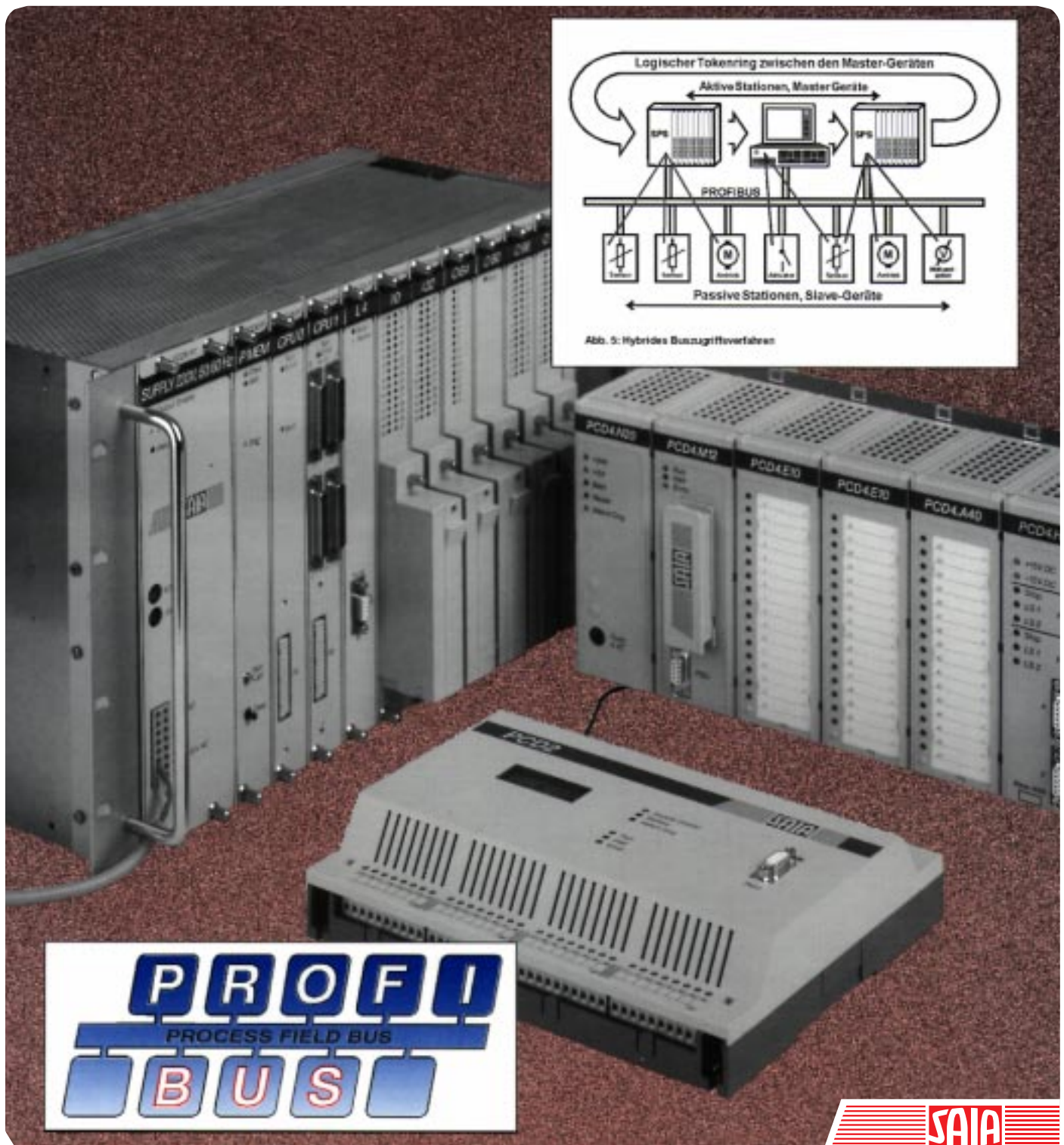


## SAIA®PCD Process Control Devices

## Manual PROFIBUS



## SAIA-Burgess Companies

<b>Switzerland</b>	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	<b>France</b>	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
<b>Germany</b>	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	<b>Netherlands</b>	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
<b>Austria</b>	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	<b>Belgium</b>	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
<b>Italy</b>	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	<b>Hungary</b>	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

## Representatives

<b>Great Britain</b>	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	<b>Portugal</b>	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
<b>Denmark</b>	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	<b>Spain</b>	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
<b>Norway</b>	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	<b>Czech Republic</b>	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
<b>Sweden</b>	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	<b>Poland</b>	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
<b>Suomi/ Finland</b>	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
<b>Australia</b>	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	<b>Argentina</b>	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

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

**SAIA® Process Control Devices**

**Manual**

**PROFIBUS-FMS**

**with SAIA® PCD**

SAIA-Burgess Electronics Ltd. 1996. All rights reserved  
Edition 26/742 E1 - 06.1996

Subject to technical changes

# Updates

---

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

Date	Chapter	Page	Description
08.11.2000	---	---	Small updates for the "Support Homepage"

# Contents

---

	Page
<b>1. Properties and general</b>	<b>1-1</b>
1.1 Properties	1-1
1.2 Principles	1-4
1.2.1 PROFIBUS Physical Layer (1)	1-4
1.2.2 PROFIBUS Data Link Layer (2)	1-4
1.2.3 PROFIBUS-FMS	1-7
<b>2. SAIA® PCD devices for the PROFIBUS network</b>	<b>2-1</b>
2.1 The processor module PCD4.M445	2-1
2.2 PROFIBUS-MS connection, type PCD7.F700 for PCD2.M120	2-2
2.3 Data for the PROFIBUS connection	2-3
2.4 Repeater PCD7.T100	2-4
2.5 Termination-Box PCD7.T160	2-5
<b>3. Planning and realizing a PROFIBUS network</b>	<b>3-1</b>
3.1 Physical Layer	3-1
3.2 Logical network structure	3-2
<b>4. The PROFIBUS configurator</b>	<b>4-1</b>
4.1 General	4-1
4.2 Network configuration with the configurator	4-2
4.3 Standard Setup	4-3
4.3.1 Configurator main menu	4-3
4.3.2 Bus definition	4-4
4.3.3 Station definition	4-5
4.3.4 Communication channels	4-6
4.3.5 Link windows	4-9
4.3.6 Object definition	4-11
4.4 Advanced Setup	4-14
4.4.1 Definition of bus parameters	4-14
4.4.2 Channel definitions	4-19
4.4.3 Object definition	4-26

	Page
<b>5. Programming</b>	<b>5-1</b>
5.1 Description of PCD instructions (SASI, SCON, STXM and SRXM)	5-1
5.1.1 SASI: Assign a channel	5-2
5.1.2 SCON: Open a communication channel	5-7
5.1.3 SRXM: Receive data (objects)	5-8
5.1.4 SRXMI: Receive indirect data (objects)	5-9
5.1.5 STXM: Transmit data (objects)	5-11
5.1.6 STXMI: Transmit indirect data (objects)	5-12
5.2 Important rules for the user program	5-14
5.3 Structure of the user program	5-15
5.3.1 Cold-start program in XOB 16	5-15
5.3.2 Main program in the COB	5-16
5.4 Program examples	5-18
5.4.1 Example 1: 2 PCD stations	5-18
5.4.2 Example 2: 1 PCD station and 1 other station	5-19
5.4.3 Example 3: 2 PCD stations and 2 other stations	5-20
<b>6. Commissioning a PROFIBUS network</b>	<b>6-1</b>
6.1 Checking and testing the hardware installation (physical layer)	6-1
6.2 Checking and testing level 7	6-2
6.2.1 Initiate	6-2
6.2.2 Abort	6-3
6.2.3 Check data exchange	6-3
6.2.4 Optimization of transmission times	6-4
6.2.5 Use of bus monitors	6-4
Appendix A: Object data types	A-1
Appendix B: Effect of bus parameters on transmission times	B-1
Appendix C: PROFIBUS services	C-1
Appendix D: Abbreviations and terminology	D-1



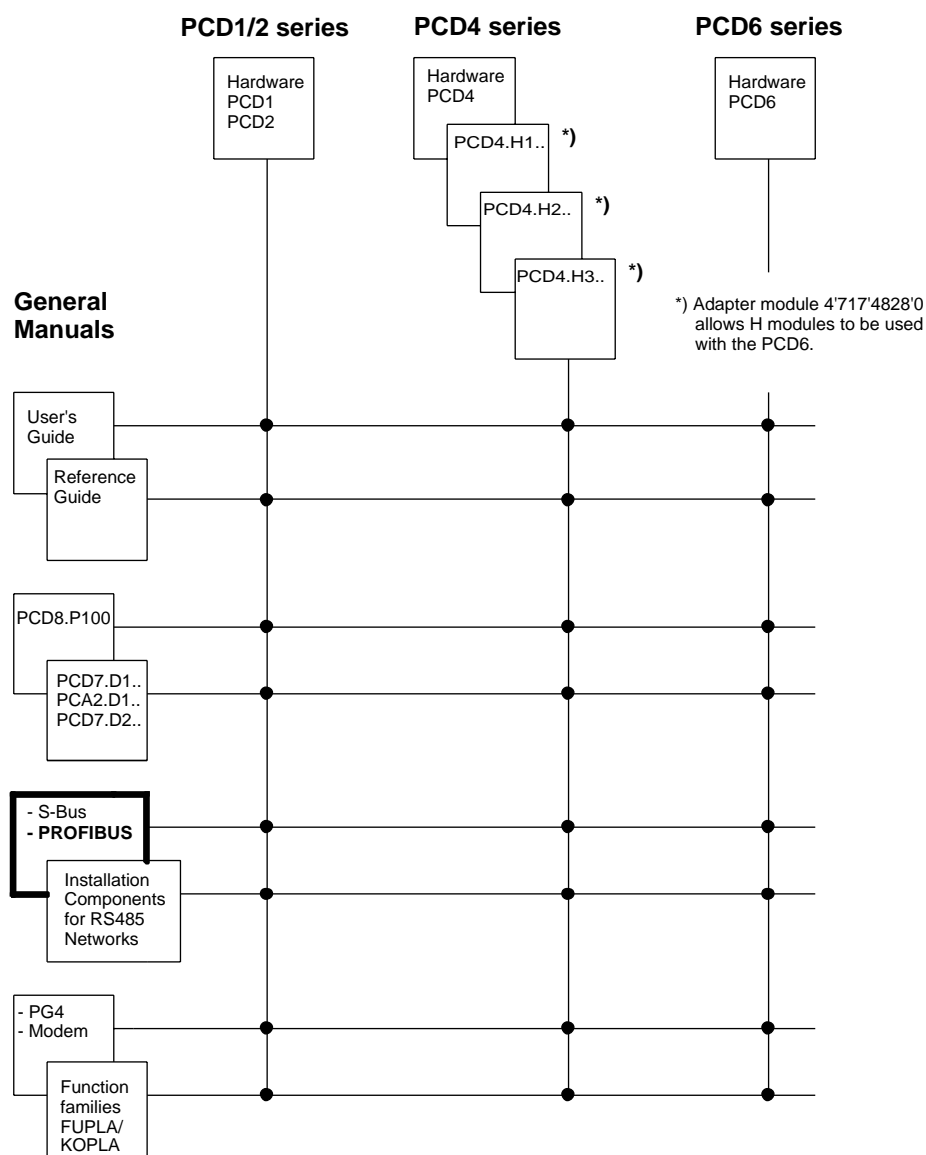
### Please note:

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 AG 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, 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 AG 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<sup>®</sup> PCD will provide you with a modern, safe programmable controller to control, regulate and monitor your installation with reliability for many years.



# 1. Properties and general

---

PROFIBUS (for *PRO*cess *FI*eld *BUS*) is the most successful open Field-bus, which can be used in wide range of applications.

## 1.1 Properties

---

- **Open**  
PROFIBUS ensures that devices of different vendors can communicate together without the need to adapt interfaces. PROFIBUS is standardized as German National standard DIN 19 245 and European standard pr EN 50170.
- **Vendor independent**  
PROFIBUS devices are offered by a wide range of qualified vendors. This enables users to select their favourite supplier and the best products for the job.
- **Optimised for a wide range of applications**  
PROFIBUS consists of an assortment of compatible products. There are three main variations of PROFIBUS corresponding to the intended application:

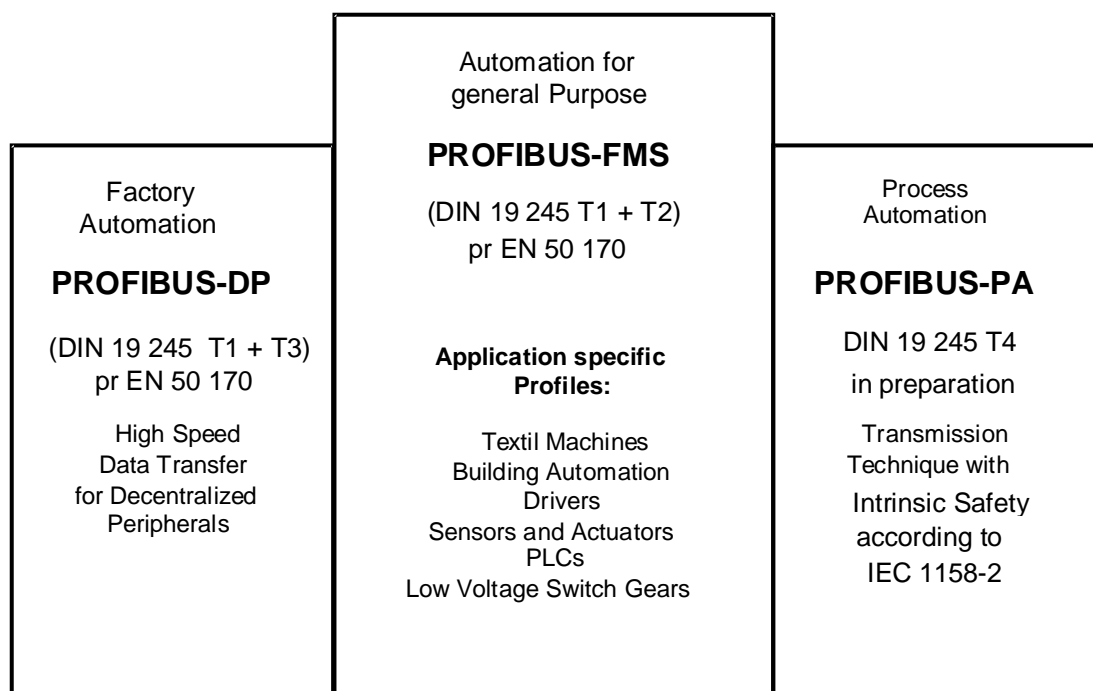


Fig. 1: The PROFIBUS Family

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

This is the universal solution for the communication tasks at the upper level (call level) and the Field Level of the industrial communication hierarchy. In order to carry out the extensive communication tasks with acyclic data transfers at medium speed, the Fieldbus Message Specification (FMS) services offer a wide range of functionality and flexibility.

PROFIBUS-FMS has been available as German National Standard DIN 19 245 Part 1 and 2 since 1990. PROFIBUS-FMS will be included into European Fieldbus Standard pr EN 50170.

**PROFIBUS-DP** (DP = Decentral Periphery)

This is the performance-optimized version of PROFIBUS, specifically dedicated to time-critical communication between automation systems and distributed peripherals. It is suitable as a replacement for costly parallel wiring of 24 V and 4 (0) to 20 mA measurement signals. PROFIBUS-DP is based on DIN 19 245 Part 1 and extends it with effective communication functions for the intended purpose. It has been available as German draft standard DIN 19 245 Part 3 since 1993. PROFIBUS-DP will be included into European Fieldbus Standard pr EN 50170.

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

PROFIBUS-PA is the PROFIBUS version for use in process automation. PROFIBUS-PA utilises the transmission technique specified in IEC 1158-2 and permits intrinsic safety and bus-powered stations. PROFIBUS-PA is based on the results of the Inter-Operable Systems Project (ISP) which has been adopted by the PROFIBUS User Organization. Device profiles define the specific functions. Device Description Language (DDL) and Function Blocks permit the complete interoperability of the devices.

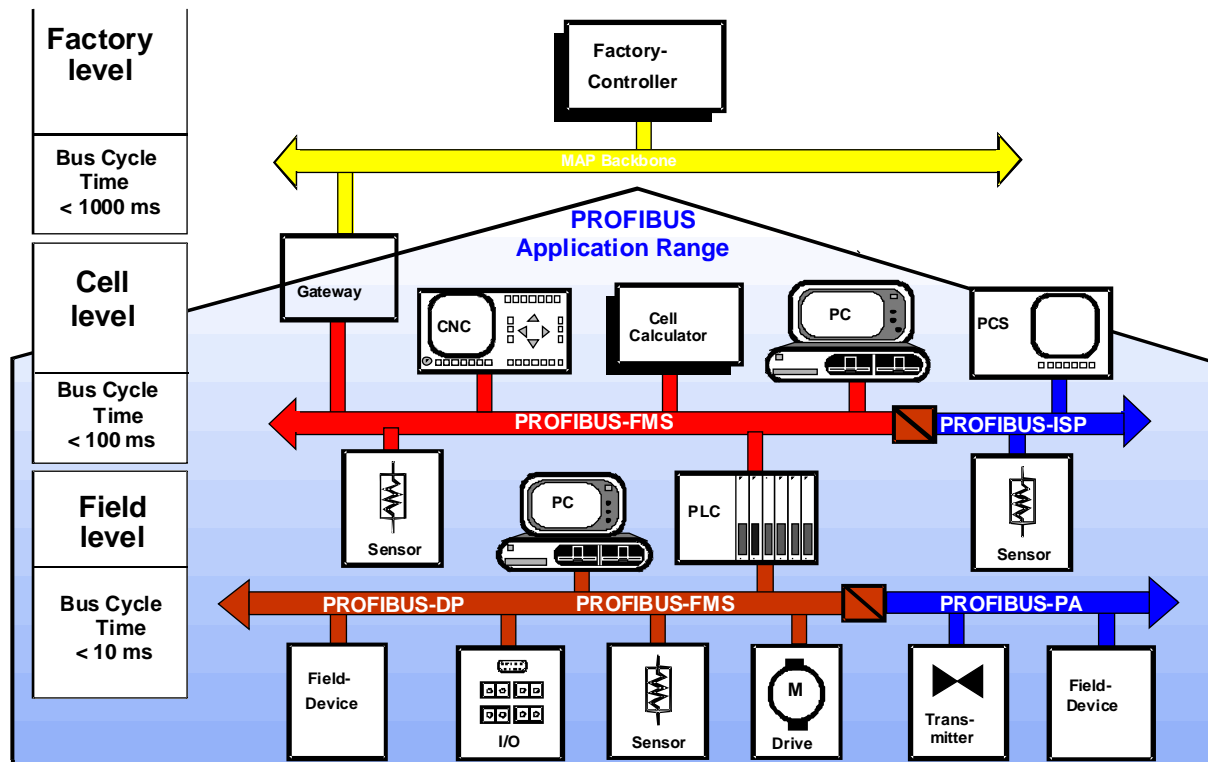


Fig. 2: PROFIBUS Application Range

- Proven**  
 PROFIBUS is an established technology with a large number of applications in building automation, manufacturing, process automation and drive control. Currently, more than 150 vendors have already discovered the rapidly growing market and offer more than 400 PROFIBUS products.
- Certified**  
 Testing of the PROFIBUS devices for conformance and interoperability by accredited test laboratories and the certification by the PNO allows users to be certain that quality and functionality are ensured in multi vendor networks.

## 1.2 Principles

---

To construct a PROFIBUS network and use it to exchange data, you do not necessarily need to read and digest fully the following chapter now. If you plan PROFIBUS communications with SAIA<sup>®</sup> PCD devices only, this is a very simple undertaking and you can proceed directly to Chapter 2.

However, if you would also like to know and understand the theory behind PROFIBUS, you will certainly find the following chapter useful.

### 1.2.1 PROFIBUS Physical Layer (1)

The area of application of a Fieldbus system is substantially affected by the selection of the transmission medium and the physical bus interface. In addition to the requirement for data integrity, the cost of provision and installation of cable are of critical significance. Hence the PROFIBUS standards defines a unique medium access protocol for different transmission techniques:

**Copper wire:** This version is defined as the base version of the transmission technique for applications in manufacturing, building automation and drive control in accordance with the US standard EIA RS-485. It uses a shielded two-conductor twisted pair cable.

**Fibre Optic:** PNO is preparing the specification of a fibre optic transmission technique in order to extend the bus length at high transmission speed and to cater for applications in highly disturbed environments. This specification is available as a proposal within a PNO guide-line.

**Intrinsic Safety:** PROFIBUS-PA permits safe transmission and bus-powered stations. The transmission technique is specified in the international standard IEC 1158-2.

### 1.2.2 PROFIBUS Data Link Layer (2)

The second Layer of the OSI Reference Model provides the functions of the Medium Access Control and data integrity as well as the execution of the transmission protocols and messages. Layer 2 in PROFIBUS is designated as Fieldbus Data Link (FDL). The Medium Access Control (MAC) defines when a station may transmit data. The MAC has to ensure that only one station has the right to transmit data at any time.

The PROFIBUS protocol has taken into account two essential requirements for the Medium Access Control:

- In the case of communication between complex automation components with equal bus access rights (masters), it has to ensure that each of these stations gets sufficient opportunity to execute its communication tasks within the precisely defined time interval.
- In the case of communication between a complex automation device and a simple peripheral device (slave), a cyclic real time data exchange is achieved as simply and as quickly as possible.

The PROFIBUS Medium Access protocol includes the token passing method for communication between complex stations (masters) and the master-slave method for communication between complex stations and simple peripheral devices (slaves). This combined method is called hybrid medium access. (Fig 3)

The **token passing method** ensures, by means of a token, the assignment of the bus access right within a precisely defined time interval. The token is a special telegram that transfers the right for transmission from one master to the next. It is circulated in a (configurable) maximum token rotation time between all masters. PROFIBUS uses the token passing method only between complex stations (masters).

The **master-slave method** allows the master (active station) that currently owns the right for data transmission to communicate with slave devices (passive stations). Each master has the ability to fetch data from the slaves and transmit data to the slaves.

With the hybrid medium access method of PROFIBUS it is possible to implement:

- a pure master-slave system
- a pure master-master system (token passing)
- a system with a combination of both methods.

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

For a certain time, after an active station receives the token, it is allowed to exercise the master function on the bus. It can communicate with all slave stations in a master-slave communication relationship and with all master stations in a master-master communication relationship.

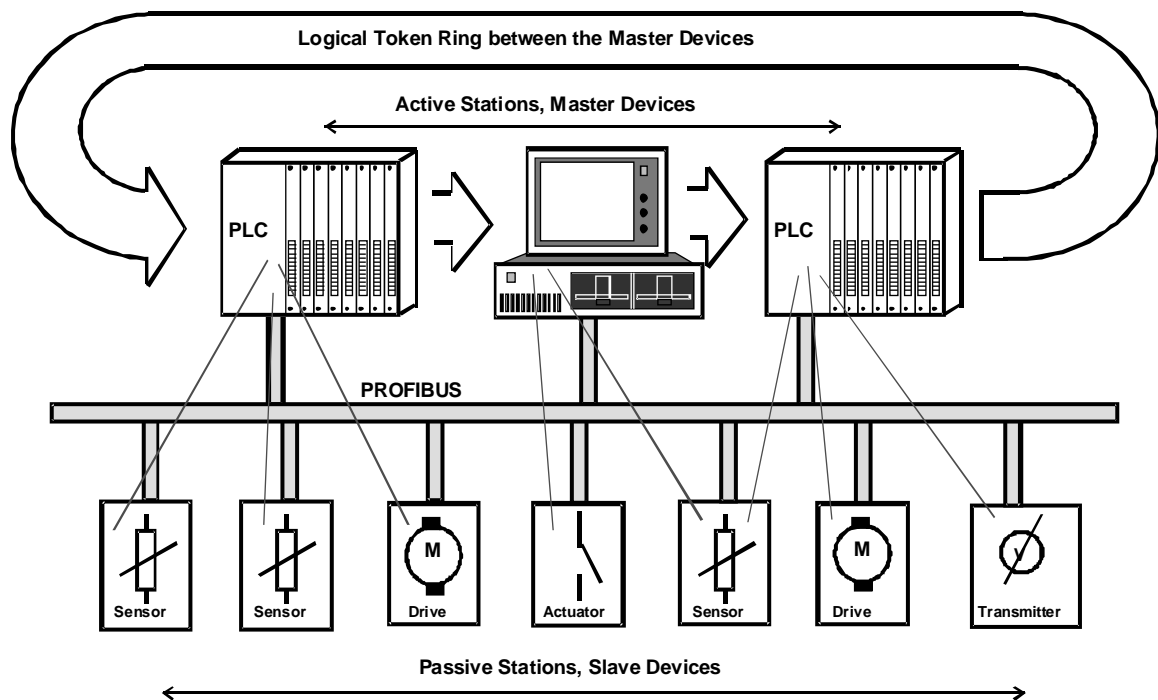


Fig. 3: Hybrid Medium Access Control

A token ring means the organizational chain of active stations building a logical ring with their station addresses. In this ring the token, the medium access right, is circulated from one master to the next in a defined sequence of increasing addresses.

In the start-up phase of the bus system, the task of the medium access control (MAC) of the active stations is to detect this logical assignment and to establish the token ring. In the operational phase, defective or switched-off (active) stations have to be eliminated from the ring and new active stations have to be included in the ring.

Additional characteristics of the PROFIBUS Medium Access Control are: to recognise defects in the transmission medium or the transceivers; to detect errors in the station addressing (e.g. multiple usage); and to detect errors in the token passing (e.g. multiple token or loss of token).

Another important task of Layer 2 is data integrity. PROFIBUS Layer 2 frames ensure high data integrity. All frames have **Hamming Distance HD = 4**. This is achieved by using special start and end delimiters, slip-free synchronisation and a parity bit for each octet, as defined in the international standard IEC 870-5-1.

The PROFIBUS Layer 2 operates in a connectionless mode. It offers both peer-to-peer transmission and multi-peer communication (broadcast and multi-cast ).

An individual subset of the Layer 2 services is used in PROFIBUS-FMS and PROFIBUS-DP. All Layer 2 services are called through Service Access Points (SAPs) from the upper Layers. In PROFIBUS-FMS these SAPs are used for the addressing of the logical communication relationships. In PROFIBUS-DP, precisely defined functions are assigned to individual SAPs. In the active and passive stations, multiple SAPs are allowed simultaneously. There is a distinction between source- (SSAP) and destination- (DSAP) service access points.

Table: Services of the PROFIBUS Data Link Layer (2).

Service	Function	Used by
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

### 1.2.3 PROFIBUS-FMS

PROFIBUS-FMS is designed for the communication between intelligent field devices and controllers and for information exchange between controllers. Hence, functionality is more important than system reaction time. Normally, information exchange is acyclic at the request of the application process.

