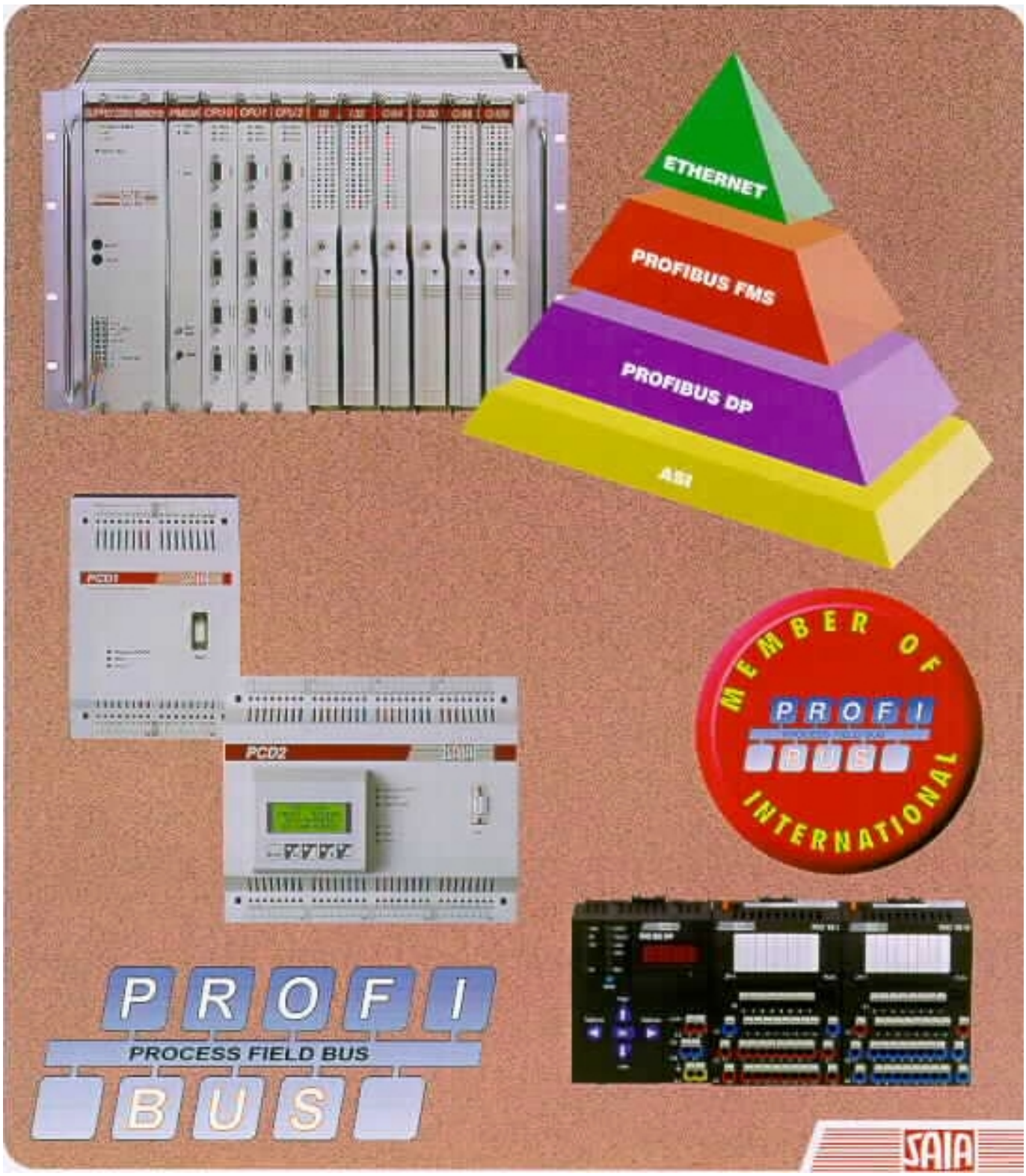
SAIA-Burgess Electronics

SWITCHES - MOTORS - CONTROLLERS

# SAIA® PCD

## Process Control Devices

# Manual
# PROFIBUS-DP




English edition 26/765 E1

SAIA-Burgess Electronics Ltd.
Bahnhofstrasse 18
CH-3280 Murten (Switzerland)
http;//www.saia-burgess.com

BA: Electronic Controllers Telephone 026 / 672 71 11
Telefax 026 / 670 44 43

## SAIA-Burgess Companies

**Switzerland** SAIA-Burgess Electronics AG
Freiburgstrasse 33
CH-3280 Murten
☎ 026 672 77 77, Fax 026 670 19 83

**Germany** SAIA-Burgess Electronics GmbH
Daimlerstrasse 1k
D-63303 Dreieich
☎ 06103 89 060, Fax 06103 89 06 66

**Austria** SAIA-Burgess Electronics Ges.m.b.H.
Schallmooser Hauptstrasse 38
A-5020 Salzburg
☎ 0662 88 49 10, Fax 0662 88 49 10 11

**Italy** SAIA-Burgess Electronics S.r.l.
Via Cadamosto 3
I-20094 Corsico MI
☎ 02 48 69 21, Fax 02 48 60 06 92

**France** SAIA-Burgess Electronics Sàrl.
10, Bld. Louise Michel
F-92230 Gennevilliers
☎ 01 46 88 07 70, Fax 01 46 88 07 99

**Nederlands** SAIA-Burgess Electronics B.V.
Hanzeweg 12c
NL-2803 MC Gouda
☎ 0182 54 31 54, Fax 0182 54 31 51

**Belgium** SAIA-Burgess Electronics Belgium
Avenue Roi Albert 1er, 50
B-1780 Wemmel
☎ 02 456 06 20, Fax 02 460 50 44

**Hungary** SAIA-Burgess Electronics Automation Kft.
Liget utca 1.
H-2040 Budaörs
☎ 23 501 170, Fax 23 501 180

## Representatives

**Great Britain** Canham Controls Ltd.
25 Fenlake Business Centre, Fengate
Peterborough PE1 5BQ UK
☎ 01733 89 44 89, Fax 01733 89 44 88

**Denmark** Malthe Winje Automation AS
Håndværkerbyen 57 B
DK-2670 Greve
☎ 70 20 52 01, Fax 70 20 52 02

**Norway** Malthe Winje Automasjon AS
Haukelivn 48
N-1415 Oppegård
☎ 66 99 61 00, Fax 66 99 61 01

**Sweden** Malthe Winje Automation AB
Truckvägen 14A
S-194 52 Upplands Våsby
☎ 08 795 59 10, Fax 08 795 59 20

**Suomi/ Finland** ENERGEL OY
Atomitie 1
FIN-00370 Helsinki
☎ 09 586 2066, Fax 09 586 2046

**Portugal** INFOCONTROL Electronica e Automatismo LDA.
Praceta Cesário Verde, No 10 s/cv, Massamá
P-2745 Queluz
☎ 21 430 08 24, Fax 21 430 08 04

**Spain** Tecnosistemas Medioambientales, S.L.
Poligono Industrial El Cabril, 9
E-28864 Ajalvir, Madrid
☎ 91 884 47 93, Fax 91 884 40 72

**Czech Republic** ICS Industrie Control Service, s.r.o.
Modranská 43
CZ-14700 Praha 4
☎ 2 44 06 22 79, Fax 2 44 46 08 57

**Poland** SABUR Ltd.
ul. Druzynowa 3A
PL-02-590 Warszawa
☎ 22 844 63 70, Fax 22 844 75 20

**Australia** Siemens Building Technologies Pty. Ltd.
Landis & Staefa Division
411 Ferntree Gully Road
AUS-Mount Waverley, 3149 Victoria
☎ 3 9544 2322, Fax 3 9543 8106

**Argentina** MURTEN S.r.l.
Av. del Libertador 184, 4° “A”
RA-1001 Buenos Aires
☎ 054 11 4312 0172, Fax 054 11 4312 0172

## After sales service

**USA** SAIA-Burgess Electronics Inc.
1335 Barclay Boulevard
Buffalo Grove, IL 60089, USA
☎ 847 215 96 00, Fax 847 215 96 06

Issue : 22.11.2000

**SAIA® Process Control Devices**

## Manual

# PROFIBUS-DP

## with SAIA® PCD


Edition 26/765 E1 - 03.1999

Subject to technical changes

# Updates

**Manual : PROFIBUS-DP with SAIA® PCD - Edition E1**

| Date | Chapter | Page | Description |
|---|---|---|---|
| | | | |
| 02.2001 | 7.3 | 7-10 | XOB → EXOB |
| | | | |
| | | | |
| | | | |
| | | | |

# Concerning this manual

The concept and practical application of PROFIBUS-DP is described in detail and complemented with the addition of various examples.

If PROFIBUS is new territory for you, it is advisable to study the whole manual.

If you already have some experience with PROFIBUS-FMS, you are advised to study the following chapters:

2. SAIA® PCD devices for the PROFIBUS-DP network
4. The PROFIBUS-DP configurator
5. Programming
6. Commissioning a PROFIBUS-DP network

If you are already familiar with PROFIBUS-DP, we recommend that you study the following chapters:

2. SAIA® PCD devices for the PROFIBUS-DP network
4. The PROFIBUS-DP configurator
5. Programming

If you wish to have a quick summary of the use of SAIA devices with PROFIBUS-DP, we recommend that you study Chapter 7:

7. Quick guide to creating a PROFIBUS-DP network

Notes

# Contents



Page

Page



Page

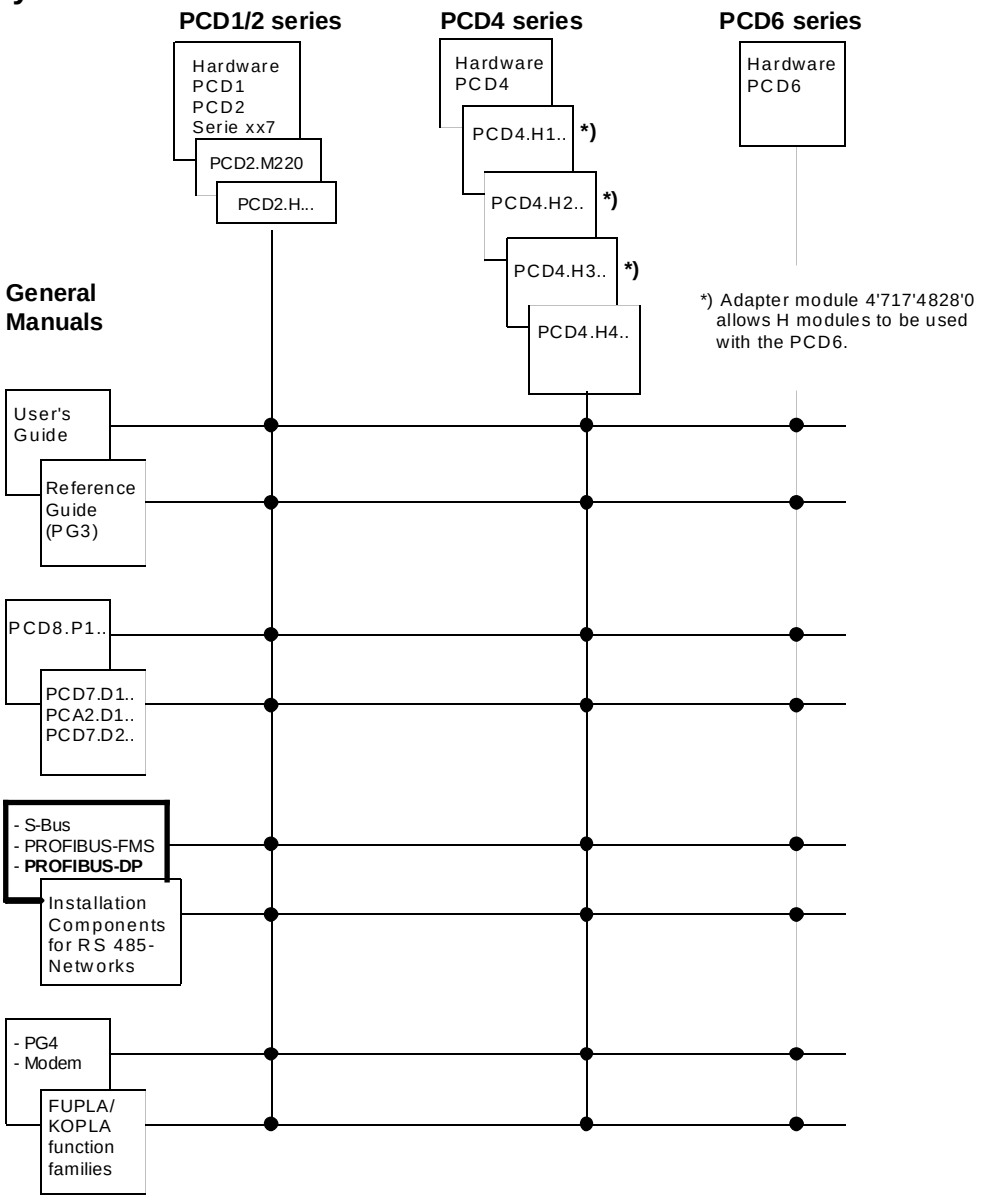
**Please note:**

A number of detailed manuals are available to aid installation and operation of the SAIA PCD. These are for use by technically qualified staff, who may also have successfully completed one of our "workshops".

To obtain the best performance from your SAIA PCD, closely follow the guidelines for assembly, wiring, programming and commissioning given in these manuals. In this way, you will also become one of the many enthusiastic SAIA PCD users.

If you have any technical suggestions or recommendations for improvements to the manuals, please let us know. A form is provided on the last page of this manual for your comments.

## Summary



# Reliability and safety of electronic controllers

SAIA-Burgess Electronics Ltd. is a company which devotes the greatest care to the design, development and manufacture of its products:

- state-of-the-art technology
- compliance with standards
- ISO 9001 certification
- international approvals: e.g. Germanischer Lloyd, UL, Det Norske Veritas, CE mark ...
- choice of high-quality componentry
- quality control checks at various stages of production
- in-circuit tests
- run-in (burn-in at 85°C for 48h)

Despite every care, the excellent quality which results from this does have its limits. It is therefore necessary, for example, to reckon with the natural failure of components. For this reason SAIA-Burgess Electronics Ltd. provides a guarantee according to the "General terms and conditions of supply".

The plant engineer must in turn also contribute his share to the reliable operation of an installation. He is therefore responsible for ensuring that controller use conforms to the technical data and that no excessive stresses are placed on it, e.g. with regard to temperature ranges, overvoltages and noise fields or mechanical stresses.

In addition, the plant engineer is also responsible for ensuring that a faulty product in no case leads to personal injury or even death, nor to the damage or destruction of property. The relevant safety regulations should always be observed. Dangerous faults must be recognized by additional measures and any consequences prevented. For example, outputs which are important for safety should lead back to inputs and be monitored from software. Consistent use should be made of the diagnostic elements of the PCD, such as the watchdog, exception organization blocks (XOB) and test or diagnostic instructions.

If all these points are taken into consideration, the SAIA PCD will provide you with a modern, safe programmable controller to control, regulate and monitor your installation with reliability for many years.

# 1. Characteristics and fundamentals

PROFIBUS (PROcess FIeld BUS) is the successful, open, industrial fieldbus standard for a broad application spectrum.

## 1.1 Characteristics

- **Open**
  PROFIBUS enables the exchange of data between devices from different manufacturers, without special interface adjustments. PROFIBUS has been standardized under German standard DIN 19 245 and European standard pr EN 50170.

- **Vendor independent**
  PROFIBUS devices are offered by many qualified manufacturers. This enables the user to select the most suitable device from a preferred manufacturer for whichever application.

- **Optimized for a wide field of use**
  PROFIBUS comprises the following three application-specific versions:

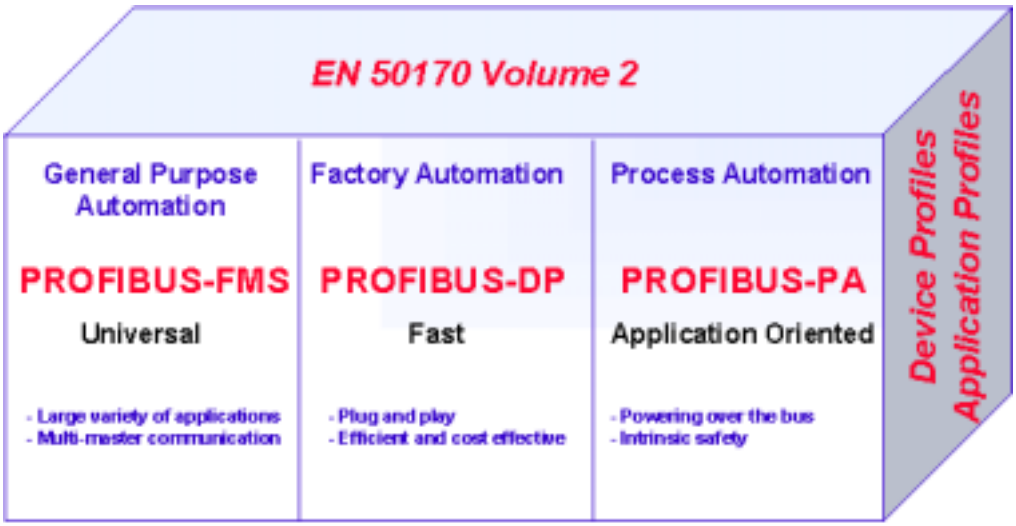



Figure 1: Overview of PROFIBUS versions

PROFIBUS has been standardized under European standard EN 50170 and the earlier German standard DIN 19245 parts 1 - 4.

**PROFIBUS-FMS** (FMS = Fieldbus Message Specification)

This is the general-purpose solution for communications tasks at the field and cell levels of the industrial communications hierarchy. Powerful FMS services open up a wide range of applications, provide great flexibility and enable comprehensive communication tasks to be managed with cyclic or acyclic data communication at medium speed.

**PROFIBUS-DP** (DP = Decentralized Peripherals**)**

Optimized for high speed, this PROFIBUS version has been especially tailored for communication between automation systems and local peripherals, enabling plug-and-play for field devices.
PROFIBUS-FMS and DP use the same transmission technology and the same bus access protocol. Both versions can therefore run simultaneously and in combination on a single cable.

**PROFIBUS-PA** (PA = Process Automation)

PROFIBUS-PA is the PROFIBUS version for process automation applications. PROFIBUS-PA uses the intrinsically safe transmission technology defined in IEC 1158-2 and enables the remote supply of stations through the bus.

The protocol functions of PROFIBUS-FMS, DP and PA are complemented by profiles which describe how PROFIBUS is used in special fields of application, such as process automation, building control technology or automated manufacturing.
Device profiles define device-specific functions. The device description language (DDL) and function blocks ensure that devices are fully interoperable.

# Transparent Communication from Sensor/Actuator up to Area

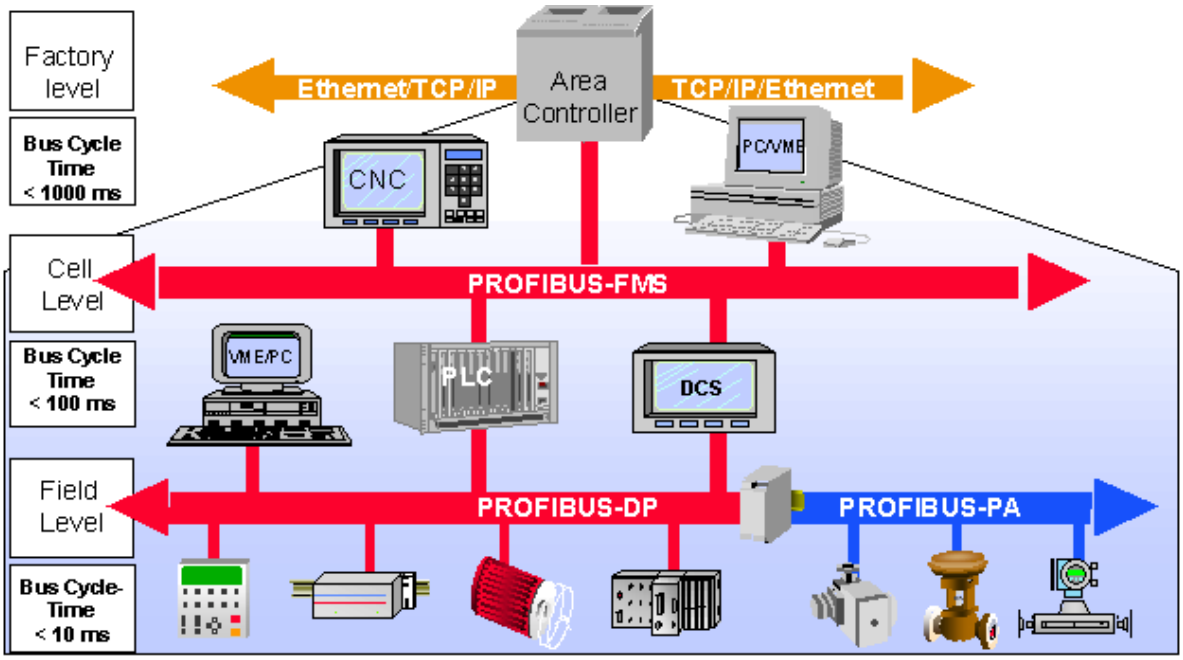



Figure 2: Communication in the field area

- **Proven**
  PROFIBUS is a complete, proven technology which has already stood the test of more than 100 000 applications in the fields of building automation, automated production, process automation and drive technology.

- **Certificated**
  Tests of conformity and interoperability performed in laboratories authorized by the PROFIBUS user organization (PNO), combined with PNO device certification, give the user an assurance of guaranteed quality and functionality, even in multi-vendor installations.

## 1.2 Basic principles

To construct a PROFIBUS network and exchange data across it, you must now read and understand the following chapters in full. If you plan PROFIBUS communication with SAIA® PCD devices only, this in itself is a very simple undertaking and you can continue directly with chapter 2.

However, if you would also like to improve your knowledge and understanding of PROFIBUS theory, you will certainly find the following chapter useful.

### 1.2.1 ISO/OSI reference model

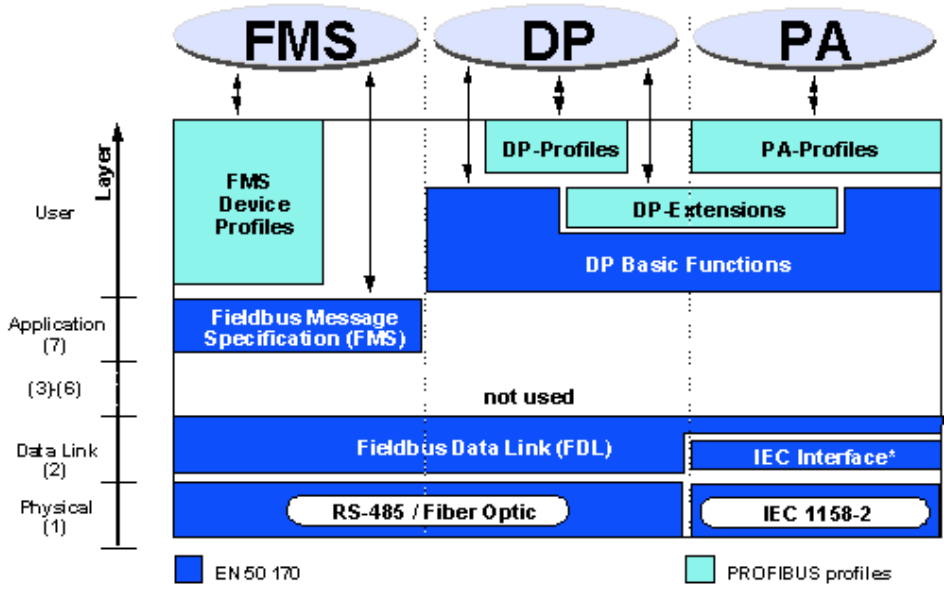



Figure 3: ISO/OSI reference model

### 1.2.2 PROFIBUS layer 1 (physical layer)

PROFIBUS protocol architecture follows the ISO / OSI (open system interconnection) reference model, in line with international standard ISO 7498.
In this model each layer takes on precisely defined tasks. Layer 1 (physical layer) defines the physical transmission technology. Layer 2 (data link layer) defines the bus access procedure. Layer 7 (application layer) defines the user level.

A fieldbus system's field of use is essentially determined by the choice of transmission medium and the physical bus interface. Apart from the requirements of transmission security, expenditure on purchasing and installing the bus cable is crucially important. The PROFIBUS standard therefore provides for various forms of transmission technology, while retaining a uniform bus protocol.

**Cable-bound transmission:** This version complies with US standard EIA RS-485 and was defined as the basic version for applications in the field of production engineering, building control technology and drive technology. It uses screened, twisted copper cable with a pair of conductors.

**Fibre-optic cable:** For applications in very interference-prone environments, and to extend working range at high transmission speeds, the PNO has worked out a specification for fibre-optic based transmission technology.

**Intrinsically safe transmission:** PROFIBUS-PA enables the intrinsically safe transmission and supply of stations across the bus. The transmission technology is described in international standard IEC 1158-2.

### 1.2.3 PROFIBUS layer 2 (data link layer)

The second layer of the OSI reference model realizes the functions of bus access control, data protection and the transaction of transmission protocols and telegrams. In PROFIBUS, layer 2 is called the fieldbus data link (FDL).

Bus access control (MAC, Medium Access Control) defines the procedure for when a station on the bus can transmit data. The MAC must ensure that, at any one time, only one station ever has authorization to transmit.

The PROFIBUS bus access method therefore includes a **token-passing procedure** for communication between complex stations (masters) and an underlying **master-slave procedure** for the communication of complex stations with low-expense peripheral devices (slaves). This combined procedure is called a **hybrid bus access procedure** (see figure 3).

In PROFIBUS, the token-passing procedure only applies between complex stations (masters).

The master-slave procedure allows the master (active station) currently holding transmission authorization to speak to its assigned slave devices (passive stations). This gives the master the opportunity of transmitting messages to the slaves or uploading messages from them.

With this hybrid access method for PROFIBUS, the following system configurations can be realized:

- Straight master-slave system.
- Straight master-master system (token passing).
- A system combining both methods.

Figure 3 shows a PROFIBUS structure with three active stations (masters) and seven passive stations (slaves). The three master devices form a logical token ring.

If an active station receives the token telegram, it can exercise for a certain time the function of master across the bus, communicating with all slave stations in a master-slave relationship and with all master stations in a master-master relationship.

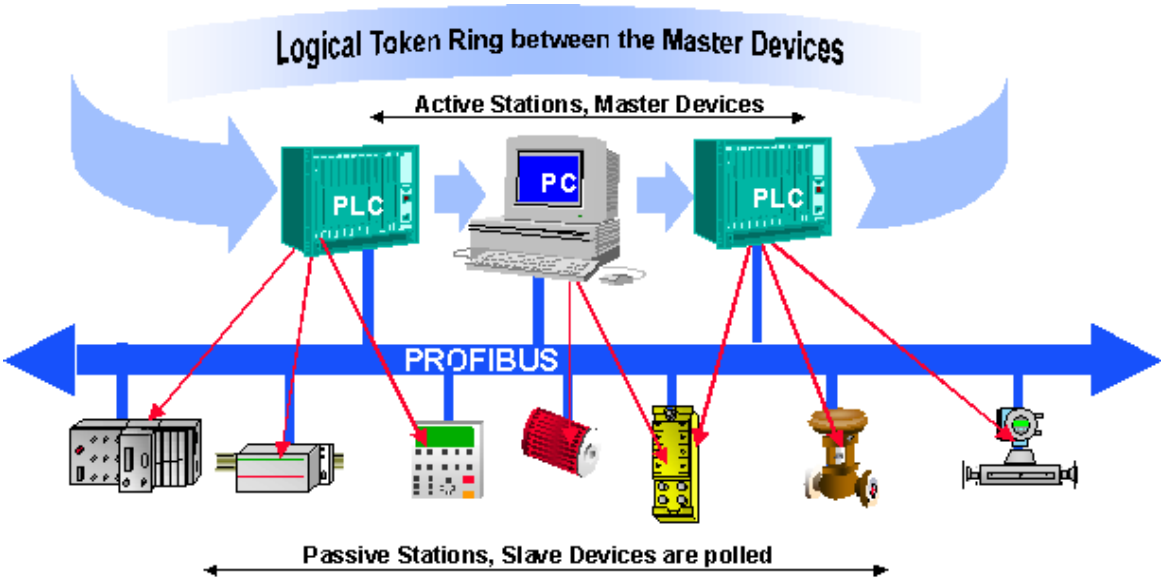



Figure 4: Hybrid bus access method

The term: "token ring" refers to the organizational stringing together of active stations which, by their bus addresses, form a logical ring. In this ring the token (bus access authorization) is passed from one master to the next in a predefined order (ascending addresses).

The task of the bus access controller (MAC) for active stations is to recognize this logical assignment in the bus system's "run-up phase" and establish the token ring. While operating in "run", any (active) station which is faulty or switched off is disconnected from the ring and any newly added active station is connected to it. These performance characteristics, together with the recognition of faults in the transmission medium, line receiver and errors of station addressing (e.g. multiple occupation) or token passing (e.g. multiple tokens or token loss), are typical features of PROFIBUS access control.

Another important task of layer 2 is data protection. The PROFIBUS layer 2 telegram formats enable great transmission security. All telegrams have a **hamming distance of HD = 4**. This is achieved by application of the rules under international standard IEC 870-5-1, selecting special start and end characters for telegrams, slip-resistant synchronization, parity bits and control bytes.

PROFIBUS-FMS and PROFIBUS-DP both use their own specific subset of layer 2 services (see table below). Services are called from higher-ranking levels via the service access points (SAPs) of level 2. With PROFIBUS-FMS these service access points are used for addressing the logical communication relationships. With PROFIBUS-DP a precisely defined function is assigned to each service access point used. For all active and passive stations, several service access points can be used in parallel. A difference is drawn between source (SSAP) and receiving service access points (RSAP).

Table: Services of the PROFIBUS data protection layer (layer 2).

| **Service** | **Function** | **Used with** |
|---|---|---|
| SDA | Send Data with Acknowledge | FMS |
| SRD | Send And Request Data With Reply | FMS DP |
| SDN | Send Data With No Acknowledge | FMS DP |
| CSRD | Cyclic Send And Request Data With Reply | FMS |

The following SAPs are occupied by PROFIBUS-DP and cannot be used for FMS in a mixed FMS- DP network:

| **Function** | **SSAP** | **DSAP** | **Service** |
|---|---|---|---|
| Data_Exchange | Default | Default | SRD |
| Chk_Cfg | 62 | 62 | SRD |
| Set_Prm | 62 | 61 | SRD |
| Slave_Diag | 62 | 60 | SRD |
| Get_Cfg | 62 | 59 | SRD |
| Global_Control | 62 | 58 | SDN |
| RD_Outp | 62 | 57 | SRD |
| RD_Inp | 62 | 56 | SRD |
| Set_Slave_Add | 62 | 55 | SRD |
| reserved | | 54 | |
| reserved | | 53 | |
| reserved | | 52 | |
| reserved | | 51 | |

### 1.2.4 PROFIBUS-DP

- **PROFIBUS layer 7 (application layer)**

  Layer 7 of the ISO/OSI reference model is not used in PROFIBUS-DP.

- **PROFIBUS DP characteristics**
  - Replaces costly parallel cabling between PLC/PC and I/Os
  - Fast: transmits 1 kByte input and output data in less than 2ms
  - Powerful tools reduce engineering costs
  - Supported by all main PLC manufacturers
  - Great diversity of products available, e.g. PLCs/PCs, I/Os, drives, valves, encoders
  - Cyclic and acyclic data transmission possible
  - Monomaster and multimaster networks can be created
  - Up to 246 Bytes of input and output data possible per station

- **PROFIBUS DP device types**
  PROFIBUS-DP defines three device types:

- **DP master class 1 (DPM1)**
  Central control which exchanges data with the local I/Os (DP slaves).
  Several DPM1s are allowed, typical devices are PLCs, PCs, VMEs

- **DP master class 2 (DPM2)**
  Project planning, monitoring or engineering tool which is used for commissioning or parameter setting/ monitoring of DP slaves.

- **DP slave**
  Local device with direct interface to input / output signals.
  Typical devices are I/Os, drives, valves, control units...

- **PROFIBUS DP systems**
  PROFIBUS-DP can operate either as a monomaster or multimaster system:

  - **Monomaster System**

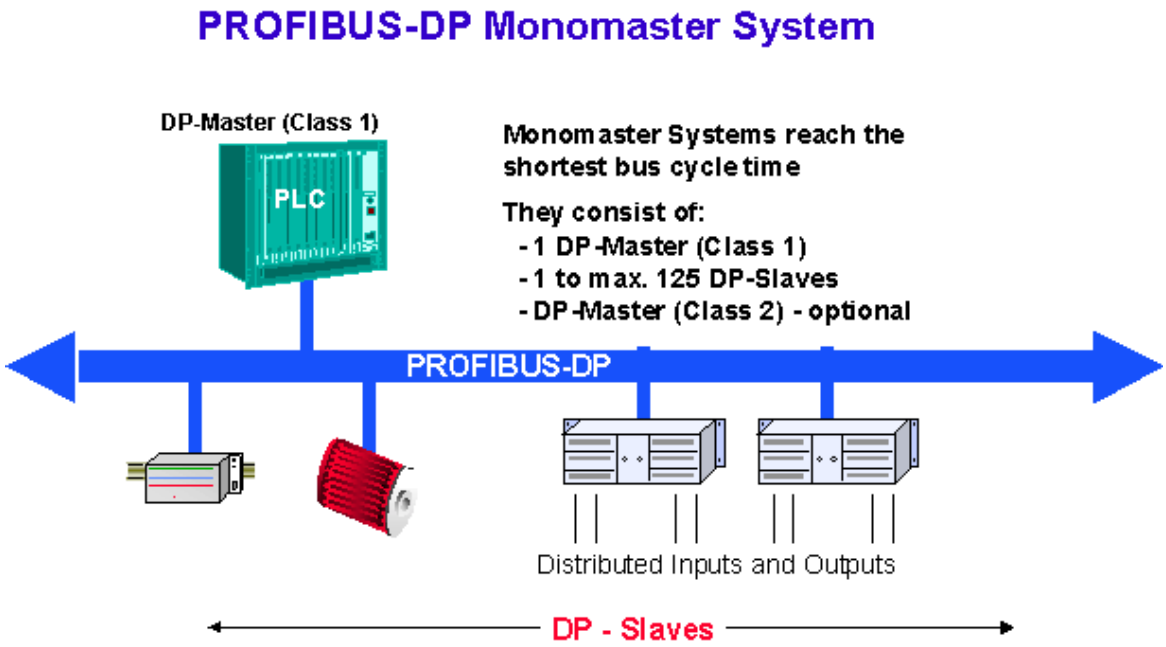



Figure 5: PROFIBUS-DP monomaster system

PROFIBUS-DP usually works as a monomaster system. This means that a master device, e.g. a PLC, is networked via PROFIBUS-DP with distributed peripheral devices, e.g. inputs and outputs. PROFIBUS-DP replaces the parallel cabling between the PLC and local peripherals.

A PROFIBUS-DP monomaster system consists of between 1 and 125 slaves, a class 1 master (i.e. the PLC) and an optional class 2 master (i.e. a configurator).

Monomaster systems achieve the shortest bus cycle times; they transmit 1 kByte of input and output data in less than 2 ms.

- **Multimaster System**

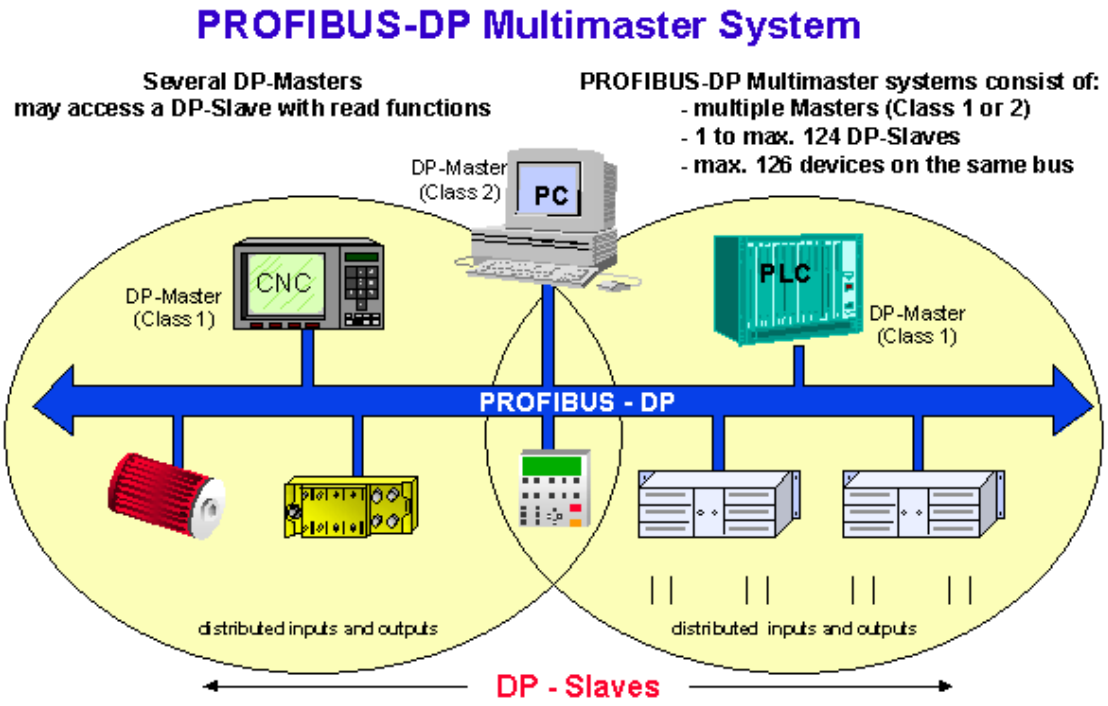



Figure 6: PROFIBUS-DP multimaster system

PROFIBUS-DP can also be used in a multimaster configuration.

Figure 6 shows two active devices, a PLC and a CNC. Each station has assigned input and output devices.

The advantage of PROFIBUS-DP is that active devices can access the data of devices assigned as common.

The DP class 2 master has the possibility of reading the diagnostic data of all devices on the bus.

- **PROFIBUS DP communication functions**

  PROFIBUS-DP communication functions are differentiated according to the following relationships:

  - Class 1 master and DP slaves
  - Class 2 master and DP slaves
  - Class 2 master and class 1 master

  Possible communication relationships:

| **Function** | **DPM1 DP slaves** | **DPM2 DP slaves** | **DPM1 DPM2** |
|---|---|---|---|
| Parameter setting/configuration | • | • | — |
| Transmission of slave diagnostic data | • | • | — |
| Transmission of master diagnostic data | — | — | • |
| Cyclic data communications | • | • | — |
| Sync + freeze commands | • | • | — |
| Set slave address | — | • | — |
| Acyclic read of input/output images | — | • | — |
| Acyclic read/write of data | •(X) | •(X) | — |
| Alarm handling functions | •(X) | •(X) | — |
| Upload/download of master parameter records | — | — | • |

(X) These expanded PROFIBUS-DP functions are specified in technical instruction no. 2.082.

PROFIBUS-DP does not support communication between several DPM1s. If this functionality is required, PROFIBUS-FMS should be used in addition.

The expanded DP functions are mainly used for intelligent slaves with many different parameters which have to be modified during operation. The new read and write functions were defined for this purpose.

It is important that the expanded DP protocol is upwardly compatible with basic DP functions. This means that devices on which the new functions have been implemented are fully interoperable with devices lacking this expanded functionality. The only restriction is that devices which lack the expanded functions cannot execute the new functions.

- **PROFIBUS DP data transmission**
  Data transmission with PROFIBUS-DP is based on a highly efficient telegram structure.

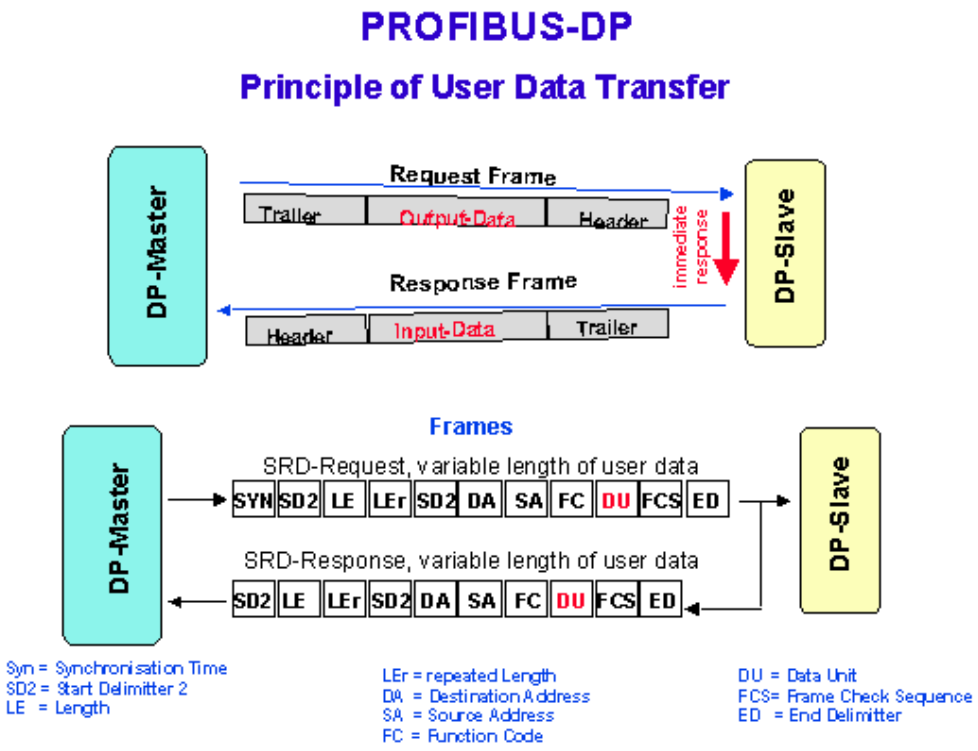



Figure 7: Principle of useful data communication with PROFIBUS-DP

The input and output data of a device are transferred in a single bus cycle. The master sends a calling telegram which contains output data for the DP slave. In its reply telegram, the DP slave transmits its input data to the master. After receipt of the calling telegram, the DP slave answers directly. Each telegram can transmit up to 244 Bytes of input or output data.

If the calling or reply telegrams have been corrupted by electromagnetic interference, the master immediately repeats the message cycle. The number of repetitions can be configured.

All PROFIBUS telegrams have high transmission security with a hamming distance of HD=4. The hamming distance is a measure of a system's error protection. This means that the higher the hamming distance, the better the system security. HD=4 meets the requirements for security-related applications.

- **PROFIBUS DP bus cycle time**
  The bus cycle time of a PROFIBUS-DP system depends mainly on the baud rate used. The baud rate is selectable between 9.6 kBit and 12 MBit, in steps.

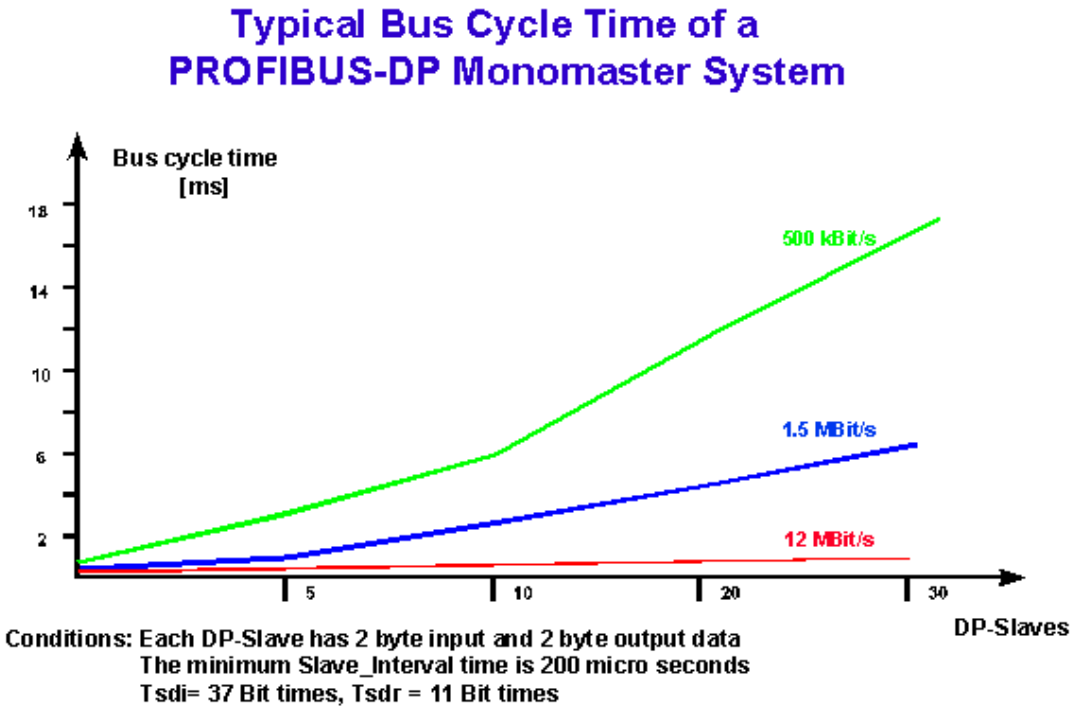



Figure 8: Bus cycle time of a PROFIBUS-DP monomaster system

Example for figure 8:
In a system configuration of 30 DP slaves working at 12 MBit and in which each slave has 2 Bytes of input and output data, a bus cycle time of approx. 1 ms is achieved.
This shows that PROFIBUS-DP can be used even in very time-critical applications.

- **Project planning with PROFIBUS DP bus cycle time**
  PROFIBUS not only describes the communication of devices with each other; it also provides a framework for open, vendor-independent project planning.

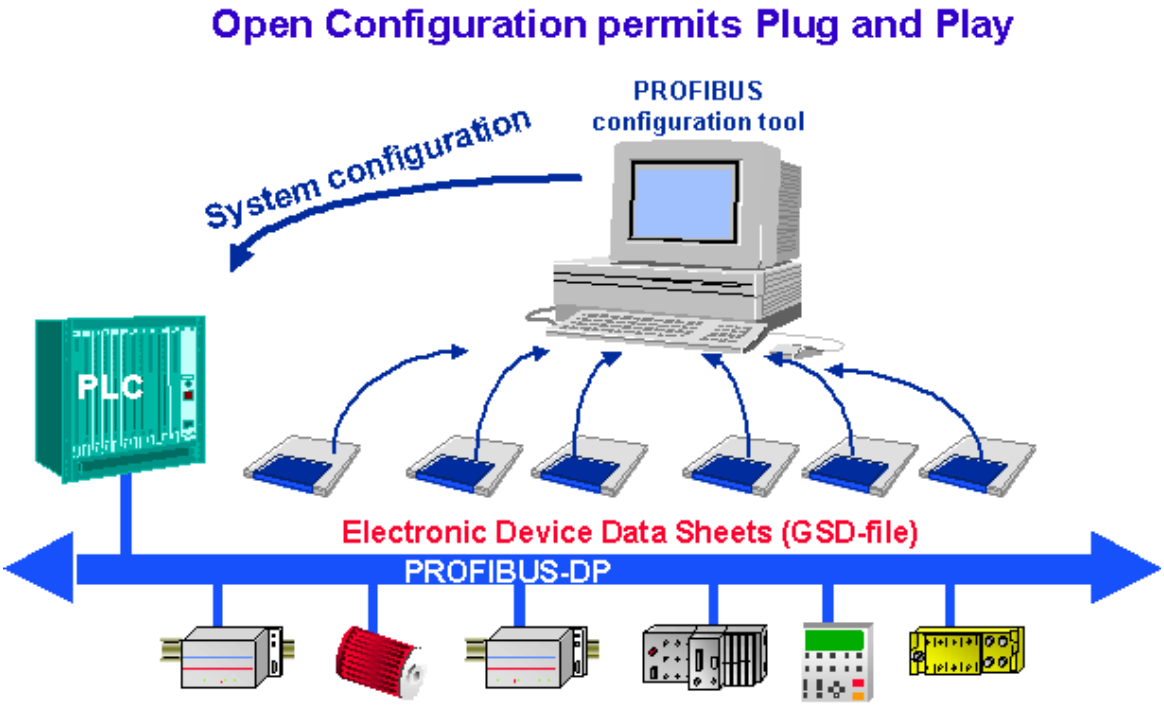



Figure 9: Project planning with PROFIBUS-DP

Project planning tools, e.g. SNET, use electronic data sheets for the devices. These electronic data sheets are called device database files or GSD files (GSD = Geräte-Stamm-Daten). These files are usually supplied on a diskette with the device. For many PROFIBUS devices, these files can also be uploaded from the PROFIBUS homepage http://www.profibus.com/.

Manufacturers describe the functionality of their devices with the help of a GSD file. The format of GSD files has a fixed definition within the PROFIBUS standard. The project planning tool reads the GSD file into its internal database and takes into consideration all the characteristics of the device in question.

This simplifies project planning and can be carried out with a vendor-independent project planning tool. PROFIBUS-DP therefore permits genuine plug-and-play solutions and reduces engineering costs.

# 2. SAIA® PCD devices for the PROFIBUS-DP network

## 2.1 PROFIBUS-DP master module

### 2.1.1 PROFIBUS-DP master card PCD7.F750

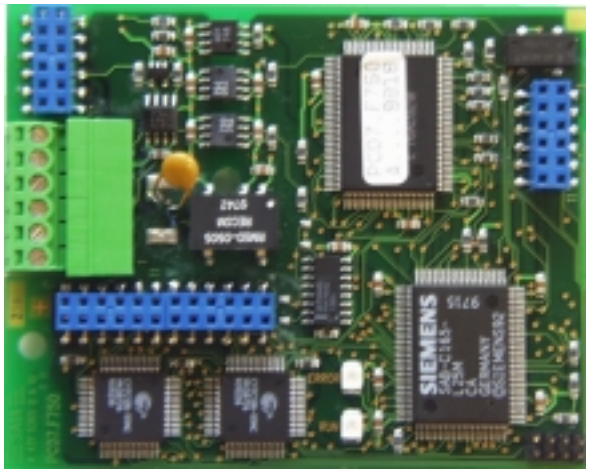


Figure 1: View of PCD7.F750

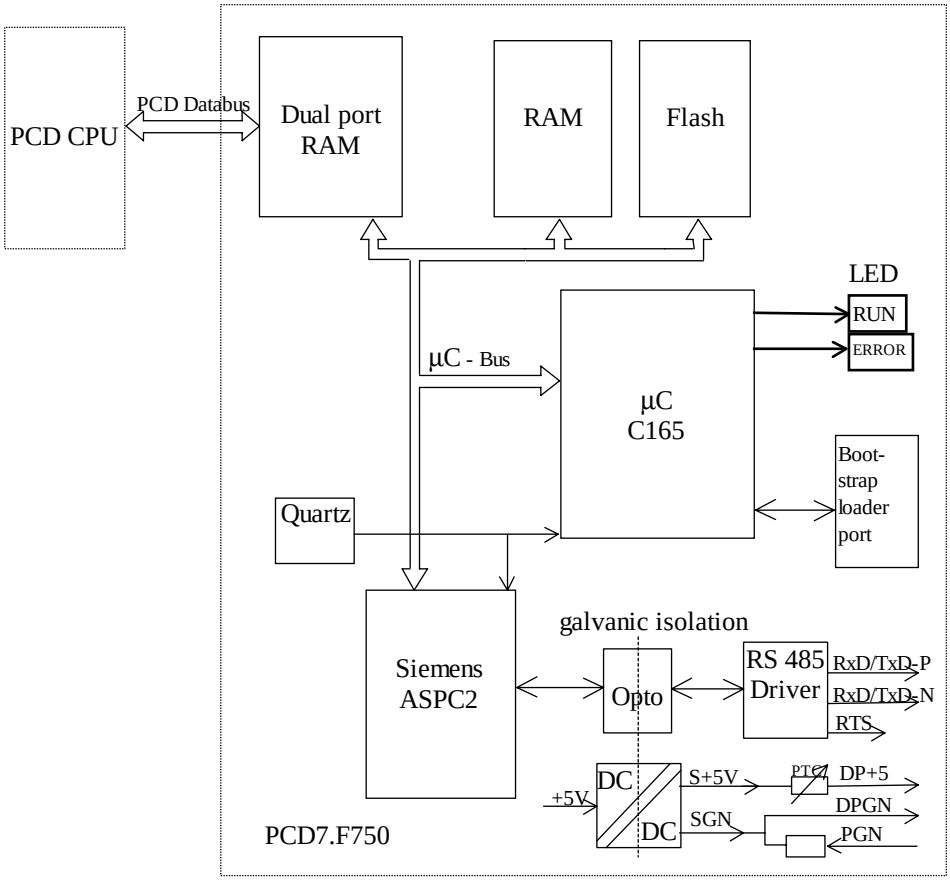



Figure 2: Block diagram PCD7.F750

### Technical data PCD7.F750

| | |
|---|---|
| Function | Profibus-DP master class 1 E (DPM1 with extension) |
| Maximum number of stations | 32 per segment / max. 126 per system (with repeaters) |
| Profibus ASIC | ASPC2 |
| Baud rate (kbit/s) | 9.6 -12000 |
| Internal current consumption from 5V bus | max. 400 mA |
| Current output DP+5V | max. 50 mA short-circuit proof with PTC |
| Galvanic isolation | between PCD-GND and PROFIBUS connection GND |

### Functions supported

| Function | Profibus-DP master class 1 E (DPM1 with extension) | |
|---|---|---|
| | Request | Response |
| Data_Exchange | X | |
| Slave_Diag | X | |
| Set_Prm | X | |
| Chk_Cfg | X | |
| Global Control | X | |
| Get_Master_Diag | | X |

### Hardware and firmware versions supporting the PCD7.F750 PROFIBUS-DP master module

| PCD System | HW | | FW PCD 1/2/6 | FW PCD7. F750 | Max. E/A Memory for DP **) | PG/Configurator |
|---|---|---|---|---|---|---|
| | Vers. | Modif. | Vers. | Vers. | Bytes | Vers. |
| PCD1.M120/M130 | C | 3 | 006 | 001 | 2942 | PG4 2.0 |
| | D | - | | | | |
| PCD2.M120/M220 | J *) | 7 | 007 | 001 | 2942 | PG4 2.0 |
| | K | - | | | | |
| PCD6.M300 | A | 234 | 002 | 001 | 2942 | PG4 2.0 |
| | B | 34 | | | | |
| | C | - | | | | |
| PCD1.M137 | A | - | 1.300 | 001 | 1024 E + 1024 A | STEP7 3.0 |
| PCD2.M127 | H | - | 1.300 | 001 | 1024 E + 1024 A | STEP7 3.0 |
| PCD2.M227 | H*) | - | 1.300 | 001 | 1024 E + 1024 A | STEP7 3.0 |

*) PCD2.M22x Version of board PCD2.M12x

**) I/O memory for DP:

This is the maximum number of bytes that can be used for I/O data exchange between the master and all slaves (number is checked in the configurator)

Example for PCD2.M120:

- 100 slaves with 29 bytes, or
- 12 slaves with 244 bytes.

**Meaning of LEDs**

The PCD7.F750 card is equipped with a Run LED and an Error LED.
The LEDs on the PCD7.F750 card are switched in the following way:

| Function | PCD7.F750 Run LED | PCD7.F750 Error LED |
|---|---|---|
| Power-up | 1 sec. on | 1 sec. on |
| then | flashing ~ 10Hz | off |
| | | |
| DP SASI instruction | flashing ~ 1Hz | 0.1 sec. on |
| then | flashing ~ 1Hz –0.01Hz | off |
| | | |
| Program running normally | flashing ~ 1Hz –0.01Hz | off |

Errors are indicated as follows:

| PCD7.F750 Run LED | PCD7.F750 Error LED |
|---|---|
| Flashing ~ 10Hz | off |
| off | On |
| Flashing ~ 1Hz | Flashing ~ 1Hz |

Possible causes of errors are:

- PCD7.F750 firmware does not match the FW of the PCD1/2/6.
- Wiring is not OK.
  (Bad cable, wires transposed, no termination)
- Too many I/O bytes used.
- Wrong configuration

Errors are stored in the PCD's history log.
To discover the precise cause of an error, the history log must be displayed.

## 2.2 PROFIBUS-DP slave module

The following PROFIBUS-DP slave modules are available:

| Module | Function |
|---|---|
| PCD7.F770 | DP slave module for PCD1 / 2 / 6.M300 |
| PCD7.F772 | DP slave module for PCD2 with port 3, RS485 |
| PCD7.F774 *) | DP slave module for PCD1 / 2<br>with port 3, RS485 and connection for PCD7.D160 terminal |
| PCD0.T770 | Profibus-DP RIO bus coupler |
| PCD0.G110 | Profibus-DP RIO with 8 I/O 24VDC |
| PCD0.G120 | Profibus-DP RIO with 16 I 24VDC |
| PCD0.G130 | Profibus-DP RIO with 16 O 24VDC |

***)** Only available as PCD7.D164 terminal set.

This set includes a plug-on ..D160 terminal with the additional RS 485 communications ports (galvanically isolated) on Port 3 and PROFIBUS-DP slave interface.

With the PCD1, Port 3 is not supported and, for the terminal, the housing cover with a recess (order number 4 104 7338 0) must also be used.

