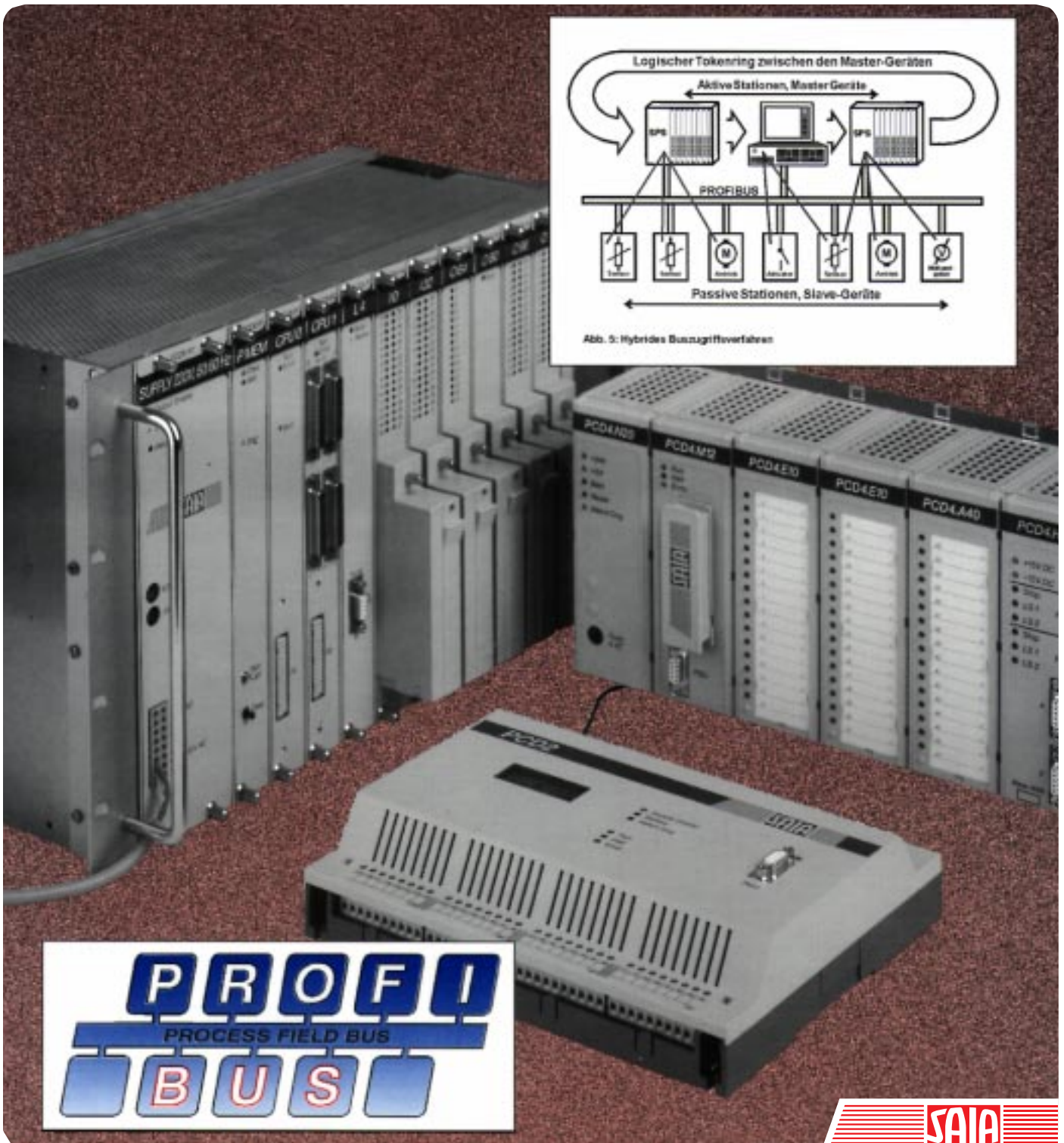


## SAIA® PCD Process Control Devices

## Handbuch PROFIBUS



---

## SAIA-Burgess Gesellschaften

<b>Schweiz</b>	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	<b>Frankreich</b>	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
<b>Deutschland</b>	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	<b>Niederlande</b>	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
<b>Österreich</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>Belgien</b>	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
<b>Italien</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>Ungarn</b>	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

---

## Vertretungen

<b>Gross-britannien</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>Dänemark</b>	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	<b>Spanien</b>	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
<b>Norwegen</b>	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	<b>Tschechische Republik</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>Schweden</b>	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	<b>Polen</b>	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
<b>Suomi/ Finnland</b>	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
<b>Australien</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>Argentinien</b>	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

---

## Kundendienst

<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**

**Handbuch**

**PROFIBUS-FMS**

**mit SAIA® PCD**

SAIA-Burgess Electronics AG 1996 - 1999. Alle Rechte vorbehalten  
Ausgabe 26/742 D2 - 01.1999

Technische Änderungen vorbehalten

# Anpassungen

---

Handbuch: PROFIBUS-FMS mit SAIA® PCD - Ausgabe D2

Datum	Abschnitt	Seite	Beschreibung
08.11.2000	---	---	Kleine Anpassungen für die Support Homepage

# Inhalt

---

	Seite
<b>1. Eigenschaften und Grundlagen</b>	<b>1-1</b>
1.1 Eigenschaften	1-1
1.2 Grundlagen	1-4
1.2.1 PROFIBUS Schicht 1 (Physical Layer)	1-4
1.2.2 PROFIBUS Schicht 2 (Data Link Layer)	1-5
1.2.3 PROFIBUS-FMS	1-7
<b>2. SAIA® PCD-Geräte für das PROFIBUS-Netzwerk</b>	<b>2-1</b>
2.1 Das Prozessormodul PCD4.M445	2-1
2.2 PROFIBUS-FMS Anschaltung Typ PCD7.F700 zum Prozess-Steuergerät PCD2.M120	2-2
2.3 Daten zum PROFIBUS-Anschluss	2-3
2.4 Repeater PCD7.T100	2-4
2.5 Termination-Box PCD7.T160	2-5
<b>3. Planung und Installation eines Netzwerkes</b>	<b>3-1</b>
3.1 Physical Layers	3-1
3.2 Logische Netzwerk-Struktur	3-2
<b>4. Der PROFIBUS-Konfigurator</b>	<b>4-1</b>
4.1 Allgemeines	4-1
4.2 Vorgehensweise für die Konfiguration	4-2
4.3 Standard-Setup	4-3
4.3.1 Konfigurator Hauptmenü	4-3
4.3.2 Bus-Definitionen	4-4
4.3.3 Stations-Definitionen	4-5
4.3.4 Kommunikations-Kanäle	4-6
4.3.5 Link-Fenster	4-9
4.3.6 Objekt-Definitionen	4-11
4.4 Erweiterter Setup	4-14
4.4.1 Definition der Bus-Parameter	4-14
4.4.2 Kanal-Definitionen	4-19
4.4.3 Objekt-Definitionen	4-27

		Seite
<b>5.</b>	<b>Programmierung</b>	<b>5-1</b>
5.1	Beschreibung der PCD-Befehle (SASI, SCON, STXM, SRXM)	5-1
5.1.1	SASI: Assignierung eines Kanals	5-2
5.1.2	SCON: Öffnen eines Kommunikationskanals	5-7
5.1.3	SRXM: Empfange Daten (Objekte)	5-8
5.1.4	SRXMI: Empfange indirekt Daten (Objekte)	5-9
5.1.5	STXM: Sende Daten (Objekte)	5-11
5.1.6	STXMI: Sende indirekt Daten (Objekte)	5-12
5.2	Regeln für die Erstellung des Anwenderprogram.	5-14
5.3	Struktur des Anwenderprogrammes	5-15
5.3.1	Kaltstartprogramm im XOB 16	5-15
5.3.2	Hauptprogramm im COB	5-16
5.4	Programmbeispiele	5-18
5.4.1	Beispiel 1: 2 PCD Stationen	5-18
5.4.2	Beispiel 2: 1 PCD und 1 Fremdstation	5-19
5.4.3	Beispiel 3: 2 PCD und 2 Fremdstationen	5-20
<b>6.</b>	<b>Inbetriebnahme eines PROFIBUS-Netzwerkes</b>	<b>6-1</b>
6.1	Überprüfung und Test der Hardware-Installation (Physical Layer)	6-1
6.2	Überprüfung und Test der Schicht 7	6-2
6.2.1	Verbindungsaufbau (Initiate)	6-2
6.2.2	Verbindungsabbau (Abort)	6-3
6.2.3	Überprüfung des Datenaustausches	6-3
6.2.4	Optimierung der Übertragungszeiten	6-4
6.2.5	Verwendung von Busmonitoren	6-4
Anhang A:	Objekt-Datentypen	A-1
Anhang B:	Einfluss der Bus-Parameter auf die Übertragungszeiten	B-1
Anhang C:	PROFIBUS-Dienste	C-1
Anhang D:	Abkürzungen und Begriffe	D-1



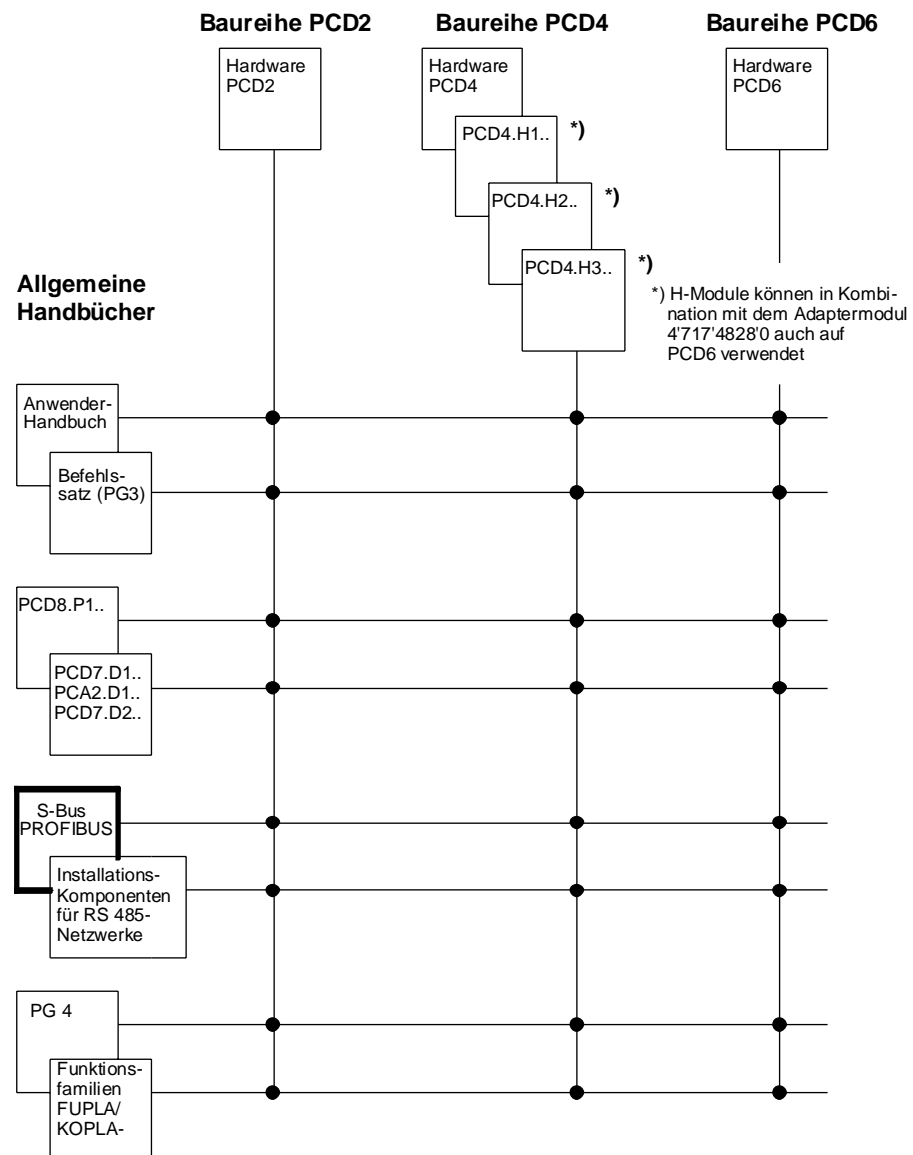
**Wichtiger Hinweis:**

Um den einwandfreien Betrieb von SAIA® PCD sicherstellen zu können, wurde eine Vielzahl detaillierter Handbücher geschaffen. Diese wenden sich an technisch qualifiziertes Personal, das nach Möglichkeit auch unsere Workshops erfolgreich absolviert hat.

Die vielfältigen Leistungen der SAIA® PCD treten nur dann optimal in Erscheinung, wenn alle in diesen Handbüchern aufgeführten Angaben und Richtlinien bezüglich Montage, Verkabelung, Programmierung und Inbetriebnahme genau befolgt werden.

Damit allerdings werden Sie zum grossen Kreis der begeisterten SAIA® PCD Anwendern gehören.

**Übersicht**



## Zuverlässigkeit und Sicherheit elektronischer Steuerungen

Die Firma SAIA-Burgess Electronics AG konzipiert, entwickelt und stellt ihre Produkte mit aller Sorgfalt her:

- Neuster Stand der Technik
- Einhaltung der Normen
- Zertifiziert nach ISO 9001
- Internationale Approbationen: z.B. Germanischer Lloyd, UL, Det Norske Veritas, CE-Zeichen ...
- Auswahl qualitativ hochwertiger Bauelemente
- Kontrollen in verschiedenen Stufen der Fertigung
- In-Circuit-Tests
- Run-in (Wärmelauf bei 85°C während 48h)

Die daraus resultierende hochstehende Qualität zeigt trotz aller Sorgfalt Grenzen. So ist z.B. mit natürlichen Ausfällen von Bauelementen zu rechnen. Für diese gibt die Firma SAIA-Burgess Electronics AG Garantie gemäss den "Allgemeinen Lieferbedingungen".

Der Anlagebauer seinerseits muss auch seinen Teil für das zuverlässige Arbeiten einer Anlage beitragen. So ist er dafür verantwortlich, dass die Steuerung datenkonform eingesetzt wird und keine Überbeanspruchungen, z.B. auf Temperaturbereiche, Überspannungen und Störfelder oder mechanischen Beanspruchungen auftreten.

Darüber hinaus ist der Anlagebauer auch dafür verantwortlich, dass ein fehlerhaftes Produkt in keinem Fall zu Verletzungen oder gar zum Tod von Personen bzw. zur Beschädigung oder Zerstörung von Sachen führen kann. Die einschlägigen Sicherheitsvorschriften sind in jedem Fall einzuhalten. Gefährliche Fehler müssen durch zusätzliche Massnahmen erkannt und hinsichtlich ihrer Auswirkung blockiert werden. So sind z.B. für die Sicherheit wichtige Ausgänge auf Eingänge zurückzuführen und softwaremässig zu überwachen. Es sind die Diagnoseelemente der PCD wie Watch-Dog, Ausnahme-Organisations-Blocks (XOB) sowie Test- und Diagnose-Befehle konsequent anzuwenden.

Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA® PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.



# 1. Eigenschaften und Grundlagen

PROFIBUS (für *PRO*cess *FI*eld *BUS*) ist der erfolgreiche, offene industrielle Feldbusstandard für ein breites Anwendungsspektrum.

## 1.1 Eigenschaften

- Offen**  
 PROFIBUS ermöglicht den Datenaustausch zwischen Geräten unterschiedlicher Hersteller ohne spezielle Schnittstellenanpassungen. PROFIBUS ist standardisiert als stabile Deutsche Norm DIN 19 245 und als europäische Norm pr EN 50170.
- Firmenneutral**  
 PROFIBUS Geräte werden von einer Vielzahl qualifizierter Hersteller angeboten. Dadurch können die Anwender das am besten geeignete Gerät ihres bevorzugten Herstellers für die jeweilige Applikation auswählen.
- Optimiert für ein breites Einsatzgebiet**  
 PROFIBUS besteht aus den drei folgenden anwendungsspezifischen Varianten:

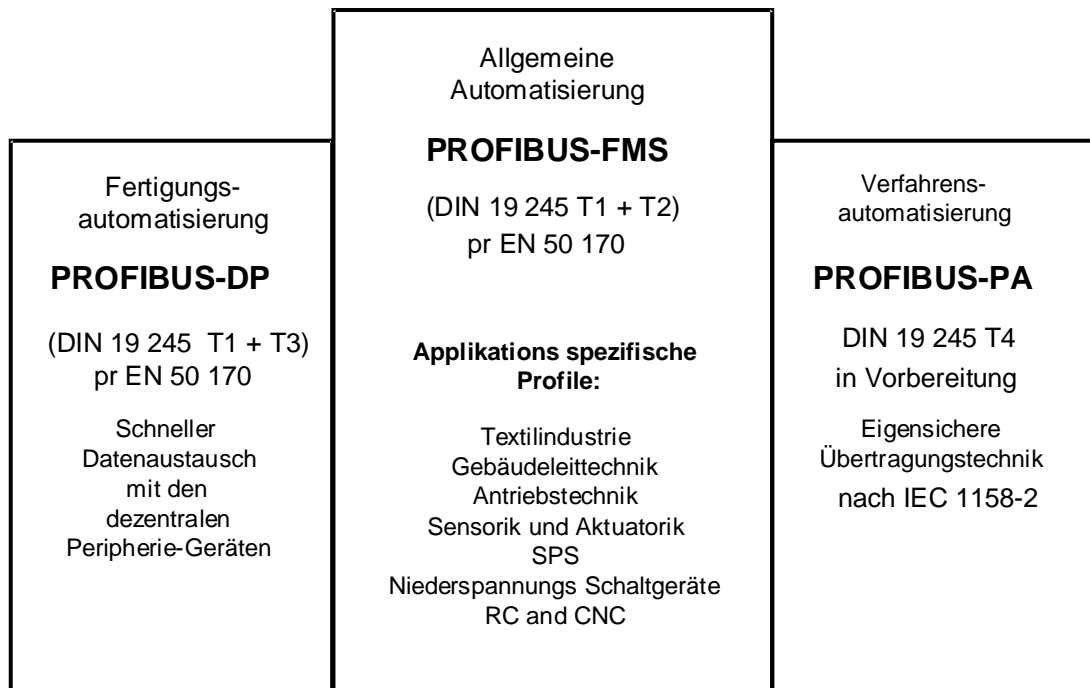


Bild 1: Übersicht PROFIBUS-Varianten

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

Dies ist die universelle Lösung für Kommunikationsaufgaben in der Feld- und Zellebene der industriellen Kommunikationshierarchie. Die leistungsfähigen FMS-Services eröffnen einen breiten Anwendungsbereich, große Flexibilität und ermöglichen es, die umfangreichen Kommunikationsaufgaben mit zyklischem oder azyklischem Datenverkehr bei einer mittleren Geschwindigkeit zu bewältigen.

Seit 1990 ist PROFIBUS-FMS als stabiler Standard DIN 19 245 Teil 1 und Teil 2 veröffentlicht. Im Zuge der europäischen Feldbusstandardisierung wird PROFIBUS-FMS in die europäische Feldbusnorm pr EN 50170 integriert.