#### • Application Layer (7)

Layer 7 of the ISO/OSI Reference Model provides the application services to the user. These services make efficient and open data transfer possible between application processes. The PROFIBUS-FMS Application Layer is specified in DIN 19 245 Part 2 and consists of:

- Fieldbus Message Specification (FMS) and
- Lower Layer Interface (LLI).

FMS describes the communication objects, the application services and the resulting models from the point of view of the communication partner.

LLI serves to adapt the application functions to the various features of Layer 2.

- **PROFIBUS Communication Model**

An application process includes all programs, resources and tasks that are not assigned to a communication Layer. The PROFIBUS communication model permits the unification of distributed application processes to a common process, using communication relationships.

That part of an application process in a field device which is communicable is called a Virtual Field Device (VFD). All objects of a real device that can be communicated with (variables, programs, data ranges) are called communication objects. They are part of the VFD. (Fig. 6).

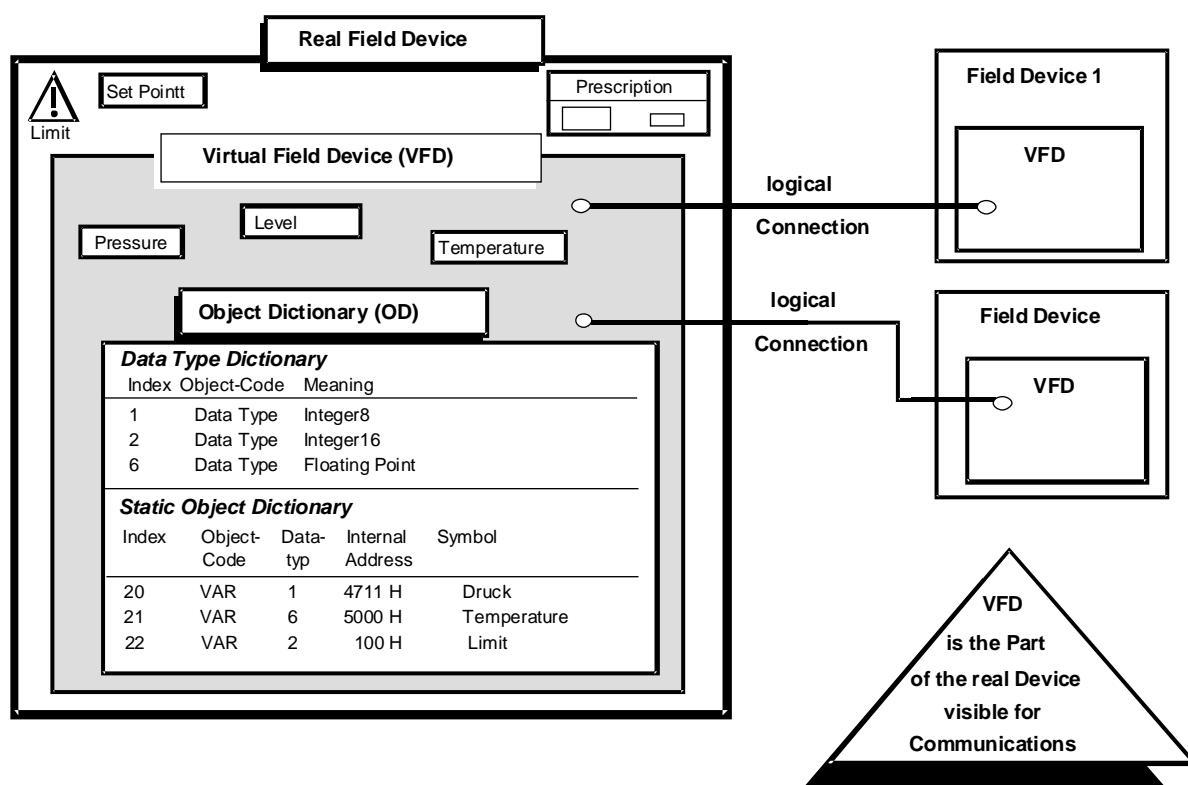


Fig 4: Virtual Field Device (VFD) with Object Dictionary

In the PROFIBUS communication model, mapping of the functions of the VFD onto the real device is provided by the Application Layer Interface (ALI).

Fig. 4 shows the relationship between the real field device and the VFD. In this example, only the variables pressure, fill level and temperature, which are part of the VFD, may be read or written via the two communication relationships.



- **Communication Objects and Object Dictionary (OD)**

All communication objects of a PROFIBUS station are entered into its local object dictionary (OD). The OD may be pre-defined at simple devices. At complex devices, the OD is configured and locally or remotely loaded into the device. The OD contains description, structure and data type. It also contains the internal addresses of the device's communication objects and its reference at the bus (index/name). The OD consists of:

- Header (contains information about the structure of the OD).
- List of Static Types (contains the list of the supported static data types).
- Static Object Dictionary (contains all static communication objects).
- Dynamic List of Variable Lists (contains the actual list of all known variable lists).
- Dynamic List of Program Invocations (contains the List of all known programs).

The individual parts of the OD are implemented only if the device supports the corresponding functions.

Static communication objects must be defined in the static Object Dictionary. They may be pre-defined by the manufacturer of the device, or defined during the configuration of the bus system. Static communication objects are mainly used for communication in the field area. PROFIBUS distinguishes the following static communication objects:

- Simple variable
- Array (row of simple variables of the same type)
- Record (sequence of simple variables with different types)
- Domain (a data range)
- Event

Dynamic communication objects are entered into the dynamic part of the OD. They may be pre-defined at configuration time, or dynamically defined, deleted or changed with the application services in the operational phase. PROFIBUS supports the following dynamic communication objects:

- Program Invocation
- Variable List (sequence of simple variables, arrays or records)

Logical addressing is the preferred method for addressing of communication objects. They are accessed with a short address, called an Index. The index is a number of type Unsigned 16. This makes efficient messages possible and minimises the protocol overhead. For each communication object an index is defined in the OD. All PROFIBUS devices must support the logical addressing method.

In addition PROFIBUS-FMS permits other optional addressing methods for special applications:

**Addressing by name.** (The symbolic name of the communication objects is transferred via the bus.)

**Physical addressing.** (Any physical memory location in the field devices may be accessed with the services PhysRead and PhysWrite.)

To prevent unauthorized access, an access protection may be optionally defined for each communication object. Access to an object or a certain group of devices is only possible with a certain password. Password and device group may be defined individually for each object in the OD. Additionally, by definition in the OD, it is possible to restrict the services which may access an object. (e.g. only read-access permitted).

- **FMS Services**

From the point of view of an application process, the communication system is a service provider offering various application services, the FMS-Services. The execution of the services is described by service primitives. A service primitive represents an interaction between the requester and the responder.

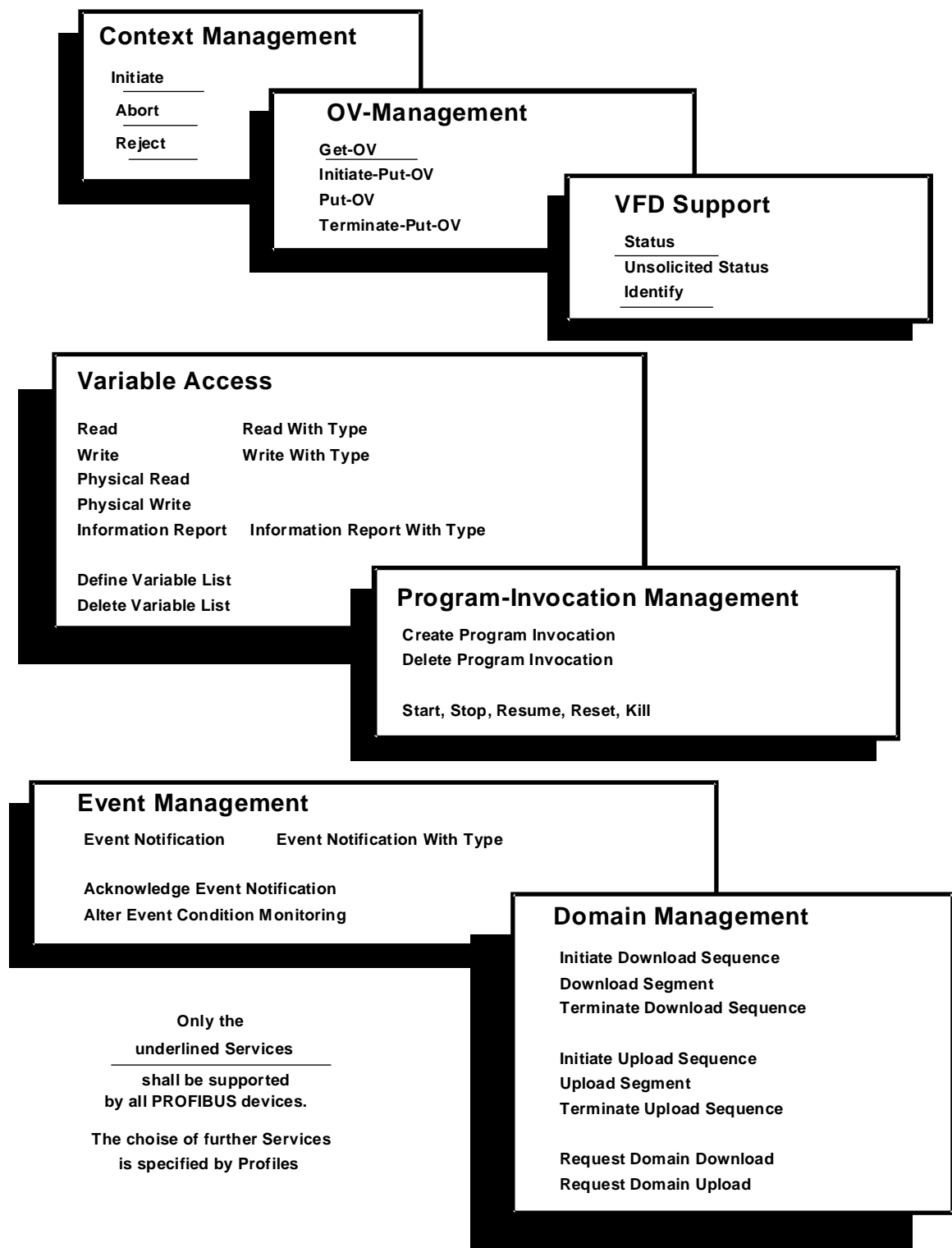


Fig. 5: Summary of the PROFIBUS Application Services

**Confirmed services** are only permitted on connection oriented communication relationships. The sequence of a confirmed service execution is shown in Fig. 5.

**Unconfirmed services** may also be used on connectionless communication relationships (broadcast, multicast). They may be transferred with high or low priority. An unconfirmed service is invoked with a "request service primitive". After the transmission over the bus, an "indication service primitive" is issued to the user(s) at the receiver(s). The "confirmation service primitive" is issued to the user(s) at the receiver(s). The "confirmation/response service primitives" are omitted for unconfirmed services.

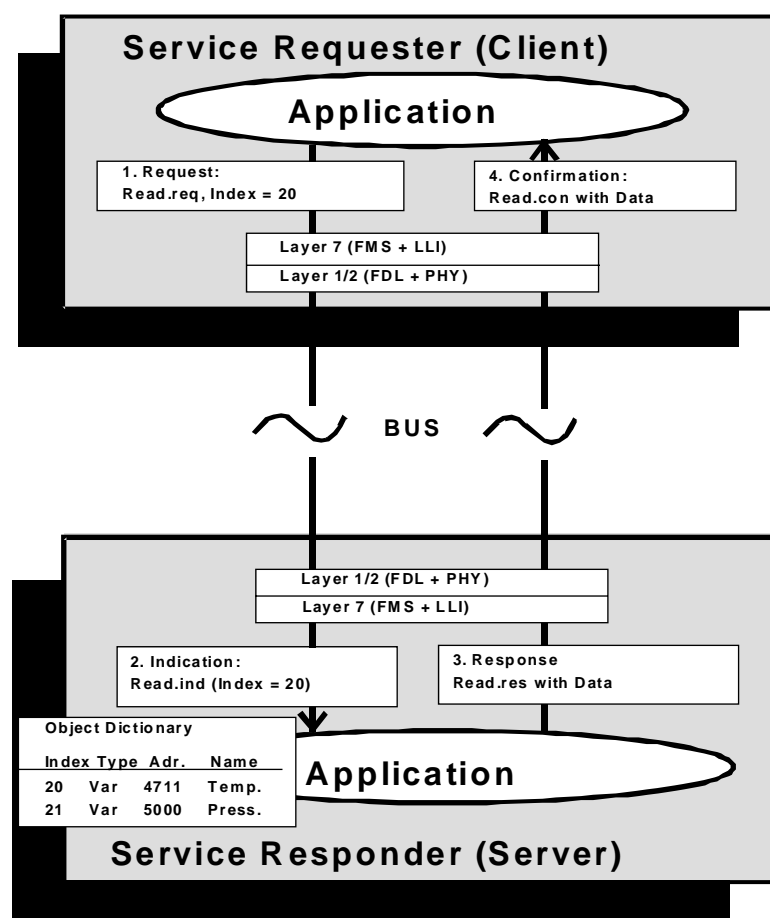


Fig. 6: Execution of a confirmed Service

The FMS services are divided into the following groups:

- **Context Management Services** allow establishment and release of logical connections, as well as the rejection of inadmissible services.

- **Variable Access Services** permit access to simple variables, records, arrays and variable lists.
- **Domain Management Services** enable the transmission of large memory areas. The application splits the data into segments for transmission.
- **Program Invocation Management Services** allow the control of program execution.
- **Event Management Services** make the transmission of alarm messages possible. These unconfirmed services may be used with high or low priority. The event messages may also be transmitted on broadcast- or multicast communication relationships.
- **VFD Support Services** permit device identification and status report. Unsolicited Status reports may also be transmitted at the initiative of the involved device on broadcast or multicast communication relationships.
- **OD Management Services** permit Object Dictionaries to be read or written.

The large number of possible PROFIBUS-FMS application services serves to fulfil the various requirements of field devices on the communication. Only a few of the application services are mandatory in implementation. The selection of further services depends on the specific application and is specified in so called "Profiles".

- **Lower Layer Interface (LLI)**

The mapping of Layer 7 onto the services of Layer 2 is effected by the LLI. It conducts the data flow control, connection monitoring and the mapping of the FMS services onto Layer 2 with consideration to the various types of device (master/slave).

The user communicates with the other application processes over logical channels, which are the **communication relationships**. For the execution of the FMS and FMA7 services, the LLI provides various types of communication relationships. They differ relative to their connection qualities (monitoring mechanism, data transfer possibilities and requirements on the remote station). The selection is done according to the requirements of the application processes. Fig. 7 shows a summary of the various communication relationships.

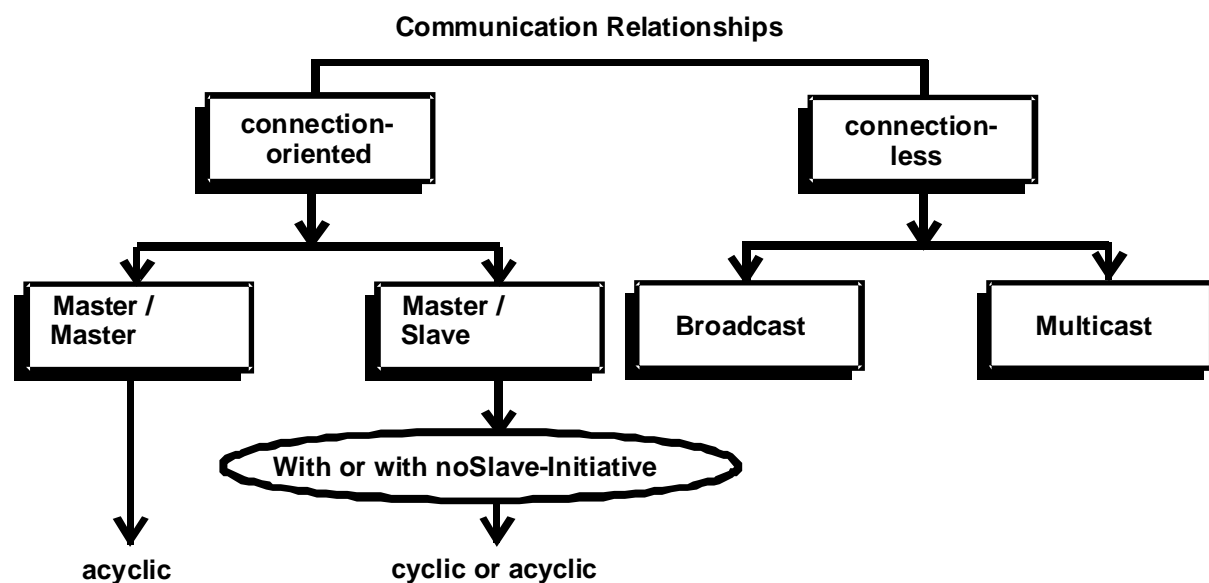


Fig. 7: Summary of the supported Communication Relationships

**Connection-oriented communication** relationships represent a logical peer-to-peer connection between two application processes. Before the connection can be used for data transmission it has to be established with an "Initiate Service". After successful establishment, the connection is protected against third party access and can be used for data transmission with the various FMS services. If an established connection is no longer needed for data communication it may be released with the "Abort Service". The LLI permits timer-controlled connection monitoring for connection-oriented communication relationships.

**Connectionless communication relationships** allow simultaneous communication between a device with several stations. Only unconfirmed services may be used.

With **broadcast** communication relationships an unconfirmed service is transmitted simultaneously to all other stations. With **multicast** communication relationships an unconfirmed service is transmitted simultaneously to a group of stations (master- or slave devices). Typical applications for these functions are process synchronisation or transmission of global alarms.

- **Connection attributes**

The connection attribute is another characteristic feature of the connection-oriented communication relationships. PROFIBUS-FMS distinguishes between open and defined connections. For **defined connections** the communication partner is fixed at configuration time and may not be changed in any phase of the connection. This prevents unauthorized access.

At open connections the communication partner is dynamically defined in the connection establishing phase. Open connections offer special advantages for simplest PROFIBUS-FMS devices for which all communication relationships are fixed by the vendor. If these pre-defined connections are defined as "open connections", the device can communicate with all other stations without additional configuration.

The slave is permitted to send unconfirmed FMS services to the assigned master only if a master-slave communication relationship is marked with the attribute "slave initiative". The LLI allows this functionality without providing the Layer 2 bus access right to the slave. It may initiate the unconfirmed services with high or low priority. Typical applications are simple devices that wish to transmit an unsolicited alarm message to the process master.

- **Cyclic and Acyclic Data Transfer**

PROFIBUS-FMS permits cyclic and acyclic data transfer.

**Cyclic data transfer** means that exactly one variable is permanently read or written over a connection. This is permitted for the services Read and Write. The LLI offers an efficient cyclic transmission mode which reduces the transmission time in comparison with acyclic data transfer.

**Acyclic data transfer** means that an application sporadically accesses various communication objects over a connection.

- **Communication Relationship List (CRL)**

All communication relationships of a PROFIBUS-FMS device must be entered into the CRL. The CRL contains their description. For simple stations it is predefined by the vendor. Otherwise the CRL is configured and loaded with the network management services, either locally or remotely via PROFIBUS.

An application addresses a communication relationship with a local short reference, called a communication reference (CREF).

From the point of view of PROFIBUS, a communication relationship is addressed with the station address, the Layer 2 Service Access Point and LLI Service Access Point.

The CRL describes the assignment between the local communication references and the addressing in Layer 2 and the LLI. Furthermore the communication context (selection of the services, message lengths ... ) and the selected connection monitoring mechanism are defined specifically for each CREF.

- **PROFIBUS-FMS-Profiles**

A wide range of functions is specified in the PROFIBUS-FMS standard in order to make the bus system universally applicable. For various application fields it is necessary to adapt the functions to those actually required. These adaptations are specified by application specific definitions, called "Profiles". Profiles ensure that PROFIBUS devices from different manufacturers, which fulfil the same profile, have the same communication functionality.

Profiles contain more information than the simple selection of communication parameters. They additionally include application specific definitions for the meaning of the communication functions as well as the interpretation of status and error indications. Profiles are defined in User Groups of the PNO.

Profiles for PROFIBUS-FMS and their current status:

Profile	Status
Building automation	PNO Guide-line
Drivers	PNO Guide-line
Sensors and Actuators	PNO Guide-line
Programmable Logic Controller	Proposal of a PNO Guide-line
Textile machines	In preparation
Low Voltage switch gears	Proposal of a PNO Guide-line

Profiles are needed for master and slave devices. A master should be able to understand the definitions of all corresponding profiles of its application range. A large range of functions is more important than the cost of the master.

A profile for a slave defines only the essential functional subset. This allows a significant simplification of the protocol implementation for the vendor and a cost reduction for the user.



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

---

### 2.1 The processor module PCD4.M445

---



Fig 8: Front panel of the PCD4.M445

This processor module has 2 independent processors (genuine multiprocessor system). In addition to the programming interface which is always there, it also has 3 additional serial interfaces which are also independent of each other. The module has further PROFIBUS coprocessor, connected via a 9-pole D-type connector on the front panel.

Further information for this module (e.g. pin assignment of the PROFIBUS connector) can be obtained from the manual "PCD4 Series - Hardware Manual", order number 26/734 E.

## 2.2 PROFIBUS-FMS connection, type PCD7.F700 for PCD2.M120

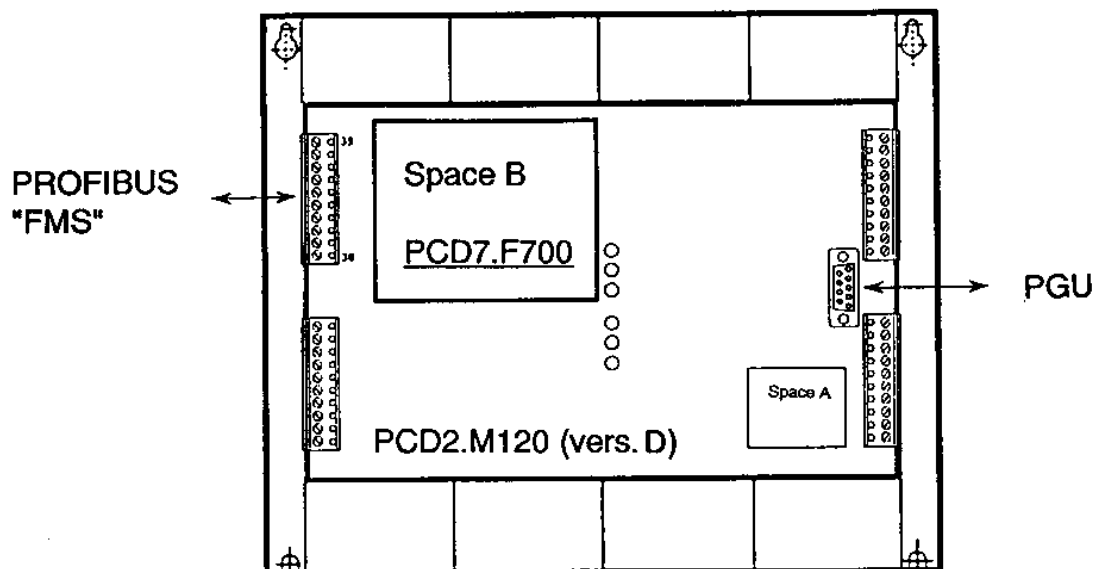


Fig: PCD2 with PCD7.F700

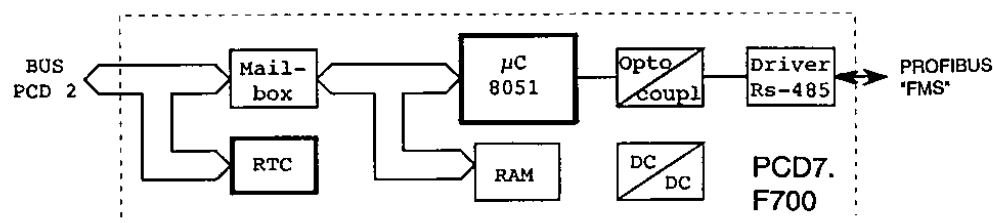


Fig 9: Block diagram of the PCD7.F700

When the PROFIBUS module PCD7.F700 is used, the PCD2 series has access to the PROFIBUS-FMS. The PCD2 is therefore enabled to be used as FMS master or FMS slave.

The PCD7.F700 plugs into connector B on the PCD2.M120 base unit (version D or more recent). PROFIBUS connection is via the row of plug-in terminals numbered 30 to 39. The module contains also the real-time clock (RTC). Further information for this can be obtained from the manual "PCD1/PCD2 Series - Hardware Manual", order number 26/737 E.

## 2.3 Data for the PROFIBUS connection

What	PCD4.M445	PCD2.M120 with PCD7.F700
FMS protocol according to DIN 19245 Parts 1 and 2	X	X
Connection as FMS master or FMS slave	X	X
Profile: Controller class 2 (enlarged)	X	X
Transmission speeds: 9.6, 19.2, 38.4 or 500 kbps	X	X
93.75 and 187.5 kbps	on request	X
PROFIBUS interface RS 485 electrically isolated	X	X
Up to 127 addressable stations (divided by PCD7.T100 repeaters into segments of 32 stations each)	X	X
Connections established simultaneously	10..99	10..19
Number of communication objects	100..499	100..199
FMS services		
Initiate	X	X
Abort	X	X
Reject	X	X
Identify (as Server)	X	X
Status	X	X
Get-OD (as Server)	X	X
Read/Write	X	X
Data types		
Boolean	X	X
Integer 8/16/32	X	X
Unsigned 8/16/32	X	X
Octet string	X	X
Bit string	X	X
Floating point	X	X

## 2.4 Repeater PCD7.T100

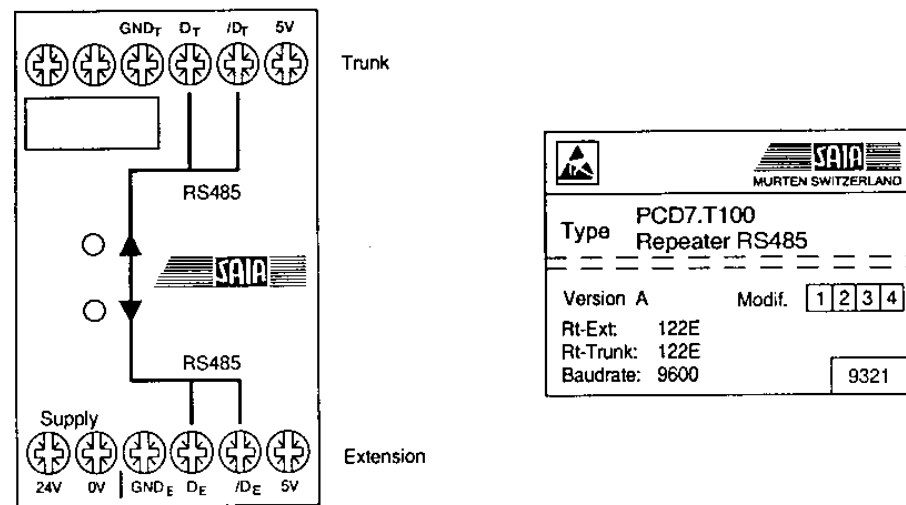


Fig. 10: Front panel and name plate

The repeater is used for decoupling the RS 485 communications network. This serves two purposes: the reprocessing of transmission signals and the electrical isolation of individual sections of the line.