### 2.2.1 PROFIBUS-DP slave card PCD7.F77x

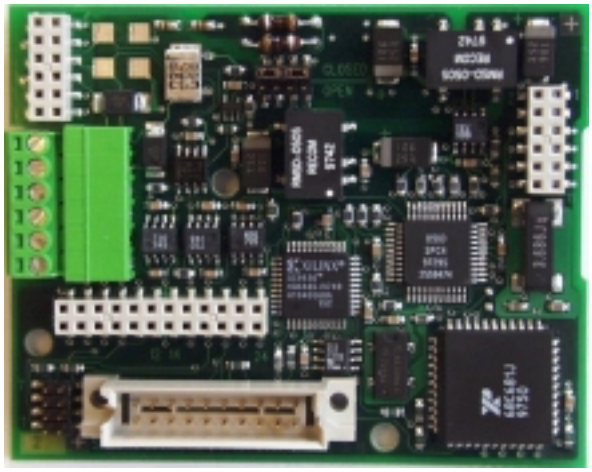


Figure 3: View of PCD7.F774

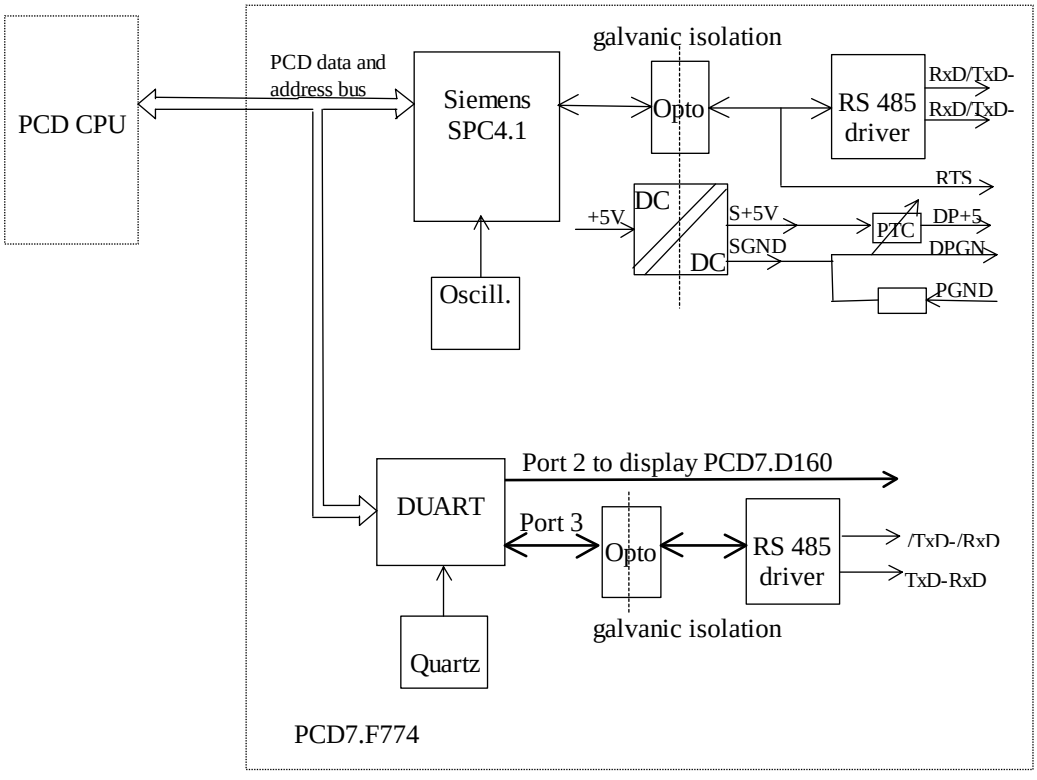



Figure 4: Block diagram PCD7.F774

### Technical data PCD7.F770 / F772 / F774

| | |
|---|---|
| Function | Profibus-DP slave E |
| Maximum number of stations | 32 per segment / max. 126 per system (with repeaters) |
| Profibus ASIC | SPC4.1 |
| Baud rate (kbit/s) | 9.6-12000 |
| Internal current consumption from 5V bus | Max. 250 mA |
| Current output DP+5V | Max. 50 mA short circuit-proof with PTC |
| Galvanic isolation | between PCD-GND and PROFIBUS GND |
| Port 2 | On PCD7.F774 for D160 display (TTL level) |
| Port 3 | On PCD7.F772 and PCD7.F774,<br>RS485 galvanically isolated from PCD-GND and PROFIBUS GND |

### Functions supported

| Function | Profibus-DP master class 1 E (DPM1 with extension) | |
|---|---|---|
| | Request | Response |
| Data_Exchange | | X |
| RD_Inp | | X |
| RD_Outp | | X |
| Slave_Diag | | X |
| Set_Prm | | X |
| Chk_Cfg | | X |
| Get_Cfg | | X |
| Global Control | | X |

### Hardware and firmware versions supporting the PCD7.F77x PROFIBUS-DP slave module

| PCD System | HW | | FW PCD 1/2/6 | Max. E/A Memory for DP **) | PG/Configurator |
|---|---|---|---|---|---|
| | Vers. | Modif. | Vers. | Bytes | Vers. |
| PCD1.M120/M130 | C | 3 | 006 | 244 E + 244 A | PG4 2.0 |
| | D | - | | | |
| PCD2.M120/M220 | J*) | 7 | 007 | 244 E + 244 A | PG4 2.0 |
| | K | - | | | |
| PCD6.M300 | A | 234 | 002 | 244 E + 244 A | PG4 2.0 |
| | B | 34 | | | |
| | C | - | | | |
| PCD1.M137 | A | - | 1.300 | 122 E + 122 A | STEP7 3.0 |
| PCD2.M127 | H | - | 1.300 | 122 E + 122 A | STEP7 3.0 |
| PCD2.M227 | H*) | - | 1.300 | 122 E + 122 A | STEP7 3.0 |

*) Version of board PCD2.M12x

*) I/O memory for DP:

This is the maximum number of bytes that can be used for the exchange of data between the master and slave. (Number is checked by the configurator.)

This value is dependent upon:

Total I/O data, number of modules, number of diagnostic bytes, etc.

**Supported PROFIBUS-DP diagnostics of PCD7.F77x slave modules**

PCD7.F77x slave modules support the 6 bytes of standard PROFIBUS-DP diagnostics (Base+0 to Base+5)
In addition, byte 7 (Base+6) of the expanded PROFIBUS-DP diagnostics is sent in the following cases:

- CPU in STOP
- CPU in HALT

Here, byte 7 (Base+6) stores the following information:

| CPU Status | Transmission of byte 7 (Base+6) | ASCII character in byte 7 |
|---|---|---|
| RUN | no | No character |
| STOP | yes | ‘S‘ (0053h) |
| HALT | yes | ‘H‘ (0048h) |

**GSD files for PROFIBUS-DP SAIA slaves**

| Slave | GSD file |
|---|---|
| PCD0.T770 | Saia1631.gsd |
| PCD0.G110 | Saia1635.gsd |
| PCD0.G120 | Saia1634.gsd |
| PCD0.G130 | Saia1633.gsd |
| PCD1.M120<br>PCD1.M120 | Saiacd10.gsd |
| PCD2.M120<br>PCD2.M220 | Saiacd20.gsd |
| PCD6.M300 | Saiacd60.gsd |

### 2.2.2 PROFIBUS-DP RIO modules

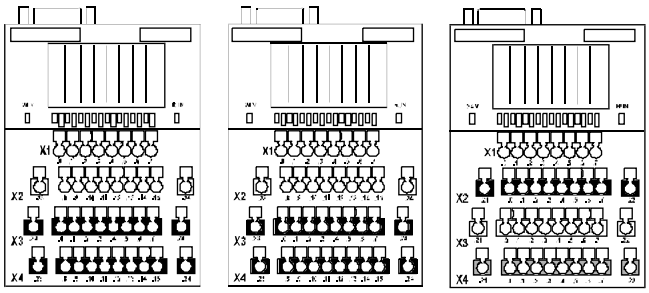


Figure 5: View of PCD0.G120, G130, G110 RIO compact modules

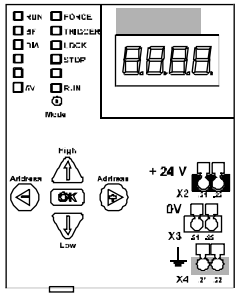


Figure 6: View of PCD0.T770 RIO bus coupler for modular system

**Technical data PCD0.T770 and PCD0.G1x0**

| | |
|---|---|
| Function | PROFIBUS-DP Slave E |
| Maximum number of stations | 32 per segment/max. 126 per system (with repeaters) |
| Profibus ASIC | Modular: SPC3, compact: LSPM2 |
| Baud rate (kbit/s) | 9.6-12000 |
| Max. number of I/Os per slave | Modular: 96, compact: 16 |
| Max. number of I/O modules per slave | Modular: 6, compact: 0 |

**Functions supported**

| Function | PROFIBUS-DP master class 1 E (DPM1 with extension) | |
|---|---|---|
| | Request | Response |
| Data_Exchange | | X |
| RD_Inp | | X |
| RD_Outp | | X |
| Slave_Diag | | X |
| Set_Prm | | X |
| Chk_Cfg | | X |
| Get_Cfg | | X |
| Global Control | | X |

Further details can be obtained from PCD0 manual 26/766.

# 2.3 Connection of PROFIBUS-DP modules

### 2.3.1 PCD7.F7xx card

The PCD7.F7xx module can be inserted in the following PCD controllers:

| | Space | F750 Master | F770 Slave | F772 Slave + Port3 RS 485 | F774 Slave + Port3 RS 485 + Display |
|---|---|---|---|---|---|
| PCD1.M120 / M130 / M137 | B | X | X | | X *) |
| PCD2.M120 / M127 | B | X | X | X | X *) |
| PCD2.M220 / M227 | B | X | X | X | |
| PCD6.M300 | 3B | X | X | | |

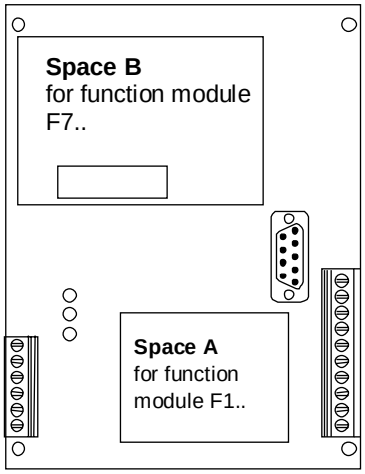
***)** Only available as a PCD7.D164 terminal set.

This set includes a ..D160 plug-on terminal with the additional RS 485 communications interfaces (galvanically isolated) on port 3 and PROFIBUS-DP slave interface.

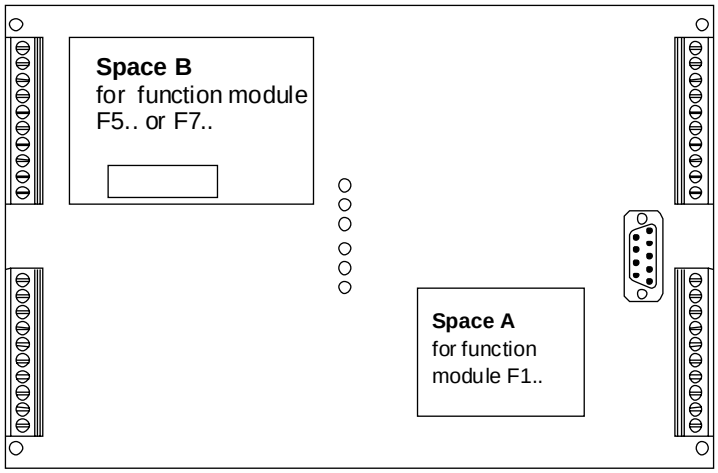
With the PCD1, Port 3 is not supported and, for the terminal, the housing cover with a recess (order number 4 104 7338 0) must also be used.

**Location of space:**

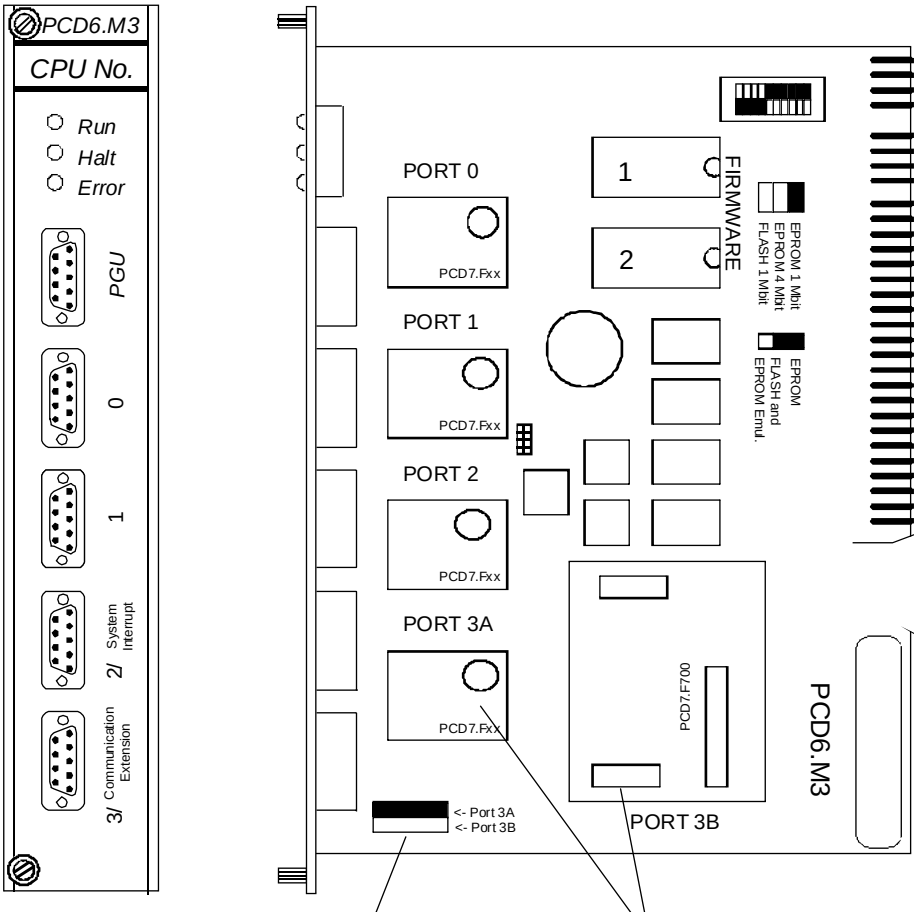
Space B on PCD1:



Space B on PCD2:




Space 3B on PCD6.M3:




Jumper for port 3A or 3B

Interface module for port 3 selectable as PROFIBUS (Port 3B) or standard (Port 3A)

**PROFIBUS-DP connection:**

For the PCD1 and PCD2, PROFIBUS-DP connection is achieved via the 6-pole connector on the PCD7.F7xx module.
With the PCD6.M3, connection is via the 9-pole, D-type connector on port 3.
With the PCD0, connection is also via the 9-pole, D-type connector.

**PCD1 / PCD2 connection:**

Connection must be made directly to the PCD7 module's 6-pole connector.
Please also note the installation instructions given in chapter 3.

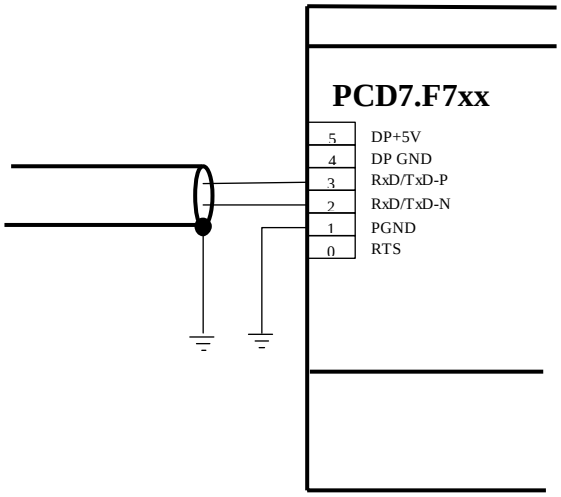



Figure 7: PCD1 / PCD2 connection

**PCD6.M3 connection:**

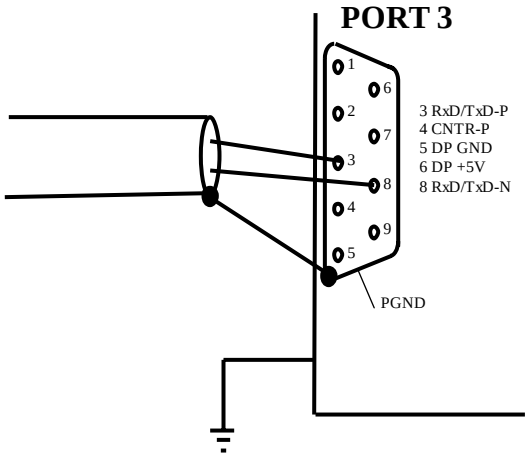



Figure 8: PCD6.M3 connection

Jumper for port 3 must be set to 3B.

### 2.3.2 PCD0.Xxxx card

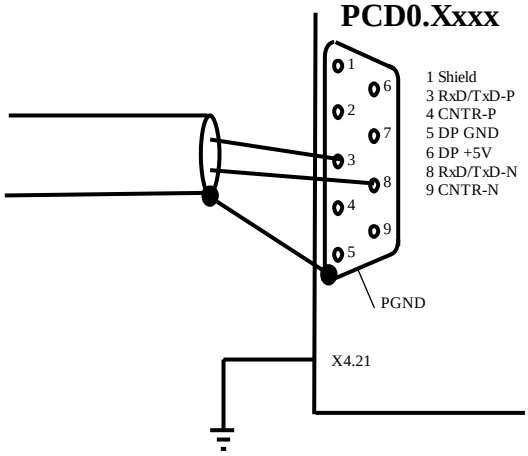



Figure 9: PCD0 connection

### 2.3.3 Meaning of connections

| Signal | Meaning | PCD7. F7xx connec-tion | PCD6. M3xx connec-tion | PCD0. Xxxx connec-tion | Standard A-B connec-tion | Standard cable green / red |
|---|---|---|---|---|---|---|
| CNTR-P / RTS | Control signal for repeater | 0 | 4 | 4 | | |
| PGND | Screen / pro-tective earth | 1 | housing | 1 | | |
| RxD/TxD-N | Receive / transmit data, minus | 2 | 8 | 8 | A | green |
| RxD/TxD-P | Receive / transmit data, plus | 3 | 3 | 3 | B | red |
| DP GND | Ground for DP +5V | 4 | 5 | 5 | | |
| DP +5V | 5V supply for line termina-tion resistors | 5 | 6 | 6 | | |
| CNTR-N | Control signal for repeater | | | 9 | | |

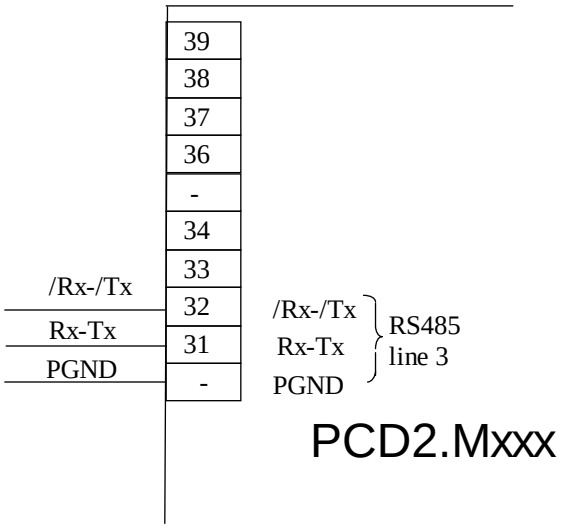
## 2.4 Ports 2 and 3 of PCD7.F77x slave modules

### 2.4.1 Port 2

Port 2 is connected directly to the PCD7.D160 plug-on terminal. Further information can be found in the PCD7.D160 manual (order number 26/753).

### 2.4.2 Port 3

Port 3 can be employed as a user-definable RS485 communications port. This port can only be used on the PCD2. Connection is via the 10-pole connector:



## 2.5 Repeater PCD7.T100

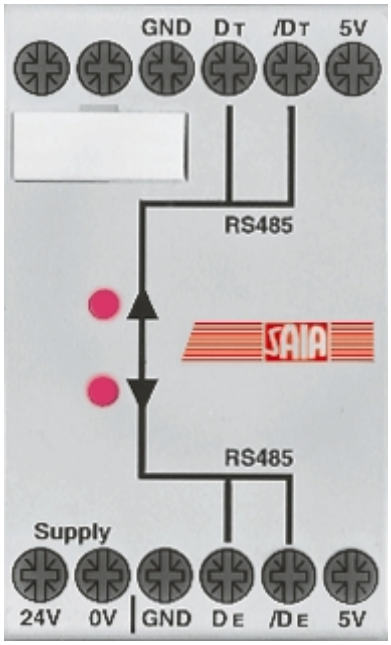



Figure 10: Repeater PCD7.T100

The repeater is used to decouple an RS 485 communications network. This serves two purposes: the reprocessing of transmission signals and the galvanic isolation of individual sections of the line.

The reprocessing of signals is necessary when

- The overall length of the bus cable exceeds the maximum length allowed.
- More than 31 stations participate in the network.

Galvanic isolation is necessary when

- The reference ground in the area of the installation demonstrates excessive potential differences.
- The reference ground of an installation is subject to excessive stray current.

A maximum of 3 PCD7.T100 repeater can be switched in series.
The PCD7.T100 repeater can be used within the operative range of **110 Baud to 500 kBaud**.
Operative ranges to 12 MBaud on request.

Detailed information can be found in the manual: “Installation components for RS 485 networks” (order number 26/740E).

## 2.6 Termination box PCD7.T160


Figure 11: Termination box PCD7.T160

For confidence and ease of maintenance when using an RS485 network it is important to install the line termination resistors separately and accessibly. The termination box serves this purpose whereby, depending on the physical characteristics of the line, a suitable line termination resistor can be selected by means of 2 jumpers. Signal lines D and /D are biased to the required off-load voltage with an electrically isolated supply.

The operating range of the PCD7.T160 termination box extends to a baud rate of 12 MBaud.

Detailed information can be found in the manual: “Installation components for RS 485 networks” (order number 26/740E).

Notes

# 3. Planning and installation of a PROFIBUS-DP network

Creating a PROFIBUS-DP network can be roughly divided into the following steps:

a) Planning and installation of the network

b) Network definition and configuration with the configurator

c) Writing the user program

d) Commissioning

This chapter does not describe any details. It is only intended as a guide and to provide ideas for the procedure to follow when planning and installing a PROFIBUS-DP network.

Detailed technical information can be found in the manual "Installation components for RS 485 networks" (order ref. 26/740E) and in the later chapters of this manual.

## 3.1 Planning and installation of the physical layer

- Construction of a layout diagram
- Defining max. network distance
- Defining max. segment distance
- Defining cable type
- Defining max. baud rate
- Possible provision of repeaters
- Defining the first and last stations

→ Provide PCD7.T160 termination boxes

→ For the correct installation of PROFIBUS, the instructions in the manual "Installation components for RS 485 networks" should be strictly adhered to.

### 3.1.1 Line parameters

The PROFIBUS-DP bus line is specified in EN 50170 as line type A.

| Parameter | Line A |
|---|---|
| Characteristic impedance (Ω) | 135…165 |
| Capacitance per unit length (pF/m) | < 30 |
| Loop resistance (Ω/km) | 110 |
| Wire diameter (mm) | 0.64 |
| Wire cross section ($mm^2$) | >0.34 |

Possible cable supplier:
Volland AG, Rümlang, Switzerland, offering the following cable types:

- Cable for static installation:
  Unitronic bus L2/F.I.P. Volland ref. 2170221

- Cable for highly flexible installation:
  Unitronic bus FD P L2/F.I.P Volland ref. 2170222

### 3.1.2 Connection of network stations

To prevent reflections at the ends of the line, each segment must be terminated at the physical extremities of its line. This means that lines are biased at the equilibrium rest potential.
According to the PROFIBUS-DP standard, this must not be done directly on PROFIBUS-DP devices, but must be achieved by means of external components.
Both the PCD7.T160 termination box and commercially available 9-pole PROFIBUS-DP D-type connectors are suitable for this purpose.

Network termination must therefore be done as follows:

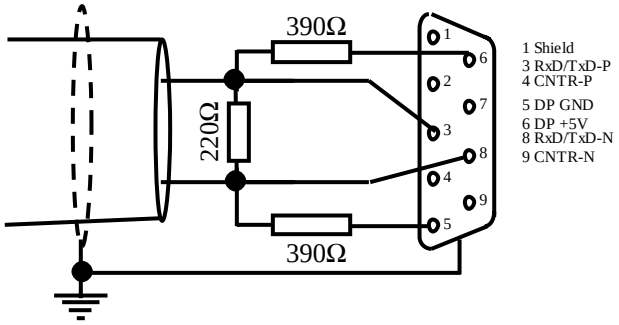
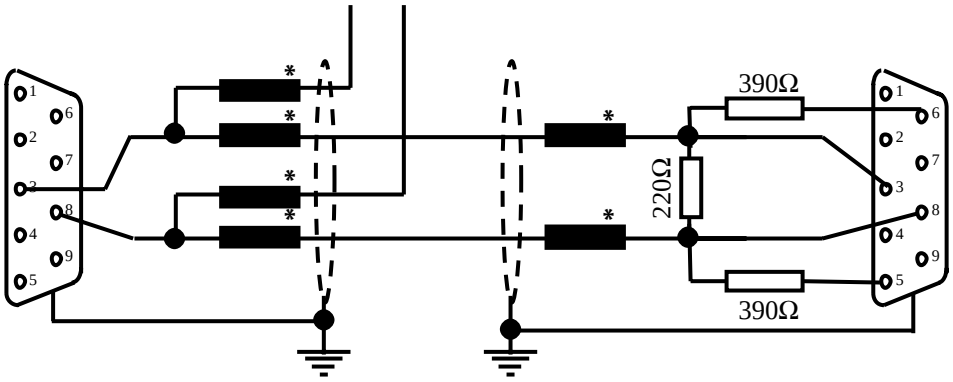



Figure 1: Network termination

For baud rates > 1.5 MBaud the following combination of plug connectors should used at all stations in the network, because of their capacitive load:




* = Series inductance of 110 nH

Figure 2: Use of series inductance

Possible supplier for 9-pole PROFIBUS-DP D-type connectors to hook up PCD controllers to the PROFIBUS-DP network:
ERNI Elektrotechnik AG, Brüttisellen, Switzerland:

- ER*bic* junction, horizontal grey: Erni ref. 103648
  (junction equipped with series inductance of 110 nH)

- ER*bic* junction, horizontal, grey with PG connector: Erni ref. 103663
  (junction equipped with series inductance of 110 nH)

- ER*bic* termination, horizontal yellow: Erni ref. 103649
  (junction equipped with series inductance of 110 nH plus termination resistors of 390Ω and 220Ω)

Figure 3: ERNI ER*bic* connector

If a 9-pole D-type connector is to be used to connect the PCD1 or PCD2 to PROFIBUS-DP, the following adapter can be used (D-type 9-pole to terminals):

Possible supplier of D-type 9-pole to terminal adapter:
Phoenix Contact AG, CH-8317 Tagelswangen, Switzerland:

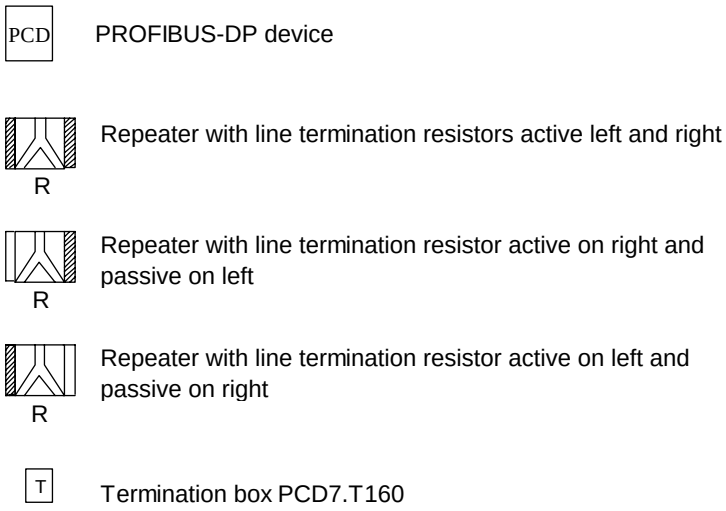
- VARIOFACE module with spring connection D-type, 9-pole, socket board: Phoenix Contact ref. 2293666


Figure 4: VARIOFACE module

### 3.1.3 Stub cables

The stubs are those cables leading from the PROFIBUS-DP network cable to PROFIBUS-DP devices.

Meaning of the symbols in the following drawings:



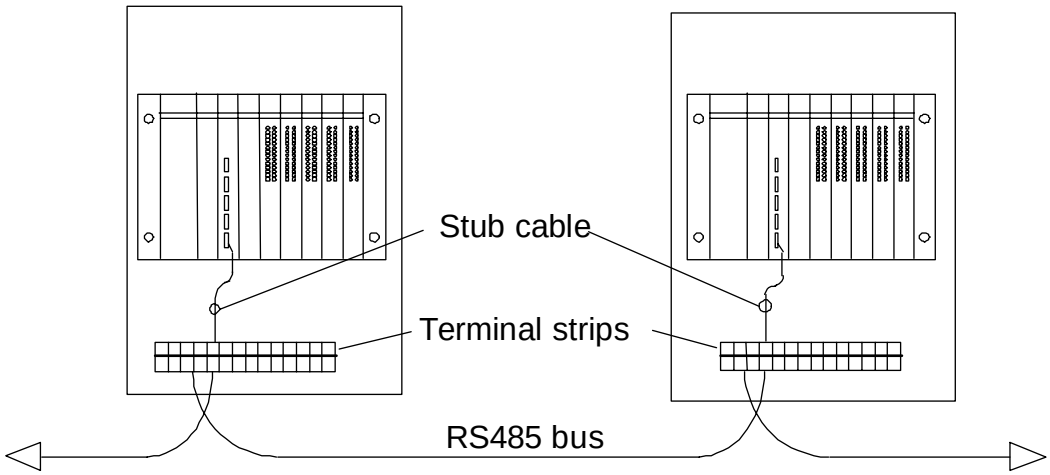




Figure 5: Stub cables

Admissible lengths for stub cable are described in the next section.

If a network in which stub cables are not allowed should, despite this fact, still require quite a long stub, this can be achieved as follows.

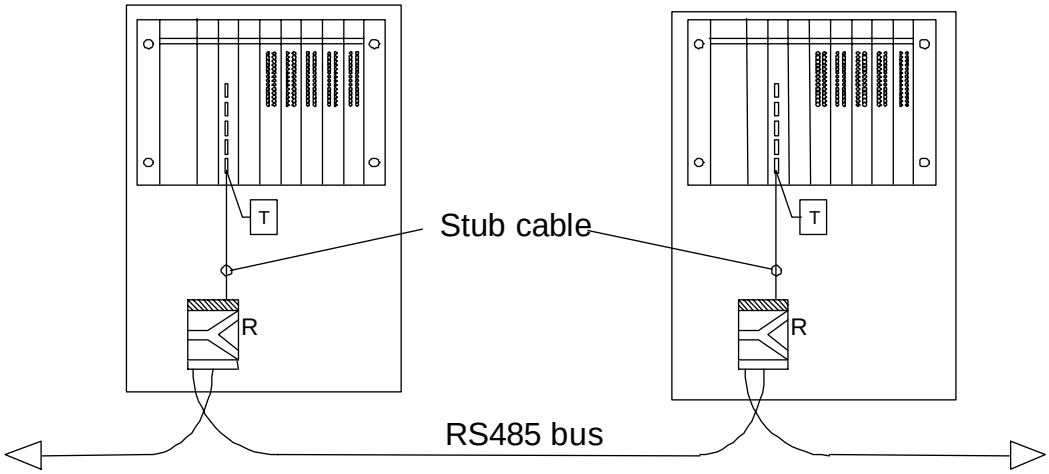



Figure 6: Stub cables with repeaters

### 3.1.4 Network layout

A maximum of 126 member stations is allowed per network.
A segment comprises a maximum of 32 members.
(Repeaters count as members.)

The overall length of cable and overall stub length depend on the bit rate.

| Bit rate Kbit/S | Linear extent per segment in metres | Length of stub cable. Total per segment in metres |
|---|---|---|
| 9.6 | 1200 | 6.6 |
| 19.2 | 1200 | 6.6 |
| 93.75 | 1200 | 6.6 |
| 187.5 | 1000 | 6.6 |
| 500 | 400 | 6.6 |
| 1500 | 200 | 6.6 |
| 3000 | 100 | 0 |
| 6000 | 100 | 0 |
| 12000 | 100 | 0 |

Networks must not be branched without special measures. By using repeaters, the following network structures are possible:

**Linear structure**

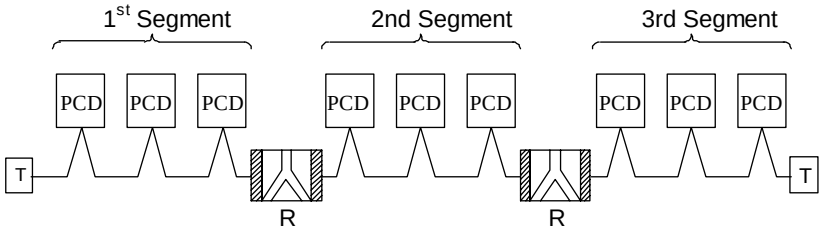



Figure 7: Network with linear structure

**Star structure**

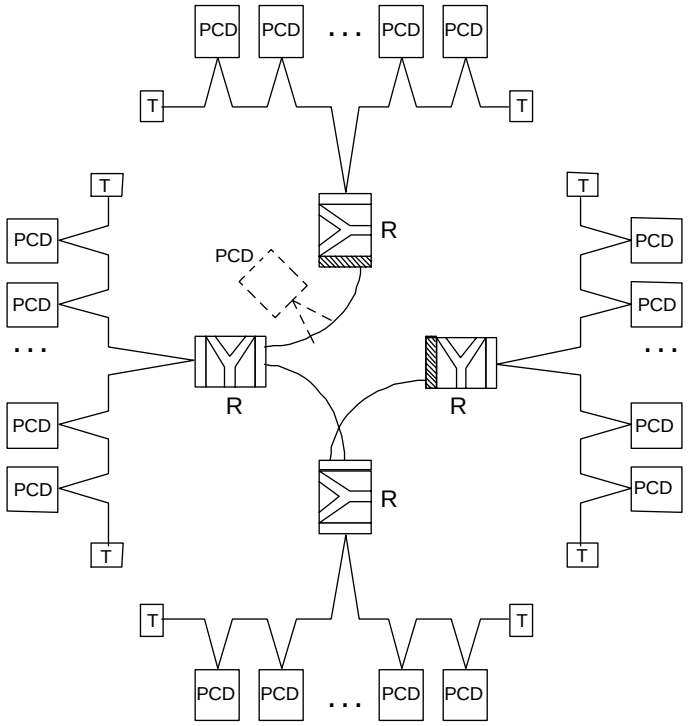



Figure 8: Network with star structure

**Tree structure**

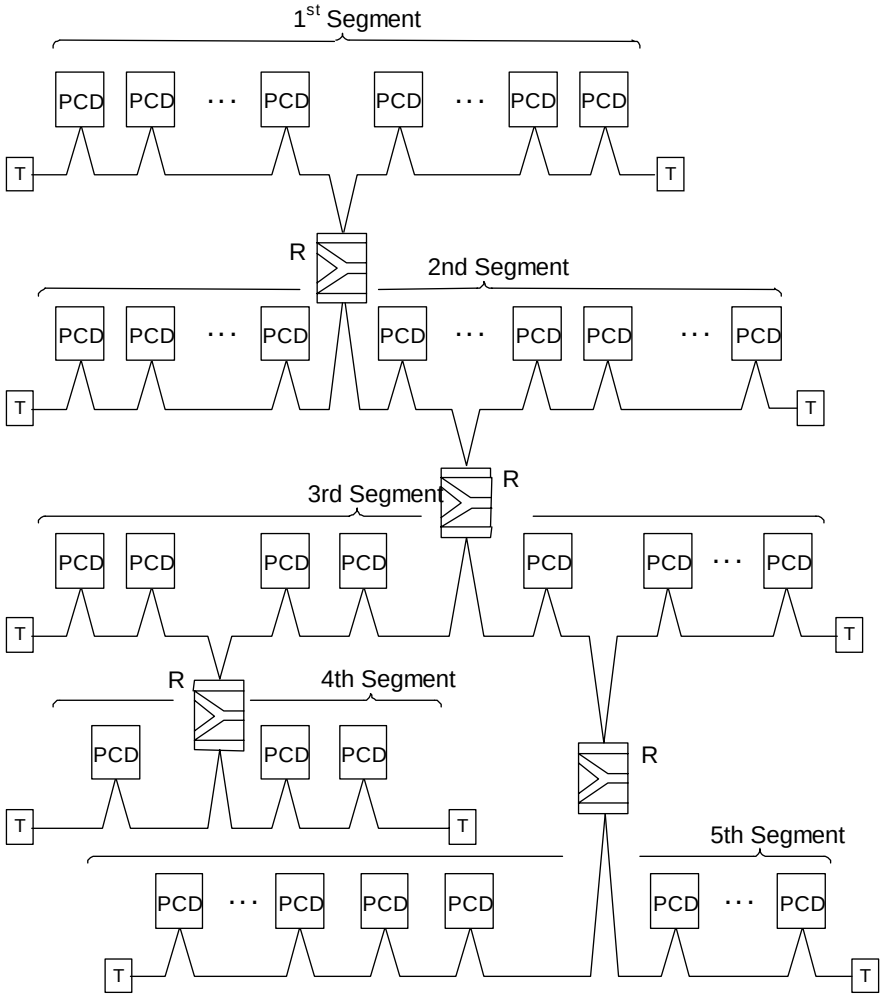



Figure 9: Network with tree structure

## 3.2 Logical network structure

### 3.2.1 Definition of bus parameters

- Baud rate
- Possible formation of groups for Global Control Service functions

| | |
|---|---|
| **Important:** | **Bus parameters must be identical for all bus members.**<br>**With mixed FMS / DP networks, the same baud rate must apply to both protocols.** |

### 3.2.2 Configuration of slaves

After describing its physical construction, slave controllers must now be integrated into the network and configured.
Slave inputs and outputs must then be assigned to the master's media (mapping).

- Load GSD files from external systems into the configurator
- Definition of network with master and all slaves
- Configuration of slaves
- Mapping of slave I/Os to master media.
- The maximum number of I/O bytes reserved in the master must not be exceeded

### 3.2.3 Checking and assessing the performance values required

- Definition of reaction times
- Definition of supply system loading

# 4. The PROFIBUS-DP configurator

The definition and configuration (bus parameters, network stations and variable definition) of a PROFIBUS-DP network can be very extensive, depending on the size of the project. This task is made significantly easier for the user by the PROFIBUS-DP configurator.

Files generated by the PROFIBUS-DP configurator can be used to configure SAIA devices (masters or slaves).

# 4.1 General

The PROFIBUS-DP configurator consists of software that runs under MS-Windows 9x/NT and higher. No special hardware is required. Windows technology is used throughout. This provides a good overview and makes parameter entry very user friendly.

For ease of use only the important parameters appear in the main windows. To reach the advanced parameters, most windows have an "advanced setup" button.

Default values are offered wherever the setting of parameters allows this. These default values can also be altered.

A value range is specified for each parameter.

The configurator ultimately creates an ASCII text file containing the definition texts for all PROFIBUS-DP connections in a station. This text file is used in the SASI instruction of the PROFIBUS channel.

As documentation, a list is generated containing all the parameters. It shows the precise configuration of the station in the PROFIBUS network. It is also possible to derive a list of all slave elements assigned in the master from this summary.

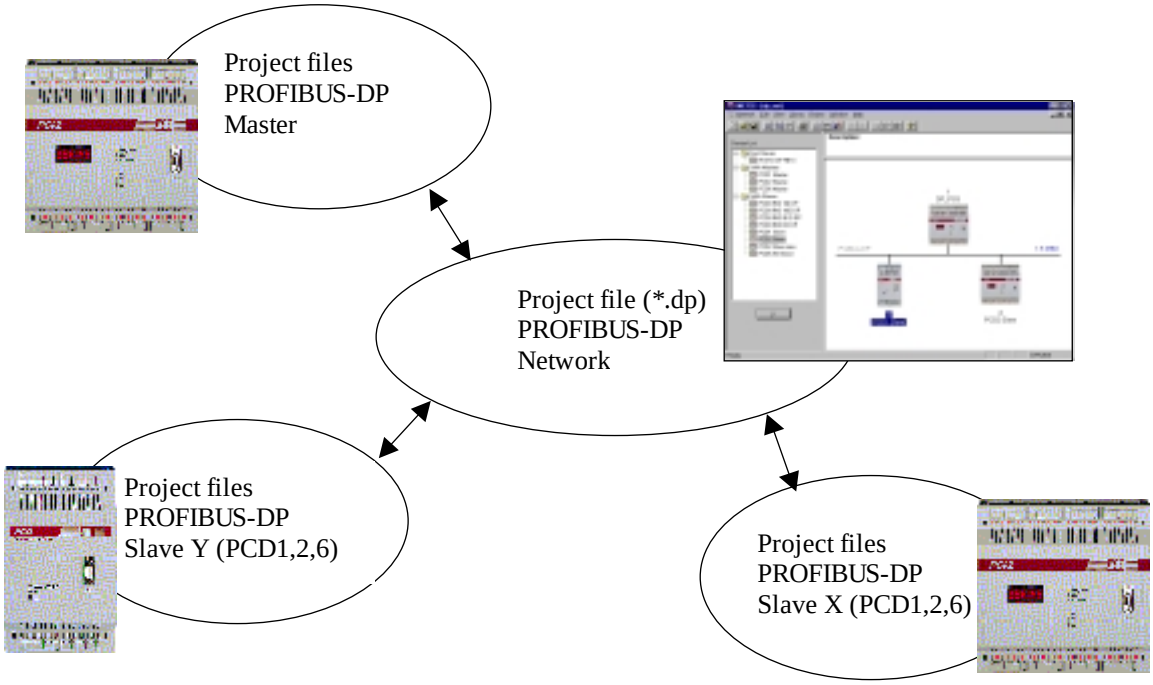
**Data structure generated by PROFIBUS-DP configurator**

The PROFIBUS-DP configurator will set up a *.def and *.src file for each SAIA master or SAIA slave (applies only to PCD1, 2, 6) contained in the PROFIBUS-DP network.

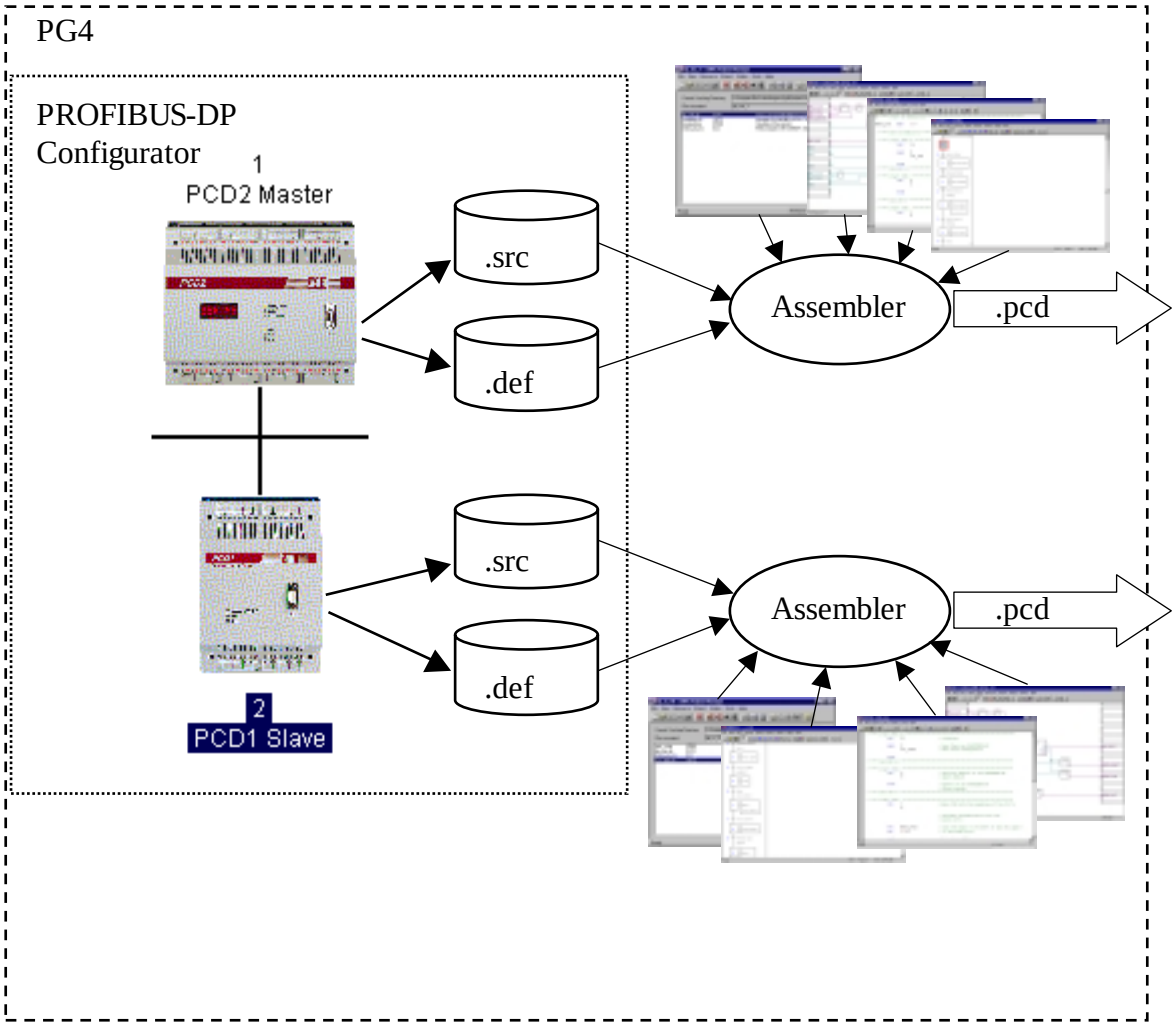
These files are then automatically linked to the user program by the PG4 programming software.

It is then possible to use any of the PG4 editors (IL, Graftec, Fupla or Kopla) to access the media defined in the configurator.
User programs previously written with PG3 programming software can be imported into the PG4 project for further processing there.

Summary of project structure:




Summary of files generated:



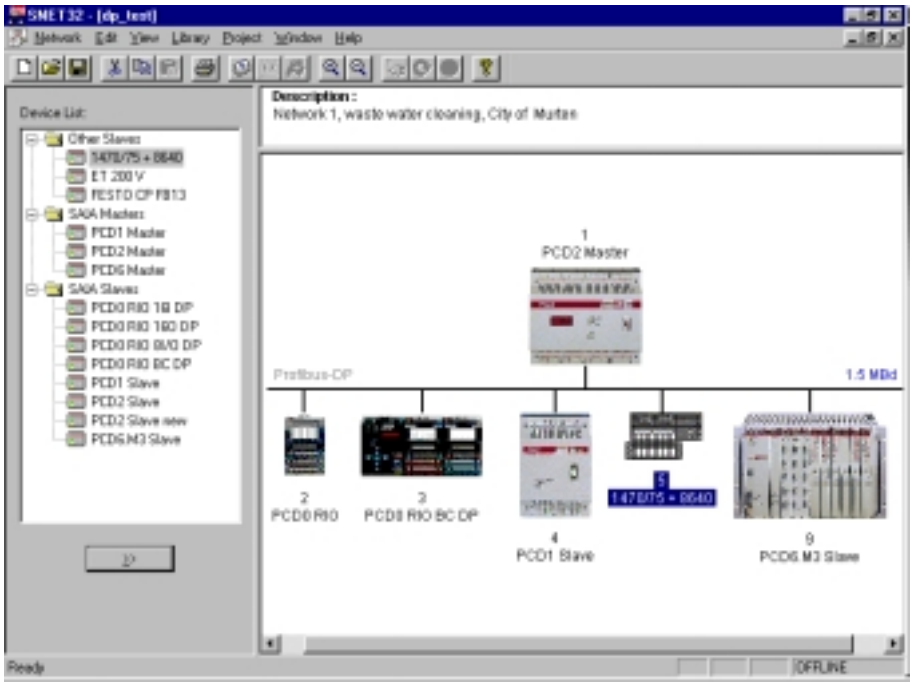
## 4.2 Procedure for PROFIBUS-DP configuration

The procedure can be divided into the following steps:

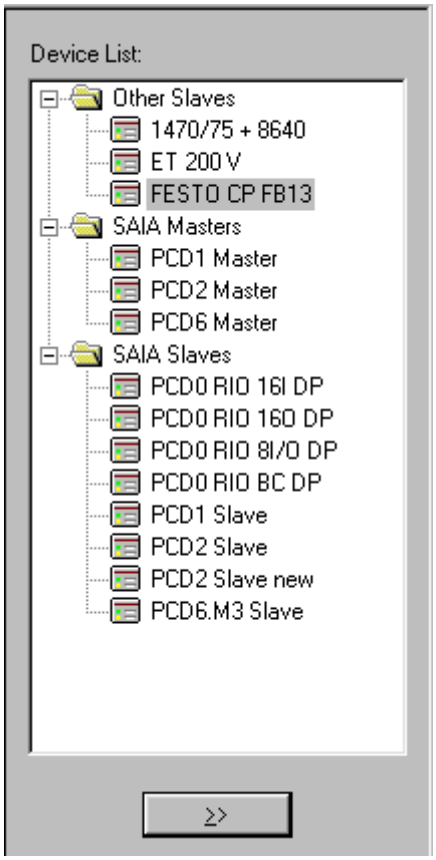
1. Start PG4
2. Open a PROFIBUS-DP project from the Project Manager
3. Configuration of the network
4. Definition of bus parameters
5. Definition of slave hardware
6. Assignment of slave I/Os to master media
7. Save the configuration
8. Generation of SASI texts for the PCD stations (Build Project)
9. Generation of documentation

## 4.3 Description of PROFIBUS-DP configurator menu

### 4.3.1 Main screen

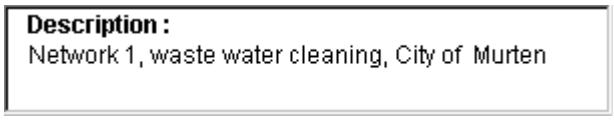



Screen description:
On the left, the 'Device List' window shows PROFIBUS-DP devices that are present in the library.

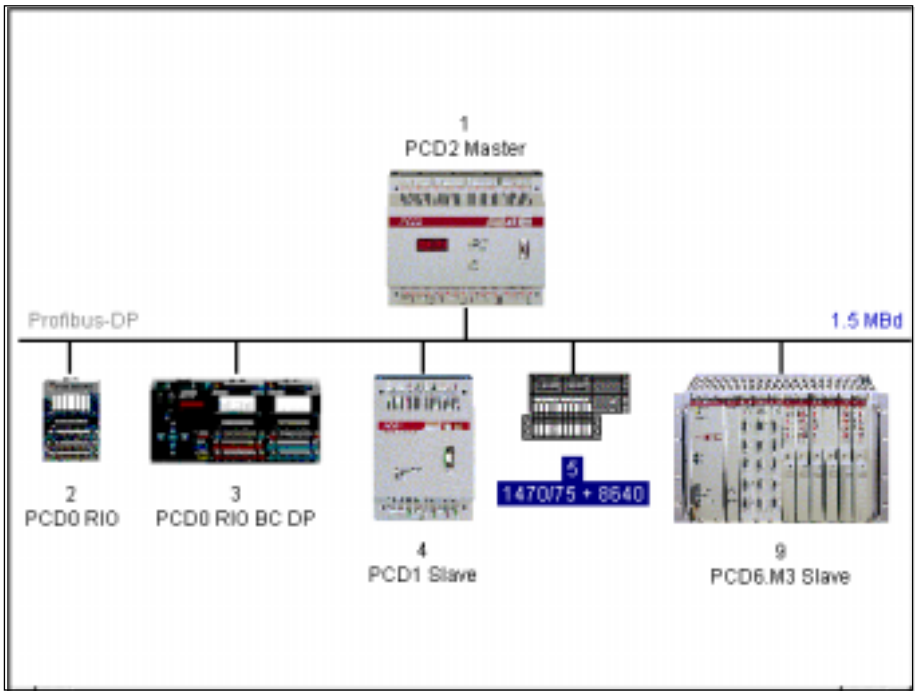


On the right, the 'Description:' window gives a description of the PROFIBUS-DP network.



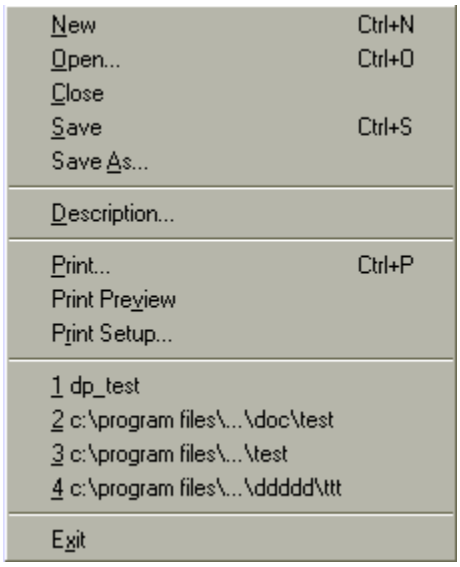

In the lower right-hand window the PROFIBUS-DP network can be seen. In this window all PROFIBUS-DP devices present in the network have been inserted graphically.




The following submenus can be selected:


- Network
- Edit
- View
- Library
- Project
- Window
- Help

### 4.3.2 'Network' submenu




**'New':** Open a new project. It is possible to choose here between a PROFIBUS-DP, SRIO or LON network.
Corresponds to this toolbar icon:

**'Open':** Open an existing project.
Corresponds to this toolbar icon:

**'Close:** Close active project.

**'Save':** Save active project under its current name.
Corresponds to this toolbar icon:

**'Save as...':** Save active project under a new name.

**'Description...':** Description of project. This description can be seen on the main screen, top right window.

**'Print...':** Print the configuration parameters of a project. It is also possible to print the parameters to an ASCII file.

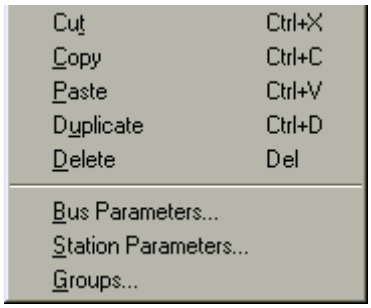
'**Print Preview':** Creates an on-screen preview of the printout. This displays all devices used, their settings and the appropriate media.

'**Print Setup...':** Definitions of printer type and paper format.

**1 .. 4:** Displays list of the last 4 projects processed.

**Exit:** Exit SNET.

### 4.3.3 'Edit' submenu




**'Cut':** Cut out and save a selected slave or master device to the clipboard. This copies the entire device configuration, i.e. installed modules and assigned media are also copied.
Corresponds to this toolbar icon:

**'Copy':** Copy a selected slave or master device to the clipboard. This copies the entire device configuration, i.e. installed modules and assigned media are also copied.
Corresponds to this toolbar icon:

**'Paste':** Insert a slave or master device currently stored in the clipboard into the active project. This copies the entire device configuration, i.e. installed modules and assigned media are also copied.
Corresponds to this following toolbar icon:

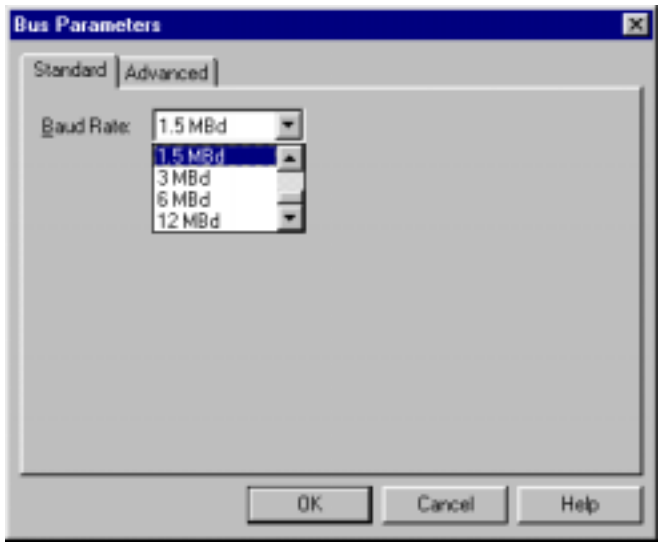
**'Duplicate':** Create a copy of a selected slave device.
This copies the entire configuration of the selected device, i.e. installed modules and assigned media are copied.

**'Delete':** Delete a selected slave or master device.

**'Bus Parameters...':**

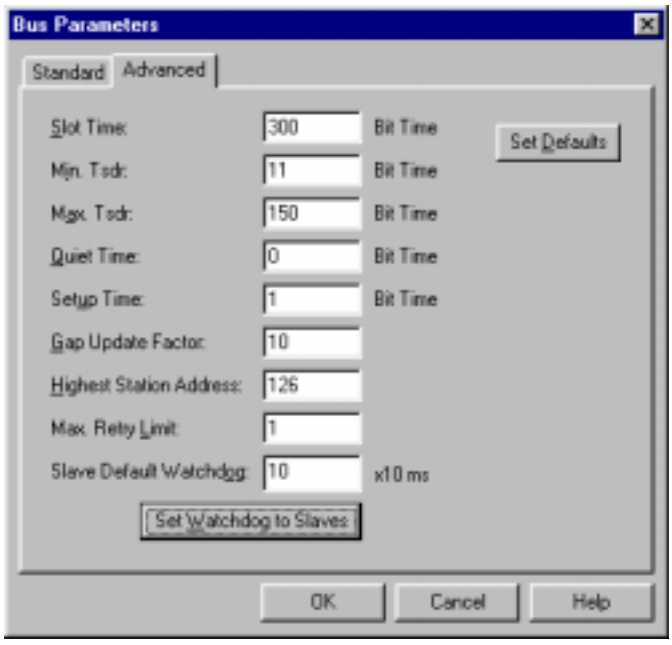
Definition of network speed and bus timeout.




The following baud rates are possible:

9.6 kBd, 19.2 kBd, 93.75 kBd, 187.5 kBd, 500 kBd, 1.5 MBd, 3 MBd, 6 MBd or 12 MBd.

Note:
If a speed is selected that is not available on all network devices, it is automatically changed to the highest possible speed supported by all devices.



| | |
|---|---|
| Slot Time: | Maximum period during which the transmitter of a data or token telegram awaits the response. The master waits for a call telegram to be answered or acknowledged until the 'Slot Time' has run out. This parameter is purely a monitoring time and therefore does not affect data throughput.<br>Range of values: 52 .. 65535 bit periods |
| Min. Tsdr: | Minimum waiting time for a slave between receiving a request telegram and sending the answer to the master. This parameter is essentially determined by the processing speed of the connection (interpreting the call, putting together a response/acknowledgement) and the length of the telegram. This parameter can be used to delay transmission of the response/acknowledgement and depends on the slowest master (telegram initiator). Only after this time is the latter ready to receive an acknowledgement.<br>Range of values: 1 .. 65535 bit periods |
| Max. Tsdr: | Maximum waiting time for a slave between receiving a request telegram and sending the answer to the master. This parameter is essentially determined by the processing speed of the connection (interpreting the call, putting together a response/acknowledgement) and the length of the telegram.<br>Range of values: 1 .. 65535 bit periods |
| Quiet Time: | Length of time a transmitter waits between sending the end of a telegram and switching on the receiver. This parameter can be used to take into account repeater transit times or modulator fade times.<br>Range of values: 0 .. 255 bit periods |
| Setup Time: | This parameter defines the time that elapses between receipt of an event (e.g. last character of telegram sent) and execution of the required reaction (e.g. enabling the receiver).<br>Range of values: 0 .. 255 bit periods |
| Gap Update Factor: | Number of token passages between two gap update cycles.<br>Range of values: 1 .. 100 |
| Highest Station Address: | Highest station address present in the network.<br>Range of values: 2 .. 126 |

Max. Retry Limit: Number of times a telegram is repeated when no ACK is received before NAK is registered. In very noisy environments, this parameter can be used to achieve safe operation.
Range of values: 1 .. 8

Slave Default Watchdog: Default watchdog time when a slave is integrated into the network.
Range of values: 0 .. 65025 x 10ms

Set Watchdog to Slaves: Copies the time defined in the 'Slave Default Watchdog' to all slaves. This function serves to ensure that all slaves have the same watchdog time.

Set Defaults: Set baud rate-dependent default values.
Default values are as follows:

| Baud rate | 9.6 | 19.2 | 93.75 | 187.5 | 500 | 1500 | 3000 | 6000 | 12000 |
|---|---|---|---|---|---|---|---|---|---|
| Slot Time | 100 | 100 | 100 | 100 | 200 | 300 | 400 | 600 | 1000 |
| Min. Tsdr | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| Max. Tsdr | 60 | 60 | 60 | 60 | 100 | 150 | 250 | 450 | 800 |
| Quiet Time | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 6 | 9 |
| Setup Time | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 8 | 16 |
| GAP | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Highest Station | 126 | 126 | 126 | 126 | 126 | 126 | 126 | 126 | 126 |
| Max. Retry | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 4 |

**Station Parameters:**

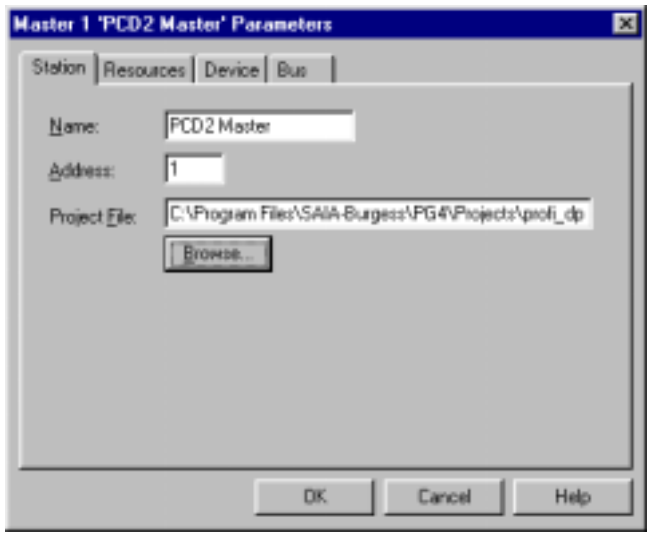
For editing the station parameters of a chosen slave or master device. Different dialog boxes are shown according to the device.