**PROFIBUS-DP** (DP = Dezentrale Peripherie)

Diese auf Geschwindigkeit optimierte PROFIBUS Variante ist speziell für die Kommunikation zwischen Automatisierungssystemen und den dezentralen Peripheriegeräten zugeschnitten. PROFIBUS-DP basiert auf DIN 19 245 Teil 1 und anwendungsspezifischen Erweiterungen, die in der Deutschen Norm DIN 19 245 Teil 3 festgelegt sind. Im Zuge der europäischen Feldbusstandardisierung wird PROFIBUS-DP in die europäische Feldbusnorm pr EN 50170 integriert.

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

PROFIBUS-PA ist die PROFIBUS-Variante für Anwendungen in der Prozessautomatisierung. PROFIBUS-PA verwendet die in IEC 1158-2 festgelegte eigensichere Übertragungstechnik und ermöglicht die Fernspeisung der Teilnehmer über den Bus. Geräteprofile legen die gerätespezifischen Funktionen fest. Device Description Language (DDL) und Function Blocks ermöglichen die vollständige Interoperabilität der Geräte.

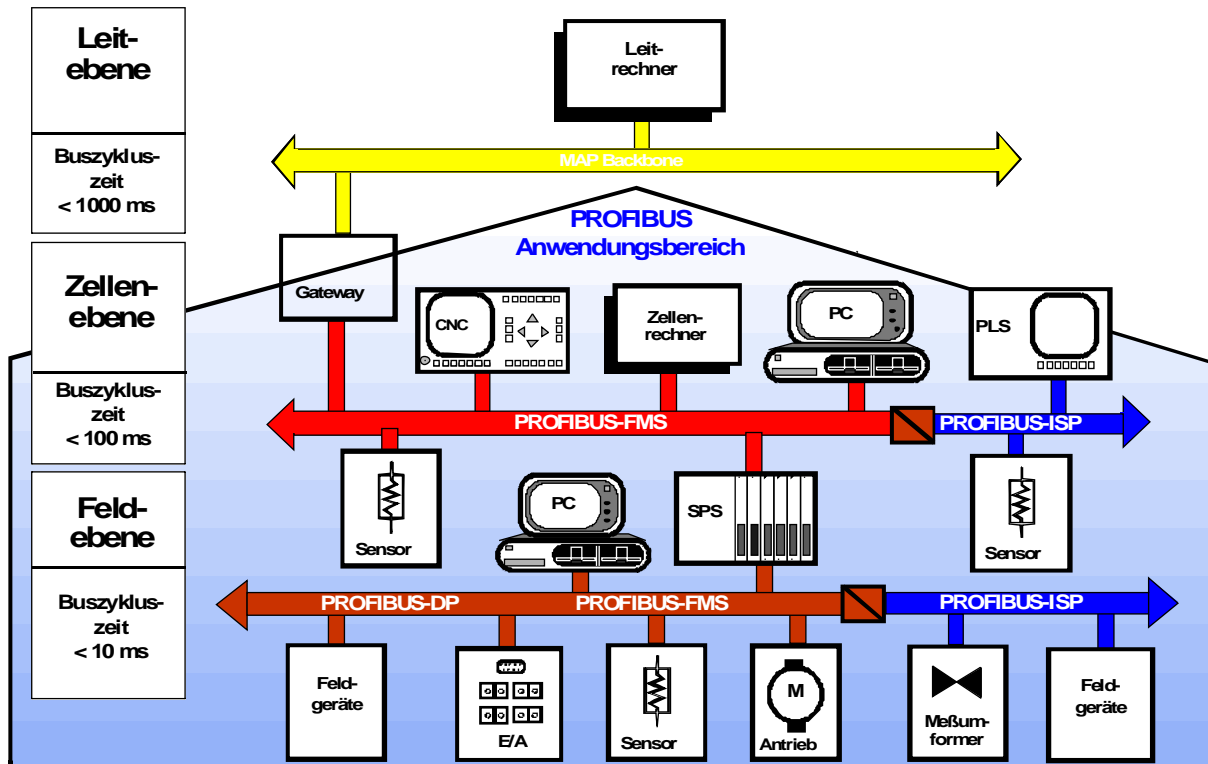


Bild 2: Kommunikation im Feldbereich

- Bewährt**  
 PROFIBUS ist eine erprobte Technologie, die sich in vielen Anwendungen im Bereich der Gebäudeautomatisierung, Fertigungsautomatisierung, Prozessautomatisierung und Antriebstechnik bereits vielfach bewährt hat.
- Zertifiziert**  
 Durch den Konformitäts- und Interoperabilitätstest bei den von der PROFIBUS Nutzerorganisation (PNO) autorisierten Testlaboratorien und eine Zertifizierung der Geräte durch die PNO, erhält der Anwender die Sicherheit, dass Qualität und Funktionalität auch in Multi-Vendor Installationen sichergestellt sind.

## 1.2 Grundlagen

---

Um ein PROFIBUS-Netzwerk aufzubauen und darüber Daten auszutauschen, ist es nicht unbedingt erforderlich, dass Sie das nun folgende Kapitel vollständig durchlesen und verstehen. Wenn Sie PROFIBUS-Kommunikation mit nur SAIA<sup>°</sup>PCD-Geräten planen, ist dies ohnehin ein sehr einfaches Unternehmen und Sie können direkt mit Kapitel 2 weiterfahren.

Wenn Sie jedoch PROFIBUS auch in der Theorie besser kennen und verstehen möchten, dann wird Ihnen das folgende Kapitel dabei sicher nützlich sein.

### 1.2.1 PROFIBUS Schicht 1 (Physical Layer)

Der Einsatzbereich eines Feldbus-Systems wird wesentlich durch die Wahl des Übertragungs-Mediums und der physikalischen Busschnittstelle bestimmt. Neben den Anforderungen an die Übertragungssicherheit sind die Aufwendungen für Beschaffung und Installation des Buskabels von entscheidender Bedeutung. Die PROFIBUS-Norm sieht daher verschiedene Ausprägungen der Übertragungstechnik unter Beibehaltung eines einheitlichen Busprotokolls vor.

**Kabelgebundene Übertragung:** Diese Version, gemäss dem US-Standard EIA RS-485, wurde als Grundversion für Anwendungen im Bereich der Fertigungstechnik, Gebäudeleittechnik und Antriebstechnik festgelegt. Es wird ein abgeschirmtes, verdrehtes Kupferkabel mit einem Leiterpaar verwendet.

**Lichtwellen Leiter:** Für Anwendungen in stark störbehafteter Umgebung, sowie zur Vergrößerung der Reichweite bei hohen Übertragungsgeschwindigkeiten, wurde in der PNO die Spezifikation einer auf Lichtwellenleitern basierenden Übertragungstechnik erarbeitet.

**Eigensichere Übertragung:** PROFIBUS-PA ermöglicht die eigensichere Übertragung und die Speisung der Teilnehmer über den Bus. Die Übertragungstechnik ist in der internationalen Norm IEC 1158-2 beschrieben.

### 1.2.2 PROFIBUS Schicht 2 (Data Link Layer)

Die zweite Schicht des OSI-Referenzmodells realisiert die Funktionen der Buszugriffssteuerung, Datensicherung sowie die Abwicklung der Übertragungsprotokolle und der Telegramme. Die Schicht 2 wird bei PROFIBUS als Fieldbus Data Link (FDL) bezeichnet.

Die Buszugriffssteuerung (MAC, Medium Access Control) legt das Verfahren fest, zu welchem Zeitpunkt ein Busteilnehmer Daten senden kann. Die MAC muss sicherstellen, daß zu einem Zeitpunkt immer nur ein Teilnehmer die Sendeberechtigung besitzt.

Das PROFIBUS-Buszugriffsverfahren beinhaltet deshalb das **Token-Passing-Verfahren** für die Kommunikation von komplexen Busteilnehmern (Master) untereinander und unterlagert das **Master-Slave-Verfahren** für die Kommunikation der komplexen Busteilnehmer mit den aufwandsarmen Peripheriegeräten (Slaves). Dieses kombinierte Verfahren wird als **hybrides Buszugriffsverfahren** (siehe Bild 3) bezeichnet.

Das Token-Passing-Verfahren wird beim PROFIBUS nur zwischen den komplexen Teilnehmern (Master) angewendet.

Das Master-Slave-Verfahren ermöglicht es dem Master (aktiver Teilnehmer), der gerade die Sendeberechtigung besitzt, die ihm zugeordneten Slave-Geräte (passive Teilnehmer) anzusprechen. Der Master hat hierbei die Möglichkeit, Nachrichten an die Slaves zu übermitteln bzw. Nachrichten von den Slaves abzuholen.

Mit dieser hybriden Zugriffsmethode von PROFIBUS können folgende Systemkonfigurationen realisiert werden:

- Reines Master-Slave-System.
- Reines Master-Master-System (Token-Passing).
- Ein System mit einer Kombination aus beiden Verfahren.

Bild 3 zeigt eine PROFIBUS-Struktur mit drei aktiven Teilnehmern (Mastern) und sieben passiven Teilnehmern (Slaves). Die drei Master-Geräte bilden einen logischen Token-Ring.

Erhält ein aktiver Teilnehmer das Token-Telegramm, so kann er für eine gewisse Zeit die "Masterfunktion" über den Bus ausüben und mit allen Slave-Teilnehmern in einer Master-Slave Kommunikationsbeziehung und mit allen Master-Teilnehmern in einer Master-Master Kommunikationsbeziehung kommunizieren.

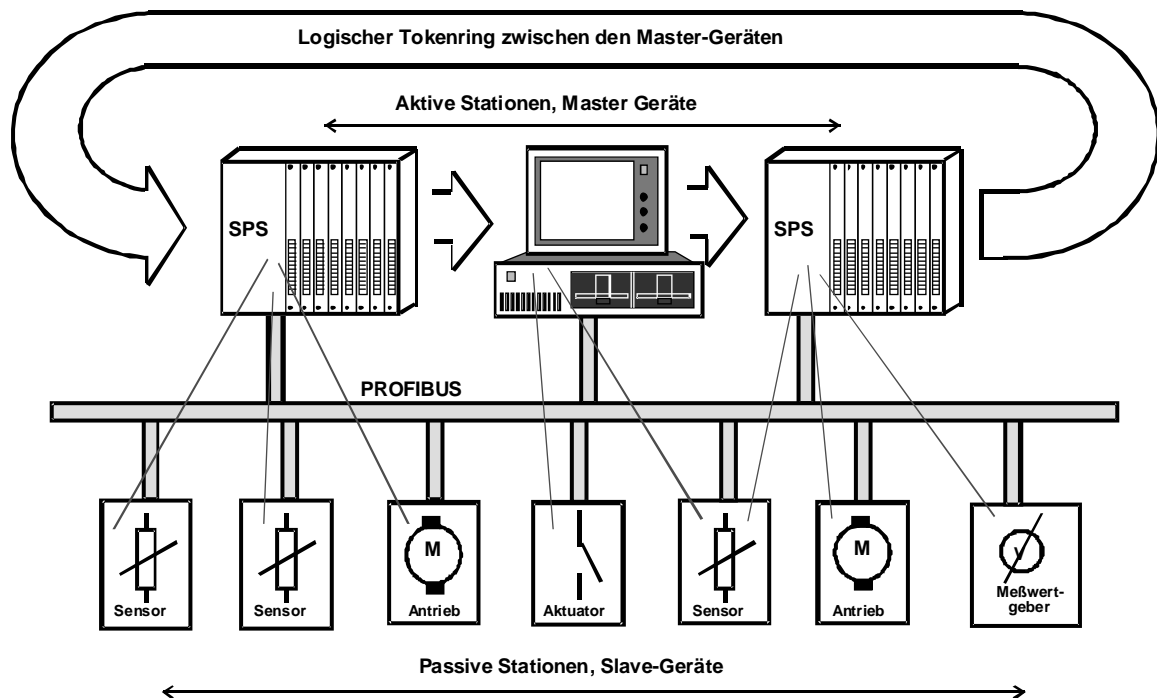


Bild 3: Hybrides Buszugriffsverfahren

Unter einem Token-Ring ist die organisatorische Aneinanderreihung von aktiven Teilnehmern, die durch ihre Busadressen einen logischen Ring bilden, zu verstehen. In diesem Ring wird der Token, die Buszugriffsberechtigung, von einem Master zum nächsten Master in einer vorgegebenen Reihenfolge (aufsteigende Adressen) weitergereicht.

Die Aufgabe der Buszugriffs-Steuerung (MAC) der aktiven Teilnehmer besteht darin, in der "Hochlauf-Phase" des Bussystems diese logische Zuordnung zu erkennen und den Token-Ring zu etablieren. Im laufenden Betrieb ist ein defekter oder abgeschalteter (aktiver) Teilnehmer aus dem Ring auszutragen, bzw. ein neu hinzukommender aktiver Teilnehmer aufzunehmen. Diese Leistungsmerkmale sind neben der Erkennung von Defekten im Übertragungsmedium und am Leitungsempfänger sowie von Fehlern bei der Teilnehmeradressierung (z.B. Mehrfachbelegung) oder der Token-Weitergabe (z.B. Mehrfachtoken oder Tokenverlust) charakteristisch für die Eigenschaften der PROFIBUS-Zugriffssteuerung.

Eine weitere wichtige Aufgabe der Schicht 2 ist die Datensicherung. Die PROFIBUS-Schicht-2 Telegrammformate ermöglichen eine große Übertragungssicherheit. Alle Telegramme weisen eine **Hamming-Distanz von  $HD = 4$**  auf. Dies wird durch die Anwendung der Vorschriften der internationalen Norm IEC 870-5-1 durch Auswahl von besonderen Start- und Ende-Zeichen der Telegramme, schlupffeste Synchronisierung, Paritätsbit und Kontrollbyte erreicht.

Bei PROFIBUS-FMS und PROFIBUS-DP wird jeweils ein spezifischer Subset der Schicht 2 Dienste verwendet, siehe nachfolgende Tabelle. Die Dienste werden über die Dienstzugangspunkte (SAPs) der Schicht 2 von den übergeordneten Schichten aufgerufen. Bei PROFIBUS-FMS werden diese Dienstzugangspunkte für die Adressierung der logischen Kommunikationsbeziehungen benutzt. Bei PROFIBUS-DP ist jedem verwendeten Dienstzugangspunkt eine genau festgelegte Funktion zugeordnet. Bei allen aktiven und passiven Teilnehmern können mehrere Dienstzugangspunkte parallel benutzt werden. Es wird zwischen Quell- (SSAP) und Ziel-Dienstzugangspunkten (RSAP) unterschieden.

Tabelle: Dienste der PROFIBUS Datensicherungsschicht (Schicht 2).

Dienst	Funktion	Benutzt bei
SDA	Send Data with Acknowledge (Daten senden mit Quittung)	FMS
SRD	Send And Request Data With Reply (Daten senden und empfangen mit Quittung)	FMS DP
SDN	Send Data With No Acknowledge (Daten senden unquittiert)	FMS DP
CSRD	Cyclic Send And Request Data With Reply (zyklisches Senden und Empfangen mit Quittung)	FMS

### 1.2.3 PROFIBUS-FMS

- **PROFIBUS Schicht 7 (Application Layer)**

Die Schicht 7 des ISO/OSI Referenzmodells stellt die für den Anwender nutzbaren Kommunikations-Dienste zur Verfügung. Diese Anwendungsdienste ermöglichen den effizienten, offenen Datenverkehr zwischen Anwendungsprozessen. Die PROFIBUS Anwendungsschicht ist in DIN 19 245 Teil 2 spezifiziert und besteht aus:

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

FMS beschreibt die Kommunikationsobjekte, die Anwendungsdienste und die daraus resultierenden Modelle aus Sicht des Kommunikationspartners.

Das LLI dient zur Anpassung der Anwendungsfunktionen an die vielfältigen Eigenschaften der PROFIBUS Schicht 2.

• **PROFIBUS Kommunikationsmodell**

Ein Anwendungsprozess umfasst alle Programme, Ressourcen und Tasks, die keiner Kommunikationsschicht zugeordnet sind. Das PROFIBUS Kommunikationsmodell gestattet es, verteilte Anwendungsprozesse über Kommunikationsbeziehungen zu einem Gesamtprozess zu vereinigen.

Der Teil eines Anwendungsprozesses in einem Feldgerät, der über die Kommunikation erreichbar ist, wird als virtuelles Feldgerät (Virtual Field Device, VFD) bezeichnet, siehe folgendes Bild.

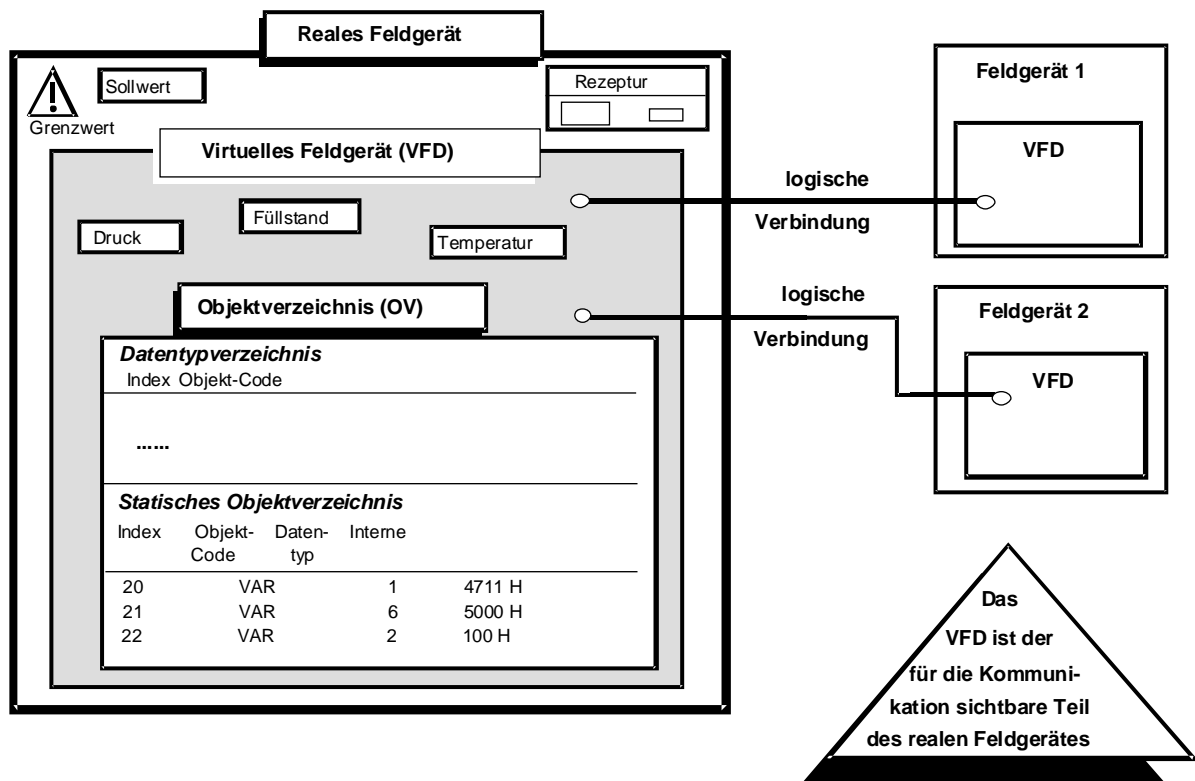


Bild 4: Virtuelles Feldgerät (VFD)

Die Abbildung der Funktionen des VFD auf das reale Gerät wird im PROFIBUS Kommunikationsmodell durch das Application Layer Interface (ALI) erbracht. Bild 4 zeigt den Zusammenhang zwischen dem realen Feldgerät und dem VFD. In diesem Beispiel sind nur die Variablen Druck, Füllstand und Temperatur Bestandteil des VFD und können über die beiden Kommunikationsbeziehungen gelesen oder geschrieben werden.