Signal reprocessing is necessary if:

- the total length of bus cable exceeds the admitted total length
- more than 32 remote stations participate in a network

Electrical isolation is necessary if:

- the reference ground in the area of the installation shows excessive potential differences (a few volts)
- the installation's reference ground is excessively charged with stray current.

The maximum number of repeaters which can be used in sequence with PROFIBUS is 3.

Further information can be obtained from the manual "Installation Components for RS 485 Networks" order number 26/740 E.

## 2.5 Termination-Box PCD7.T160

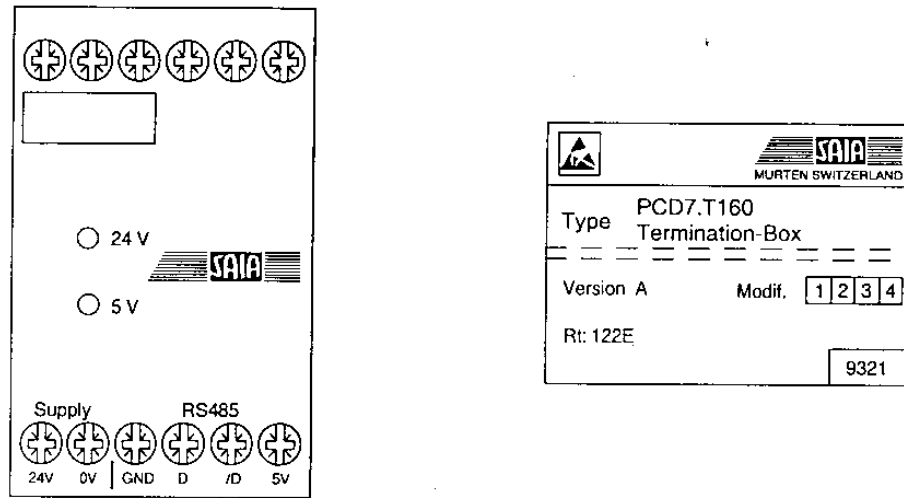


Fig 11: Front panel and name plate

For the save use of an RS 485 network, and for ease of service, it is important to install the line termination resistors separately and to arrange them clearly. The termination box is provided for this purpose where, depending on the physical properties and length of the line, a suitable line termination resistance can be set by a DIL switch. An electrically isolated supply provides bias voltage at the necessary off-load potential for signal lines D and /D.

Further information can be obtained from the manual "Installation Components for RS 485 Networks" order number 26/740 E.

**Notes :**

## 3. Planning and realizing a PROFIBUS network

---

The realization of a PROFIBUS network can be divided into the following steps:

- a) Planning of the network
- b) Definition and configuration of the network with a configurator
- c) Creation of the user program for the data exchange
- d) Commissioning of the network

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 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 Physical layer

---

- Creation of an overview diagram
  - Definition of the max. network distance
  - Definition of the max. segment distance
  - Definition of the cable type
  - Definition of the max. baud rate
  - Eventual provide for a repeater
  - Definition of the beginning and end station
- to provide for PCD7.T160 termination boxes
- For a correct installation of the Profibus, the instructions in the manual "Installation components for RS 485 networks" should be observed.

## 3.2 Logical network structure

---

### 3.2.1 Definition of the bus parameters

- Baud rate
- Highest station number
- GAP update factor
- Target rotation time
- etc.

<b>Important:</b> The bus parameters must be identical for every bus station.
---

### 3.2.2 Definition of the communication relations

Now the logical communication channels have to be defined as the physical structure has been described.

- Type of device and its function (Master/Slave)
- The stations' addresses
- Connection channel
- Connection type (Cyclic/acyclic)
- Service Access Points (SAP)
- Access rights
- etc.

### 3.2.3 Creating of the object dictionary

It is planned, which data respectively process objects have to be exchanged over the bus.

For every single station the planer creates a source object dictionary. The dictionary contains the communication objects which puts a station at disposal of others as also a complete or partly copy of the partner station's objects.

During the operation every station can accede to another station's objects through a defined communication channel.

### 3.2.4 Checking respectively valuation of the required performance values

- Determination of the response time
- Determination of the network loading



## 4. The PROFIBUS configurator

---

The definition and configuration (bus parameters, communication relationship list and object dictionary) of a PROFIBUS network can be very extensive, depending on the size of the project. This task is made significantly easier for the user by the PROFIBUS configurator.

The configurator can be used to configure all SAIA devices.

### 4.1 General

---

The configurator consists of software which runs under MS Windows 3.1 and higher. No special hardware is required. Windows technology is used throughout. This provides a good overview and makes parameter entry very user friendly.

To enable less skilful users to configure simply, only the main parameters appear in the individual windows. To reach the totality of all parameters, it is possible in most windows to switch over to an “advanced setup”.

Standard values are offered wherever the setting of parameters allows this. These standard values can, of course, also be adjusted.

A value range is specified for all parameters.

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

As documentation, a clear list is generated with all parameters. It shows the precise configuration of the station in the PROFIBUS network. It is also possible to derive a list of all communications channels and the defined objects from this list.

## **4.2 Network configuration with the configurator**

---

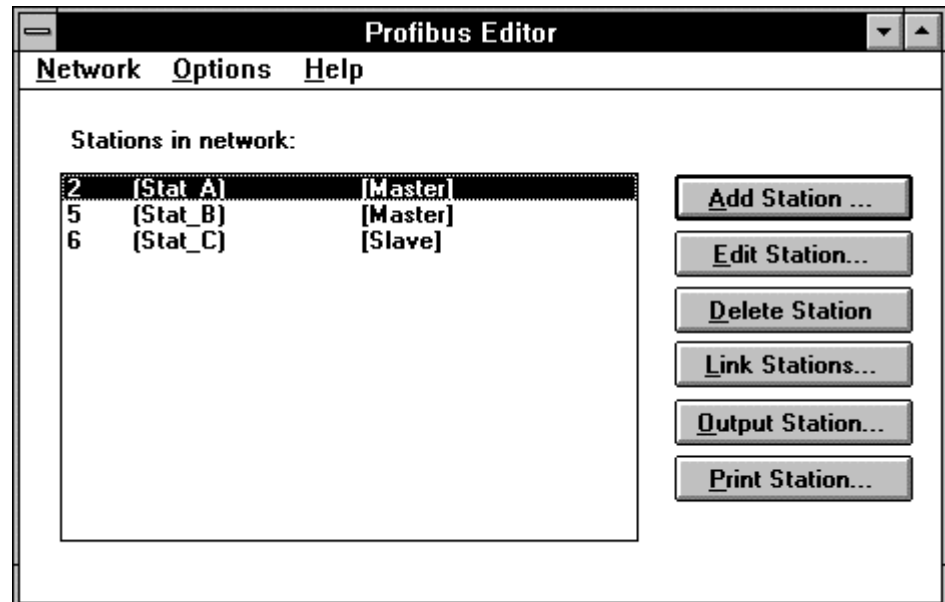
The configuration procedure can be divided into the following steps:

1. Definition of bus parameters
2. Definition of stations
3. Definition of communications channels
4. Creation of communications objects (link stations)
5. Creation of communications objects
6. Generation of SASI texts for PCD stations (output station)
7. Saving the configuration
8. Creation of documentation

## 4.3 Standard setup

### 4.3.1 Configurator main menu

This is the basic window of the configurator, which lists all stations exchanging data with each other.



**Network:** In this menu new or existing configurations are opened or saved. The “Bus Definition ...” window is also reached from here.

**Options:** Access to change the offset of the text base address for configuration texts (standard value: 3000).

**Help:** Help file for user (not implemented).

**Add station...:** Access to “Station Definition” window. Used to add a new station to the network.

**Edit station...:** Access to “Station Definition” window. Used to edit a previously defined station.

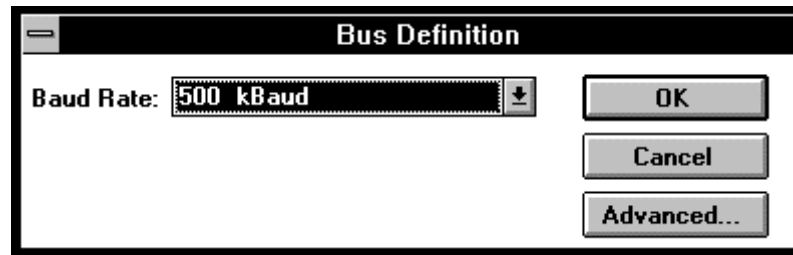
**Delete station:** Removal of a station from the network.

**Link station...:** Access to the “Link Stations” function.

**Output station...:** Creates the configuration file for the selected station.

**Print station...:** Prints the configuration parameters of the selected station.

### 4.3.2 Bus definition



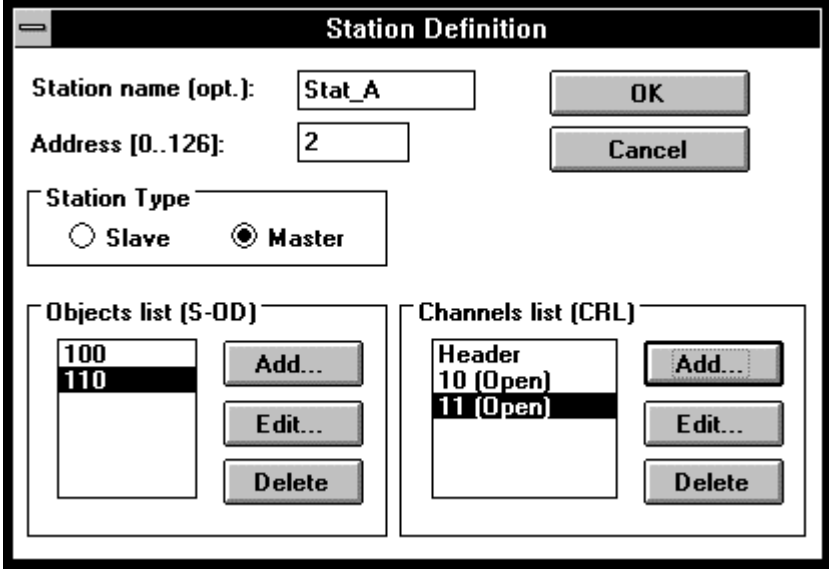
**Baud rate:** A choice can be made from the following baud rates:

9.6, 19.2, 38.4, 93.75 (PCD7.F700 only),  
187.5 (PCD7.F700 only) and 500 kbps

Bus parameters are identical for all stations in the network.

**Advanced...:** Access to advanced setup.

### 4.3.3 Station definition



The image shows a 'Station Definition' dialog box. It has a title bar with a minus sign. Inside, there are two text input fields: 'Station name (opt.):' with the value 'Stat\_A' and 'Address [0..126]:' with the value '2'. To the right of these fields are 'OK' and 'Cancel' buttons. Below the address field is a 'Station Type' section with two radio buttons: 'Slave' and 'Master', with 'Master' selected. At the bottom, there are two list boxes. The left one is titled 'Objects list (S-OD)' and contains '100' and '110'. The right one is titled 'Channels list (CRL)' and contains 'Header', '10 (Open)', and '11 (Open)'. To the right of each list box are three buttons: 'Add...', 'Edit...', and 'Delete'.

**Station name:** Name of station

**Address:** Address of station in network

**Station Type:** Select either master or slave.

**Object list:** Lists the communication objects of the station

S-OD = Static Object Dictionary  
(S-OV = Statisches Objekt-Verzeichnis)

**Channel list:** Lists the virtual channels of the station

CRL = Communication Relationship List  
(KBL = Kommunikations-Beziehungsliste)

#### 4.3.4 Communication channels

In PROFIBUS, stations wishing to exchange data with each other must have a mutual communication relationship. Each station therefore has its own **communication relationship list (CRL)** setting out all defined communication relationships. The communication relationship list is structured as a table with one line containing the relevant data for each projected communication relationship.

The connections listed are accessed via the line index, the so-called **communication reference (CREF)**. In our configurator, the **channel number** represents the communication reference (CREF). The communication channel is a virtual transmission path across which information can be exchanged.

A channel contains the following information:

- Address of partner station
- **Service Access Points** used (LSAP and RSAP)
- Connection type (cyclic, acyclic, etc.)
- List of permitted services (e.g. Read/Write, etc.)
- The size of data buffer required (PDU length)

The following rules apply for the communication channel:

- A single channel can only be used for one defined connection.
- If a connection is open, a channel can be used for connection with several partner devices (often utilized for non-configurable slaves).
- A single station can have up to 90 (10 for PCD2 with PCD7.F700) channels defined and running in parallel.

### Service Access Point (SAP)

The SAP forms the logical interface between the application layer (FMS layer) and the FDL layer (Field Data Link).

The following are differentiated:

- LSAP (Local Service Access Point) = SAP of the station itself
- RSAP (Remote Service Access Point) = SAP of partner station

The following rules apply:

- Poll SAP = common SAP for all slave stations (default poll SAP = 128)
- If a master-master connection is defined, the same SAP can only be used once for a single connection (channel).
- For an open connection, the same rule applies by analogy as for the communication channel.

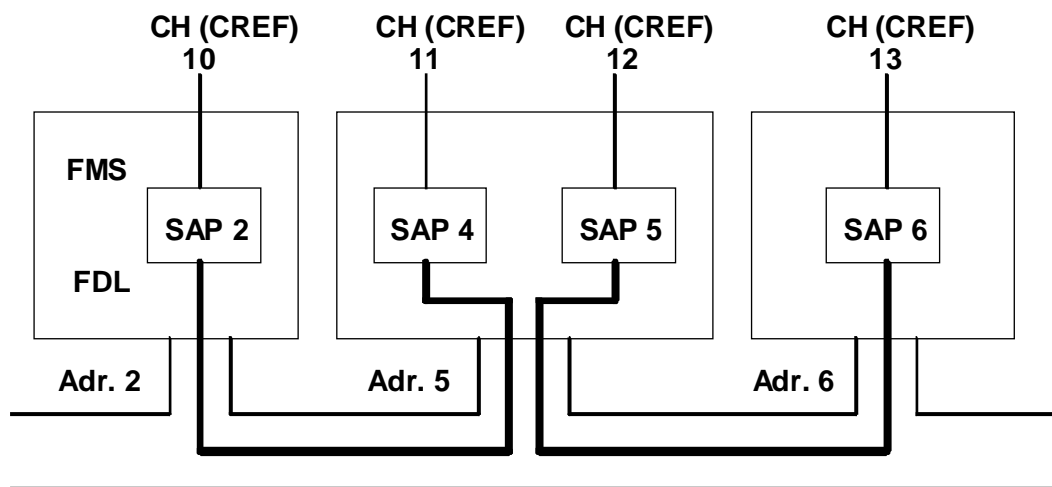
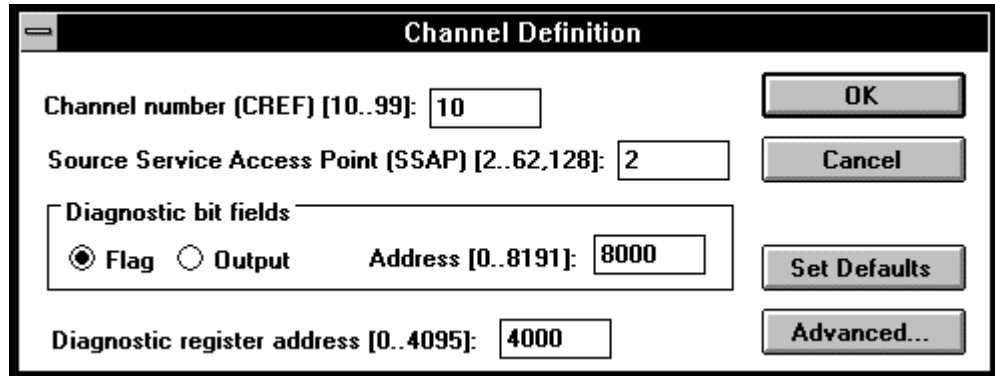


Figure 12: Examples of communication relationships with channel and SAP numbers.

### Channel definition

In this window, parameters are set for the communication channels of a station.



The image shows a 'Channel Definition' dialog box with the following fields and buttons:

- Channel number (CREF) [10..99]:** A text box containing the value '10'.
- Source Service Access Point (SSAP) [2..62,128]:** A text box containing the value '2'.
- Diagnostic bit fields:** A group box containing:
  - ☒ **Flag** ☐ **Output**
  - Address [0..8191]:** A text box containing the value '8000'.
- Diagnostic register address [0..4095]:** A text box containing the value '4000'.
- Buttons:** 'OK', 'Cancel', 'Set Defaults', and 'Advanced...'.

**Channel number:**

This parameter is used for definition of the SASI text.

**Source Service Access Point (SSAP):**

Service access point of the station

**Diagnostic bit field:**

This corresponds to the diagnostic elements <dia\_elem> in the SASI text for general communications: base address of 8 flags (or outputs).

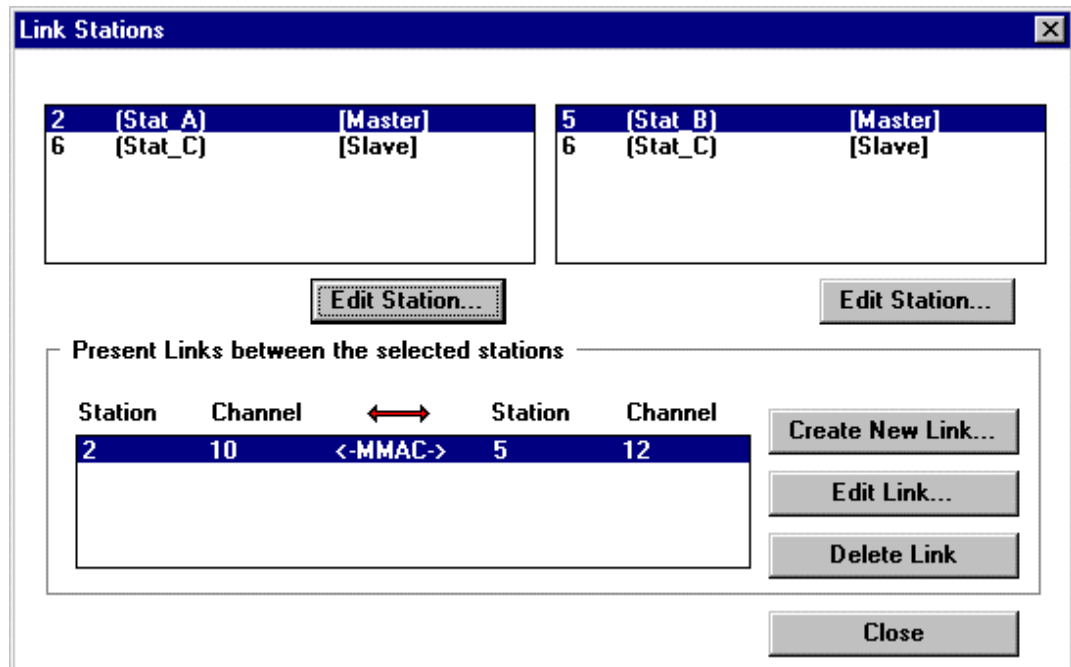
**Diagnostic register address:**

This corresponds to the diagnostic register <dia\_reg> in the SASI text for general communications: address of a PCD register.



### 4.3.5 Link windows

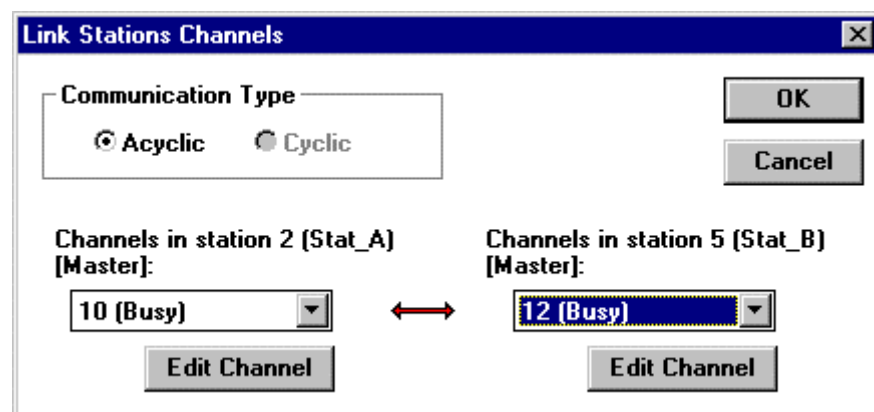
The virtual communication link (communication relationship) must now be defined. Two windows are provided for this: “Link Stations” and “Link Station Channels”.



Each of the top two boxes of the “Link Stations” window contains a station which has been selected for linking. The lower box shows the existing links between these two stations. Links can now be cancelled, modified or newly defined. For the communication relationships listed, the type of link is indicated in the middle.

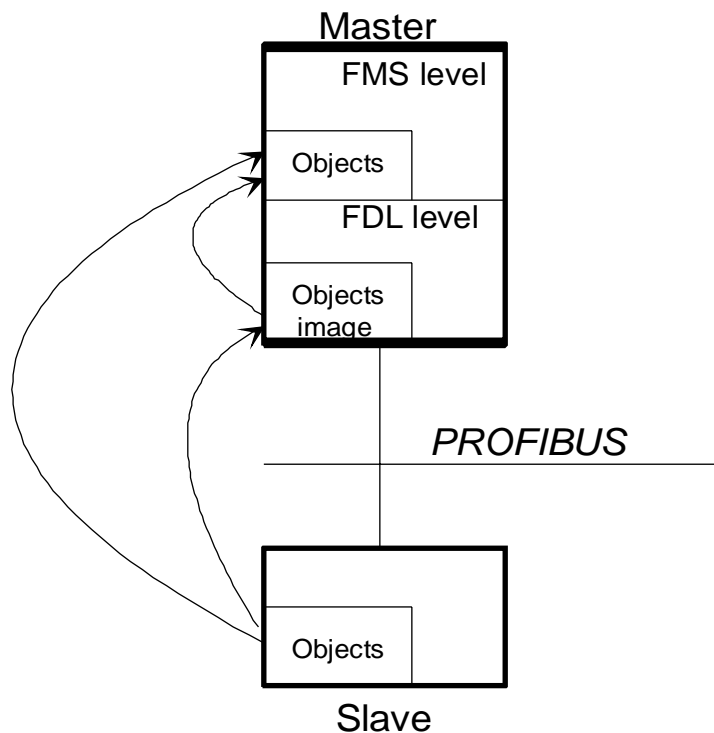
Example above: MMAC = Master-Master-Acyclic

The ‘Create New Link’ and ‘Edit Link’ options lead to the “Link Station Channels” window.



In the previous window (Link Stations) the link was made at station level. Here the link occurs at channel level.

- **Communication Type**



**Acyclic** communications means that the data is transferred spontaneously to the remote device, during processing of an SRXM/STXM instruction.

**Cyclic** communications are mainly used only for connections with simple slave devices (e.g. sensors). As soon as the first STXM/SRXM instruction has been processed, the specified objects from the FDL layer are exchanged cyclically. A picture of the objects is set up in the memory of the FDL layer. For the user there is no direct, discernible difference between cyclic and acyclic communications.

The advantage of cyclic data transfer is its fast accessing of objects for real-time applications using the specially provided features.

However, cyclic data transfer puts a considerable load on bus communications. For this reason cyclic data transfer should only be used for signals where time is critical.

#### 4.3.6 Object definition

PROFIBUS provides for object-oriented programming, so that even the most diverse devices can exchange data with each other.

Therefore, before data can be exchanged, the communication objects and their meaning or structure must be made known to the system. The definition of objects takes place in the configurator for all stations on the bus.

Each station keeps a complete or partial copy of the objects in its partner station.

Example:

Station A is to transfer flags 1000-1007 to outputs 0-7 of station B. Two objects must be defined:

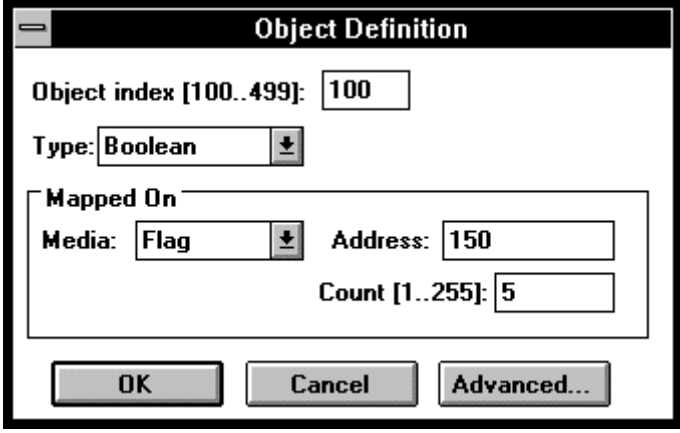
1. Object xx for station A:  
Packet of 8 boolean type elements which are defined as flags 1000-1007.
2. Object yy for station B:  
Packet of 8 boolean type elements which are defined as outputs 0-7.

It is now possible for station A to send the following telegram:  
Write (send) my object xx to object yy of station B.

The following rules apply:

- The objects of a station are in principle (access rights can be defined) accessible to all stations on the bus via the communication channels.
- When objects are transmitted, the source and destination objects must have compatible data structures and formats.
- It should be ensured that object size (number of bytes) does not exceed the defined buffer size (PDU length) of the channel used.

- **Object format for data types:**  
Boolean, integer 8/16/32, unsigned 8/16/32 and floating point



The dialog box is titled "Object Definition". It contains the following fields and controls:

- Object index [100..499]:** A text box containing the value "100".
- Type:** A dropdown menu showing "Boolean".
- Mapped On:** A section containing:
  - Media:** A dropdown menu showing "Flag".
  - Address:** A text box containing "150".
  - Count [1..255]:** A text box containing "5".
- At the bottom are three buttons: "OK", "Cancel", and "Advanced...".