The following are possible:

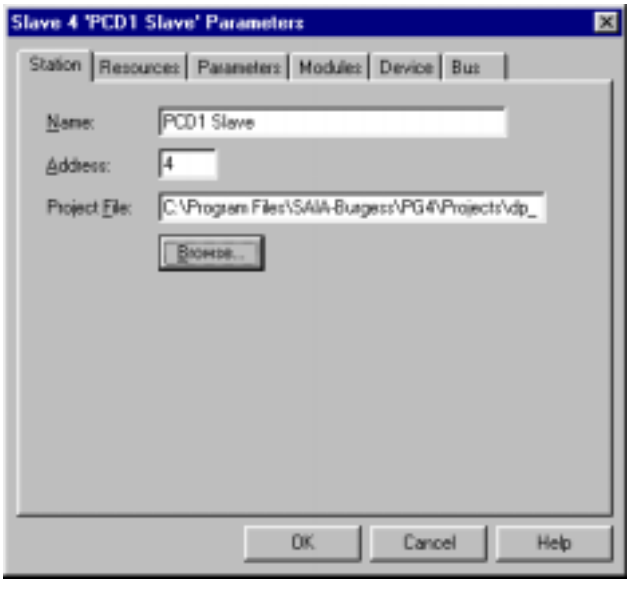
- SAIA master
- SAIA slave
- SAIA PCD0 slave
- other slaves

The following layouts are possible:

For SAIA master devices:



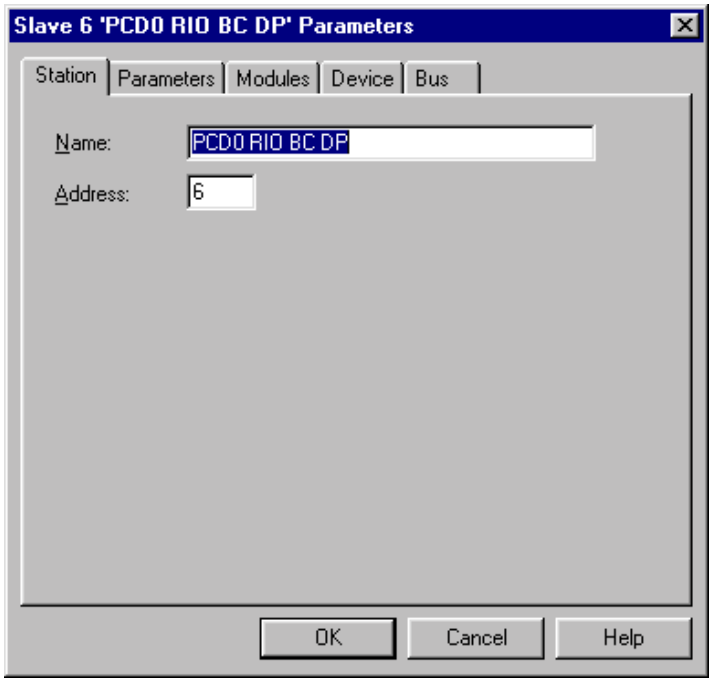

For SAIA PCD1, 2, 6 slave devices:



| | |
|---|---|
| Name: | Station name (max. 32 characters). |
| Address: | Network address of device (range: 1-125). |
| Project File: | Enter the PG4 project to which the device data generated by the PROFIBUS-DP configurator should be assigned. For SAIA devices only. |
| Browse: | Look for a PG4 project. For SAIA devices only. |

For SAIA PCD0 slaves and non-SAIA devices:



Explanation of individual menu items:

**Station:** Definition of device name and PROFIBUS-DP address.

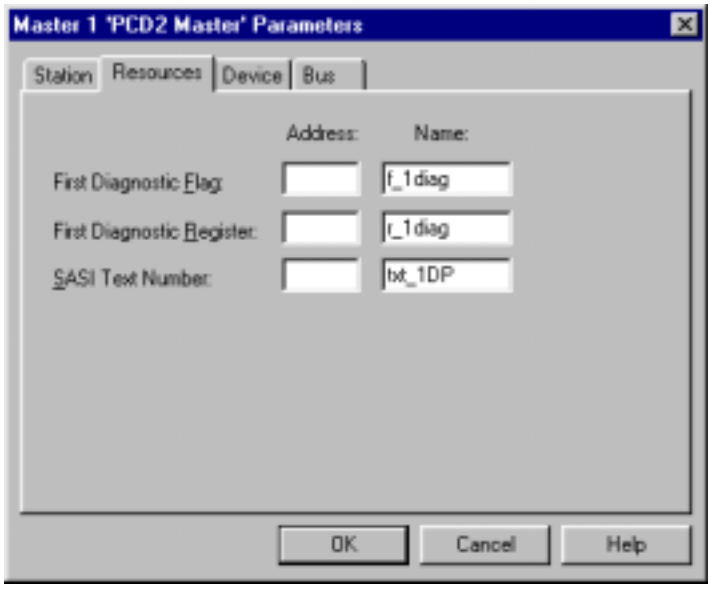
Name: Station name (max. 32 characters).

Address: Network address of device (range: 1 - 125).

Project File: Enter the PG4 project to which the device data generated by the PROFIBUS-DP configurator should be assigned. For SAIA devices only.

Browse: Look for a PG4 project. For SAIA devices only.

**Resources:** Only selectable for SAIA master or slave devices.

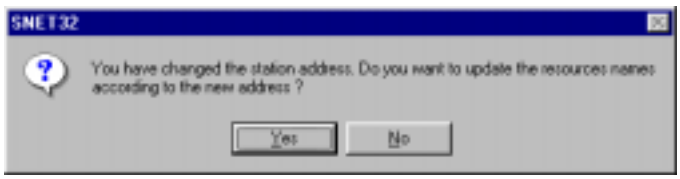



First Diagnostic Flag: Base address and/or symbolic name of the first diagnostic flag. 8 Flags are reserved.

First Diagnostic Register: Base address and/or symbolic name of the first diagnostic register. Up to 70 registers are reserved. This number depends on the extended diagnostics of the slave devices used.

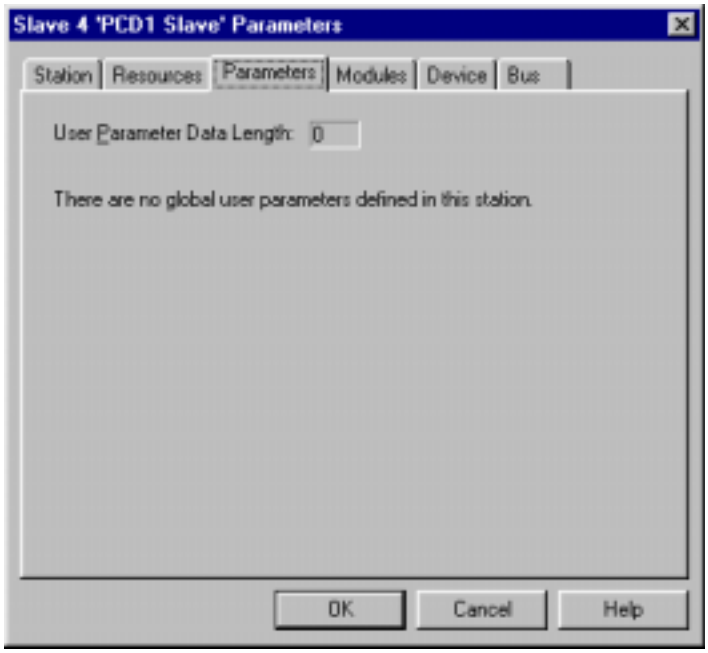
SASI Text Number: Text address and/or symbolic name of SASI text

If the network address of a SAIA PCD1, 2 or 6 device has been changed, the option to adapt symbolic names to the new station number is offered when exiting this window.

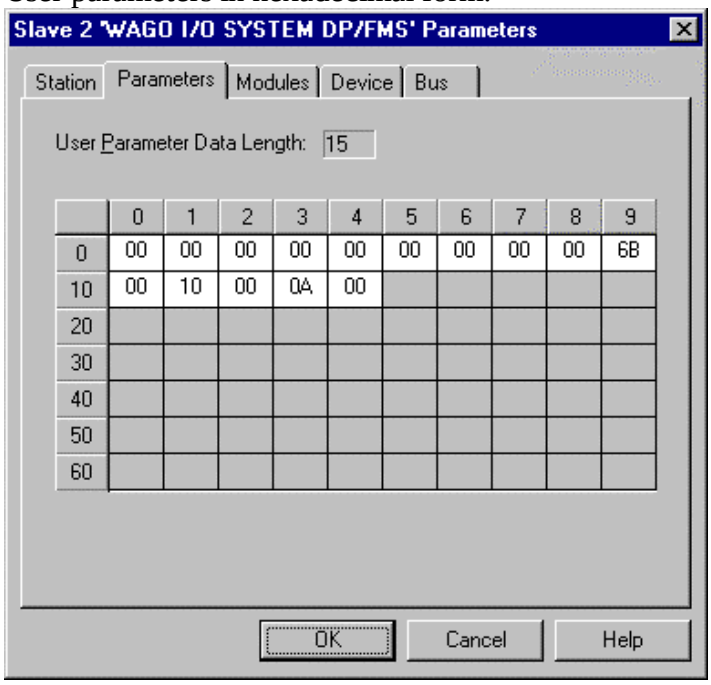



**Parameters:** Selectable with non-SAIA slave devices.
The user parameters of the device can be edited from this page. The user parameters are device-specific and have a different meaning for each device. The meaning of user parameters should be taken from the device descriptions.

Depending on the PROFIBUS-DP device, user parameters can be represented in hexadecimal form or plain text.
If no user parameters are selectable, the following frame appears:

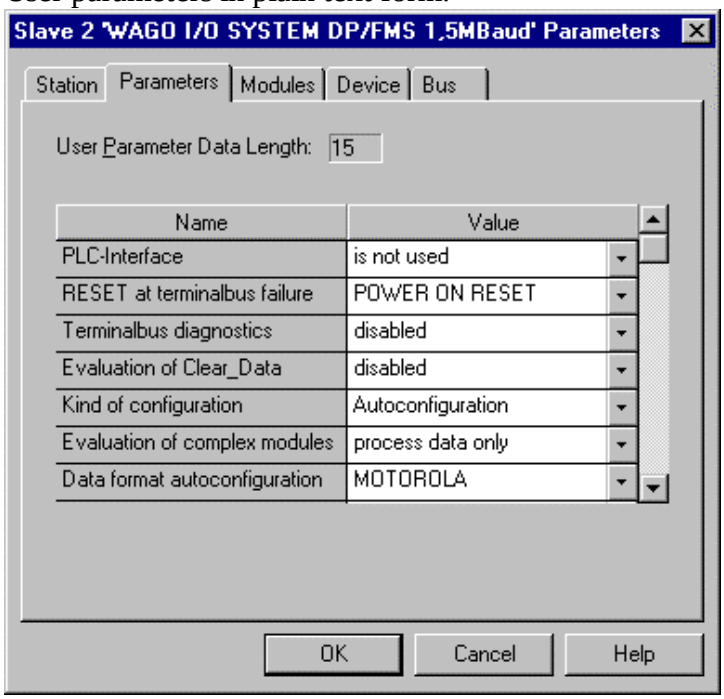


User parameters in hexadecimal form:




User parameters must be entered in the correct position and format.
See device description.

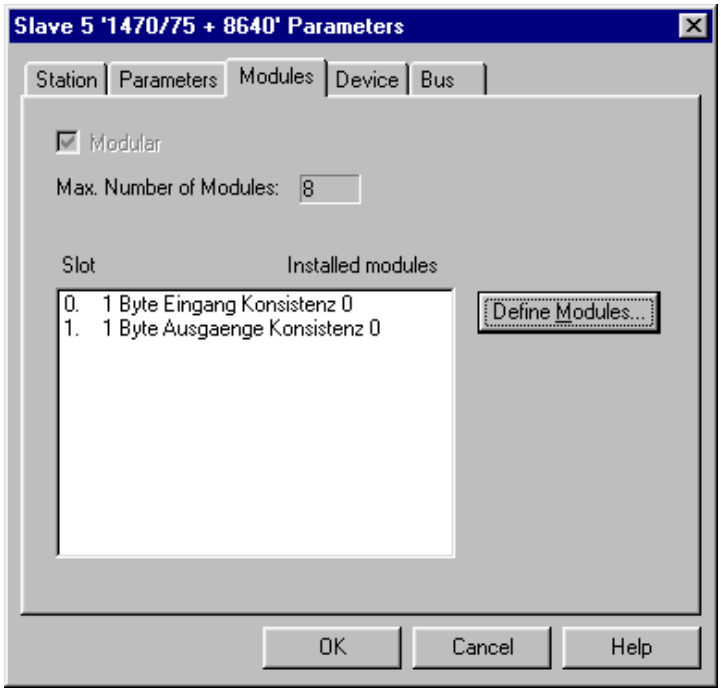
User parameters in plain text form:



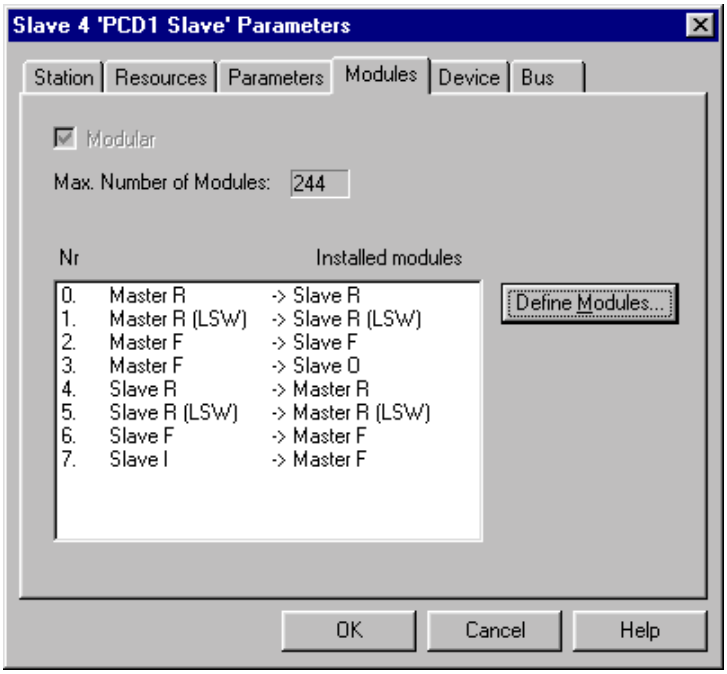

User parameters can be selected from a list of choices. The user does not have to worry about the position and correct entry.

**Modules:** Shows the modules possible for a device and allows these modules to be assigned and configured.
For SAIA slaves, the messages between a SAIA master and a SAIA slave are defined on this page.

Module definition for non-SAIA slaves:




Module definition for SAIA slave:



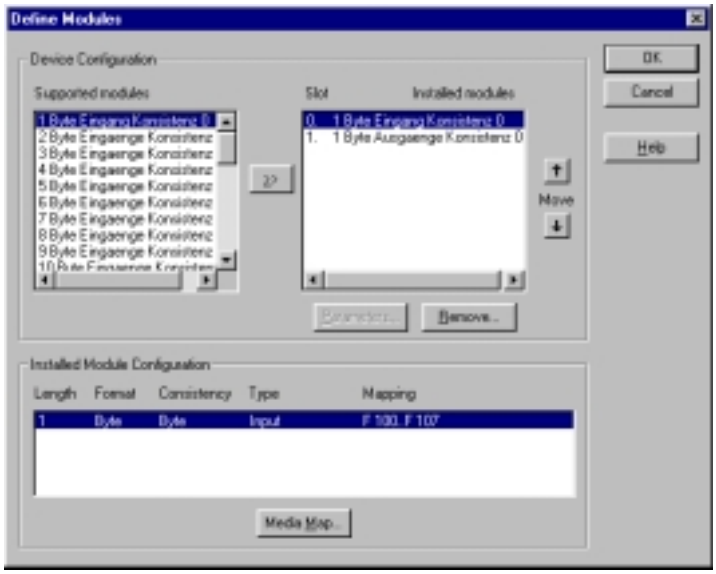
Modular: This indicates whether the device is a modular or compact unit. Modular devices can be expanded with modules. Compact devices cannot be expanded.

Max. Number of Modules: This indicates the maximum number of modules that the device supports.

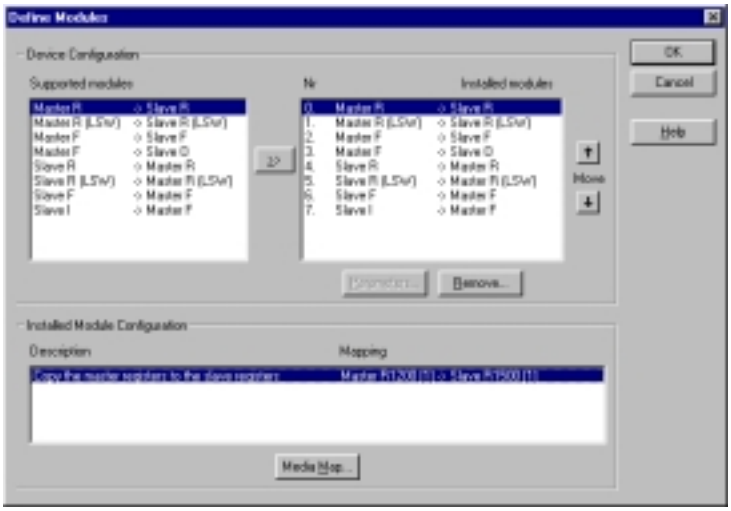
Installed modules: List of modules installed in the device.

Define Modules: Insertion or deletion of modules in the device. Definition of media in the master, which are assigned (mapped) to modules of the device.

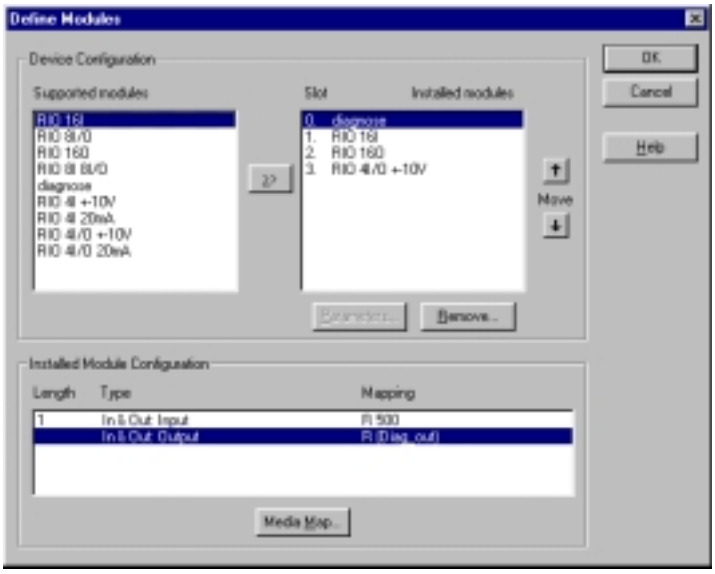
Module definition for non-SAIA slaves:




Module definition for SAIA PCD1,2,6 slaves:

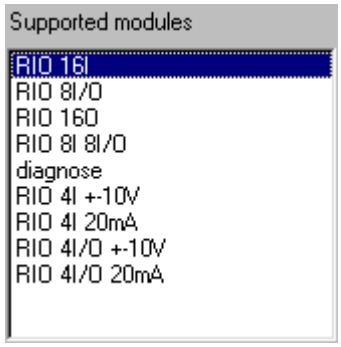


Module definition for SAIA PCD0 slaves:




Supported modules:

List of all modules that are supported by the device.

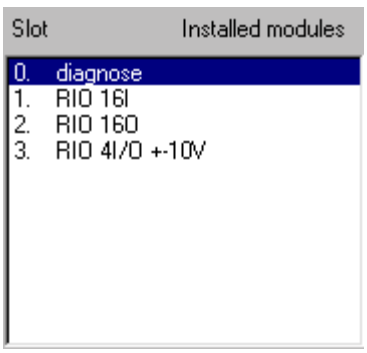




Inserts the selected module from module list into the device configuration. Note: For each newly inserted module, configuration data must be mapped to the master media.

Installed modules:

List of modules already installed in the device.

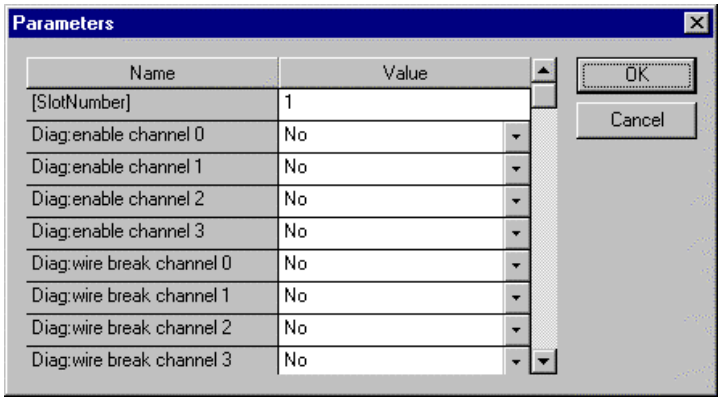




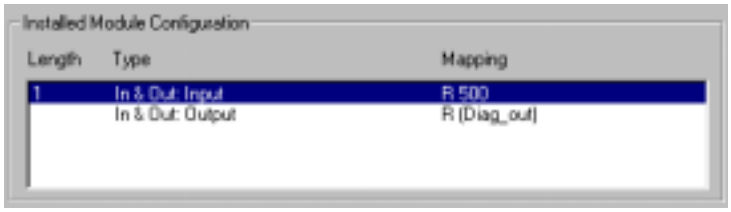
Shifts the selected module up or down in the sequence of installed modules.

Remove: Removes the selected module from the list of installed modules.

Parameters: Definition of module parameters for the module selected from the list of installed modules.

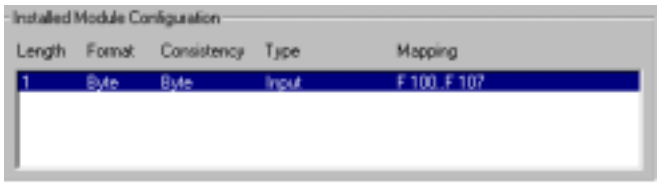



Installed Module Configuration: Display of module configurations for slave devices. This display is dependent upon device type:



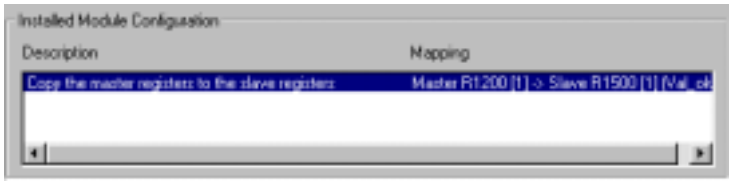
For all non-SAIA slaves:

Length: Data length
Format: Byte or word
Consistency: Byte/word or over the entire data length
Type: Input, output or input/output
Mapping: Assignment to master PCD media



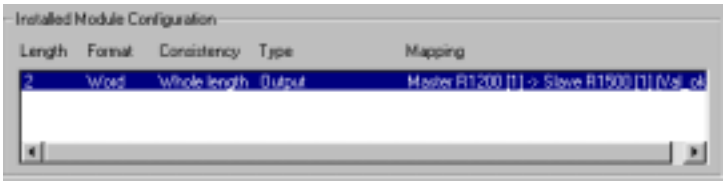

For SAIA PCD1, 2, 6 slaves, when a SAIA master is configured:

Description: Description of data transfer
Mapping: Assignment to master PCD media



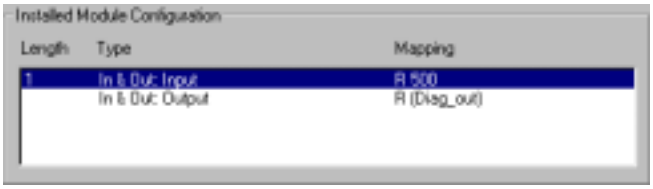

For SAIA PCD1,2,6 slaves, when no master is configured:

Length: Data length
Format: Byte or word
Consistency: Byte/word or over the entire data length
Type: Input, output or input/output
Mapping: Assignment to master PCD media



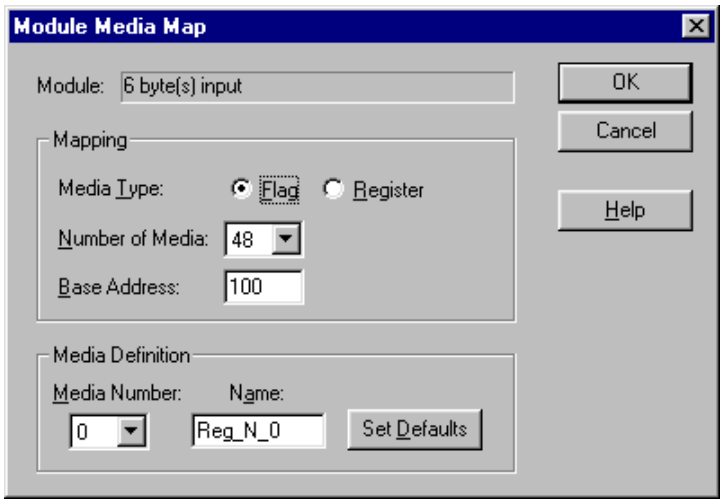

For SAIA PCD0 slaves:

Length: Data length
Type: Input, output or input/output
Mapping: Assignment to master PCD media



**Media Map:** Assignment of selected modules to master PCD media. Mapping enables the user to access remote I/Os or the registers of remote devices via master PCD media (flags or registers). Depending on the slave device used, the media map frame is structured differently.

Media map for non-SAIA slaves:



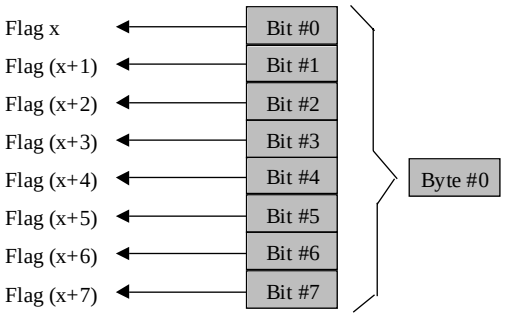

Module: Module description

Mapping: Definition of master PCD media with which this module will be accessed.

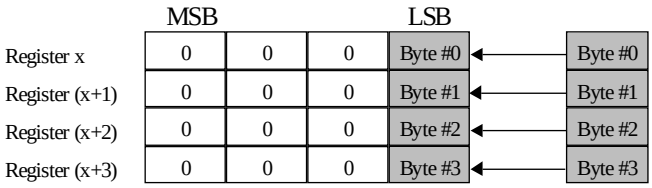
Media Type: Definition of master PCD media type (flag or register) with which the module the module will be accessed.

Number of media: Number of media to be used for accessing the module from the master.
By default, bytes are mapped to flags and words to registers (2 words per register, if consistency is over the entire length; 1 word per register, if consistency is over one word). When required, the user can modify this default mapping in accordance with module specifications.
For example, only one byte per register might be mapped (only using the register's LSB), or 4 bytes might be copied to one register, or 4 bytes might be stored in two registers.
If bytes are mapped to registers, by default 4 bytes per register are used when there is consistency over the entire length and 1 byte per register when there is consistency over one byte.
All unused bytes are set at 0.

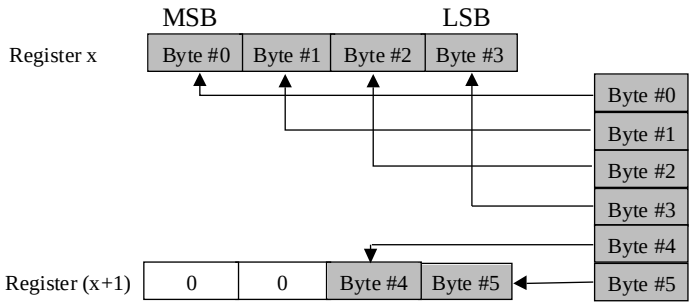
Example:
1 byte is mapped to 8 flags: (1 bit per flag)



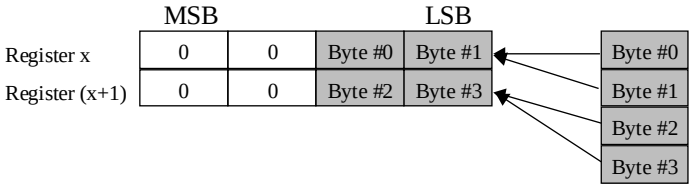

4 bytes are mapped to 4 registers
(1 byte per register)




6 bytes are mapped to 2 registers
(4 bytes per register)




4 bytes are mapped to 2 registers
(2 bytes per register)




Base Address: Base address for the media. This entry is not compulsory in the PG4, since media can be assigned with a symbolic name.
If the base address is not entered, a symbolic name must be entered for each media. Each media will have a name that starts with the name specified in the base address and ends with a sequential number.

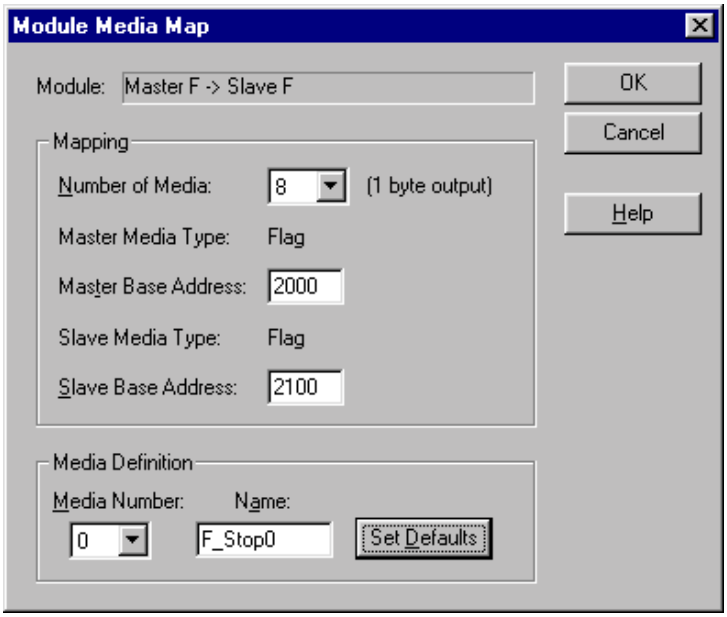
Media Definition: Entry of symbolic media names

Media Number: Choice of media number

Name: Name belonging to the appropriate media number.
These names are then used in the master program.

Set Defaults: This sets default values for all media names.
When the last digit of a default name is a number, this number is incremented for the next media.
If the last character of the default name is not a number, a 0 is attached to the name and the names of subsequent media are incremented.
Caution: To use the 'Set Defaults' button, 'Media Number' must be at 0.

Media map for SAIA slaves:




In contrast to non-SAIA slaves, master and slave media are defined in this window.

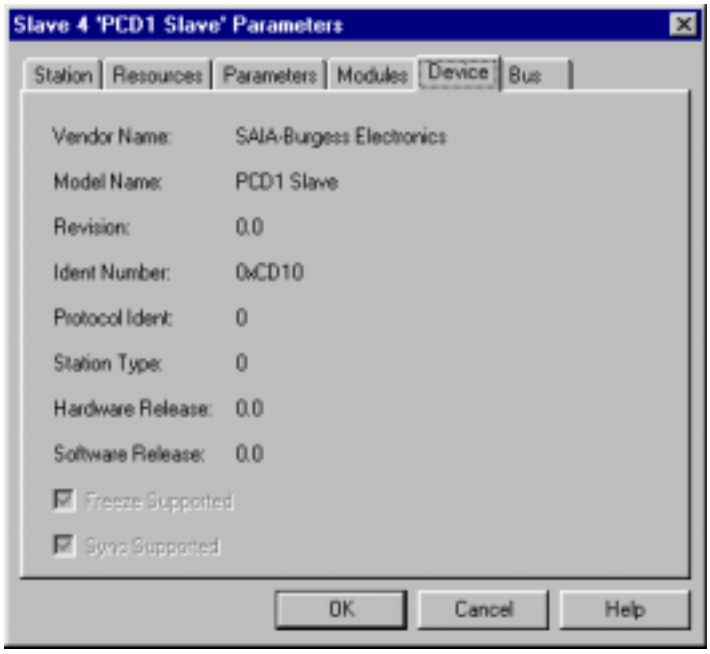
Module Description of module configuration.

Mapping: Definition of master PCD media with which this module will be accessed.

Number of media: Number of media with which the module will be accessed from the master.
(1, 2, 4 or 8 media for register transfer, 8 for flag transfer).

| | |
|---|---|
| Master Media Type: | Type of media used in master PCD to access slave device media. |
| Master Base Address: | Base address for media.<br>This entry is not obligatory in the PG4, since media can be mapped with symbolic names. If the base address is not entered, a symbolic name must be entered in the media definition. |
| Slave Media Type: | Media type for slave PCD, by means of which the module configuration (flags, inputs, outputs or registers) of the actual slave is accessed. |
| Slave Base Address: | Slave base address for the first media to be mapped.<br>This entry is not obligatory in the PG4, since media can be mapped with symbolic names. If the base address is not entered, a symbolic name must be entered in the media definition. |
| Media Definition: | Entry of the symbolic media name. This name will be the same for master and slave media. |
| Media Number: | Choice of media number. |
| Name: | Name belonging to the relevant media number. These names are then used in master and slave programs. |
| Set Defaults: | Set default values for all media names.<br>See description above. |

**Device:** Displays device-specific data.
This data is uploaded from the GSD file and does not refer to PROFIBUS-DP devices actually installed.
The parameters cannot be modified.




Vendor Name: Name of device manufacturer

Model Name: Name of device

Revision: Last device update

Ident Number: Unique PROFIBUS-DP number. This number is checked every time a connection is established between master and slave.
If this number does not match the number present in the slave, no exchange of data takes place between the controllers.

Protocol Ident:

Station Type:

Hardware Release: Hardware version of device.

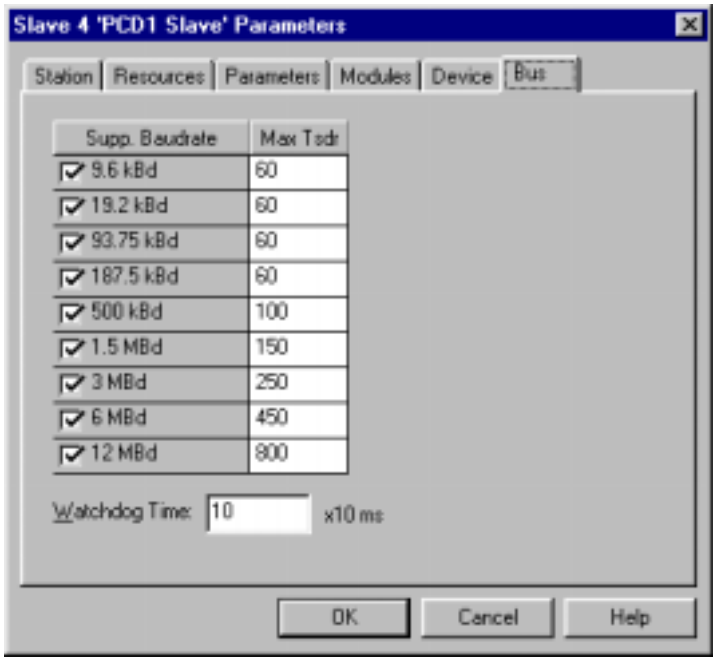
Software Release: Software version of device.

Freeze Supported: Indicates whether a PROFIBUS-DP device supports freeze mode.

Sync Supported Indicates whether a PROFIBUS-DP device supports sync mode.

**Bus:** Displays device-specific bus data.
This data is uploaded from the GSD file and does not refer to PROFIBUS-DP devices actually installed.
The parameters cannot be modified.

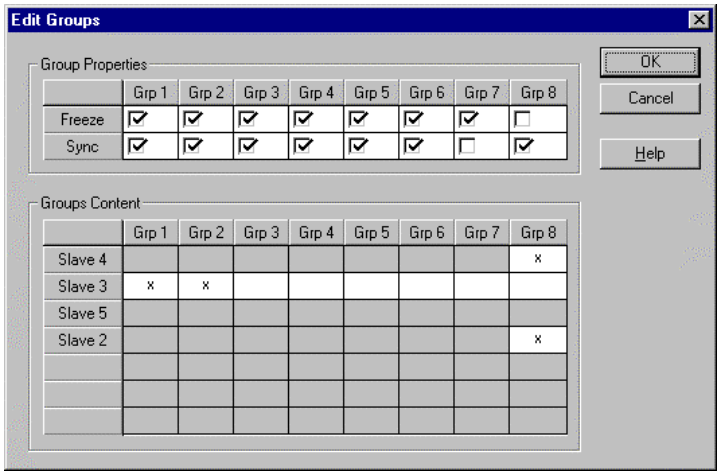



Supp. Baudrate: Indicates the baud rates supported by a device.

Max Tsdr: Indicates maximum 'Tsdr Time' (in bit periods) for each baud rate supported by the module.

Watchdog Time: A watchdog time can be entered in this field.
Each slave can have a different watchdog time.
From menu item 'Bus parameters, Advanced' it is possible to execute the option 'Set Watchdog to slaves', so that the time defined there will be written in this field.

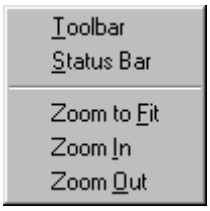
**Groups:** Assignment of PROFIBUS-DP devices to the 8 groups defined in the standard.
It is also possible to assign to groups the 'Global Control Services' supported.




Group Properties: Definition of 'Global Control Services' 'Sync' and 'Freeze'.
For each group.
Click on the 'Freeze' or 'Sync' fields to activate or deactivate the service.

Groups Content: Definition of which PROFIBUS-DP slave devices belong to which group.
To insert a slave in a group, the slave must support the 'Global Control Services' of that group.

### 4.3.4 'View' submenu




**Toolbar:** Shows or hides the tool bar below the menu.

**Status Bar:** Shows or hides the status bar at the bottom at the screen.

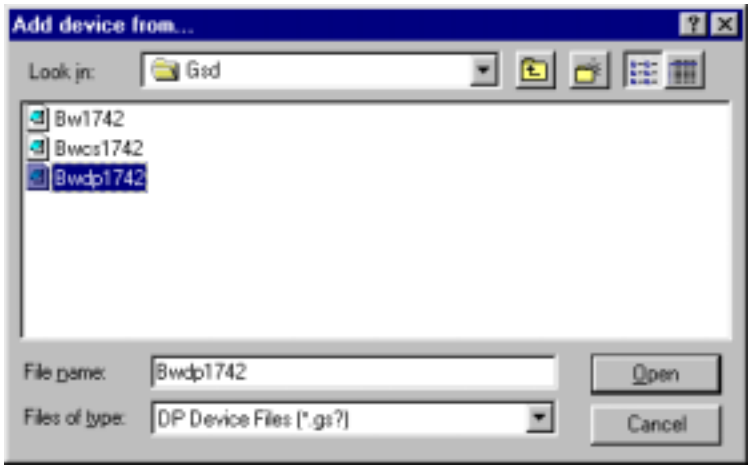
**Zoom to Fit:** With this option all devices in the network are displayed on the screen all the time.

**Zoom In:** Magnification of network screen contents.
Corresponds to this toolbar icon:

**Zoom Out:** Reduces size of network screen contents.
Corresponds to this toolbar icon:

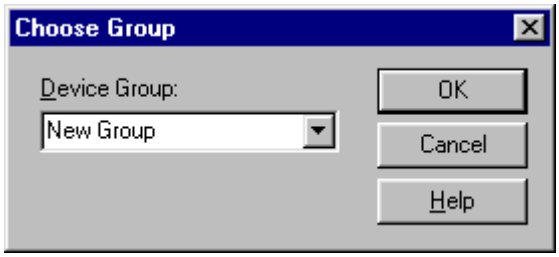
### 4.3.5 'Library' submenu


**Add Device:** Insert new PROFIBUS-DP device.
These devices must have a file with the extension '.gs?'.



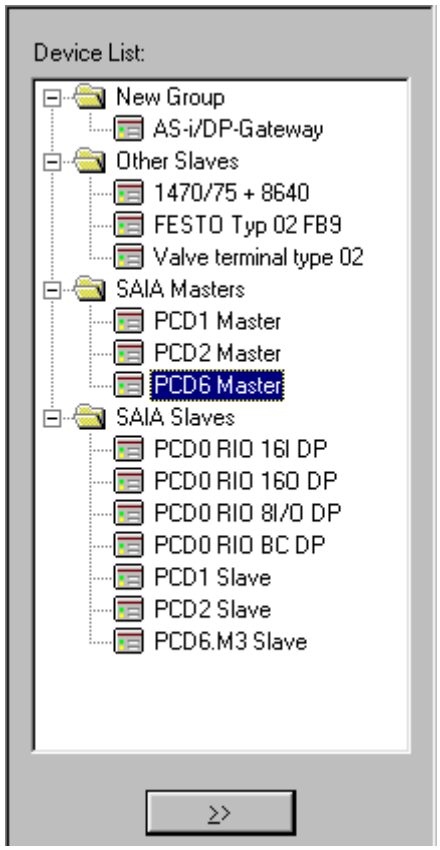

After selecting the '.gs?' file, the device can be assigned to a device group.

This can be done either by saving the device in an existing group, or in a new one. To define a new group, the new group name is simply entered in the entry field.



This new group is then automatically inserted in the device list:

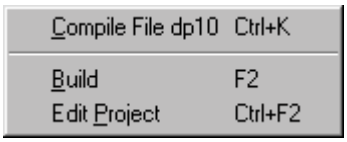



**Remove Device:** Delete a PROFIBUS-DP device from the device list. Deleting a PROFIBUS-DP device from the device list can also be done by pressing the <DEL> key.

**Rename Group:** Rename a device group.
By clicking once on the device group text with the mouse button, the group name can also be edited.

**Caution:** If the PROFIBUS-DP device to be inserted has its own bitmap drawing, which should be displayed in the configurator when the device is called, this *.bmp file must be stored in the directory containing the configurator.

### 4.3.6 'Project' submenu

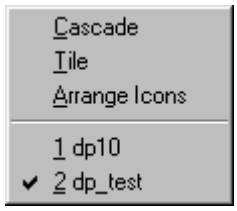



**Compile File:** The project selected is compiled, i.e. '.def' and '.src' files are constructed for all slave and master stations defined in the network.
Corresponds to this toolbar icon:

**Build:** The SAIA device selected in the PROFIBUS-DP network is assembled and linked together with the compiled configurator files.
Corresponds to this toolbar icon:

**Edit Project:** This calls the PG4 Project Manager for the SAIA device selected in the PROFIBUS-DP network.
Corresponds to this toolbar icon :

### 4.3.7 'Window' submenu



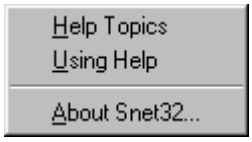

**Cascade:** All open projects are represented on the screen. Representaion takes the form of an overlapping cascade, so that each project title is visible.

**Tile:** All open projects are represented on the screen. Representation is in the form of Windows, without any overlapping of projects.

**Arrange Icons:** Orderly display of all minimized projects.

**1…10** Selection list of all open projects.

### 4.3.8 'Help' submenu

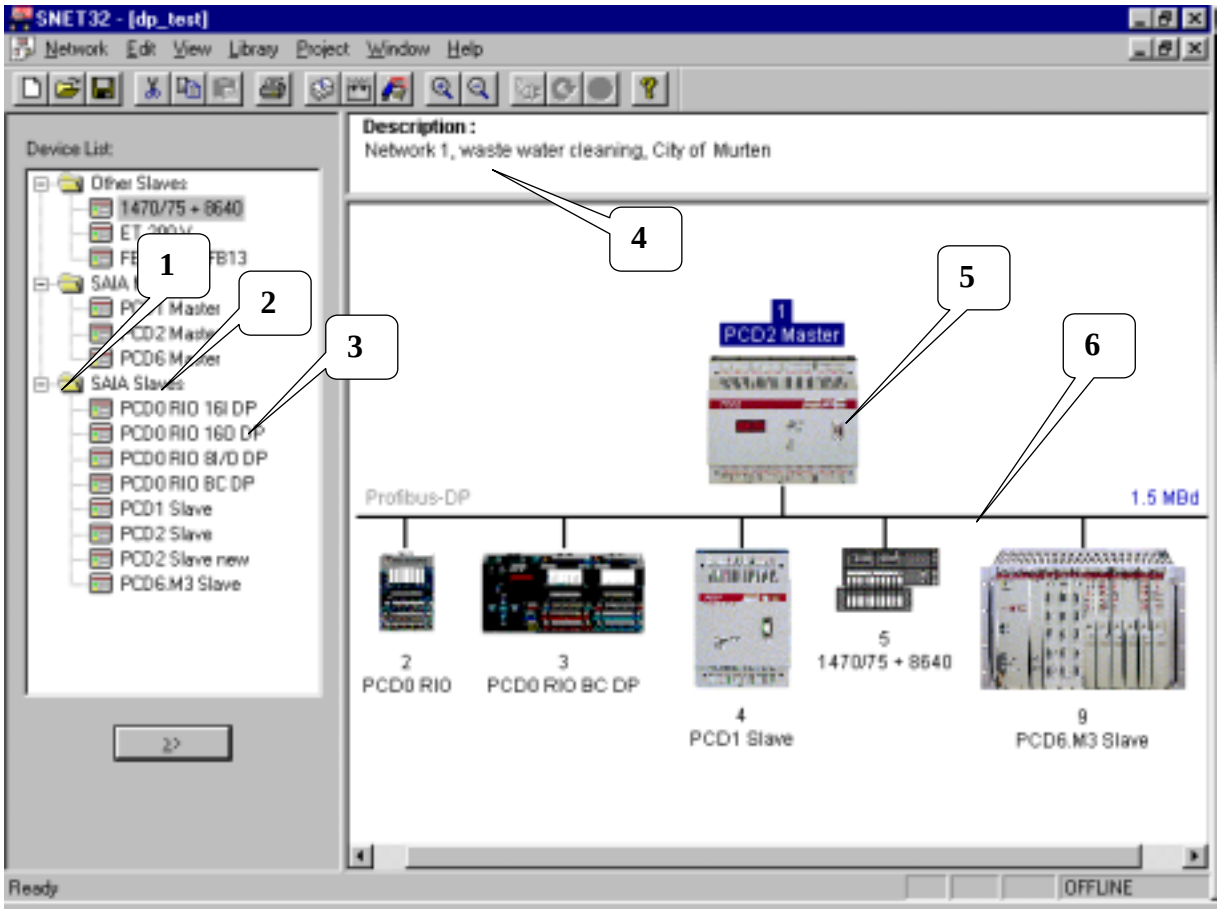



**Help Topics:** Summary of help topics.

**Using Help:** Description of how Help should be used.

**About Snet32:** Indicates the version number and the name of the licence holder.
Corresponds to this toolbar icon :

### 4.3.9 Mouse-sensitive elements in configurator




This section shows which elements of the PROFIBUS-DP configurator can also be edited with the mouse.
The meaning of individual items is described in the preceding chapters.

**1**
**left button:** Double-click: open or close folder.

**right button:**


**2**
**left button:** Edit device group name.

**right button:**

**3**

**left button:** Double-click: insert selected device into the network.

**right button:**


**4**

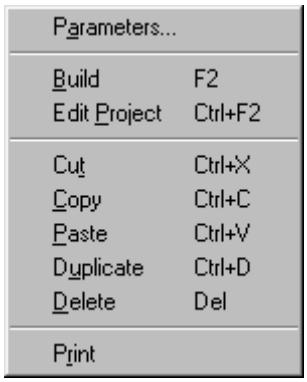
**left button:** Double-click: open editor for network description.

**right button:**


**5**

**left button:** Drag and drop with the device selected.
Double-click: open 'Parameter' entry window

**right button:**




**6**

**left button:** Double-click: open 'Bus Parameters' menu

**right button:**

Notes

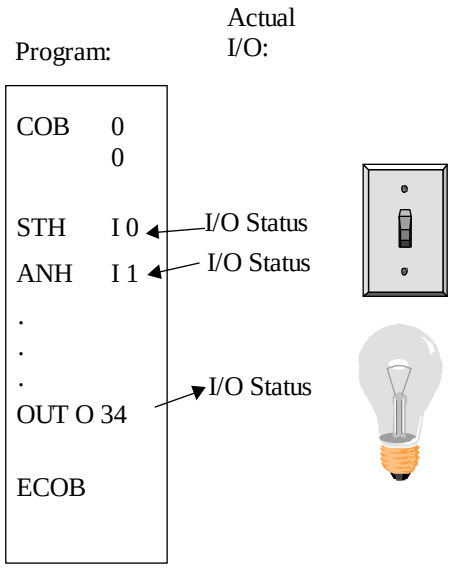
# 5. Programming

## 5.1 Access to slave I/Os and registers in a PCD user program with PROFIBUS-DP

When accessing PCD media, it is necessary to point out an important difference between a program running with PROFIBUS-DP and one which is running directly, i.e. without PROFIBUS-DP.

**Program without PROFIBUS-DP:**

If the input or output of a conventional user program without PROFIBUS-DP is read or written, the status of the actual input is momentarily read or written.



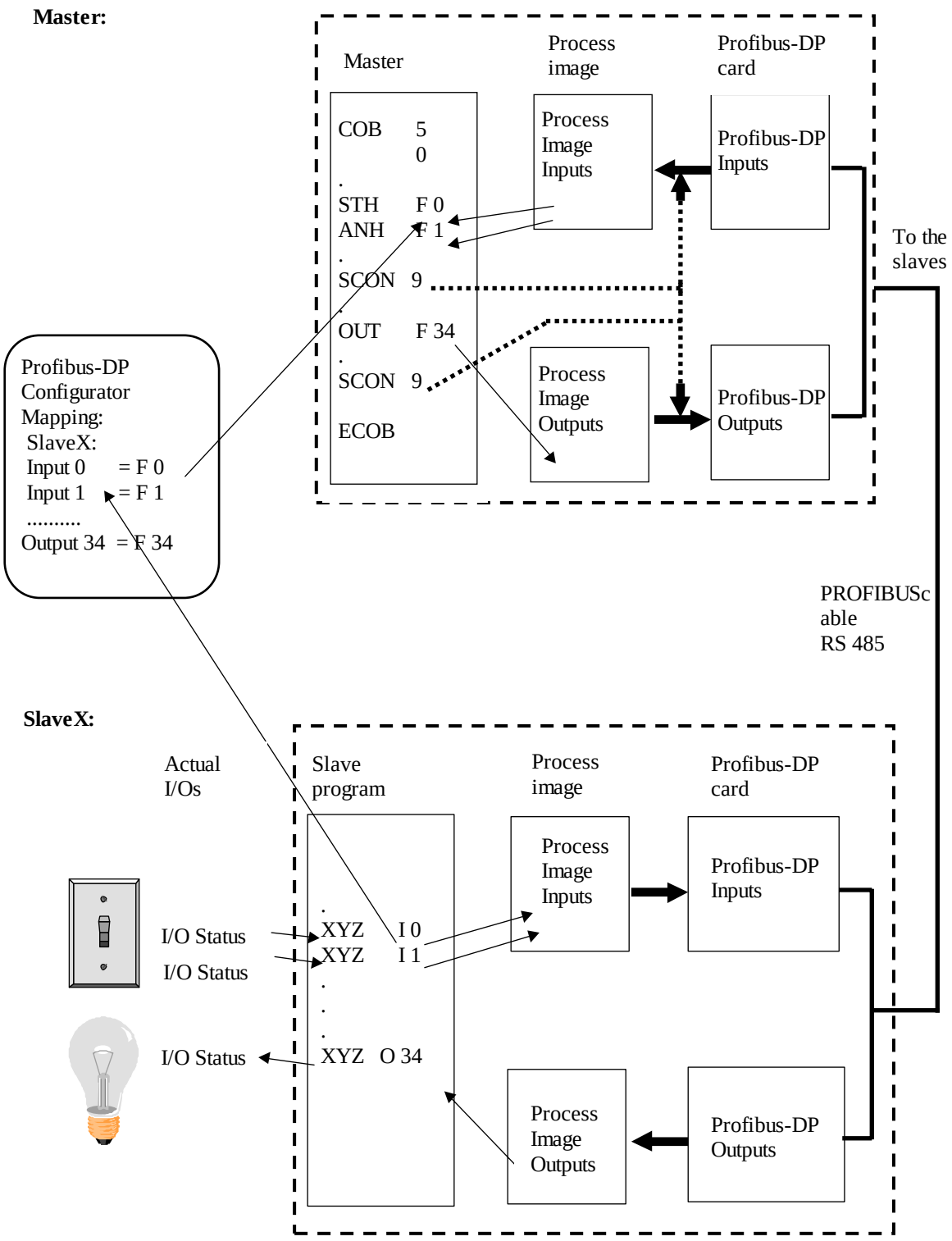
**Program with PROFIBUS-DP slaves:**

If the input or output of a PROFIBUS-DP slave is read or written in a user program with PROFIBUS-DP, the status of the actual input or output is **not** then momentarily read or written. Instead the status is modified in the process image memory of the PCD controller. Data from the process image memory of the PCD controller are then exchanged, either automatically or by the user program, between the PCD controller's process image memory and memory on the PROFIBUS-DP card. This exchange of data between memory on the PROFIBUS-DP card and the slave subsequently takes place automatically in a cyclical sequence and cannot be influenced by the user program.

**Process map memory of PCD controller:**

The PCD controller's process image memory stores all data (I/Os and registers) exchanged between the master and any slaves. This involves differentiating between an input and an output image memory. The PCD controller's input image memory stores any inputs or registers that are read by slaves. This data is then read in the master controller's application program. The PCD controller's output image memory stores any outputs or registers that are written to slaves. This data is described in the master controller's application program.

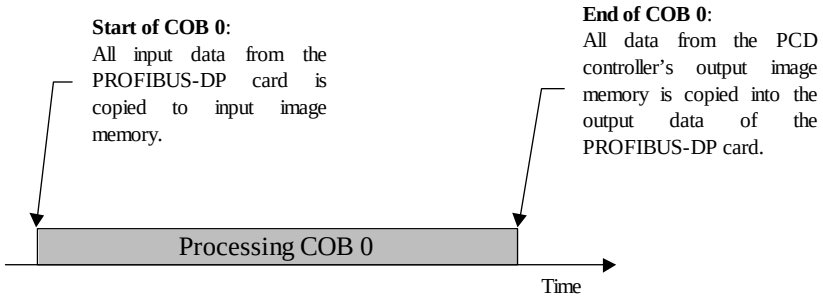
The assignment of I/Os or registers to slave devices in the PCD controller's image memory takes place via the 'Mapping' menu in the PROFIBUS-DP configurator. This involves mapping the I/Os and registers of slave devices to flags and registers in the master controller.


Master:
Master
Process image
Profibus-DP card
COB 5
0
STH F 0
ANH F 1
SCON 9
OUT F 34
SCON 9
ECOB
Process Image Inputs
Profibus-DP Inputs
Process Image Outputs
Profibus-DP Outputs
To the slaves
Profibus-DP Configurator Mapping:
SlaveX:
Input 0 = F 0
Input 1 = F 1
.....
Output 34 = F 34
PROFIBUScable RS 485
SlaveX:
Actual I/Os
Slave program
Process image
Profibus-DP card
I/O Status
I/O Status
I/O Status
XYZ I 0
XYZ I 1
XYZ O 34
Process Image Inputs
Profibus-DP Inputs
Process Image Outputs
Profibus-DP Outputs

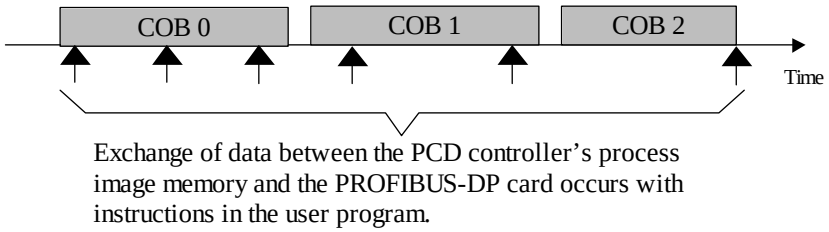
**Exchange of data between the PCD controller's process image memory and the PROFIBUS-DP card.**

The exchange of data between the PCD controller's process image memory and the PROFIBUS-DP card memory can take place in two different ways.
These are:

1. **Default model:**
   The exchange of data between the PCD controller's process image memory and the PROFIBUS-DP card memory takes place automatically. This involves copying all input data from the PROFIBUS-DP card memory to the process image memory when the COB 0 command is processed.
   At the end of COB 0, when the ECOB command is processed, all data from output image memory is copied to the PROFIBUS-DP card's output data memory.
   The exchange of data between the PCD controller's process image memory and the PROFIBUS-DP card memory only occurs as COB 0 is processed. No exchange of data takes place if COB 0 is not processed in the user program.




2. **Advanced model:**
   For this model, the exchange of data between the PCD controller's process image memory and PROFIBUS-DP card memory takes place with instructions in the user program.
   This means that the exchange of data between the PCD controller's process image memory and PROFIBUS-DP card memory can at any time be forced with special user program instructions.



## 5.2 Description of PCD instructions (SASI, SCON)

The following instructions are supported with PROFIBUS-DP:

| | |
|---|---|
| SASI | Assign serial interface |
| SASII | Like SASI but indirect |
| SCON | Serial connect<br>Command to force the exchange of data |
| SCONI | Like SCON but indirect |

### 5.2.1 SASI: Assigning a channel

**Description:** PROFIBUS-DP assignment takes place, as with most PCD communications modes, by the SASI instruction. SASI calls a text which contains the necessary information for initializing a PROFIBUS-DP channel. If communication is to take place across several PROFIBUS-DP channels, a separate SASI instruction should be edited for each channel. For PROFIBUS-DP channel no. 9 is used. Channel 8 is already reserved for future PCDs, which may possess two PROFIBUS-DP interfaces.

**Structure:**

| SASI | Channel<br>Text |
|---|---|

| | | |
|---|---|---|
| Channel | 9 (or 8) | |
| Text | txt_DP | Symbolic name of the text containing the assignment information. This text is generated by the PROFIBUS-DP configurator. |

**Example:**

```
SASI    9       ; Initialize channel 9
        Txt_DP  ; Definition text for
                  PROFIBUS-DP
```

**Flags:** The error (E) flag is set if the definition text is missing or invalid, or if the FW does not support PROFIBUS-DP.

**SASI-Text:** The SASI text is generated by the PROFIBUS-DP configurator and has the following format:

Master:
"MODE:DPM;CONF:DBXxxxx;DIAG:Fyyyy,Rzzzz"

Slave:
"MODE:DPS;CONF:DBXxxxx;DIAG:Fyyyy,Rzzzz"

xxxx: Specific number of a DBX containing all PROFIBUS-DP information.
yyyy: Specific number of the first diagnostic flag or diagnostic output.
zzzz: Specific number of the first diagnostic register.

**Diagnosis:** Diagnosis of a PROFIBUS-DP communication takes place in the usual way for the PCD, i.e. for each communications channel, 8 flags are assigned for rough diagnosis and up to a maximum of 70 registers for fine diagnosis. These diagnostic resources are defined in the configurator.

#### 5.2.1.1 Diagnostic flags with PROFIBUS-DP

| Address | Name | Description |
|---|---|---|
| xxxx | SLAVE_ERR | Slave error<br>Error in the slave |
| xxxx+1 | GCS_BUSY | Global Control Service<br>is processing |
| xxxx+2 | SERV_BUSY | Service function<br>is processing |
| xxxx+3 | DATA_EXCH | Data exchange<br>Exchange of data between master and slave |
| xxxx+4 | | Not used |
| xxxx+5 | | Not used |
| xxxx+6 | CONF_RCV | Configuration received<br>Slave has received a configuration telegram from the master |
| xxxx+7 | CONF_STAT | Configuration status<br>Indicates whether configuration data is OK |

Description of diagnostic flags:

**Slave_error (SLAVE_ERR)**
Master: H = Error in one or more slaves
L = No error in slaves
Slave: H = Error in slave
L = No error in Slave

Master:
The number of the slave that generated the error can be obtained from diagnostic registers +3 to +6. This flag is set low when, after completion of a 'Read slave diagnostic data' telegram, there are no longer any errors present.

