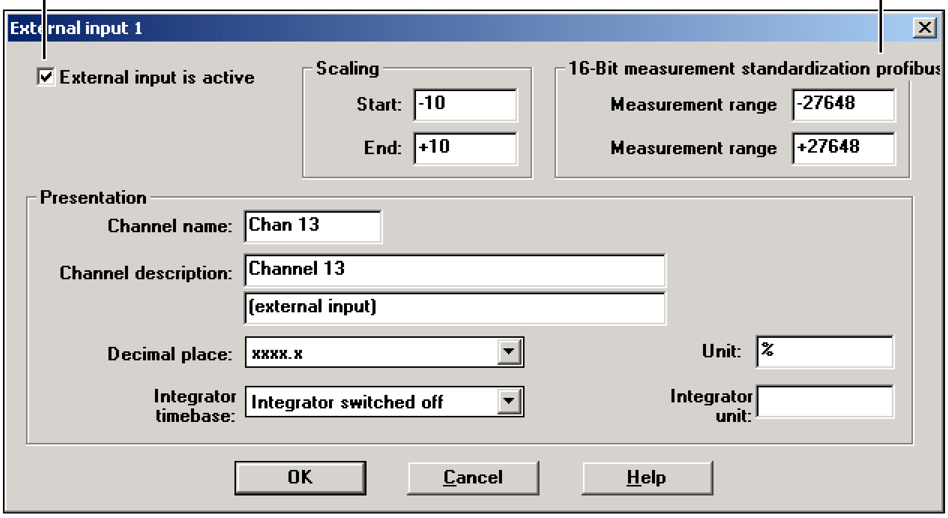
| | Input meas. range of PLC | Standardiza-tion range of PLC | Range paperless recorder | Scaling paperless recorder |
|---|---|---|---|---|
| Range start | -10V | -27648 | -27648 | -10 |
| Range end | +10V | +27648 | +27648 | +10 |




First of all, the “Edit” button has to be activated for the external measurement inputs, ...

... then the exernal input can be activated and ...

... finally, standardization can be carried out.



## 4.7 Bit-by-bit coding of binary signals

The signals described below are available for paperless recorders with program version 100.03.05 or above and can be addressed using the GSD generator.

| Address | Access | Data type | Signal designation |
|---|---|---|---|
| Alarm_Group1-6 | R/O | bit0 | alarm group 1<br>0 = no alarm<br>1 = at least 1 limit infringed in group |
| | R/O | bit1 | alarm group 2 |
| | R/O | bit2 | alarm group 3 |
| | R/O | bit3 | alarm group 4 |
| | R/O | bit4 | alarm group 5 |
| | R/O | bit5 | alarm group 6 |
| | R/O | bit6-7 | not used |
| | | | |
| Int. logic inp. 1-7 | R/O | bit0 | logic input 1<br>0 = open / 1 = closed |
| | R/O | bit1 | logic input 2 |
| | R/O | bit2 | logic input 3 |
| | R/O | bit3 | logic input 4 |
| | R/O | bit4 | logic input 5 |
| | R/O | bit5 | logic input 6 |
| | R/O | bit6 | logic input 7 |
| | R/O | bit7 | not used |
| | | | |
| Additional_ dig.signals | R/O | bit0 | combination alarm<br>0 = no alarm<br>1 = at least 1 limit infringed in device |
| | R/O | bit1 | diskette reserve signal<br>0 = disk. reserve not yet reached<br>1 = replace diskette |
| | R/O | bit2 | fault<br>0 = no fault / 1 = fault |
| | R/O | bit3-7 | not used |
| | | | |

| Address | Access | Data type | Signal designation |
|---|---|---|---|
| Output1-6 | R/O | bit0 | relay output 1<br>0 = not active / 1 = active |
| | R/O | bit1 | relay output 2 |
| | R/O | bit2 | relay output 3 |
| | R/O | bit3 | relay output 4 |
| | R/O | bit4 | relay output 5 |
| | R/O | bit5 | open-collector output<br>0 = not active / 1 = active |
| | R/O | bit6-7 | not used |
| | | | |
| Ext.logic inp.1-6 | R/W | bit0 | external logic input 1<br>0 = open / 1 = closed |
| | R/W | bit1 | external logic input 2 |
| | R/W | bit2 | external logic input 3 |
| | R/W | bit3 | external logic input 4 |
| | R/W | bit4 | external logic input 5 |
| | R/W | bit5 | external logic input 6 |
| | R/W | bit6-7 | not used |

## 4.8 External inputs on faults in the data exchange

As long as there is no data exchange between the PLC and the recorder, the external analog inputs of the recorder are treated as “invalid” (display --------). Thus it is possible to detect, during the evaluation of the measurement data, that there were no valid values present for this period. This only applies to the external measurement inputs.

All other data (binary signals, batch texts, ...) will be frozen and remain at their current values.

When using recorders in a PROFIBUS-DP system, please take note of their data format.

**Integer values**

Integer values are transmitted in the following format:

- first the high byte,
- then the low byte.

**Float values/ real values**

The float/real values of the cyclic data for the paperless recorder are transmitted using the IEEE-754 standard format (32 bits).

The float/real values for the acyclic data of the paperless recorder are transmitted in the MODbus format.

The IEEE-754 standard format and the MODbus format differ in the transmission sequence of the individual bytes. In the MODbus format bytes 1 and 2 are swapped with bytes 3 and 4 (first the high byte, then the low byte).

**Single-float format (32 bits) as per IEEE 754 standard**

| | | | |
|---|---|---|---|
| SEEEEEEE | EMMMMMMM | MMMMMMMM | MMMMMMMM |
| byte 1 | byte 2 | byte 3 | byte 4 |

S - sign bit (bit31)
E - exponent in complement to base 2 (bit23 — bit30)
M - 23 bit normalized mantissa (bit0 — bit22)

Example:
calculation of the real number from sign, exponent and mantissa.

byte1 = 40h, byte2 = F0, byte 3 = 0, byte 4 = 0

40F00000h = 0100 0000 1111 0000 0000 0000 0000 0000b

S = 0
E = 100 0000 1
M = 111 0000 0000 0000 0000 0000

Value = $-1^{S} \cdot 2^{exponent-127} \cdot (1 + M_{b22} \cdot 2^{-1} + M_{b21} \cdot 2^{-2} + M_{b20} \cdot 2^{-3} + M_{b19} \cdot 2^{-4} + \ldots)$

Value = $-1^{0} \cdot 2^{129-127} \cdot (1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4})$

Value = $1 \cdot 2^{2} \cdot (1 + 0.5 + 0.25 + 0.125 + 0)$

Value = $1 \cdot 4 \cdot 1.875$

Value = 7.5

**MODbus float format**

| Address x | | Address x+1 | |
|---|---|---|---|
| MMMMMMMM | MMMMMMMM | SEEEEEEE | EMMMMMMM |
| byte 3 | byte 4 | byte 1 | byte 2 |

After/before the transmission from/to the device, the bytes of the float value have to be swapped accordingly.

Many compilers (e.g. Microsoft C++, Turbo C++, Turbo Pascal, Keil C51) store the float values in the following order (Intel compatibility):

**float value**

| Storage address x | Storage address x+1 | Storage address x+2 | Storage address x+3 |
|---|---|---|---|
| MMMMMMMM | MMMMMMMM | EMMMMMMM | SEEEEEEE |
| byte 4 | byte 3 | byte 2 | byte 1 |

Please check how float values are stored in your application. If necessary, the bytes have to be swapped accordingly.