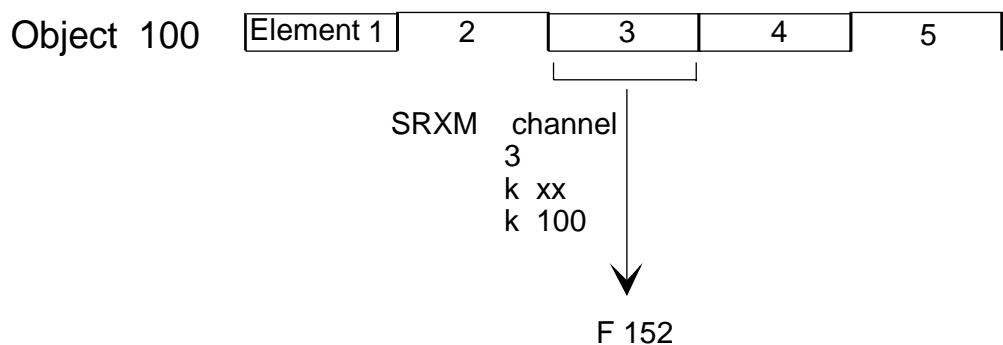
Object      

Element 1	2	3			nmax
-----------	---	---	--	--	------

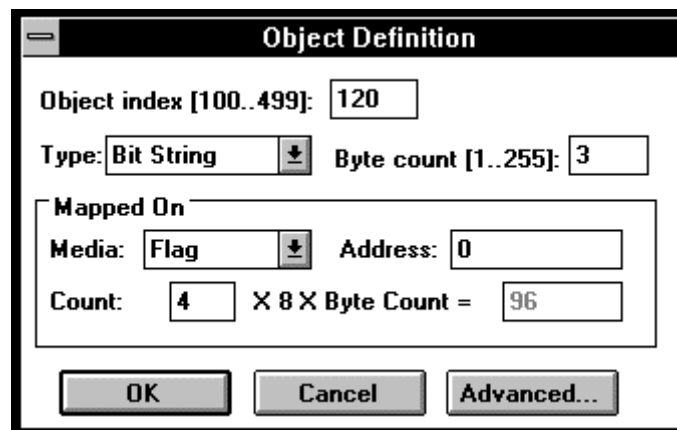
where: nmax = count (number of elements)

Example:

Object index      100  
 Element type:    Boolean  
 Count:            5  
 Media address:   Flag 150



- Object format for data types: **Bit string and octet string**



**Object Definition**

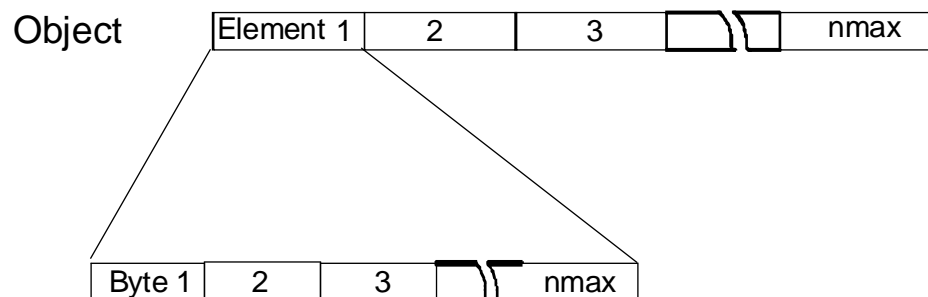
Object index [100..499]: 120

Type: Bit String  Byte count [1..255]: 3

Mapped On

Media: Flag  Address: 0

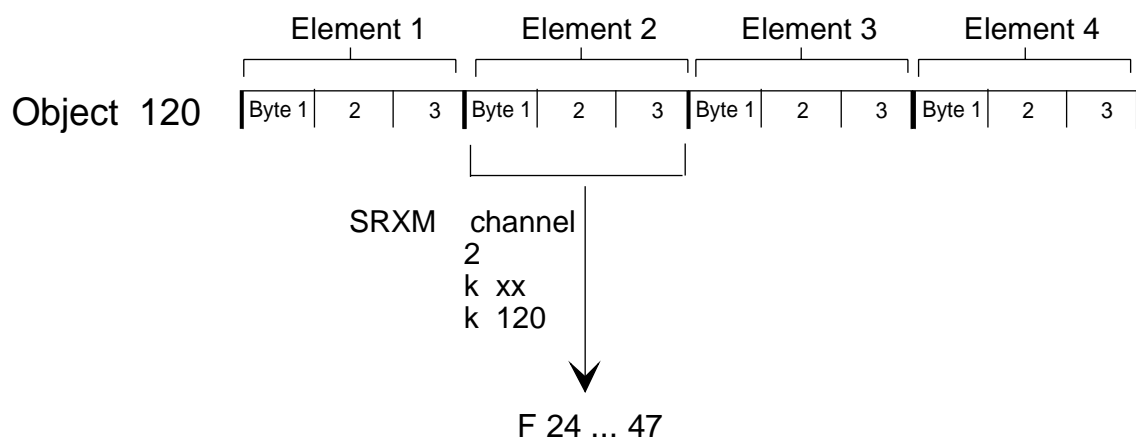
Count: 4 X 8 X Byte Count = 96



where: - Element-nmax = count (number of elements)  
 - Byte-nmax = Byte count (number of bytes per element)

Example:

Object index 120  
 Element type: Byte String  
 Byte count: 3  
 Count: 4  
 Media address: Flag 100

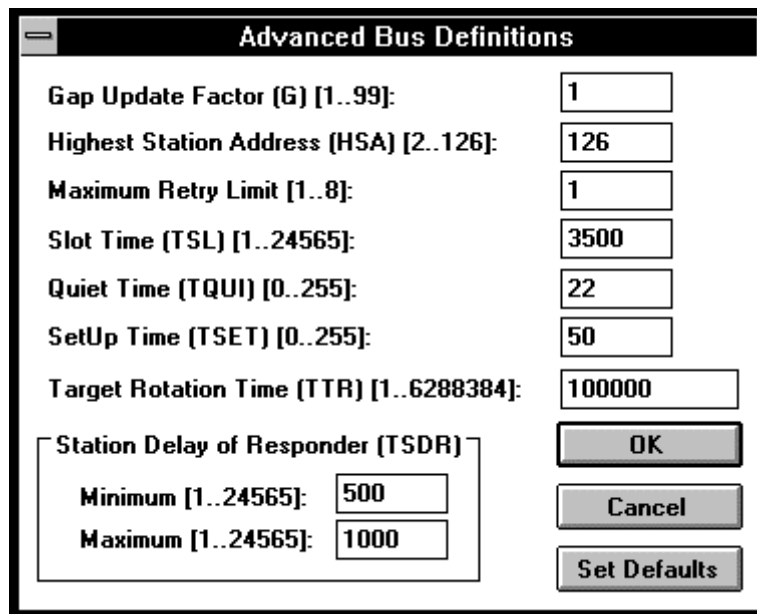


For more information on data types see appendix A.

## 4.4 Advanced Setup

### 4.4.1 Definition of bus parameters

 **Bus parameters must be the same for all stations on the bus !**



The default values for bus parameters are dependent on baud rate. The following data applies for a baud rate of 500 kBd.

#### Gap Update Factor (G):

Range: 1..99  
in multiples of the maximum target rotation time ( $T_{TR}$ )

Default value: 1

This parameter defines the time, in multiples of the target rotation time, between checks of whether any new stations have been added to the bus. It is only a significant parameter for active stations (master).

$$\text{Gap update time} = G * T_{TR}$$

The following rule applies:

Data throughput on the bus increases in direct proportion to the length of  $G * T_{TR}$ .

**Highest Station Address (HSA):**

Range: 2..126  
Default value: 126

This parameter defines the highest station address on the bus. When checking whether new, active stations wish to enter the logical token ring, polling only extends up to this address (i.e. GAP update).

If later the number of stations is increased, this parameter must be modified for all active devices. For time-sensitive applications it is recommended that the smallest possible value is set, as this can significantly increase data throughput.

**Maximum Retry Limit:**

Range: 1..8  
Default value: 1

This parameter specifies how often a layer 2 telegram is repeated, if no answer has been received within the slot time. In high noise environments, this parameter enables reliable operation to be achieved.

This parameter is only necessary for active stations.

**Slot Time (T<sub>SL</sub>):**

Range: 1..24565 bit times  
Default value: 3500 = 3500 \* 2 µs = 7 ms

This parameter defines the timeout for a transmitted, layer 2 telegram.

The master waits for the response to or acknowledgement of a call telegram until the slot time has elapsed. This parameter deals purely with monitoring time and therefore does not affect data throughput.

The following formula can be used to determine this parameter:

$$T_{SL} \gg \text{Max\_}T_{SDR} (1,5..2 * \text{Max\_}T_{SDR})$$

**Quiet Time (T<sub>QUI</sub>):**

Range: 0..255 bit times  
Default value:  $22 = 22 * 2\mu s = 44\mu s$

With this parameter, the switch-over times of repeaters or dying times of modulators can be taken into account.

**SetUp Time (T<sub>SET</sub>):**

Range: 0..255 bit times  
Default value:  $50 = 50 * 2\mu s = 100\mu s$

This parameter defines the time which elapses between the arrival of an event (e.g. last character of telegram transmitted) and the execution of the required reaction (e.g. release of receiver).

**Target Rotation Time (T<sub>TR</sub>):**

Range: 1..6288384 bit times  
Default value:  $100,000 = 100,000 * 2\mu s = 200 \text{ ms}$

From the target rotation time, whichever station is active calculates its own maximum token holding time.

This parameter is only significant for active stations.

The system reaction time is strongly influenced by the target rotation time.

Formula for determining this parameter:

$$T_{TR} \gg nb_M * (2 \text{ Max\_}T_{SDR} + 3000)$$

$nb_M$ : number of master stations in the network

This formula assumes that each master can send a telegram on receipt of the token.



**Minimum Station Delay of Responder (Min\_T<sub>SDR</sub>):**

Range: 1..24565 bit times  
Default value:  $500 = 500 * 2\mu\text{s} = 1 \text{ ms}$

This parameter is essentially determined by the processing speed of the connection (interpretation of the call, composition of the reply/acknowledgement) and the length of the telegram. With this parameter, the transmission of the response/acknowledgement can be delayed and follows the slowest master (telegram initiator). Only after this time is the station ready to receive the acknowledgement.

i.e.:

- Min\_T<sub>SDR</sub> is determined by the slowest master (telegram initiator)
- Min\_T<sub>SDR</sub> > T<sub>QUI</sub> (quiet time)

**Maximum Station Delay of Responder (Max\_T<sub>SDR</sub>)**

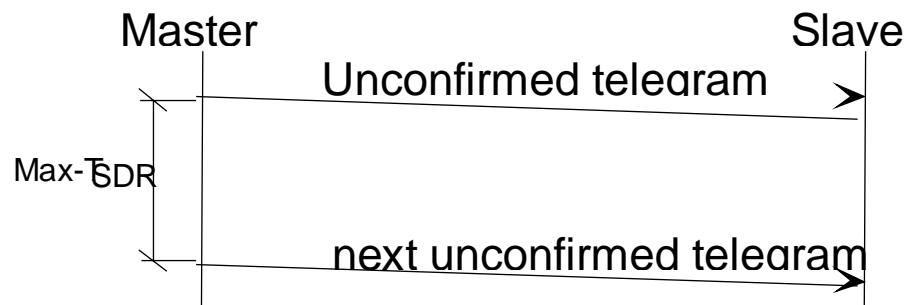
Range: 1..24565 bit times  
Default value:  $1000 = 1000 * 2\mu\text{s} = 2 \text{ ms}$

This parameter is essentially determined by the processing speed of the connection (interpretation of the call, composition of the response/acknowledgement) and the length of the telegram.

There are two different possibilities:

a) Unconfirmed transmitting telegrams:

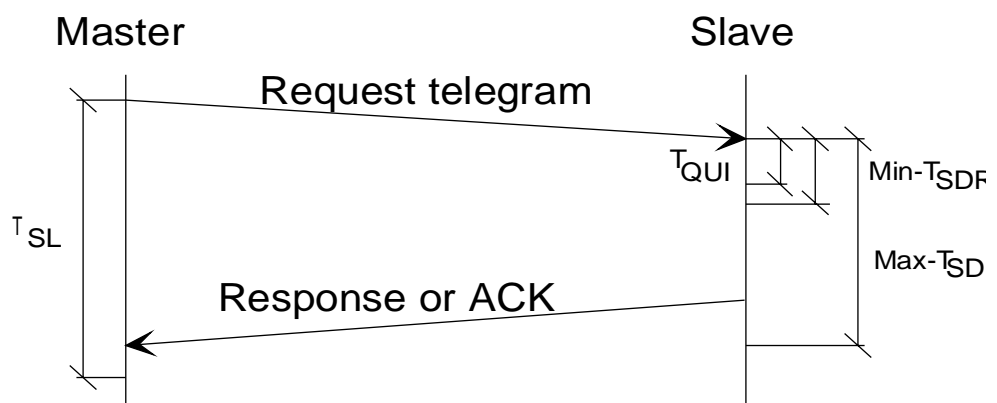
This parameter can be used to define a delay after a transmitted but unconfirmed telegram, before the next telegram is sent. The value set follows the slowest telegram receiver. In such cases, this parameter is only significant for active stations.



i.e.: Max. T<sub>SDR</sub> is determined by the slowest slave.

b) Confirmed request telegrams:

In such cases, this parameter is significant for the slave station. The parameter defines the maximum time which can elapse after a request telegram has been received, before a response or acknowledgement must be returned.



i.e.: Max. T<sub>SDR</sub> < T<sub>SL</sub> (slot time)

#### 4.4.2 Channel definitions

- **Types of communication relations**

PROFIBUS differentiates fundamentally between **connectionless** and **connection-oriented** communication relations. Since the logical return channel is missing in a connectionless communication relation, the user only has available the unconfirmed features for multi or broadcast.

Unconfirmed services are **not** yet supported by SAIA devices.

Confirmed services must be transmitted via connection-oriented communication relations, since the return of the acknowledgement requires a return channel for addressing purposes. In contrast to connectionless communication, before data transfer the connection must be initialized in an establishment phase (SCON instruction) and released again on termination of data transmission.

With connection-oriented communication relations, an additional differentiation is made between a **defined** and an **open connection**.

- **Defined connection:**

With this connection the communications partner is already defined in the planning phase.



For this example, Master 3 has the attributes for the channel 10 communication relation defined as follows:

Remote address:	5
Local SAP:	2
Remote SAP:	5
Connection type:	defined, acyclic, master/master (MMAZ)

- **Open connection:**

This type of connection is mainly used for slave devices which cannot be configured. Open connections can only be opened by the remote device.



For this example, Slave 3 has the attributes for the channel 10 communication relation defined as follows:

Remote address: 255  
 Local SAP: 2  
 Remote SAP: 255  
 Connection type: open, acyclic

The value 255 for remote address and RSAP are global addresses and enable a remote device with any address and any SAP to open this connection. This allows a number of master devices to use the same SAP. However, it is still only possible for one of these connections to be open at the same time.

- **Header Channel Definition**

The image shows a software dialog box titled 'Header Channel Definition'. It contains three input fields with corresponding labels and three buttons on the right. The first field is 'Channel number (CREF):' with the value '0'. The second field is 'Poll Service Access Point [2..62,128]:' with the value '128'. The third field is 'Control Interval (ASS/ABI-CI):' with the value '3000'. The buttons are 'OK', 'Cancel', and 'Set Defaults'.

The communication relationship list (CRL) has a header with data which applies for all communication links.

- \*) Open connections can be defined as follows in the configurator:
- For a defined channel, there is no link created with any other channel.
  - For a single channel, several links are created with channels of various stations.

**Channel number:** CREF = Communication Reference

Range: 0

Reference number for the header. Cannot be modified.

**Poll SAP:**

Range: 2..62, 128, 255

Default value: 128, for a master station

255, for a slave station

This parameter specifies the SAP through which the poll list should be processed, if such a list is present. The poll SAP is used for all communication relationships with passive stations.

**Control Interval:** (ASS/ABT-CI)  
Associate Abort Control Interval

Range:  $0..(2^{32}-1) * 10 \text{ ms}$

Default value:  $3000 = 3000 * 10 \text{ ms} = 3 \text{ s}$

Timeout for establishment and termination of a link (SCON instruction). The value set applies for all link channels.

Formula for determining the parameter:

Control interval  $\gg G * T_{TR}$

- **Advanced Channel Definition**

Channel Definition Advanced			
<b>Fms features supported:</b>		<b>Maximum service counters:</b>	
Request:	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	Send confirmed (SCC) [1..15]:	<input type="text" value="1"/>
Response:	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	Receive confirmed (RCC) [1..15]:	<input type="text" value="15"/>
	<input checked="" type="checkbox"/> Get OV	Send acknowledged (SAC):	<input type="text" value="0"/>
		Receive acknowledged (RAC):	<input type="text" value="0"/>
<b>Maximum PDU length:</b>			
High priority message:			
Received: 0	Sent: 0		
Low priority message:			
Received [0,31..242]: <input type="text" value="241"/>	Sent [0,31..242]: <input type="text" value="241"/>		
Acyclic/Cyclic control interval (ACI/CCI): <input type="text" value="0"/>			
		<input type="button" value="OK"/>	
		<input type="button" value="Cancel"/>	
		<input type="button" value="Set Defaults"/>	
		<input type="button" value="Access rights..."/>	

**FMS features supported:**

Defines the FSM services supported for this channel.

Request:	Services for request telegrams
Response:	Services for response telegrams (Get OD: read object dictionary, is always supported)

The responder (server or slave) must support at least all services of the requester (master). This precondition is checked when the link is established.

**Maximum service counters:**

Send confirmed (SCC):

Range:	0..15
Default value:	1

This parameter specifies the maximum number of outstanding responses which the station as client on this channel can process. For a straight server, the value 0 is entered.

For a master with a cyclic link, the value 0 is entered.

i.e.:  $SCC \leq RCC$  of partner

Parallel transmission services are not supported by SAIA devices. Any value higher than 1 is therefore irrelevant and without influence.

Receive confirmed (RCC):

Range:	0..15
Default value:	15

This parameter specifies the maximum number of confirmed services (requests) which the station as server on this channel can accept, without sending a response.

For a channel with a cyclic link, the value 0 is entered.

i.e.:  $RCC \geq SCC$  of partner

**Send unconfirmed (SAC) (acknowledged):**

Default value: 0

Maximum permissible number of unconfirmed, parallel services as sender.

Unconfirmed and parallel services are not supported by SAIA devices.

**Receive unconfirmed (RAC) (acknowledged):**

Default value: 0

Maximum permissible number of unconfirmed, parallel services as receiver.

Unconfirmed services are not supported by SAIA devices.

**Maximum PDU length of high priority message (MPRH & MPSH):**

Range: 0

Telegrams with high priority are not supported. Only unconfirmed telegrams have high priority.

**Maximum PDU length of low priority message:**

Range: 0, 31..242 Bytes

Default value: 241 Bytes

**Received (MPRL):**

This parameter specifies the maximum permissible PDU size (Protocol Data Units) of FMS telegrams with low priority for the receiver.

i.e.:

Max. PDU-Received > = Max. PDU-Sent of partner

This precondition is checked during any Initiate (SCON instruction).

**Sent (MPSL):**

This parameter specifies the maximum permissible PDU size (Protocol Data Units) of FMS telegrams with low priority for the sender.

i.e.:

Max. PDU-Send < = Max. PDU-Received of partner

This precondition is checked during any Initiate (SCON instruction).

**Acyclic/Cyclic control interval:**

## a) Cyclic connections (CCI):

Range: 0..(2<sup>32</sup>-1) in units of 10 ms

Default value: 3000

This value specifies the control interval for cyclic connections.

Monitoring is obligatory for the master in any cyclic connection, optional for the slave.

Rule: For a master the value should always be > 0.

## b) Acyclic connections (ACI)

Range: 0..(2<sup>32</sup>-1) in units of 10 ms

Default value: 3000 for master-master connections  
0 for master-slave connections

This value specifies the interval for monitoring acyclic connections. If no live telegrams are transferred along this connection, both stations send an idle telegram three times per interval. If within this interval either station fails to receive an idle or live telegram, it shuts down the communication channel.

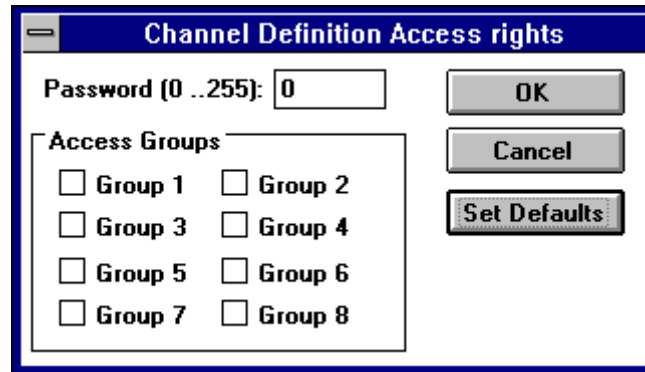
i.e.: The interval must be the same for both partners.  
This is checked when the connection is established.

Rule: ACI > 0 for master-master connections  
ACI = 0 for master-slave connections



- **Channel Definition Access rights**

A password with various access groups can be assigned to each communication channel. This enables access to protected objects from the remote device.



The image shows a dialog box titled "Channel Definition Access rights". It has a blue title bar. Inside, there is a text field labeled "Password (0 ..255):" with the value "0" entered. To the right of this field are three buttons: "OK", "Cancel", and "Set Defaults". Below the password field is a section titled "Access Groups" which contains eight checkboxes arranged in two columns, labeled "Group 1" through "Group 8". All checkboxes are currently unchecked.

**Password (PW):**

Range: 0..255  
Default value: 0 = no password

The password valid here for this channel is sent to the partner station when the connection is established (SCON instruction). If later an object is accessed via this connection, the partner checks whether the password received when the connection was established matches the password entered at the object.

**Access Groups (GR):**

In addition to the password, a channel can be assigned with a total of 8 different access groups. The access groups are also advised during establishment of the connection, and checked later when the partner accesses any objects.

### 4.4.3 Object definition

- **Object Access rights**

Although access rights can be defined in the configurator, they are not supported by the SAIA PROFIBUS firmware.

For each object, various access rights can be defined, which are compared with the access rights of the partner device when a connection is opened.

**Access rights:**

- **Communication partners:**  
Defines access for partner devices without a password.
- **Password processor:**  
Defines access for partner devices with a password.
- **Group:**  
Defines access for partner devices belonging to a certain group (1..8).

**Password:**

Range: 0..255  
Default value: 0 = no password

A password can be assigned to the object. Only devices with the same password have the access rights defined in the password processor.

**Access Groups:**

Here the object can be assigned with certain access groups, in addition to the password.

## 5. Programming

---

### 5.1 Description of PCD instructions (SASI, SCON, STXM and SRXM)

---

The following instructions are supported by PROFIBUS:

SASI	Assign serial interface
SASII	Like SASI but indirect
SCON	Serial connect
SCONI	Like SCON but indirect
SRXM	Serial receive media
SRXMI	Serial receive media indirect
STXM	Serial transmit media
STXMI	Serial transmit media indirect

The description of the configurator showed two peculiarities of PROFIBUS compared with the communication methods which are already familiar (S-Bus, Mode D):

1. With PROFIBUS, telegram transmission is via communication channels. This method allows several channels for a single partner station. Each channel allows various services, e.g. cyclic and acyclic connection. The PCD instruction therefore does not specify the address of the partner station, but the index of the channel across which data is to be exchanged.
2. Objects are used for the transfer of data. The write and read instructions contain the index of these objects, not the actual media to be transferred (I, O, F, R). The connection between the objects and the media is known to the firmware, due to the “Object Definition” in the SASI instruction, and its correct use is automatic.

### 5.1.1 SASI Assign a channel

**Description:** PROFIBUS assignment takes place, as with most PCD communications modes, with the SASI instruction. SASI calls a text which contains the necessary information for initializing a virtual channel. If communication is to take place across several virtual channels, a separate SASI instruction should be used for each channel.

**Format:**

SASI	channel text
------	-----------------

channel	10..99 10..19	for PCD4.M445 for PCD2 with PCD7.F700
---------	------------------	--

text	T_As_xx	Symbolic name of the text containing the assignment information.
------	---------	--

**Example:**

SASI	12	; Initialize channel 12
	T_As_12	; Definition text for
		; channel 12

**Flags:** The error (E) flag is set if the definition text is missing or invalid.

**Diagnostics:** Diagnostics of a PROFIBUS communication takes place in the usual way for the PCD, i.e. for each communication channel, 8 flags are assigned for rough diagnostics plus a register for fine diagnostics. These diagnostic resources are defined in the configurator.

**Diagnostic flags:**

Adresse	Nom	Description
xxxx	RBSY	Receiver busy
xxxx+1	RFUL	Receive buffer full
xxxx+2	RDIA	Receiver diagnostic
xxxx+3	TBSY	Transmitter busy
xxxx+4	TFUL	Not used
xxxx+5	TDIA	Transmitter diagnostic
xxxx+6	XBUY	Cross busy
xxxx+7	NEXE	Not executed

**Receiver Busy (SBSY)** not used.

**Receive Buffer Full (RFUL)**

Client: not used

Server: High indicates that a write telegram has been received and therefore PCD media have been written.

**Receiver Diagnostic (RDIA)**

Set high when an error is detected during receipt of a telegram. A detailed description of the error can be obtained from the diagnostic register (bits 0..15). The flag is reset as soon as all receiver diagnostic bits (0..15) have been reset in the diagnostic register.

**Transmitter Busy (TBSY)**

Client: Set high for as long as transmission is underway. This applies during execution of any STXM, SRXM or SCON instructions. It is reset when the response or acknowledgement has been received.

Server: Not used

**Transmitter Diagnostic (TDIA)**

Set high when an error is detected during transmission of a telegram. A detailed description of the error can be obtained from the diagnostic register (bits 16..31). The flag is reset as soon as all transmitter diagnostic bits (16..31) have been reset in the diagnostic register.

**Not executed (NEXE)**

Set high when, after three attempts, it has not been possible to execute an instruction (STXM or SRXM). The flag is reset at the next instruction.

Client:           Service requester (master)  
 Server:           Service provider (master or slave)

**Diagnostic registers:**

	Bit	Designation	Description
R E C E P T E U R	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9	Format error	Invalid instruction
	10		
	11		
	12	Range error	Invalid object
	13		
	14		
	15		
E M E T T E U R	16		
	17		
	18		
	19		
	20	Negative response	Negative response (NAK)
	21		
	22		
	23		
	24		
	25	Format error	Invalid instruction
	26		
	27		
	28	Range error	Invalid object, or resource outside range
	29		
	30	Response error	Response does not correspond to enquiry
	31	Program error	STXM/SRXM instruction not allowed

**Format error (bit 9)**

Set high when the dual-port RAM contains an invalid instruction (<> read, write, init, abort).

**Range error (bit 12)**

Set high when an invalid object is received:

- With a “Get\_OD\_i” instruction, when the object does not exist (index outside the range or not defined during assignment).
- With a “Read\_i” instruction, when the source object does not exist (index outside the range or not defined during assignment) or sub-index > total resources of object.
- With a “Write-i” instruction, when the destination object does not exist (index outside the range or not defined during assignment), or sub-index > total resources of object, or total bytes received <> total bytes in object.