**Global Control Service (GCS_BUSY)**
Master: H = Global Control Service is busy
L = Global Control Service has finished
Slave: Not used

Global Control Services are:
Freeze, Unfreeze, Sync and Unsync.

**Service (SERV_BUSY)**
Master: H = Service function is busy
L = Service function has finished
Slave: Not used.

Service functions are:
- Stop data exchange between the PCD controller's process image memory and PROFIBUS-DP card memory.
- Read slave diagnostic data.
- Activate or deactivate a slave.

**Data Exchange (DATA_EXCH)**
Master: H = Data exchange on the PROFIBUS-DP network is running.
L = Data exchange on the PROFIBUS-DP network has halted.
Slave: H = Connection with master established (executing data exchange).
L = No data exchange connection with master.
The flag becomes = L only after the watchdog time is elapsed.

**Configuration received (CONF_RCV)**
Master: Not used.
Slave: H = Slave has received a configuration telegram from master.
L = Slave has not received a configuration telegram from master.

**Configuration status (CONF_STAT)**
Master: Not used.
Slave: H = The configuration telegram from the master corresponds to the slave configuration.
L = The configuration telegram from the master does not correspond to the slave configuration.

#### 5.2.1.2 Diagnostic registers with PROFIBUS-DP

Diagnostic registers are grouped by the following areas:

- Service area
- Station area
- Standard PROFIBUS-DP diagnostic area
- Expanded PROFIBUS-DP diagnostic area

The maximum size of diagnostic registers is defined by the 'Max_Diag_Data_Len' parameter from the slave device GSD file, since slave diagnostic data is stored in the diagnostic registers.
'Max_Diag_Data_Len' can have a maximum size of 244 bytes. When there is more than one slave, the largest 'Max_Diag_Data_Len' parameter always applies.
At present, the diagnostic registers are only used by the master.

Division of diagnostic registers:

| Areas | Address | Description |
|---|---|---|
| Service area | Base +0 | Result of Global Control Service GCS |
| | Base +1 | Result of IL instruction SCON(I) Fct. 0,1,8,9 |
| | Base +2 | Result of IL instruction SCON(I) Function #7 |
| Station area | Base +3 | Error status station 0…31 |
| | Base +4 | Error status station 32…63 |
| | Base +5 | Error status station 64…95 |
| | Base +6 | Error status station 96…126 |
| Standard Profibus- DP Diagnostic | Base +7 | Length of PROFIBUS-DP diagnostic (byte 6…243) |
| | Base +8 | Standard DP diagnostic (byte 0 and 1) |
| | Base +9 | Standard DP diagnostic (byte 2 … 5) |
| Expanded Profibus DP Diagnostic | Base +10 | Expanded DP diagnostic (byte 6…9) |
| | Base +11 | Expanded DP diagnostic (byte 10…13) |
| | Base +12 | Expanded DP diagnostic (byte 14…17) |
| | Base +13 | Expanded DP diagnostic (byte 18...21) |
| | / | / |
| | / | / |
| | Base +69 | Expanded DP diagnostic (byte 242 and 243) |

Description of diagnostic registers:

**Result GCS (base + 0)**

In this register the result of the 'Global Control Service' is stored. The 'Global Control Service' is triggered by function codes 13..16 of the SCON instruction. The result codes are the same as described under: 'Result of IL instruction SCON(I) Fct. 0, 1, 8, 9 (Base + 1)'.

**Result of IL instruction SCON(I)**
**Fct. 0, 1, 8, 9 (base + 1)**

In this register the results of the following functions are stored:

- Run / Stop Data Exchange
  SCON wit function code 0.
- Read slave diagnostics.
  SCON with function code 1.
- Activate or deactivate slave.
  SCON with function code 8 or 9.

The following values are possible here:

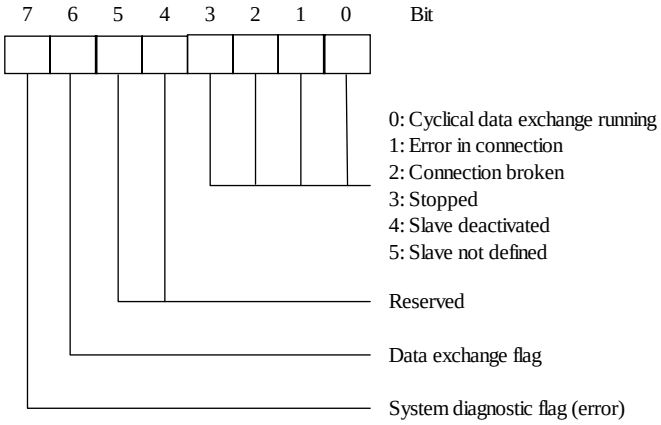
| Wert | Description |
|---|---|
| 0 | Instruction has been successfully completed |
| 1 | Incorrect parameter (contact your local SAIA agent) |
| 2 | Not possible (contact your local SAIA agent) |
| 3 | No local resources (contact your local SAIA agent) |
| 4 | DP error (contact your local SAIA agent) |
| 5 | Slave is not OK |
| 6 | Not defined |
| 7 | Status conflict (contact your local SAIA agent) |
| 8 | Error in acyclic master-slave data exchange (contact your local SAIA agent) |
| | |
| 20 | Timeout |
| 21 | Station number does not exist |
| 22 | Instruction executed more than once (Diag Flag base+2 has not been checked) |
| 23 | Incorrect DP response |
| 24 | Incorrect parameter |

**Result of IL instruction SCON(I) Fct. 7 (base + 2)**

In this register the result of the following function is stored:

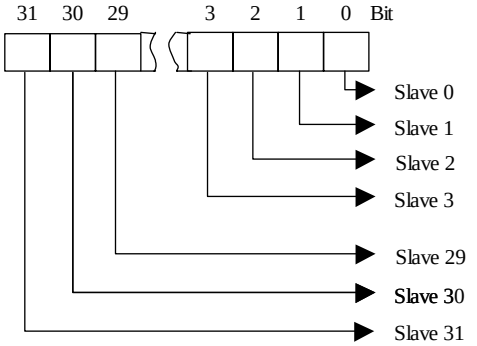
- Read station status.
  SCON with function code 7.

The register is coded here as follows:




**Error status stations 0…31 (base + 3)**

Each bit in this register corresponds to the station number of a slave device. As soon as an error occurs in a slave device, the relevant bit is set high. The bit is set low when, after completion of a 'Read slave diagnostic data' telegram, there is no longer any error present.




**Error status stations 32…63 (base + 4)**

Same function as for diagnostic register (base + 3) with errors for stations 32 to 63.

**Error status stations 64…95 (base + 5)**

Same function as for diagnostic register (base + 3) with errors for stations 64 to 95.

**Error status stations 96…125 (base + 6)**

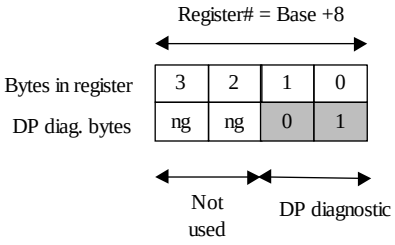
Same function as for diagnostic register (base + 3) with errors for stations 96 to 125.

**Length of PROFIBUS-DP diagnostic bytes 6…243 (base +7)**

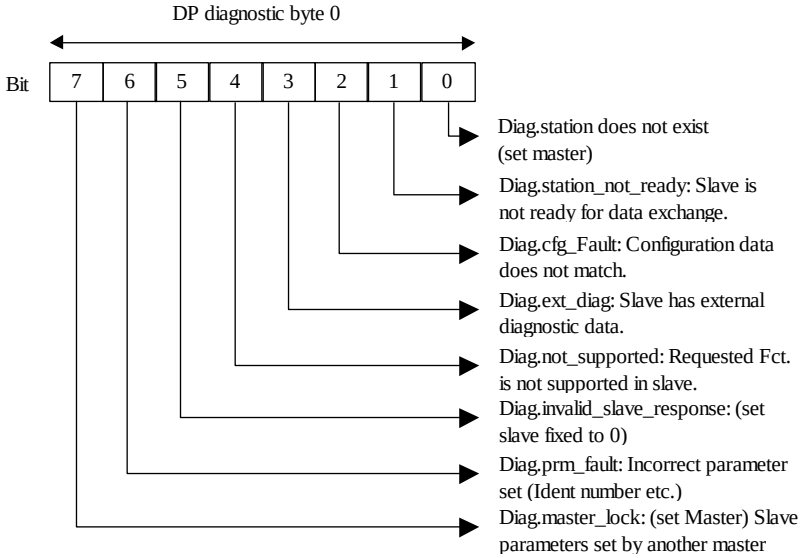
In this register, after an SCON instruction with function 1, the total length of diagnostic data (standard PROFIBUS-DP + external PROFIBUS-DP diagnostic) is stored in bytes. The length of diagnostic data differs in each slave device, amounting to no less than 6 bytes and no more than 244 bytes.

**Standard DP diagnostic: bytes 0 and 1 (base +8)**

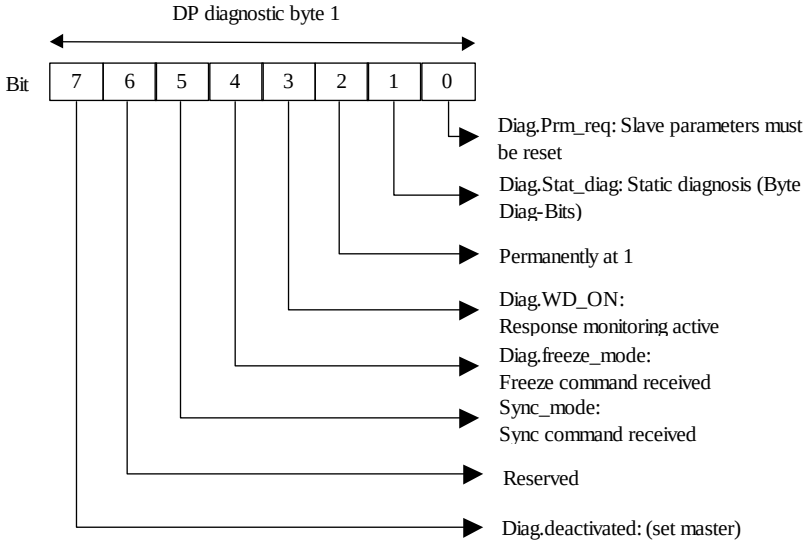
In this register the first two bytes of standard PROFIBUS-DP diagnostic data are stored. Division into diagnostic registers is as follows:




Meaning of PROFIBUS-DP diagnostic byte 0:

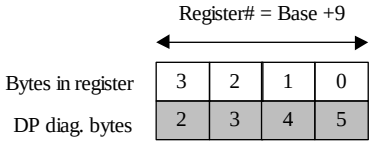


Meaning of PROFIBUS-DP diagnostic byte 1:

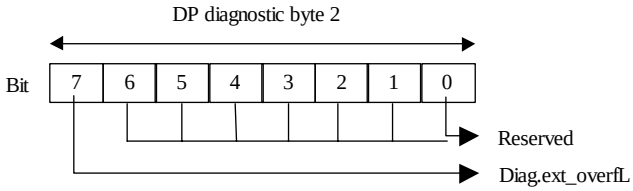



**Standard DP diagnostic: bytes 2 to 5 (base +9)**

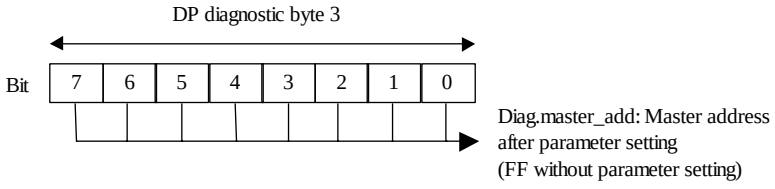
In this register bytes 2 to 5 of the standard PROFIBUS-DP diagnostic data are stored.
The division is as follows:



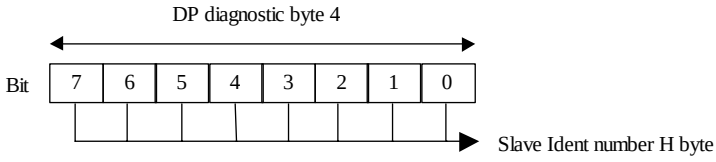

Meaning of PROFIBUS-DP diagnostic byte 2:



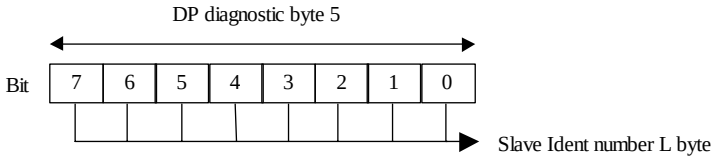

Meaning of PROFIBUS-DP diagnostic byte 3:




Meaning of PROFIBUS-DP diagnostic byte 4:

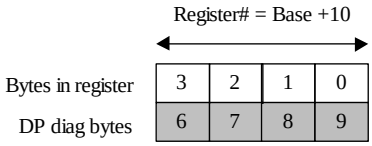


Meaning of PROFIBUS-DP diagnostic byte 5:

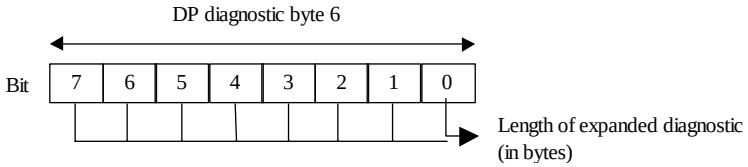



**Expanded DP diagnostic: bytes 6 to 9 (base +10)**

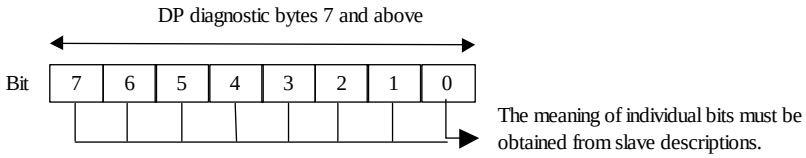
In this register bytes 6 to 9 of the expanded PROFIBUS-DP diagnostic are stored.
The division is as follows:



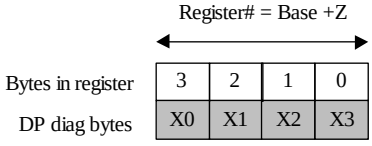

Meaning of PROFIBUS-DP diagnostic byte 6:




Meaning of PROFIBUS-DP diagnostic bytes 7 and above:




**Expanded DP diagnostic: bytes X0 to X3 (base +Z)**
In these registers the expanded diagnostic information is stored.
The division is always as follows:



### 5.2.2 SCON instructions to affect data exchange

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | | 9, 8 |
| Fct. code | 0..16 | Function code<br>This function code triggers a specific function. |
| Parameter | 0..255 | Parameters for the specific function |

**Example:** Request diagnostic data of slave 4

```
STH       SERV_BUSY     ; If diagnostic flag xxxx+2
JR     H  next          ; is not H, then SCON
SCON      9             ; PROFIBUS-DP channel 9
          1             ; Fct.code 1 = read slave
          4             ; diagnostic data of slave 4

next:
```

**Flags:** The error flag is set if the channel is unassigned.

### 5.2.3 SCONI instructions to affect data exchange

**Structure:**

| | |
|---|---|
| SCONI | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | R 0-4095 | Registers with channel numbers 9, 8 |
| Fct. code | R 0-4095 | Registers with function codes 0..16 |
| Parameter | R 0-4095 | Registers with the parameters for the specific function 0..255 |

**Example:** All data are to be exchanged between the PCD controller's process image memory and that of the PROFIBUS-DP card, controlled by the user program.

```
LD      R 2000          ; Load register 2000
          9       ; with channel 9
LD      R 2001          ; Load register 2001
          3       ; wit Fct. code 3 =
                  ; Force data exchange
LD      R 2002          ; Load register 2002
          0       ; with parameter 0 =
                  ; Input and output
                  ; image memory
SCONI   R 2000          ; SCONI instruction
        R 2001
        R 2002
```

**Flags:** The error flag is set if the channel is unassigned.

### 5.2.3.1 Description of Fct. codes and parameters for SCON(I) instruction

| Fct Code | | Para-meter | Description | Diagnostic affected | |
|---|---|---|---|---|---|
| Master | Slave | | | Flag | Reg |
| 0 | | 0 | Stop data exchange between master and slaves | 2, 3 | 1 |
| 1 | | Slave no. 0..126 | Read slave diagnostic data | 0, 2 | 3 - 6<br>0,7,8,9<br>10-69 |
| 2 | 2 | | **Start / Stop default data exchange between image memory and the PROFIBUS-DP card** | | |
| | | 0 | Stop default model data exchange for all slaves between the entire image memory and the PROFIBUS-DP card (COB 0; ECOB) | | |
| | | 1 | Start default model data exchange for all slaves between the entire image memory and the PROFIBUS-DP card (COB 0; ECOB) | | |
| | | 2 | Stop data exchange for all slaves between input image memory and the PROFIBUS-DP card (Start of COB 0) | | |
| | | 3 | Start data exchange for all slaves between input image memory and the PROFIBUS-DP card (Start of COB 0) | | |
| | | 4 | Stop data exchange for all slaves between output image memory and the PROFIBUS-DP card (End of COB 0) | | |
| | | 5 | Start data exchange for all slaves between output image memory and the PROFIBUS-DP card (Ende von COB 0) | | |
| 3 | 3 | 0 | Force data exchange for all slaves between the entire image memory and the PROFIBUS-DP card | | |
| | | 1 | Force data exchange for all slaves between input image memory and the PROFIBUS-DP card | | |
| | | 2 | Force data exchange for all slaves between output image memory and the PROFIBUS-DP card | | |
| 4 | | Slave no. 0..126 | Force data exchange for a slave device between input image memory and the PROFIBUS-DP card | | |
| 5 | | Slave no. 0..126 | Force data exchange for a slave device between output image memory and the PROFIBUS-DP card | | |
| 6 | | Slave no. 0..126 | Force data exchange for a slave device between the entire image memory and the PROFIBUS-DP card | | |
| 7 | | Slave no. 0..126 | Read status of a slave | | 2 |
| 8 | | Slave no. 0..126 | Deactivate slave | 2 | 1 |
| 9 | | Slave no. 0..126 | Activiate slave | 2 | 1 |
| 10 | | Group no. 0..255 | Force data exchange for a group of slaves between input image memory and the PROFIBUS-DP card | | |
| 11 | | Group no. 0..255 | Force data exchange for a group of slaves between output image memroy and the PROFIBUS-DP card | | |
| 12 | | Group no. 0..255 | Force data exchange for a group of slaves between the entire image memory and the PROFIBUS-DP card | | |
| 13 | | Group no. 0..255 | FREEZE | 1 | 0 |
| 14 | | Group no. 0..255 | UNFREEZE | 1 | 0 |
| 15 | | Group no. 0..255 | SYNC | 1 | 0 |
| 16 | | Group no. 0..255 | UNSYNC | 1 | 0 |

### 5.2.3.2 SCON(I) 0: stop data exchange between master and slave

This instruction can be used to stop data exchange on the PROFIBUS-DP network. To restart data exchange, it is necessary to execute a 'Restart' - 'Cold' on the PCD. With this instruction all slave outputs are set to 0. This instruction is mainly used in XOB 0, so that slave outputs are not left in an undefined state before powering off the master. Diagnostic flag +2 is set high as soon as this instruction executes. When the instruction is finished, the flag is set low. This instruction may only be executed when diagnostic flag +2 is low.

When the instruction has been executed and the status of diagnostic flag +2 is low, the result of the operation is written to diagnostic register +1. A description of the response code is given in section 5.2.1.2 'Diagnostic registers with PROFIBUS-DP'. Diagnostic flag +3 shows the status of data exchange on the PROFIBUS-DP network.

Diagnostic flag +3: L = Data exchange on the PROFIBUS-DP network has stopped.
H = Data exchange on the PROFIBUS-DP network is running.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

Channel 9, 8

Fct. code 0

Parameter 0 ; Stop data exchange on the
; PROFIBUS-DP network

**Flags:** The error flag is set if the channel is unassigned or if the instruction is called when diagnostic flag +2 is not high.

**Example:** Stop data exchange on the PROFIBUS-DP network:

```
STH       SERV_BUSY     ; If diagnostic flag +2
JR     H  NEXT          ; is not H, then SCON
SCON      9             ; PROFIBUS-DP channel 9
          0             ; Fct.code 0
          0             ; Stop PROFIBUS-DP
NEXT:
```

#### 5.2.3.3 SCON(I) 1: read slave diagnostic data

With this instruction the diagnostic data of the slave can be read. Diagnostic data is mostly read when an error has been detected in the slave. This is indicated by setting diagnostic flag +0. The user can then identify the faulty slave by means of diagnostic registers +3 to + 6 and read the diagnostic data of that slave. As soon as this instruction is executed, diagnostic flag +2 is set high and, when the instruction is finished, reset low. When the instruction has been executed and the status of diagnostic flag +2 is low, the result of the operation is written to diagnostic register +1. A description of the response code is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'. This instruction may only be executed when the status of diagnostic flag +2 is 0.

When the instruction is finished, in diagnostic registers +3 to + 6 the relevant bit for the slave to which the instruction was addressed is set low. The following values are stored in the diagnostic registers:
Diagnostic register +7: Length of expanded PROFIBUS-DP diagnostic
Diagnostic register +8: Standard PROFIBUS-DP diagnostic bytes 0 and 1
Diagnostic register +9: Standard PROFIBUS-DP diagnostic bytes 2 to 5
Diagnostic register +10: Expanded PROFIBUS-DP diagnostic bytes 6 to 9
etc.
A description of the response code is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | 9, 8 | |
| Fct. code | 1 | |
| Parameter | 0..126 | ; Station number |

**Flags:** The error flag is set if the channel is unassigned or if the instruction has been called when diagnostic flag +2 is high.

**Example:** Read slave diagnostic data from slave 5:

```
STH       SLAVE_ERR      ; If diagflag +0 = H
ANL       SERV_BUSY      ; and no SCON is active
JR     L  NEXT           ; (diagflag +2 = L), then SCON
SCON      9              ; PROFIBUS-DP channel 9
          1         ; Fct.code 1
          5              ; Slave no. 5
NEXT:
```

#### 5.2.3.4 SCON(I) 2: start / stop default data exchange between image memory and the PROFIBUS-DP card

With this instruction default data exchange between the image memory and the PROFIBUS-DP card can be started or stopped. Default data exchange refers to the data exchange that is executed automatically when COB 0 starts up and when it ends. This data exchange can be changed to the following function:

Parameters:

0 Stop default model data exchange for all slaves between the entire image memory and the PROFIBUS-DP card (COB 0; ECOB)

1 Start default model data exchange for all slaves between the entire image memory and the PROFIBUS-DP card (COB 0; ECOB)

2 Stop data exchange for all slaves between input image memory and the PROFIBUS-DP card (Start COB 0)

3 Start data exchange for all slaves between input image memory and the PROFIBUS-DP card (Start COB 0)

4 Stop data exchange for all slaves between output image memory and the PROFIBUS-DP card (End COB 0)

5 Start data exchange for all slaves between output image memory and the PROFIBUS-DP card (End COB 0)

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | 9, 8 | |
| Fct. code | 2 | |
| Parameter | 0..5 | ; Parameter |

**Flags:** The error flag is set if the channel is unassigned.

**Example:** Stop data exchange for all slaves between input image memory and the PROFIBUS-DP card (Start COB 0)

```
SCON    9       ; PROFIBUS-DP channel 9
        2       ; Fct.code 2
        2       ; Parameter 2
```

#### 5.2.3.5 SCON(I) 3: force data exchange for all slaves between the image memory and the PROFIBUS-DP card

With this instruction, data exchange between the image memory of all slaves and the PROFIBUS-DP card can at any time be forced in the user program. This forcing can take place in the following way:

Parameters:

0 Force data exchange for all slaves between the entire image memory and the PROFIBUS-DP card

1 Force data exchange for all slaves between input image memory and the PROFIBUS-DP card

2 Force data exchange for all slaves between output image memory and the PROFIBUS-DP card

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | | 9, 8 |
| Fct. code | 3 | |
| Parameter | 0..2 | ; Parameter |

**Flags:** The error flag is set if the channel is unassigned.

**Example:** Force data exchange for all slaves between the entire image memory and the PROFIBUS-DP card

```
SCON    9       ; PROFIBUS-DP channel 9
        3               ; Fct.code 3
        0       ; Parameter 0 = input and
                        ; output image memory
```

### 5.2.3.6 SCON(I) 4, 5, 6: force data exchange for a slave between the image memory and the PROFIBUS-DP card

With these instructions data exchange between the image memory of a slave and the PROFIBUS-DP card can at any time be forced in the user program. This forcing can take place in the following way:

Fct Code:

4 Force data exchange for a slave between input image memory and the PROFIBUS-DP card.

5 Force data exchange for a slave between output imate memory and the PROFIBUS-DP card.

6 Force data exchange for a slave between the entire image memory and the PROFIBUS-DP card.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | |
|---|---|
| Channel | 9, 8 |
| Fct. Code | 4, 5, 6 ; Fct code |
| Parameter | 0..126 ; Slave number |

**Flags:** The error flag is set if the channel is unassigned.

**Example:** Force data exchange for slaves 12 between output image memory and the PROFIBUS-DP card.

```
SCON    9       ; PROFIBUS-DP channel 9
        5       ; Fct.code 5
        12              ; Slave 12
```

### 5.2.3.7 SCON(I) 7: read status of a slave

With this instruction the status of a slave can be read. After execution of the instruction, the slave's status is written to diagnostic register +2. A description of diagnostic register +2 is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | 9, 8 | |
| Fct. code | 7 | |
| Parameter | 0..126 | ; Slave number |

**Flags:** The error flag is set if the channel is unassigned.

**Example:** Read status of slave 34.

```
SCON    9       ; PROFIBUS-DP channel 9
        7       ; Fct.code 7
        34              ; Slave 34
```

### 5.2.3.8 SCON(I) 8, 9: deactivate / activate slave

With this instruction a slave can be activated or deactivated. When the instruction is executed, diagnostic flag +2 is set high and when the instruction finishes, it is set low. After the instruction has been executed and the status of diagnostic flag +2 is low, the result of the operation is written to diagnostic register +1. A description of the response code is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'. This instruction may only be executed if the status of diagnostic flag +2 is 0. The deactivation or activation of a slave is triggered by the following Fct. codes:

Fct Code:

8 Deactivate slave
9 Activate slave

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | 9, 8 | |
| Fct. code | 8, 9 | ; deactivate / activate slave |
| Parameter | 0..126 | ; Slave number |

**Flags:** The error flag is set if the channel is unassigned or if the instruction is called when diagnostic flag +2 is high.

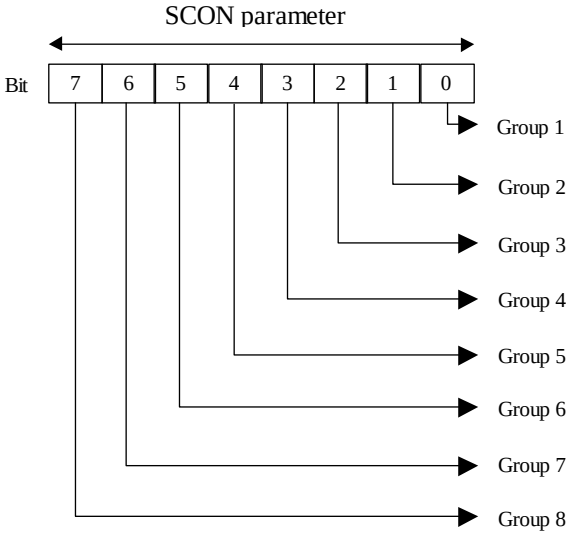
**Example:** Deactivate slave 32.

```
STH     SERV_BUSY       ; If diagnostic flag +2
JR   H  NEXT            ; is not high, then SCON

SCON    9               ; PROFIBUS-DP channel 9
        8           ; Fct.code 8
        32          ; Slave 32
NEXT:
```

### 5.2.3.9 SCON(I) 10, 11, 12: force data exchange for a group of slaves between the image memory and the PROFIBUS-DP card

With these instructions, data exchange between the image memory of one or more groups of slaves and the PROFIBUS-DP card can at any time be forced in the user program. Assigning a slave to a group takes place with the PROFIBUS-DP configurator.

PROFIBUS-DP supports the formation of a maximum of 8 groups. These groups can be assigned as many slaves as required. The choice of group in the SCON parameter is bit-oriented according to the following pattern:




Forcing can be applied here to more than one group at a time. This forcing can take place in the following way:

Fct code:

10 Force data exchange for a group of slaves between input image memory and the PROFIBUS-DP card.

11 Force data exchange for a group of slaves between output image memory and the PROFIBUS-DP card.

12 Force data exchange for a group of slaves between the entire image memory and the PROFIBUS-DP card.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

| | | |
|---|---|---|
| Channel | 9, 8 | |
| Fct. code | 10, 11, 12 | ; Fct code |
| Parameter | 0..255 | ; Group number |

**Flags:** The error flag is set if the channel is unassigned.

**Example:** Force data exchange for groups 1 and 2 between input image memory and the PROFIBUS-DP card.

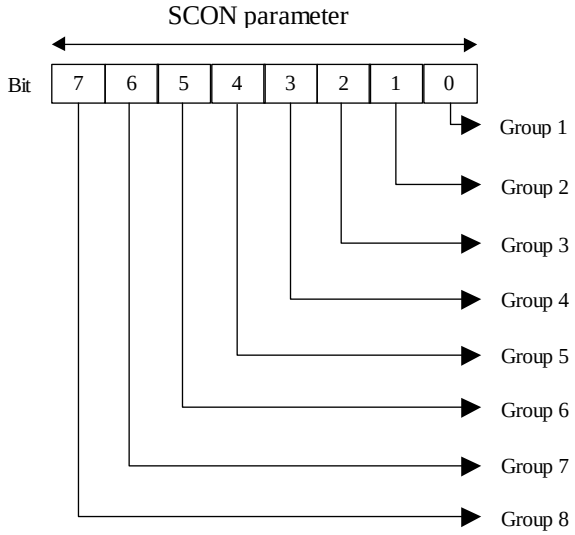
```
SCON    9; PROFIBUS-DP channel 9
        10      ; Fct.code 10
        3; Groups 1 and 2 (00000011)
```

### 5.2.3.10 SCON(I) 13, 14: Global Control Service Freeze, Unfreeze

With these instructions, the 'Freeze' and 'Unfreeze' commands can be triggered for one or more groups of slaves. The instruction is used for the purpose of input synchronization. With the 'Freeze' instruction, the master causes a slave or group of slaves simultaneously to freeze inputs in their present state. The slaves addressed therefore stop their inputs at exactly the same time. In the next data cycle (Data_exch) the slaves transmit the frozen inputs to the master. Any changes at the inputs are not recognized by the slaves and are also not passed on the the master. After the conclusion of this action, the master sends an 'Unfreeze' instruction to the group. Input changes are now sent again from the slave to the master in the normal data cycle. It is permissible for the master, after one 'Freeze' instruction, to send further 'Freeze' instructions to the slaves. In this case the current status of inputs is frozen each time and sent to the master in the next data cycle.

Diagnostic flag +1 is set high as soon as this instruction starts up. When the instruction has finished, the flag is set low and the result of the operation is written to diagnostic register +0. A description of the response code in diagnostic register +0 is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'.

This instruction may only be executed if the status of diagnostic flag +1 is low. and L hat. Assigning a slave to a group takes place with the PROFIBUS-DP configurator. PROFIBUS-DP supports the formation of a maximum of 8 groups. These groups can be assigned as many slaves as required. The choice of group in the SCON parameter is bit-oriented according to the following pattern:




A 'Freeze' or 'Unfreeze' instruction can be executed here on several groups simultaneously.

Fct. code to trigger 'Freeze' or 'Unfreeze' instructions:

13 Start freeze instruction.
14 Start unfreeze instruction.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

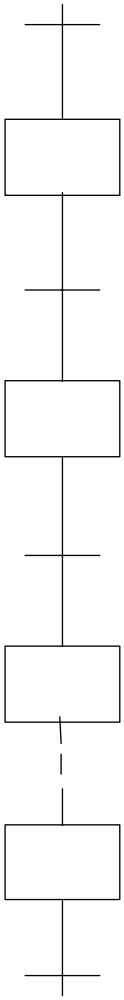
Channel 9, 8

Fct. code 13, 14 ; Fct code

Parameter 0..255 ; Group number

**Flags:** The error flag is set if the channel is unassigned or if the instruction is called when diagnostic flag +1 is high.

**Example:** Execute freeze and unfreeze sequence for the slaves of group 5.



```
STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue

SCON    9               ; PROFIBUS-DP channel 9
        13      ;  Freeze
        16      ;  Group 5 (00010000)

STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue

LD   T  3               ; Load timer with
        100             ;   value 100, delay so that
                        ;   the slaves transmit their
                        ;   frozen inputs to the
STL  T  3               ;   master

STL  F  XX      ; Process the
                        ;   frozen I/Os of
                        ;   slaves

SCON    9               ; PROFIBUS-DP channel 9
        14      ;  Unfreeze
        16      ;  Group 5 (00010000)

STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue
```

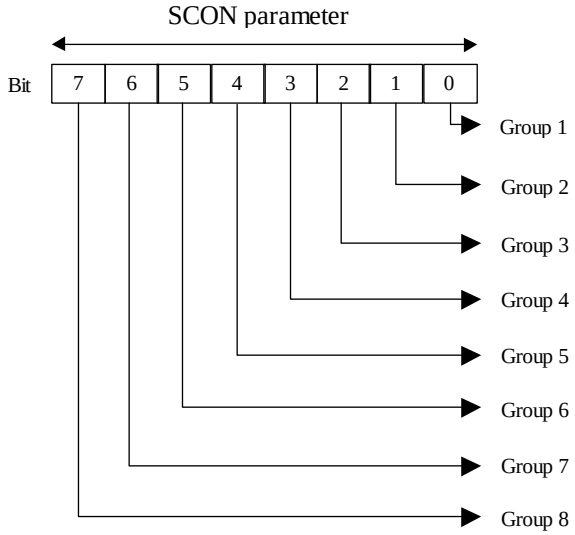
### 5.2.3.11 SCON(I) 15, 16: Global Control Service Sync, Unsync

With these instructions, the 'Sync' and 'Unsync' commands can be triggered for one or more groups of slaves. The instruction is used to synchronize the outputs.

With the 'Sync' instruction, the master causes a slave or group of slaves simultaneously to freeze outputs in their present state. In the next data cycle (Data_exch) the master transfers the output image to the slaves, without the slaves copying this image to their outputs. After the conclusion of this action, the master sends an 'Unsync' instruction to the group. All slave outputs are now switched on or off at precisely the same time and these outputs are again refreshed in the normal data cycle. It is permissible for the master, after one 'Sync' instruction, to send further 'Sync' instructions to the slaves. In each case the current output image is copied to the outputs at exactly the same time.

Diagnostic flag +1 is set high as soon as this instruction starts up. When the instruction has finished, the flag is set low and the result of the operation is written to diagnostic register +0. A description of the response code in diagnostic register +0 is given in section 5.2.1.2, 'Diagnostic registers with PROFIBUS-DP'. This instruction may only be executed when the status of diagnostic flag +1 is low.

Assigning a slave to a group takes place with the PROFIBUS-DP configurator. PROFIBUS-DP supports the formation of a maximum of 8 groups. These groups can be assigned as many slaves as required. The choice of group in the SCON parameter is bit-oriented according to the following pattern:




A 'Sync' or 'Unsync' instruction can be executed here on several groups simultaneously.

Fct. code to trigger 'Sync' or 'Unsync' instructions:

15 Start sync instruction.
16 Start unsync instruction.

**Structure:**

| | |
|---|---|
| SCON | Channel<br>Fct code<br>Parameter |

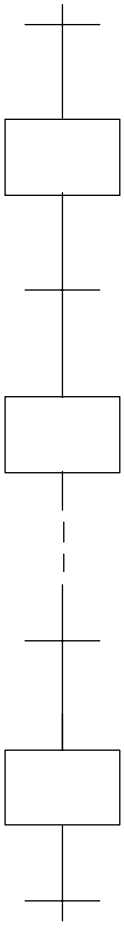
Channel 9, 8

Fct. code 15, 16 ; Fct code

Parameter 0..255 ; Group number

**Flags:** The error flag is set if the channel is unassigned or if the instruction is called when diagnostic flag +1 is high.

**Example:** Execute a 'Sync' and 'Unsync' sequence for the slaves of group 3.



```
STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue

SCON    9               ; PROFIBUS-DP channel 9
        15      ;  Sync
        4               ;   Group 3 (00000100)

STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue

OUT   F XX              ; Set outputs
                        ;   of slaves
LD    T 5               ; Load timer 5 with
      400               ;   value 400

STL   T 5               ; Wait until timer = 0

SCON    9               ; PROFIBUS-DP channel 9
        16      ;  Unsync
        4               ;   Group 3 (00000100)

STL     GCS_BUSY        ; If diagnostic flag +1
                        ;   is low, then continue
```

### 5.2.4 History list messages

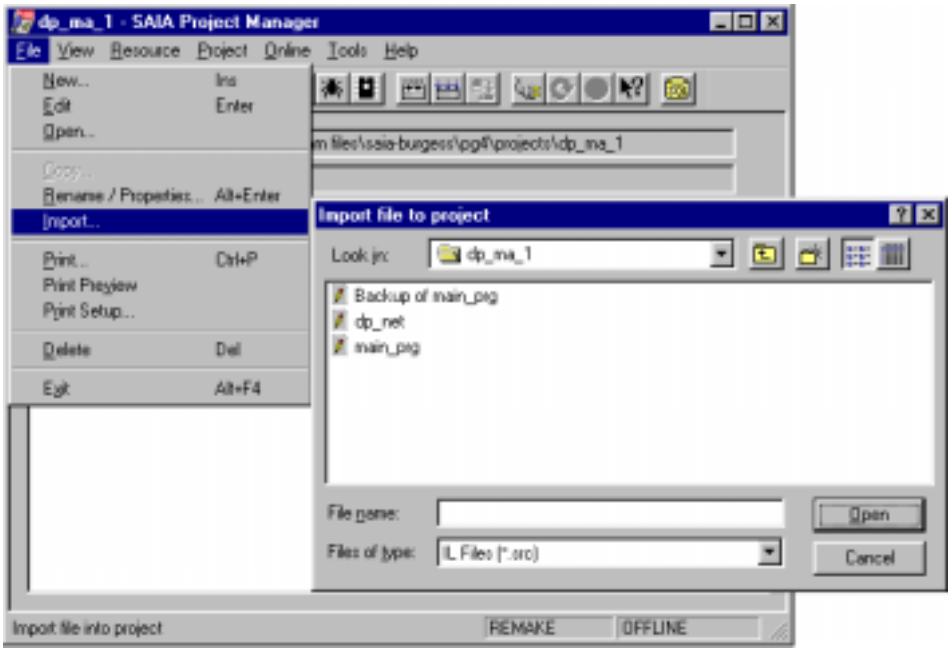
In case of problems with PROFIBUS-DP the following error message is stored in the history log:

**PROF DP FAIL xxx**

| ERR# | Description |
|---|---|
| 0 | Key word MODE: not found |
| 0 | Wrong mode specified |
| 0 | Key word CONF: not found |
| 0 | DBX key word not specified |
| 0 | DBX number error |
| 0 | DBX number to large |
| 0 | DBX does not exist |
| 0 | Key word DIAG: not found |
| 0 | Flag or output key word not specified in DIAG |
| 0 | Error in address of diag flag or output |
| 0 | Range error diag flag or output |
| 0 | Register key word not specified in DIAG |
| 0 | Range error diag register |
| 1 | PROFIBUS-DP HW card not present |
| 2 | Error in instruction |
| 3 | DBX structure error |
| 4 | DBX type not for DP master (no PROFIBUS DBX) |
| 5 | FW-DBX version not compatible |
| 6 | No IN RING message after timeout on initialization |
| 7 | Semaphore error for data exchange (info to PCD support) |
| 8 | DBX error: data transfer function not implemented |
| 9 | Incompatible PCD7.F750 and PCD hardware |

### 5.2.5 Employing PG3 user programs in PROFIBUS-DP projects

The data generated by the PROFIBUS-DP configurator can only be further processed with PG4 programming software. User programs written with the PG3 programming software can if required be integrated into the PG4 programming software.
When doing this, however, it is necessary to check that no conflict arises between the resources of files written with the PG3 and the dynamic resource management of the PG4.



## 5.3 Rules for writing the user program

To begin with, here again are the main rules for the PROFIBUS-DP user program:

- As usual with SAIA® PCD, each communications interface must be initialized with a SASI instruction.
  This normally occurs in XOB 16.

- In the master PCD, media are reserved for all I/Os and registers of PROFIBUS-DP slaves. Access to these slave I/Os and registers takes place in the master program via these master media. The master media are grouped in the image memory. If slave I/Os are accessed in the master program, it is always this image memory that is accessed, not the actual I/Os.

- Data exchange between the image memory and the PROFIBUS-DP card (and the slaves) can be automatic or controlled by the user program. Only after data exchange has been executed between the image memory and the PROFIBUS-DP card (and the slaves) will slave I/Os be read or written.

- To enable automatic data exchange between the image memory and PROFIBUS-DP card memory (and the slaves), the beginning of COB 0 (COB 0) and end of COB 0 (ECOB) must be processed. At the beginning of COB 0 slave inputs are copied from the PROFIBUS-DP card memory to the master PCD's input image memory. At the end of COB 0 the master PCD's output image memory is copied to the PROFIBUS-DP card memory.

- By means of SCON instructions, it is possible in the user program to force data exchange between the image memory and the PROFIBUS-DP card memory.

# 5.4 Structure of the user program

### 5.4.1 Cold-start program in XOB 16

In XOB 16 the PROFIBUS-DP interface is initialized with the SASI instruction. The SASI texts generated by the PROFIBUS-DP configurator are used for this purpose.

Example:

```
XOB     16

SASI    9               ; Channel 9
        txt_1DP         ; Text from configurator

EXOB
```

### 5.4.2 Main program in COB

To enable automatic data exchange between the image memory and PROFIBUS-DP card memory (and the slaves), the beginning of COB 0 (COB 0) and end of COB 0 (ECOB) must be processed. Data exchange between the image memory and the PROFIBUS-DP card memory (and the slaves) can if required be controlled by the user program or forced. This is mainly used in large user programs when reactions are needed to time-critical I/O signals from slaves.

Example 1:

Automatic data exchange between the image memory and the PROFIBUS-DP card memory in BLOCTEC structure.

```
COB     0               ; Copy slave inputs from PROFIBUS-DP
        0               ;  card memory to master PCD's input
                        ;  image memory.
STH     XX
ANL     YY              ; Code with which the slave media
OUT     ZZ              ;  are accessed

ECOB                    ; Data from output image memory
                        ;  are copied into the PROFIBUS-DP
                        ;  card memory.
```

Example 2:

Automatic data exchange and data exchanged forced by the user program between the image memory and the PROFIBUS-DP card memory in a BLOCTEC structure.

```
COB     0           ; Copy slave inputs from PROFIBUS-DP
        0           ;  card memory into the master PCD's
                    ;  input image memory.

STH     F XX
ANL     F YY        ; Code with which the slave media
OUT     F BZ        ;  are accessed

CPB     2           ; Call PB with which to copy the
                    ;  slave inputs from the PROFIBUS-DP
                    ;  card memory into the master PCD's
                    ;  input image memory

STH     F XX
ANL     F YY        ; Code with which the slave media
OUT     F AZ        ;  are accessed

ECOB                ; Data from output image memory are
                    ;  copied to the PROFIBUS-DP
                    ;  card memory


PB      2           ; PB to refresh the inputs

SCON    9           ; Channel 9
        3           ;  Fct. code 3 with parameter 1 =
        1           ;  Force data exchange for all slaves
                    ;  between input image memory and
                    ;  PROFIBUS-DP card
EPB
```

Example 3:

The slave I/Os must be coherent in a structured program with several COBs, inside a program cycle.
In order to keep the same data status throughout the entire program, the I/Os are not accessed in COB0.

```
COB      0          ; Copy slave inputs from PROFIBUS-DP
         0          ;  card memory to the master PCD's input
                    ;  image memory.
                    ;  COB 0 is only used for data exchange.
                    ;  The data provided is then processed
                    ;  by the other COBs.
ECOB                ; Data from output image memory is
                    ;  copied to PROFIBUS-DP card
                    ;  memory.


COB      1          ; Actual user program in which.
         0          ;  the slave media are accessed.

STH    F XX
ANL    F YY         ; Code with which the slave media
OUT    F AZ         ;  are accessed


ECOB                ; End of COB 1

COB      2
         0

STH    F XX
ANL    F YY         ; Code with which the slave media
ANL    F AZ         ;  are accessed
OUT    F AY

ECOB                ; End of COB 2
```

## 5.5 Program examples

### 5.5.1 Example 1

**Task:**

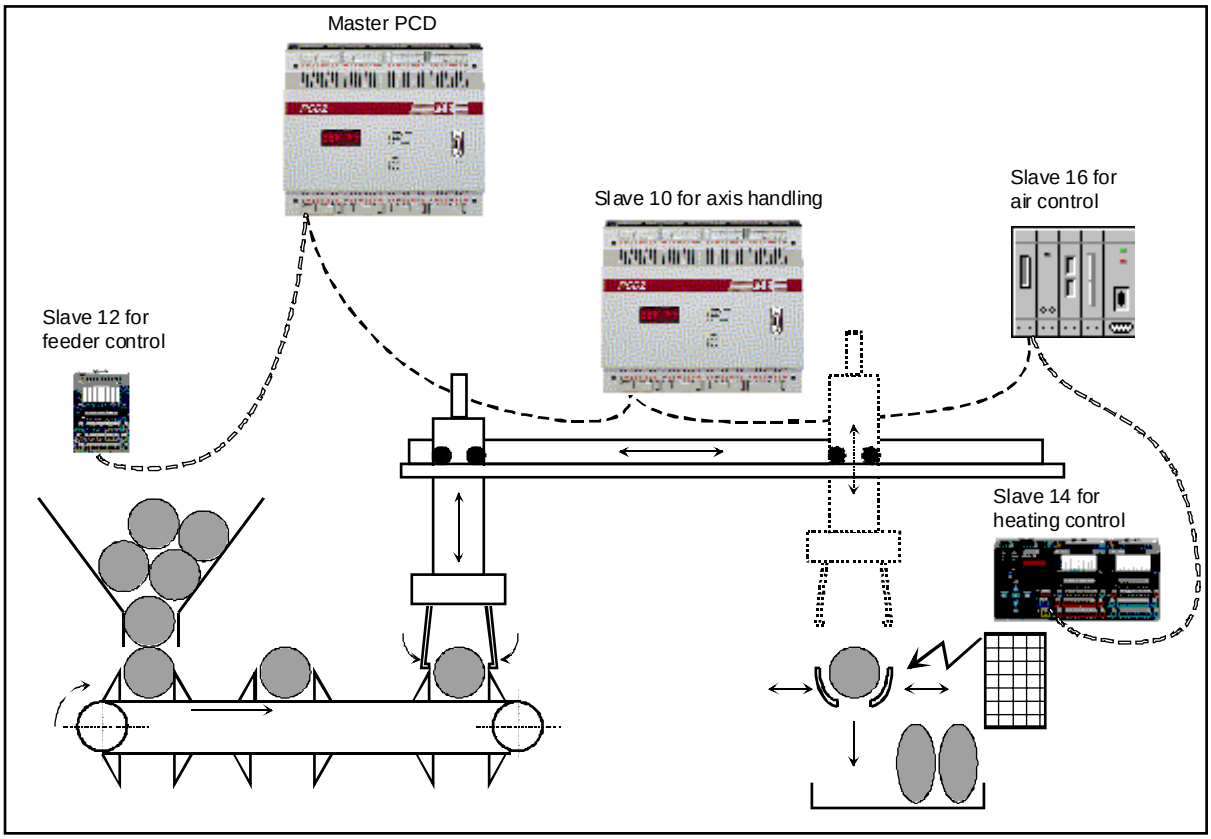
Program for master that reads an input (Emerg_st12) from slave 12 and writes the status of the input to output O 0 (air_valve) of the slave.

**Solution:**

```
XOB     16

SASI    9               ; Channel 9
        txt_1DP         ; Text from configurator

EXOB


COB     0               ; Copy slave inputs from PROFIBUS-DP
        0               ;   card memory to the master PCD's input
                        ;   image memory.

STH     Emerg_st12      ; If I 0 of slave 12 = H, then set
OUT     Air_valve       ;   O 0 of slave 16 = H.

ECOB                    ; Data from the output image memory
                        ;   are copied to the PROFIBUS-DP
                        ;   card memory.
```

### 5.5.2 Example 2

The following installation is to be automated with PROFIBUS-DP:




Function of installation:

On this machine, plastic spheres are formed into oval shapes by heating. The machine has the following functions:

Slave 12 is responsible for the feeder. In the feeder, the spheres are separated out and fed by conveyor belt to the handling device's receiving station.

Slave 10 is responsible for the handling and controls two pneumatic cylinders (horizontal and vertical) which convey the plastic spheres from the feeder to the heating station. The whole handling sequence is controlled directly in the slave.

The handling tongs are also controlled by station 10.

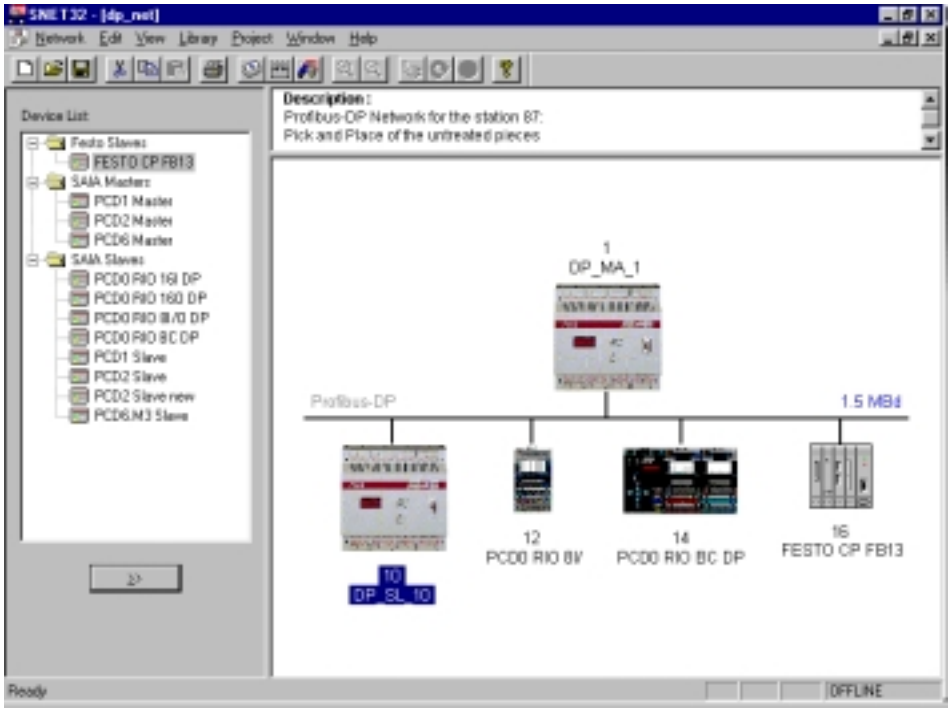
Slave 14 is responsible for forming the plastic spheres with heating tongs and a heating regulator. The formed part is finally ejected into a container by opening the heating tongs.

Slave 16 is responsible for the provision of compressed air.

The following devices are used in the installation:

- A PCD2 as master with: PROFIBUS address 1
  1 input module at addresses 0..7
  1 output module at addresses 64..71
  The PG4 programs are stored in path:
  ..\PG4\Projects\dp_ma_1
- A PCD2 as slave with: PROFIBUS address 10
  1 input module at addresses 16..23
  1 output module at addresses 32..39
  The PG4 programs are stored in path:
  ..\PG4\Projects\dp_sl_10
- A PCD0.G110, RIO 8 I/O DP with PROFIBUS address 12
  8 I/Os
- A PCD0.T770, RIO BC DP with: PROFIBUS address 14
  First module: PCD0.E120, RIO 16I
  Second module: PCD0.A410, RIO 16O
  Third module: PCD0.B120, RIO 8I 8I/O
  Fourth module: PCD0.W710, RIO 4AI/4AO ±10VDC
- A Festo valve island CP-FB13-E with: PROFIBUS address 16
  Branch 0 with 16 inputs and 16 outputs

The result will appear as follows with the PROFIBUS-DP configurator:



This involves defining the following variables in the PROFIBUS-DP configurator:

| Station | Module / Bit | Message | Media Slave | Media Master | Symbol Name |
|---|---|---|---|---|---|
| 10 | 0 / 0 | M->S | F ? | F ? | Emrg_Stop |
| 10 | 0 / 1 | M->S | F ? | F ? | Start_x |
| 10 | 0 / 2 | M->S | F ? | F ? | Start_z |
| 10 | 0 / 3 | M->S | F ? | F ? | Reset_cnt |
| 10 | 0 / 4 | M->S | F ? | F ? | Heat_ok |
| 10 | 0 / 5 | M->S | F ? | F ? | Speed_1 |
| 10 | 0 / 6 | M->S | F ? | F ? | Speed_2 |
| 10 | 0 / 7 | M->S | F ? | F ? | Open_Grid |
| 10 | 1 / 0 | S->M | F ? | F ? | Pce_in_pos |
| 10 | 1 / 1 | S->M | F ? | F ? | Job_end |
| 10 | 1 / 2 | S->M | F ? | F ? | X_ismoving |
| 10 | 1 / 3 | S->M | F ? | F ? | free_10_3 |
| 10 | 1 / 4 | S->M | F ? | F ? | free_10_4 |
| 10 | 1 / 5 | S->M | F ? | F ? | free_10_5 |
| 10 | 1 / 6 | S->M | F ? | F ? | free_10_6 |
| 10 | 1 / 7 | S->M | F ? | F ? | free_10_7 |
| 10 | 2 / 0 | S->M | I 0 | F 1000 | Rest_Stop |
| 10 | 2 / 1 | S->M | I 1 | F 1001 | Limt_x_lef |
| 10 | 2 / 2 | S->M | I 2 | F 1002 | Limt_x_rig |
| 10 | 2 / 3 | S->M | I 3 | F 1003 | Limt_z_up |
| 10 | 2 / 4 | S->M | I 4 | F 1004 | Limt_z_dwn |
| 10 | 2 / 5 | S->M | I 5 | F 1005 | Posok_feed |
| 10 | 2 / 6 | S->M | I 6 | F 1006 | Posok_heat |
| 10 | 2 / 7 | S->M | I 7 | F 1007 | Emerg_st10 |
| 10 | 3 / 0 | M->S | R ? | R ? | Nbr_pieces |
| 10 | 4 / 0 | S->M | R ? | R ? | Nbr_act_pi |
| 10 | 4 / 1 | S->M | R ? | R ? | New_pos_x |
| 12 | 0 / 0 | S<->M | I/O 0 | F ? | free_12_0 |
| 12 | 0 / 1 | S<->M | I/O 0 | F ? | free_12_1 |
| 12 | 0 / 2 | S<->M | I/O 0 | F ? | free_12_2 |
| 12 | 0 / 3 | S<->M | I/O 0 | F ? | free_12_3 |
| 12 | 0 / 4 | M->S | O 4 | F ? | Vibra_on |
| 12 | 0 / 5 | M->S | O 5 | F ? | Lamp_ok |
| 12 | 0 / 6 | M->S | O 6 | F ? | Lamp_nok |
| 12 | 0 / 7 | M->S | O 7 | F ? | Belt_on |
| 12 | 0 / 8 | S->M | I 0 | F ? | Emerg_st12 |
| 12 | 0 / 9 | S->M | I 1 | F ? | Feed_void |
| 12 | 0 / 10 | S->M | I 2 | F ? | Stack_void |
| 12 | 0 / 11 | S->M | I 3 | F ? | Stack_full |
| 12 | 0 / 12 | S<->M | | F ? | free_12_12 |
| 12 | 0 / 13 | S<->M | | F ? | free_12_13 |
| 12 | 0 / 14 | S<->M | | F ? | free_12_14 |
| 12 | 0 / 15 | S<->M | I 0 | F ? | free_12_15 |

| | | | | | |
|---|---|---|---|---|---|
| 14 | 0 / 0 | S->M | R ? | R ? | diag_i_14 |
| 14 | 0 / 1 | M->S | R ? | R ? | diag_o_14 |
| 14 | 1 / 0 | S->M | I 0 | F ? | Heat_is_0 |
| 14 | 1 / 1 | S->M | I 1 | F ? | Heat_is_1 |
| 14 | 1 / 2 | S->M | I 2 | F ? | Heat_is_2 |
| 14 | 1 / 3 | S->M | I 3 | F ? | Heat_is_3 |
| 14 | 1 / 4 | S->M | I 4 | F ? | Heat_is_4 |
| 14 | 1 / 5 | S->M | I 5 | F ? | Heat_is_5 |
| 14 | 1 / 6 | S->M | I 6 | F ? | Heat_is_6 |
| 14 | 1 / 7 | S->M | I 7 | F ? | Heat_is_7 |
| 14 | 1 / 8 | S->M | I 8 | F ? | Heat_is_8 |
| 14 | 1 / 9 | S->M | I 9 | F ? | Heat_is_9 |
| 14 | 1 / 10 | S->M | I 10 | F ? | Heat_is_10 |
| 14 | 1 / 11 | S->M | I 11 | F ? | Heat_is_11 |
| 14 | 1 / 12 | S->M | I 12 | F ? | Heat_is_12 |
| 14 | 1 / 13 | S->M | I 13 | F ? | Heat_is_13 |
| 14 | 1 / 14 | S->M | I 14 | F ? | Heat_is_14 |
| 14 | 1 / 15 | S->M | I 15 | F ? | Heat_is_15 |
| 14 | 2 / 0 | M->S | O 0 | F ? | Heat_os_0 |
| 14 | 2 / 1 | M->S | O 1 | F ? | Heat_os_1 |
| 14 | 2 / 2 | M->S | O 2 | F ? | Heat_os_2 |
| 14 | 2 / 3 | M->S | O 3 | F ? | Heat_os_3 |
| 14 | 2 / 4 | M->S | O 4 | F ? | Heat_os_4 |
| 14 | 2 / 5 | M->S | O 5 | F ? | Heat_os_5 |
| 14 | 2 / 6 | M->S | O 6 | F ? | Heat_os_6 |
| 14 | 2 / 7 | M->S | O 7 | F ? | Heat_os_7 |
| 14 | 2 / 8 | M->S | O 8 | F ? | Heat_os_8 |
| 14 | 2 / 9 | M->S | O 9 | F ? | Heat_os_9 |
| 14 | 2 / 10 | M->S | O 10 | F ? | Heat_os_10 |
| 14 | 2 / 11 | M->S | O 11 | F ? | Heat_os_11 |
| 14 | 2 / 12 | M->S | O 12 | F ? | Heat_os_12 |
| 14 | 2 / 13 | M->S | O 13 | F ? | Heat_os_13 |
| 14 | 2 / 14 | M->S | O 14 | F ? | Heat_os_14 |
| 14 | 2 / 15 | M->S | O 15 | F ? | Heat_os_15 |
| 14 | 3 / 0 | S->M | I 0 | F ? | Emerg_st14 |
| 14 | 3 / 1 | S->M | I 1 | F ? | Piece_okh |
| 14 | 3 / 2 | S->M | I 2 | F ? | Clamb_open |
| 14 | 3 / 3 | S->M | I 3 | F ? | Clamb_clos |
| 14 | 3 / 4 | S->M | I 4 | F ? | Air_ok |
| 14 | 3 / 5 | S->M | I 5 | F ? | Start_heat |
| 14 | 3 / 6 | S->M | I 6 | F ? | Free_14_6 |
| 14 | 3 / 7 | S->M | I 7 | F ? | Free_14_7 |
| 14 | 3 / 8 | S->M | I 8 | F ? | Close_clam |
| 14 | 3 / 9 | S->M | I 9 | F ? | Open_clamb |
| 14 | 3 / 10 | S->M | I / O 10 | F ? | free_14_18 |
| 14 | 3 / 11 | S->M | I / O 11 | F ? | free_14_19 |
| 14 | 3 / 12 | S->M | I / O 12 | F ? | free_14_20 |
| 14 | 3 / 13 | M->S | O 13 | F ? | Heat_great |
| 14 | 3 / 14 | M->S | O 14 | F ? | Heat_less |
| 14 | 3 / 15 | M->S | O 15 | F ? | Handl_work |
| 14 | 4 / 0 | S->M | I O | R ? | Heat_in_0 |
| 14 | 4 / 1 | S->M | I 1 | R ? | Heat_in_1 |
| 14 | 4 / 2 | S->M | I 2 | R ? | Heat_in_2 |
| 14 | 4 / 3 | S->M | I 3 | R ? | Heat_in_3 |
| 14 | 4 / 4 | M->S | O 1 | R ? | Heat_out_0 |
| 14 | 4 / 5 | M->S | O 2 | R ? | Heat_out_1 |
| 14 | 4 / 6 | M->S | O 3 | R ? | Heat_out_2 |
| 14 | 4 / 7 | M->S | O 4 | R ? | Heat_out_3 |

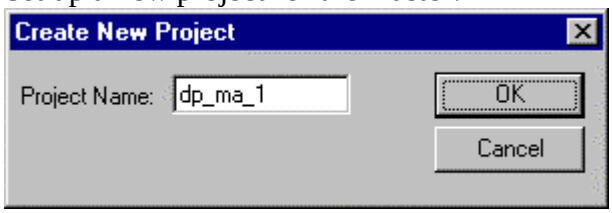
| | | | | | |
|---|---|---|---|---|---|
| 16 | 0 / 0 | S->M | I 0 | F ? | Air_start |
| 16 | 0 / 1 | S->M | I 1 | F ? | Air_P_Ok |
| 16 | 0 / 2 | S->M | I 2 | F ? | Air_Oil_Ok |
| 16 | 0 / 3 | S->M | I 3 | F ? | free_16_3 |
| 16 | 0 / 4 | S->M | I 4 | F ? | free_16_4 |
| 16 | 0 / 5 | S->M | I 5 | F ? | free_16_5 |
| 16 | 0 / 6 | S->M | I 6 | F ? | free_16_6 |
| 16 | 0 / 7 | S->M | I 7 | F ? | free_16_7 |
| 16 | 0 / 8 | S->M | I 8 | F ? | free_16_8 |
| 16 | 0 / 9 | S->M | I 9 | F ? | free_16_9 |
| 16 | 0 / 10 | S->M | I 10 | F ? | free_16_10 |
| 16 | 0 / 11 | S->M | I 11 | F ? | free_16_11 |
| 16 | 0 / 12 | S->M | I 12 | F ? | free_16_12 |
| 16 | 0 / 13 | S->M | I 13 | F ? | free_16_13 |
| 16 | 0 / 14 | S->M | I 14 | F ? | free_16_14 |
| 16 | 0 / 15 | S->M | I 15 | F ? | free_16_15 |
| 16 | 0 / 16 | M->S | O 0 | F ? | free_16_16 |
| 16 | 0 / 17 | M->S | O 1 | F ? | free_16_17 |
| 16 | 0 / 18 | M->S | O 2 | F ? | free_16_18 |
| 16 | 0 / 19 | M->S | O 3 | F ? | free_16_19 |
| 16 | 0 / 20 | M->S | O 4 | F ? | free_16_20 |
| 16 | 0 / 21 | M->S | O 5 | F ? | Air_valve |
| 16 | 0 / 22 | M->S | O 6 | F ? | Air_ready |
| 16 | 0 / 23 | M->S | O 7 | F ? | Air_nready |
| 16 | 0 / 24 | M->S | O 8 | F ? | free_16_24 |
| 16 | 0 / 25 | M->S | O 9 | F ? | free_16_25 |
| 16 | 0 / 26 | M->S | O 10 | F ? | free_16_26 |
| 16 | 0 / 27 | M->S | O 11 | F ? | free_16_27 |
| 16 | 0 / 28 | M->S | O 12 | F ? | free_16_28 |
| 16 | 0 / 29 | M->S | O 13 | F ? | free_16_29 |
| 16 | 0 / 30 | M->S | O 14 | F ? | free_16_30 |
| 16 | 0 / 31 | M->S | O 15 | F ? | free_16_31 |

A question mark (?) by the media address means that these media addresses are assigned automatically by the PG4.
Media with the symbol name free_*_* are reserve I/Os

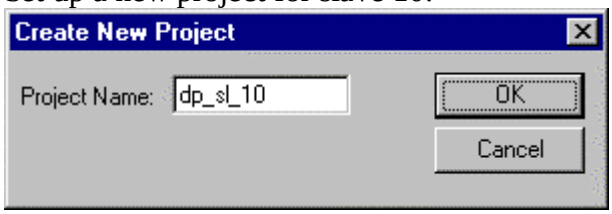
#### 5.5.2.1 Creating the structure of the example project

- Start PG4.
- Configuration of PROFIBUS-DP network project.
  The master and all slaves are configured in this network.
- Create projects for the master and slave 10 in the PG4's project library.
  In these projects, user programs are created for the controllers. None of the other slaves require user programs, since they are non-intelligent slaves, i.e. these slaves have a standard-PROFIBUS-DP program to communicate with the master and cannot process any project-specific user program.