- **Kommunikationsobjekte und Objektverzeichnis (OV)**

Alle Kommunikationsobjekte eines PROFIBUS-Teilnehmers werden in seinem lokalen Objektverzeichnis eingetragen. Das Objektverzeichnis kann bei einfachen Geräten vordefiniert sein. Bei komplexen Geräten wird das Objektverzeichnis projiziert und lokal oder remote in das Gerät geladen. Das OV enthält Beschreibung, Struktur und Datentyp sowie die Zuordnung zwischen der geräteinternen Adresse der Kommunikationsobjekte und der Bezeichnung am Bus (Index/Name). Das OV besteht aus:

- Header (enthält Informationen über die Struktur des OV).
- Liste der statischen Datentypen (Liste der unterstützten statischen Datentypen)
- Statisches Objektverzeichnis (enthält alle statischen Kommunikationsobjekte)
- Dynamische Liste der Variablen Listen (Liste der aktuell bekannten Variablen-Listen)
- Dynamische Programm Liste (Liste der aktuell bekannten Programme)

Die einzelnen Abschnitte des OV's müssen nur dann vorhanden sein, wenn das Gerät die jeweiligen Funktionen auch unterstützt.

Statische Kommunikationsobjekte werden im statischen Objektverzeichnis eingetragen und können durch den Hersteller des Gerätes vordefiniert oder bei der Projektierung des Bussystems festgelegt werden. Bei der Kommunikation im Feldbereich werden überwiegend statische Kommunikationsobjekte benutzt. PROFIBUS kennt folgende statische Kommunikationsobjekte:

- Simple Variable (Einfach-Variable)
- Array (Reihe von Einfach-Variablen gleichen Typs)
- Record (Reihe von Einfach-Variablen unterschiedlichen Typs)
- Domain (Datenbereich)
- Event (Ereignismeldung)

Dynamische Kommunikationsobjekte werden im dynamischen Teil des OV (Variablenlisten-Verzeichnis / Program-Invocation-Verzeichnis) eingetragen. Sie können vordefiniert oder in der Betriebsphase mit den Anwendungsdiensten definiert, gelöscht oder verändert werden.

PROFIBUS unterstützt folgende dynamische Kommunikationsobjekte:

- Program-Invocation (Programm)
- Variable-List (Reihe von Simple Variables, Arrays oder Records)

Bei PROFIBUS ist die logische Adressierung als bevorzugte Methode für die Adressierung der Kommunikationsobjekte festgelegt. Dabei erfolgt der Zugriff auf die Kommunikationsobjekte über eine Kurzadresse, den sogenannten Index. Der Index ist eine Zahl vom Typ Unsigned16. Dadurch werden effiziente Telegramme ermöglicht und der Protokoll-Overhead reduziert. Für jedes Kommunikationsobjekt eines Gerätes wird im OV ein Index festgelegt. Die logische Adressierung muss von allen PROFIBUS-Teilnehmern unterstützt werden.

Zusätzlich ermöglicht PROFIBUS-FMS für besondere Anwendungen folgende optionale Adressierungsverfahren:

- Die Adressierung mit Namen: Dabei wird der symbolische Name der Kommunikationsobjekte über den Bus übertragen.
- Die physikalische Adressierung: Dabei kann eine beliebige physikalische Speicheradresse mit den Services Physical-Read und Physical-Write in den Feldgeräten angesprochen werden.

Jedes Kommunikationsobjekt kann (optional) vor unberechtigtem Zugriff geschützt werden. Dazu ist es möglich, den Zugriff auf ein Objekt nur mit einem bestimmten Passwort oder für eine bestimmte Gerätegruppe zu ermöglichen. Passwort und Gerätegruppe können im Objektverzeichnis spezifisch für jedes Objekt festgelegt werden. Zusätzlich können die erlaubten Services für den Zugriff auf ein Objekt durch Eintrag in das Objektverzeichnis eingeschränkt werden (z. B. nur lesender Zugriff erlaubt).

- **PROFIBUS-FMS Services**

Aus Sicht eines Anwendungsprozesses stellt das Kommunikationssystem eine Dienstleistung, die sogenannten FMS-Services, zur Verfügung. Die Abwicklung dieser Dienste wird durch Dienstprimitiven beschrieben. Sie stellen eine Interaktion zwischen dem Dienstanforderer (Requester) und dem Diensterbringer (Responder) dar.

Die folgende Abbildung gibt einen Überblick über die möglichen PROFIBUS-Dienste.

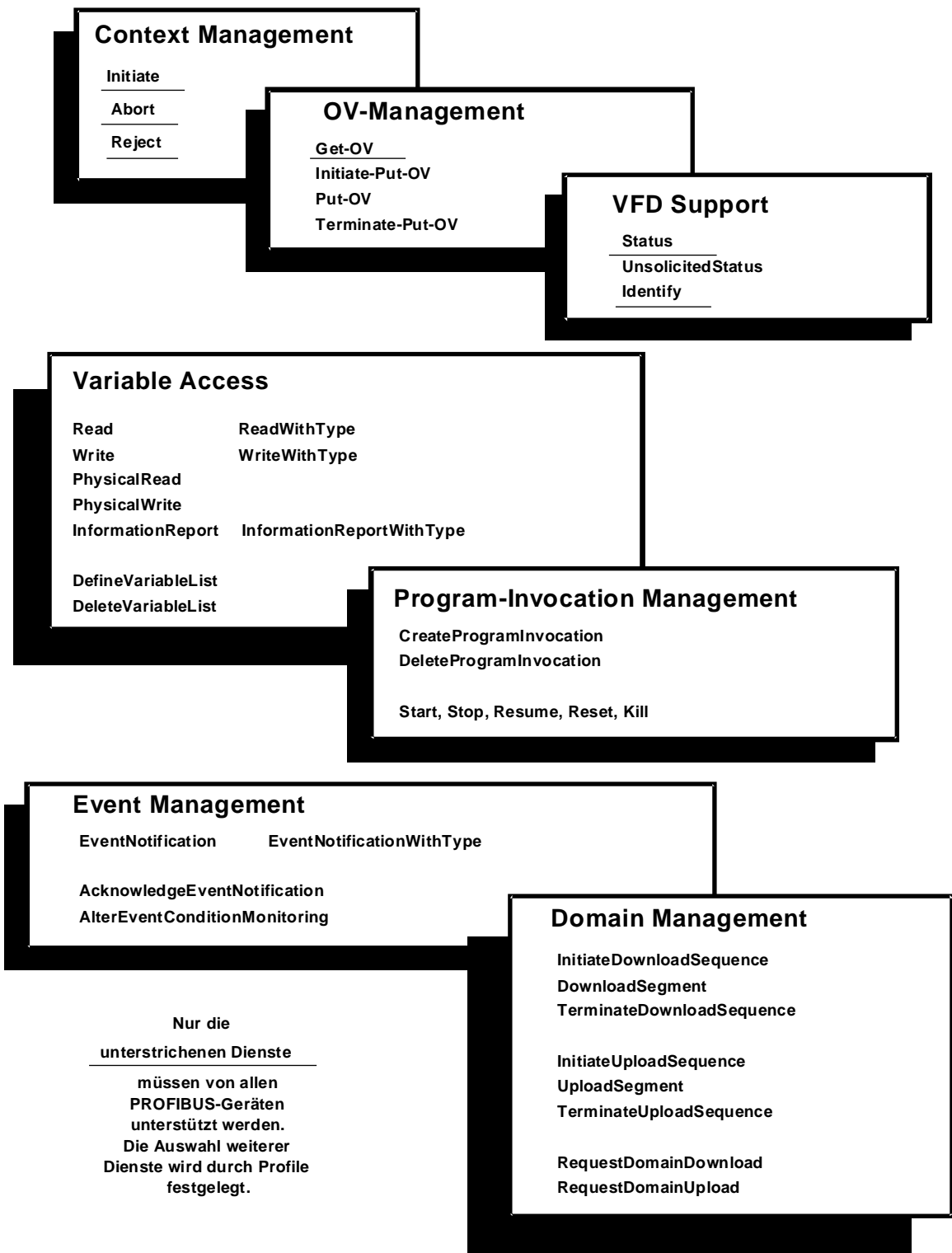


Bild 5: Übersicht über die PROFIBUS-FMS Dienste

**Bestätigte Dienste** sind nur auf verbindungsorientierten Kommunikationsbeziehungen möglich. Der Ablauf eines bestätigten Dienstes ist in der nächsten Abbildung dargestellt.

Unbestätigte Dienste können auch auf verbindungslosen Kommunikationsbeziehungen (Broadcast, Multicast) angewendet werden. Sie können mit hoher oder niedriger Priorität übertragen werden. Ein unbestätigter Dienst wird mit einer Request-Dienstprimitive angefordert. Nach der Übertragung über den Bus wird dem Applikationsprozess bei den Empfängern eine Indication-Dienstprimitive übergeben. Die Confirmation-/Response-Dienstprimitiven sind bei unbestätigten Services nicht vorhanden.

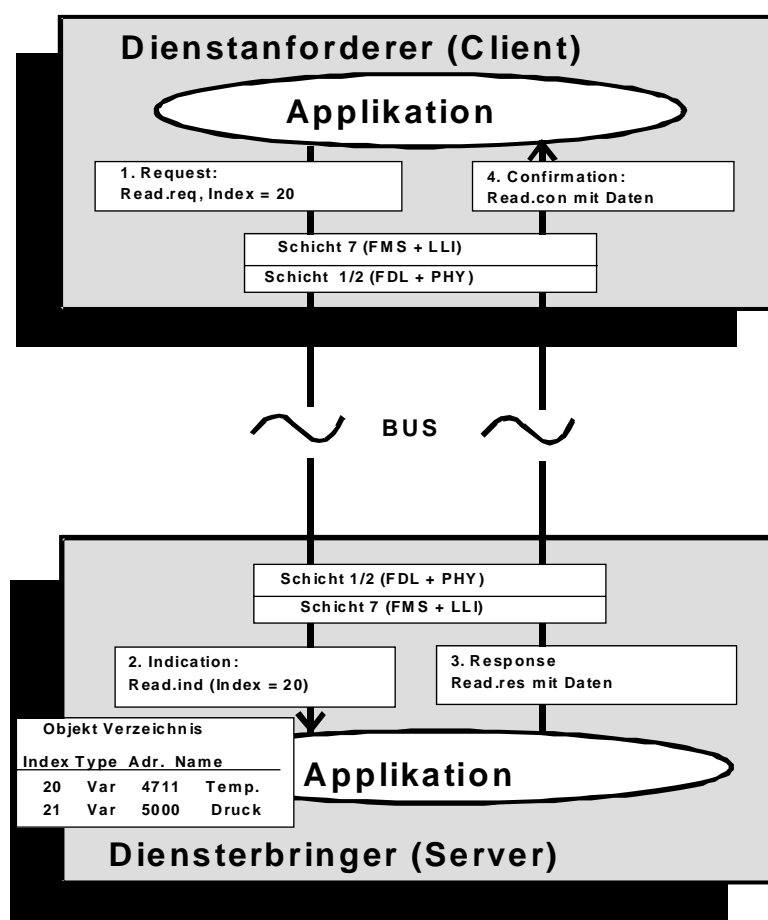


Bild 6: Ablauf eines bestätigten Dienstes

Die FMS-Services werden in folgende Gruppen eingeteilt:

- Die **Context Management Services** ermöglichen das Einrichten und Beenden von logischen Verbindungen sowie das Abweisen von unzulässigen Services.

- Die **Variable Access Services** ermöglichen den Zugriff auf Einfachvariablen, Records, Arrays und Variablenlisten.
- Die **Domain Management Services** ermöglichen die Übertragung von zusammenhängenden Speicherbereichen. Bei der Übertragung werden grosse Datenmengen vom Anwendungsprozess in Segmente unterteilt.
- Die **Program Invocation Management Services** ermöglichen die Programmsteuerung.
- Die **Event Management Services** ermöglichen die Übertragung von Alarmmeldungen. Diese unbestätigten Services können mit hoher oder niedriger Priorität übertragen werden. Die Alarmmeldung kann auch über Broadcast- oder Multicast-Kommunikationsbeziehungen gesendet werden.
- Die **VFD Support Services** ermöglichen die Identifikation und Zustandsabfrage eines Gerätes. Zustandsmeldungen können auch unaufgefordert, auf Initiative des betroffenen Gerätes, über Broadcast oder Multicast Kommunikationsbeziehungen versendet werden.
- Die **OV Management Services** ermöglichen das Lesen und Schreiben von Objektverzeichnissen.

Die grosse Anzahl der möglichen PROFIBUS Anwendungsdienste dient dazu, die vielfältigen Anforderungen der unterschiedlichen Feldgeräte an die Kommunikation zu erfüllen. Nur wenige der Anwendungsdienste müssen zwingend implementiert werden. Die Auswahl weiterer Dienste hängt von der jeweiligen Anwendung ab. Diese Auswahl wird spezifisch für die jeweiligen Einsatzbereiche in sogenannten Profilen festgelegt.

- **Lower-Layer-Interface (LLI)**

Die Anpassung der Schicht 7 an die Schicht 2 wird durch das LLI realisiert. Das LLI übernimmt die Flußkontrolle und Verbindungsüberwachung sowie die Abbildung der FMS-Services auf die Schicht 2, unter Beachtung der verschiedenen Gerätetypen (Master/Slave).

Der Anwender kommuniziert mit den anderen Anwendungsprozessen über logische Kanäle, den **Kommunikationsbeziehungen**. Das LLI stellt für die Abwicklung der FMS- und FMA7-Services verschiedene Typen von Kommunikationsbeziehungen zur Verfügung. Die Kommunikationsbeziehungen unterscheiden sich bezüglich ihrer Verbindungsqualitäten (Überwachungsmechanismen, Übertragungsmöglichkeiten und Anforderungen an die Kommunikationspartner) voneinander. Die Auswahl erfolgt nach den Anforderungen der Anwendungsprozesse. Die folgende Abbildung gibt eine Zusammenfassung der möglichen Kommunikationsbeziehungen.

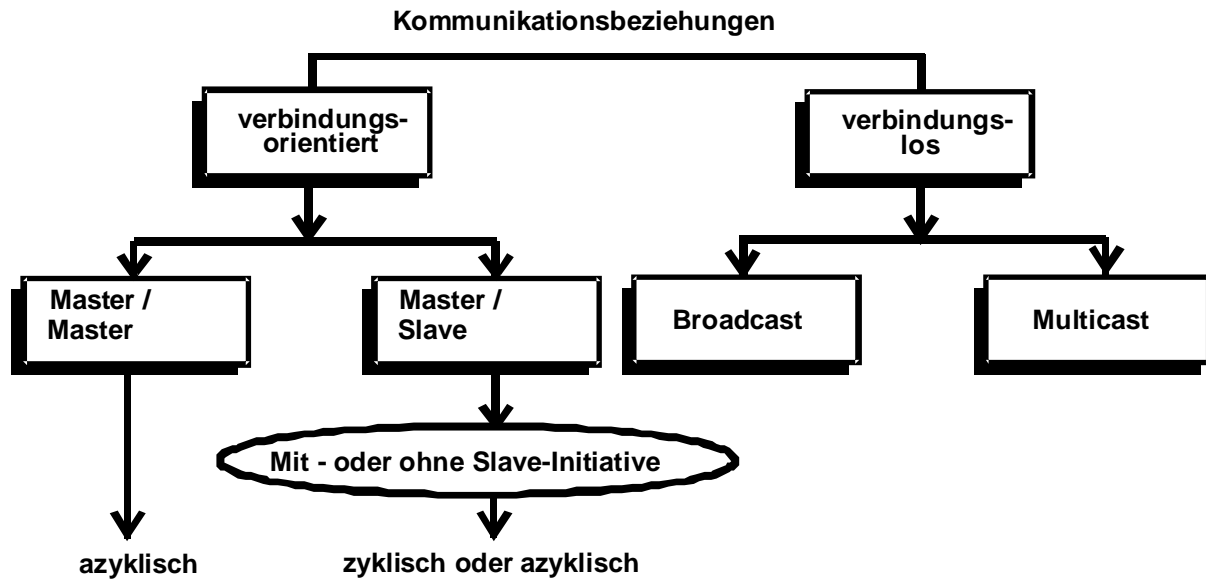


Bild 7: Übersicht über die möglichen Kommunikationsbeziehungen

**Verbindungsorientierte Kommunikationsbeziehungen** stellen eine logische Punkt-zu-Punkt Verbindung zwischen zwei Anwendungsprozessen dar. Vor der Nutzung einer Verbindung für die Datenübertragung muß diese mit einem Initiate-Service aufgebaut werden. Nach dem erfolgreichen Verbindungsaufbau steht die gegen Fremdzugriffe geschützte Verbindung für die Datenübertragung mit beliebigen FMS-Services zur Verfügung. Wird eine aufgebaute Verbindung für die Datenkommunikation nicht mehr benötigt, dann kann sie mit dem Abort-Service abgebaut werden. Das LLI ermöglicht eine zeitgesteuerte Verbindungsüberwachung für verbindungsorientierte Kommunikationsbeziehungen.

**Verbindungslose Kommunikationsbeziehungen** ermöglichen die gleichzeitige Kommunikation eines Gerätes mit mehreren Teilnehmern. Es können nur unbestätigte Dienste benutzt werden.

Bei Broadcast-Kommunikationsbeziehungen wird ein unbestätigter FMS-Service gleichzeitig an alle anderen Teilnehmer und entsprechend bei Multicast-Kommunikationsbeziehungen gleichzeitig an eine Gruppe von Teilnehmern (Master- oder Slave-Geräte) gesendet. Typische Anwendungen dieser Funktionen sind die Synchronisierung von Prozessen oder die Übertragung von Global-Alarmen.

- **Verbindungsattribute**

Ein weiteres Unterscheidungsmerkmal der verbindungsorientierten Kommunikationsbeziehungen ist das Verbindungsattribut. PROFIBUS-FMS unterscheidet offene und definierte Verbindungen. Bei **definierten Verbindungen** wird der Kommunikationspartner zum Projektierungszeitpunkt eindeutig festgelegt und kann in keiner Phase der Verbindung verändert werden. Dadurch wird ein unberechtigter Zugriff verhindert.