**Negative response (bit 20)**

Set high:

- On receipt of a negative confirmation after an “Initiate” instruction (opening a channel). Check: PDU length, FMS characteristics, OD version, password, profile number.
- On receipt of a negative confirmation for a “Confirmed service” (read, write). Check: source and destination objects, and sub-index.

**Format error (bit 25)**

Set high when the “taco” buffer contains an invalid instruction (<> read, write, init, abort).

**Range error (bit 28)**

- With an SRXM or STXM instruction, when:
  - a) The object was not defined during assignment.
  - b) The object index is outside the range (index < K 100 or index > K 499).
  - c) The index of the remote object is greater than K 16383.
  - d) The sub-index is greater than 255.
  - e) The sub-index is greater than the total resources of the object.
- With a “Read\_c” instruction, when the total bytes received does not correspond to the total bytes in the object.

**Response error (bit 30)**

Set high when the response received does not correspond to the enquiry.

- The service of the telegram for confirmation does not agree with the enquiry telegram(Init, read, write).
- The “Invoke\_ID” of the confirmation telegram does not agree with the “Invoke\_ID” of the enquiry (read\_C, write\_c).

**Program error (bit 31)**

Set high during execution of an instruction which is not allowed.

- With an STXM or SRXM instruction, when the channel is closed (poll XBSY).
- Executing an SCON, STXM or SRXM instruction, when a similar instruction is already being processed (TBSY flag not polled before execution of the instruction).



### 5.1.2 SCON: Open a communication channel

**Description:** Before exchanging information, it is necessary to open (initialise) the virtual communication channel first. For this operation, the SCON command is used. In order not to charge the network when it is not necessary, we advise to close (abort) the channel, when the link is not longer used.

**Format:**

SCON	channel (station) state
------	-------------------------------

channel	10..99 10..19	for PCD4.M445 for PCD2 with PCD7.F700
(station)	0	Not used. This parameter is defined in the channel description. It becomes superfluous with PROFIBUS.
state	0 / 1	0 = close the channel 1 = open the channel

**Example:**

```

STL      TBSY
ANL      XBSY
JR       L next
SCON   10      ; channel 10
          0      ;
          1      ; open

next:
```

**Flags:** The error flag is set if the channel is not assigned.

### 5.1.3 SRXM: Receive data (objects) from a remote station

**Description:** This command reads data (objects) from the remote station and copies them in the local PCD. A similar instruction format to the one for the mode MD/SD is used.

**Format:**

SRXM	channel	
	count	
	source	; remote
	dest	; local

channel	10..99	for PCD4.M445
	10..19	for PCD2 with PCD7.F700

count	0..255	Sub-index from source and destination object. Note: this forces to have the same structure in both objects. 0 = no sub-index.
-------	--------	---

source	K 0..16383	Source object index (remote station)
--------	------------	--------------------------------------

destination		Destination object index (own station)
	K 100..499	PCD4.M445
	K 100..199	PCD2 with PCD7.F700

**Example:**

STL	TBSY	
ANH	XBSY	
JR	L next	
SRXM	13	; channel 13
	0	; without sub-index
	K 22	; copies object 22 from the
		; remote station
	K 150	; to the object 150 of the
		; own station

next:

**Flags:** The error flag is set if the channel is not assigned.

### 5.1.4 SRXMI: Receive indirect data (objects) from a remote station

**Description:** With this instruction, data (objects) are read from a remote station. It is possible to select direct or indirect addressing of channel operands. For other operands, addressing is always indirect, via registers.

This instruction is supported with effect from the following firmware versions:

PCD4.M445 V005  
PCD2.M120 V003

Utility version:

V1.9 or higher

This instruction can neither be indexed nor be parameterised

#### Format:

SRXMI	channel	; 10..99, R 0..4095
	count	; R 0..4095
	source	; K [R 0..4095]
	destination	; K [R 0..4095]

channel      10..99      for PCD4.M445  
                 10..19      for PCD2 with PCD7.F700  
         or      R 0..4095      for indirect addressing

count      R 0..4095      register contains the sub-index 0..255 of the source and the destination object (0 = complete object).

source      K [R 0..4095]      register contains 0..16383 of the source object (remote station)

destination      K [R 0..4095]      register contains index of the destination object (own station)  
PCD4.M445, index: 100..499  
PCD2.M120, index: 100..199

**Example:**

LD	R 50	;
	12	; channel 12
LD	R 51	
	0	; sub-index = 0
LD	R 52	
	172	; source object index 172
LD	R 53	
	145	; destination object index 145
STL	TBSY	
ANH	XBSY	
JR	L next	

**SRXMI**

R 50	; channel number in R 50
R 51	; sub-index in R 51
K 52	; source object index in R 52
K 53	; dest object index in R 53

next:

**Flags:** The error flag is set if the channel is not assigned.

### 5.1.5 STXM: Transmit data (objects) to a remote station

**Description:** This command transmits (write) data from the local PCD to the remote station. A similar instruction format to the one known for the mode MD/SD is used.

**Format:**

STXM	channel	
	count	
	source	; local
	destination	; remote

channel      10..99      for PCD4.M445  
                  10..19      for PCD2 with PCD7.F700

count          0..255      Sub-index from source and destination object.  
                                  Note: this forces to have the same structure in both objects.  
                                  0 = no sub-index

source                              Index of the source object (own station)  
                                  K 100..499      PCD4.M445  
                                  K 100..199      PCD2 with PCD7.F700

destination    K 0..16383      Index of the destination object (remote station)

**Example:**

STL            TBSY  
 ANH           XBSY  
 JR             L next

**STXM**        11                    ; channel 11  
                  3                    ; sub-index (element) 3  
                  K 122                ; copy object 122 from the  
    ; own station  
                  K 150                ; to the object 150 of the-  
    ; remote station

next:

**Flags:**                    The error flag is set if the channel is not assigned.

### 5.1.6 STXMI: Transmit indirect data (objects) to a remote station.

**Description:** This command transmits (write) data from the local PCD to the remote station. It is possible to select direct or indirect addressing of channel operands. For other operands, addressing is always indirect, via registers.

This instruction is supported with effect from the following firmware versions:

PCD4.M445 V005  
PCD2.M120 V003

Utility version:

V1.9 or higher

This instruction can neither be indexed nor be parametrised

#### Format:

STXMI	channel	; 10..99, R 0..4095
	count	; R 0..4095
	source	; K [R 0..4095]
	destination	; K [R 0..4095]

channel	10..99	for PCD4.M445
	10..19	for PCD2 with PCD7.F700
or	R 0..4095	for indirect addressing
count	R 0..4095	Register contains a sub-index 0..255 of the source and the destination object (0 = no sub-index).
source	K [R 0..4095]	Register contains index of the source object (own station) PCD4.M445, index: 100..499 PCD2.M120, index: 100..199
destination	K [R 0..4095]	Register contains index 0..16383 of the destination object (remote station).

**Example:**

LD	R 100	;
	12	; channel 12
LD	R 101	
	0	; sub-index = 0
LD	R 102	
	172	; source object index 172
LD	R 103	
	45	; dest. object index 45
STL	TBSY	
ANH	XBSY	
JR	L next	
<b>STXMI</b>	R 100	; channel number in R 100
	R 101	; sub-index in R 101
	K 102	; source object index in R 102
	K 103	; dest. object index in R 103

next:

**Flags:** The error flag is set if the channel is not assigned.

## 5.2 Important rules for the user program

---

Starting with a review of the main rules for the PROFIBUS user program:

- The user program for PROFIBUS communications must be loaded into CPU 1, as this is the only one with access to the PROFIBUS controller.
- As usual with SAIA<sup>®</sup> PCD, each communications interface (communications channel in PROFIBUS) must be initialized with a SASI instruction. This is normally done in XOB 16.
- Before data can be exchanged, a communications channel must be opened by an SCON instruction.
- One channel should only be opened by one station (normally by the client).
- If an SCON instruction is processed for a channel which is already open, it will subsequently be closed.
- SCON and STXM/SRXM instructions may only be processed when the TBSY flag is low.
  - Before each instruction, poll the TBSY flag.
- Communications are only possible via an open channel.
  - Before each STXM/SRXM instruction, poll XBSY flag.



## 5.3 Structure of the user program

---

### 5.3.1 Cold-start program in XOB 16

In XOB 16 the file generated by the configurator (\*.CON), which contains the SASI texts, is integrated into the user program with \$INCLUDE.

As a rule, the communications channels are also initialized in XOB 16.

Example:

```
$INCLUDE Stat_A.CON ; File generated by the
                    ; configurator
                    ; Stat_A = Name of
                    ; PROFIBUS station
.
.

XOB 16

SASI 10            ; Channel 10
      T_As_10      ; SASI text for channel 10
SASI 11            ;
      T_As_11      ;

.
.
EXOB
```

### 5.3.2 Main program in the COB

In the main program, the SCON instruction is programmed to open the communications channel. In principle, the channel could also be opened in XOB 16. However, since the channel can be closed if there is an error, or shut down at any time by the remote station, it is preferable to monitor the channel cyclically in a COB, opening it if necessary with the SCON instruction.

The communications instructions (STXM/SRXM) can be placed either in a cyclical program or a GRAFTEC structure.

Example 1:

Communications in BLOCTEC structure without evaluation of diagnostic elements.

```

COB          0
              0

              STL      TBSY_10      ; TBSY = Low
              ANL      XBSY_10      ; AND XBSY = Low
              JR        L      NEXT1 ; then
              SCON      10           ; open channel 10
              0
              1

NEXT1:        STL      TBSY_11      ; TBSY = Low
              ANL      XBSY_11      ; AND XBSY = Low
              JR        L      NEXT2 ; then
              SCON      11           ; open channel 11
              0
              1

NEXT2:        STL      TBSY_10      ; TBSY = Low
              ANH      XBSY_10      ; AND XBSY = High
              JR        L      NEXT3 ; then
              STXM      10           ; transmit via channel 10
              0
              K 100      ; source object 100 of own station
              K 150      ; to destination object 150 of the
              ; partner

NEXT3:        STL      TBSY_11      ; TBSY = Low
              ANH      XBSY_11      ; AND XBSY = High
              JR        L      END    ; then
              SRXM      11           ; transmit via channel 11
              0
              K 120      ; source object 120 of the partner
              K 250      ; to destination object of own
              ; station
              .
              .

END:          ECOB

```

## Example 2:

Communications in a GRAFTEC structure without evaluation of diagnosis.

```

COB          0
              0

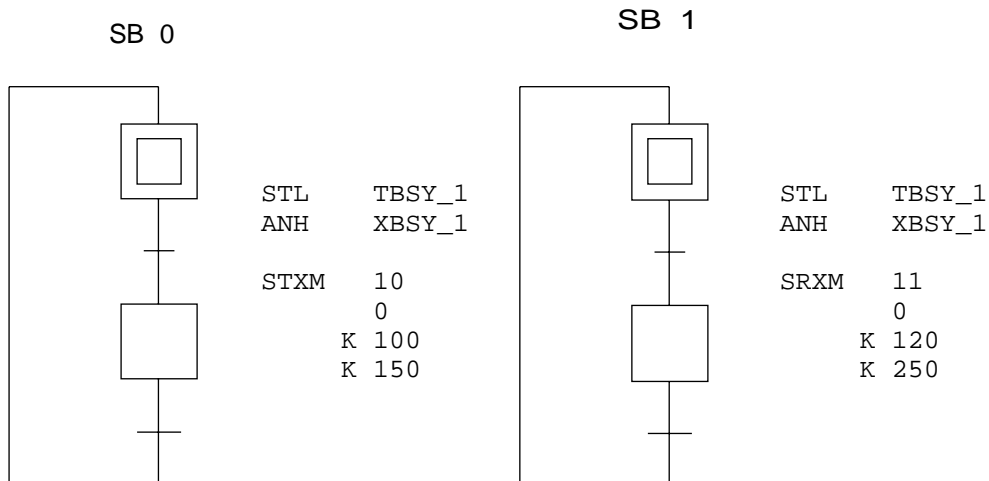
STL          TBSY_10      ; TBSY = Low
ANL          XBSY_10      ; AND XBSY = Low
JR    L      EXT1         ; then
SCON         10          ; open channel 10
              0
              1

NEXT1: STL    TBSY_11      ; TBSY = Low
ANL          XBSY_11      ; AND XBSY = Low
JR    L      NEXT2        ; then
SCON         11          ; open channel 11
              0
              1

NEXT2: CSB    0            ; Comms SB for channel 10
CSB          1            ; Comms SB for channel 11

ECOB

```

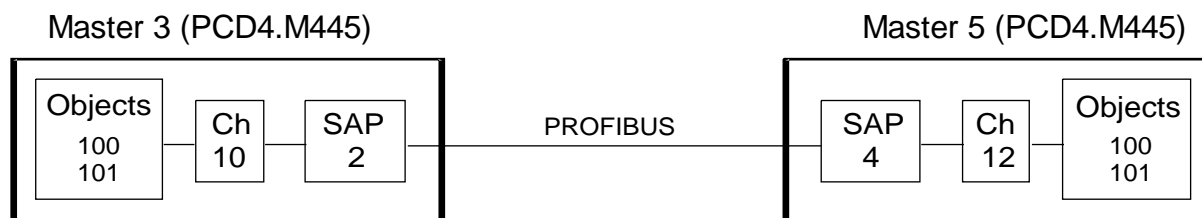


## 5.4 Program examples

The following program examples are available on diskette.

### 5.4.1 Example 1: 2 PCD stations

#### Network:



#### Task:

Both stations transfer their local inputs 0..7 to outputs 32..39 of the remote station. The communications channel is opened by Master 3.

- Definition of communications connections:

#### Master 3 (PCD4.M445):

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
10	2	Master 5	12	4	acyclic

#### Master 5 (PCD4.M445):

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
12	4	Master 3	10	2	acyclic

- Definition of objects:

#### Master 3:

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
100	Bit string/1	r/w	I 0..7	Master 5	101	Bit string/1	r/w	O 32..39
101	Bit string/1	r/w	O 32..39	Master 5	100	Bit string/1	r/w	I 0..7

#### Master 5:

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
100	Bit string/1	r/w	I 0..7	Master 3	101	Bit string/1	r/w	O 32..39
101	Bit string/1	r/w	O 32..39	Master 3	100	Bit string/1	r/w	I 0..7

### 5.4.2 Example 2: 1 PCD station and 1 other station

#### Network:



#### Task:

Master 2 transfers the local inputs 0..7 to outputs 8..15 of the Weidmüller slave. The slave inputs 0..7 shall be read and copied to the outputs 32..39.

#### • Definition of communications connections:

##### Master 2 (PCD4.M445):

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
10	128	Slave 3	10	3	acyclic

##### Slave 3 (Weidmüller):

Own station		Partner station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
10	3	Master 2	10	128	acyclic

#### • Definition of objects:

##### Master 2:

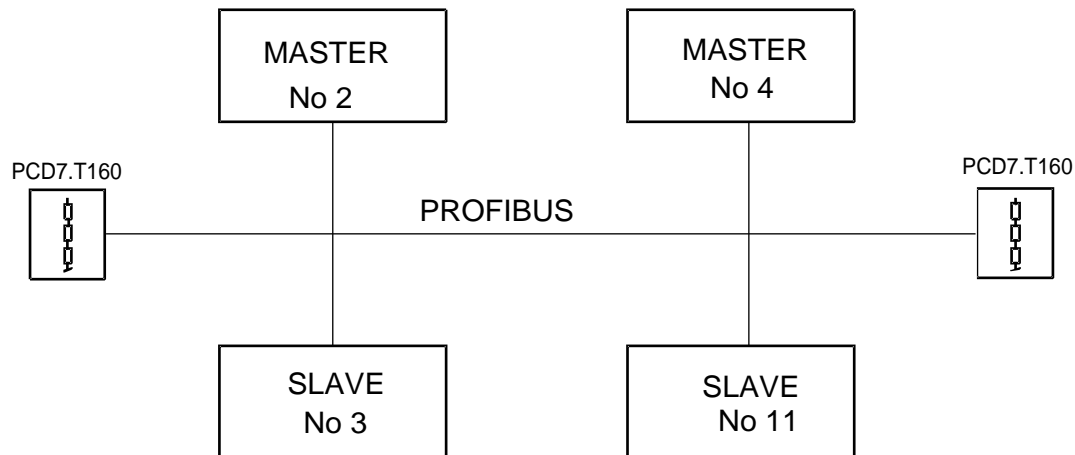
Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
100	Octet-String/1	r/w	I 0..7	Slave 3	51	Octet-String/1	r/w	O 16..23
101	Octet-String/1	r/w	O 32..39	Slave 3	50	Octet-String/1	r/w	I 0..7

##### Slave 3:

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
50	Octet-String/1	r/w	I 0..7	Master 2	101	Octet-String/1	r/w	O 32..39
51	Octet-String/1	r/w	O 8..15	Master 2	100	Octet-String/1	r/w	I 0..7

### 5.4.3 Example 3: 2 PCD stations and 2 other stations

#### Network:



#### Bus participating stations:

Master 2:	PCD4.M445
Master 4:	PCD4.M445
Slave 3:	Weidmüller remote I/O with digital inputs/outputs
Slave 11:	Festo valve island with digital inputs/outputs

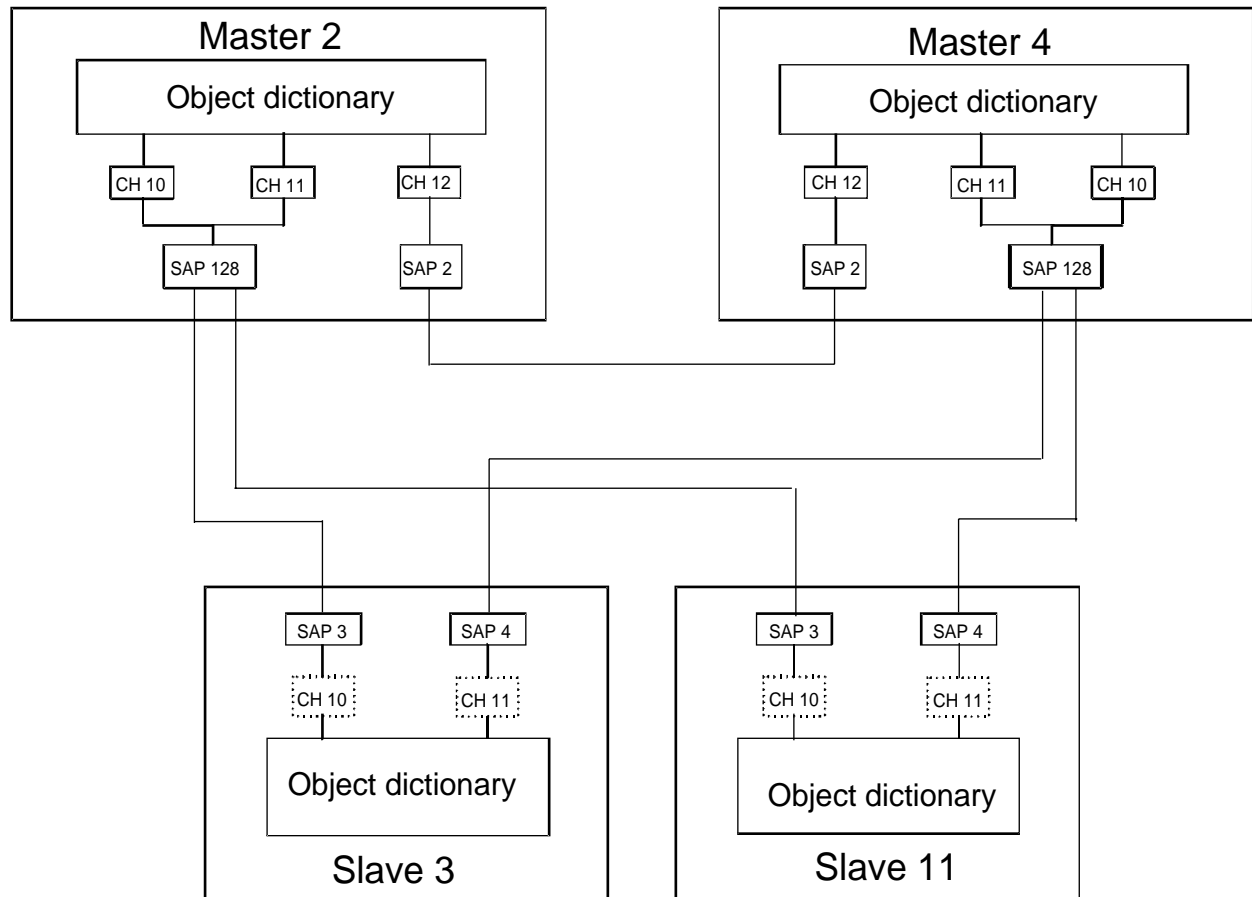
#### Tasks:

##### a) Master 2:

- Copy register 150 to R 2500 on Master 4.
- Read inputs 0..7 from Slave 3 and copy to outputs 32..39.
- Read inputs 0..15 from Slave 11 and copy to outputs 48..63.
- Copy inputs 0..7 to outputs 8..15 on Slave 3.
- Open communications channel to Master 4.

##### b) Master 4:

- Read inputs 0..7 from Slave 3 and copy to outputs 32..39.
- Read inputs 0..15 from Slave 11 and copy to outputs 48..63.
- Copy inputs 0..15 to outputs 16..31 on Slave 11.

**Communication relations:**

- **Definition of communications connections:**

**Master 2 (PCD4.M445):**

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
12	2	Master 4	12	2	acyclic
10	128	Slave 3	10	3	acyclic
11	128	Slave 11	10	3	acyclic

**Master 4 (PCD4.M445):**

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
12	2	Master 2	12	2	acyclic
11	128	Slave 3	11	4	cyclic
10	128	Slave 11	11	4	acyclic

**Slave 3 (Weidmüller):**

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
10	3	Master 2	10	128	acyclic
11	4	Master 4	11	128	cyclic

**Slave 11 (Festo):**

Local station		Remote station			Connection type
Chan.	LSAP	System	Chan.	RSAP	
10	3	Master 2	11	128	acyclic
11	4	Master 4	10	128	acyclic



- **Definition of objects:**

**Master 2 (PCD4.M445):**

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
100	Integer-32/1	r/w	R 150	Master 4	100	Integer-32/1	r/w	R 2500
101	Bit string/1	r/w	O 32..39	Slave 3	50	Bit string/1	r/w	I 0..7
102	Bit string/1	r/w	I 0..7	Slave 3	51	Bit string/1	r/w	O 8..15
103	Bit string/2	r/w	O 48..63	Slave 11	40	Bit string/2	r/w	I 0..15

**Master 4 (PCD4.M445):**

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
100	Integer-32/1	r/w	R 2500	Master 2	100	Integer-32/1	r/w	R 150
101	Bit string/1	r/w	O 32..39	Slave 3	50	Bit string/1	r/w	I 0..7
102	Bit string/2	r/w	I 0..15	Slave 11	42	Bit string/2	r/w	O 16..31
103	Bit string/2	r/w	O 48..63	Slave 11	40	Bit string/2	r/w	I 0..15

**Slave 3 (Weidmüller):**

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
50	Bit string/1	r/w	I 0..7	Master 2	101	Bit string/1	r/w	O 32..39
50	Bit string/1	r/w	I 0..7	Master 4	101	Bit string/1	r/w	O 32..39
51	Bit string/1	r/w	O 8..15	Master 2	102	Bit string/1	r/w	I 0..7

**Slave 11 (Festo):**

Local station				Remote station				
Object	Data type/count	Access	Media	System	Object	Data type/count	Access	Media
40	Bit string/2	r/w	I 0..15	Master 2	103	Bit string/2	r/w	O 48..63
40	Bit string/2	r/w	I 0..15	Master 4	103	Bit string/2	r/w	O 48..63
42	Bit string/2	r/w	O 16..31	Master 4	102	Bit string/2	r/w	I 0..15

**Notes :**

## 6. Commissioning a PROFIBUS network

---

Commissioning a PROFIBUS network is completed in two steps:

- a) Checking and testing the hardware installation (physical layer)
- b) Checking and testing layer 7 (configuration and exchange of data)

### 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 any adjusted line termination resistors, repeaters, ...
- Check the bus line for through transmission
- Check the electrical signal level

etc.

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 level 7

---

These tests are to check the network configuration and application program for data exchange.

### 6.2.1 Initiate

These tests are to check projected communication relationships (connection channels) and parameters set with the configurator. Experience shows that the greatest problems occur during initiation of a connection.

Initiation can be checked as follows:

1. Channel assignment by manual execution of the SASI instruction. This is achieved in the debugger with the “Instruction...” command, or by processing the user program in “Trace” (step-by-step execution). The error flag should be observed. If an error occurs, the following points should be checked:
  - Syntax of SASI instruction
  - In the configurator, was the configuration file (\*.CON) generated with the “Output Station” command?
  - Was this configuration file included in the user program (\$include xx.CON)?
  - Was the user program assembled, linked and downloaded into CPU 1 (PCD4.M445 only)?
2. Initiation checking by manual execution of the “SCON” instruction (debugger with “Instruction...” or “Trace”). Successfully executed initiation is indicated by an XBSY flag which has been set high. If after a few seconds the XBSY flag is still not high, the following points should be checked:
  - Syntax of SCON instruction
  - Was the appropriate channel assigned in advance?
  - Is the remote station on the network and switched on?
  - Was the channel previously assigned at the remote station?
  - Check the channel and SAP numbers set and used
  - Check the initiate control interval set (Control Interval in “Header Channel Definition” menu)
  - Has bit 20 (negative response) been set in the diagnostic register?

If so, the following causes are possible:

- FMS-PDU size (max. message length) not compatible.
- FMS features not supported.
- Incorrect password.
- Cyclic control interval for monitoring link is incompatible.
- Profile number incompatible.
- Version number of object dictionary incompatible

→ The points listed above should be checked for both the local and the remote stations (see also chapter 4.4.2 of this manual).

### 6.2.2 Abort

The “SCON” instruction can be used to abort links either from the client or from the server. In special error situations, an abort is also executed by the protocol software.

The following error situation leads, among other things, to an automatic abort:

- Control interval exceeded (ACI).

### 6.2.3 Check data exchange

After initiation of the link, it is usually access to variables which most produces most problems. Errors in the user program or configuration are often the cause.

When checking, it is advisable to test all planned access to variables manually with the debugger using the STXM or SRXM instructions, or to run through the user program in “Trace” (step-by-step execution). During this, the diagnostic resources of the communication channel used should be observed. Possible causes of incorrect transfer can also be found in chapter 5.1.1 “SASI diagnosis” of this manual.