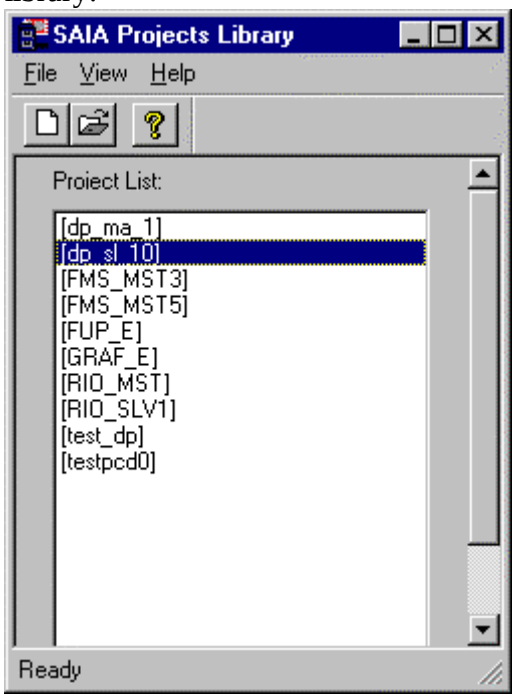
Set up a new project for the master:




Set up a new project for slave 10:

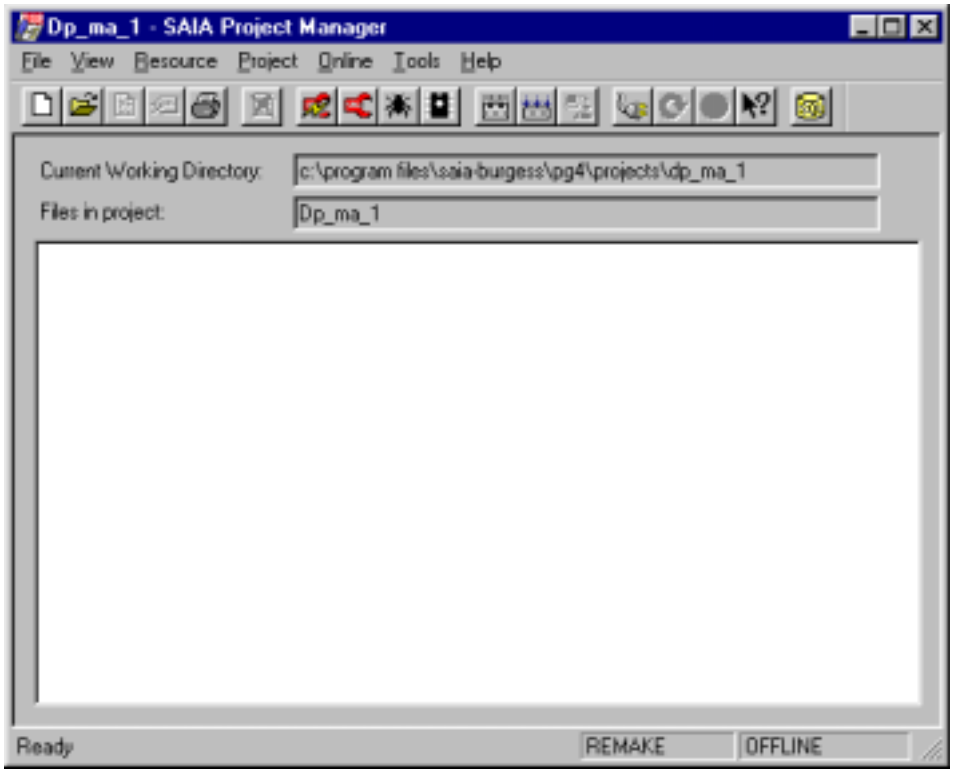



After both new projects have been created, they appear in the project library:

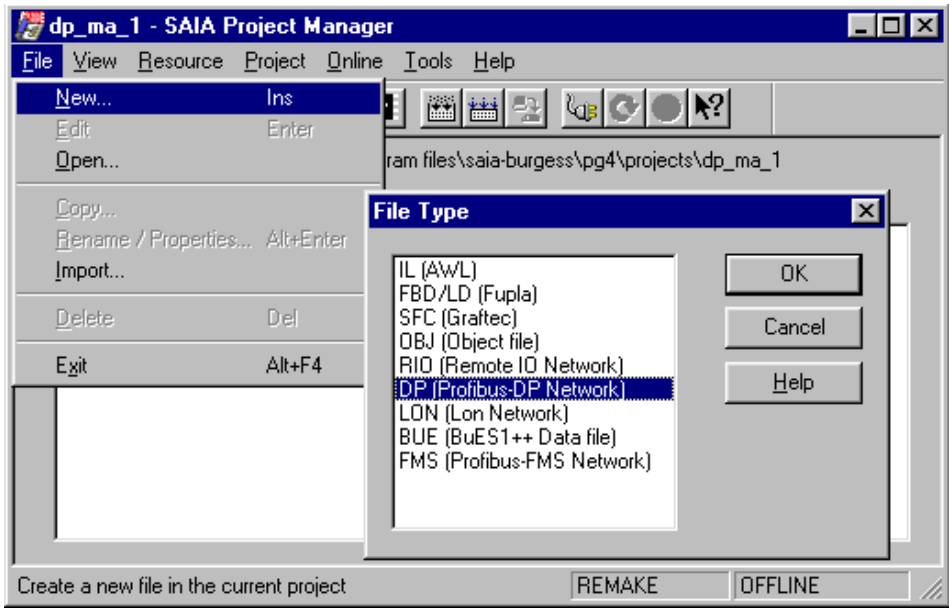


In this example, the PROFIBUS-DP network project is stored in the master directory.

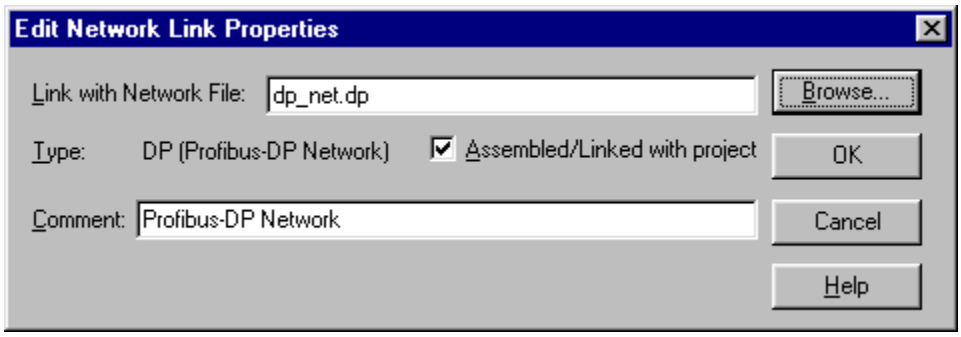
Call the project manager for project 'dp_ma_1' from the project library:

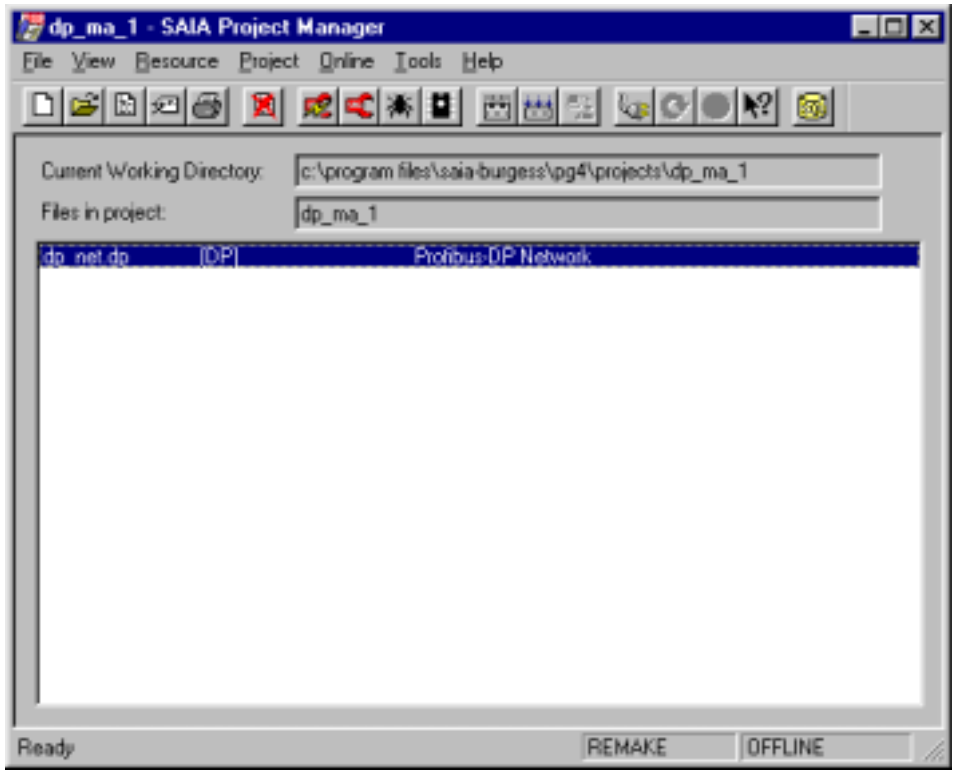



Enter the PROFIBUS-DP network project. Select 'File '- 'New', then choose DP (PROFIBUS-DP Network):



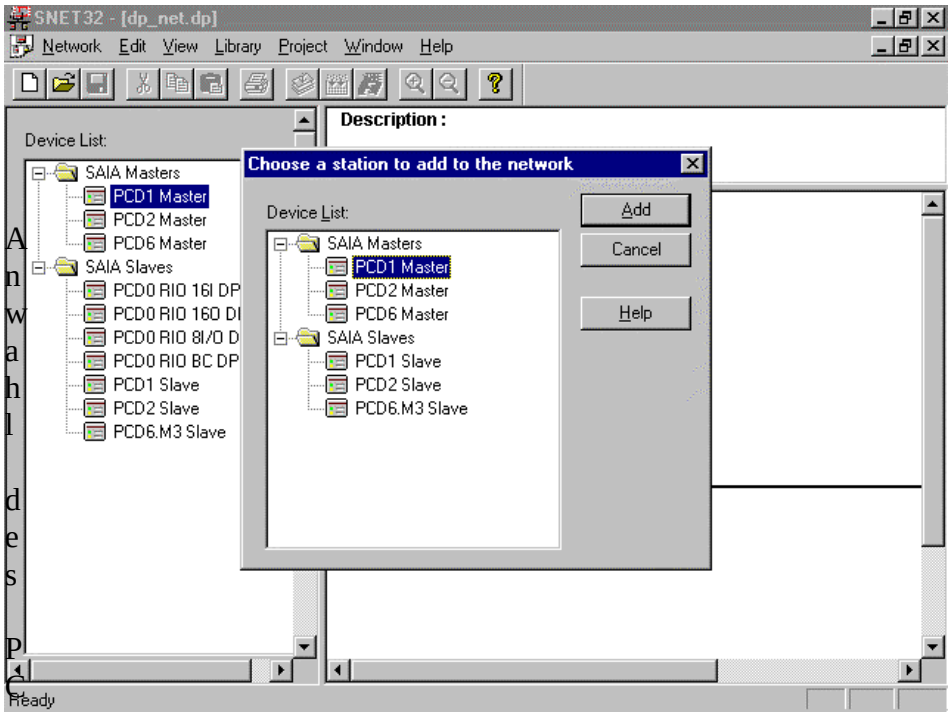
Enter the name of the PROFIBUS-DP network. The network file should be stored in the master project library. If a network includes intelligent SAIA slaves, all slaves access the same network.



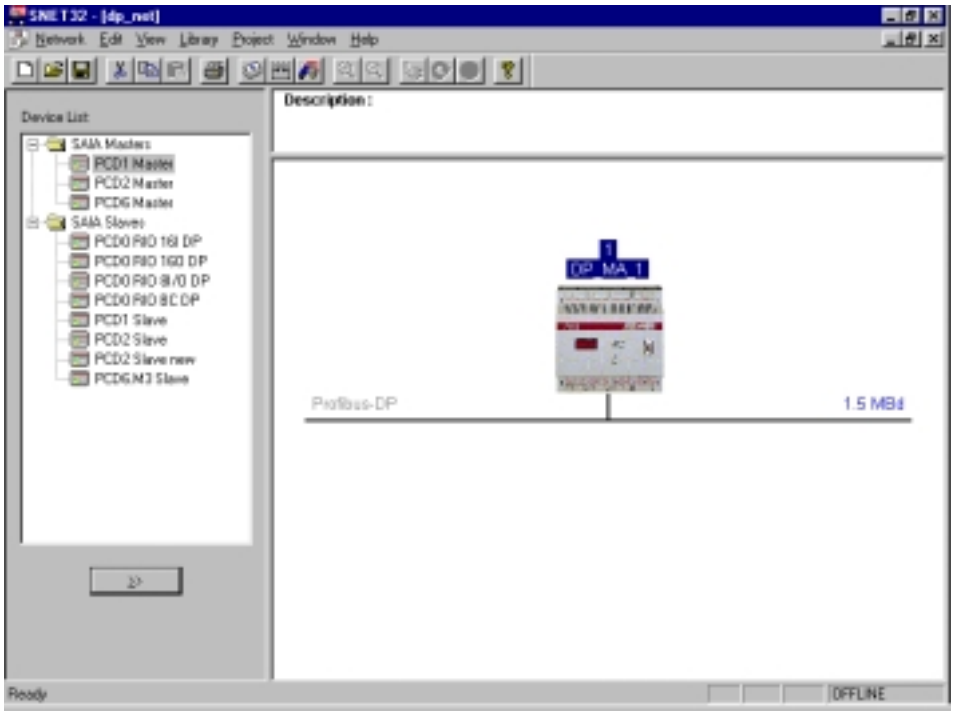



### 5.5.2.2 Configuration of the network

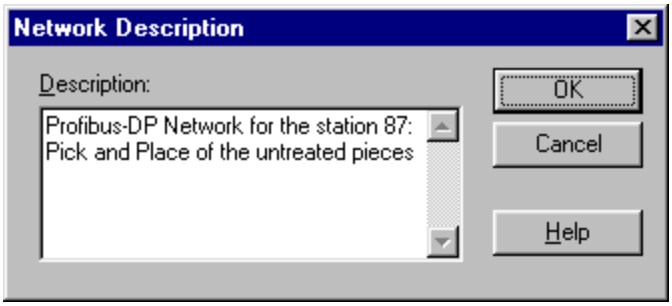
Start PROFIBUS-DP configurator:
Double-click on file 'dp_net.dp'. The PROFIBUS-DP configurator is loaded.
If no SAIA master or intelligent SAIA slave has yet been defined, an entry prompt appears on the screen to choose one of the SAIA devices.



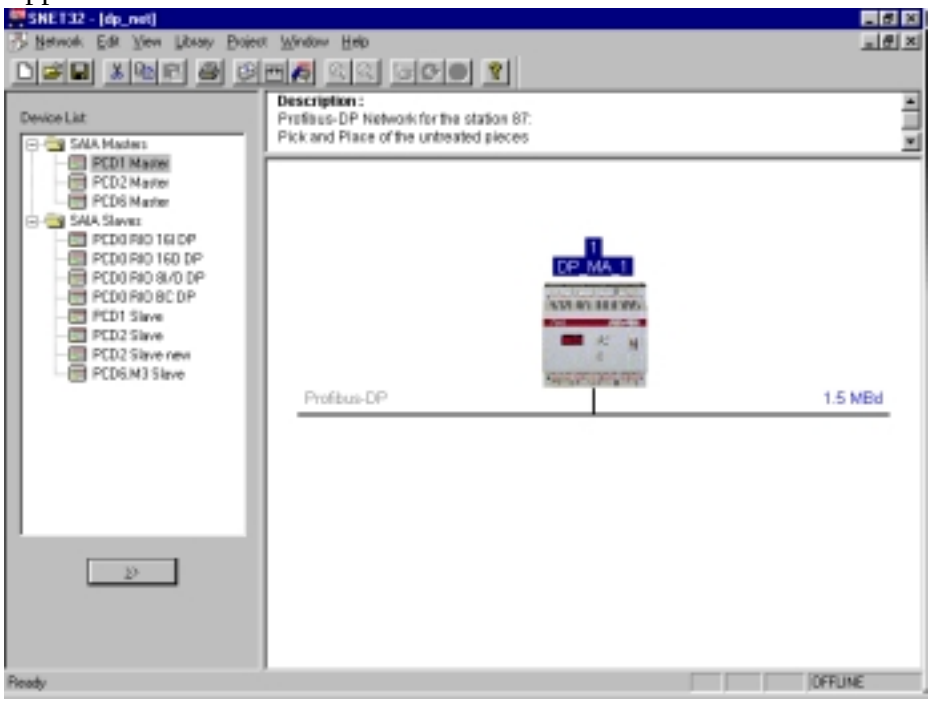

Choice of PCD2 master:



Enter the network comment by double-clicking on the 'Description' field:



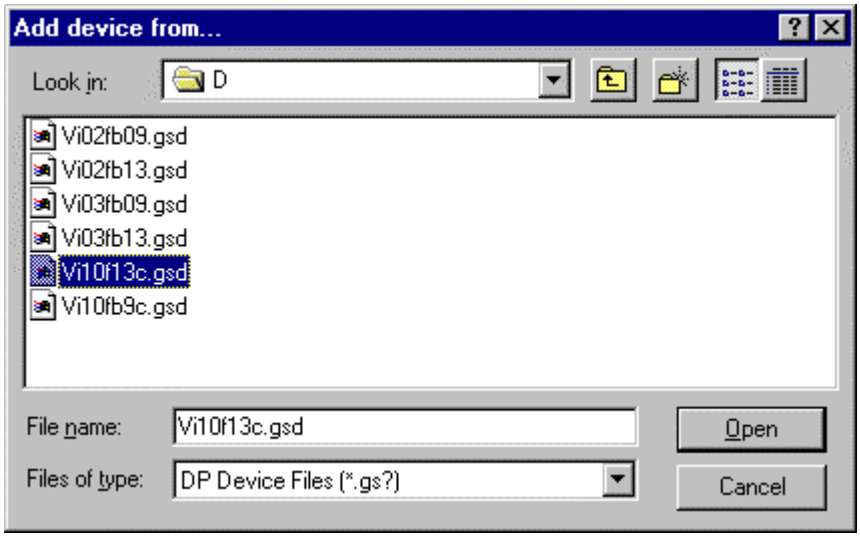

Appearance of test:




The 'Festo' valve island does not yet feature in the device list.
Its registration in the device list occurs when the '*.gs*' file of the relevant slave is loaded.

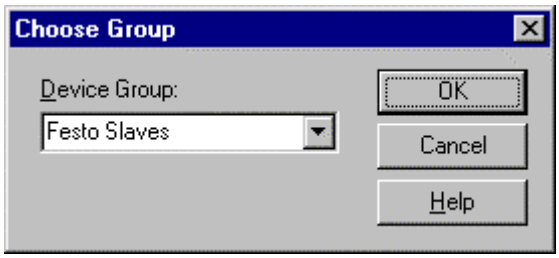
Select the 'Library' menu to insert a new slave in the device list.

Choice of corresponding '*.GS*' file.

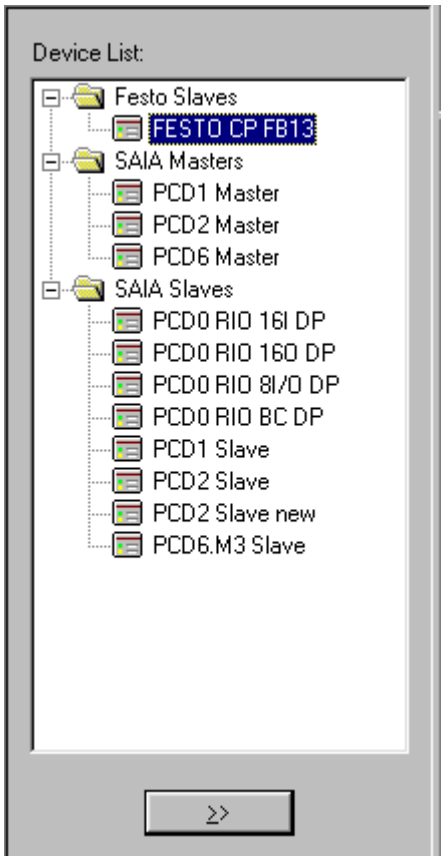



The slave can then be assigned to an existing device group, or a new device group can be formed.

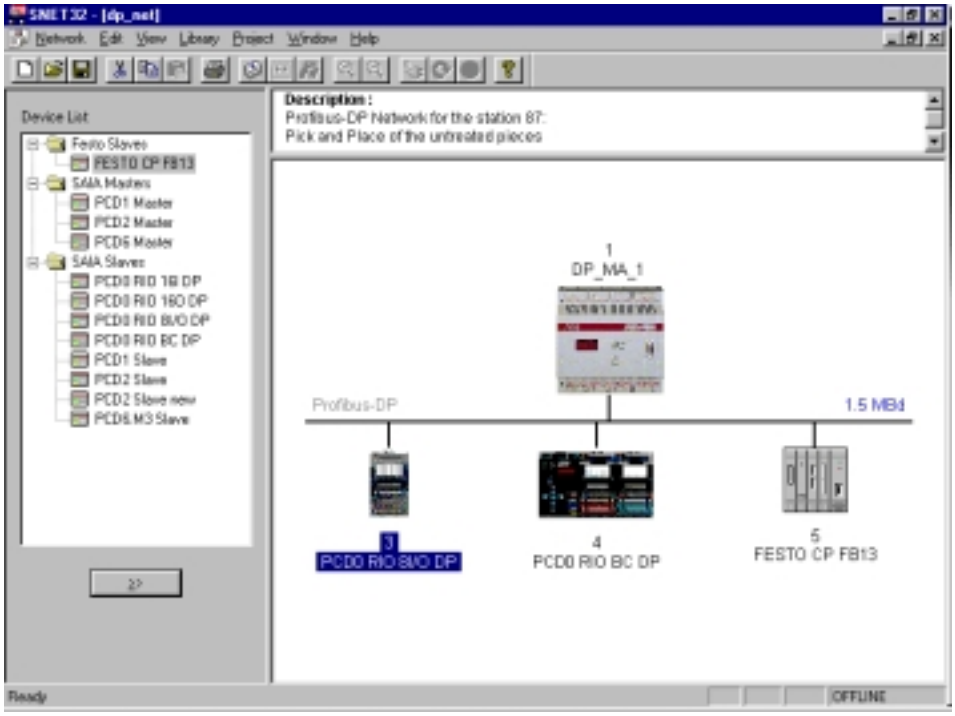
To form a new device group, it is simply necessary to enter the name of the group in the entry field.



The valve island has been included in the device list under device group 'Festo Slaves'.

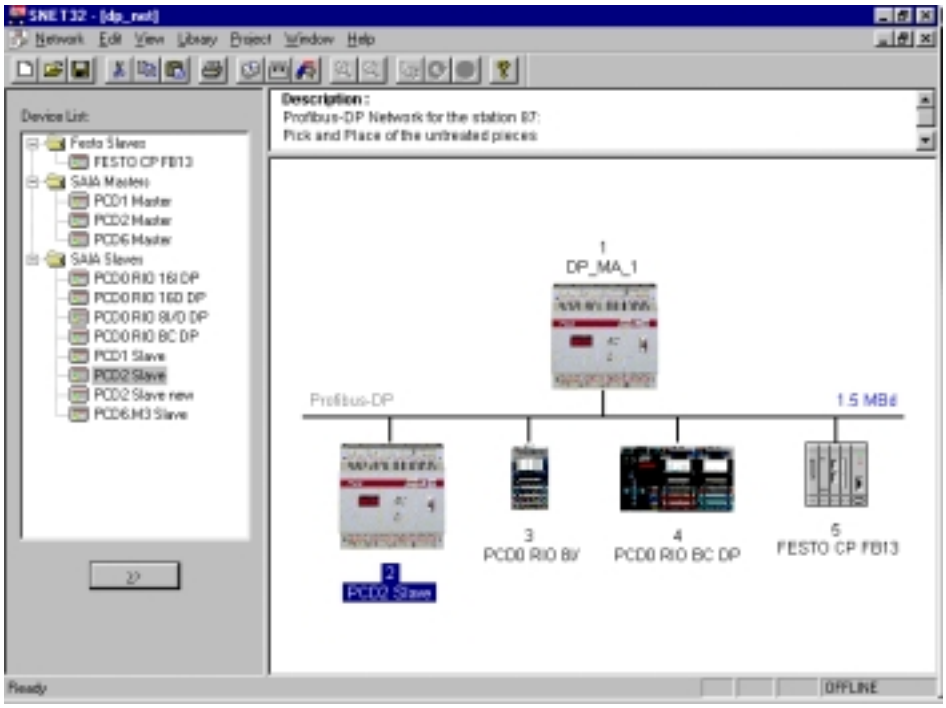



Configuration of the network by inserting slaves.
The slaves are automatically provided with a serial PROFIBUS-DP station number.



The intelligent PCD2 slave can be added in two differend ways:

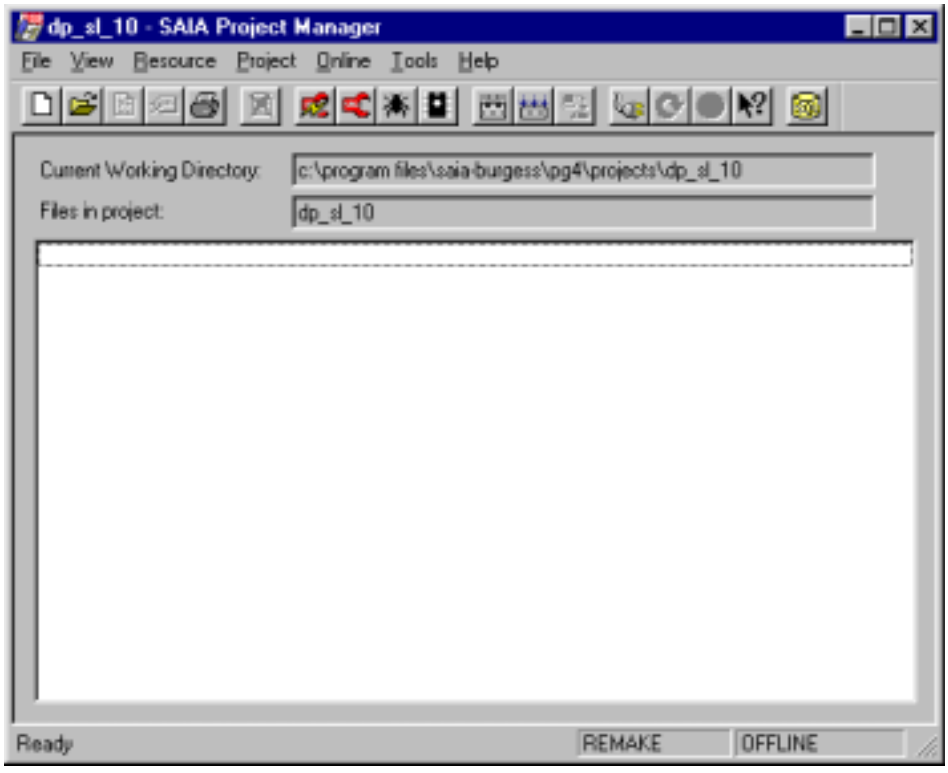
- Insertion in the existing network project by selection from the device list:



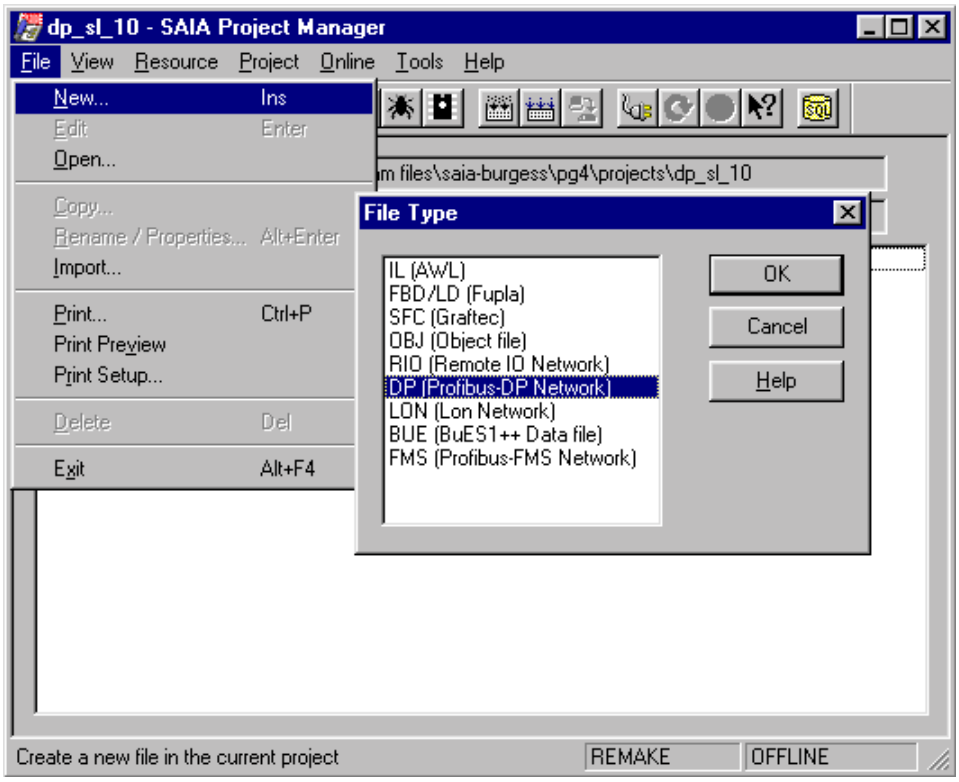

or

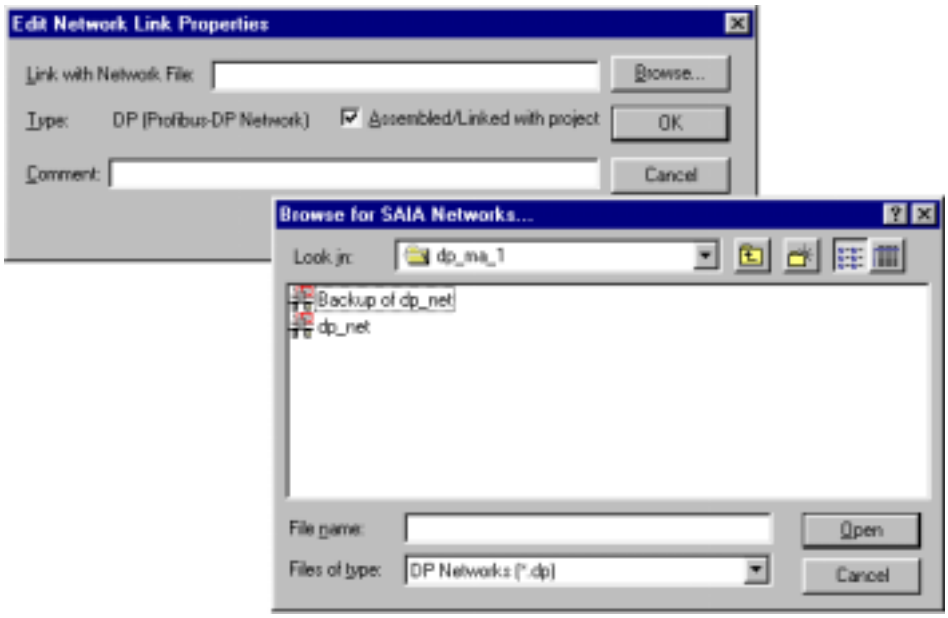
- Calling the PG4 slave project.
  In this project, reference is then made to the network project in the master:

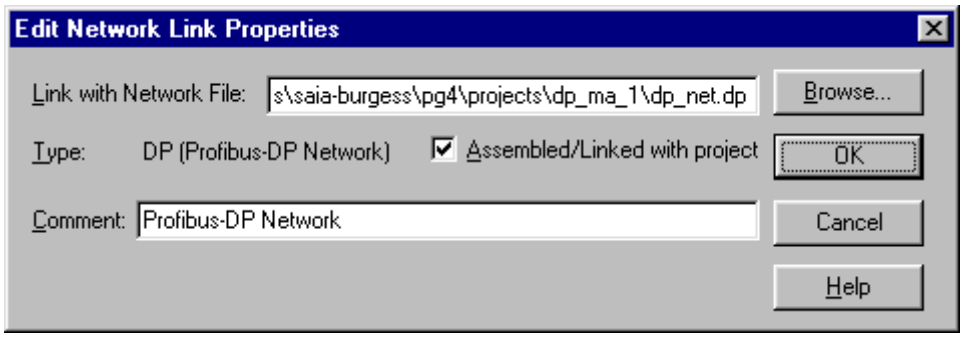
Call PG4 slave project dp_sl_10:

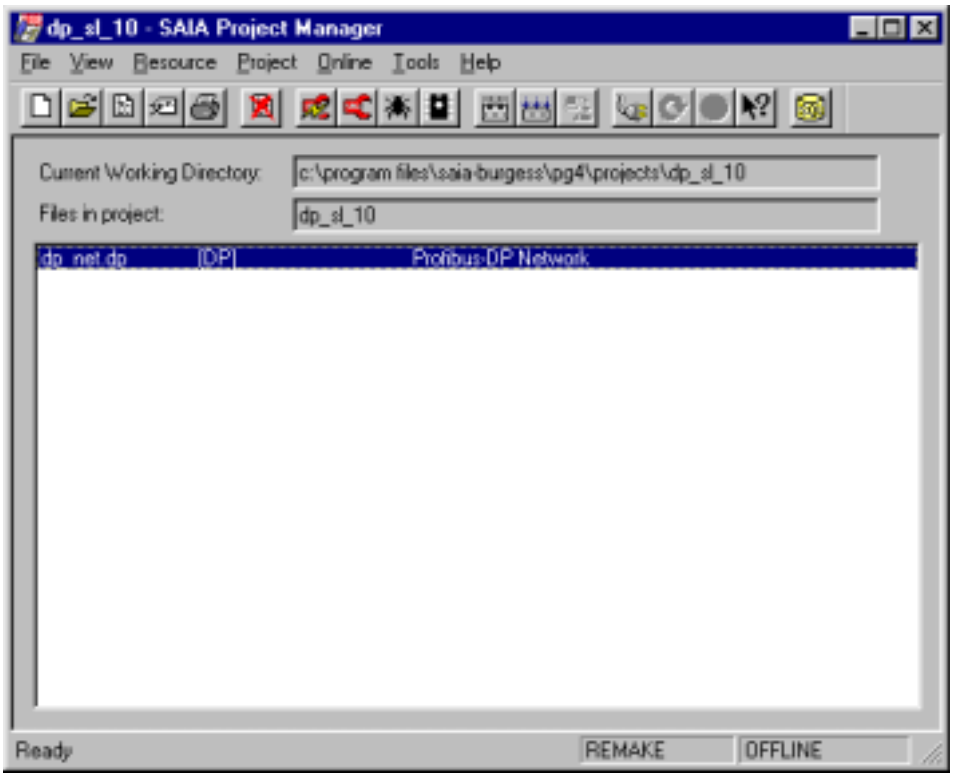


Insertion of PROFIBUS-DP network with ‘File New‘.
This involves referring to the existing network project dp_net in the master.

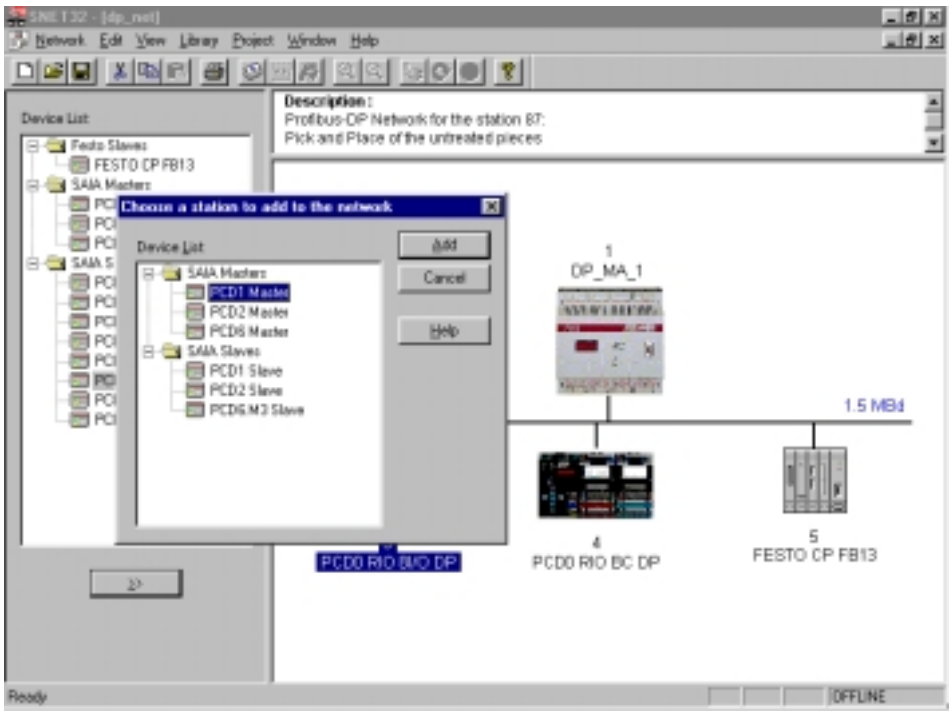


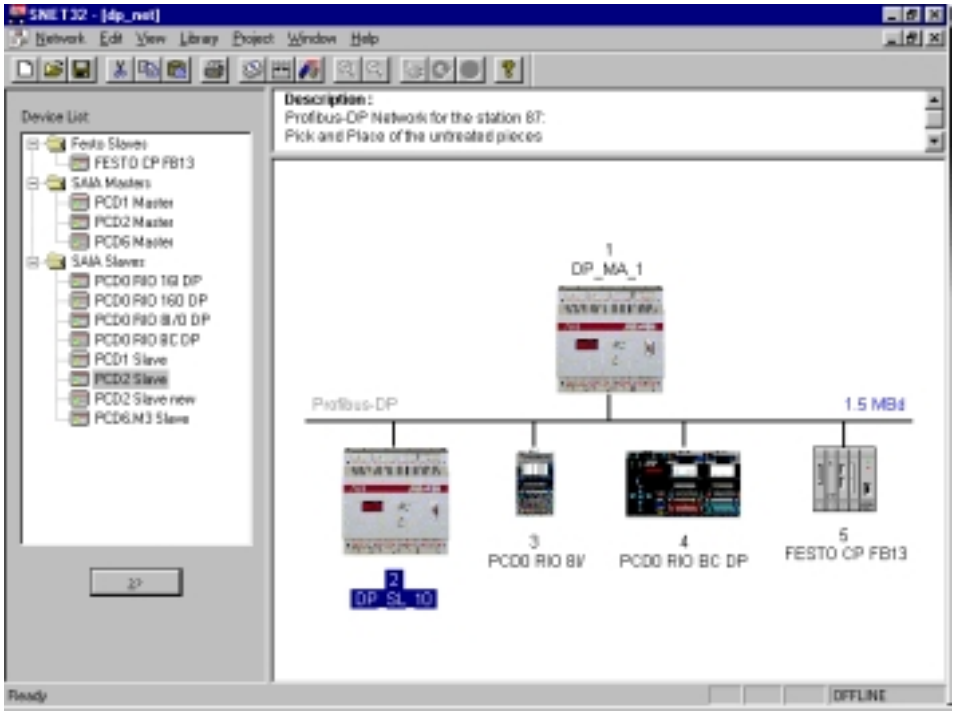





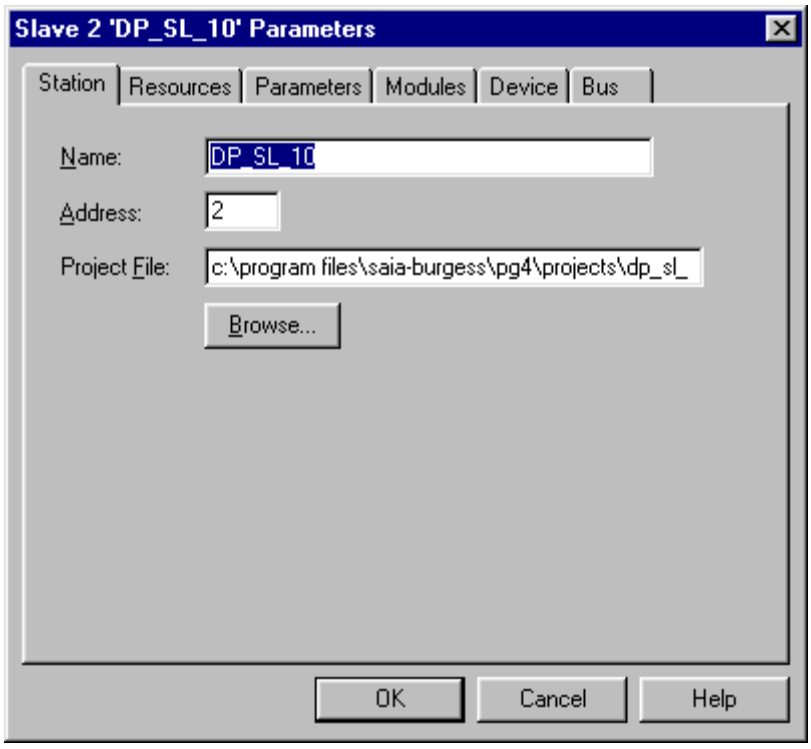

Call PROFIBUS-DP network.
When the PROFIBUS-DP network is called, a check is made that the project calling the configurator is already present with a station in the network.
If no PROFIBUS-DP station in the network has yet been defined by the calling project, an entry prompt appears to choose a SAIA master or slave.



Choice of PCD2 slaves from the device list.
The slave designation of the PG4 project is automatically adopted here.



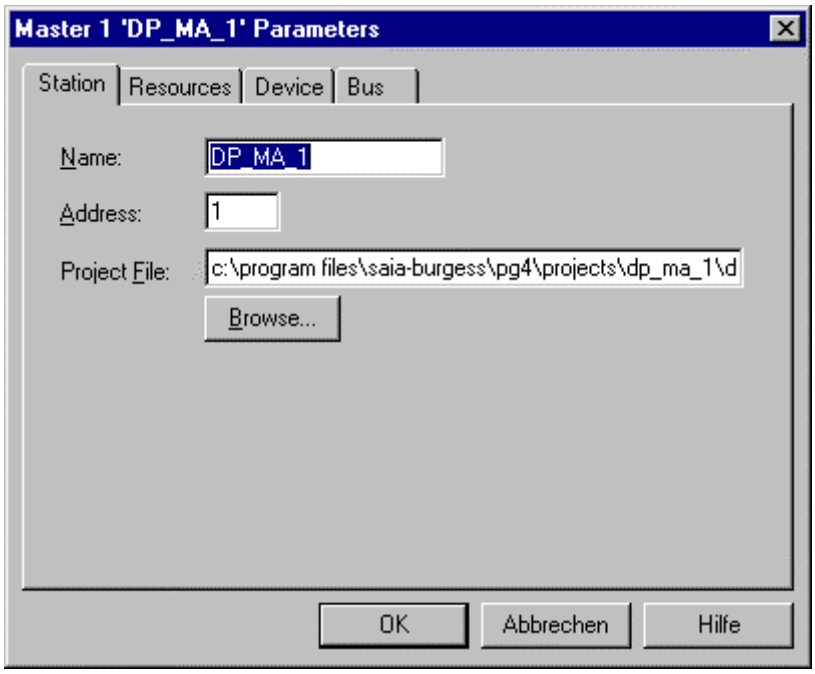

The project belonging to the slave has also been adopted.



#### 5.5.2.3 Setting parameters for the stations

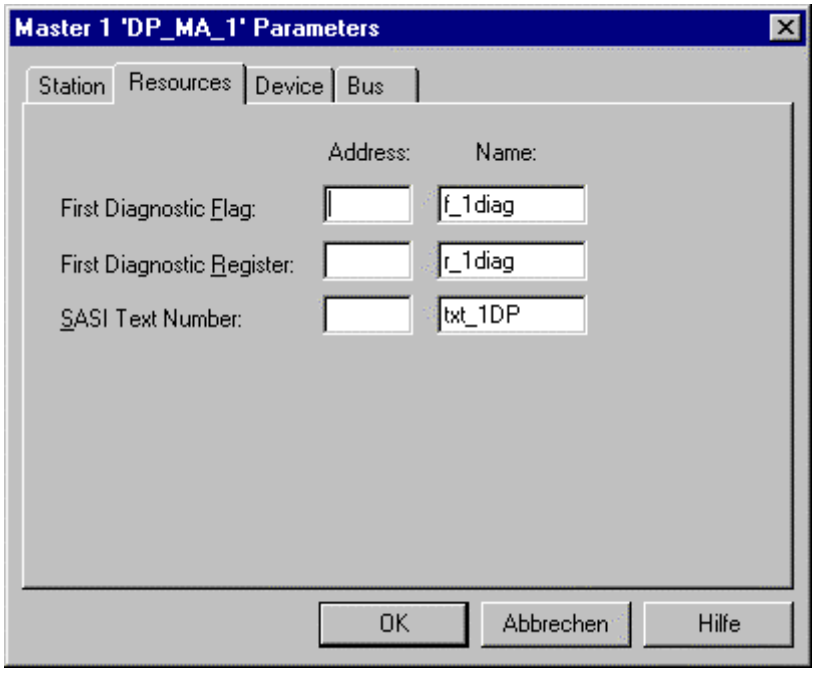
Setting parameters for the master

Double-click on the master device:




The address is correct and does not have to be changed. The project file was entered previously when the PROFIBUS-DP configurator was started up and must not be changed.
Files generated by the configurator are stored in this project file.

Choice of resources:

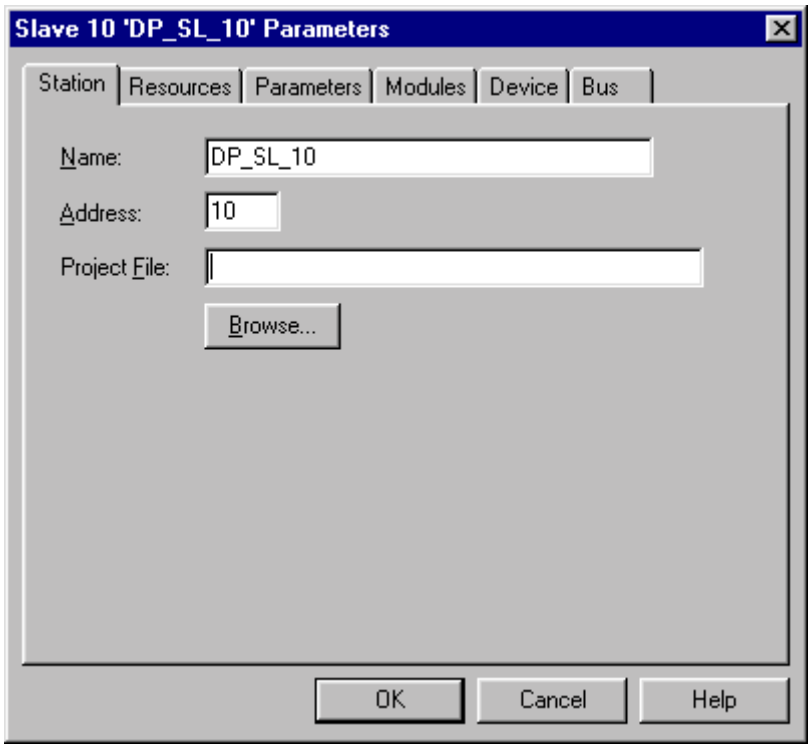



These have also been set correctly and do not have to be adusted.
This completes the setting of master parameters.

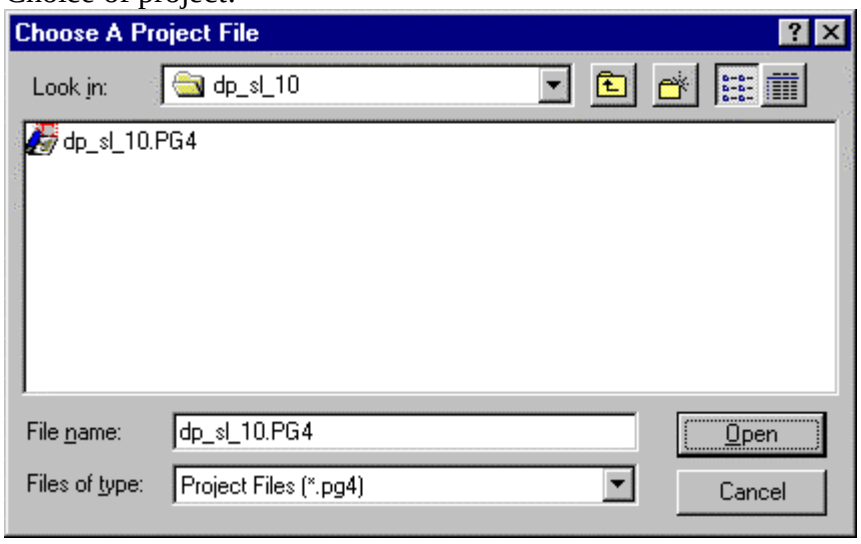
Setting parameters for slave 10

Double-click on the slave PCD2 with address 2:

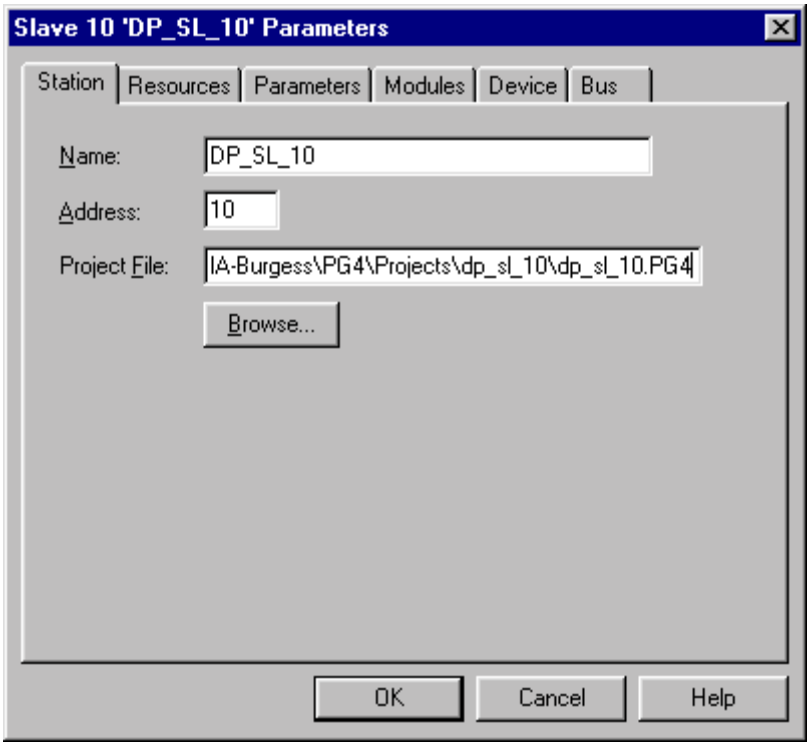
Address 2 was assigned automatically by the configurator. This must be changed to 10.
If it is not already present, the project file belonging to the slave must also be entered in the 'Project File' field.
Using the 'Browse' button, a search can be made for the project file.
It is also possible to generate a new station in the network project from an existing project file.



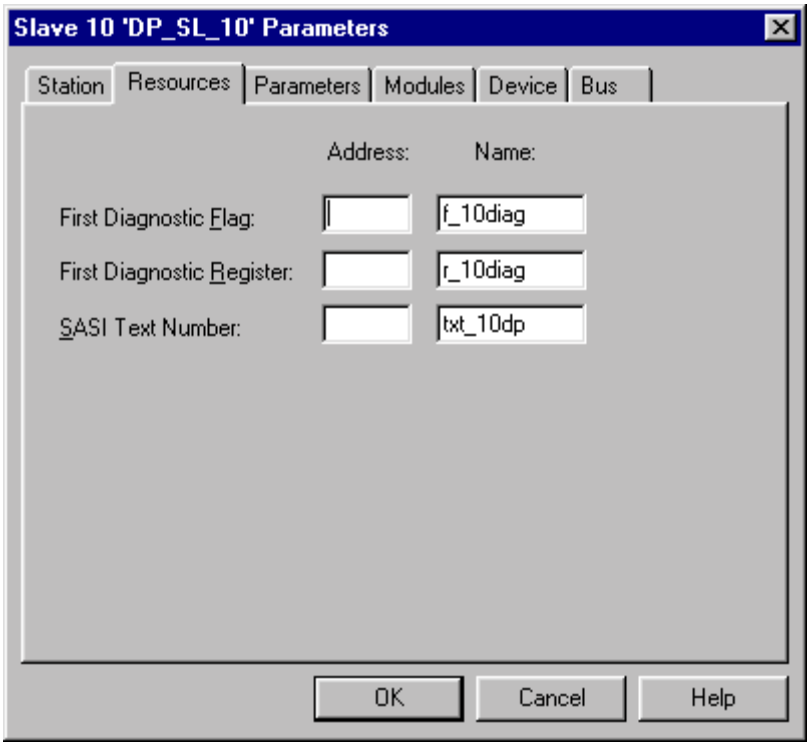

Choice of project:





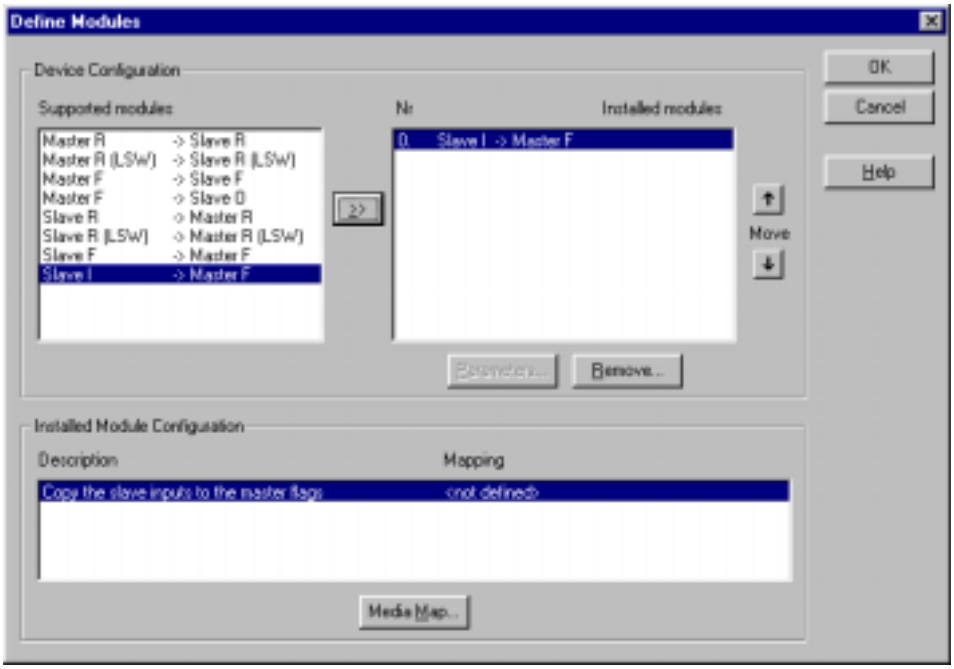

Definition of slave resources:



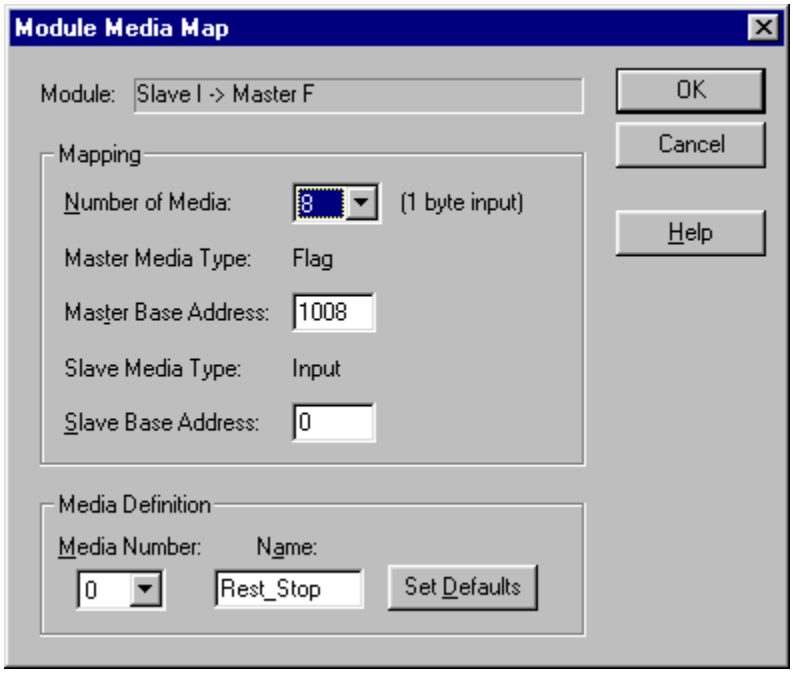

The resources are defined with a default symbol. These symbolic names refer to the PROFIBUS-DP address automatically assigned by the configurator.
When the station address is changed, the symbolic name is adjusted automatically.