Bei **offenen Verbindungen** wird der Kommunikationspartner erst in der Verbindungsaufbauphase festgelegt. Besondere Vorteile bieten offene Verbindungen für einfachste PROFIBUS-Geräte, bei denen alle Kommunikationsbeziehungen durch den Hersteller fest vorgegeben sind. Werden die vordefinierten Verbindungen als offene Verbindungen gekennzeichnet, kann das Gerät ohne Projektierung mit jedem anderen PROFIBUS-Teilnehmer kommunizieren.

Wird eine Master-Slave Kommunikationsbeziehung mit dem Verbindungsattribut "Slave-Initiative" versehen, so ist es für den Slave möglich, unbestätigte FMS-Services an den zugeordneten Master abzusetzen. Das LLI ermöglicht diese Funktionalität, ohne dem Slave die Schicht-2 Buszugriffsberechtigung zu übergeben. Der Slave darf die unbestätigten Dienste mit hoher oder niedriger Priorität absetzen. Typische Anwendungen hierfür sind einfache Geräte, die eine Alarmmeldung unaufgefordert an den Prozessmaster senden müssen.

- **Zyklischer und azyklischer Datenverkehr**

PROFIBUS-FMS ermöglicht zyklischen- und azyklischen Datenverkehr.

**Zyklischer Datenverkehr** bedeutet, daß über eine Verbindung mit den Read- oder Write-Services genau eine Variable ständig gelesen oder geschrieben wird. Das LLI stellt hierfür eine besonders effiziente Form der Dienstabwicklung zur Verfügung. Dadurch werden die Übertragungszeiten gegenüber dem azyklischen Datenverkehr wesentlich verkürzt.

**Azyklischer Datenverkehr** bedeutet, dass über eine Verbindung verschiedene Kommunikationsobjekte sporadisch auf Anforderung des Anwendungsprozesses angesprochen werden.

- **Kommunikationsbeziehungsliste**

Alle Kommunikationsbeziehungen eines PROFIBUS-FMS Gerätes müssen in die KBL eingetragen werden. Die KBL enthält deren Beschreibung. Sie ist bei einfachen Teilnehmern durch den Hersteller fest vordefiniert. In allen anderen Fällen wird die KBL projektiert und lokal oder remote mit den Netzmanagement-Services geladen.

Vom Anwendungsprozess wird eine Kommunikationsbeziehung durch eine lokale Kurzbezeichnung, der Kommunikationsreferenz (KR) adressiert.

Aus Bus-Sicht wird eine Kommunikationsbeziehung durch die Teilnehmeradresse, Schicht-2-Dienstzugangspunkt und LLI-Dienstzugangspunkt adressiert.

Die KBL enthält die Zuordnung zwischen den lokalen Kommunikationsreferenzen, der Schicht-2- und der LLI-Adressierung. Weiterhin werden spezifisch für jede KR der Kommunikations-Kontext (Auswahl der Services, Telegrammlängen ...) und die anzuwendenden Überwachungsmechanismen angegeben.

- **PROFIBUS-Profile**

Um die universelle Anwendbarkeit des Bussystems zu ermöglichen, legt PROFIBUS-FMS einen sehr großen Funktionsumfang fest. Für die verschiedenen Anwendungsbereiche ist es erforderlich, den tatsächlich benötigten Funktionsumfang auf die konkreten Anforderungen abzustimmen. Diese Auswahl wird durch anwendungsspezifische Festlegungen (Profile) getroffen. Sie stellen sicher, daß Geräte von unterschiedlichen Herstellern die das gleiche Profil erfüllen, auch dieselbe Kommunikationsfunktionalität besitzen.

Die Inhalte eines Profils gehen jedoch noch über die Festlegung von reinen Kommunikationsparametern hinaus. Zusätzlich werden anwendungsspezifische Vereinbarungen über die Bedeutung der Kommunikationsfunktionen sowie die Interpretation von Status- und Fehlermeldungen getroffen.

Profile werden von Benutzergruppen in der PNO erarbeitet. Die folgende Tabelle gibt eine Übersicht über die momentan verfügbaren Profile für PROFIBUS-FMS.

Anwendungsbereich	Status
Gebäudeautomatisierung	PNO Richtlinie
Drehzahlveränderbare Antriebe	PNO Richtlinie
Sensoren und Aktuatoren	PNO Richtlinie
Programmable Controller	Entwurf PNO Richtlinie
Textilmaschinen	In Vorbereitung
Niederspannungs Schaltgeräte	Entwurf PNO Richtlinie

Profile werden für Master und Slave Geräte benötigt. Ein Master muss die Festlegungen aller Profile innerhalb seines Anwendungsbereichs unterstützen. Hierbei ist eine grosse Funktionalität wichtiger als die mögliche Kostenersparnis.

Ein Profil für Slaves hingegen, definiert nur den unbedingt erforderlichen funktionalen Subset. Dadurch wird eine deutliche Vereinfachung der Implementierung für den Hersteller und eine Kostenersparnis für den Anwender erreicht.



## 2. SAIA® PCD-Geräte für das PROFIBUS-Netzwerk

---

### 2.1 Das Prozessormodul PCD4.M445

---



Bild 8: Frontansicht PCD4.M445

Dieses Doppelprozessormodul enthält 2 vollständig unabhängige CPU zur Bearbeitung von Steuerungs- und Kommunikationsprogrammen sowie zusätzlich einen Co-Prozessor zum Anschluss an den **PROFIBUS-FMS**.

Neben der immer vorhandenen Programmierschnittstelle verfügt das Modul noch über 3 weitere voneinander unabhängige serielle Schnittstellen.

Der PROFIBUS-Anschluss erfolgt über den 9-poligen D-Sub-Stecker (weiblich).

Detaillierte Informationen (z.B. Pin-Belegung und Anschluss des Steckers) zu dem Modul, sind im Handbuch der PCD4 Hardware 26/734 zu finden.

## 2.2 PROFIBUS-FMS Anschaltung Typ PCD7.F700 zum Prozess-Steuergerät PCD2.M120

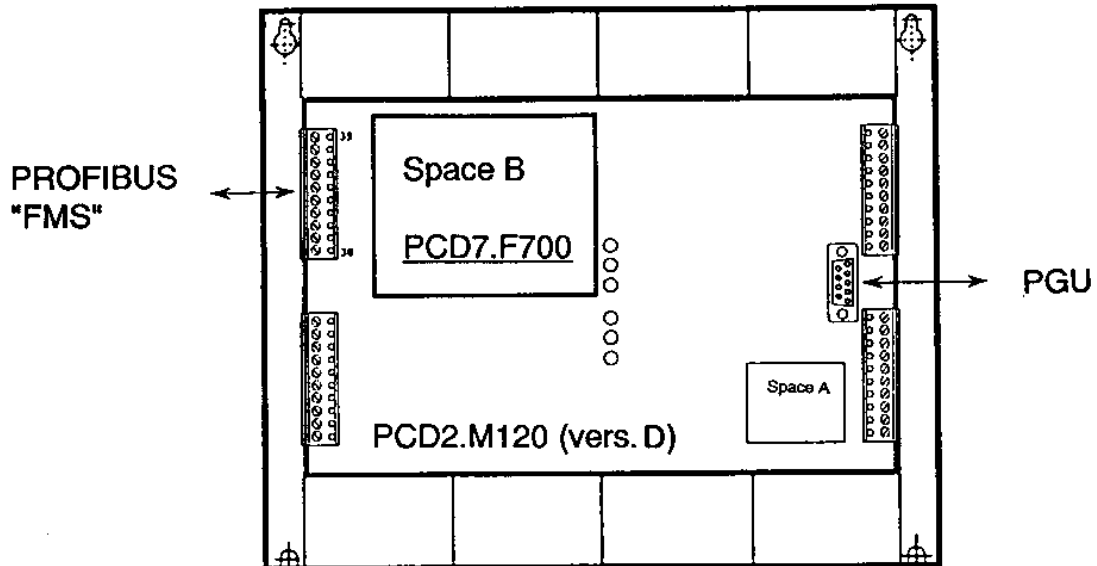


Bild: PCD2 mit PCD7.F700

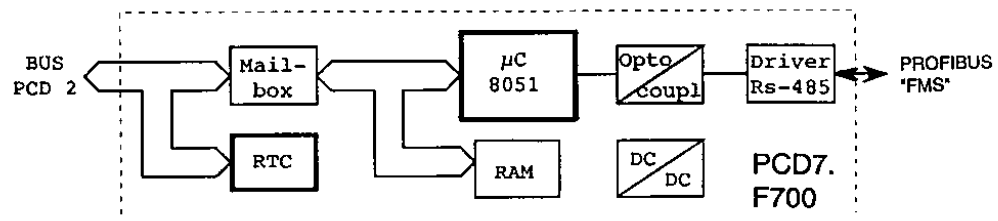


Bild 9: Architektur des PCD7.F700

Unter Verwendung der PROFIBUS-Anschaltung PCD7.F700 erhält die PCD2-Baureihe Anschluss an den PROFIBUS-FMS. Damit kann die PCD2 als FMS-Master oder FMS-Slave eingesetzt werden.

Das Modul PCD7.F700 wird am Steckplatz B des Basisgerätes PCD2.M120 aufgesteckt (Version D oder höher). Der Profibus-Anschluss erfolgt über die steckbaren Klemmen 30 bis 39. Das Modul verfügt neben dem PROFIBUS-Anschluss auch noch über eine Hardware-Uhr. Detaillierte Informationen zu diesem Modul finden Sie im Handbuch der PCD2 Hardware 26/737.

## 2.3 Daten zum PROFIBUS-Anschluss

Was	PCD4.M445	PCD2.M120 mit PCD7.F700
FMS-Protokoll nach DIN 19245 Teile 1 und 2	X	X
Anschluss als FMS-Master oder FMS-Slave	X	X
Geräteprofil: Controllerklasse 2 (erweitert)	X	X
Übertragungsraten: 9.6, 19.2, 38.4 oder 500 kbps	X	X
93.75 und 187.5 kbps	auf Anfrage	X
Galvanische Trennung der RS 485 Profibus-Schnittstelle	X	X
Bis 127 Stationen adressierbar (unterteilt mittels Repeater PCD7.T100 in Segmente zu 32 Stationen)	X	X
Gleichzeitig aufgebaute Verbindungen (Kanäle)	10..99	10..19
Anzahl Kommunikations-Objekte	100..499	100..199
FMS-Dienste		
Initiate	X	X
Abort	X	X
Reject	X	X
Identify (als Server)	X	X
Status	X	X
Get-OV (als Server)	X	X
Read/Write	X	X
Datentypen		
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

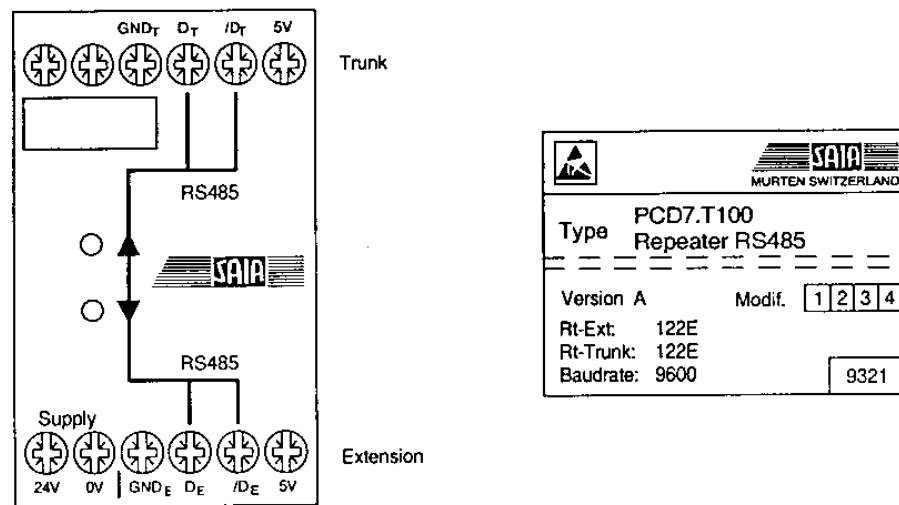


Bild 10: Frontplatte mit Typenschild

Der Repeater dient der Entkopplung eines RS 485-Kommunikations-Netzwerkes. Dabei werden zwei Ziele verfolgt, die Wiederaufbereitung der Übertragungssignale und die galvanische Trennung einzelner Leitungsabschnitte.

Die Wiederaufbereitung der Signale ist erforderlich, wenn

- die Gesamtlänge des Buskabels die maximal zulässige Länge übersteigt.
- mehr als 31 Teilnehmer am Netzwerk beteiligt sind.

Die galvanische Trennung ist erforderlich, wenn

- die Bezugserde im Bereich der Anlage zu grosse Potentialunterschiede aufweist.
- die Bezugserde der Anlage von vagabundierenden Strömen zu stark belastet wird.

Es dürfen maximal 3 Repeater in Serie geschaltet werden.

Detailinformationen finden Sie im Handbuch Installationskomponenten für RS 485-Netzwerke (Bestellnummer 26/740D).

## 2.5 Termination-Box PCD7.T160

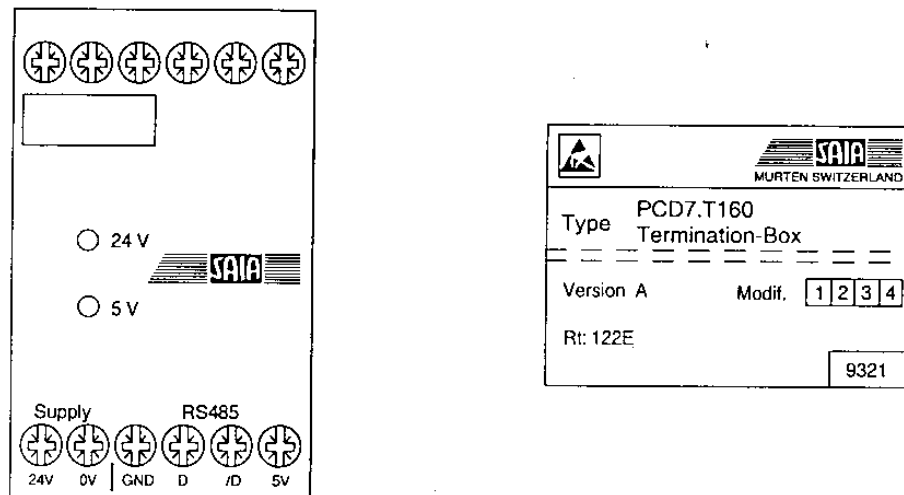


Bild 11: Frontplatte und Typenschild

Für eine sichere und servicefreundliche Nutzung eines RS485-Netzwerkes ist es wichtig, die Abschlusswiderstände separat und übersichtlich zu installieren. Diesem Zweck dient die Termination-Box, wo in Abhängigkeit der physikalischen Eigenschaften der Leitung der geeignete Abschlusswiderstand mittels 2 Steckbrücken (Jumper) eingestellt werden kann. Die Signalleitungen D und /D werden mit einer galvanisch getrennten Speisung auf das nötige Ruhepotential vorgespannt.

Detailinformationen finden Sie im Handbuch "Installationskomponenten für RS 485-Netzwerke" (Bestellnummer 26/740D).

Notizen:

## 3. Planung und Installation eines PROFIBUS-Netzwerkes

---

Die Realisierung eines PROFIBUS-Netzwerkes kann grob in folgende Schritte unterteilt werden:

- a) Planung und Installation des Netzwerkes
- b) Definition und Konfiguration des Netzes mit dem Konfigurator
- c) Erstellung des Anwenderprogrammes für den Datenaustausch
- d) Inbetriebnahme

In diesem Kapitel werden keine Details beschrieben, es soll lediglich ein Leitfaden und Gedankenstütze darstellen für die Vorgehensweise bei der Planung und Installation eines PROFIBUS-Netzwerkes. Die technischen Detailinformationen finden Sie im Handbuch "Installations-Komponenten für RS 485-Netzwerke" (Bestellnummer 26/740D) sowie in den folgenden Kapiteln dieses Handbuches.

### 3.1 Planung und Installation des Physical Layers

---

- Erstellung eines Übersichtschemas
  - Bestimmung der max. Netzwerk-Distanz
  - Bestimmung der max. Segment-Distanz
  - Bestimmung des Kabeltyps
  - Bestimmung der max. Baudrate
  - Eventuell Repeater vorsehen
  - Bestimmen der Anfangs- und Endstation
- PCD7.T160 Abschlussboxen vorsehen
- Für eine korrekte Installation des PROFIBUS, sind unbedingt die Anweisungen im Handbuch "Installationskomponenten für RS 485-Netzwerke" zu beachten.

## 3.2 Logische Netzwerk-Struktur

---

### 3.2.1 Bestimmung der Busparameter

- Baudrate
- Höchste Stationsnummer
- GAP-Update-Faktor
- Target Rotation Time
- usw.

<b>Wichtig:</b> Die Busparameter müssen für alle Teilnehmer am Bus identisch sein.
--

### 3.2.2 Festlegen der Kommunikationsbeziehungen

Nachdem der physikalische Aufbau beschrieben ist, müssen nun die logischen Kommunikationskanäle definiert werden.

- Gerätetypen und deren Funktion (Master/Slave)
- Teilnehmeradressen
- Verbindungskanäle
- Verbindungstyp (Zyklisch/Azyklisch)
- Service Access Points (SAP)
- Zugriffsrechte
- usw.

### 3.2.3 Erstellen der Objektverzeichnisse

Hier wird geplant, welche Daten respektive Prozessobjekte über den Bus ausgetauscht werden sollen.

Der Projektierer erstellt ein Source-Objektverzeichnis für jede einzelne Station. Das Objektverzeichnis enthält die Kommunikationsobjekte, welche eine Station anderen zur Verfügung stellt, sowie eine vollständige oder teilweise Kopie der Objekte von der Partnerstation.

Während des Betriebes hat jede Station via eines definierten Kommunikationskanales Zugriff auf die Objekte einer anderen Station.

### 3.2.4 Überprüfung respektive Abschätzung der geforderten Leistungswerte

- Bestimmen der Reaktionszeiten
- Bestimmen der Netzbelastung



## 4. Der PROFIBUS-Konfigurator

---

Die Definition und Konfiguration (Busparameter, Kommunikationsbeziehungsliste und Objektverzeichnis) eines PROFIBUS-Netzwerkes kann je nach Grösse des Projektes recht umfangreich sein. Diese Aufgabe wird dem Anwender durch die Verwendung des PROFIBUS-Konfigurators erheblich erleichtert. Mit dem Konfigurator können alle SAIA-Geräte konfiguriert werden.

### 4.1 Allgemeines

---

Der Konfigurator besteht aus einer Software, die unter MS-Windows 3.1 und höher läuft. Es wird keine spezielle Hardware benötigt. Die Fenster-technik wird durchgehend angewandt. Daraus resultiert eine angenehme Übersicht und die Parametereingabe wird sehr anwenderfreundlich.