#### **6.2.4 Optimization of transmission times**

Optimum response and transmission times are achieved by adjusting or optimizing the bus parameters. However, this is not a trivial matter and requires a very good knowledge of PROFIBUS. Therefore, if there are no special demands on transmission times, it is advisable to leave the bus parameters at their standard values. These have been chosen to enable communications with a wide range of devices.

If, despite this, it is ever necessary to adjust the parameters, the rules and recommendations in chapter 4.4.1 “Definition of bus parameters” must always be followed.

Appendix B contains examples which clarify the effect of individual parameters on response and transmission times.

#### **6.2.5 Use of bus monitors**

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

A bus monitor is a testing tool for the commissioning, maintenance and diagnostics of PROFIBUS 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 TMG I-Tec (Germany)**

This bus monitor consists of MS-DOS software and a PC-PROFIBUS add-on card, which provides the physical connection to the bus.

This monitor can trace and analyse FMS and DP telegrams with time stamping up to baud rates of 500 kbps. 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 always 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 FMS 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 FMS services and the context management services.

Appendix C contains information about FMS services and their coding, and a description of context management services with possible error codes.

**Notes :**



## Appendix A: Object data types

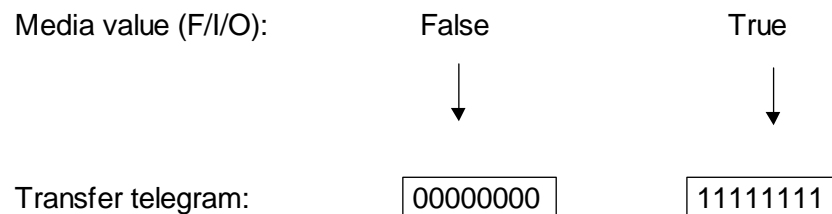
---

- **Boolean**

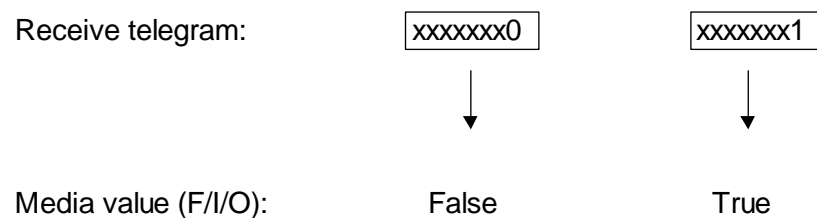
Range:            0        =        FALSE  
                     1        =        TRUE

Transfer PCD media → send-telegram

- The value "false" is transmitted in the telegram as 00000000.
- The value "true" is transmitted in the telegram as 11111111.



Transfer telegram → PCD media



Only the LSBit is taken in consideration when a received boolean telegram is copied to the PCD media.

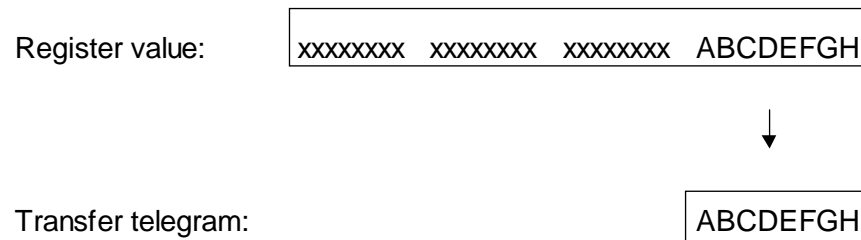
- **Integer 8/16/32**

Range: ....

Transfer PCD media (register) → send-telegram

For objects of type "integer 8" or "integer 16", the upper significant bits of the PCD are not taken in consideration.

Example: "integer-8"



Transfer receive telegram → PCD media (register)

When a received "integer 8" or "integer 16" telegram is copied to the PCD media, the upper significant bits are set with the same value as the sign bit.

Example: "integer 8"



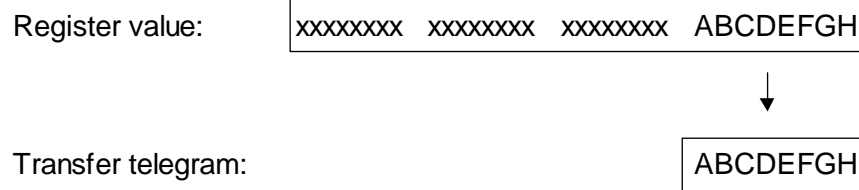
## • Unsigned 8/16/32

Range: ....

Transfer PCD media (register) → send-telegram.

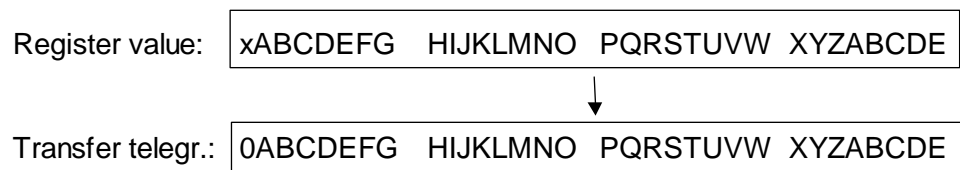
For objects of type "unsigned 8" or "unsigned 16", the upper significant bits of the PCD are not taken in consideration.

Example: "unsigned 8"



Example: "unsigned-32"

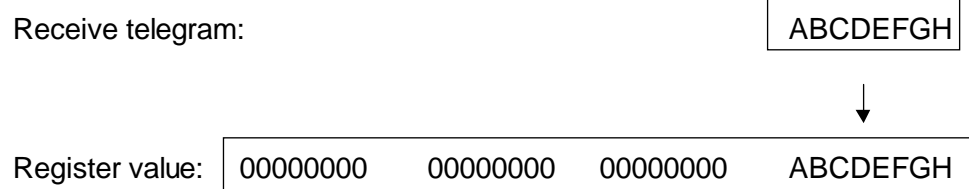
For objects of type "unsigned 32" the MSBit (bit 32) is set low (sign bit).



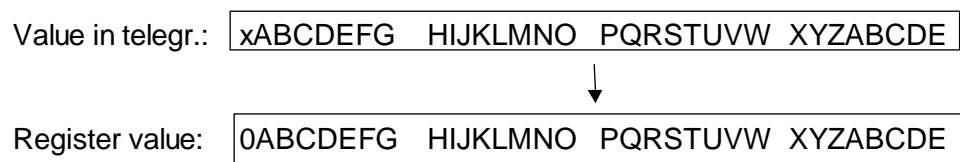
Transfer receive telegram → PCD media (register)

When a received "unsigned 8" or "unsigned 16" telegram is copied to the PCD media, the upper significant bits are set zero.

Example: "integer 8"



Example: "unsigned 32". The sign bit (bit 32) is set low.



- **Floating Point**

Range: ....

Transfer PCD media (register) → send-telegram.

For the transfer the value is converted from the Motorola format to the IEEE format.

Register value:

Floating point in Motorola format

↓ conversion Motorola → IEEE format

Transfer telegram:

floating point in IEEE format

Transfer receive telegram → PCD media (register)

A received floating point telegram is converted from the IEEE to the Motorola format.

Received telegram:

Floating point in IEEE format

↓ conversion IEEE → Motorola format

Media value:

Floating point in Motorola format

## • OCTET-String

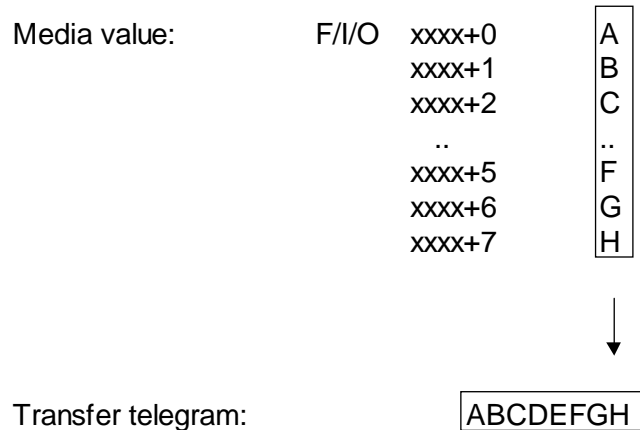
Range: .....

Transfer PCD media (register) → send-telegram.

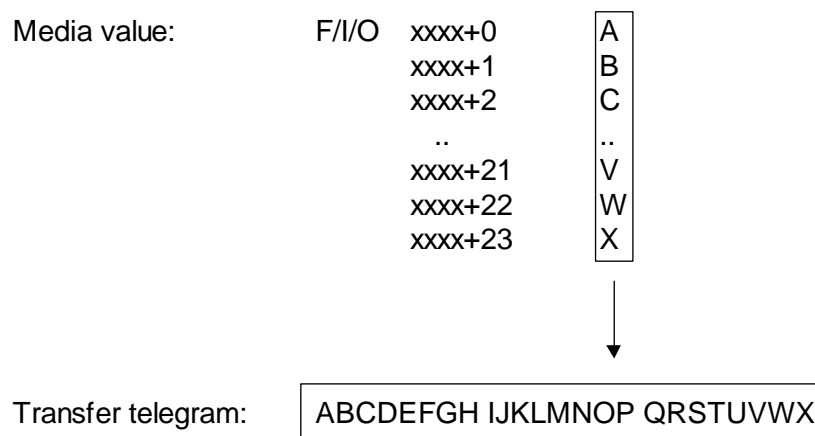
At least 1 string with 8 bit (1 byte) or a number of n strings (n bytes) will be transferred.

Near by the media element with the lowest address is transferred as MSBit in the telegram. The element with the highest address will be transferred as LSBit. This corresponds with the "sensor/actuator" profile.

Example: Octet string (1)



Example: Octet string (3)

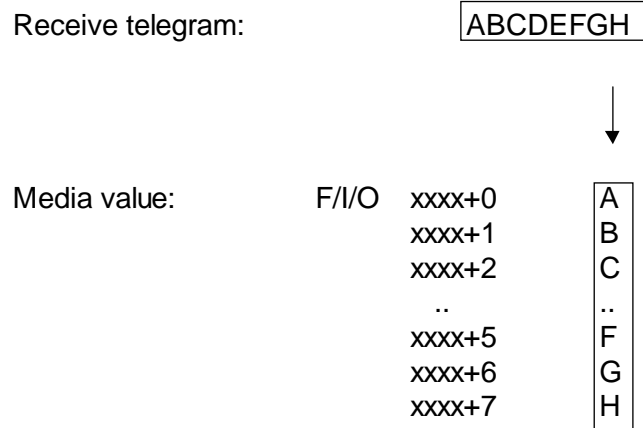


Transfer receive telegram → PCD media (register):

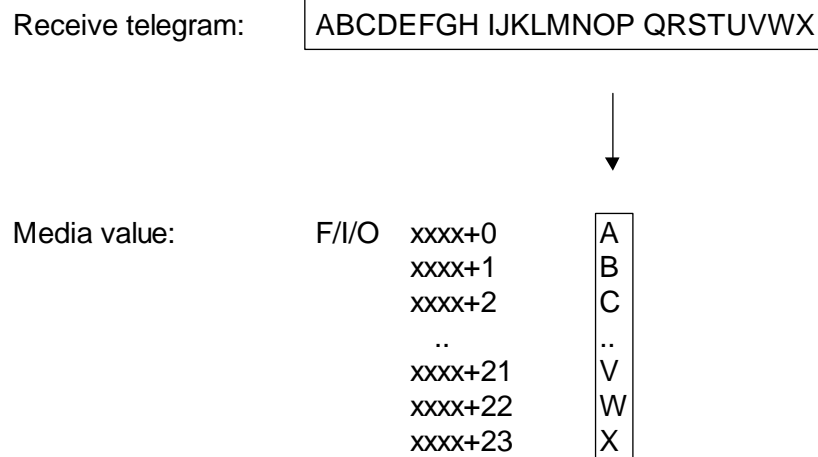
At least 1 string with 8 bit (1 byte) or a number of n strings (n bytes) will be transferred.

Near by the MSBit is transferred in the telegram to lowest media address. The LSBit is transferred to the highest media address. This corresponds with the "sensor/actuator" profile.

Example: Octet string (1)



Example: Octet string (3)



## • BIT-String

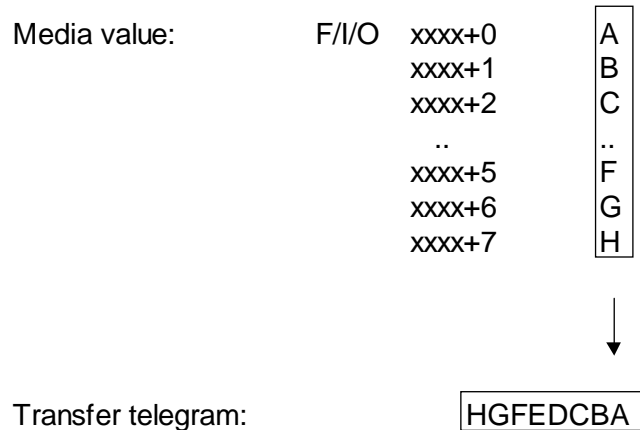
Range: .....

Transfer PCD media (register) → send-telegram:

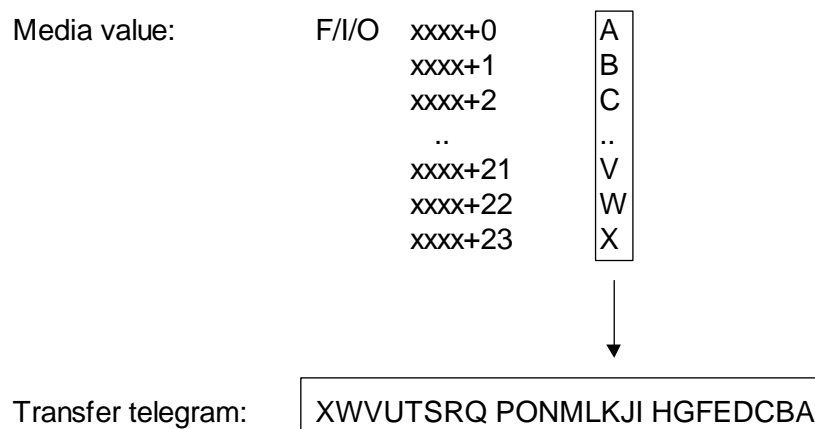
At least 1 string with 8 bit (1 byte) or a number of n strings (n bytes) will be transferred.

Near by the media element with the lowest address is transferred as LSBit in the telegram. The element with the highest address is transferred as MSBit. This corresponds with the DIN 19245, part 2 standard.

Example: Bit string (1)



Example: Bit string (3)

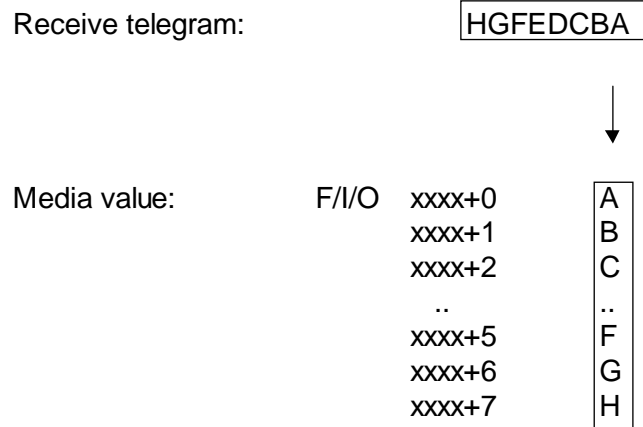


Transfer receive telegram → PCD media (register):

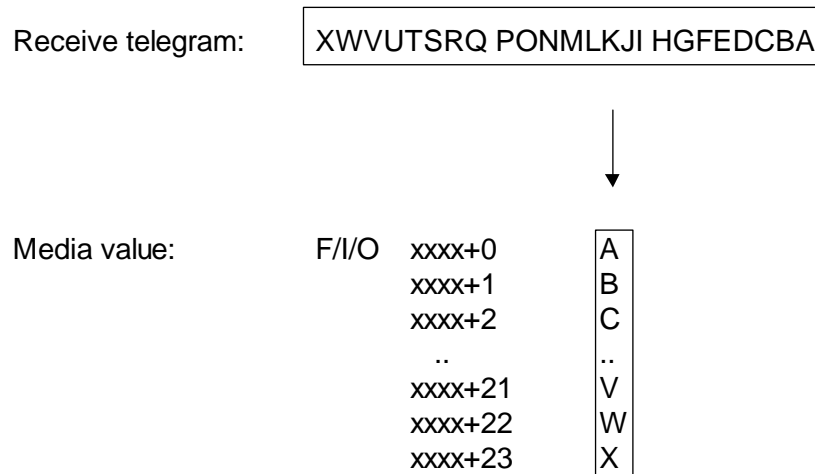
At least 1 string with 8 bit (1 byte) or a number of n strings (n bytes) will be transferred.

Near by the MSBit is transferred in the telegram to highest media address. The LSBit will be transferred to the lowest media address. This corresponds with the DIN 19245, part 2 standard.

Example: Bit string (1)



Example: Bit string (3)