Definition of media which are exchanged between the master and the slave: for example, if slave inputs 0..7 are written to master flags 1008..1015, parameters must be set for this as follows:

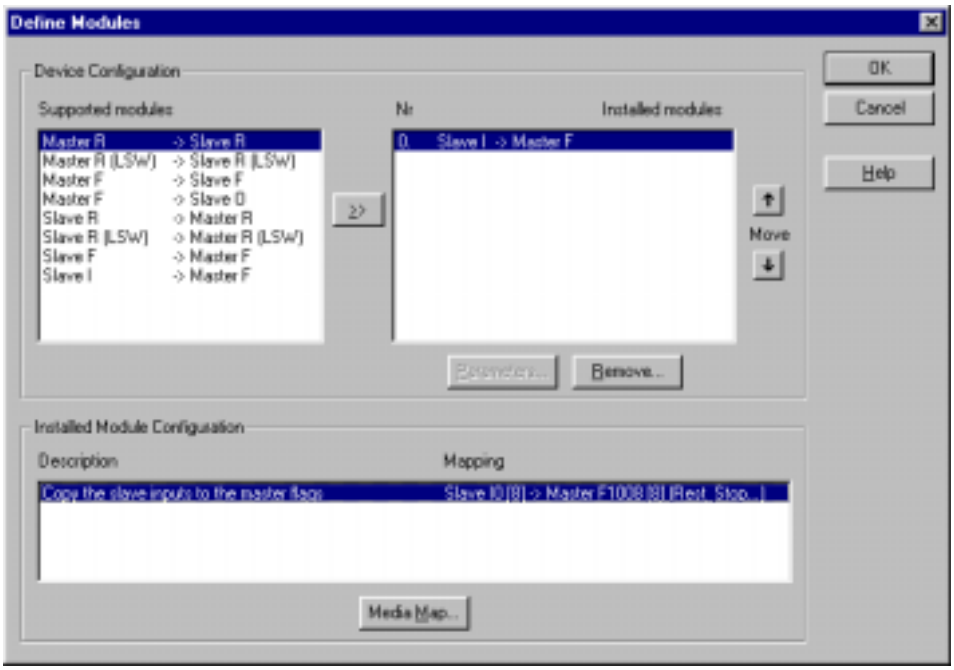
Definition of modules:




Mapping master and slave media:

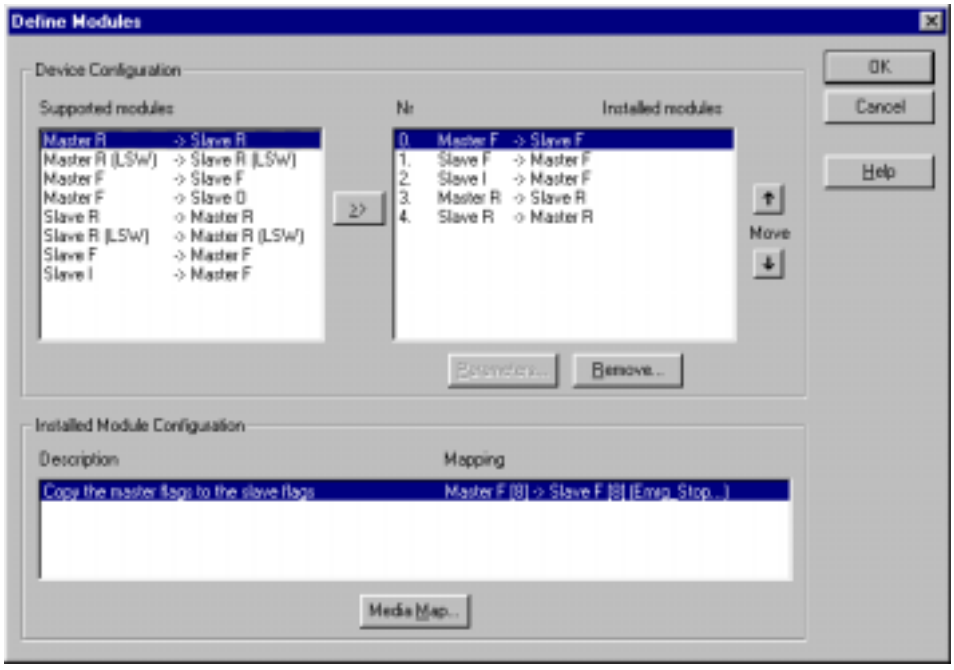



If media are assigned a symbolic name, this name can be used in the master program and in the slave program.



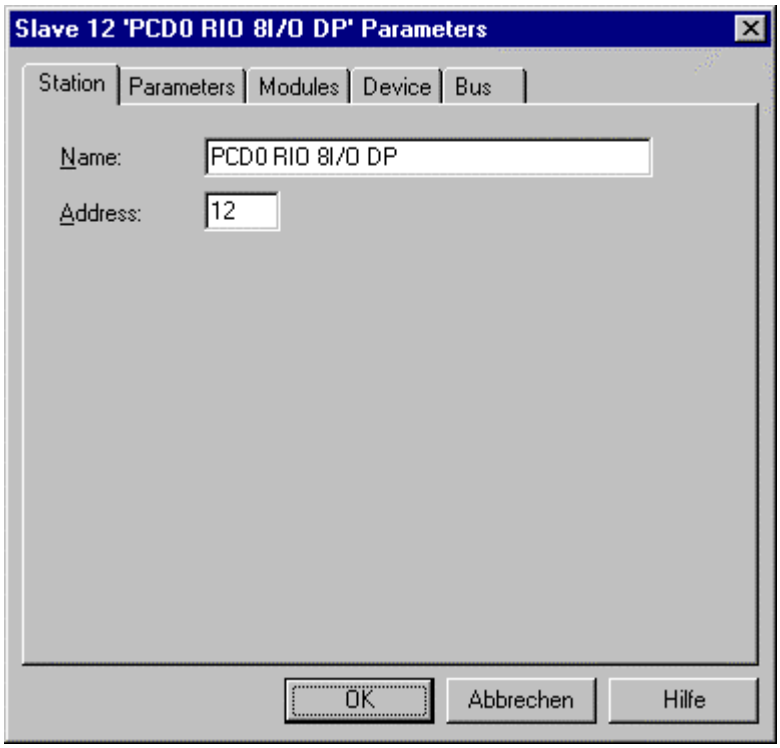

If even more data must be exchanged between the master and the slave, this can be entered as described above.

The final configuration of slave 10 appears as follows:

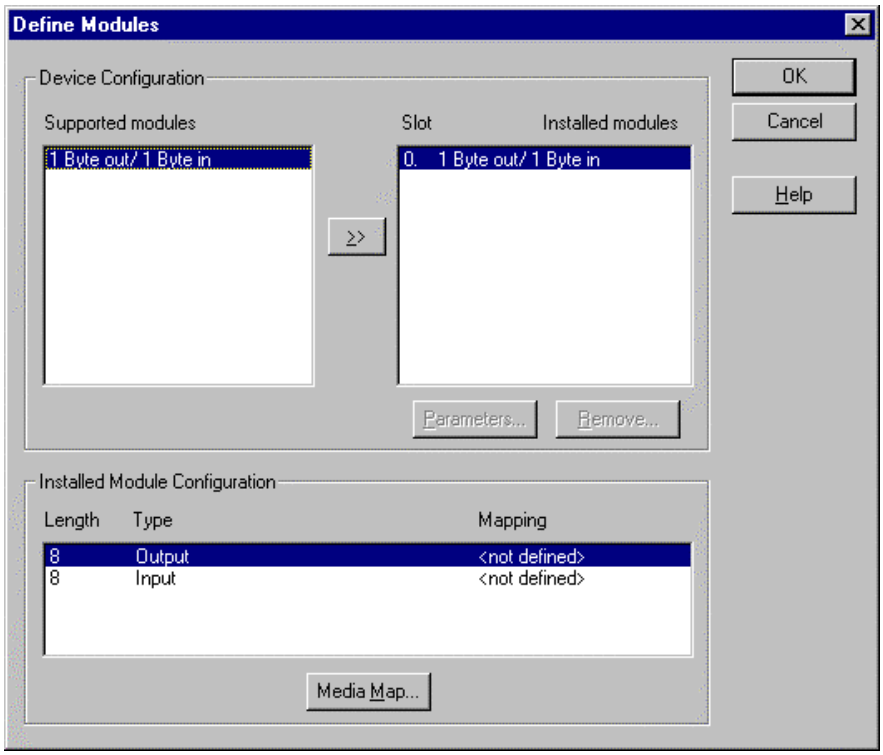


Setting parameters for slave 12

Double-click on slave PCD0.G100 (compact module) with address 3. Address 3 was assigned automatically by the configurator. This must be changed to 12.




Define master media to be used for accessing slave.
This slave is not capable of modular expansion. Therefore, no additional modules can be defined during configuration of the device.



In this slave the I/Os are divided as follows:

Inputs are addresses 0..3,
Outputs are addresses 4..7.

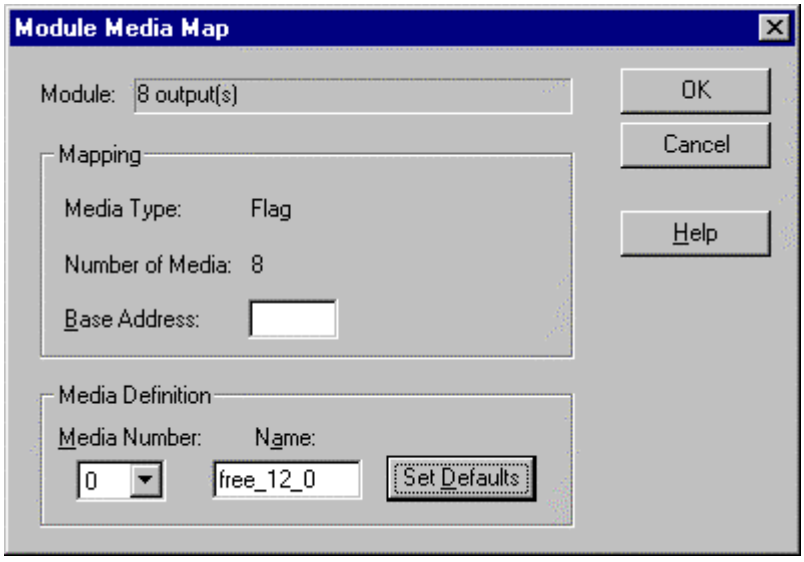
Media are addressed with the following symbolic names:

I 0: Emerg_st12
I 1: Feed_void
I 2: Stack_void
I 3: Stack_full
O 4: Vibra_on
O 5: Lamp_ok
O 6: Lamp_nok
O 7: Belt_on

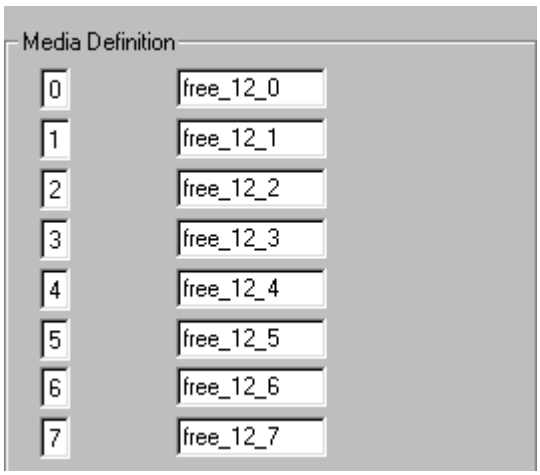
Any I/Os that are not required must also be addressed with a symbolic name, as it is only determined whether an I/O is an input or an output when the user program is in run.

The simplest way of defining unused I/Os is to proceed as follows:
Define all I/Os in the module with a symbolic name. For this, activate the 'Set default' button after entering the name at medium no. 0. All media are therefore defined with a similar name.

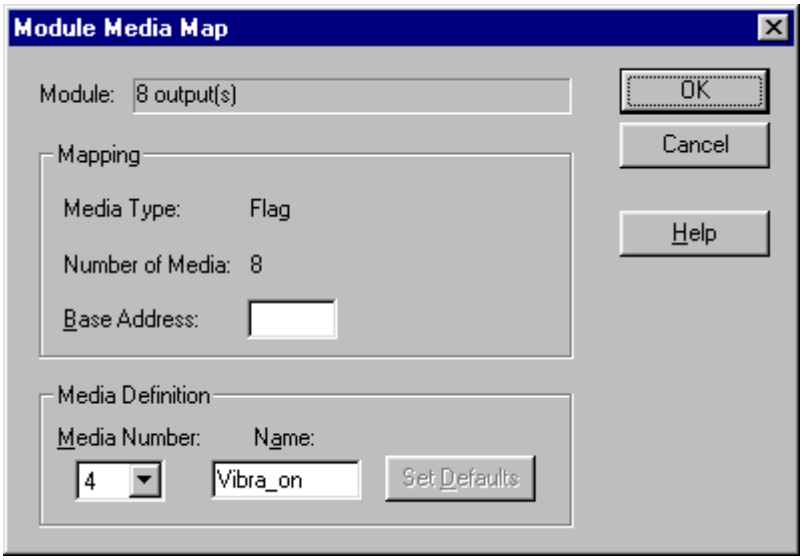
The 8 flags have now been numbered serially with the names 'free_12_0' to 'free_12_7'.



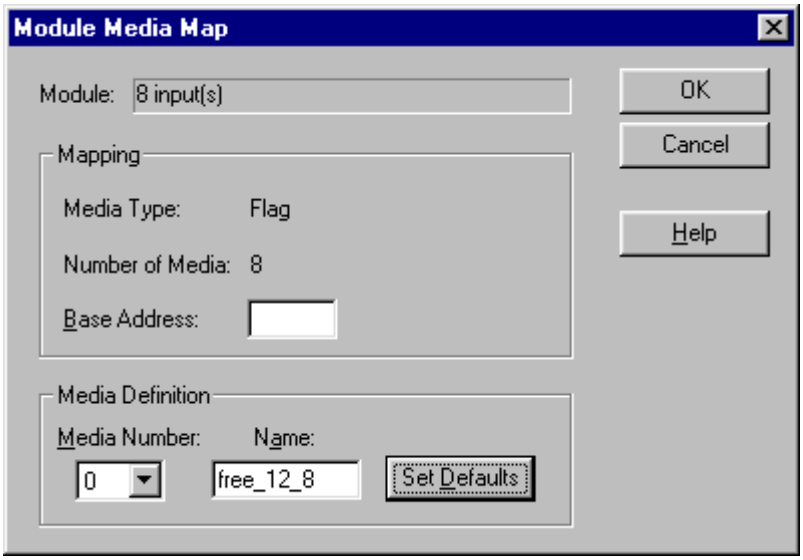
After activating the 'Set Default' button, the 8 flags are numbered serially as follows:

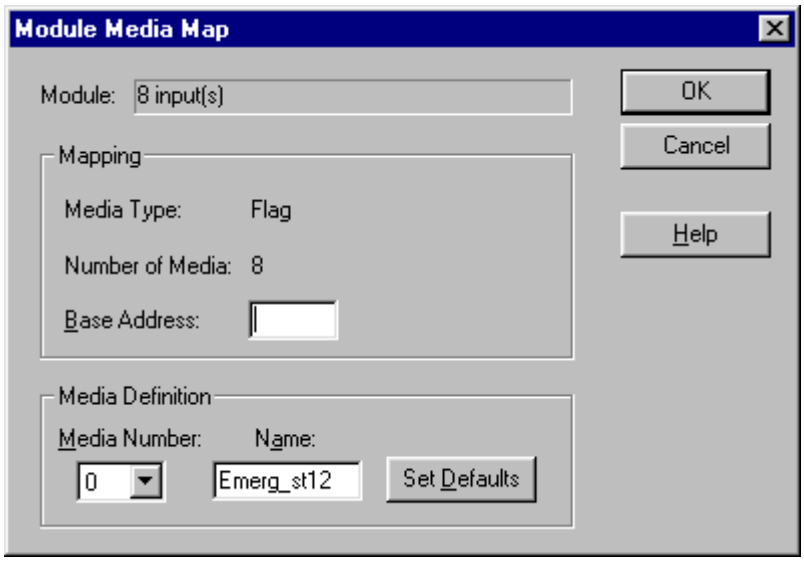



The correct symbol name can now be entered by the media used.

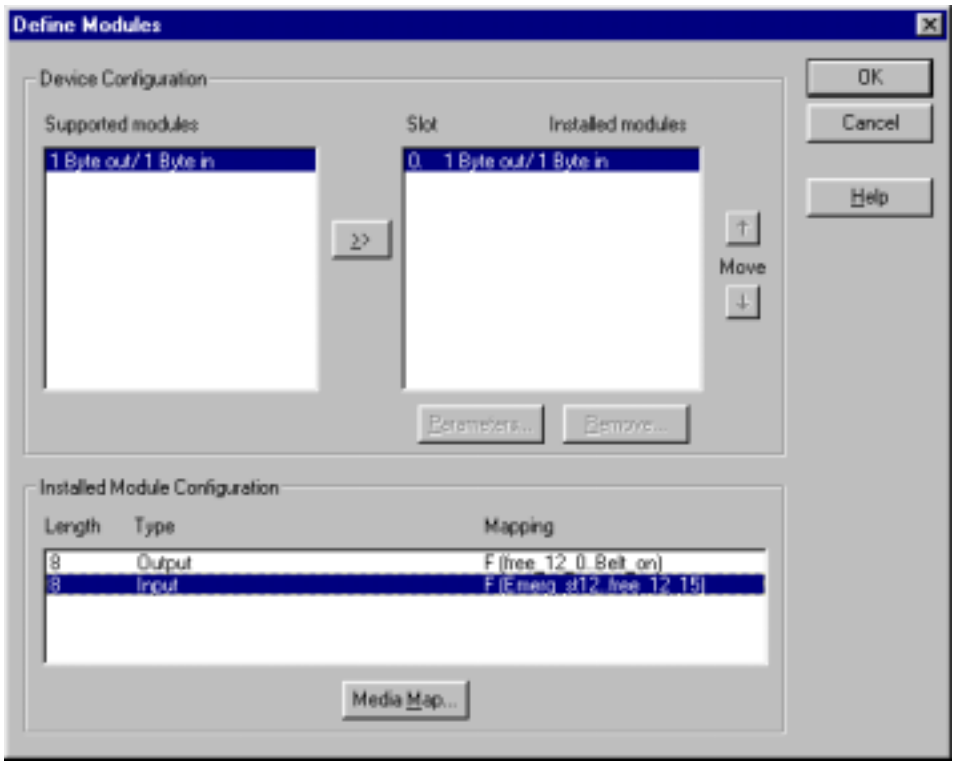


The same applies for the module's inputs:



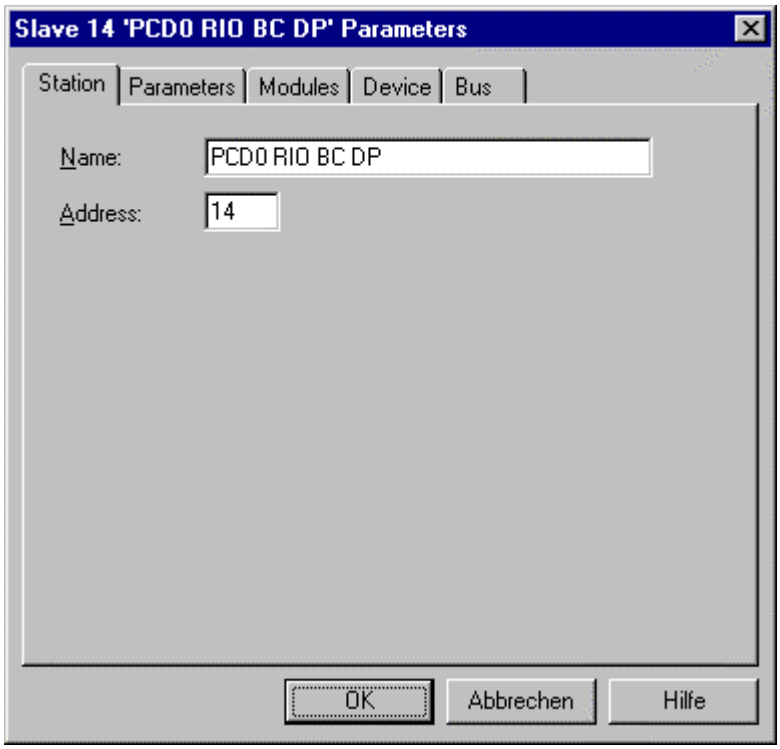



In the list of modules defined, the symbolic assignment of I/Os can then be seen (in each case, the first and last module only):



Setting parameters for slave 14

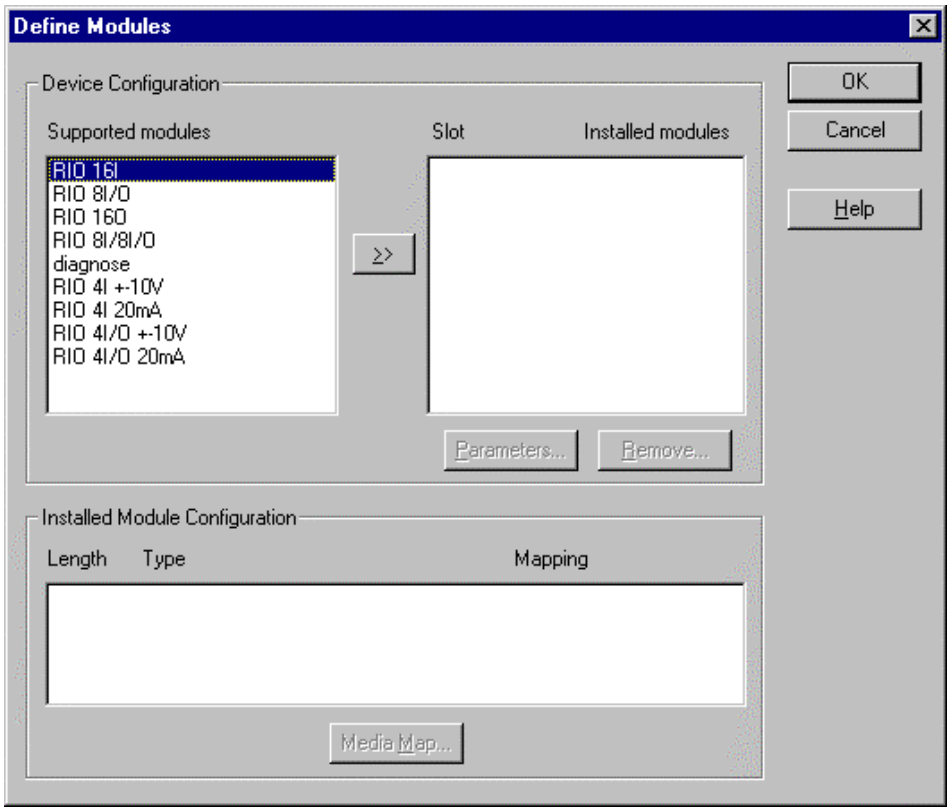
Double-click on the slave PCD0 (modular) with address 4:
Address 4 was assigned automatically by the configurator.
This must be changed to 14.




Define the slave's hardware allocation.

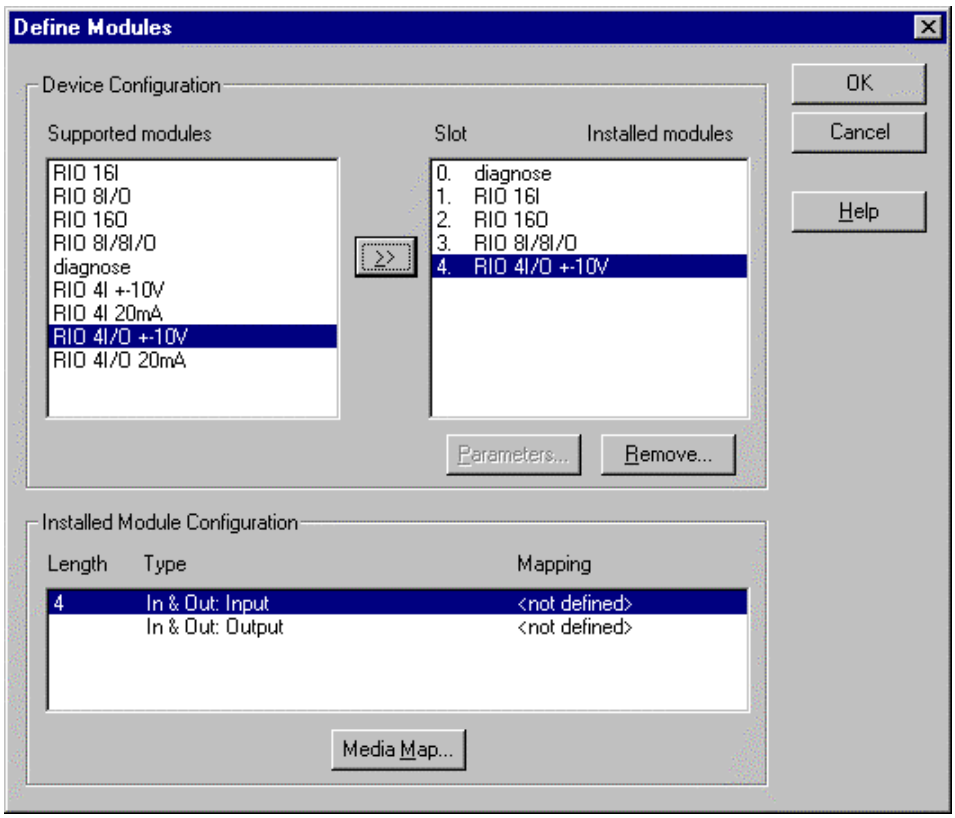
This is a question of defining the I/O modules used by the slave.

For the PCD0.T770, the first slot defined must always be a diagnostic module. The I/O modules must then be defined in the same order as they are arranged on the PCD0.




This appears as follows for the hardware installation below:

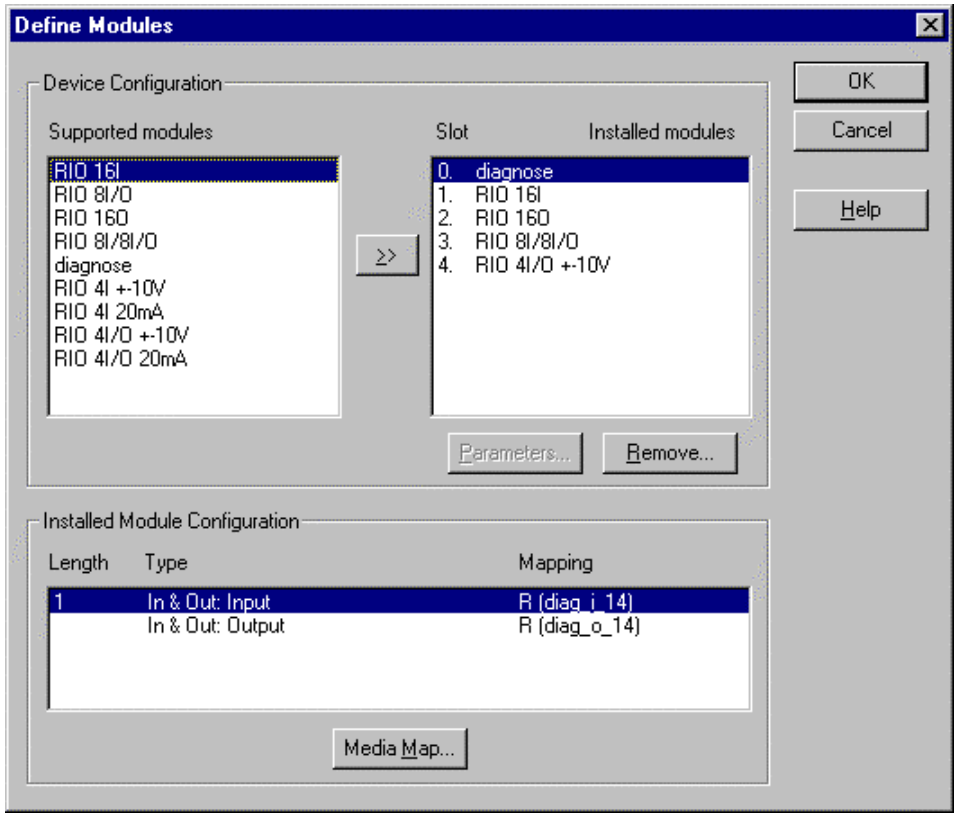
First slot: PCD0.E120, RIO 16I
Second slot: PCD0.A410, RIO 16O
Third slot: PCD0.B120, RIO 8I 8I/O
Fourth slot: PCD0.W710, RIO 4AI/4AO ±10VDC



Define the master media to be used for accessing the slave.
The master media must be defined here. For each installed module, the media must be mapped in the master.

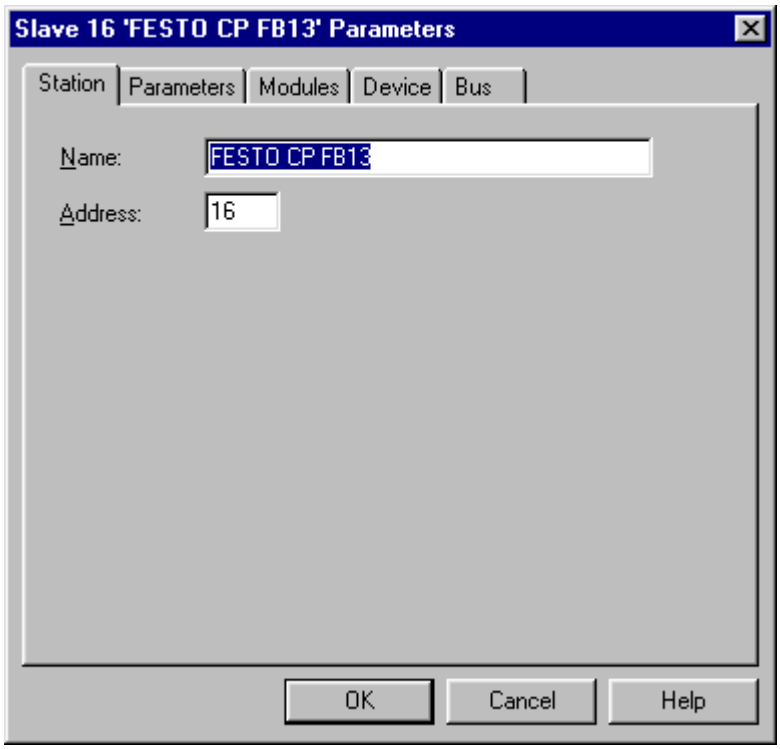
This takes place in the same way as for the compact PCD0 (slave no. 12).

After the successful definition of all I/Os, this will appear roughly as follows:



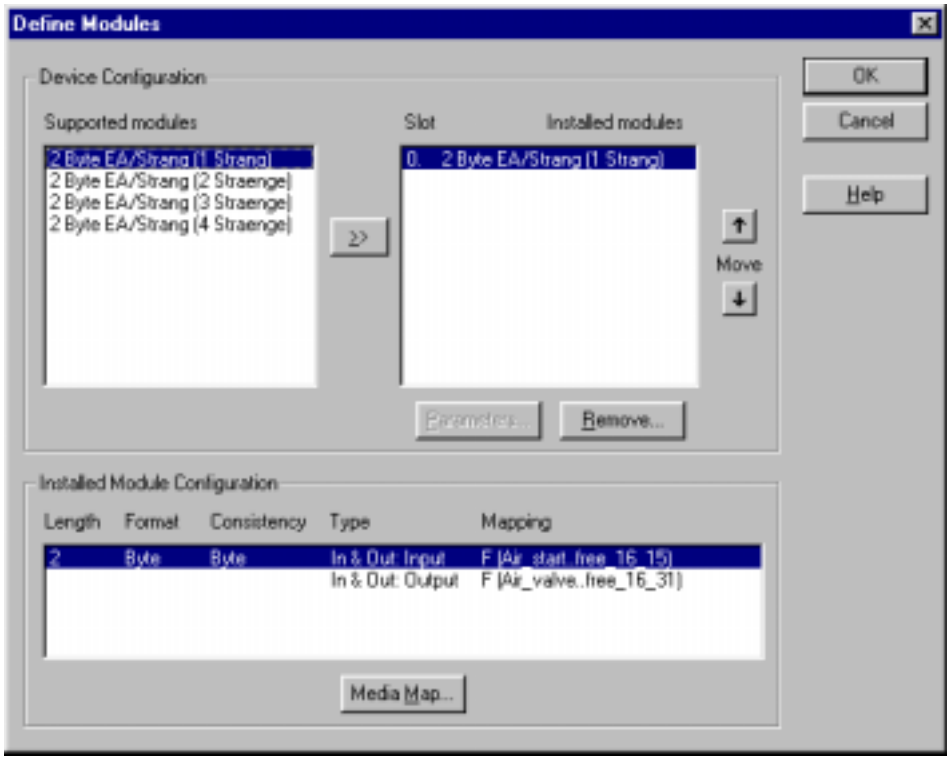
Setting parameters for slave 16

Double-click on the Festo valve island with address 5:
Address 5 was assigned automatically by the configurator.
This must be changed to 16.



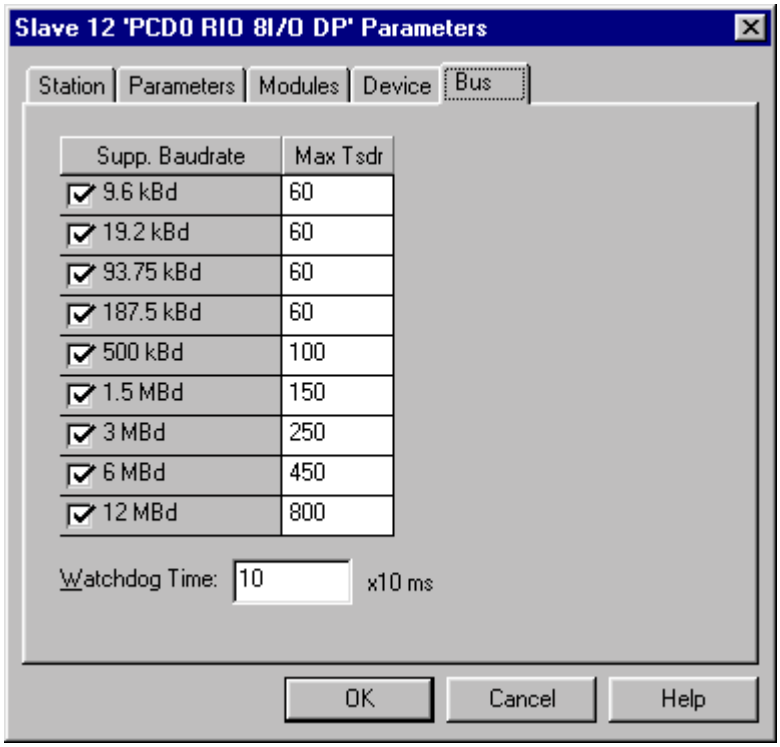

With the Festo valve island, one branch with 16 inputs and 16 outputs is used.

Media entry takes place as described above.



Definition of slave watchdog time:

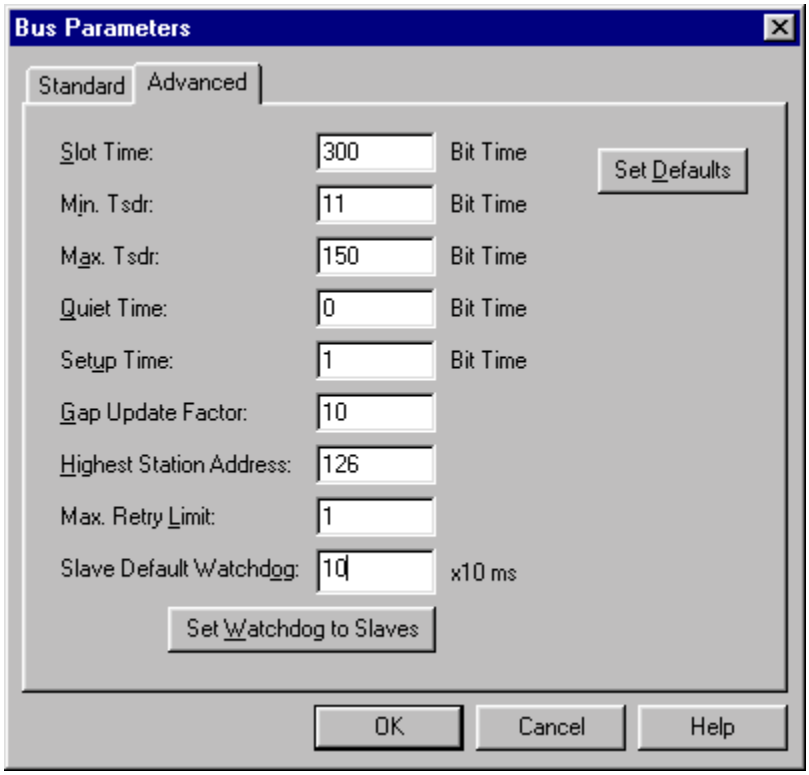
If required, the slave watchdog time can be defined for each slave individually or for all slaves in common. To enter this for each slave individually, select the 'Bus' submenu in the slave.




It is then possible to enter a watchdog monitoring time for each slave.

A watchdog time of 0 deactivates the slave watchdog.

If the watchdog monitoring time is to be the same for all slaves, this is achieved by double-clicking on the PROFIBUS-DP line in the PROFIBUS-DP network window.

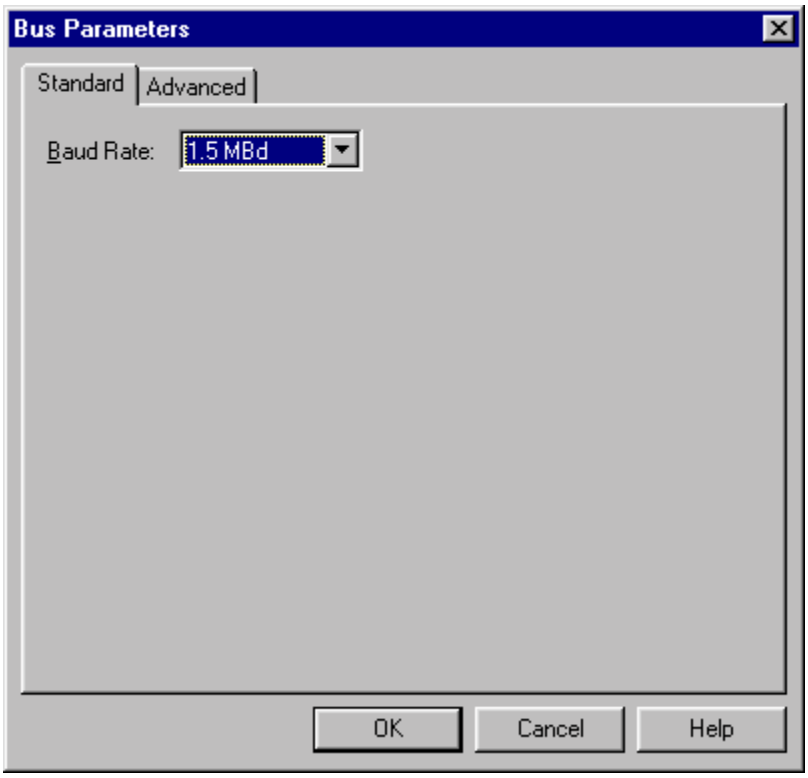



Activating the 'Set Watchdog to Slaves' button transmits to all slaves the watchdog time set in this window.

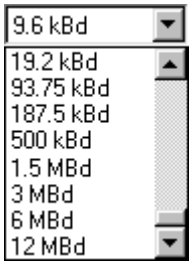
#### 5.5.2.4 Changing network parameters

If required, network parameters like bus speed, watchdog monitoring time, etc. can also be modified. This is done by double-clicking on the PROFIBUS-DP line in the PROFIBUS-DP network window.

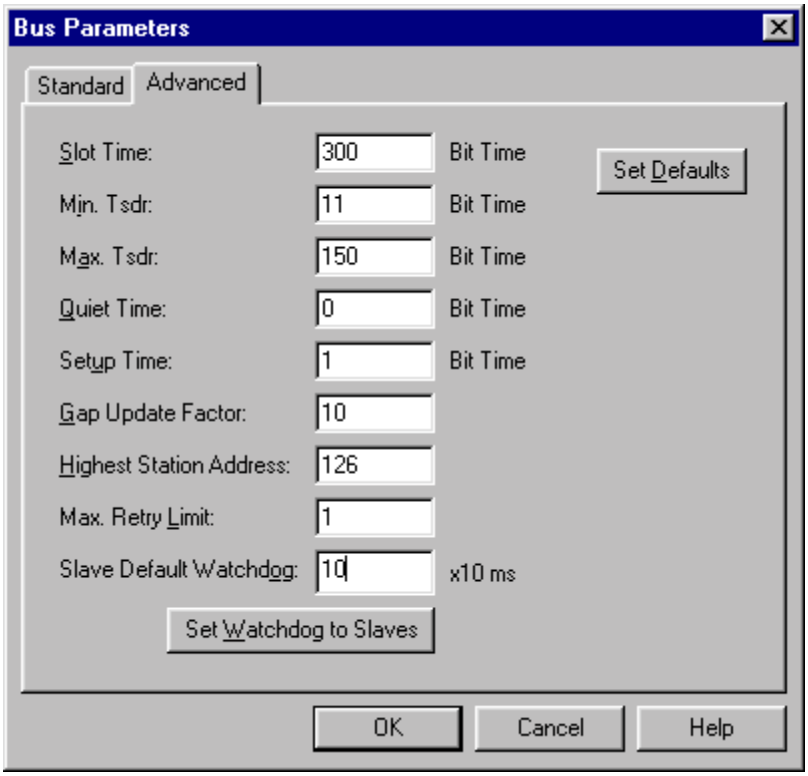
The following frame then appears:




One of the following baud rates can be chosen:



In menu item 'Advanced' the following frame appears:




Normally default parameters are adequate. However, if required they can be modified. By activating the 'Set Watchdog to Slaves' button, the watchdog time set in this window is transmitted to all slaves. A watchdog time of 0 deactivates the slave watchdog.

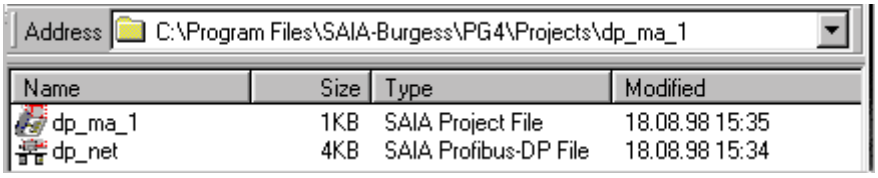
#### 5.5.2.5 Further processing of data

When all stations have been configured and their parameters set, the PROFIBUS-DP project must be compiled. This generates the '*.src' and '*.def' files for the master station and for the intelligent slave stations. These files are then linked to the actual user program and produce the program that will run. The files are stored in the appropriate file directory.

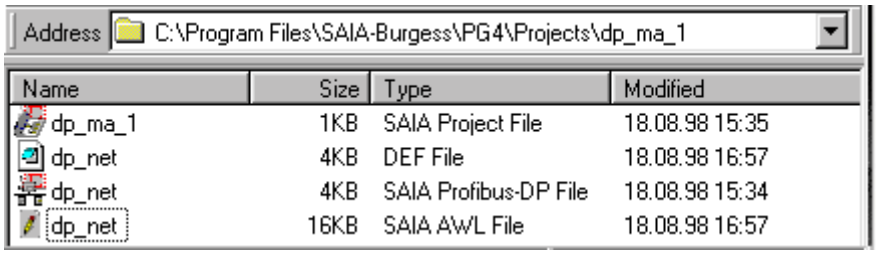
If the data is subsequently further processed with PG4 programming software, the linking of PROFIBUS-DP files is done automatically by the PG4.

In contrast, if the user program is further processed with PG3 programming software, the PROFIBUS-DP file must be integrated into the user program with the ‘$INCLUDE *.DEF‘ instruction.

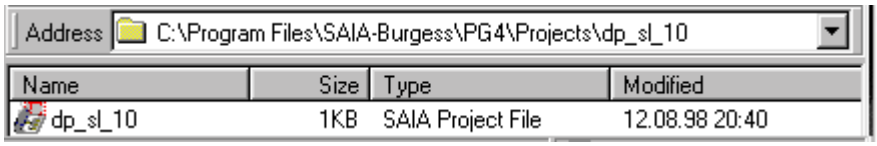
File directory of station 'dp_ma_1' before compiling:



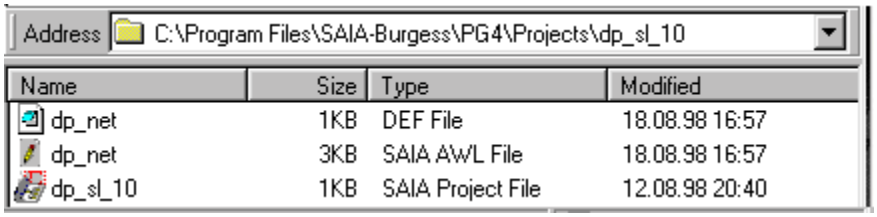

File directory of station 'dp_ma_1' after compiling:




File directory of station 'dp_sl_10' before compiling:



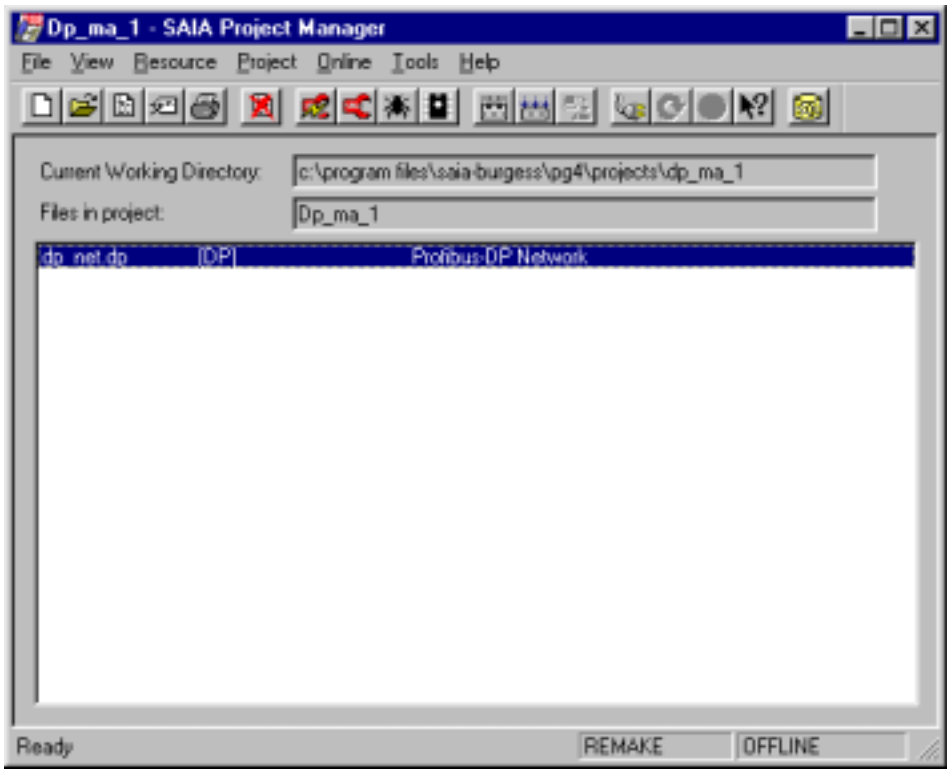

File directory of station 'dp_sl_10' after compiling:



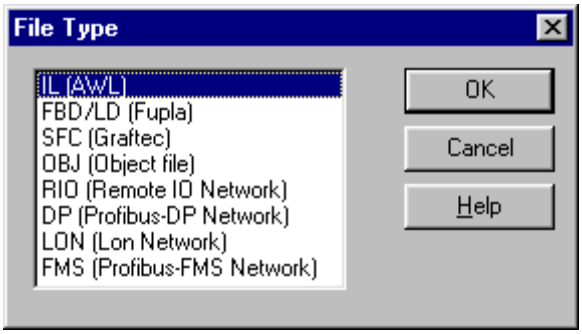

This concludes the configuration and definition of the PROFIBUS-DP network.

Writing the user program in the master

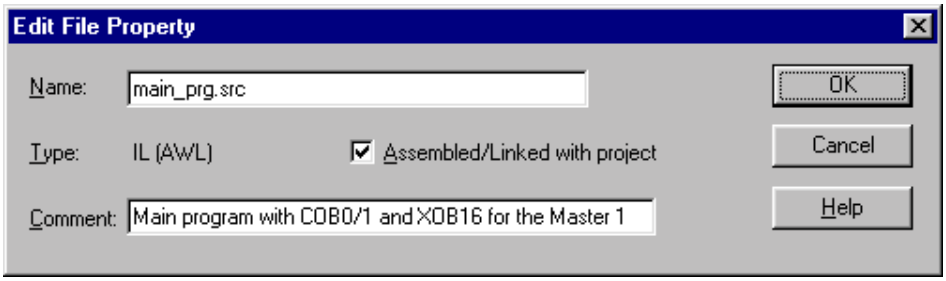
If project 'dp_ma_1' is edited, the following Project Manager window appears:



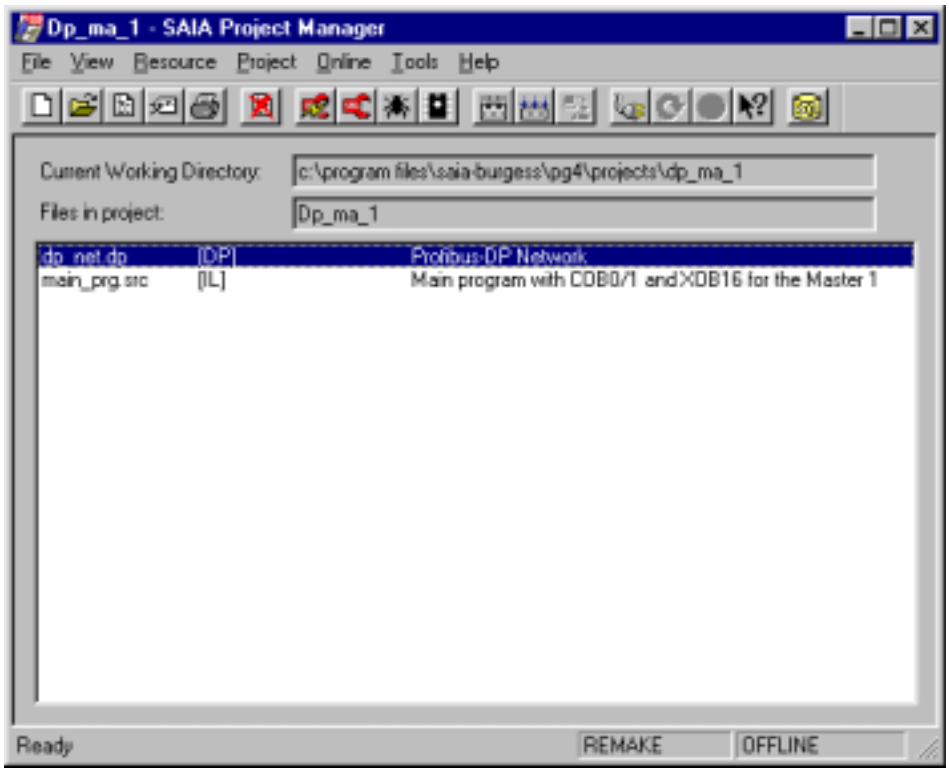

To enable entry of the user program, a new file must first be opened:



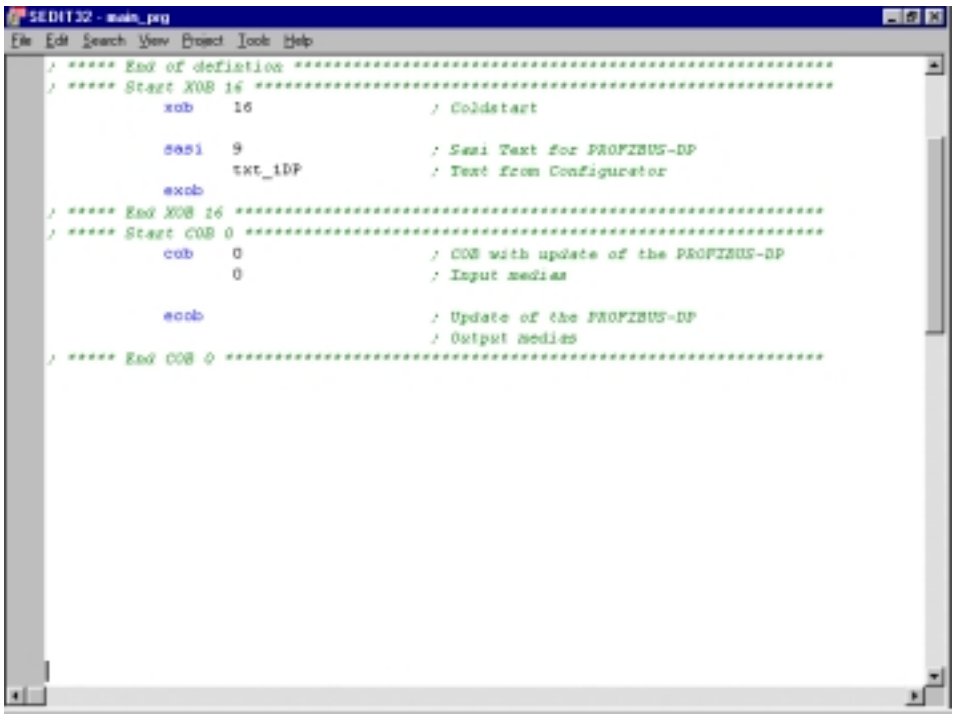

The name of the IL file can then be entered.



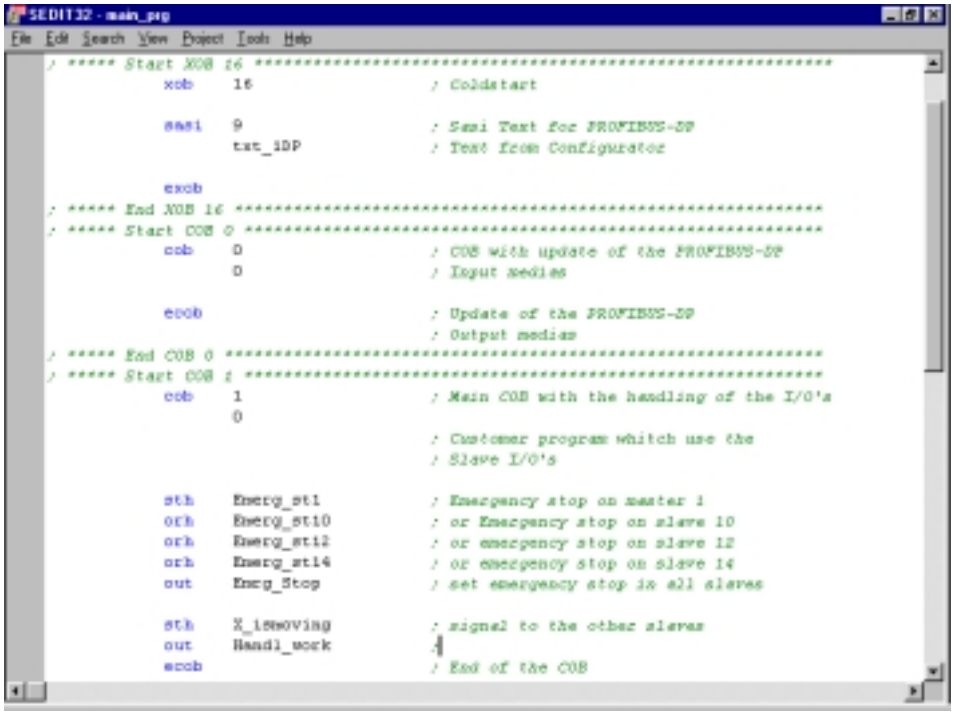



The following basic program can now be edited in the master PCD:

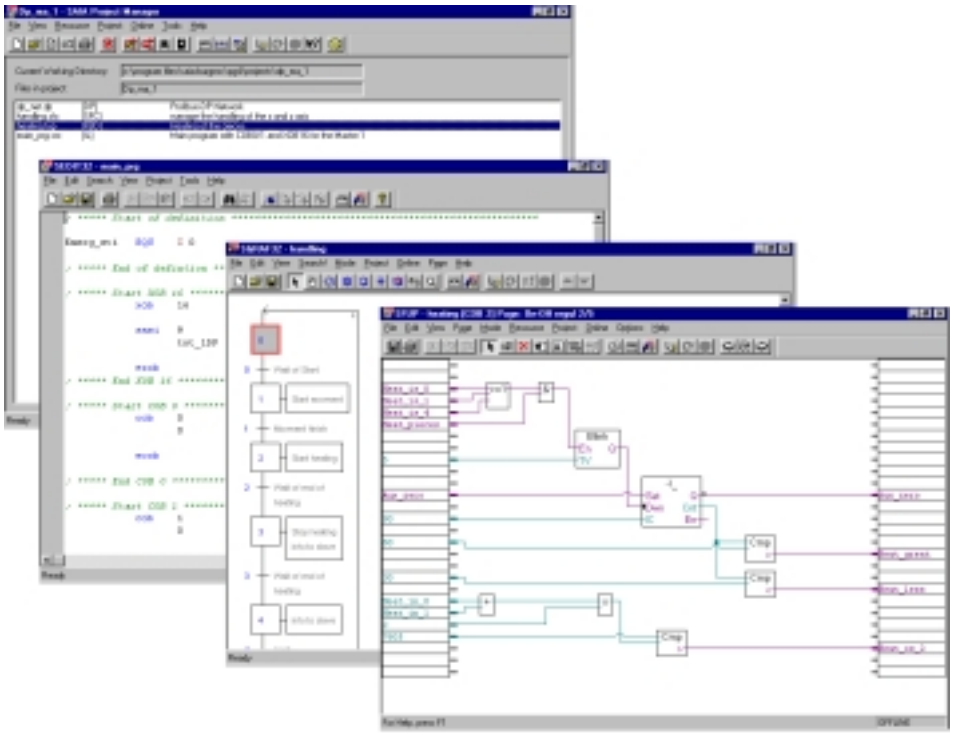



This basic program still produces a user program that will run.