Um auch weniger gewandten Anwendern die Konfigurierung auf einfache Weise zu ermöglichen, erscheinen nur die wichtigsten Parameter in den einzelnen Fenster. Um die Gesamtheit aller Parameter zu erreichen, kann in den meisten Fenster in einen sogenannten "Advanced-Setup" weitergeschaltet werden.

Standardwerte werden überall dort angeboten, wo dies die Parametrierung zulässt. Selbstverständlich können aber auch diese Standardwerte angepasst werden.

Für alle Parameter ist ein Wertebereich vorgegeben.

Der Konfigurator erzeugt schlussendlich eine ASCII-Textdatei mit den Definitionstexten für alle PROFIBUS-Kanäle einer Station. Diese Textdatei wird im SASI-Befehl des PROFIBUS-Kanals verwendet.

Als Dokumentation wird eine übersichtliche Liste mit allen Parametern erzeugt. Es kann daraus die genaue Konfiguration der Station im PROFIBUS-Netzwerk ersehen werden. Auch eine Liste aller Kommunikationskanäle und die definierten Objekte kann dieser Liste entnommen werden.

## 4.2 Vorgehensweise für die Konfiguration

---

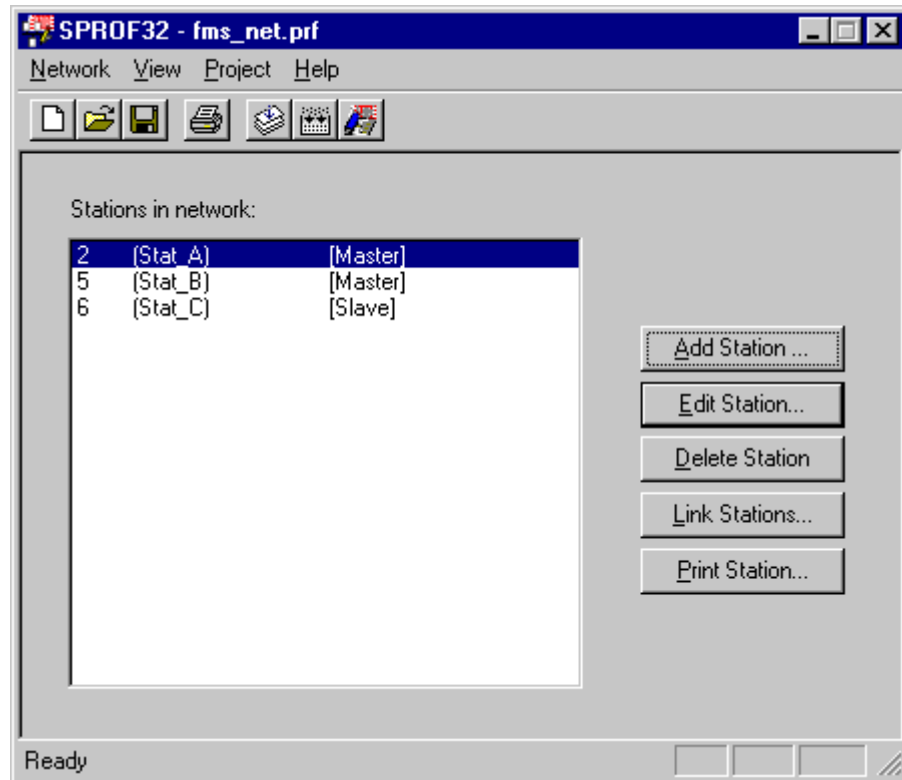
Die Vorgehensweise kann in folgende Schritte unterteilt werden:

1. Definition der Busparameter
2. Definition der Stationen
3. Definition der Kommunikationskanäle
4. Erstellen der Kommunikationsverbindungen (Link stations)
5. Erstellen der Kommunikationsobjekte
6. Generierung der SASI-Texte für PCD-Stationen (Output Station)
7. Speichern der Konfiguration
8. Erzeugung der Dokumentation

## 4.3 Standard-Setup

### 4.3.1 Konfigurator-Hauptmenü

In diesem Basisfenster des Konfigurators werden alle Stationen, welche untereinander Daten austauschen, aufgelistet.



**Network:** In diesem Menü werden Konfigurationen neu eröffnet, bestehende geöffnet sowie Konfigurationen abgespeichert. Auch das Fenster "Bus-Definition..." wird von hier aus erreicht.

**Options:** Zugang, um den Offset der Textbasisadresse für die Konfigurationstexte zu ändern (Standardwert: 3000)

**Help:** Hilfedatei für den Anwender (nicht implementiert).

**Add Station...:** Zugang zum Fenster "Station Definition". Dient zum Zufügen einer neuen Station ins Netzwerk.

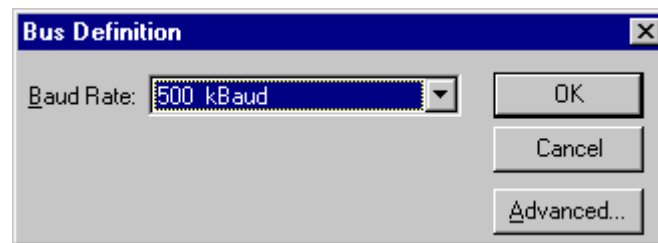
**Edit Station...:** Zugang zum Fenster "Station Definition". Dient zum Editieren einer bereits definierten Station.

**Delete Station:** Entfernen einer Station aus dem Netzwerk.

**Link Station...:** Zugang zur Funktion "Link Stations".

**Print Station...:** Druckt die Konfigurations-Parameter der gewählten Station aus.

### 4.3.2 Bus-Definitionen



**Baud Rate:** Es kann zwischen den folgenden Übertragungsgeschwindigkeit (Baud Rate) gewählt werden:

9.6, 19.2, 38.4, 93.75 (nur PCD7.F700), 187.5 (nur PCD7.F700) und 500 kbps

Die Busparameter sind für alle Stationen im Netzwerk identisch.

**Advanced...:** Zugang zum erweiterten Setup

### 4.3.3 Stations-Definitionen

**Station name:** Name der Station

**Address:** Adresse der Station im Netzwerk

**Station Type:** Wahl zwischen Master oder Slave

**Object list:** Verzeichnis der Kommunikations-Objekte der Station

**Channels list:** Verzeichnis der virtuellen Kanäle der Station

**Object list:** S-OD = Static Object Dictionary  
S-OV = Statisches Objekt-Verzeichnis

Verzeichnis der Kommunikations-Objekte der Station.

**Channels list:** CRL = Communication Relationship list  
KBL = Kommunikations-Beziehungsliste

Verzeichnis der virtuellen Kanäle der Station.

#### 4.3.4 Kommunikations-Kanäle

Bei PROFIBUS müssen die Stationen, welche untereinander Daten austauschen wollen, Kommunikationsbeziehungen zueinander unterhalten. Jede Station hat somit eine eigene **Kommunikationsbeziehungsliste (KBL)**, in welcher alle definierten Kommunikationsverbindungen eingetragen sind. Der Aufbau der Kommunikationsbeziehungsliste entspricht einer Tabelle, bei der für jede projektierte Kommunikationsbeziehung eine Zeile mit den entsprechenden Daten existiert. Der Zugriff auf die eingetragenen Verbindungen geschieht über den Zeilenindex, die sogenannte **Kommunikationsreferenz (KR)**. Bei unserem Konfigurator entspricht die **Kanalnummer** der Kommunikationsreferenz (KR). Der Kommunikationskanal ist ein virtueller Übertragungsweg über den Informationen ausgetauscht werden können.

Ein Kanal enthält folgende Informationen:

- Adresse der Partnerstation
- Die verwendeten **Service Access Points** (LSAP und RSAP)
- Verbindungstyp (zyklisch, azyklisch, usw.)
- Liste der erlaubten Dienste (z.B. Read/Write, usw.)
- Die Grösse der benötigten Datenpuffer (PDU-Länge)

Für den Kommunikationskanal gelten folgende Regeln:

- Derselbe Kanal kann nur für eine definierte Verbindung verwendet werden.
- Bei einer offenen Verbindung, kann ein Kanal für Verbindungen mit mehreren Partnergeräten verwendet werden (wird oft bei nicht-konfigurierbaren Slaves angewandt).
- Für dieselbe Station können bis zu 90 (10 bei PCD2 mit PCD7.F700) Kanäle definiert und parallel betrieben werden.

#### Service Access Point (SAP)

Ein SAP bildet die logische Schnittstelle zwischen der Anwenderschicht (FMS-Ebene) und der darunter liegenden FDL-Schicht (Field Data Link).

Es wird unterschieden zwischen

- LSAP (Local Service Access Point) = SAP der eigenen Station und
- RSAP (Remote Service Access Point) = SAP der Partnerstation

Es gelten folgende Regeln:

- Poll SAP = gemeinsamer SAP für alle Slavestationen (Default Poll SAP = 128)
- Bei einer definierten Master-Master-Verbindung kann der gleiche SAP nur einmal für eine einzige Verbindung (Kanal) verwendet werden.
- Für eine offene Verbindung gilt sinngemäss die gleiche Regel wie für den Kommunikationskanal.

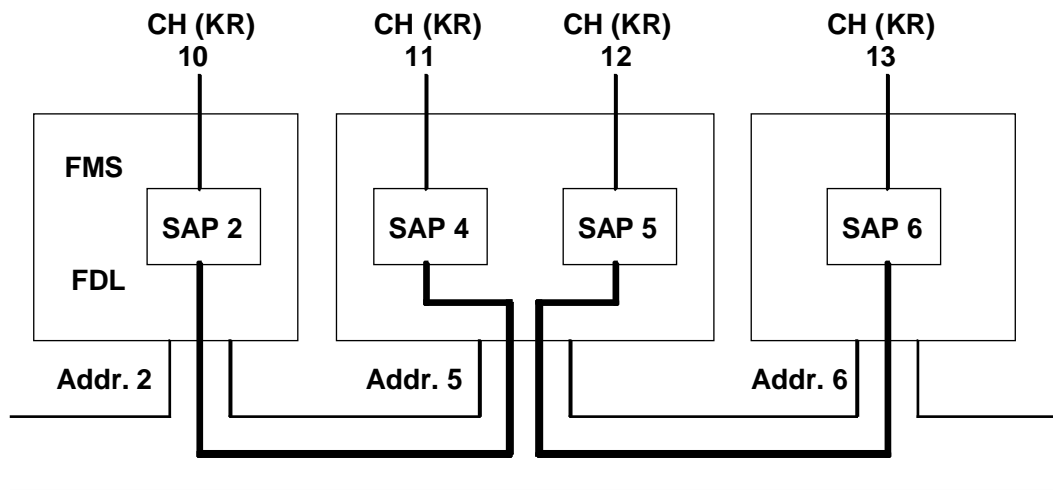


Bild 12: Beispiele Kommunikationsbeziehungen mit Kanal- und SAP-Nummern.

### Kanal-Definition (channel definition)

In diesem Fenster werden die Parameter des Kommunikations-Kanals der Station festgelegt.

#### Channel number:

Kanal Nummer. Dieser Parameter wird für die Definition des SASI-Textes verwendet.

#### Source Service Access Point (SAP):

Dienstzugangspunkt der Station

#### Diagnostic bit field:

Dies entspricht den Diagnose-Elementen <dia\_elem> im SASI-Text für die allgemeine Kommunikation: Basis-Adresse von 8 Flags (oder Ausgängen).

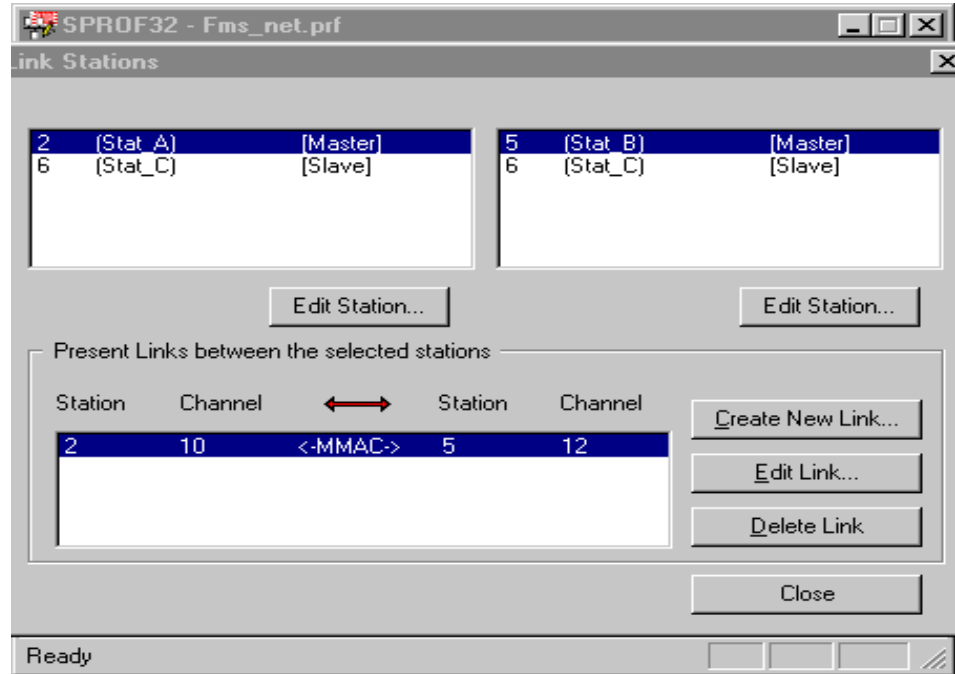
#### Diagnostic register address:

Dies entspricht dem Diagnose-Register <dia\_reg> im SASI-Text für die allgemeine Kommunikation: Adresse eines PCD-Registers.



### 4.3.5 Link-Fenster

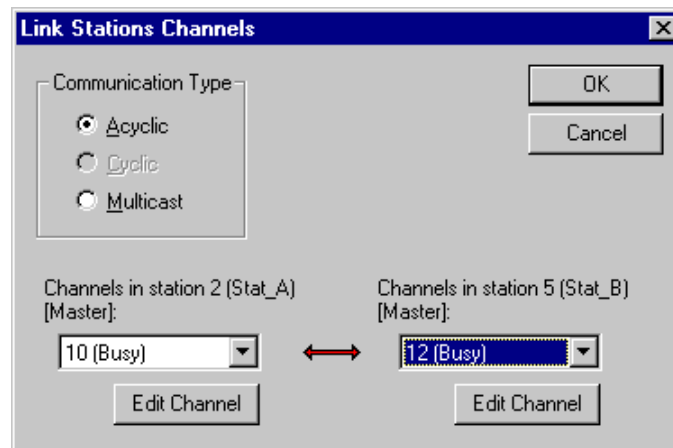
Es muss nun noch die virtuelle Kommunikationsverbindung (Kommunikationsbeziehung) definiert werden. Es stehen zu diesem Zweck die Fenster "Link Stations" und "Link Stations Channels" zur Verfügung.



Im Fenster "Link Stations" werden in den beiden oberen Teil-Fenster je eine Station gewählt die zu verbinden sind. Das untere Teil-Fenster zeigt die bereits existierenden Verbindungen zwischen diesen beiden Stationen. Es können nun Verbindungen aufgehoben, modifiziert oder neu definiert werden. Bei den aufgelisteten Kommunikationsbeziehungen ist in der Mitte jeweils der Verbindungstyp angegeben.

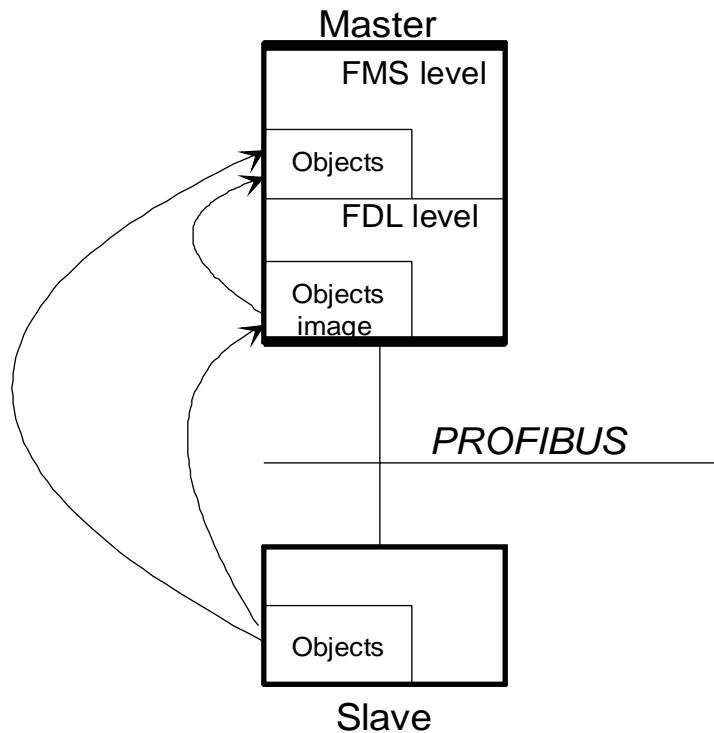
Beispiel oben: MMAC → Master-Master-Azyklisch

Die Option 'Create new Link' und 'Edit Link' führen zum Fenster 'Link Stations Channels'.



Im vorangehenden Fenster (Link Stations) wurde die Verbindung auf Stationsebene durchgeführt. Hier geschieht die Verbindung auf Kanalebene.

- **Communication Type**



**Azyklischer Datenverkehr** bedeutet, dass die Daten spontan, bei der Abarbeitung eines SRXM/STXM-Befehls zum Partnergerät übertragen werden.

**Zyklischer Datenverkehr** kommt meistens nur bei Verbindungen mit einfachen Slavegeräten (z.B. Sensoren) zur Anwendung. Sobald der erste STXM/SRXM-Befehl abgearbeitet wurde, werden in der Folge die angegebenen Objekte von der FDL-Schicht zyklisch ausgetauscht. Dazu wird im Speicher der FDL-Schicht ein Abbild der Objekte angelegt. Für den Anwender ist zwischen zyklischem und azyklischem Datenverkehr kein direkter Unterschied feststellbar.

Der Vorteil im zyklischen Datenverkehr liegt im schnellen Zugriff auf Objekte bei Echtzeitanwendungen unter Verwendung der speziell dafür vorgesehenen Dienste.

Allerdings wird durch den zyklischen Datenverkehr die Kommunikation auf dem Bus erheblich belastet. Aus diesem Grunde soll zyklischer Datenverkehr nur für zeitkritische Signale angewendet werden.

### 4.3.6 Objekt-Definitionen

PROFIBUS sieht eine objektorientierte Arbeitsweise vor, damit auch die unterschiedlichsten Geräte untereinander Daten austauschen können.

Bevor also Daten ausgetauscht werden können, müssen dem System die Kommunikationsobjekte und deren Bedeutung bzw. Struktur bekannt gegeben werden. Die Definition der Objekte geschieht für alle Busteilnehmer im Konfigurator.

Eine Station führt jeweils eine vollständige oder teilweise Kopie der Objekte ihrer Partnerstation.