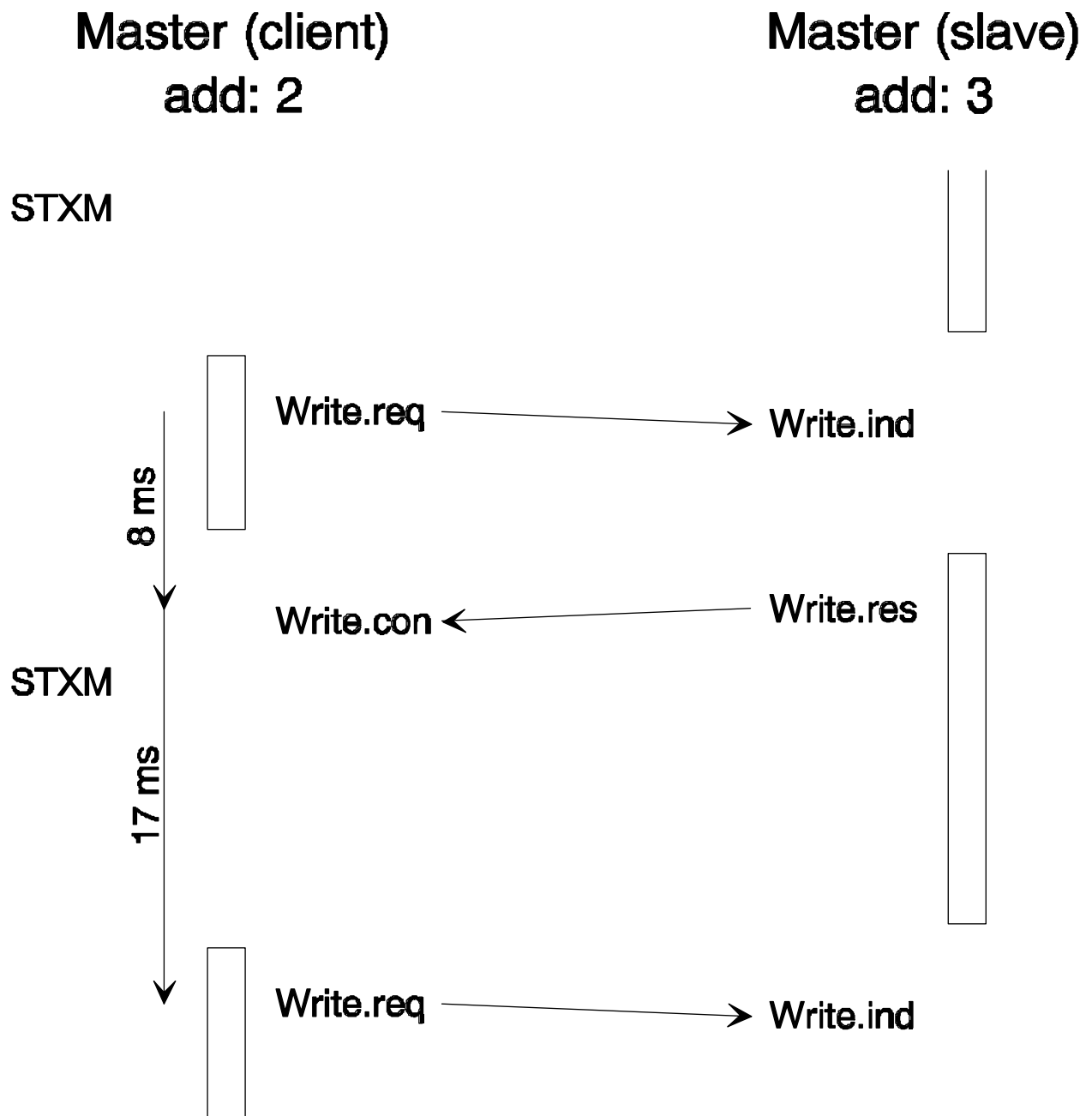


## Appendix B: Effect of bus parameters on transmission times

---

### Master -> Master

default parameters



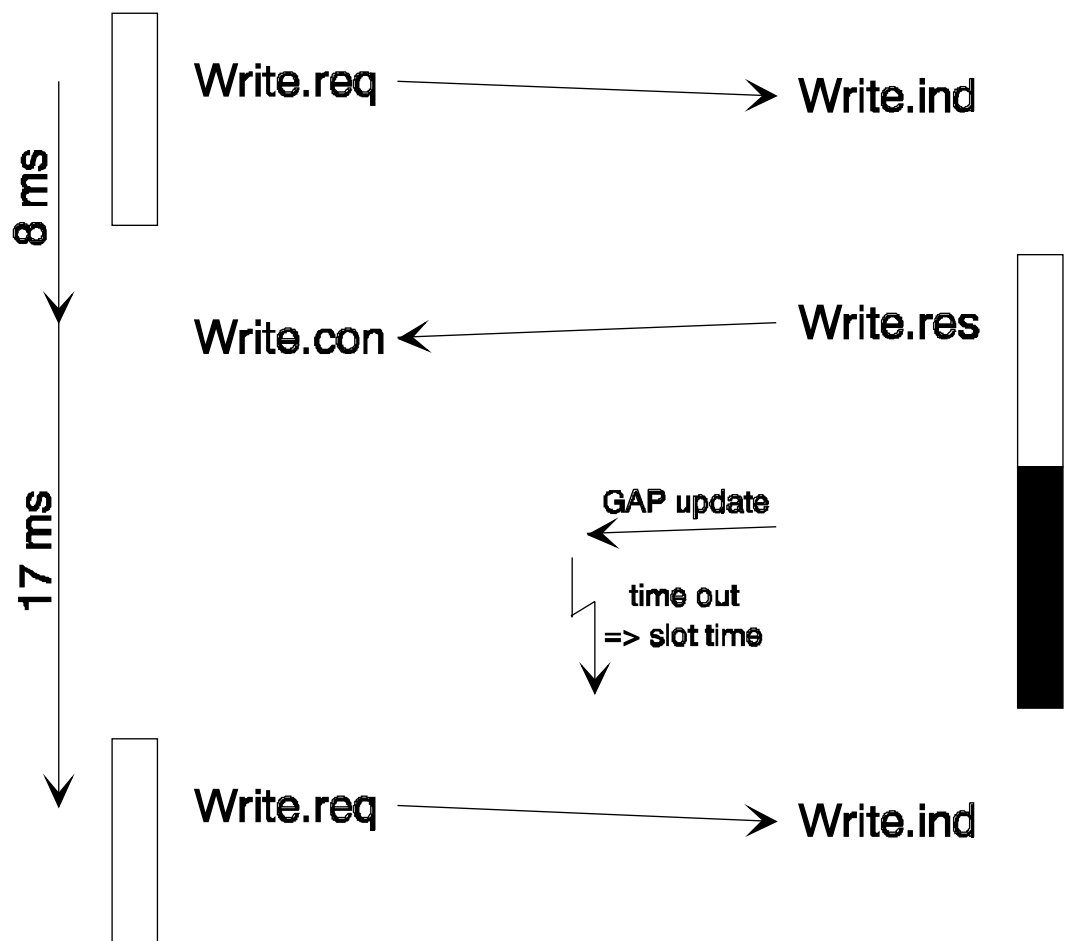
**Master -> Master**

default parameters

**Master (client)  
add: 2**

STXM

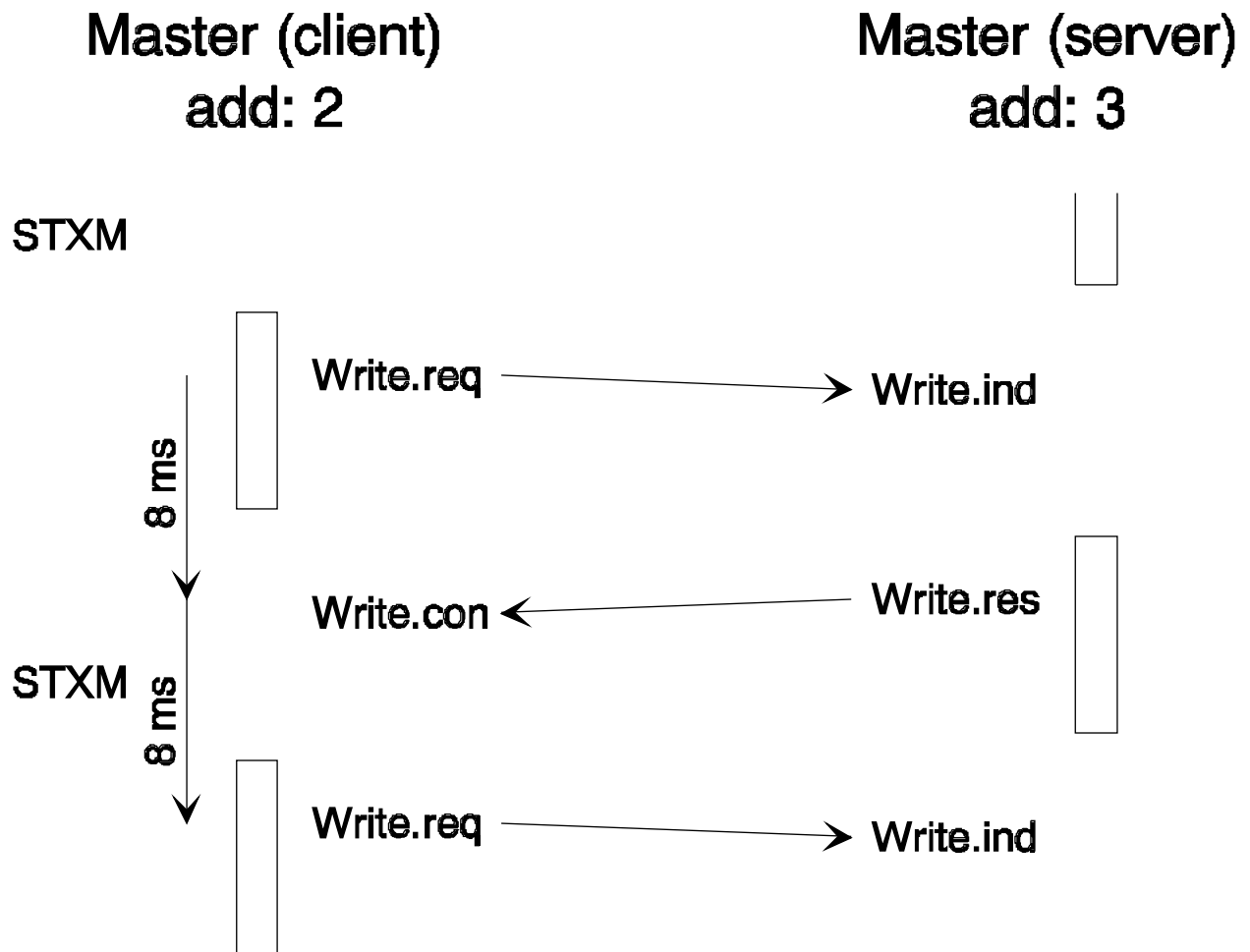
STXM



**Master -> Master**

highest station add: 3

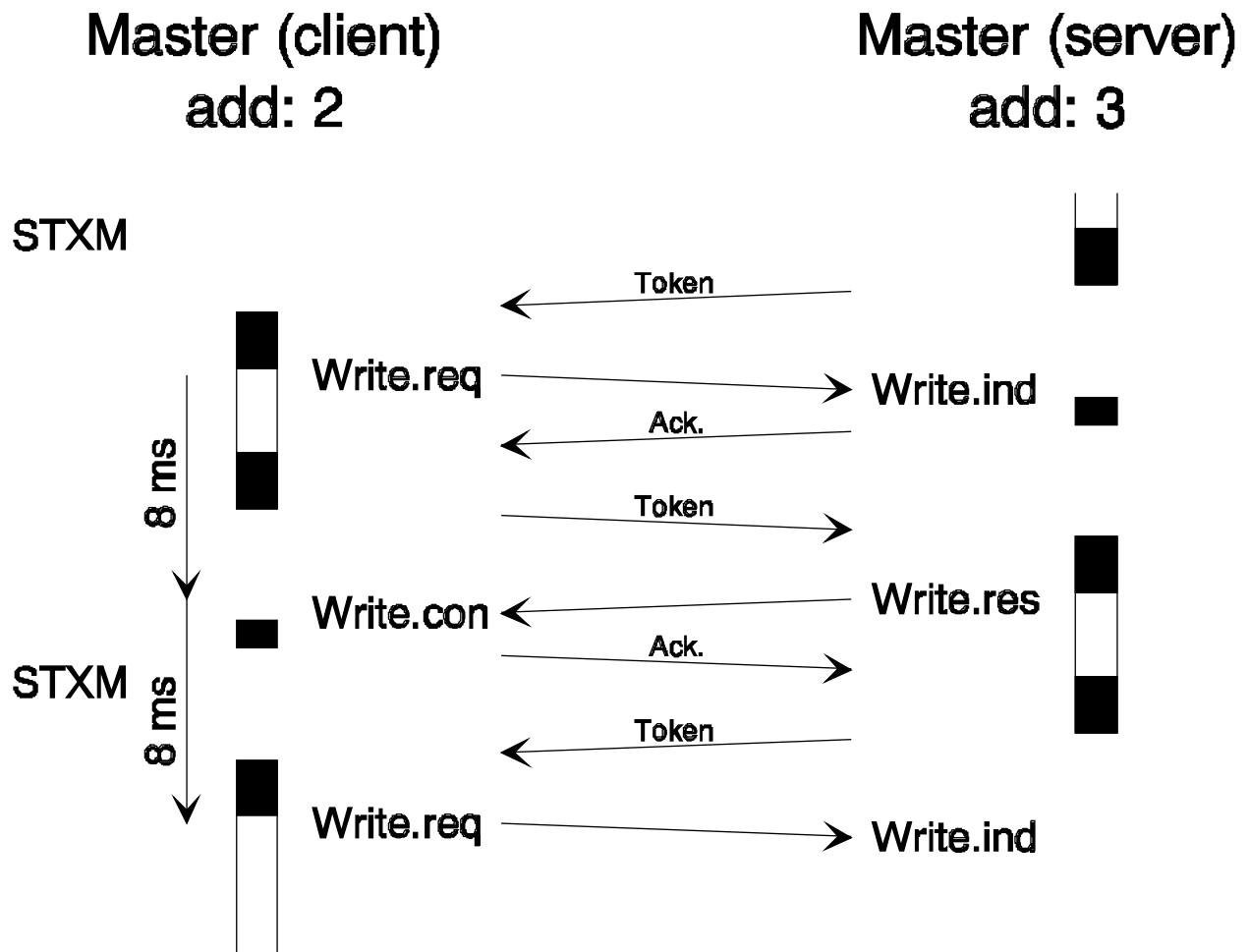
gap update: 99



**Master -> Master**

highest station add: 3

gap update: 99



**Master -> Master**

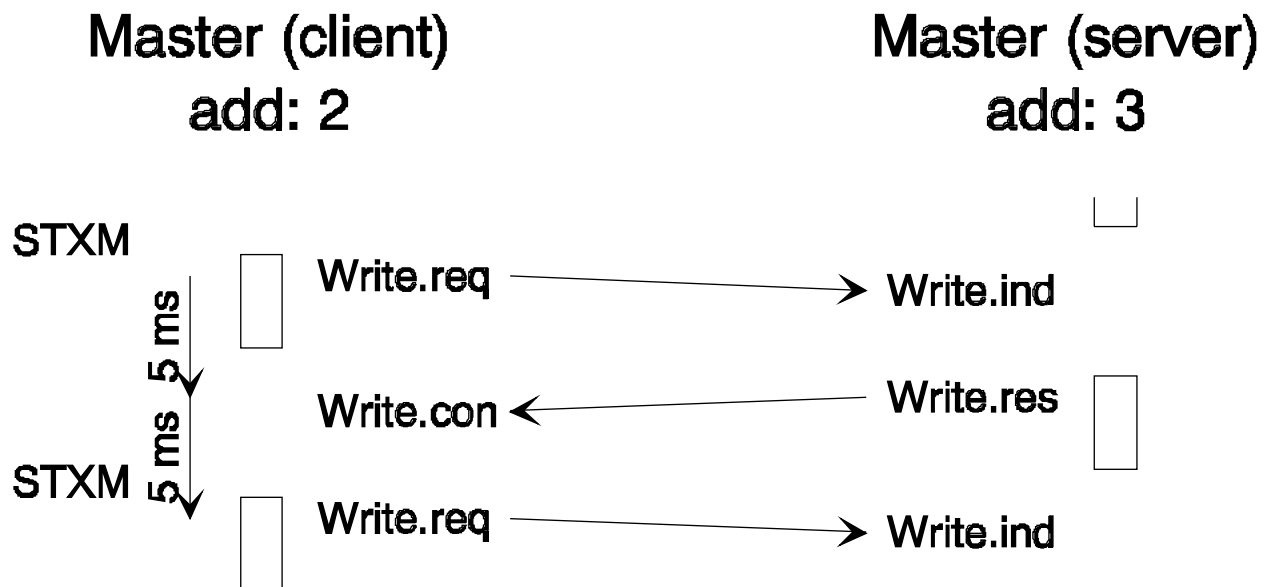
highest station add: 3

gap update: 99

TSDR min: 50

TSDR max: 100

(slot time: 200)



**Master -> Master**

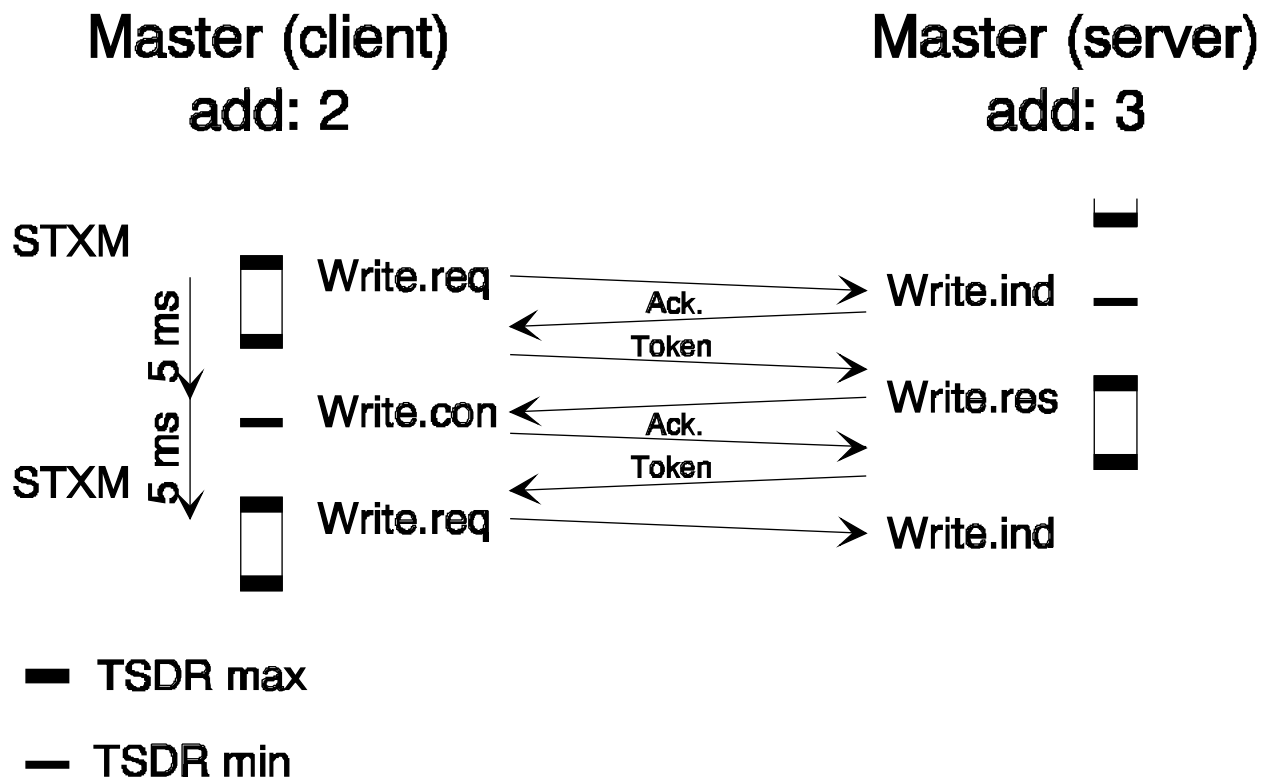
highest station add: 3

gap update: 99

TSDR min: 50

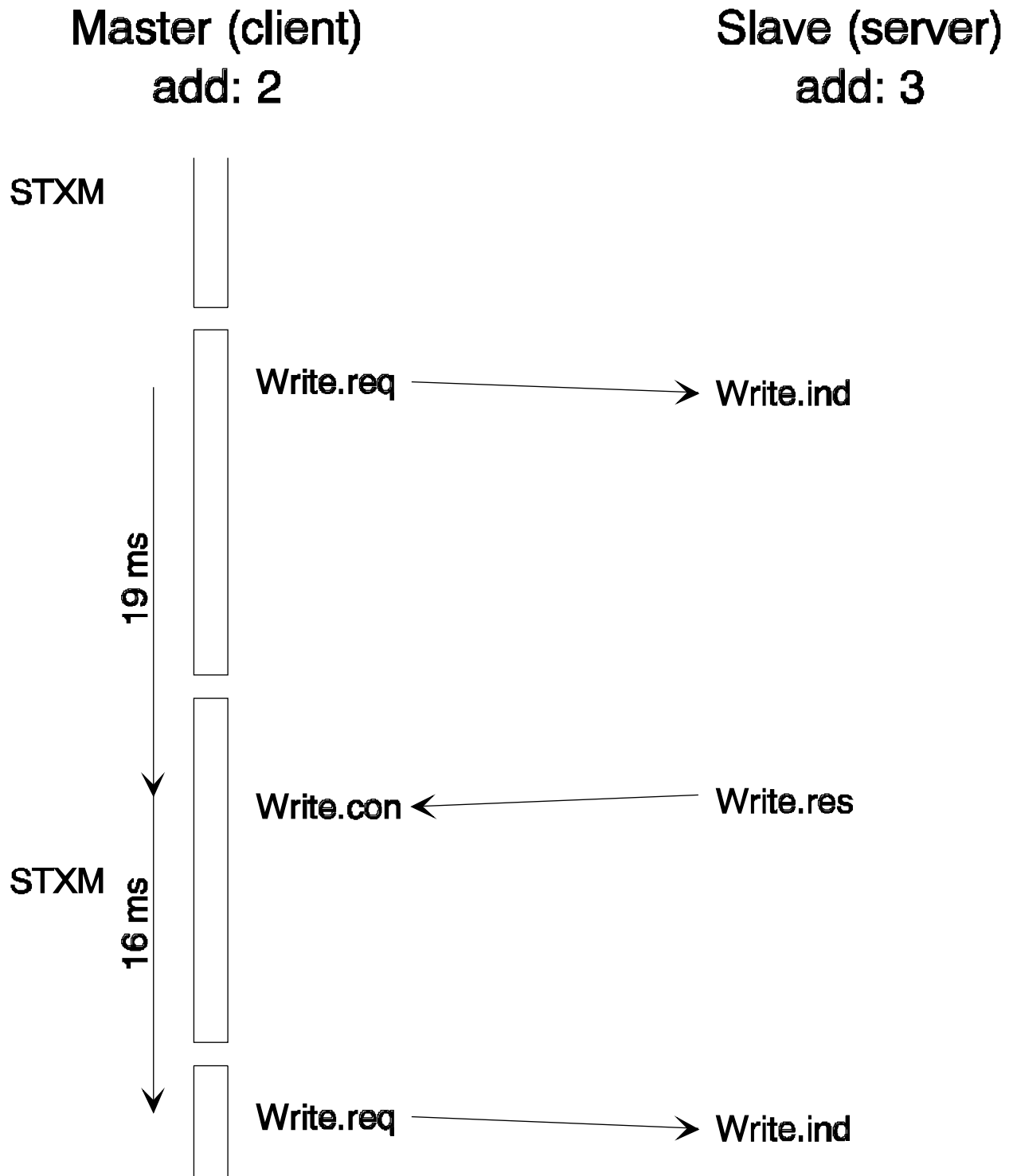
TSDR max: 100

(slot time: 200)



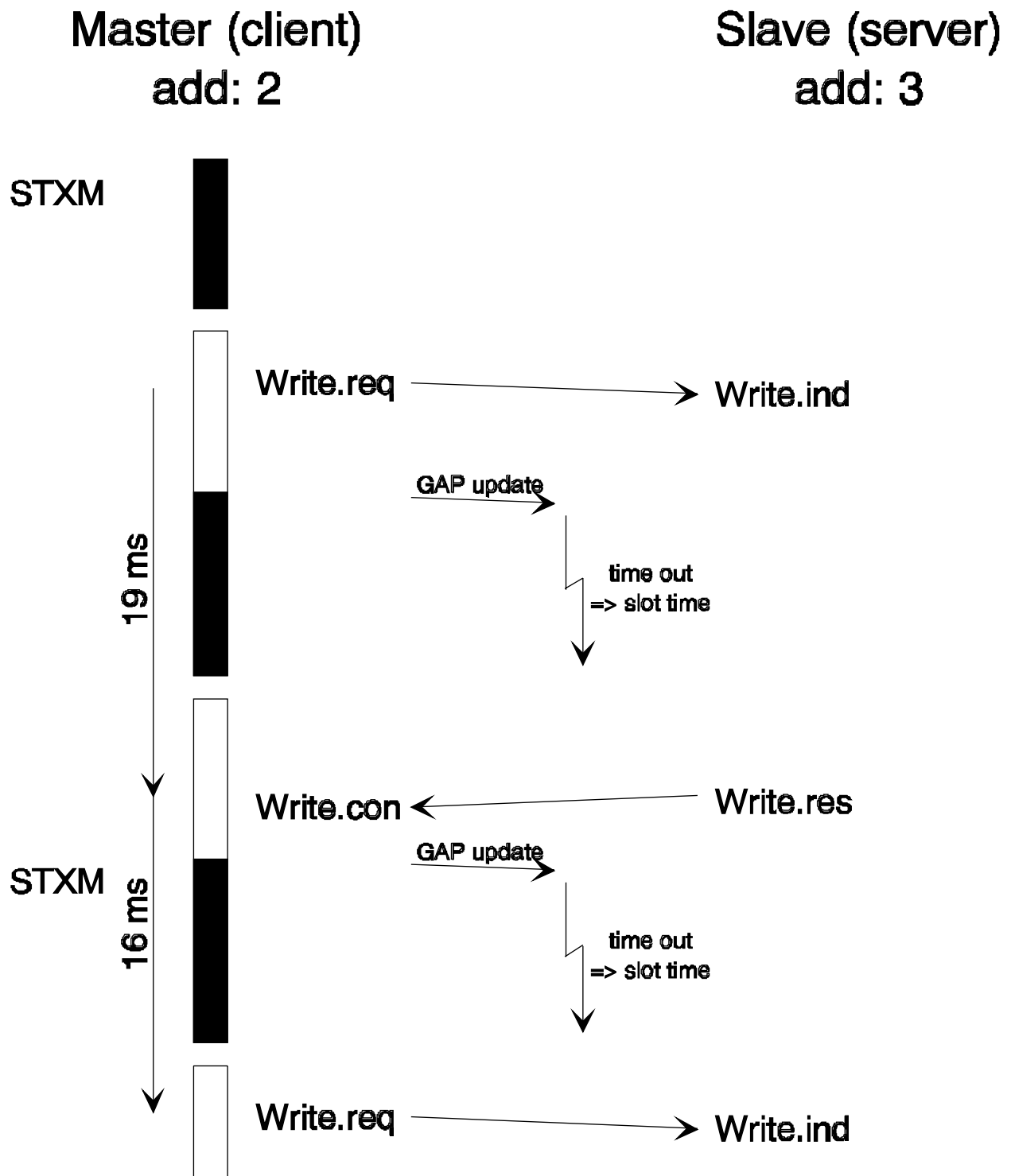
**Master -> Slave**

default parameters



Master -> Slave

default parameters

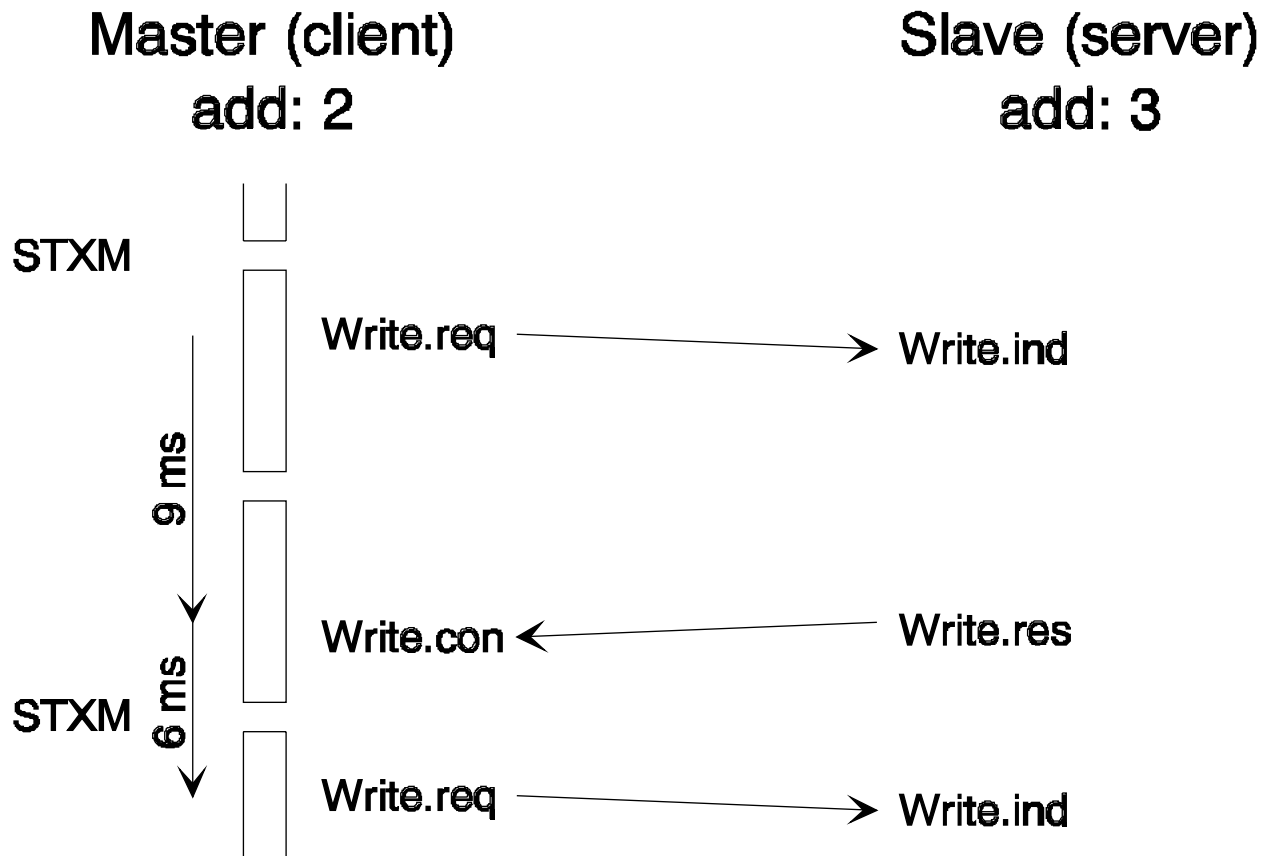




**Master -> Slave**

Highest station add: 2

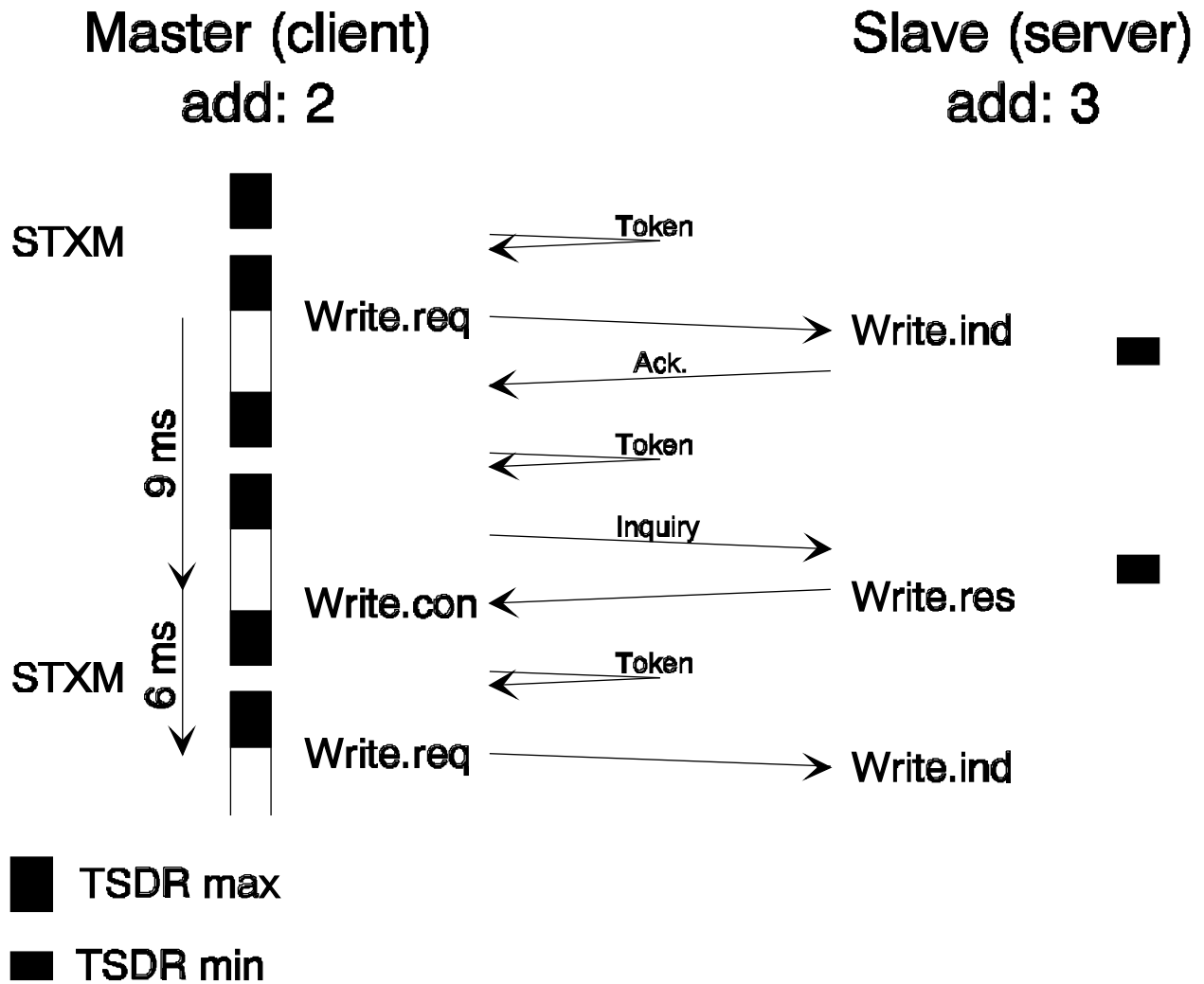
gap update: 99



**Master -> Slave**

Highest station add: 2

gap update: 99



**Master -> Slave**

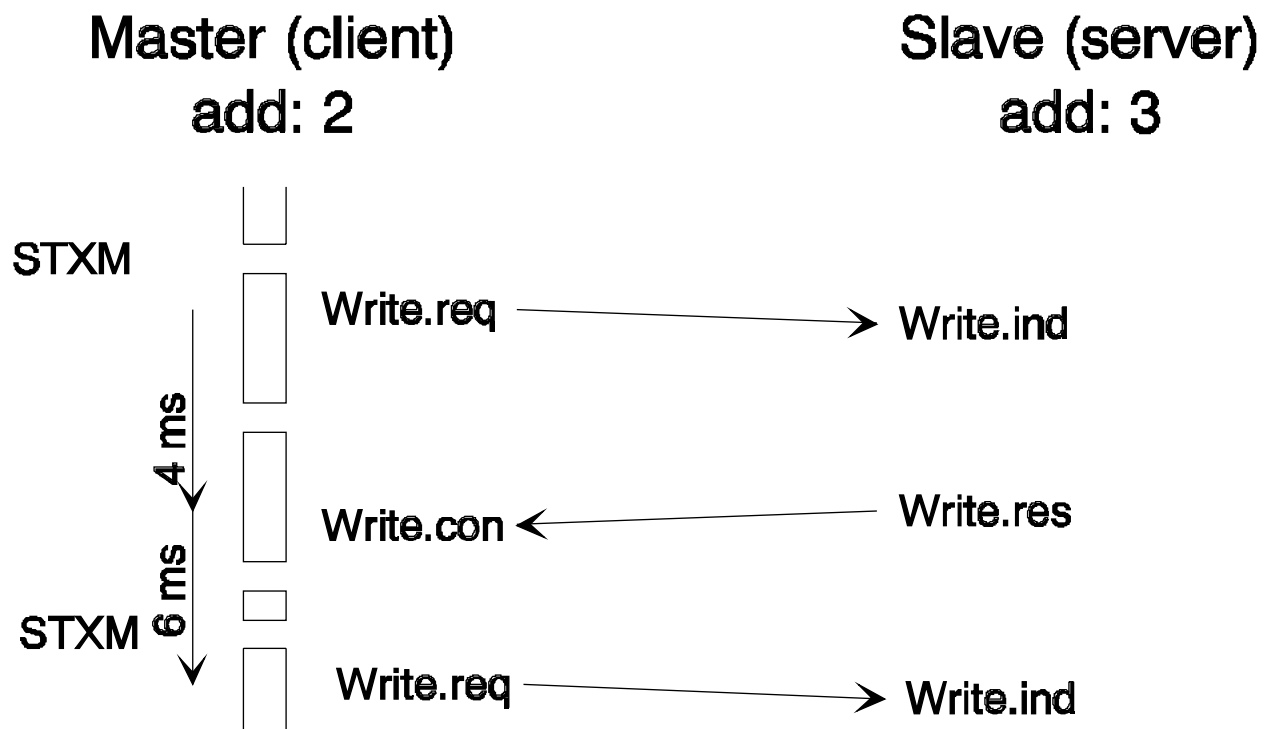
Highest station add: 2

gap update: 99

TSDR min: 50

TSDR max: 100

(slot time: 200)