The code for handling the remote I/Os can now be entered:

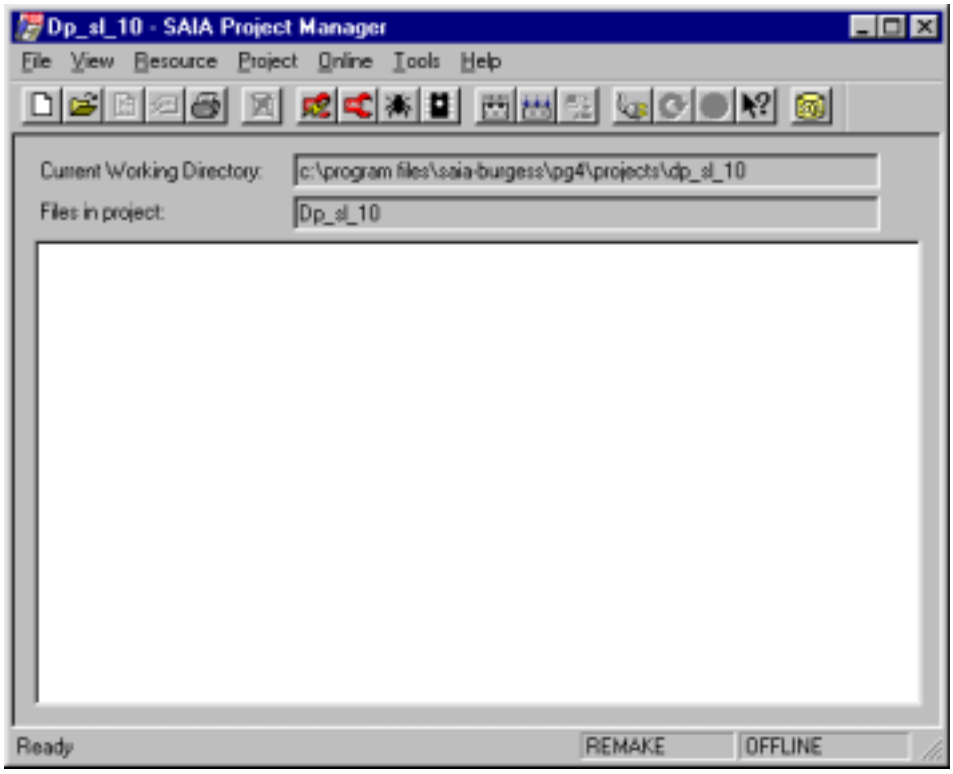



When using PROFIBUS-DP, all available editors can be utilized. Since the handling sequence calls for sequential control, it is logical to write this part of the program in GRAFTEC. On the other hand, if required FUPLA can be used to access existing Fboxes. Control of the entire installation can then appear as follows using IL, FUPLA and GRAFTEC:



Writing the user program in slave 10

Call project 'dp_sl_10':




Writing the user program takes the same points into consideration as for the master.

- Enter IL code for the SASI instruction.
- Program COB 0 for data exchange.
- Enter IL, FUPLA and GRAFTEC programs.

The slave project can then appear as follows:

### 5.5.3 Example 3

**Task:**

A master from a non-SAIA controller (Siemens S7) must communicate with a SAIA PCD1 type slave.
This must involve reading or writing the following data in the PCD1:

| No | Message | # Word/Byte | Media Slave |
|---|---|---|---|
| 0 | Master → Slave | 16 Word | R 100 – R 107 |
| 1 | Slave → Master | 16 Word | R 200 – R 207 |
| 2 | Master → Slave | 1 Byte | F 100 – F 107 |
| 3 | Slave → Master | 1 Byte | F 200 – F 207 |
| 4 | Slave → Master | 1 Byte | I 0 – I 7 |
| 5 | Master → Slave | 1 Byte | O 32 – O 39 |

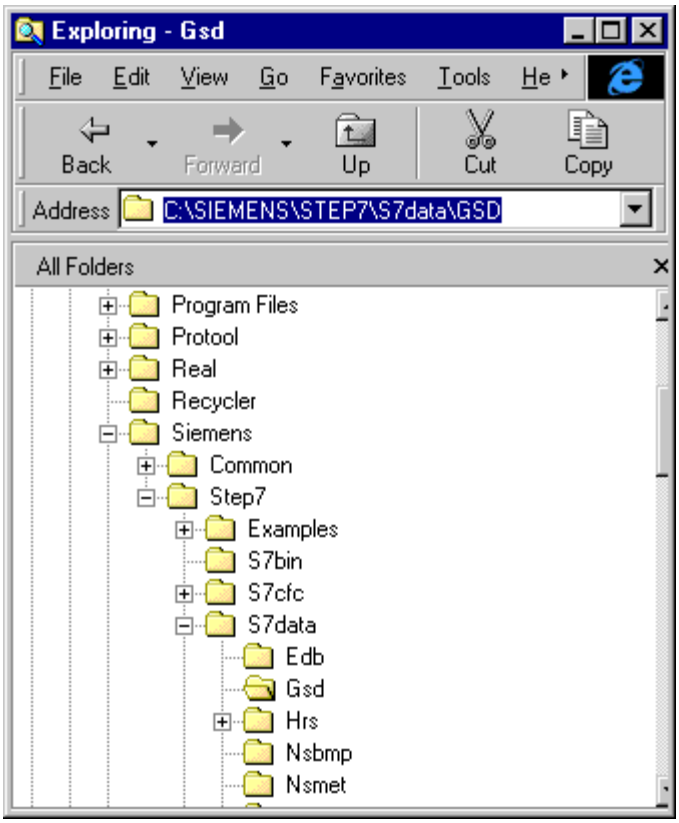
**Solution:**

Since both the master controller and the slave controller contain a user program with PROFIBUS-DP, both network configurators must be used by the master and by the slave to generate the relevant configuration data and program files for the devices. When doing this, care should be taken that PROFIBUS-DP messages between master and slave are defined identically (sequence, size, etc.) in both configuration tools.

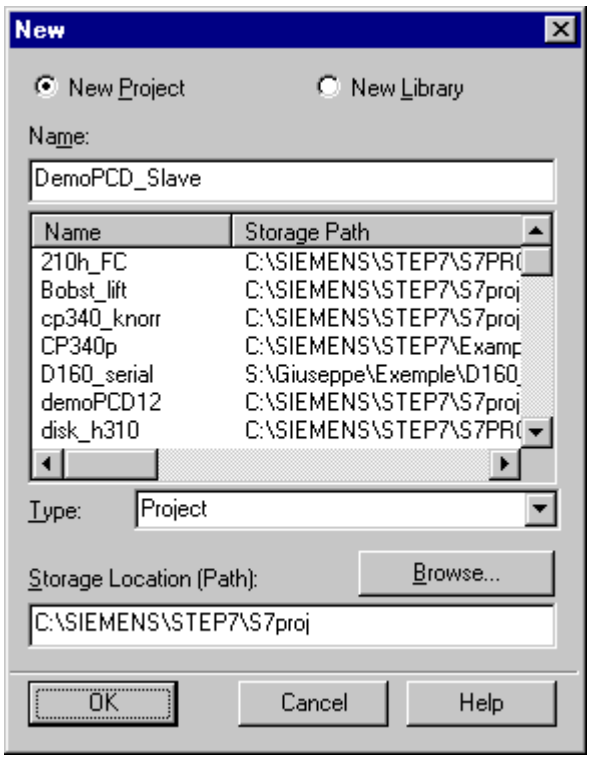
The following example has been written with Siemens Step-7 programming software version 4.

Configuration of the Siemens S7 master:

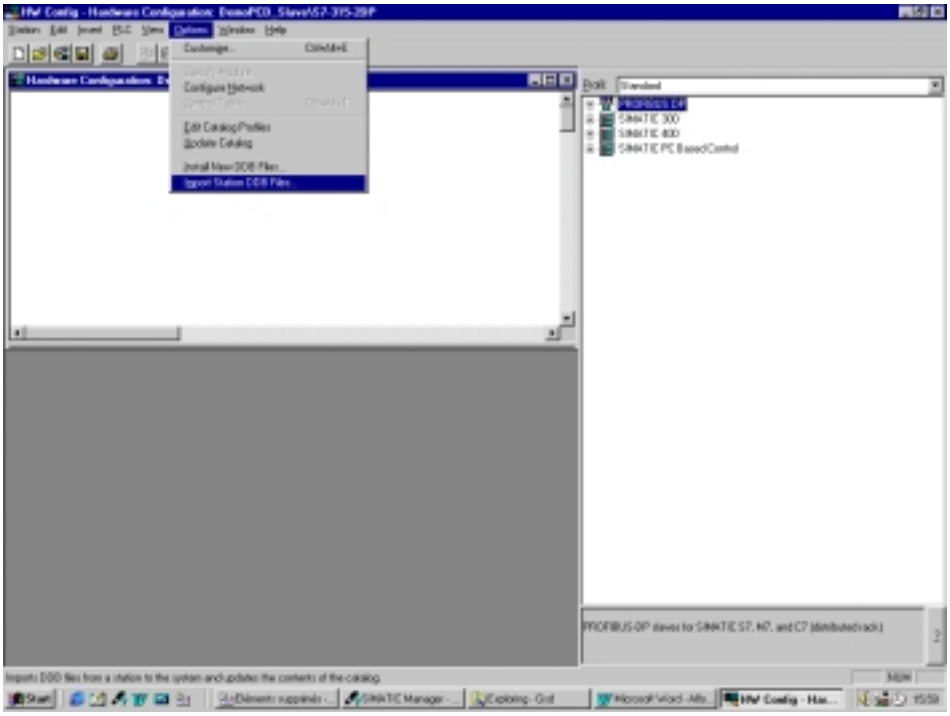
Copy the *.GSD file from the PCD1 into the following directory:



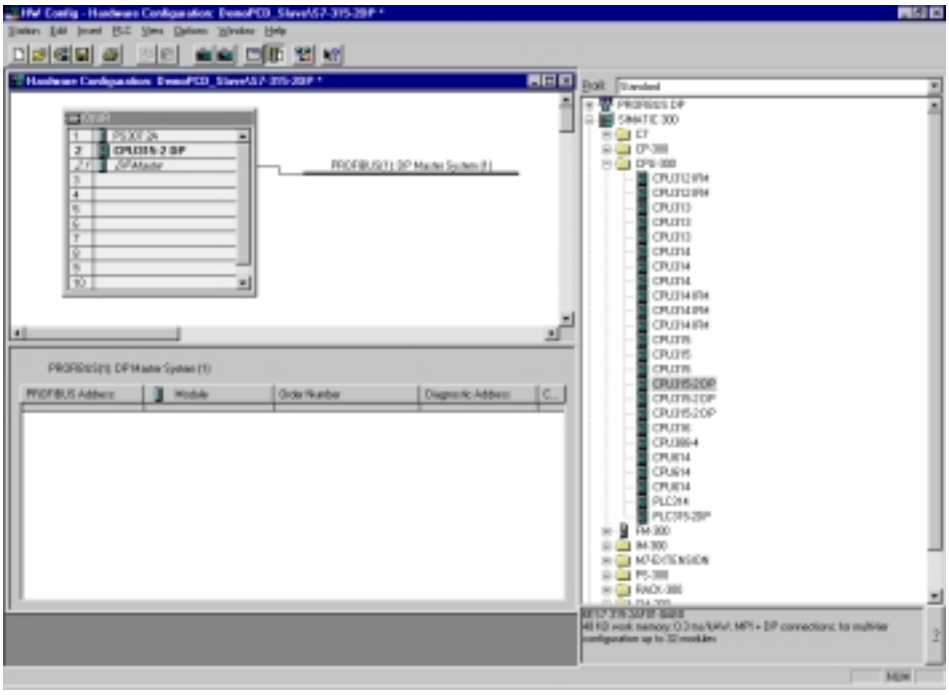

Define a new project:



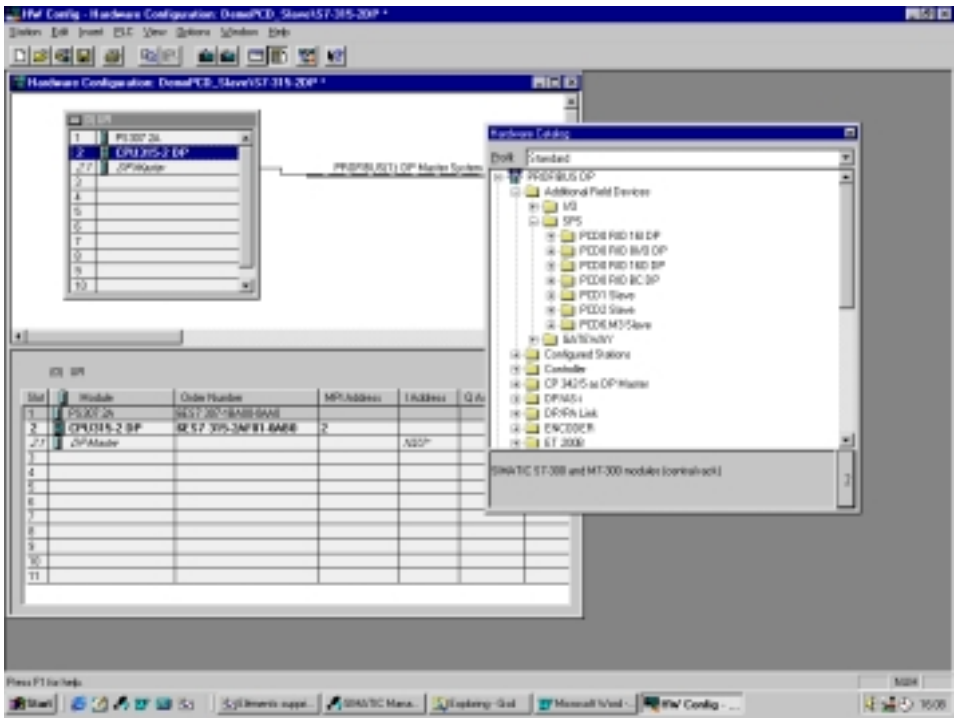
Insert the PCD1 *.GSD file into the Siemens S7 configurator with Import Station DDB Files.



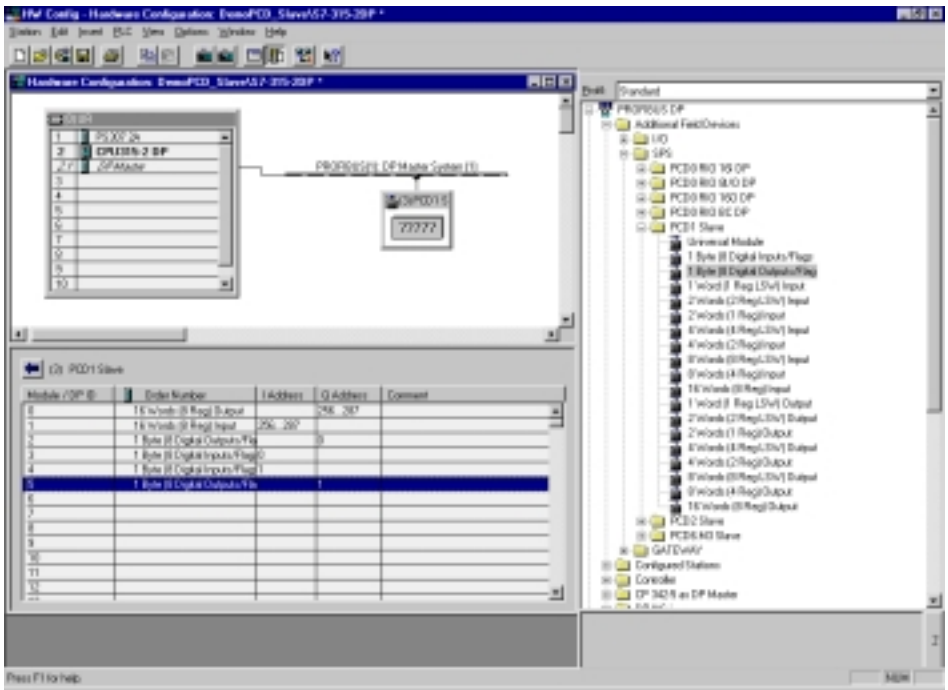
Hardware configuration of master:
- Rack
- Power Supply
- CPU-315-2DP Master module



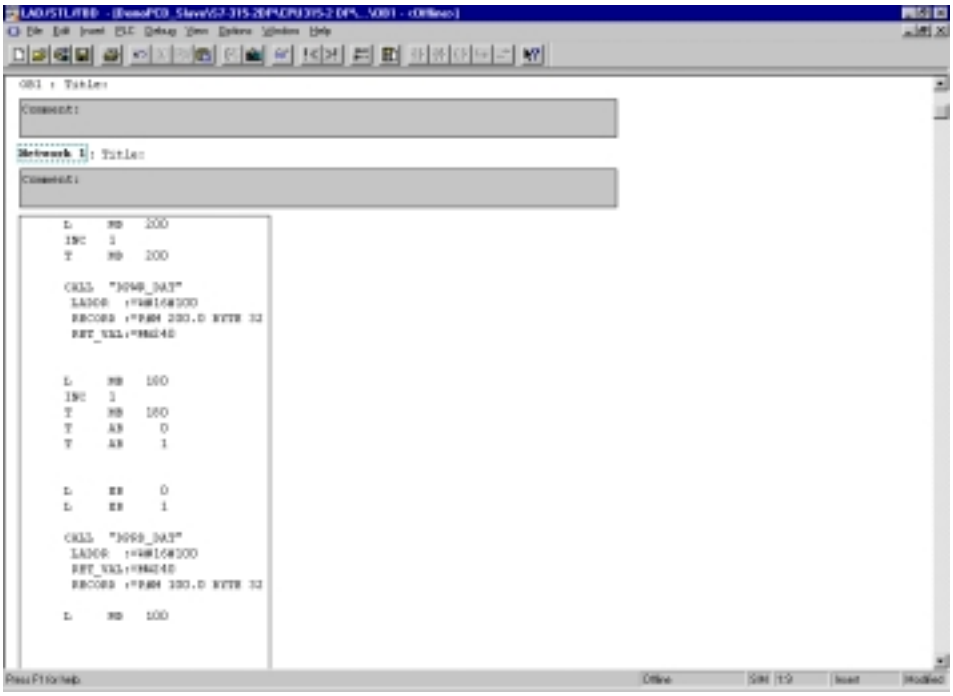
Choice of PCD1 from the PROFIBUS-DP menu:



Configuration of I/O data to be exchanged with the slave:

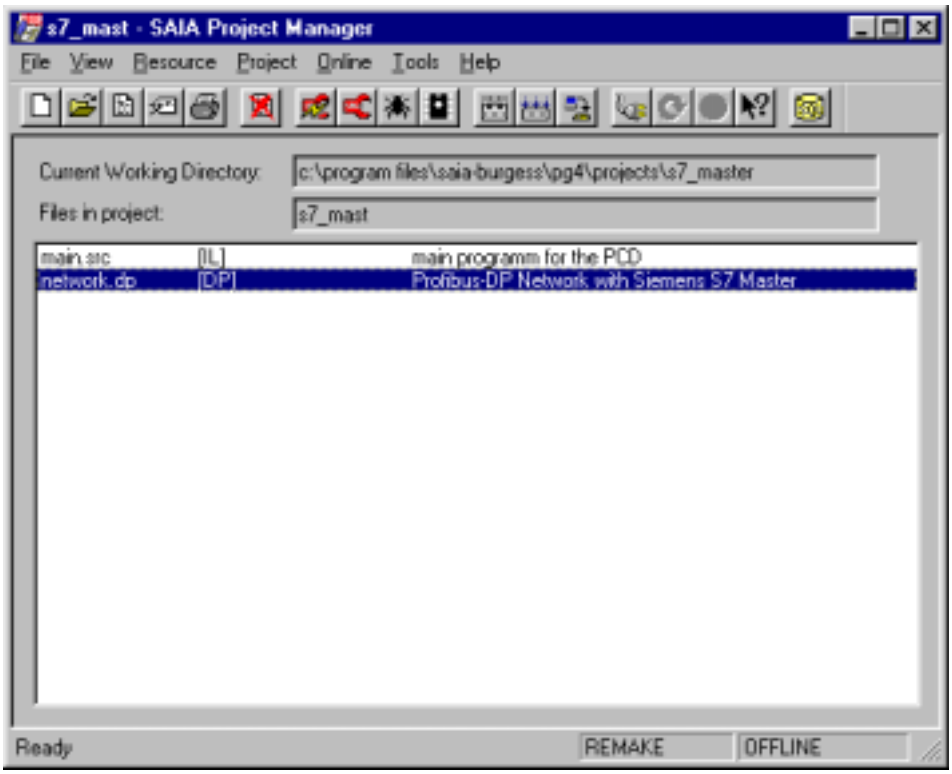


User program in master for processing slave data:

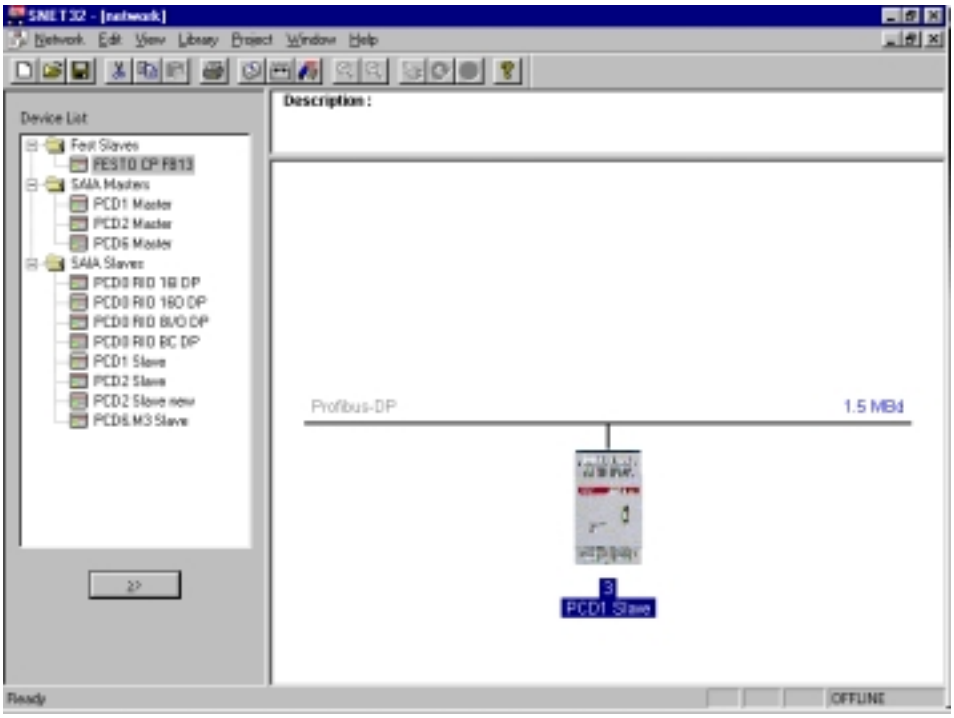


Configuration of SAIA PCD1 slave:

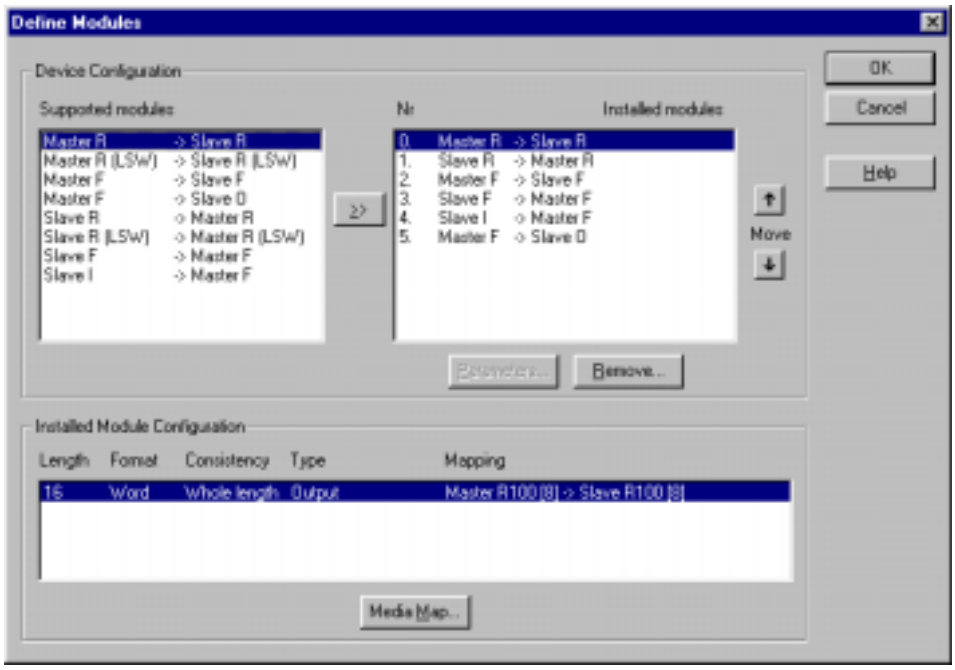
Write a new PG4 project with a PROFIBUS-DP and an IL file:



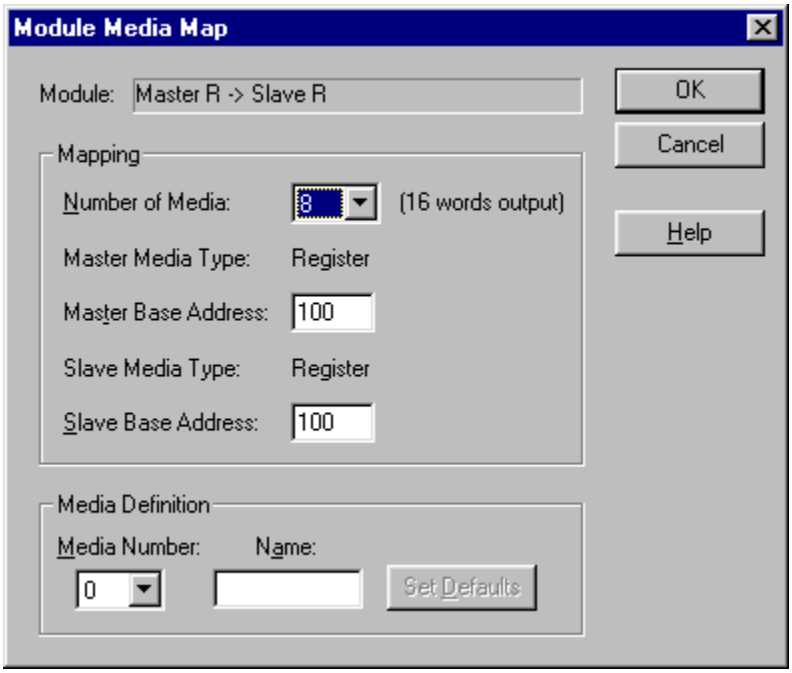

Definition of network:
Master controller does not have to be defined.



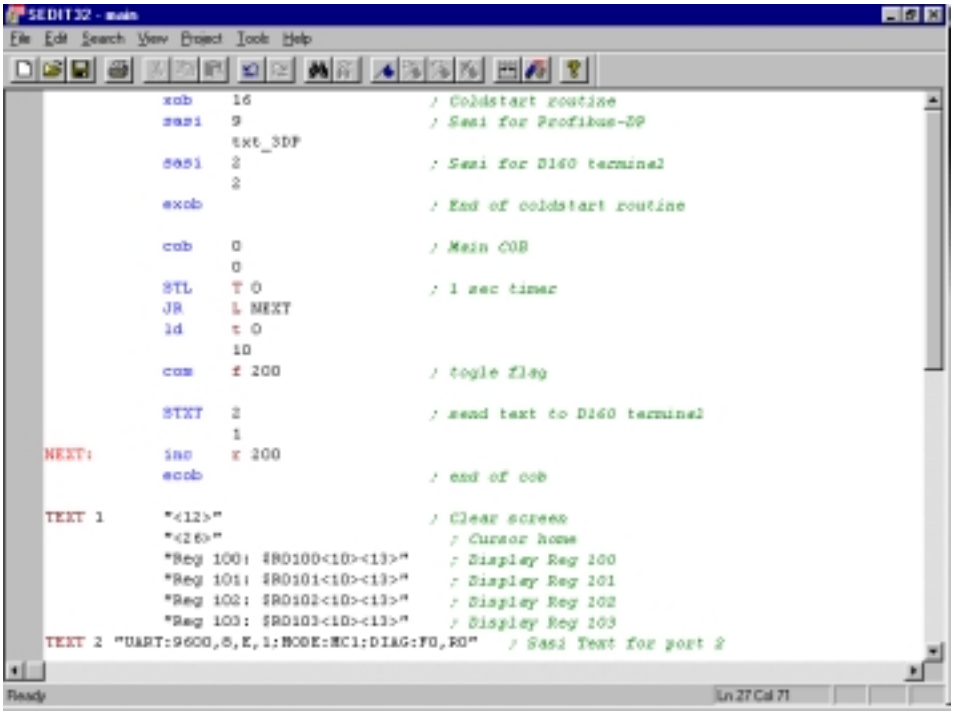
Define messages to be exchanged between the master and the slave. Here data must be defined with the same order, size and direction of data exchange as for the master.




The media map of message 0 appears as follows:



Write the user program for the slave:



# 6. Commissioning a PROFIBUS-DP network

Commissioning a PROFIBUS-DP network is completed in two steps:

a) Checking and testing the hardware installation (physical layer)

b) Checking and testing layer 7 (configuration and data exchange)

## 6.1 Checking and testing the hardware installation (physical layer)

Experience shows that the main causes of communications problems lie in inadequate or incorrect hardware installation. For this reason, great importance must be attached to this part of the commissioning process.

The following checks and tests should be carried out scrupulously:

- Connection and laying of bus cable (terminals and plugs, screening, stub cables, laying power cables, etc.)
- Check and adjust line termination resistors, repeaters, etc.
- Check the bus line for through transmission
- Check the electrical signal level

A detailed description of the checks and tests (also called static tests) can be found in the manual "Installation components for RS 485 networks" (order ref. 26/740 E).

## 6.2 Checking and testing data exchange

These tests are to check data exchange between the master and slaves and the correct mapping of slave I/Os to master media.

### 6.2.1 Checking data exchange

These tests are to check data exchange between the master and the slaves.

1. Create the network with the PROFIBUS-DP configurator. For this all stations present in the network must be defined with all I/Os.
   The correct baud rate must also be entered.
2. Write a basic program in the PG4, which must contain the following program statements:

```
XOB   16          ; Cold start routine
SASI  9           ; Start PROFIBUS-DP
      txt_1DP
EXOB

COB   0           ; Update inputs from PROFIBUS-DP
      0
ECOB              ; Update outputs from PROFIBUS-DP
```

3. Assemble and link the basic program and the PROFIBUS-DP configuration.
4. Load the program into the master controller.
5. Start the program.
6. Check diagnostic flag DATA_EXCH.
   This flag signals the operating status of data exchange on the PROFIBUS-DP network.
   High means that data exchange is running on the PROFIBUS-DP network.
   Low means that data exchange is not running on the PROFIBUS-DP network.

Possible errors:

- No master in network.
- No PROFIBUS-DP card plugged into PCD.
- Incorrect FW or HW version.
- No SASI instruction.
- No COB 0, ECOB statement.

7. Check diagnostic flag SLAVE_ERR.
   If this flag is low, slave stations are addressed without error. If this flag is high, one or more slave stations has an error.
   To discover the faulty station, diagnostic registers base+3 to base+6 can be used. The diagnostic data of the faulty slave can then be read with the SCON instruction using function code 1.

Possible errors:

- Station number in slave does not match the station number in the PROFIBUS-DP configurator.
- Several slaves have the same station number.
- Wiring is incorrect. (Short circuit between A and B, or A and B have been muddled).
- Slave does not support the selected baud rate.
- Slave has been wrongly configured.
- The slave's GSD file does not match the slave device.

### 6.2.2 Checking the mapped media

This test checks the correct addressing of slave I/Os.
After data exchange between the master and the slaves has been checked according to section 6.2.1, the following test can be executed:

1. The program described in section 6.2.1 is loaded into the master controller.
2. Start the program.
3. In the PG4 the debugger is used to access individual media of the slave I/Os.
   For this purpose the slave inputs are displayed with a 'Display Flag' instruction and the slave outputs are described with a 'Write Flag' instruction.
4. Check that the right I/Os on the right slaves are read or switched on/off.

Possible errors:

- Station number in slave does not match the station number in the PROFIBUS-DP configurator.
- Several slaves have the same station number.
- Several I/Os have been mapped to the same media.
- Slave has been incorrectly configured.
- The slave's GSD file does not match the slave device.
- PROFIBUS-DP media are already used in the user program.

### 6.2.3 Use of bus monitors

If the checks and tests described in the previous sections do not produce the desired results, a bus monitor must be used for further tests and analyses.

A bus monitor is a testing tool for the commissioning, maintenance and diagnosis of PROFIBUS-DP networks. As a passive tool, it does not affect the bus in any way and does not require a station address, nor does it have to be considered during planning.

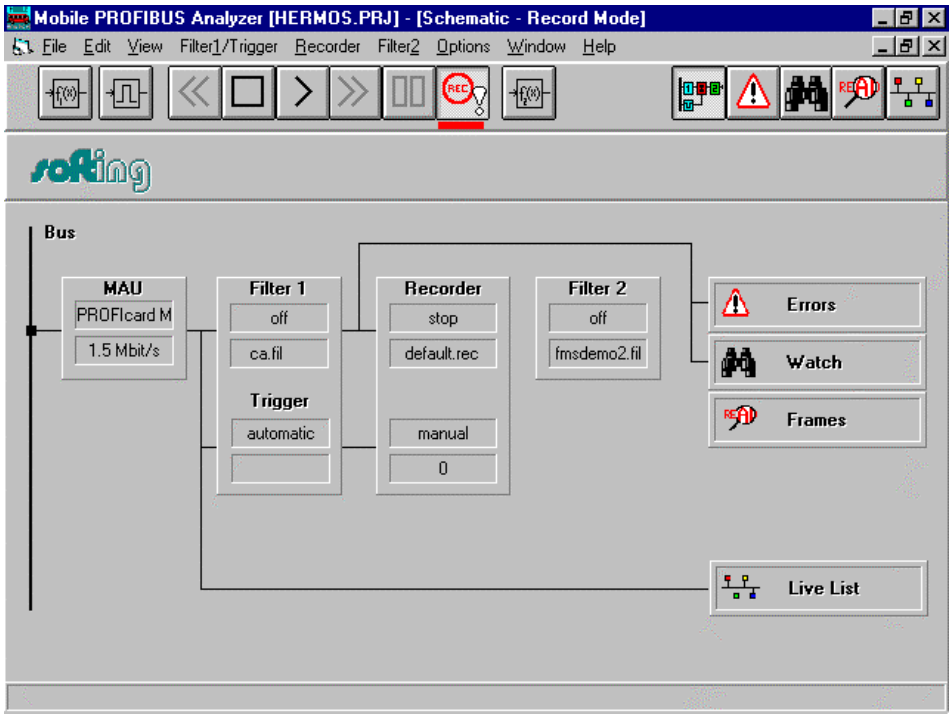
In online operation, the bus monitor traces telegram communications and displays either the "Live List" of stations connected to the bus, or certain bus characteristics. In offline operation the data recorded can be assessed and telegram analysis can be carried out on layers 2 or 7 of the communication model.

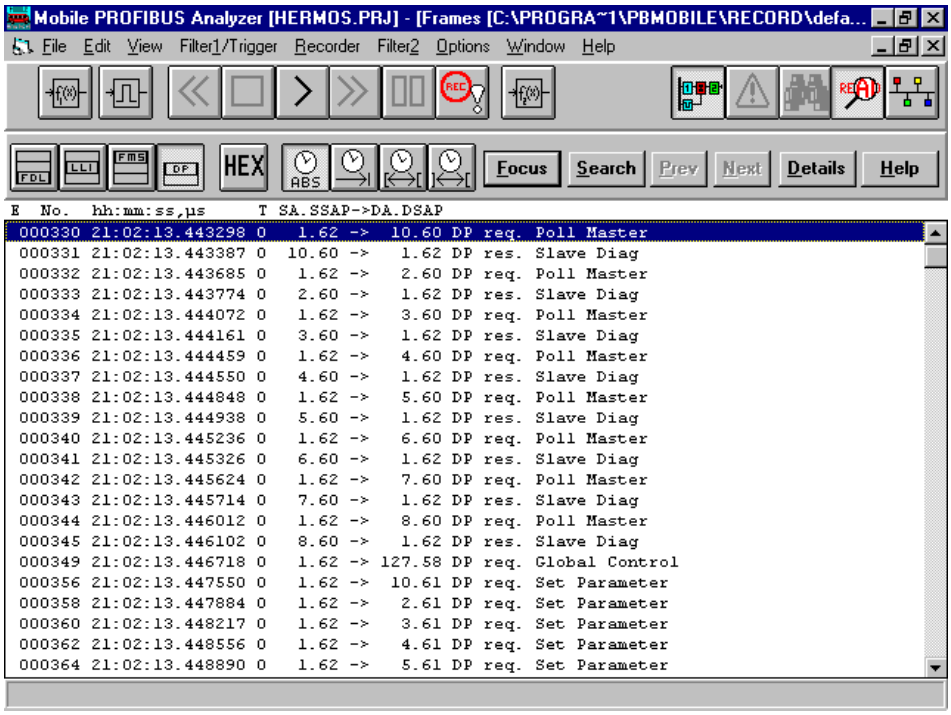
**Bus monitor manufactured by Softing (Germany)**

This bus monitor consists of MS-WINDOWS software, a PCMCIA card and a TAP (Terminal Access Point) adapter, which provides the physical connection between the PCMCIA card and the PROFIBUS.

This monitor can be used with a portable PC to trace and analyse FMS and DP telegrams with time stamping up to baud rates of 12 Mbps. Extensive adjustable filter functions allow detailed fault finding and error analysis.

More information and addresses can be found in the electronic PROFIBUS product catalogue (available from the PROFIBUS user organization).



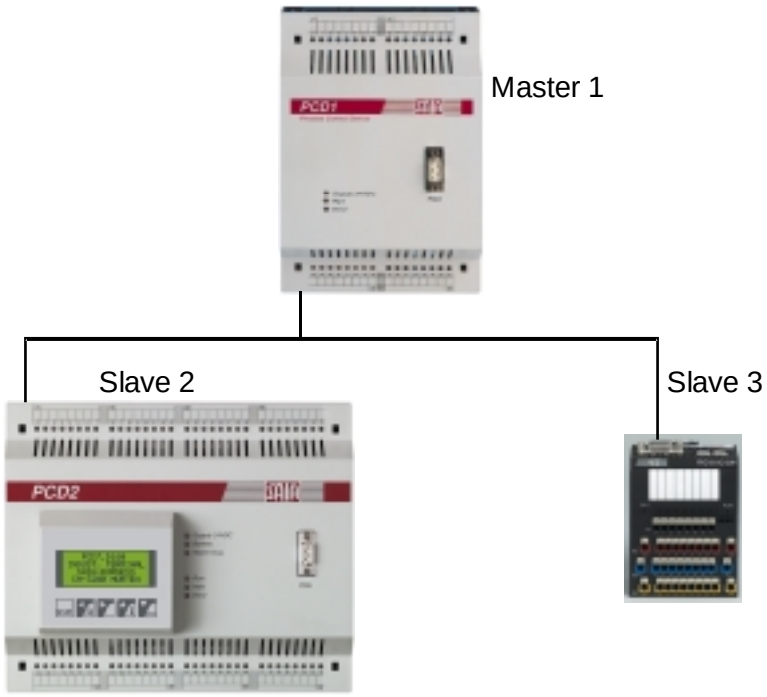



### Working with the bus monitor

Unfortunately, use of a bus monitor still requires the user to have very extensive and thorough knowledge of PROFIBUS. The user must at least be in a position to interpret and understand the DP telegrams traced, in order to carry out an analysis and error search. This means that the user must, for example, be able to interpret the coding of DP telegrams.

# 7. Quick guide to create a PROFIBUS-DP network

A network is to be created with a PCD1 master controller, a PCD2 compact slave controller and a PCD0 compact slave controller.




This involves realizing the following functions via PROFIBUS-DP:

Master 1 (PCD1):
HW: no I/O cards

Function: Increment register 'Val_Sec' in second time.
Transmit 'Val_Sec' to slave 2.
Read inputs I0..I7 of slave 2 and copy these inputs to outputs O0..O7 of slave 3

Slave 2 (PCD2 with D160 terminal plugged on):
HW: Input module with 8 inputs I0..I7 at address 0

Function: Display 'Val_Sec' value on D160 terminal

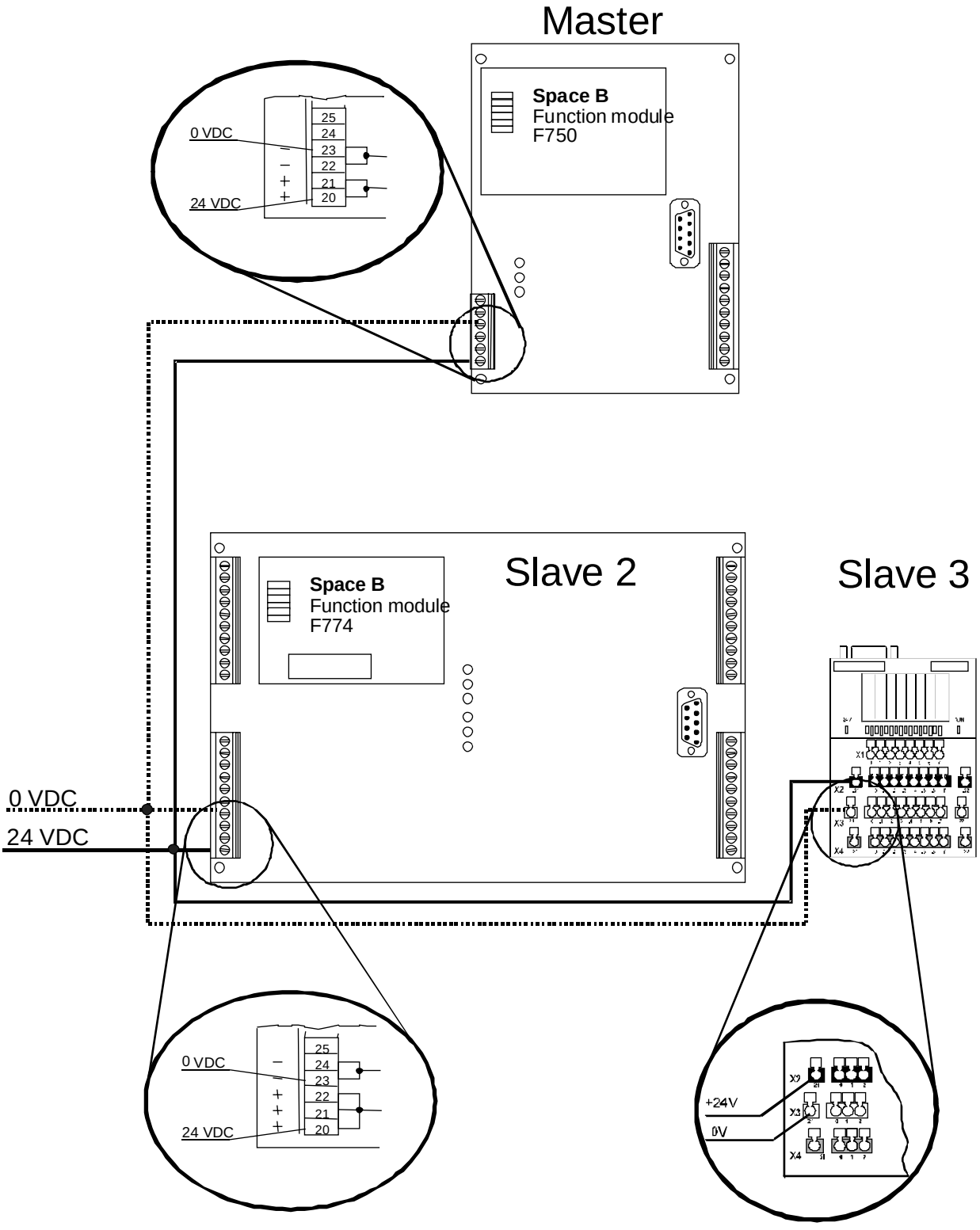
Slave 3 (PCD0.G110 with 8 I/O):
HW/Function: Provision of I/Os

# 7.1 Electrical connection

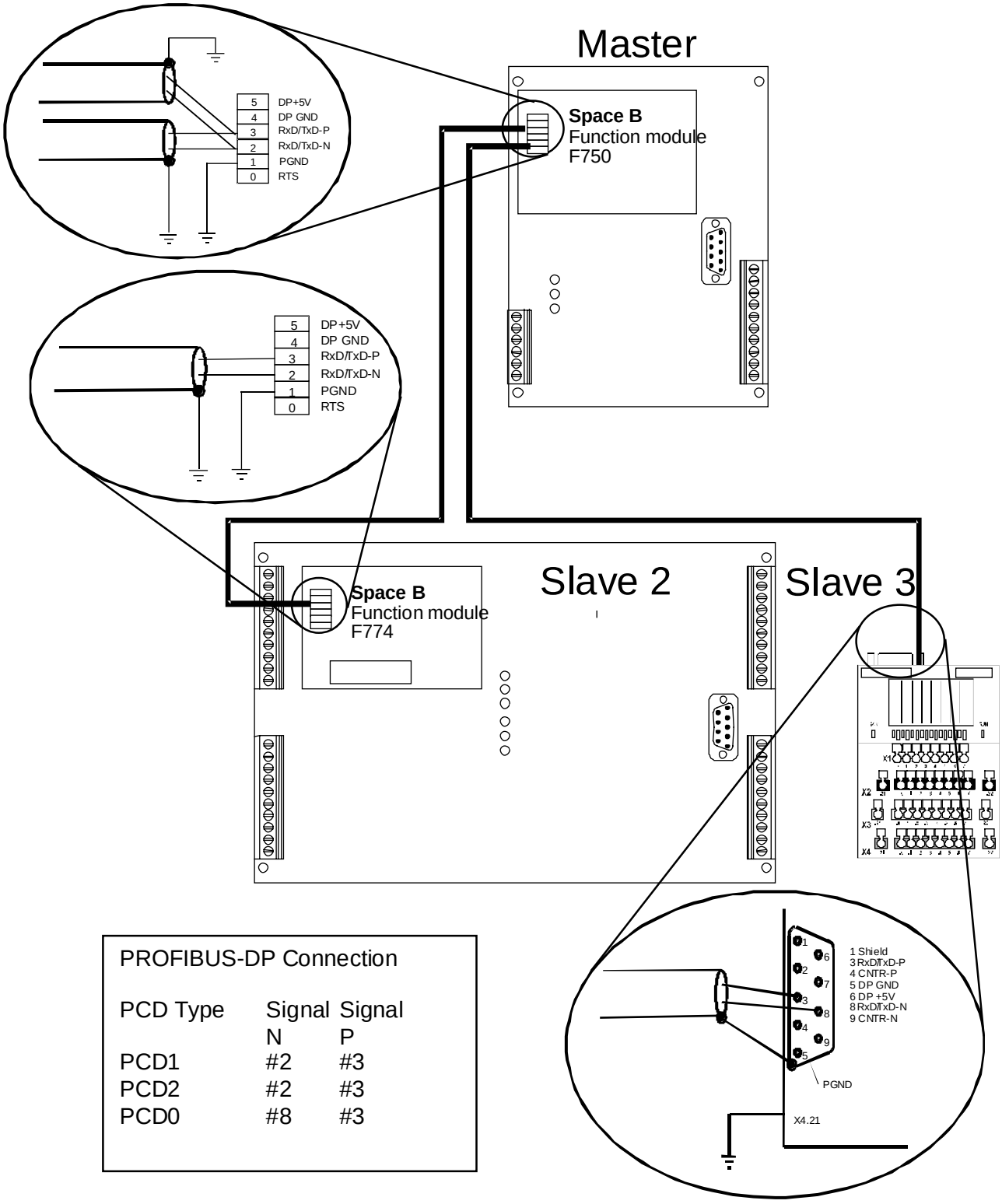
### 7.1.1 Connection of supply

Lift off cover of PCD1 and PCD2 and connect 24VDC supply according to the following diagram.
Function module PCD7.F750 plugs onto space B of the PCD1 and Function module PCD7.F774 plugs onto space B of the PCD2.



### 7.1.2 RS 485 connection

Connect the PROFIBUS-DP line according to the following diagram:

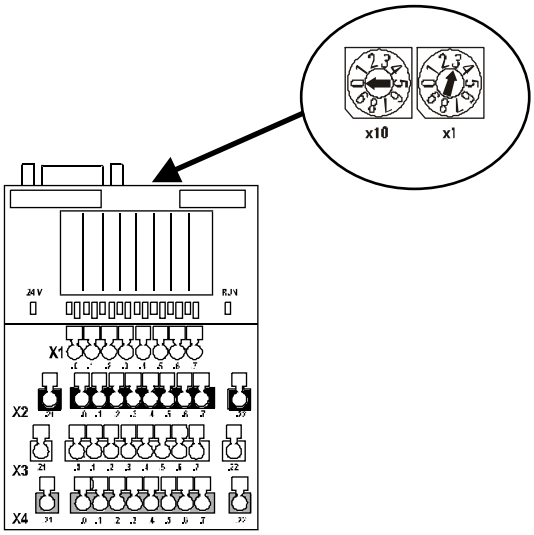



PROFIBUS-DP Connection

| PCD Type | Signal N | Signal P |
|---|---|---|
| PCD1 | #2 | #3 |
| PCD2 | #2 | #3 |
| PCD0 | #8 | #3 |

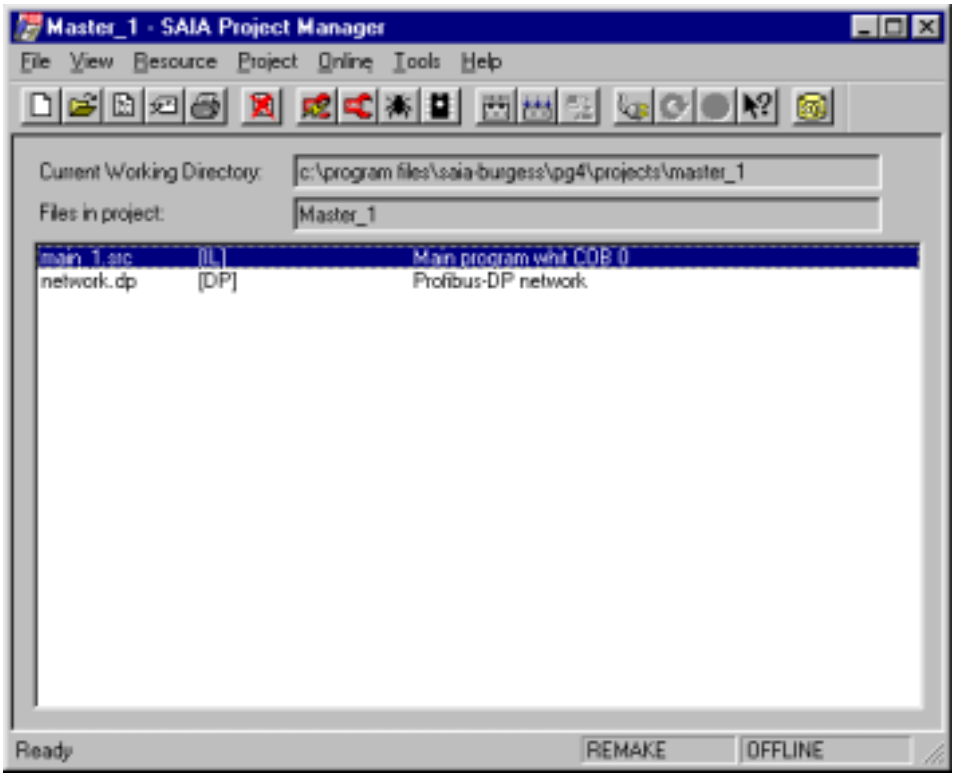
### 7.1.3 Setting the PROFIBUS-DP address

The PROFIBUS-DP address must be set at slave 3 (PCD0) with a rotary switch.
Rotary switch x10: position 0
Rotary switch x1: position 3
With the PCD1 and PCD2 the address is set via the PROFIBUS-DP configurator.

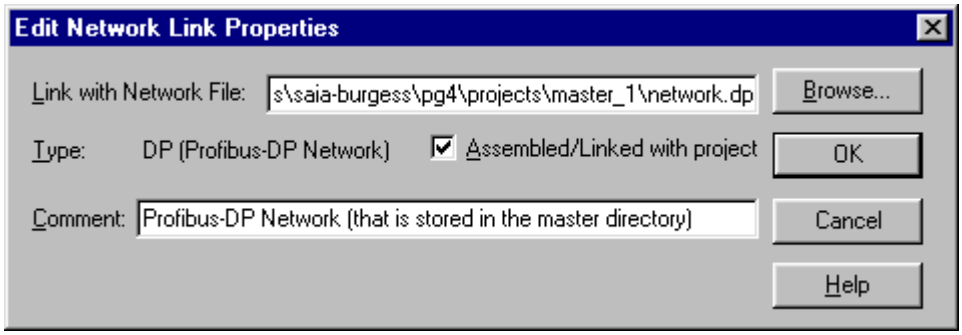


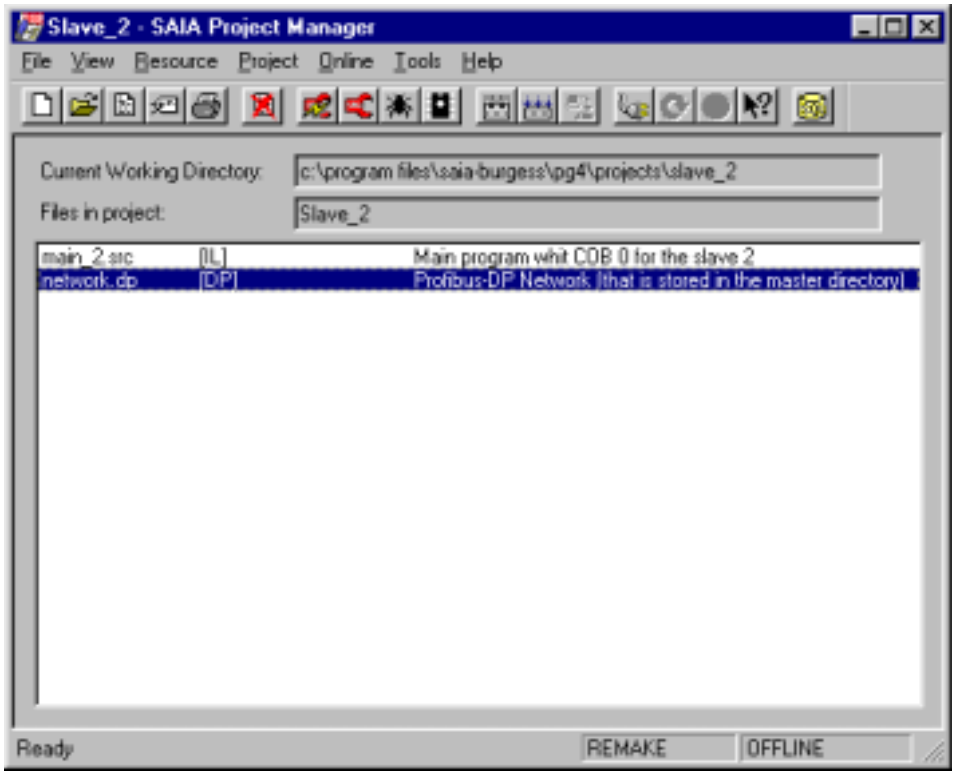
## 7.2 Configuration with the PROFIBUS-DP configurator

- Start up PG4 programming software
- Create two new PG4 projects
  Master_1
  Slave_2
- Load project Master_1
- Set up two files in project Master_1:
  IL with file name Main_1
  DP (Profibus-DP network) with filename Network



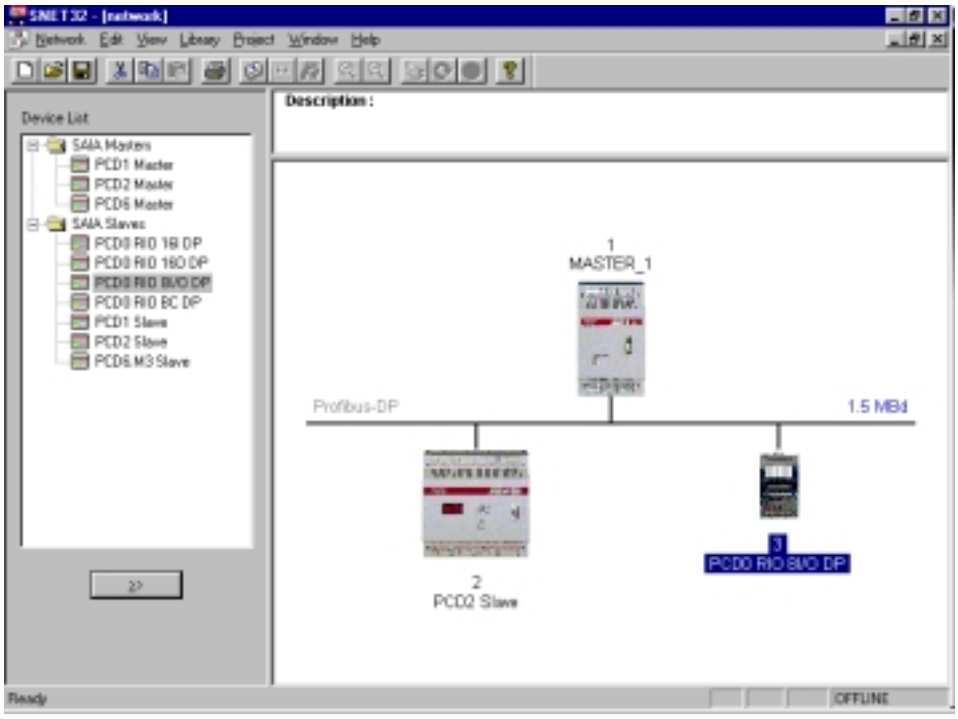

- Load project Slave_2
- Set up two files in project Slave_2
  IL with filename Main_2
  DP (Profibus-DP network) with filename Network, which references the DP file of the project Master_1.
  With the 'Browse' function it is now possible to locate the file 'Network.dp' in the project Master_1.