Beispiel:

Die Station A soll die Flags 1000-1007 auf die Ausgänge 0-7 der Station B übertragen. Es sind 2 Objekte zu definieren:

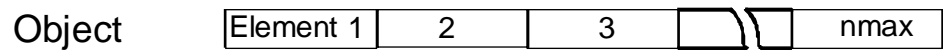
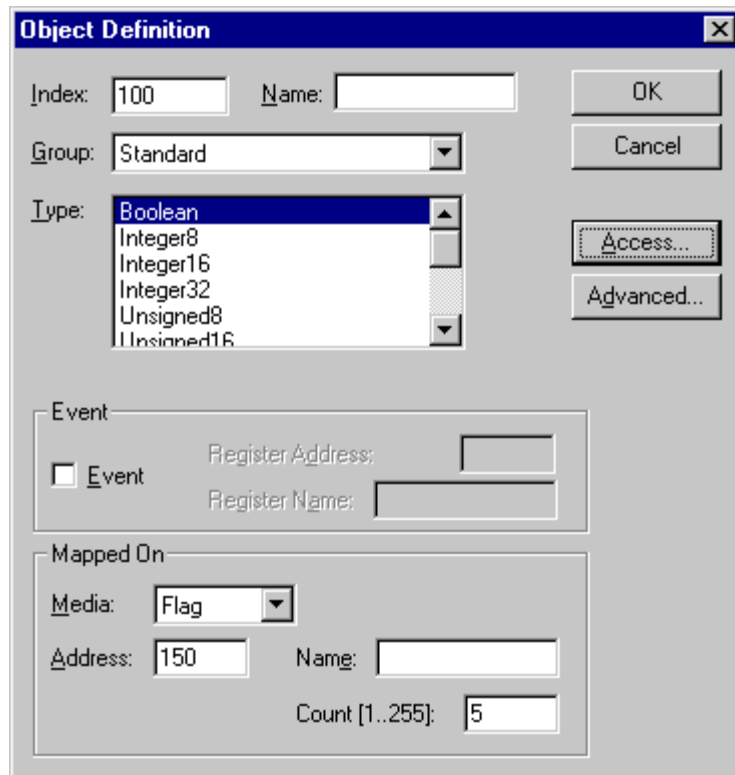
1. Ein Objekt xx für die Station A:  
Paket von 8 Elementen vom Typ Boolean die als Flag 1000-1007 definiert werden.
2. Ein Objekt yy für die Station B:  
Paket von 8 Elementen vom Typ Boolean die als Ausgänge 0-7 definiert werden.

Es ist nun möglich, dass Station A das folgende Telegramm sendet: Schreibe (sende) mein Objekt xx zum Objekt yy der Station B.

Es gelten folgende Regeln:

- Die Objekte einer Station sind grundsätzlich (Zugriffsrechte können definiert werden) allen Busteilnehmern via die Kommunikationskanäle zugänglich.
- Bei der Übertragung von Objekten müssen das Source- und Quellobjekt kompatible Datenstrukturen und -Formate aufweisen.
- Es ist darauf zu achten, dass die Objektgröße (Anzahl Bytes) die definierte Puffergröße (PDU-Länge) des verwendeten Kanals nicht übersteigt.

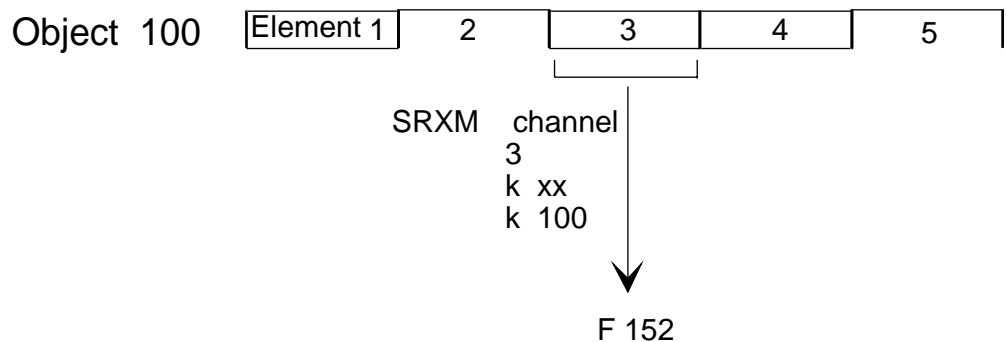
- **Objektformat für Datentypen:**  
**Boolean, Integer-8/16/32, Unsigned-8/16/32 und Floating Point**



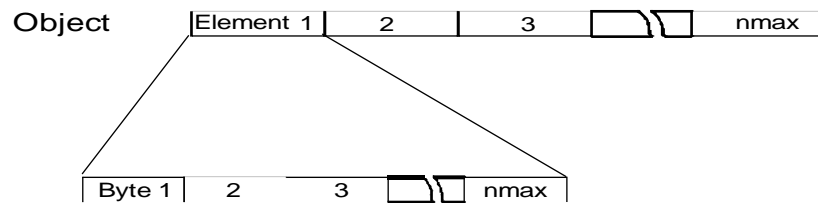
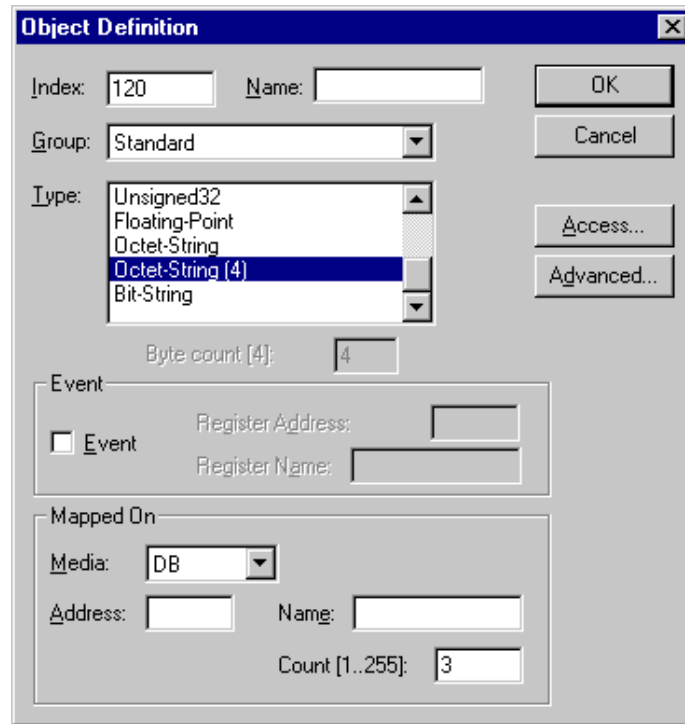
wobei: nmax = count (Anzahl Elemente)

Beispiel:

Object index:            100  
 Element-Typ:            Boolean  
 Count:                    5  
 Media-Adresse:         Flag 150



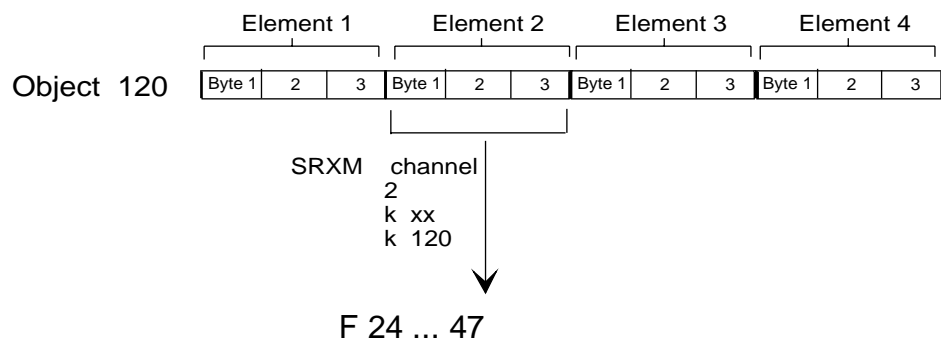
• **Objektformat für Datentypen: Bit String und Octet String**



- wobei:
- Element-nmax = count (Anzahl Elemente)
  - Byte-nmax = Byte count (Anzahl Byte pro Element)

Beispiel:

Object index: 120  
 Element-Typ: Bit String  
 Byte count: 3  
 Count: 4  
 Media address: Flag 0



Für mehr Informationen über Datentypen siehe Anhang A.

## 4.4 Erweiterter Setup

### 4.4.1 Definition der Bus-Parameter



**Busparameter müssen für alle Busteilnehmer identisch sein !**

Die Default-Werte für die Busparameter sind abhängig von der Baudrate. Die nachfolgenden Angaben gelten für eine Baudrate von 500 kBd.

#### Gap Update Factor (G):

Wertebereich: 1..99

in Vielfachen der maximalen Target Rotation Time ( $T_{TR}$ )

Default-Wert: 1

Der Parameter definiert die Zeit, in Vielfachen der Target Rotation Time, nach welcher geprüft wird, ob neue Teilnehmer am Bus hinzugekommen sind. Der Parameter hat nur bei aktiven Stationen (Master) eine Bedeutung.

$$\text{GAP-Aktualisierungszeit} = G * T_{TR}$$

Es gilt folgende Regel:

Je grösser  $G * T_{TR}$ , desto grösser ist der Datendurchsatz auf dem Bus.

**Highest Station Address (HSA):**

Wertebereich: 2..126

Default-Wert: 126

Der Parameter legt die höchste Teilnehmeradresse am Bus fest. Beim Überprüfen, ob neue aktive Stationen in den logischen Tokenring eintreten wollen, wird nur bis zu dieser Adresse hin abgefragt (sogenannte GAP-Aktualisierung).

Bei einer nachträglichen Erhöhung der Teilnehmerzahl muss dieser Parameter bei allen aktiven Geräten geändert werden. Bei zeitkritischen Anwendungen empfiehlt es sich, den kleinstmöglichen Wert einzustellen, weil damit der Datendurchsatz wesentlich erhöht werden kann.

**Maximum Retry Limit:**

Wertebereich: 1..8

Default-Wert: 1

Der Parameter gibt an wie oft ein Schicht-2-Telegramm wiederholt wird, wenn die Antwort nicht innerhalb der Slot-Time eingetroffen ist. In stark störbehafteter Umgebung kann mit diesem Parameter ein sicherer Betrieb erreicht werden.

Der Parameter ist nur bei aktiven Teilnehmern notwendig.

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

Wertebereich: 1..24565 Bitzeiten

Default-Wert: 3500 = 3500 \* 2 us = 7 ms

Der Parameter definiert das Timeout für ein ausgesendetes Schicht-2-Telegramm.

Der Master wartet auf die Antwort oder Quittung eines Aufruftelegrammes bis die Slot Time abgelaufen ist. Bei dem Parameter handelt es sich um eine reine Überwachungszeit und hat somit keinen Einfluss auf den Datendurchsatz.

Zur Bestimmung des Parameters kann folgende Faustregel angewendet werden:

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

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

Wertebereich: 0..255 Bitzeiten

Default-Wert: 22 = 22 \* 2 µs = 44 µs

Durch diesen Parameter können Repeater-Umschaltzeiten bzw. Modulator-Ausklingzeiten berücksichtigt werden.

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

Wertebereich: 0..255 Bitzeiten

Default-Wert: 50 = 50 \* 2 µs = 100 µs

Dieser Parameter definiert die Zeit, die vom Eintreffen eines Ereignisses (z.B. letztes Telegrammzeichen ausgesendet) bis zur Ausführung der erforderlichen Reaktion (z.B. freigeben des Empfängers) verstreicht.

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

Wertebereich: 1..6288384 Bitzeiten

Default-Wert: 100'000 = 100'000 \* 2 µs = 200 ms

Aus der Target Rotation Time (oder Token-Soll-Umlaufszeit) berechnet sich der jeweils aktive Teilnehmer seine eigene maximale Token-Haltezeit.

Der Parameter ist nur bei aktiven Teilnehmern von Bedeutung.

Mit dieser Zeit wird die Systemreaktionszeit stark beeinflusst.

Faustregel zur Bestimmung dieses Parameters:

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

nb<sub>M</sub>: number of master stations in the network

Die Regel basiert auf der Annahme, dass jeder Master beim Erhalt des Tokens ein Telegramm senden kann.



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

Wertebereich: 1..24565 Bitzeiten

Default-Wert: 500 = 500 \* 2  $\mu$ s = 1 ms

Der Parameter wird im Wesentlichen durch die Verarbeitungsgeschwindigkeit der Anschaltung (Interpretation des Aufrufs, Zusammenstellen der Antwort/Quittung) und die Telegrammlänge bestimmt. Mit dem Parameter kann das Senden der Antwort/Quittung verzögert werden und richtet sich nach dem langsamsten Master (Telegramm-Initiator). Erst nach dieser Zeit ist dieser bereit zum Empfang der Quittung.

Es gilt:

- Min\_T<sub>SDR</sub> wird bestimmt durch den langsamsten Master (Telegramm-Initiator)
- Min\_T<sub>SDR</sub> > T<sub>QUI</sub> (Quiet time)

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

Wertebereich: 1..24565 Bitzeiten

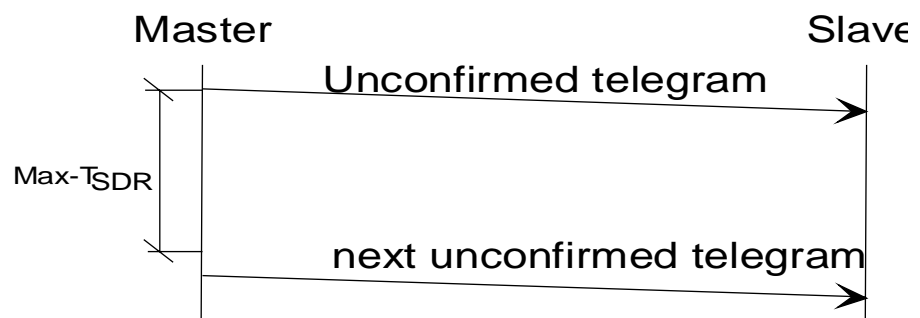
Default-Wert: 1000 = 1000 \* 2  $\mu$ s = 2 ms

Der Parameter wird im Wesentlichen durch die Verarbeitungsgeschwindigkeit der Anschaltung (Interpretation des Aufrufs, Zusammenstellen der Antwort/Quittung) und die Telegrammlänge bestimmt.

Es werden zwei Fälle unterschieden:

a) Unbestätigte Sende-Telegramme:

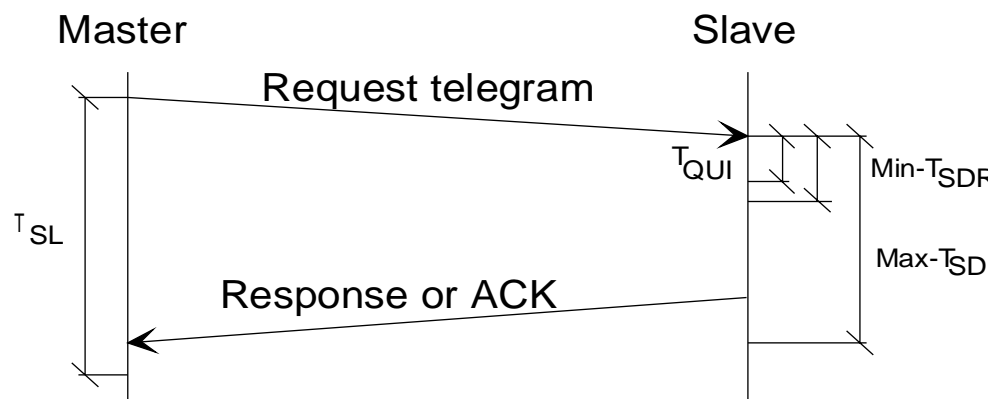
Mit dem Parameter kann nach einem gesendeten unbestätigtem (unconfirmed) Telegramm eine Ruhepause definiert werden bis das nächste Telegramm gesendet wird. Der einzustellende Wert richtet sich nach dem langsamsten Telegramm-Empfänger. Für diesen Fall ist der Parameter nur bei aktiven Teilnehmern von Bedeutung.



Es gilt:  $\text{Max-T}_{\text{SDR}}$  wird bestimmt durch den langsamsten Slave.

b) Bestätigte Request-Telegramme:

Für diesen Fall ist der Parameter beim Slave von Bedeutung. Der Parameter definiert die maximale Zeit, welche nach einem empfangenen Request-Telegramm verstreichen darf bis eine Antwort oder Quittung zurückgesendet werden muss.



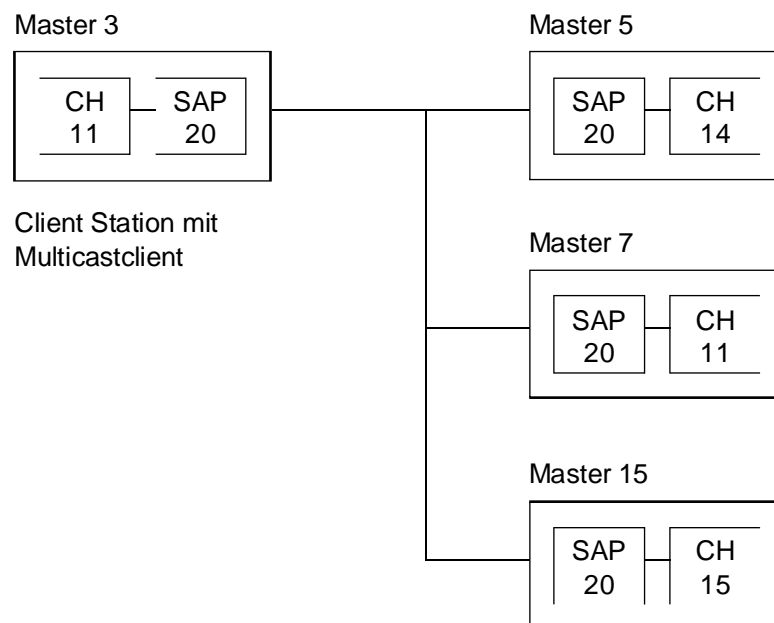
Es gilt:  $\text{Max-T}_{\text{SDR}} < T_{\text{SL}}$  (Slot time)

#### 4.4.2 Kanal-Definitionen

- **Kommunikationsbeziehungs-Typen**

Grundsätzlich unterscheidet PROFIBUS zwischen **verbindungslosen** und **verbindungsorientierten** Kommunikationsbeziehungen. Da bei einer verbindungslosen Kommunikationsbeziehung der logische Rückkanal fehlt, stehen dem Anwender hier nur die unbestätigten Dienste für Multi- bzw. Broadcast zur Verfügung.

Unbestätigte Dienste werden von SAIA-Geräten unterstützt.



Bestätigte Dienste müssen über verbindungsorientierte Kommunikationsbeziehungen übertragen werden, da der Rücklauf der Quittung aus Adressierungsgründen einen Rückkanal erfordert. Im Unterschied zur verbindungslosen Kommunikation muss vor der Übertragung von Daten die Verbindung in einer Verbindungsaufbauphase initialisiert (SCON-Befehl) und nach Beendigung der Datenübertragung wieder freigegeben werden.