**Master -> Slave**

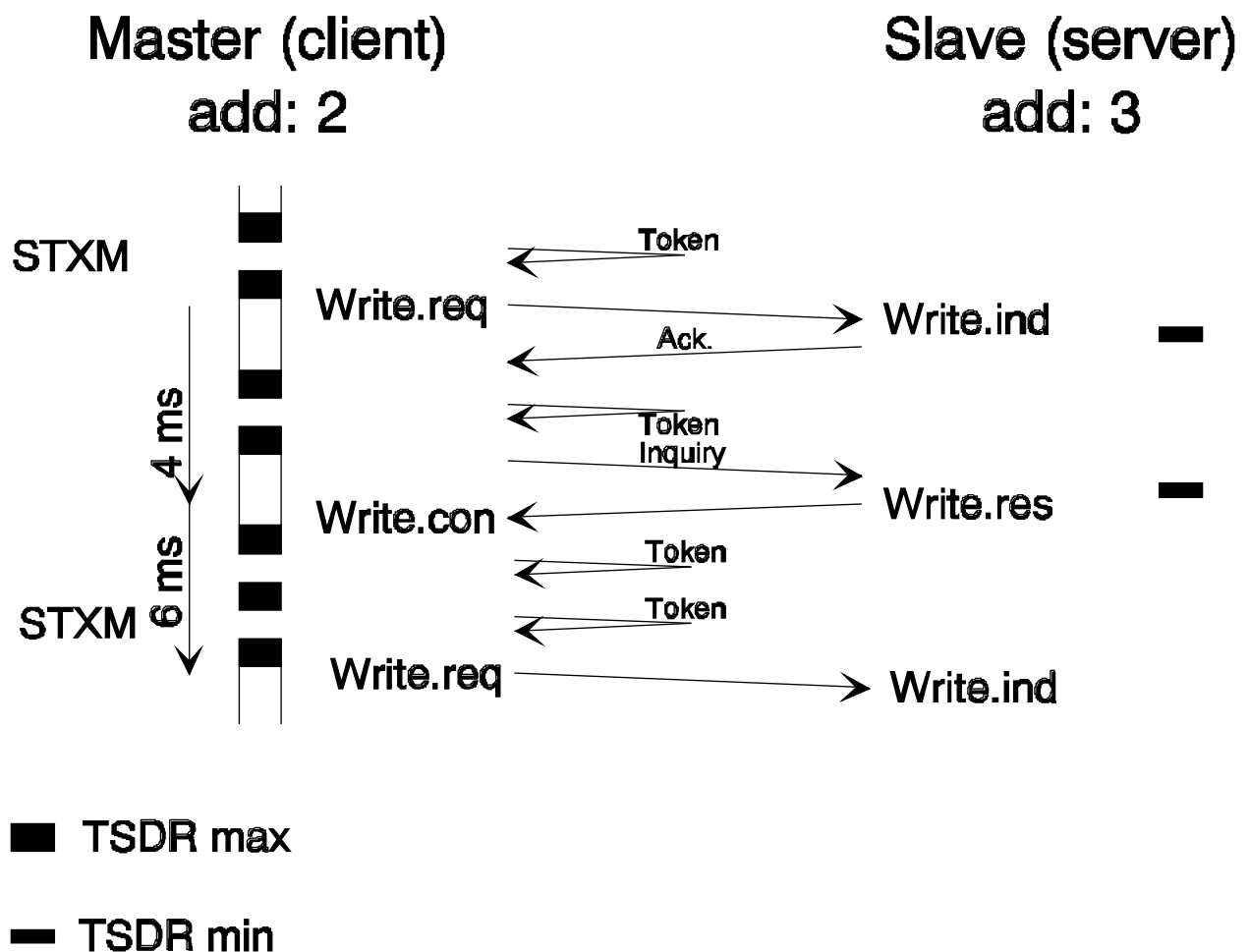
Highest station add: 2

gap update: 99

TSDR min: 50

TSDR max: 100

(slot time: 200)



## Appendix C: PROFIBUS services

### • PROFIBUS services

Services	Slave Cli Serv		Master Cli Serv		cfd/ unc
Initiate	-	M	M	M	cfd
Abort	O	M	M	M	unc
Reject	M	M	M	M	unc
Status	-	M	M	M	cfd
Unsolicited-Status	O	O	O	O	unc
Identify	-	M	M	M	cfd
Get-OV	-	M	M	M	cfd
Initiate-Put-OV	-	O	O	O	cfd
Put-OV	-	O	O	O	cfd
Terminate-Put-OV	-	O	O	O	cfd
Initiate-Download-Sequence	-	-	O	O	cfd
Download-Segment	-	-	O	O	cfd
Terminate-Download-Sequence	-	-	O	O	cfd
Request-Domain-Download	-	-	O	O	cfd
Initiate-Upload-Sequence	-	O	O	O	cfd
Upload-Segment	-	O	O	O	cfd
Terminate-Upload-Sequence	-	O	O	O	cfd
Request-Domain-Upload	-	-	O	O	cfd
Create-Program-Invocation	-	O	O	O	cfd
Delete-Program-Invocation	-	O	O	O	cfd
Start	-	O	O	O	cfd
Stop	-	O	O	O	cfd
Resume	-	O	O	O	cfd
Reset	-	O	O	O	cfd
Kill	-	O	O	O	cfd
Read	-	O	O	O	cfd
Write	-	O	O	O	cfd
Read-With-Type	-	O	O	O	cfd
Write-With-Type	-	O	O	O	cfd
Phys-Read	-	O	O	O	cfd
Phys-Write	-	O	O	O	cfd
Information-Report	O	O	O	O	unc
Information-Report-With-Type	O	O	O	O	unc
Define-Variable-List	-	O	O	O	cfd
Delete-Variable-List	-	O	O	O	cfd
Event-Notification	O	O	O	O	unc
Event-Notification-With-Type	O	O	O	O	unc
Acknowledge-Event-Notification	-	O	O	O	cfd
Alter-Event-Condition-Monitoring	-	O	O	O	cfd

Cli: Client (Dienstanforderer)

Serv: Server (Diensterbringer)

cfd: Confirmed-Service

unc: Unconfirmed-Service

M: Mandatory (Pflicht-Dienst)

O: Optional

## • Coding of services

The PROFIBUS standard provides the following coding for the FMS services.

A difference is drawn between:

**Request and Confirmation:** Services provided by a station as client.

**Indication and Response:** Services provided by a station as server.

### Indication and Response

24	<input type="checkbox"/>	Get-OV (lang)
25	<input type="checkbox"/>	Unsolicited Status
26	<input type="checkbox"/>	Put-OV
27	<input type="checkbox"/>	Download (Initiate, Terminate)
28	<input type="checkbox"/>	Upload (Initiate, Terminate)
29	<input type="checkbox"/>	Request-Domain-Download
30	<input type="checkbox"/>	Request Domain Upload
31	<input type="checkbox"/>	Program Invokation (Create, Delete)
32	<input type="checkbox"/>	Start, Stop, Resume, Reset
33	<input type="checkbox"/>	Kill
34	<input type="checkbox"/>	Read
35	<input type="checkbox"/>	Write
36	<input type="checkbox"/>	Read with Type
37	<input type="checkbox"/>	Write with Type
38	<input type="checkbox"/>	Physical Read
39	<input type="checkbox"/>	Physical Write
40	<input type="checkbox"/>	Information Report
41	<input type="checkbox"/>	Information Report with Type
42	<input type="checkbox"/>	Variable List (Define, Delete)
43	<input type="checkbox"/>	Event Notification
44	<input type="checkbox"/>	Event Notification with Type
45	<input type="checkbox"/>	Acknowledge Event Notification
46	<input type="checkbox"/>	Alter Event Condition Monitoring
47	<input type="checkbox"/>	Adressierung mit Namen
Bits		

### Request and Confirmation

0	<input type="checkbox"/>	Get-OV (lang)
1	<input type="checkbox"/>	Unsolicited Status
2	<input type="checkbox"/>	Put-OV
3	<input type="checkbox"/>	Download (Initiate, Terminate)
4	<input type="checkbox"/>	Upload (Initiate, Terminate)
5	<input type="checkbox"/>	Request-Domain-Download
6	<input type="checkbox"/>	Request Domain Upload
7	<input type="checkbox"/>	Program Invokation (Create, Delete)
8	<input type="checkbox"/>	Start, Stop, Resume, Reset
9	<input type="checkbox"/>	Kill
10	<input type="checkbox"/>	Read
11	<input type="checkbox"/>	Write
12	<input type="checkbox"/>	Read with Type
13	<input type="checkbox"/>	Write with Type
14	<input type="checkbox"/>	Physical Read
15	<input type="checkbox"/>	Physical Write
16	<input type="checkbox"/>	Information-Report
17	<input type="checkbox"/>	Information-Report-with-Type
18	<input type="checkbox"/>	Variable List (Define, Delete)
19	<input type="checkbox"/>	Event-Notification
20	<input type="checkbox"/>	Event-Notification-with-Type
21	<input type="checkbox"/>	Acknowledge-Event-Notification
22	<input type="checkbox"/>	Alter-Event-Condition-Monitoring
23	<input type="checkbox"/>	Adressierung mit Namen
Bits		

Examples:

- Coding of services for a SAIA<sup>®</sup> PCD with a master-master connection:

00 30 00 80 30 00

- Coding of services for a SAIA<sup>®</sup> PCD with a master-slave connection:

00 30 00 00 00 00  
Master station

00 00 00 80 30 00  
Slave station

## • Context management services

The context includes all the arrangements for a communications link. Context management services are required for initialization and release of a communications connection. Inadmissible services are rejected.

### 1. Initiate

With this service a connection between two communications partners is initialized. This communications link must have been entered in the communications link list (KBL) before execution of the initiate service. On initialization, agreements are made on PDU length (message length) and the services which can be invoked via this connection are named. The current version numbers of the object register are also transferred.

Parameter Name	.req	.ind	.res	.con
<b>Argument</b>	M	M=		
Version-OV *	M	M=		
Profile-Name *	M	M=		
Access-Protection-Supported *	M	M=		
Password *	M	M=		
Access-Groups *	M	M=		
Max-PDU-Sending-High-Prio *				
Max-PDU-Sending-Low-Prio *				
Max-PDU-Receiving-High-Prio *				
Max-PDU-Receiving-Low-Prio *				
FMS-Services-Supported *				
<b>Result (+)</b>			S	S =
Version-OV **			M	M=
Profile-Name **			M	M=
Access-Production-Supported **			M	M=
Password **			M	M=
Access-Groups **			M	M=
<b>Result (-)</b>			S	S =
Error-Type			M	M=
Max-PDU-Sending-High-Prio **				M
Max-PDU-Sending-Low-Prio **				M
Max-PDU-Receiving-High-Prio **				M
Max-PDU-Receiving-Low-Prio **				M
FMS-Services-Supported **				M

\* → parameter from service requester  
 \*\* → parameter from service supplier

The first six parameters of the initiate service must be supplied by the requester at the interface ALI/FMS. The rest are generated by the FMS (Fieldbus Message Specification).

**Version OV: (OV → OD = Object Dictionary)**

Version identification of the object register.

**Profile name:**

Profile name of visible string type

**Access protection supported:**

Indicates whether access protection is supported.

**Password:**

This parameter contains the password which applies to all access to the remote station's objects via this communications link.

**Access groups:**

Contains an assignment to certain access groups. communicating station's objects via this communications link.

**Max PDU Sending High Prio:**

Maximum possible length of FMS PDU in high priority transmission.

**Max PDU Sending Low Prio:**

Maximum possible length of FMS PDU in low priority transmission.

**Max PDU Receiving High Prio:**

Maximum possible length of FMS PDU in high priority reception.

**Max PDU Receiving Low Prio :**

Maximum possible length of FMS PDU in low priority reception.

**FMS Services Supported:**

This parameter provides information on which services can be executed by the sender of this initial service.



## 2. Abort

With this service, an existing communications link between two communicating stations is released. The connection can be aborted both by the service requester (client) and by the service provider (server).

Parameter Name	.req	.ind
Argument	M	M=
Locally-Generated		M
Abort-Identifier		M=
Reason-Code		M=
Abort-Detail	M	C

### Locally Generated:

This parameter indicates whether the abort was triggered locally or by a remote communicating station.

### Abort Identifier:

This parameter indicates the layer (0 - USER, 1 - FMS, 2 - LLI, 3 - FDL) in which the cause of the aborted connection has been identified.

### Reason Code:

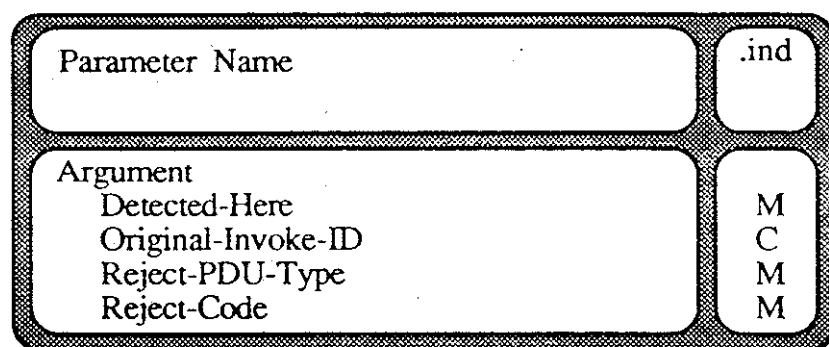
This parameter indicates the reason for the abort.

### Abort Detail:

This byte string type parameter contains additional information on the reason for the abort. With application process errors, the meaning is defined by the profile.

### 3. Reject

With the reject service, the FMS layer rejects an inadmissible PDU (message). The reject service is generated whenever the job given by the service requester cannot be handled by the virtual field device or the communications process. For example, inadmissible messages include those containing a requested service which is not executable or not allowed. Another reason for generation of a reject can be excessive message length, so that it no longer fits in the buffer provided. Incorrect execution of the read service, e.g. because of incorrect index specification, only leads to a negative response.



#### Detected Here:

This parameter indicates whether the error was detected in the local device or in the remote communicating station.

#### Original Invoke ID:

The job number of the rejected PDU.

#### Reject PDU Type:

This parameter indicates the type of the rejected PDU.

Types are differentiated as follows:

- 1 Confirmed Request PDU
- 2 Confirmed Response PDU
- 3 Unconfirmed PDU
- 4 Unrecognized PDU type

#### Reject Code:

Reason for rejection of the PDU.

#### **4. Error codes**

##### **FMS reject codes**

0	Other error
1	Invoke ID already exists
2	Max. number of parallel unconfirmed services exceeded
3	Service not supported (connection-oriented CREF)
4	Service not supported (connectionless CREF)
5	PDU length exceeded
6	Service primitive not allowed
41h	Max. number of parallel, unconfirmed services exceeded

##### **FMS reject PDU types**

1	Confirmed Request DDU
2	Confirmed Response PDU
3	Unconfirmed PDU
4	Unrecognized PDU type

##### **Initiate Error codes**

0	Other error
3	Version OD incompatible
4	FMS user rejects initiate
5	Password error
6	Profile number

##### **Initiate Error Codes (FMS and FMA7)**

1	Maximum PDU length exceeded
2	Features not supported

**FMS Abort Reason Codes**

- 0 FMS-CRL error
- 1 User error
- 2 FMS-PDU error
- 3 LLI service primitive not allowed
- 4 LLI error
- 5 PDU length too great
- 6 Service not supported
- 7 Invoke-ID does not exist
- 8 Max. number of parallel services exceeded
- 9 Status conflict in FMS
- 10 Service error (response does not agree with request)
- 11 Invoke-ID already exists
- 12 FMS-Disable: FMA7 loads the CRL

**Abort details for user errors**

- 1 CREF not defined
- 2 Illegal service code
- 3 Optional string with length 0
- 4 Name addressing with unknown access code
- 6 List is empty
- 7 Error in integer variable
- 8 Illegal error class
- 9 Illegal PI state
- 10 Illegal type description
- 11 Link has not been initiated
- 12 Initiate-Response: link is not at initiate stage
- 13 Initiate-Request: link is not in CLOSED state
- 14 Illegal LSAP
- 15 Characteristic length <> 40 (for management only)

**FMA7 Abort Reason Codes**

- 0 FMA7-CRL error
- 1 User error
- 2 FMA7-PDU error
- 3 LLI service primitive not allowed
- 4 LLI error
- 5 PDU length too great
- 6 Service not supported
- 7 Response error (request does not exist)
- 8 Max. number of parallel services exceeded
- 9 Status conflict in FMA7
- 10 Service error (response does not agree with request)

<b>Instance Identifier</b>		<b>User Abort Reason Codes</b>	
0	User	0	Disconnect
1	LLI user (FMS or FMA7)	1	Version OD incompatible
2	LLI	2	Password error
3	FDL/PHY	3	Profile number incompatible
		4	Limited service permitted
		5	Put-OD Loading-Interacting

### **LLI Abort Reason Codes**

0	LLI-LLI context check negative
1	Illegal LLI-PCD received in initiate phase
2	Illegal LLI-PDU received in the data transfer phase
3	Unknown or incorrect LLI-PDU received
4	DAT_ACK_PDU received and SAC = 0
5	Number of parallel services exceeded (indication by FDL)
6	Invoke-ID unknown
7	Priority error
8	Local error on remote station
9	Timer 1 finished (initiation monitoring)
10	Timer 3 finished (connection monitoring)
11	E Timer finished
12	Error during LSAP enable (AD contains status)
13	Illegal FDL primitive in initiate phase
14	Illegal FDL primitive in data transfer phase
15	Unknown FDL primitive
16	Unknown LLI primitive
17	Illegal LLI primitive in initiate phase
18	Illegal LLI primitive in data transfer phase
19	CRL entry not ok
20	Conflict in initiate phase
21	Run error with cyclic data transfer
22	Number of parallel services exceeded (request by FMS)
23	CRL is downloaded from FMA7, LLI disabled
24	Confirm / indication mode error
25	Illegal FMA 1/2 primitive received
26	Illegal FMS primitive on cyclic connection
27	FMS-PDU length on cyclic connection exceeded

**Notes :**

## **Appendix D: Abbreviations and terminology**

---

### **Abort**

Service which aborts a logical communication channel.

→ Service, Initiate

### **Array**

Data type. Field of structured variables. Sequence of simple variables of the same data type.

→ Record

### **Boolean**

Data type. Can accept the values FALSE (0) or TRUE (255).

### **Broadcast**

A message which is sent simultaneously to all stations in the network.

Not supported by SAIA devices.

→ Multicast

### **Client**

Service requester (generally the master). In the communication sense, an application process which assigns tasks, in order to use the function (objects) of a remote application process with the help of services.

→ Server

### Communication channel

Virtual transmission route for a virtual connection between two stations.

A channel contains the following information:

- remote station address
- **Service Access Points** used (LSAP and RSAP)
- connection type (cyclic, acyclic, etc.)
- list of services allowed (e.g. Read/Write, etc.)
- size of data buffer required (PDU length)

The following rules apply to the communication channel:

- A single channel can only be used for one defined connection.
- With an open connection, a channel can be used for connections with several remote devices (often applied to non-configurable slaves).
- For a single station, up to 90 channels (10 for the PCD7.F700) can be defined and run in parallel.

### Communication Object

Data block for communication. PROFIBUS provides for object-oriented programming, so that even the most varied devices can exchange data mutually.

Before data can be exchanged, the system must be advised of the communication objects and their meaning or structure. For all stations, object definition takes place in the configurator.

Each station keeps a complete or partial copy of the objects in its remote station.

The following rules apply:

- The objects of a station are basically accessible to all stations on the bus via the communication channels (access rights can be defined).
- During transmission of objects, the source and source object must have compatible data structures and data formats.

→ Object dictionary

### Confirmation

Confirmation of receipt of a request to the local server, at a previously started request.

→ Service primitive, Request, Indication



**Connection, Connection oriented communication relationship**

Specification of a logical communication channel between two stations (point-to-point connection).

**Connection monitoring**

Continuous control of the operating capability of a logical connection.

**Context Management**

Management services (Initiate, Abort, Reject)

→ Service

**CREF, Communication Reference, Communication Channel Number**

Reference number for the logical addressing (identification) of connections.

**CRL, Communication Relationship List**

List containing data for the projected connections.

**Default Value**

Standard value used when no other value is explicitly entered.

**DP**

Decentralized peripheral

**Event Management Services**

Services which process event-dependent messages (object type: event).

Not supported by SAIA devices.

**FDL, Fieldbus Data Link**

Name for the data protection layer (2) in PROFIBUS.

**FMS, Fieldbus Message Specification**

Defines the application services in PROFIBUS.

**GAP**

The address range between the local station address of an active station up to the next active station in the logical ring.

**Identify**

Service for requesting the device identification of another station on the bus.

Supported by SAIA devices only as servers.

**Indication**

Indicates the arrival of a service request at the remote server.

→ Service primitive, Confirmation

**Lower Layer Interface, LLI**

Is part of application layer 7 in PROFIBUS FMS

**Master**

Active station, which participates in token passing.

→ Slave, Token passing

**MMAC**

Master-master connection for acyclic data transfer

**MSAC**

Master-slave connection for acyclic data transfer

**MSAC\_SI**

Master-slave connection for acyclic data transfer with slave initiative

**MSCY**

Master-slave connection for cyclic data transfer

**MSCY\_SI**

Master-slave connection for cyclic data transfer with slave initiative.

Not supported by SAIA devices.

**Multicast**

A message which is sent simultaneously to several stations in the network.

Not supported by SAIA devices.

→ Broadcast

**Object Dictionary, OD**

Dictionary of all attainable (defined) communication objects. It contains, e.g.: details of structure, data type, access rights and object memory address.

**PA**

Process Automation

**Password**

Decimal number, access protection for an object or connection.

**PNO**

→ PROFIBUS user organization.

**Profile**

Selection of functions and performance features within the limits of the overall specification of the PROFIBUS standard.

**RAC**

Receive Acknowledged request Counter.  
Maximum allowed number of unconfirmed, parallel services as receiver.

**RCC**

Receive Confirmed request Counter.  
Indicates the maximum number of confirmed services (requests) the station can accept as a server on this channel, without sending a response.

**Read**

Service for reading an object from another station.  
→ Service, Write

**Record**

Data type. Sequence of simple variables with different data types.  
Not supported by SAIA devices.  
→ Array

**Reject**

Service by which an invalid service request is rejected.  
→ Service

**Repeater**

Bi-directional repeater.

**Request**

Service request by the local user.  
→ Service primitive, Response

**Response**

Response from the remote server to an indication

→ Service primitive, Indication, Request

**SAC**

Send Acknowledged request Counter.

Maximum allowed number of unconfirmed, parallel services as sender.

**SCC**

Send Confirmed request Counter.

Indicates the maximum number of outstanding requests the station can process as a client on this channel.

**Server**

Master or slave. An application process for task execution, which makes available the functionality (objects) of its virtual field device to a client.

**Service**

An object oriented operation; defined under DIN 19245 - parts 1 & 2.

For example, a read or write task to a master or slave is a service.

There is a difference between:

- Confirmed and unconfirmed services.

With confirmed services, execution of the task is accompanied by a confirmation from the server.

With unconfirmed services, there is no confirmation that the task has been executed.

Unconfirmed services (multicast or broadcast telegrams) are not supported by SAIA devices.

- Parallel services

People talk of parallel services on a communication channel when, on a single server at any one time, there are several tasks from one client, or when at any one time a client has started several tasks on a server, which have not yet been executed.

The invoke identification allows differentiation between the individual tasks.

A SAIA<sup>®</sup> PCD can only process parallel services as server.

**Service Access Point, SAP (2..62, 128)**

An SAP is the logical interface between the application layer (FMS layer) and the lower FDL layer (Field Data Link).

A difference is drawn between:

LSAP (Local Service Access Point)

and

RSAP (Remote Service Access Point)

Data is exchanged across these logical interfaces.

A SAP also contains information on the type of connection

- master/slave
- cyclic/acyclic
- etc.

The following rules apply:

- Poll SAP = common SAP for all slave stations (default poll SAP = 128)
- With a defined master-master connection, the same SAP can only be used once for a single connection (channel).
- For an open connection, the same rule applies by analogy as for the communication channel.

**Service primitive**

An abstract resource, which enables the connection between user and the service of a lower layer. A request to execute a service (.req) is made and processed as an indication (.ind). Confirmed services are answered with a response (.res), which is received as a confirmation (.con) by the client.

→ Request, Indication, Response, Confirmation

**Slave**

Passive station, which does not participate in token passing and therefore can only send following a request from a master.

→ Master

**Target Rotation Time**

Token rotation time. As a setpoint, it is a bus parameter projected by the user (TTR). As an observed value, it is the actual time measured during which all active stations held the token once.

→ Token, Token Passing

**Token**

A telegram with predefined bit sequence, by means of which control of the bus system is passed from one station to the next.

**Token Passing**

Bus access control, in which the token telegram is used to pass bus access rights from master to master. Only a station (master) which possesses the token can act on the bus.

→ Token, Master, Slave

**VFD, Virtual Field Device**

Abstract model for describing the data and behaviour of an automation system from the point of view of a remote communicating station.

**Write**

Service for describing the object of another station.

→ Service, Read

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<sup>®</sup> PCD

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

**Your suggestions :**