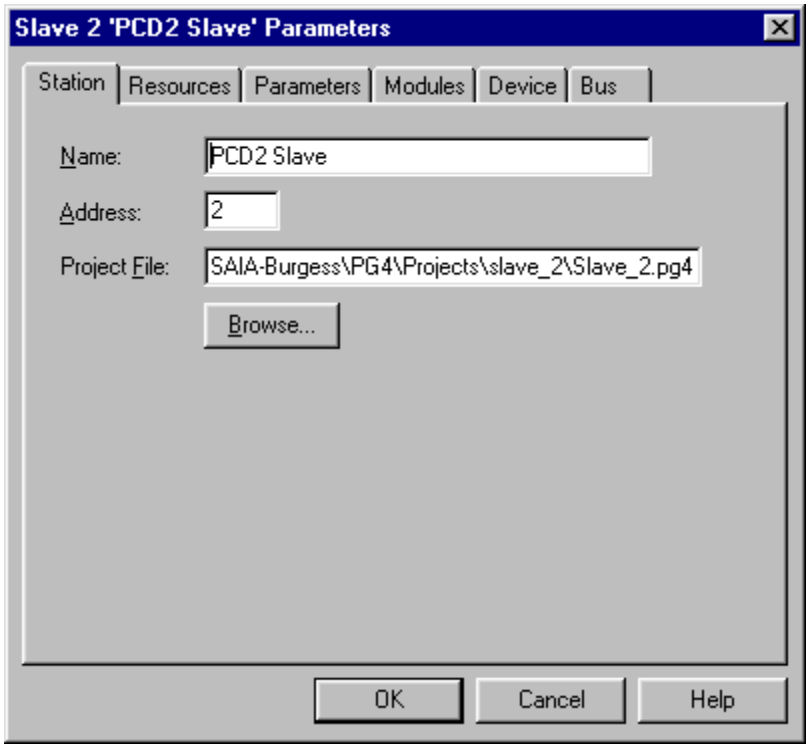



- Change to project Master_1
- Call file network.dp by double-clicking on file name network.dp.
- Insert master PCD1, slave PCD2 and slave PCD0 RIO.8 I/O by double-clicking on the appropriate devices in the 'Device List'

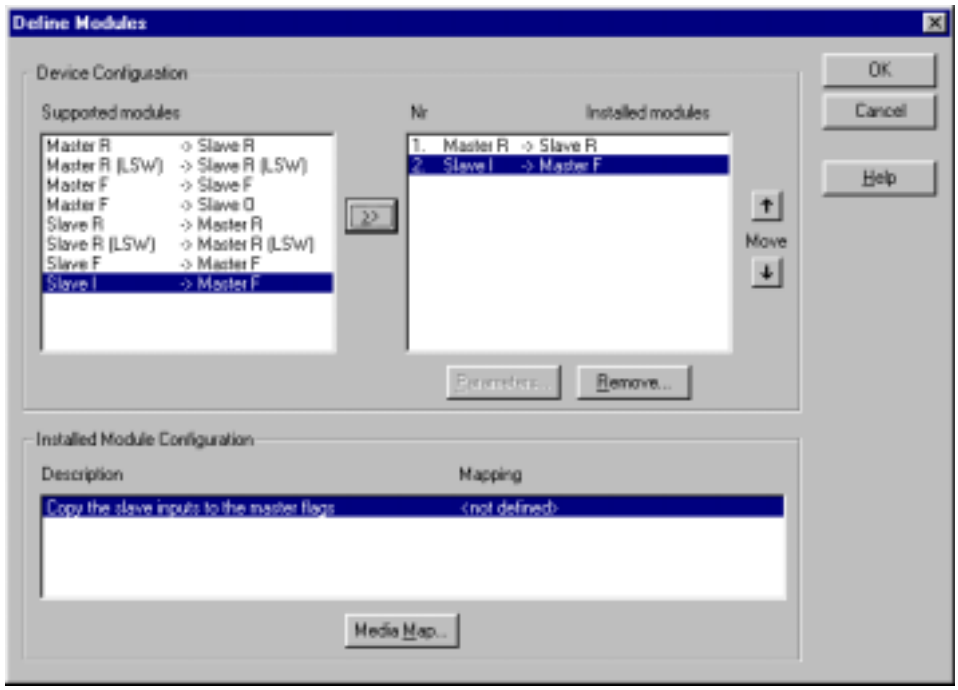
The network then appears as follows:



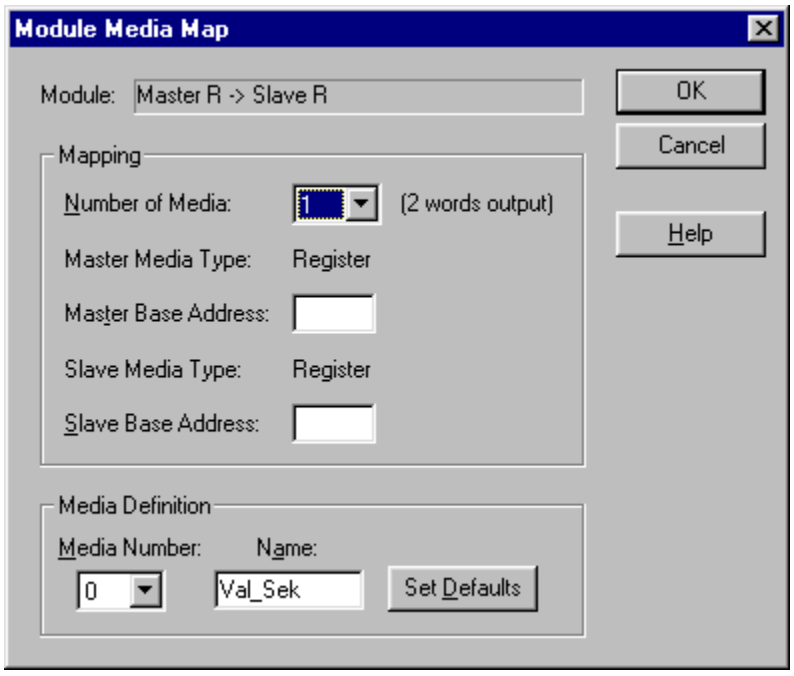
- Double-click on slave 2:
  With the 'Browse' button, the slave's project path must be referenced to the project Slave_2.



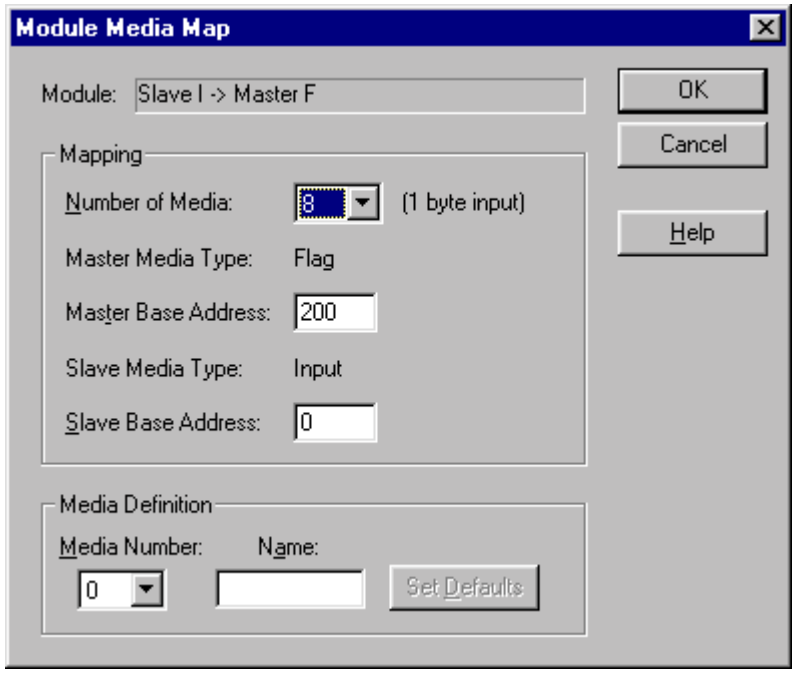

- Under menu item 'Modules', define two modules:
  Master R → Slave R
  Slave I → Master F



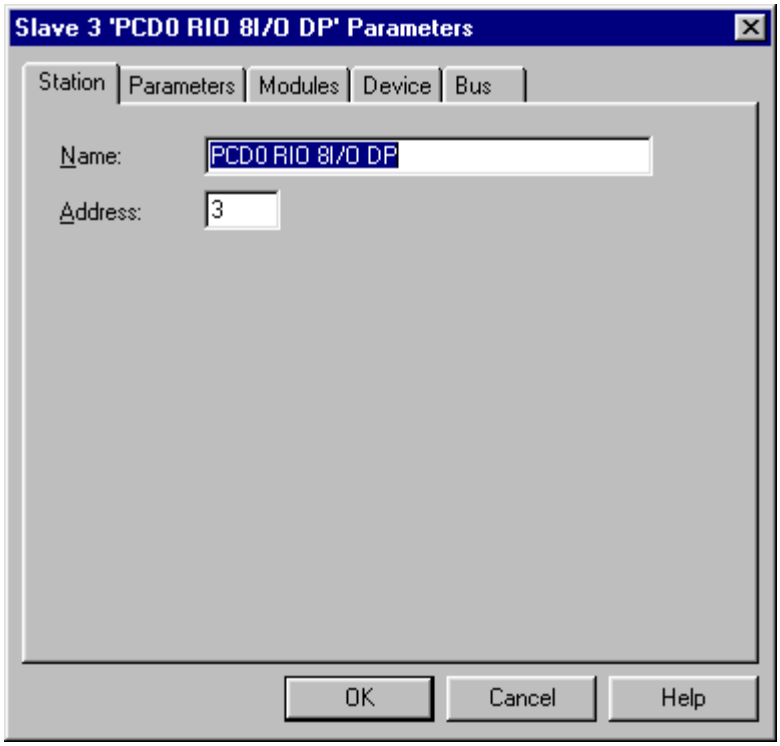
- Map slave media to master media with the 'Media Map' button
- Enter the symbolic name 'Val_Sec' at module Master R -> Slave R



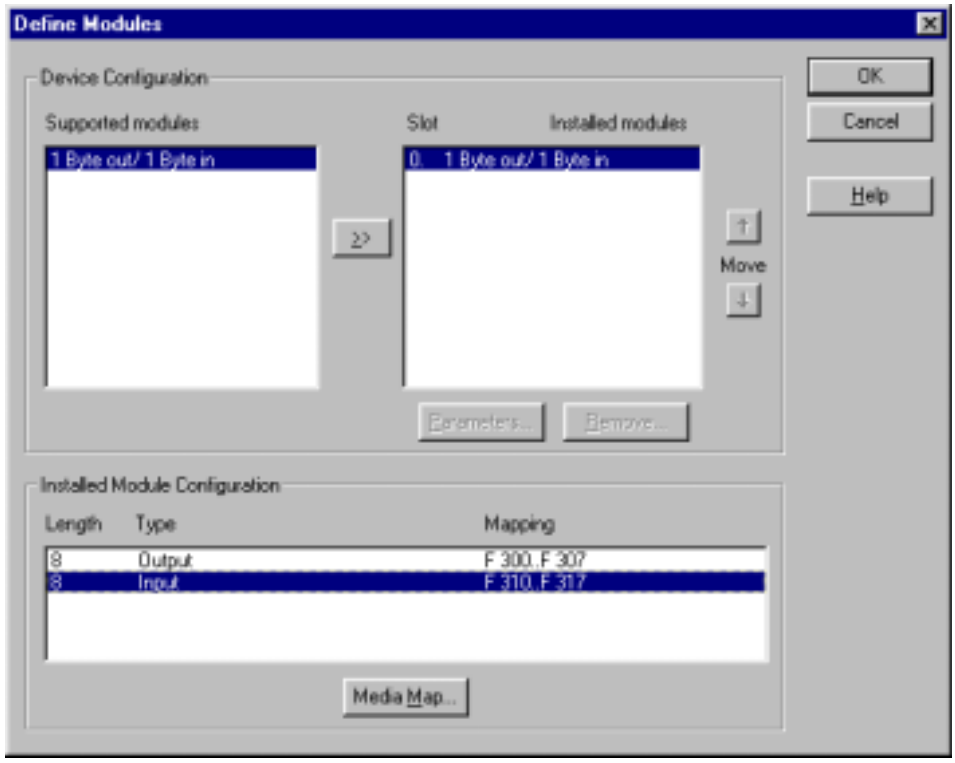

- Enter the absolute address F200 for the master and I0 for the slave at module Slave I –> Master F



- Double-click on slave 3




- Select 'Modules'
- Select 'Media Map'
- Define the absolute address for the module inputs and outputs:
  Outputs 0..7 to flags 300..307
  Inputs 0..7 to flags 310..317

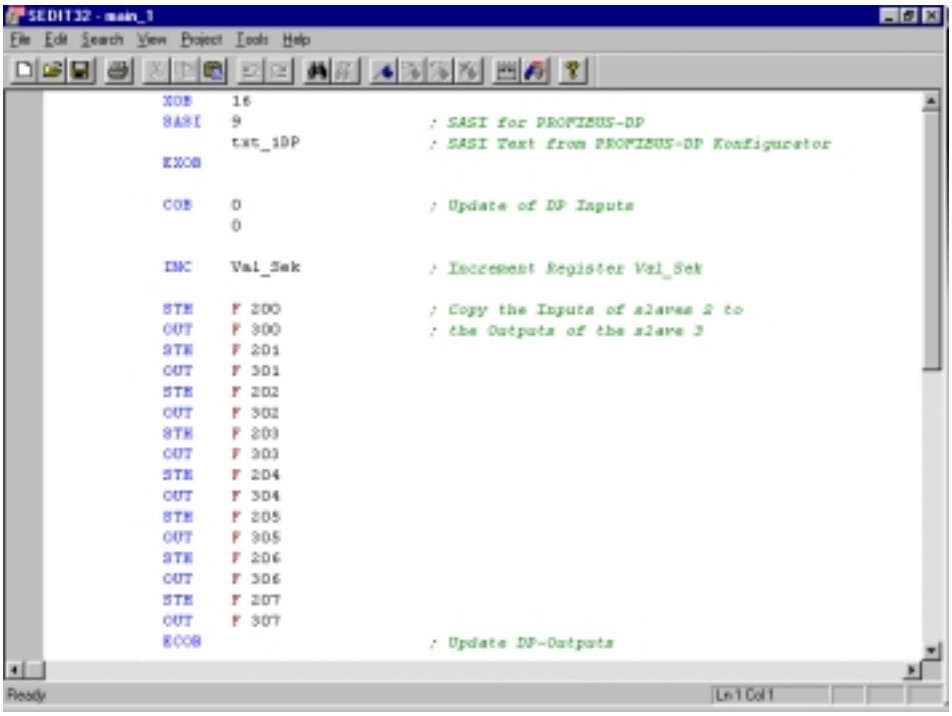


## 7.3 Writing user programs

### 7.3.1 User program in the master

Load the IL editor 'Main_1' in project 'Master_1' and enter the following program:

```
XOB     16
SASI    9               ; SASI for PROFIBUS-DP
        txt_1DP         ; SASI text from PROFIBUS-DP configurator
EXOB

COB     0               ; Update DP inputs
        0

INC     Val_Sec         ; Increment register Val_Sec
STH     F 200           ; Copy inputs of slave 2 to
OUT     F 300           ; outputs of slave 3
STH     F 201
OUT     F 301
STH     F 202
OUT     F 302
STH     F 203
OUT     F 303
STH     F 204
OUT     F 304
STH     F 205
OUT     F 305
STH     F 206
OUT     F 306
STH     F 207
OUT     F 307

ECOB                    ; Update DP outputs
```

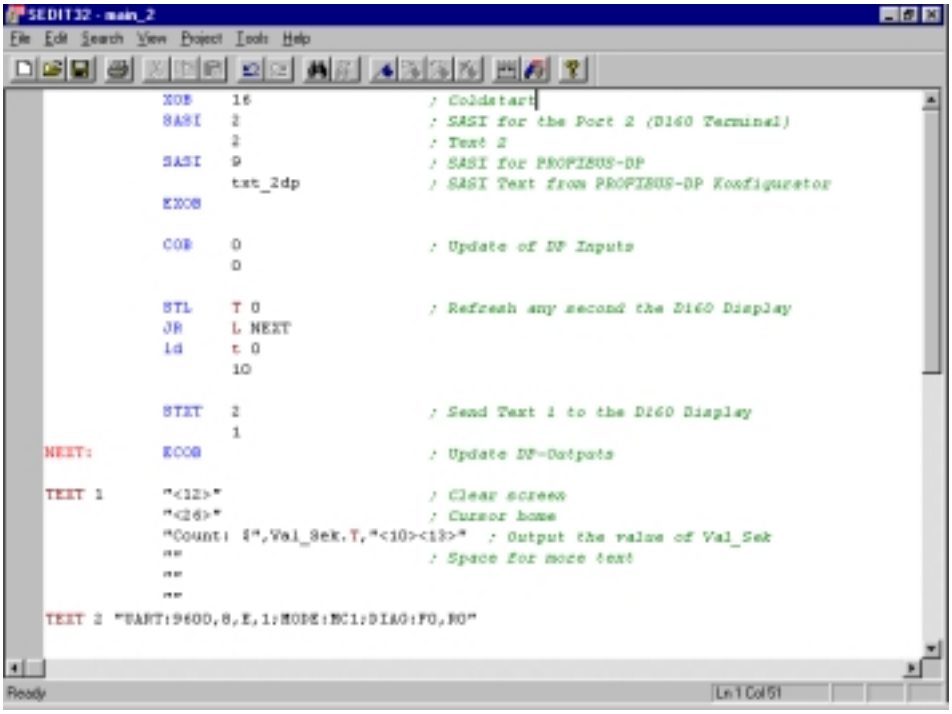



- The project can then be loaded into the master controller with 'Build' and 'Download'.

### 7.3.2 User program in Slave_2

Call the IL editor Main_2 in project Slave_2 and enter the following program:

```
XOB     16              ; Cold start
SASI    2               ; SASI for Port 2 (D160 Terminal)
        2               ; Text 2
SASI    9               ; SASI for PROFIBUS-DP
        txt_2dp         ; SASI text from PROFIBUS-DP configurator
EXOB

COB     0               ; Update DP inputs
        0

STL     T 0             ; New value every second to D160 display
JR      L       NEXT
ld      t 0
        10
STXT    2               ; Transmit Text 1 to the D160 display
        1
ECOB                    ; Update DP outputs

TEXT 1  "<12>"          ; Delete screen
        "<26>"          ; Cursor home
        "Count: $",Val_Sec.T,"<10><13>" ; Output register contents Val_Sec
        ""              ; Space reserved for more text
        ""
        ""
TEXT 2  "UART:9600,8,E,1;MODE:MC1;DIAG:F0,R0"
NEXT:
ECOB                    ; Update DP outputs
```




- The project can then be loaded into the slave controller and tested with 'Build' and 'Download'.

## 7.4 Commissioning user programs

The following functions are now possible with the network.

- Inputs 0 .. 7 of slave 2 are copied to outputs 0 .. 7 of slave 3.
- Register 'Val_Sec' is incremented in the master and transmitted to slave 2.
- Register 'Val_Sec' is written in 1 second time to the display in slave 2.

If you are able to execute the functions listed above, you have installed and programmed everything correctly.

If PROFIBUS-DP fails to function correctly, various software diagnostic capabilities are available to the user in the master and in the slave to locate the error. (See chapters 4 and 5).
However, experience shows that most faults arise on the hardware side. The following points should therefore be checked:

- Are the P and N connections of the RS485 PROFIBUS-DP line correctly hooked up?
- 24VDC supply for devices?
- Controller hardware and firmware versions.
- Is the PCD7.F750 card’s RUN LED flashing every second on the PCD1 when the PCD1 is in RUN?
- Is the address of the PCD0 right?
- Is the BF LED switched off on the PCD0?
- Does the configuration in the PROFIBUS-DP configurator match the hardware used?

# Appendix A. Machine state diagram of a PROFIBUS-DP slave

## 1. Machine state diagram

To understand better how PROFIBUS-DP works, a brief description of the machine state diagram of a DP slaves is reproduced below. The machine state diagram describes how a PROFIBUS station must behave in whatever situation to guarantee conformity. The full description can be obtained from EN 50 170.

In the following illustration, the diagram's machine states are marked in ellipses. Events denote crossing from one state to another. The vertical arrows indicate state transition.

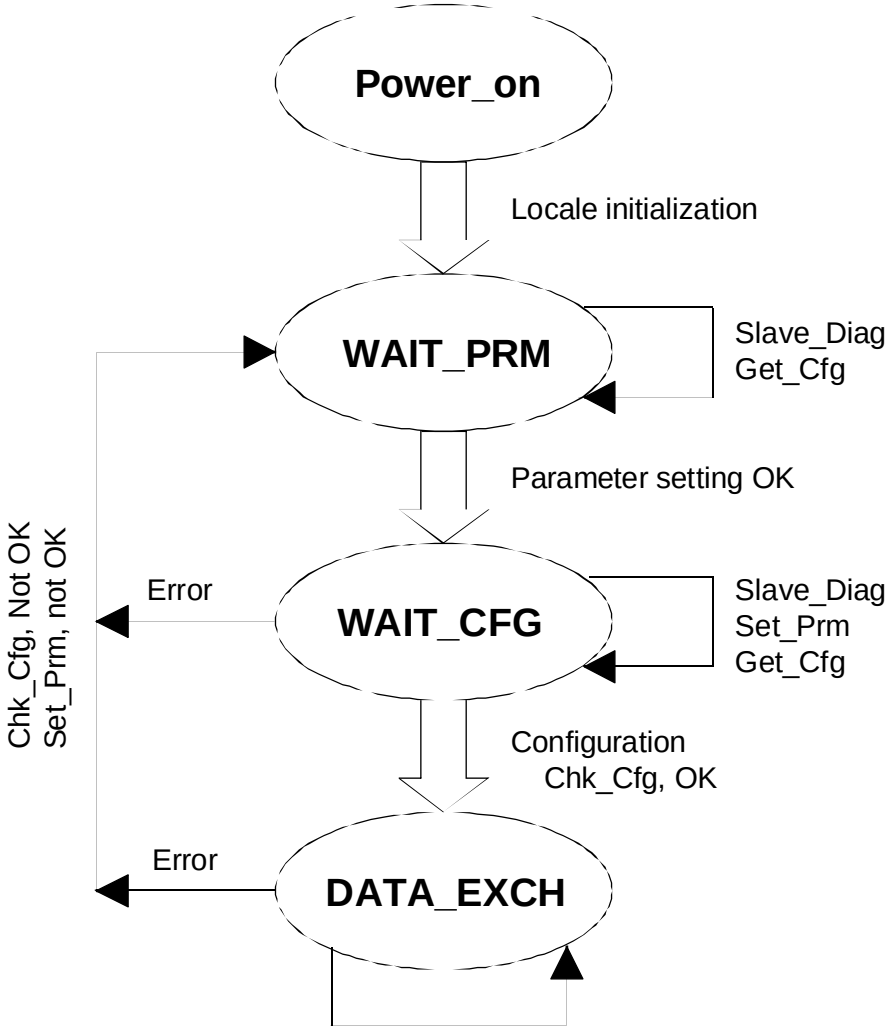



Figure: Machine state diagram of a PROFIBUS-DP slave

**2. Power_On**

Only in the "Power - On" state will a slave station accept a "Set - Slave - Address" telegram from a class 2 master to change station address. For this, the slave must include a non-volatile memory medium in which to store the address.

**3. Wait_Prm (await setting of parameters)**

After the internal run-up, the slave expects a parameter telegram (or "Get-Cfg" telegram). All other types of telegram are either rejected or not processed by the slave. Data exchange is not yet possible.

As a minimum, the parameter telegram carries the information laid down by the standard, such as ident number, sync/freeze capability, watchdog time etc. In addition, user-specific parameter data is possible. The meaning of this data is defined by the application alone.

**4. Wait_Cfg (await configuration)**

The configuration telegram defines the number of input and output bytes. The master informs the slave how many I/O bytes are to be exchanged with that slave in each message cycle. With intelligent slaves, the ASIC transfers the configuration to the application for checking. The result of this verification is then either a correct, incorrect or, for a modular slave, an adaptable configuration.

An additional possibility exists in which the"Get_Cfg" telegram is used to query a master for the configuration of any chosen slave. A slave will accept a "Get_Cfg" telegram, whatever its state.

**5. Data_Exch (data exchange)**

When both the parameters and the configuration have been accepted, the slaves accepts 'Data_Exch' state, i.e. it exchanges useful data with the master.

In DATA_EXCH state, the slave station accepts the following telegrams: 'Data_Exch_ok', 'Rd_Inp', 'Rd_Outp', commands (Sync, Freeze ... ), 'Slave_Diag', 'Chk-Cfg_ok', 'Prm_ok', 'Get_Cfg'.

# Appendix B. CPU load and reaction time when using PROFIBUS-DP

## 1. CPU load when using PROFIBUS-DP

The majority of data exchange with PROFIBUS-DP is processed by the PROFIBUS-DP card: PCD7.F7xx. However, the PCD's CPU must use part of its capacity for data exchange with the PROFIBUS-DP card. The capacity used by the CPU for processing its various tasks is called the CPU load and has a direct influence on the CPU's cycle time, i.e. the CPU's cycle time gets longer the more tasks it has to process. In relation to PROFIBUS-DP, total CPU load can be divided into the following partial loads:

- Program load (T_Progr)
  (processing time for the actual program)
- Normal load for PROFIBUS-DP (T_Normal)
  (Time required by the CPU for updating diagnostic media, processing PROFIBUS-DP routines, checking IL instructions, etc..)
- Communications load (T_Com)
  (Time required by the CPU to exchange data between the PCD's process image memory and the PROFIBUS-DP card memory).

Total load (T_Cycl_Total ) can be calculated from these partial loads.

**Total load =**
**Program load + Normal load + Communications load**

This corresponds to:
**Total cycle time =**
**Cycle time user prog. + Diag update time + I/O update time PCD-DP**

This corresponds to:
**T_Cycl_Total = T_Progr + T_Normal + T_Com**

Comments on individual loads:

### 1.1 CPU program load

This load corresponds to the cycle time of the actual user program (without the PROFIBUS-DP part) and is different in each user program. Program load can be determined by measuring the cycle time (e.g. using instruction SYSRD 7000 to read the 1 ms counter in each cycle and subtracting this value from the old one).

### 1.2 Normal load for PROFIBUS-DP

When working with PROFIBUS-DP, the PCD's CPU must continuously perform some update tasks (for diagnostic media). These update tasks are carried out continuously in the background of the actual user program. This normal load depends only on the number of slaves the master has to deal with.
The diagram below shows the effect of the number of slaves on CPU normal load.

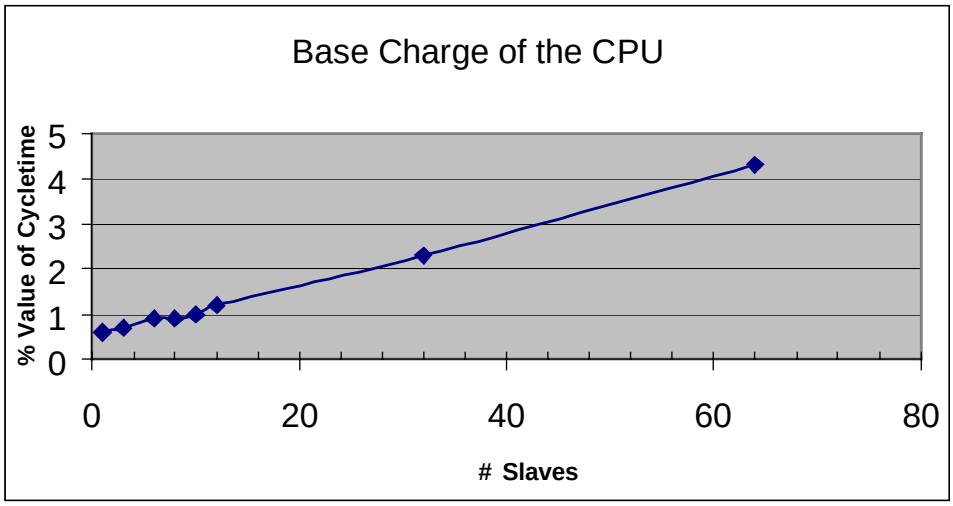



Figure 1: CPU normal load with PROFIBUS-DP

### 1.3 Communications load

Communications load corresponds to the time required by the PCD's CPU to exchange PROFIBUS-DP I/O data between the PCD's process image memory and the PROFIBUS-DP card memory.

Communications load depends on the number and type (bytes or words) of PROFIBUS-DP I/O information, which the master has to read or write. Communications load is independent of cycle time. Regardless of user program length, the communications load will remain the same.

The following diagrams show CPU processing times in relation to media (bytes or words) and the number of slaves:

The amount of I/O data is given per slave. i.e.
8 PCD flags produce 1 byte, 1 PCD register produces 2 words.

#### 1.3.1 Data exchange with byte information

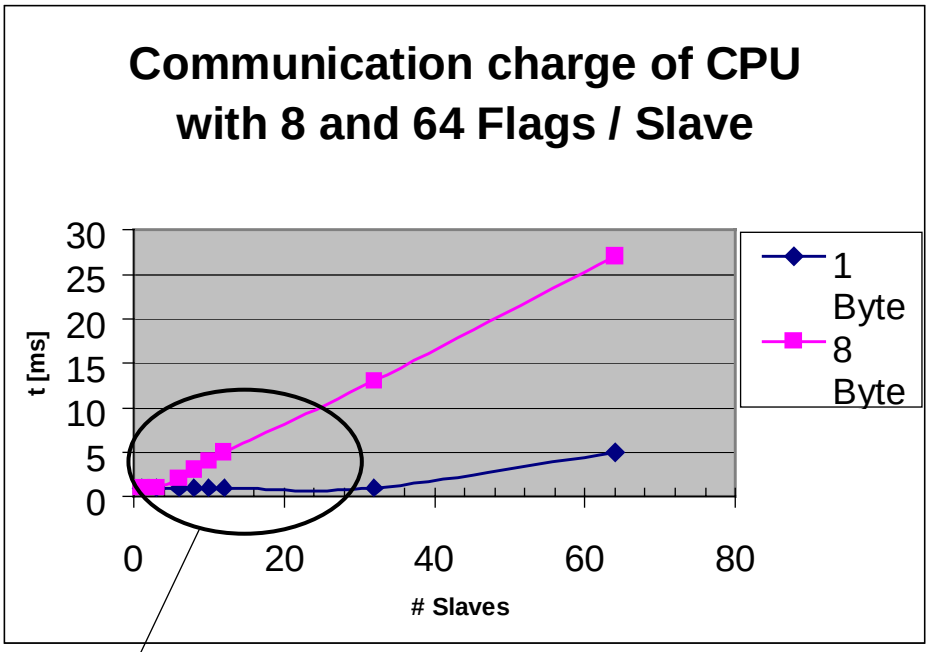



Figure 2: Communications load with 8 and 64 flags; up to 64 slaves

Figure 3 shows the range up to 12 slaves in detail.

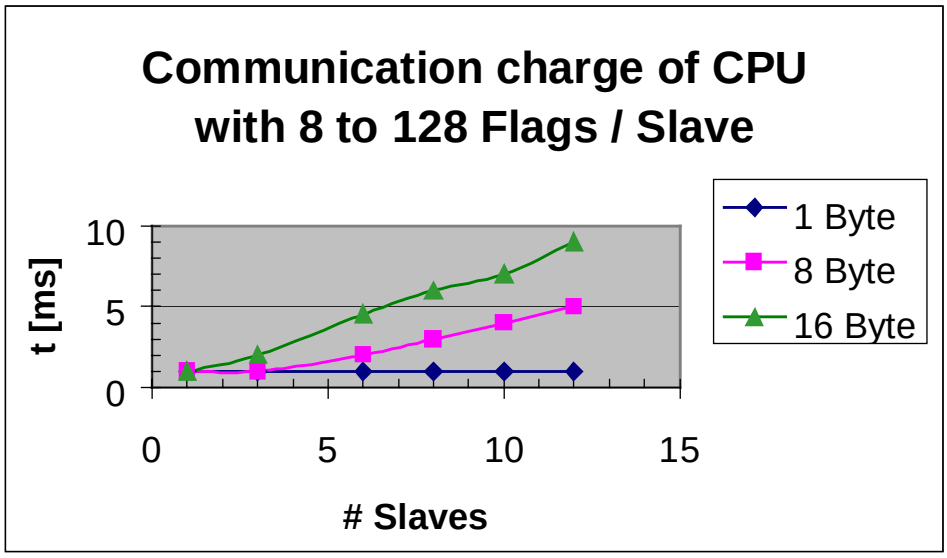



Figure 3: Communications load with 8, 64 and 128 flags; up to 12 slaves.

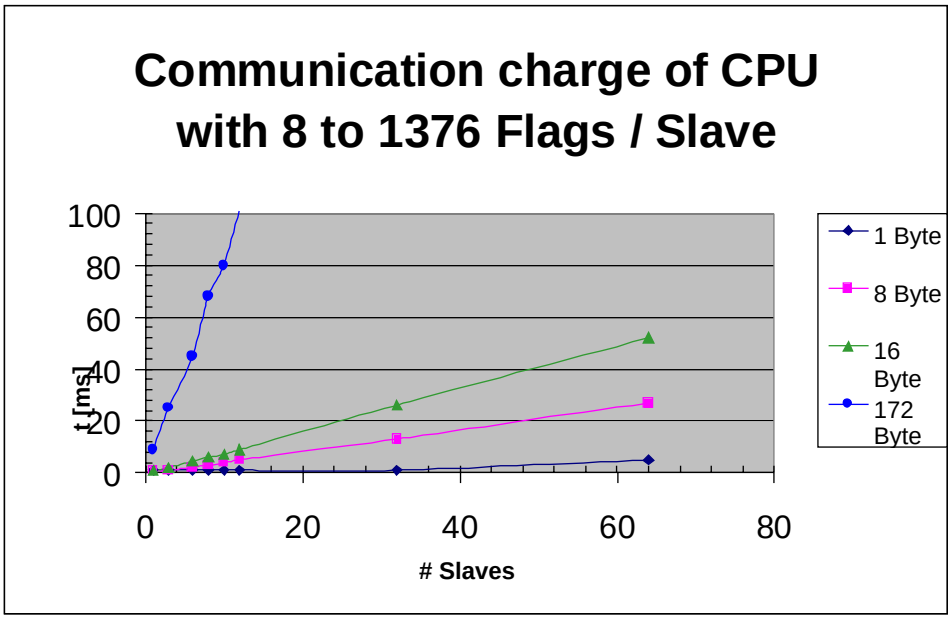



Figure 4: Communications load with 8, 64, 128 and 1376 flags; up to 64 slaves

### 1.3.2 Data exchange with word information

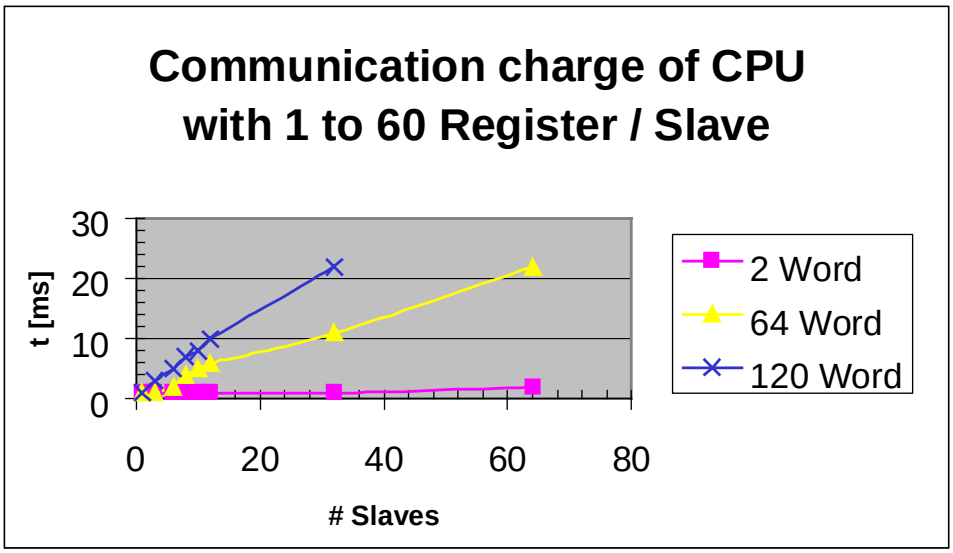



Figure 5: Communications load between 1 and 60 registers; 64 slaves

**Calculation of CPU cycle time**

Calculation of program cycle time when working with PROFIBUS-DP:
For this, the following parameters must be known:

- Cycle time of program without PROFBIUS-DP communication
- Number of slaves
- Number and type of I/O data for PROFIBUS-DP.

Calculation of the cycle time is then based on the following formula:

**Total load =**
**Program load + Normal load + Communications load**

**Total cycle time =**
**Cycle time user prog. + Diag update time + I/O update time PCD-DP**

Example:

| | | |
|---|---|---|
| Program load | (cycle time user program) | = 20 ms |
| Normal load | (number of slaves) | = 12 |
| Communications load | (number of DP I/Os/slave) | = 8 flags (1byte) |

From the preceding tables, the following values can be read.
The cycle time with DP is:

20 ms + 1.2% of 20ms + 1 ms = 21.25 ms

i.e., Processing 96 inputs from 12 slaves results in a 1.25 ms increase to the cycle time.

### 1.4 Comparison of CPU load between S-Bus and PROFIBUS-DP

This graph compares CPU load under PROFIBUS-DP against that under SAIA S-Bus (S-Bus with 38.4 kBaud, PROFIBUS-DP with 12 MBaud, reading 8 flags (1 byte) for 8 slaves).

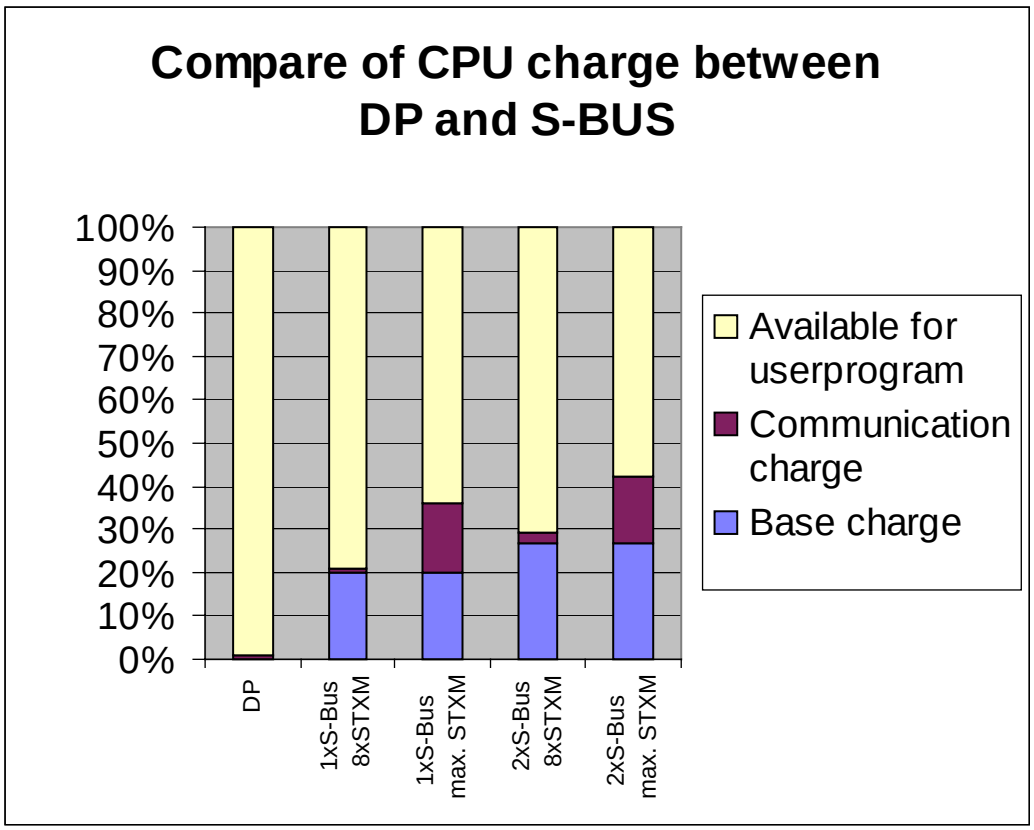



Figure 6: Comparison of CPU load S-Bus / DP

The graph demonstrates that, when an S-Bus connection is employed, depending on the number of STXM instructions, only approx. 64% of CPU capacity can still be used for the application program. If S-Bus is used on 2 ports at once, CPU capacity for the application program is thereby reduced to approx. 58%.

With PROFIBUS-DP, CPU capacity for the user program is approx. 99%, due to the PROFIBUS-DP card. This means that, with PROFIBUS-DP, roughly between 20% and 45% more CPU capacity is available to the user program than with S-Bus.

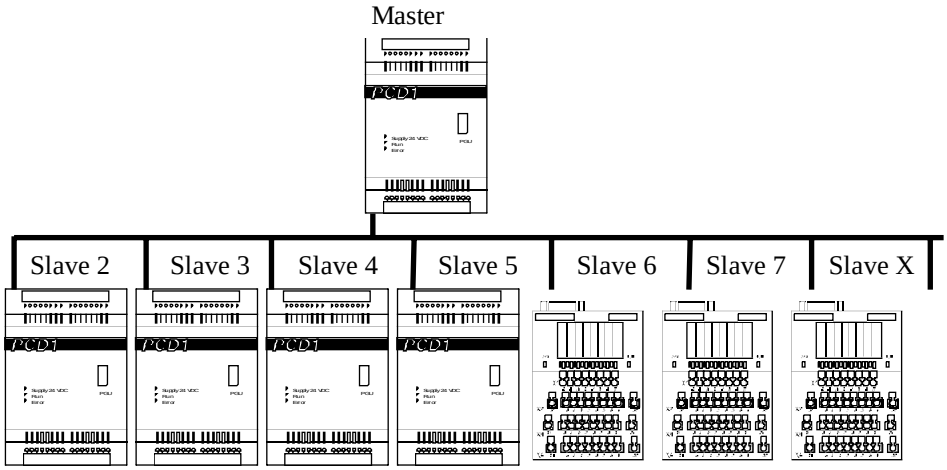
## 2. Reaction times with PROFIBUS-DP

Because of the individual PROFIBUS-DP processor and the very fast transmission rates, with PROFIBUS-DP very fast reaction times to I/O signals are possible.

The reaction time depends on the following factors:

- Cycle time of the user program.
- Communications load
- PROFIBUS-DP baud rate
- Point in time when PROFIBUS-DP media in the user program are read or written

In the following diagrams, the reaction time is based on the following test structure:




The master is to send slave X the following messages:

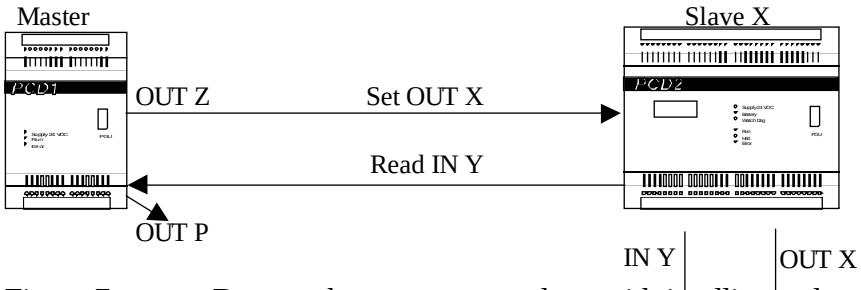



Figure 7: Data exchange master – slave with intelligent slave

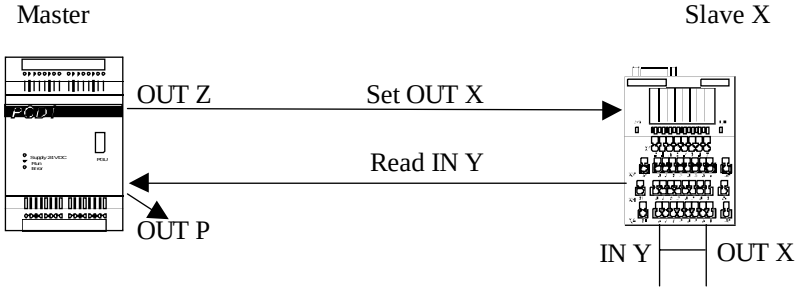



Figure 8: Data exchange master – slave with non-intelligent slave

Every second, the master changes the status of output 'Out Z'. This output is copied to flag 'W', which is defined as an output on the slave. Using PROFIBUS-DP, flag 'W' is transferred to the slave. Flag 'W' is copied in the slave from output 'Out X'.

On the slave, output 'Out X' is electrically connected with input 'In Y'.

The status of input 'In Y' is sent banckk to the master and copied to flag 'V'. The status of flag V is then copied to output 'Out P'.

The time difference between switching on outputs 'Out Z' and 'Out P' is always measured at the slave with the highest address.

This data exchange can be represented in a graph as follows:

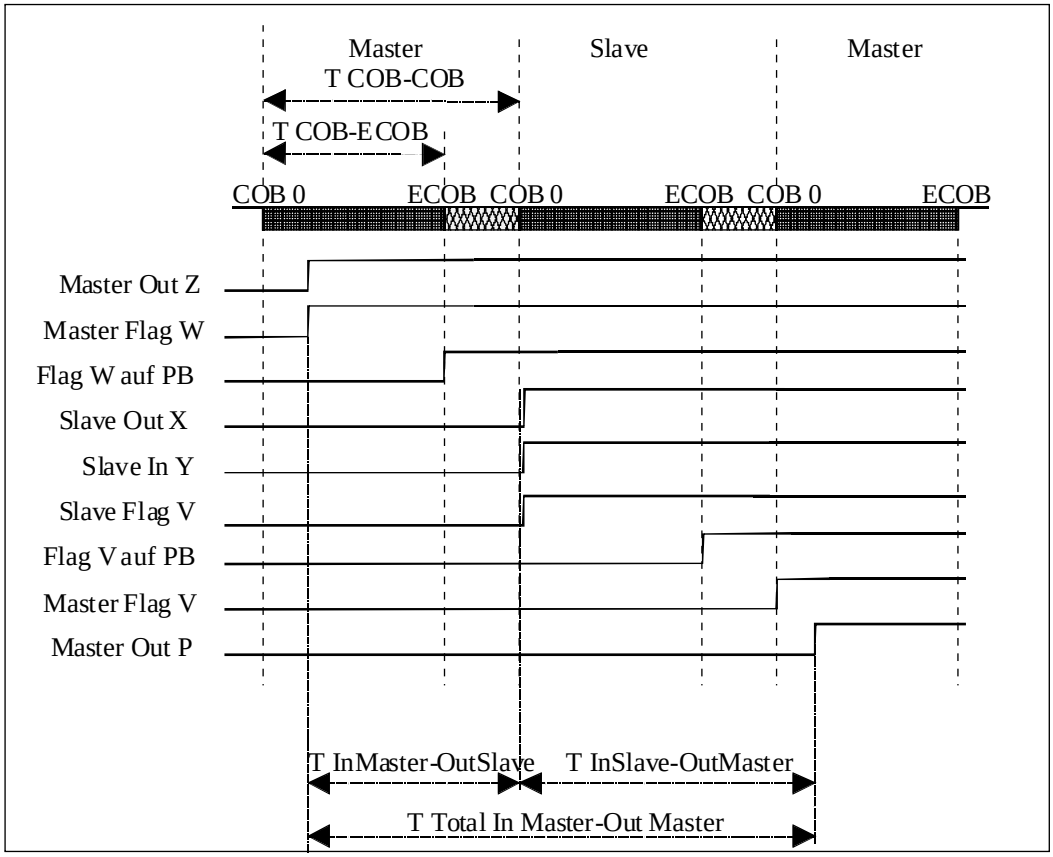



Figure 9: Data flow chart of test structure

Table of measurement results with PCD0 slaves:

| Baudrate [MBaud] | # PCD0 Slaves | # Total Bytes | T COB-ECOB | T COB-COB | T InMaster-OutSlave | T InSlave-Out-Master | T Total In Master-Out Master | T cycle Profibus Master Slave | T cycle Profibus Message - Message |
|---|---|---|---|---|---|---|---|---|---|
| | | | mi-cro | mili | mili | mili | mili | micro | mili |
| 12 | 90 | 2160 | 400 | 22 | 28 | 38 | 66 | 12 | 22.71 |
| 12 | 40 | 960 | 400 | 10 | 12.8 | 17.2 | 30 | 12 | 9.96 |
| 12 | 32 | 768 | 400 | 8.3 | 9.6 | 14.4 | 24 | 12 | 8.12 |
| 12 | 16 | 384 | 400 | 4.1 | 5.3 | 7.5 | 12.8 | 12 | 4.65 |
| 12 | 8 | 192 | 400 | 2.4 | 3 | 6.7 | 9.7 | 12 | 2.95 |
| 12 | 4 | 96 | 400 | 1.5 | 1.8 | 5.1 | 6.9 | 12 | 1.65 |
| 12 | 2 | 48 | 400 | 0.98 | 1.6 | 4.1 | 5.7 | 12 | 0.79 |
| 12 | 1 | 24 | 400 | 0.65 | 0.8 | 3.8 | 4.6 | 12 | 0.67 |
| | | | | | | | | | |
| 6 | 90 | 2160 | 400 | 22 | 31 | 35 | 66 | 22 | 22.16 |
| 6 | 32 | 768 | 400 | 8.2 | 11.6 | 12.4 | 24 | 22 | 7.78 |
| 6 | 4 | 96 | 400 | 1.5 | 2 | 5.1 | 7.1 | 22 | 1.3 |
| | | | | | | | | | |
| 1.5 | 90 | 2160 | 400 | 21.9 | 72 | 59 | 131 | 85 | 43.9 |
| 1.5 | 32 | 768 | 400 | 8.3 | 18 | 22 | 40 | 85 | 16.35 |
| 1.5 | 16 | 384 | 400 | 4.1 | 9.4 | 11.8 | 21.2 | 85 | 8.6 |
| 1.5 | 8 | 192 | 400 | 2.4 | 5.7 | 6.5 | 12.2 | 85 | 5.13 |
| 1.5 | 4 | 96 | 400 | 1.5 | 3.2 | 5.7 | 8.9 | 85 | 2.08 |
| 1.5 | 2 | 48 | 400 | 0.95 | 1.3 | 5.3 | 6.6 | 85 | 1.34 |
| 1.5 | 1 | 24 | 400 | 0.76 | 1.25 | 4.1 | 5.35 | 85 | 0.58 |
| | | | | | | | | | |
| 0.5 | 90 | 2160 | 400 | 21.9 | 180 | 129 | 309 | 250 | 101 |
| 0.5 | 32 | 768 | 400 | 8 | 38 | 43 | 81 | 250 | 36.9 |
| 0.5 | 4 | 96 | 400 | 1.3 | 8.7 | 7.1 | 15.8 | 250 | 4.7 |
| | | | | | | | | | |
| 0.1875 | 90 | 2160 | 400 | 22 | 370 | 244 | 614 | 666 | 229 |
| 0.1875 | 32 | 768 | 400 | 8.2 | 100 | 87 | 187 | 666 | 82.5 |
| 0.1875 | 4 | 96 | 400 | 1.32 | 12 | 13 | 25 | 666 | 10.98 |
| | | | | | | | | | |
| 0.09375 | 90 | 2160 | 400 | 21.8 | 550 | 450 | 1000 | 1323 | 438 |
| 0.09375 | 32 | 768 | 400 | 7.8 | 250 | 164 | 414 | 1323 | 155 |
| 0.09375 | 4 | 96 | 400 | 1.3 | 30 | 22 | 52 | 1323 | 22.2 |
| | | | | | | | | | |
| 0.0192 | 90 | 2160 | 400 | 21.8 | 550 | 450 | 1000 | 1323 | 438 |

Graphical representation of measurement results:

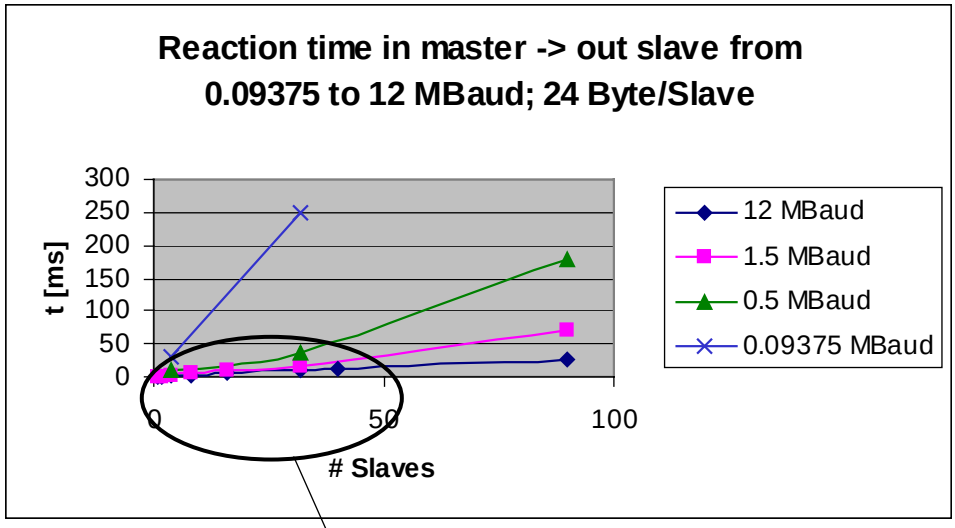



Figure 10: Reaction time with 93.75 and 500kBaud, 1.5 and 12 MBaud

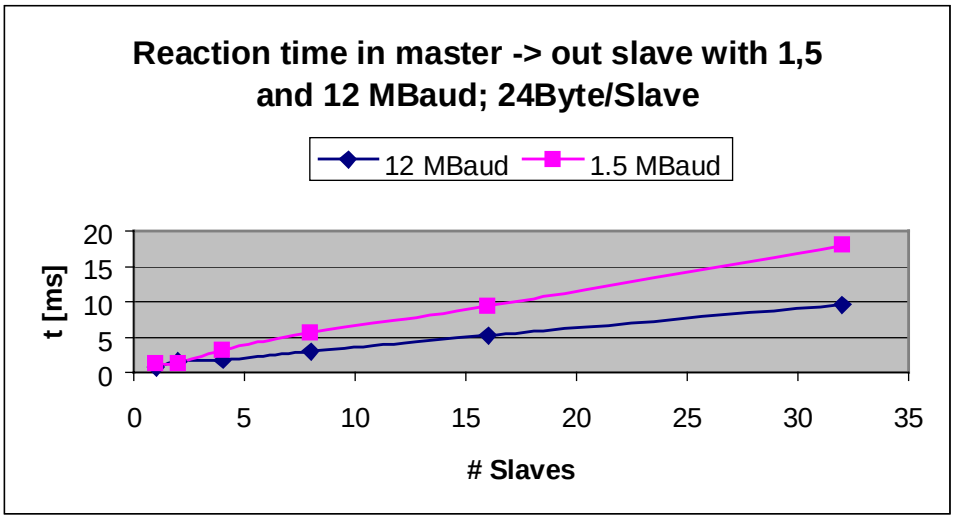



Figure 11: Reaction time with 1.5 and 12 MBaud

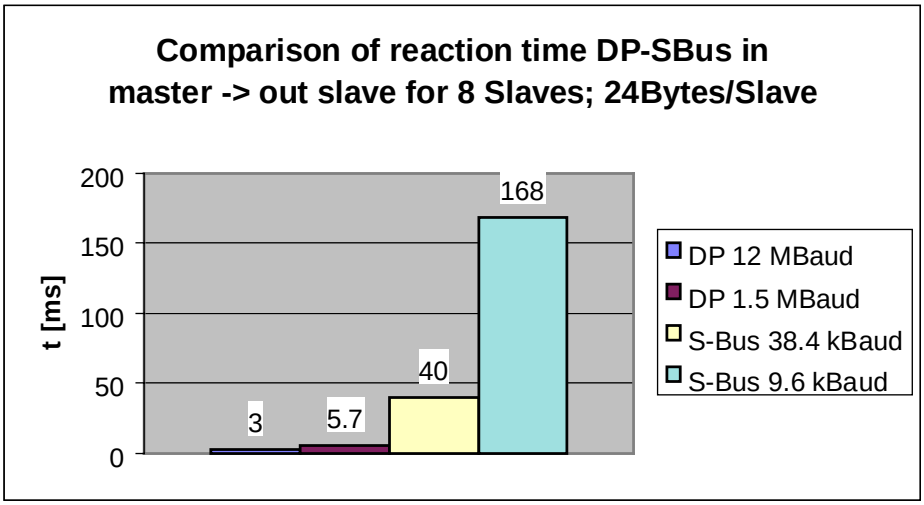



Figure 12: Comparison of reaction times between PROFIBUS-DP and S-Bus

The above graph clearly shows that PROFIBUS-DP allows very short reaction times to be realized, which cannot be achieved with S-Bus, for example.

Notes

# Appendix C. Tested Non-SAIA PROFIBUS-DP devices

**Up to now the following Non-SAIA PROFIBUS-DP devices have been tested.**

| DP-Slave | | | | |
|---|---|---|---|---|
| **Supplier** | **Type** | **Master/ Slave** | **result** | **Remark** |
| Hirschmann | LWL-Converter OZD Profi G4a | Converter | OK | |
| Siemens | LWL-Converter Sinec L2FO OLM / S4 | Converter | OK | |
| ABB | Drive ABB-ACS600-NPBA-02 | Slave | OK | |
| Bihl+Wiedemann | AS-i/DP-Gateway | Slave | OK | |
| Bürkert | Magnet valve system 8640 | Slave | OK | |
| Festo | CP- Magnet valve system CP FB13 | Slave | OK | |
| Festo | Magnet valve CP FB09 E | Slave | OK | |
| Heidenhain | Encoder EnDat/DP-Gateway | Slave | OK | |
| Mannesmann Rexroth | HNC100 | Slave | OK | |
| Murr Elektronik | Terminals ME MBS GP | Slave | OK | |
| Siemens | Terminals ET200L-SC | Slave | Not OK | This device works with a Siemens Master only |
| Siemens | Terminals ET200B 16DI | Slave | OK | |
| Siemens | Terminals ET200B 16DO | Slave | OK | |
| Siemens | Terminals ET200B 4AI | Slave | OK | |
| Siemens | Drive CB15 | Slave | OK | |
| SMC | Magnet valve EX 121-SPR1 | Slave | OK | |
| VIPA | Terminals ET 200V | Slave | OK | |
| Wago | Terminals Wago I/O System | Slave | OK | |
| Weidmüller | Terminals Winbloc 8 DI | Slave | OK | |
| Weidmüller | Terminals Winbloc 8 DO | Slave | OK | |
| Weidmüller | Terminals Winbloc 3AI 1 AO | Slave | OK | |
| Weidmüller | Terminals Winbloc 16 DI /16 DO 0.5 A eco | Slave | OK | |
| | | | | |

| DP Master | | | | |
|---|---|---|---|---|
| **Supplier** | **Type** | **Master/ Slave** | **result** | **Remark** |
| Siemens | Simatic S 7 | Master | OK | |
| | | | | |

| PA devices | | | | |
|---|---|---|---|---|
| **Supplier** | **Type** | **Master/ Slave** | **result** | **Remark** |
| Bürkert | Magnet valve 6520 Namur | PA Slave | OK | |
| Endress+ Hauser | Pressure measuring unit Del-tabar S | PA Slave | OK | |
| Hartmann & Braun | Temperature sensor Contrans T TM 211 | PA Slave | OK | |
| Pepperl+ Fuchs | DP/PA Coupler | DP/PA Coupler | OK | |
| Siemens | Pressure sensor Sitrans | PA Slave | OK | |

Note about PA devices:

The use of these devices requires detailed PROFIBUS know-how.

---

The test reports of each tested device can be seen on the PCD Support homepage under "Communications" - "PROFIBUS-DP".

URL: http://www.saia-burgess.com/pcdsupport/

From :

Company :
Department :
Name :
Address :

Tel. :

Date :

Send back to :

SAIA-Burgess Electronics Ltd.
Bahnhofstrasse 18
CH-3280 Murten (Switzerland)
http://www.saia-burgess.com

BA : Electronic Controllers

PROFIBUS-DP with SAIA® PCD

If you have any suggestions concerning the SAIA® PCD, or have found any errors in this manual, brief details would be appreciated.

**Your suggestions :**