Bei verbindungsorientierten Kommunikationsbeziehungen wird zusätzlich zwischen einer **definierten und offenen Verbindung** unterschieden.

- **Definierte Verbindung**

Bei dieser Verbindung wird der Kommunikationspartner bereits in der Projektierungsphase festgelegt.



Für dieses Beispiel sind bei Master 3 die Attribute für die Kommunikationsbeziehung des Kanals 10 wie folgt definiert:

Remote Adresse:	5
Lokaler SAP:	2
Remote SAP:	5
Verbindungsart:	definiert, azyklisch, Master/Master (MMAZ)

- **Offene Verbindung**

Diese Verbindungsart kommt hauptsächlich bei Slavegeräten zur Anwendung, welche nicht konfiguriert werden können. Offene Verbindungen können nur vom Partnergerät geöffnet werden.



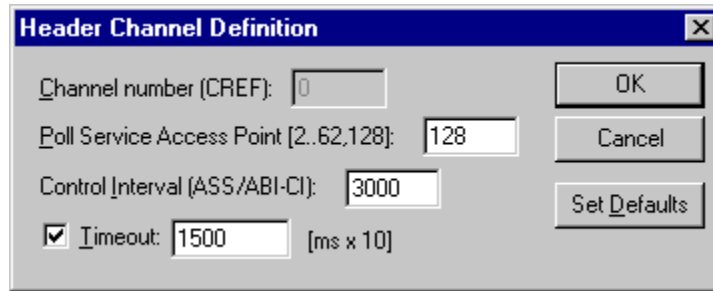
Für dieses Beispiel sind bei Slave 3 die Attribute für die Kommunikationsbeziehung des Kanals 10 wie folgt definiert:

Remote Adresse:	255
Lokaler SAP:	2
Remote SAP:	255
Verbindungsart:	offen, azyklisch

Die Werte 255 für Remote Adresse und RSAP sind Globaladressen und ermöglichen einem Partnergerät mit beliebiger Adresse und einem beliebigen SAP diese Verbindung zu eröffnen. Dies ermöglicht, dass mehrere Mastergeräte denselben SAP benutzen können. Es kann jedoch immer nur eine dieser Verbindungen gleichzeitig geöffnet sein. \*)

- \*) Offene Verbindungen können im Konfigurator wie folgt definiert werden:
  - a) Für einen definierten Kanal wird keine Verbindung (Link zu einem anderen Kanal) erstellt.
  - b) Für denselben Kanal werden mehrere Verbindungen (Links) zu Kanälen von unterschiedlichen Stationen erstellt.

• **Header Channel Definition**



Die Kommunikationsbeziehungsliste (KBL) verfügt über einen Header mit Angaben, die für alle Kommunikationsverbindungen gelten.

**Channel number:** CREF = Communication Reference  
 KR = Kommunikations-Referenz

Wertebereich: 0

Referenznummer für den Header, kann nicht geändert werden.

**Poll SAP:**

Wertebereich: 2..62, 128, 255

Default-Wert: 128, bei einer Master-Station  
 255, bei einer Slave-Station

Dieser Parameter gibt an, über welchen SAP die Pollliste abgearbeitet werden soll, falls eine solche vorhanden ist. Der Poll-SAP wird für alle Kommunikationsbeziehungen zu passiven Teilnehmern verwendet.

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

Wertebereich:  $0..(2^{32}-1) * 10 \text{ ms}$   
 Default-Wert:  $3000 = 3000 * 10 \text{ ms} = 30 \text{ s}$

Timeout-Wert für Verbindungsauf- und -abbau (SCON-Befehl). Der eingestellte Wert gilt für alle Verbindungskanäle.

Faustregel zur Bestimmung des Parameters:  
 $\text{Control interval} \gg G * T_{TR}$

**Timeout:**

Wenn während dieser Zeit, nach einem STXH/SRXH-Befehl, keine Antwort von der Partnerstation empfangen wurde, wird der Kanal geschlossen und der Dienst annulliert.

**Überwachungszeit:**

Defaultwert: 0: 15 sek  
 1 .. 1638375: multipliziert mit 10 ms = Timeoutwert mit einer Auflösung von 250 ms.

Beispiel: 1 .. 25 = Timeout von 250 ms  
 26 .. 50 = Timeout von 500 ms  
 1638376: = kein Timeout

- **Advanced Channel Definition**

**Advanced Channel Definition**

Fms features supported:

Request:

Read  Unsolicited Status  
 Info Report  Write  
 Event Notification

Response:

Get OV  Unsolicited Status  
 Read  Write  
 Info Report  Event Notification

Maximum service counters:

Send confirmed (SCC) [1..15]: 1  
 Receive confirmed (RCC) [1..15]: 1  
 Send acknowledged (SAC) [1..15]: 1  
 Receive acknowledged (RAC) [1..15]: 1

Maximum PDU length:

High priority message:  
 Received [0,3..242]: 241 Sent [0,3..242]: 0

Low priority message:  
 Received [31..242]: 241 Sent [31..242]: 241

Acyclic/Cyclic control interval (ACI/CCI): 0

OK  
 Cancel  
 Set Defaults  
 Access rights...

**FMS features supported:**

Definiert die für diesen Kanal unterstützten FMS-Dienste .

Request: Dienste für Anfrage-Telegramme  
Response: Dienste für Antwort-Telegramme  
(Get OV: lese Objektverzeichnis, ist immer unterstützt)

Der Responder (Server oder Slave) muss mindestens alle Dienste des Requesters (Master) unterstützen. Diese Bedingung wird beim Verbindungsaufbau überprüft.

Info Report: Dienste für Broadcast, Multicast

**Maximum service counters:**

Send confirmed (SCC):

Wertebereich: 0..15  
Default-Wert: 1

Dieser Parameter gibt an, wieviele ausstehende Antworten der Teilnehmer als Client auf diesem Kanal maximal verarbeiten kann. Bei einem reinem Server wird der Wert 0 eingetragen.

Bei einem Master mit zyklischer Verbindung wird der Wert 0 eingetragen.

Es gilt:  $SCC \leq RCC$  des Partners

Von SAIA-Geräten werden parallele Sende-Dienste nicht unterstützt, deshalb ist ein grösserer Wert als 1 sinnlos und hat keinen Einfluss.

Receive confirmed (RCC):

Wertebereich: 0..15  
Default-Wert: 15

Dieser Parameter gibt an, wieviele bestätigte Dienste (Anfragen) der Teilnehmer als Server auf diesem Kanal maximal annehmen kann, ohne eine Antwort zu senden.

Bei einem Kanal mit zyklischer Verbindung wird der Wert 0 eingetragen.

Es gilt:  $RCC \geq SCC$  des Partners

Send unconfirmed (SAC) (acknowledged):

Default-Wert: 0

Maximal zulässige Anzahl der unbestätigten, parallelen Dienste als Sender.

Unbestätigte und parallele Dienste werden von SAIA-Geräten nicht unterstützt.

Receive unconfirmed (RAC) (acknowledged):

Default-Wert: 0

Maximal zulässige Anzahl der unbestätigten, parallelen Dienste als Empfänger.

Unbestätigte Dienste werden von SAIA-Geräten nicht unterstützt.

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

Wertebereich: 0

Telegramme mit hoher Priorität werden nicht unterstützt.  
Nur unbestätigte Telegramme sind hochprioritär.

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

Wertebereich: 0, 31..242 Byte

Default-Wert: 241 Byte

**Received (MPRL):**

Dieser Parameter gibt die maximal zulässige PDU-Grösse (Protocol Data Units) der FMS-Telegramme mit niedriger Priorität für den Empfänger an.

Es gilt:

Max.-PDU-Received > = Max.-PDU-Sent des Partners

Diese Bedingung wird bei einem Initiate (SCON-Befehl) überprüft.



**Sent (MPSL):**

Dieser Parameter gibt die maximal zulässige PDU-Grösse (Protocol Data Units) der FMS-Telegramme mit niedriger Priorität für den Sender an.

Es gilt:

Max.-PDU-Sent  $\leq$  Max.-PDU-Received des Partners

Diese Bedingung wird bei einem Initiate (SCON-Befehl) überprüft.

**Acyclic/Cyclic control interval:**

## a) Zyklische Verbindungen (CCI):

Wertebereich:  $0..(2^{32}-1)$  in Einheiten von 10 ms

Default-Wert: 3'000

Dieser Wert gibt das Überwachungsintervall für zyklische Verbindungen an.

Die Überwachung ist beim Master einer zyklischen Verbindung zwingend vorgeschrieben, beim Slave optional.

Regel: bei einem Master soll der Wert immer  $> 0$  sein

## b) Azyklische Verbindungen (ACI):

Wertebereich:  $0..(2^{32}-1)$  in Einheiten von 10 ms

Default-Wert: 3'000 für Master-Master Verbindungen  
0 für Master-Slave Verbindungen

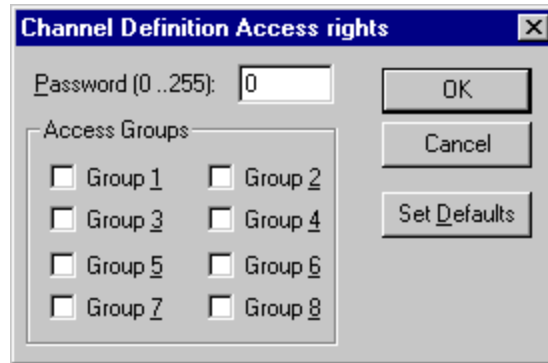
Dieser Wert gibt das Zeitintervall für die Verbindungsüberwachung auf azyklischen Verbindungen an. Wenn keine Nutztelegramme auf dieser Verbindung übertragen werden, dann wird von jedem der beiden Teilnehmer dreimal pro Intervall ein Idle-Telegramm gesendet. Empfängt einer der beiden Teilnehmer innerhalb des Intervalls keine Nutz- oder Idle-Telegramme, so schliesst er den Kommunikationskanal.

Es gilt: Das Intervall muss bei beiden Partnern identisch sein. Dies wird beim Verbindungsaufbau überprüft.

Regel: ACI  $> 0$  für Master-Master Verbindungen  
ACI = 0 für Master-Slave Verbindungen

- **Channel Definition Access rights**

Jedem Kommunikationskanal kann ein Passwort mit verschiedenen Zugriffsgruppen zugeordnet werden. Damit wird der Zugriff auf geschützte Objekte beim Partnergerät ermöglicht.



**Password (PW):**

Wertebereich: 0..255  
Default-Wert: 0 = kein Passwort

Das hier für diesen Kanal gültige Passwort wird beim Verbindungsaufbau (SCON-Befehl) zum Partner geschickt. Wird später über diese Verbindung auf ein Objekt zugegriffen, so prüft der Partner, ob das beim Verbindungsaufbau empfangene Passwort mit dem, beim Objekt eingetragenen Passwort übereinstimmt.

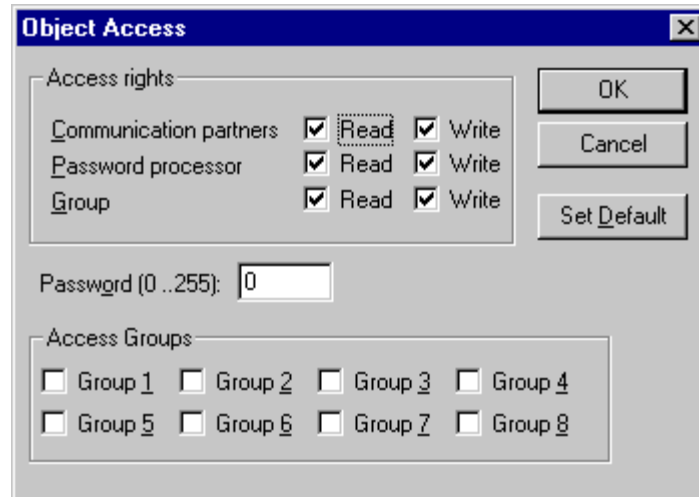
**Access groups (GR):**

Zusätzlich zum Passwort können einem Kanal insgesamt acht verschiedene Zugriffsgruppen zugeordnet werden. Die Zugriffsgruppen werden ebenfalls beim Verbindungsaufbau mitgeteilt und beim späteren Zugriff auf Objekte vom Partner überprüft.

### 4.4.3 Objekt-Definitionen

- **Objekt-Zugriffsrechte (Object access)**

Zugriffsrechte können im Konfigurator zwar eingestellt, werden jedoch von der SAIA-PROFIBUS-Firmware nicht unterstützt.



Für jedes Objekt können verschiedene Zugriffsrechte definiert werden, welche mit den Zugriffsrechten des Partnergerätes beim Eröffnen einer Verbindung verglichen werden.

**Access rights:**

- **Communication partners:**  
Definiert den Zugriff für Partnergeräte ohne Passwort.
- **Password processor:**  
Definiert den Zugriff für Partnergeräte mit Passwort.
- **Group:**  
Definiert den Zugriff für Partnergeräte, die einer bestimmten Gruppe (1..8) angehören.

**Password:**

Wertebereich: 0..255

Default-Wert: 0 = kein Passwort

Dem Objekt kann ein Passwort zugeordnet werden. Nur die Geräte mit dem gleichen Passwort, haben die im Password Processor definierten Zugriffsrechte.

**Access Groups:**

Hier können dem Objekt zusätzlich zum Passwort, auch noch bestimmte Zugriffsgruppen zugeordnet werden.

**Advanced Object Definitions:**

The screenshot shows a dialog box titled "Object Advanced" with a close button (X) in the top right corner. It is divided into two main sections: "Read Indicator" and "Write Indicator".

**Read Indicator:**

- Radio buttons:  None,  Flag,  Output
- Address [0..8191]:  Name:

**Write Indicator:**

- Radio buttons:  None,  Flag,  Output
- Address [0..8191]:  Name:

On the right side of the dialog, there are two buttons: "OK" and "Cancel".

Für jedes Objekt können Les- und Schreib-Instruktionen definiert werden. Durch die Abfrage dieser Instruktionen kann im Anwenderprogramm festgestellt werden, ob eine Partnerstation die eigenen Objekte beschrieben oder gelesen hat. \*)

**Read Indicator**

Indikation für Lesezugriff

**Write Indicator**

Indikation für Schreibzugriff

- \*) Werden die einzelnen Objekte von einer Partner-Station gelesen oder geschrieben, so wird der Indikator auf "H" gesetzt. Das Zurücksetzen auf "L" muss der Anwender im PCD-Programm selbst programmieren.

## 5. Programmierung

---

### 5.1 Beschreibung der PCD-Befehle (SASI, SCON, STXM und SRXM)

---

Folgende Befehle werden bei PROFIBUS unterstützt:

SASI	Assign serial interface Assigniere den Kommunikationskanal
SASII	Like SASI but indirect Wie SASI, jedoch indirekt
SCON	Serial connect Öffne den Kommunikationskanal
SCONI	Like SCON but indirect Wie SCON, jedoch indirekt
SRXM	Serial receive media Empfange Daten (Objekte) von einer Partner-Station
SRXMI	Serial receive media indirect Empfange Daten (Objekte) von einer Partner-Station
STXM	Serial transmit media Sende Daten (Objekte) zu einer Partner-Station
STXMI	Serial transmit media indirect Sende Daten (Objekte) zu einer Partner-Station

Bei der Beschreibung des Konfigurators wurden zwei Eigenheiten des PROFIBUS gegenüber der bisher bekannten Kommunikation (S-Bus, Mode-D) aufgezeigt:

1. Mit PROFIBUS erfolgt die Telegrammübertragung über Kommunikations-Kanäle. Diese Methode erlaubt mehrere Kanäle zu ein- und derselben Partner-Station. Jeder Kanal erlaubt unterschiedliche Dienste, z.B. zyklische und azyklische Verbindung. Es werden daher im PCD-Befehl nicht die Adresse der Partner-Station, sondern der Index des Kanals, über welchen die Informationen ausgetauscht werden sollen, angegeben.
2. Zur Übertragung von Daten werden Objekte verwendet. Die Schreib- und Lesebefehle enthalten den Index dieser Objekte und nicht die zu übertragenden Medien (I,O,F,R) selbst. Der Zusammenhang zwischen den Objekten und den Medien ist der Firmware dank der "Objekt-Definition" im SASI-Befehl bekannt und wird automatisch richtig eingesetzt.

### 5.1.1 SASI Assignierung eines Kanals

**Beschreibung:** Die Assignierung des PROFIBUS geschieht, wie die meisten Kommunikations-Modi der PCD, mit dem Befehl SASI. SASI ruft einen Text auf, der die für die Initialisierung eines virtuellen Kanals notwendige Information enthält. Soll über mehrere virtuelle Kanäle kommuniziert werden, so ist für jeden Kanal ein eigener SASI-Befehl zu editieren.

**Aufbau:**

SASI	Kanal Text
------	---------------

Kanal      10..99      für PCD4.M445  
                  10..19      für PCD2 mit PCD7.F700

Text            T\_As\_xx      Symbolischer Name des Textes, welcher die Information für die Assignierung enthält. (Wird vom PROFIBUS-Konfigurator generiert).

**Beispiel:**      SASI            12            ; Initialisiere Kanal 12  
    T\_As\_12      ; Definitionstext für Kanal 12

**Flags:**            Das Error (E) Flag wird gesetzt, falls der Definitionstext fehlt oder ungültig ist.

**Diagnose:**        Die Diagnose einer PROFIBUS-Kommunikation erfolgt in der für PCD bekannten Weise, d.h. jedem Kommunikationskanal sind 8 Flags zur Grob- und ein Register zur Fein-Diagnose zugewiesen. Die Definition dieser Diagnose-Elemente erfolgt im Konfigurator.

**Diagnose-Flags:**

Adresse	Name	Beschreibung
xxxx	RBSY	Receiver busy Empfänger Busy
xxxx+1	RFUL	Receive buffer full Empfangsbuffer voll
xxxx+2	RDIA	Receiver diagnostic Empfänger-Diagnose
xxxx+3	TBSY	Transmitter busy Sender Busy
xxxx+4	TFUL	Nicht benützt
xxxx+5	TDIA	Transmitter diagnostic Sender-Diagnose
xxxx+6	XBSY	Cross busy Kanal geöffnet
xxxx+7	NEXE	Not executed Nicht ausgeführt

**Receiver Busy (RBSY)** nicht benützt.

**Receive Buffer Full (RFUL)**

Client: nicht benützt.

Server: High zeigt an, dass ein Schreibtelegramm empfangen und somit PCD-Media geschrieben wurden.

**Receiver Diagnostic (RDIA)**

Wird high gesetzt, wenn während dem Empfang eines Telegrammes ein Fehler festgestellt wird. Die detaillierte Beschreibung des Fehlers kann dem Diagnoseregister (Bit 0..15) entnommen werden. Das Flag wird zurückgesetzt, sobald im Diagnoseregister alle Empfängerdiagnose-Bit (0..15) zurückgesetzt sind.

**Transmitter Busy (TBSY)**

Client: Wird high gesetzt, solange eine Übertragung statt findet. Dies ist der Fall bei der Ausführung eines STXM-, SRXM- oder SCOM-Befehls.

Wird zurückgesetzt, wenn Antwort oder Quittung empfangen wird.

Server: nicht benützt

**Transmitter Diagnostic (TDIA)**

Wird high gesetzt, wenn während dem Senden eines Telegrammes ein Fehler festgestellt wird. Die detaillierte Beschreibung des Fehlers kann dem Diagnoseregister (Bit 16..31) entnommen werden. Das Flag wird zurückgesetzt, sobald im Diagnoseregister alle Senderdiagnose-Bit (16..31) zurückgesetzt sind.

**Not Executed (NEXE)**

Wird high gesetzt, wenn ein Befehl (STXM oder SRXM) nach dreimaligem Versuch nicht ausgeführt werden konnte. Das Flag wird beim nächsten Befehl wieder zurückgesetzt.

Client:       Dienstanforderer (Master)  
 Server:       Diensterbringer (Master oder Slave)

**Diagnose-Register:**

	Bit	Bezeichnung	Beschreibung
E M P F Ä N G E R	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9	Format error	Ungültiger Befehl
	10		
	11		
	12	Range error	Ungültiges Objekt
	13		
	14		
S E N D E R	15		
	16		
	17		
	18		
	19		
	20	Negative response	Negative Antwort (NAK)
	21	No response	keine Antwort nach Timeout
	22		
	23		
	24		
	25	Format error	Ungültiger Befehl
	26		
	27		
	28	Range error	Ungültiges Objekt oder Element ausserhalb des Bereichs
	29		
	30	Response error	Antwort entspricht nicht der Anfrage
	31	Program error	Programm Fehler: STXM/SRXM-Befehl nicht erlaubt



**Format error (Bit 9)**

Wird high gesetzt, wenn das Dualport-RAM einen ungültigen Befehl (< > read, write, init, abort) enthält.

**Range error (Bit 12)**

Wird high gesetzt, wenn ein ungültiges Objekt empfangen wird:

- Bei einem "Get\_OV-i" existiert das Objekt nicht. (Index ausserhalb des Bereichs oder bei der Assignierung nicht definiert.)
- Bei einem "Read\_i" existiert das Quellobjekt nicht. (Index ausserhalb des Bereichs oder bei der Assignierung nicht definiert.) oder Sub-Index > Anzahl Elemente des Objekts.
- Bei einem "Write-i" existiert das Zielobjekt nicht. (Index ausserhalb des Bereichs oder bei der Assignierung nicht definiert.) oder Sub-Index > Anzahl Elemente des Objekts oder Anzahl empfangener Bytes < > Anzahl Bytes des Objekts.

**Negative response (Bit 20)**

Wird high gesetzt, wenn

- bei Empfang einer negativen Bestätigung nach einem "Initiate" (Öffnen eines Kanals. Zu überprüfen: PDU-Länge, FMS-Eigenschaften, OV-Version, Passwort, Profil-Nummer)
- bei Empfang einer negativen Bestätigung bei einem "Confirmed service" (read, write). Zu überprüfen: Quell- und Destinationsobjekt sowie Sub-Index.

**Format error (Bit 25)**

Wird high gesetzt, wenn der "taco"-Buffer eine ungültigen Befehl (< > read, write, init, abort) enthält.

**No response (Bit 21)**

Wird auf high gesetzt, wenn nach einem STXM, SRXM-Befehl innerhalb der Timeout-Zeit keine Antwort empfangen wurde. (Nur wenn der Kanal offen ist). Der Kanal wird danach geschlossen und der nicht ausgeführte Befehl wird abgebrochen.

**Range error (Bit 28)**

- Bei einem SRXM bzw. einem STXM:
  - a) Das Objekt wurde beim Assignieren nicht definiert.
  - b) Der Index des Objekts ist ausserhalb des Bereichs (Index < K 100 oder Index > K 499).
  - c) Der Index des Objekts des Partners ist grösser als K 16383
  - d) Der Sub-Index ist grösser als 255
  - e) Der Sub-Index ist grösser als die Anzahl Elemente des Objekts.
- Bei einem "Read\_c" entspricht die Anzahl empfangener Bytes nicht der Anzahl Bytes des Objekts.

**Response error (Bit 30)**

Wird high gesetzt, wenn die empfangene Antwort nicht der Anfrage entspricht.

- Der Dienst des Telegramms für die Bestätigung stimmt nicht mit dem Anfragetelegramm überein (Init, read, write).
- Die "Invoke\_ID" des Bestätigungs-Telegramms stimmt nicht mit der "Invoke\_ID" der Anfrage überein (read\_C, write\_c).

**Program error (Bit 31)**

Wird high gesetzt, bei der Ausführung eines nicht erlaubten Befehls.

- STXM- oder SRXM-Befehl bei einem geschlossenen Kanal (XBSY abfragen).
- Ausführung eines SCON-, STXM- oder SRXM-Befehls, wenn bereits ein solcher Befehl bearbeitet wird (TBSY-Flag wurde vor der Ausführung des Befehls nicht abgefragt).

### 5.1.2 SCON: Öffnen eines Kommunikationskanals

**Beschreibung:** Bevor über einen Kanal Telegramme übertragen werden können, ist dieser Kanal zu öffnen. Dies geschieht mit dem Befehl SCON, der bereits beim LAN1 die gleiche Funktion hat.

Nicht notwendig bei Broadcast und Multicast.

**Aufbau:**

SCON	Kanal (Station) Status
------	------------------------------

Kanal            10..99 für PCD4.M445  
                   10..19 für PCD2 mit PCD7.F700

(Station)        0        Nicht verwendet. Dieser Parameter ist für PROFIBUS bereits in der Definition des Kanals enthalten.

Status 0 / 1        0 = Schliessen des Kanals  
                          1 = Öffnen des Kanals

**Beispiel:**

```

STL            TBSY
ANL            XBSY
JR             L next
SCON        10            ; Kanal 10
                 0            ;
                 1            ; öffnen
    
```

next:

**Flags:** Das Error-Flag wird gesetzt, falls der Kanal nicht assigniert ist.

**5.1.3 SRXM: Empfange Daten (Objekte) von einer Partner-Station**

**Beschreibung:** Mit diesem Befehl werden Daten von einer Partner-Station gelesen.

**Aufbau:**

SRXM	Kanal	
	Count	
	Quelle	; remote = Partner-Station
	Ziel	; local = eigene Station

Kanal 10..99 für PCD4.M445  
 10..19 für PCD2 mit PCD7.F700

Count 0..255 Sub-Index des Quell- und des Ziel-Objekts.  
 Bemerkung: Dies zwingt dazu, die gleiche Struktur für beide Objekte einzuhalten.  
 0 = Gesamtes Objekt übertragen.

Quelle K 0..16383 Index des Quellen-Objekts (Partner-Station)

Ziel Index des Ziel-Objekts (eigene Station)  
 K 100..499 PCD4.M445  
 K 100..199 PCD2 mit PCD7.F700

**Beispiel:**

```

STL      TBSY
ANH      XBSY
JR       L next
SRXM   13      ; Kanal 13
          0      ; Ohne Sub-Index
          K 22   ; Kopiere Objekt 22 von der
                ; Partner-Station
          K 150  ; auf Objekt 150 der eigen-
                ; en Station

next:
    
```

**Flags:** Das Error-Flag wird gesetzt, falls der Kanal nicht assigniert ist.

**5.1.4 SRXMI: Empfange indirekt Daten (Objekte) von einer Partner-Station**

**Beschreibung:** Mit diesem Befehl werden Daten (Objekte) von einer Partner-Station gelesen. Die Adressierung der Operanden erfolgt für den Kanal wahlweise direkt oder indirekt. Für die anderen Operanden ist die Adressierung immer indirekt über Register.

Der Befehl wird ab folgenden Firmware-Versionen unterstützt:

PCD4.M445 V005  
PCD2.M120 V003

Utility-Version:

V1.9 oder höher

Der Befehl kann nicht indexiert oder parametrisiert werden.

**Aufbau:**

SRXMI	Kanal	; 10..99, R 0..4095
	Count	; R 0..4095
	Quelle	; K [R 0..4095]
	Ziel	; K [R 0..4095]

Kanal 10..99 für PCD4.M445  
10..19 für PCD2 mit PCD7.F700  
oder R 0..4095 für indirekte Adressierung

Count R 0..4095 Das Register enthält ein Sub-Index 0..255 des Quell- und des Ziel-Objekts (0 = gesamtes Objekt).

Quelle K [R 0..4095] Register enthält Index 0..16383 des Quellen-Objekts (Partner-Station)

Ziel K [R 0..4095] Register enthält Index des Ziel-Objekts (eigene Station)  
PCD4.M445, Index: 100..499  
PCD2.M120, Index: 100..199

**Beispiel:**

```

LD      R 50      ;
        12        ; Kanal 12
LD      R 51
        0         ; Sub-index = 0
LD      R 52
        172       ; Quell-Objekt Index 172
LD      R 53
        145       ; Ziel-Objekt Index 145
STL     TBSY
ANH     XBSY
JR      L next

SRXMI  R 50      ; Kanalnummer in R 50
        R 51      ; Sub-index in R 51
        K 52      ; Quell-Objekt Index in R 52
        K 53      ; Ziel-Objekt Index in R 53
    
```

next:

**Flags:** Das Error-Flag wird gesetzt, falls der Kanal nicht assigniert ist.

**5.1.5 STXM:      Sende Daten (Objekte) zu einer Partner-Station**

**Beschreibung:** Mit diesem Befehl werden Daten zu einer Partner-Station geschrieben.

**Aufbau:**

STXM	Kanal	
	Count	
	Quelle	; local = eigene Station
	Ziel	; remote = Partner-Station

Kanal      10..99      für PCD4.M445  
             10..19      für PCD2 mit PCD7.F700

Count      0..255      Sub-Index des Quell- und des Ziel-Objekts.  
                             Bemerkung: Dies zwingt dazu, die gleiche Struktur für beide Objekte einzuhalten.  
                             0 = Gesamtes Objekt übertragen

Quelle                      Index des Quell-Objekts (eigene Station)  
                             K 100..499      PCD4.M445  
                             K 100..199      PCD2 mit PCD7.F700

Ziel                      K 0..16383      Index des Ziel-Objekts (Partner-Station)

**Beispiel:**

STL            TBSY  
 ANH           XBSY  
 JR             L next

**STXM**      11                      ; Kanal 11  
             3                      ; Sub-Index (Element) 3  
             K 122                  ; Kopiere Objekt 122 der  
                                     ; eigenen Station  
             K 150                  ; auf Objekt 150 der Partner-  
                                     ; Station

next:

**Flags:**

Das Error-Flag wird gesetzt, falls der Kanal nicht assigniert ist.

### 5.1.6 STXMI: **Sende indirekt Daten (Objekte) zu einer Partner-Station**

**Beschreibung:** Mit diesem Befehl werden Daten (Objekte) zu einer Partner-Station geschrieben.

Die Adressierung der Operanden erfolgt für den Kanal wahlweise direkt oder indirekt. Für die anderen Operanden ist die Adressierung immer indirekt über Register.

Der Befehl wird ab folgenden Firmware-Versionen unterstützt:

PCD4.M445 V00C  
PCD2.M120 V004

Utility-Version:

V1.9 oder höher

Der Befehl kann nicht indexiert oder parametrisiert werden.

**Aufbau:**

STXMI	Kanal	; 10..99, R 0..4095
	Count	; R 0..4095
	Quelle	; K [R 0..4095]
	Ziel	; K [R 0..4095]

Kanal	10..99 10..19 oder R 0..4095	für PCD4.M445 für PCD2 mit PCD7.F700 für indirekte Adressierung
-------	------------------------------------	---

Count	R 0..4095	Register enthält ein Sub-Index 0..255 des Quell- und des Ziel-Objekts (0 = gesamtes Objekt).
-------	-----------	--

Quelle	K [R 0..4095]	Register enthält Index des Quellen-Objekts (eigene Station) PCD4.M445, Index: 100..499 PCD2.M120, Index: 100..199
--------	---------------	---

Ziel	K [R 0..4095]	Register enthält Index 0..16383 des Ziel-Objekts (Partner-Station).
------	---------------	---



**Beispiel:**

```

LD      R 100      ;
          12        ; Kanal 12
LD      R 101
          0        ; Sub-index = 0
LD      R 102
          172      ; Quell-Objekt Index 172
LD      R 103
          45       ; Ziel-Objekt Index 45
STL     TBSY
ANH     XBSY
JR      L next

STXMI  R 100      ; Kanalnummer in R 100
          R 101      ; Sub-index in R 101
          K 102      ; Quell-Objekt Index in R 102
          K 103      ; Ziel-Objekt Index in R 103

```

next:

**Flags:** Das Error-Flag wird gesetzt, falls der Kanal nicht assigniert ist.

## 5.2 Regeln für die Erstellung des Anwenderprogrammes

Zu Beginn noch einmal die wichtigsten Regeln für das PROFIBUS-Anwenderprogramm:

- Das Anwenderprogramm für die PROFIBUS-Kommunikation muss beim Modul PCD4.M445 in die **CPU 1** geladen werden, da nur diese zum PROFIBUS-Controller Zugriff hat.
- Wie bei SAIA<sup>®</sup> PCD üblich, muss jede Kommunikationsschnittstelle (bei PROFIBUS der Kommunikationskanal) mit einem SASI-Befehl initialisiert werden. Dies geschieht normalerweise im XOB 16.
- Bevor über einen Kommunikationskanal Daten ausgetauscht werden können, muss dieser mit dem SCON-Befehl geöffnet werden.
- Ein Kanal soll nur von einer Station (normalerweise vom Client) geöffnet werden.
- SCON- und STXM/SRXM-Befehle dürfen nur abgearbeitet werden wenn das TBSY-Flag = low ist.  
→ Unmittelbar vor jedem Befehl TBSY-Flag abfragen.
- Datenaustausch ist nur über einen geöffneten Kanal möglich.  
→ Unmittelbar vor jedem STXM/SRXM-Befehl XBSY-Flag abfragen.
- TBSY-Flag muss immer vor dem XBSY-Flag abgefragt werden.

## 5.3 Struktur des Anwenderprogrammes

---

### 5.3.1 Kaltstartprogramm im XOB 16

Im XOB 16 wird die vom Konfigurator erzeugte Datei (\*.CON), welche die SASI-Texte enthält, mit \$INCLUDE in das Anwenderprogramm eingebunden.

In der Regel werden die Kommunikationskanäle ebenfalls im XOB 16 initialisiert.

Beispiel:

```
$INCLUDE Stat_A.CON ; Vom Konfigurator erzeugte
                    ; Datei
.                   ; Stat_A = Name der
                    ; 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 Hauptprogramm im COB

Im Hauptprogramm wird der SCON-Befehl programmiert um den Kommunikationskanal zu öffnen. Grundsätzlich kann der Kanal auch schon im XOB 16 geöffnet werden. Da der Kanal jedoch im Fehlerfall sowie jederzeit von der Partnerstation geschlossen werden kann, empfiehlt es sich, den Kanal in einem COB zyklisch zu überwachen und nötigenfalls mit dem SCON-Befehl zu öffnen.

Die Befehle für den Datenaustausch (STXM/SRXM) können sowohl in einem Umlaufprogramm wie auch in einer GRAFTEC-Struktur programmiert werden.

Beispiel 1:

Datenaustausch in BLOCTEC-Struktur ohne Auswertung der Diagnose-Elemente.

```

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 st.
      K  150        ; to dest. obj. 150 of the p.

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 p.
      K  250        ; to dest. object of own st.

      ....

END:    ECOB
    
```

Beispiel 2:

Datenaustausch in einer GRAFTEC-Struktur ohne Auswertung der Diagnose.

```

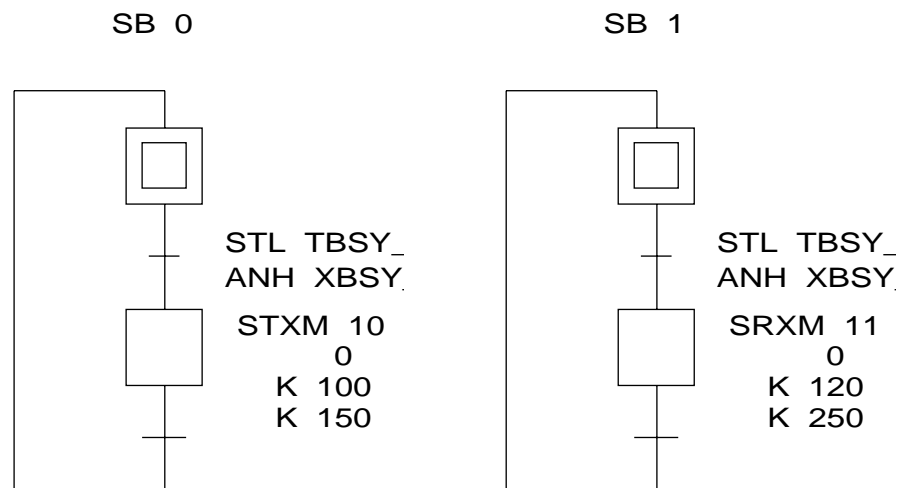
COB    0
      0

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

      ECOB
    
```

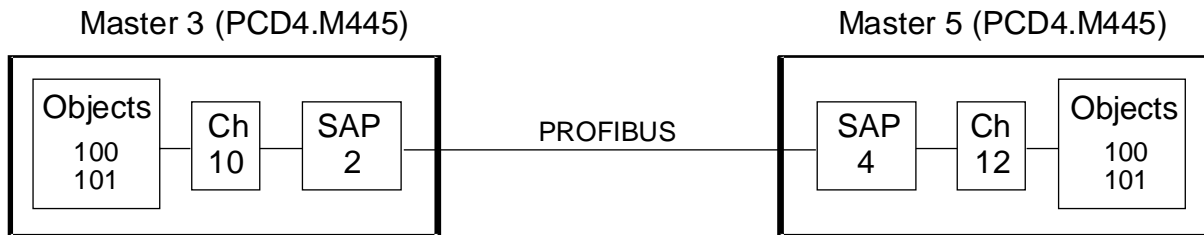


## 5.4 Programmbeispiele

Die Lösungen für die nachfolgend aufgeführten Beispiele sind auf Diskette erhältlich.

### 5.4.1 Beispiel 1

Netzwerk:



**Aufgabe:**

Beide Stationen übertragen die eigenen Eingänge 0..7 auf die Ausgänge 32..39 der Partnerstation. Der Kommunikationskanal wird von Master 3 geöffnet.

- **Definition der Kommunikationsverbindungen:**

**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 der Objekte:**

**Master 3:**

Local station				Remote station				
Object	Data typ/count	Access	Media	System	Object	Data typ/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 typ/count	Access	Media	System	Object	Data typ/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 Beispiel 2

Netzwerk:



**Aufgabe:**

Master 2 überträgt die eigenen Eingänge 0..7 auf die Ausgänge 16..23 des Weidmüller-Slaves. Die Eingänge 0..7 des Slaves sollen gelesen und auf die Ausgänge 32..39 kopiert werden.

- **Definition der Kommunikationsverbindungen:**

**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 der Objekte:**

**Master 2:**

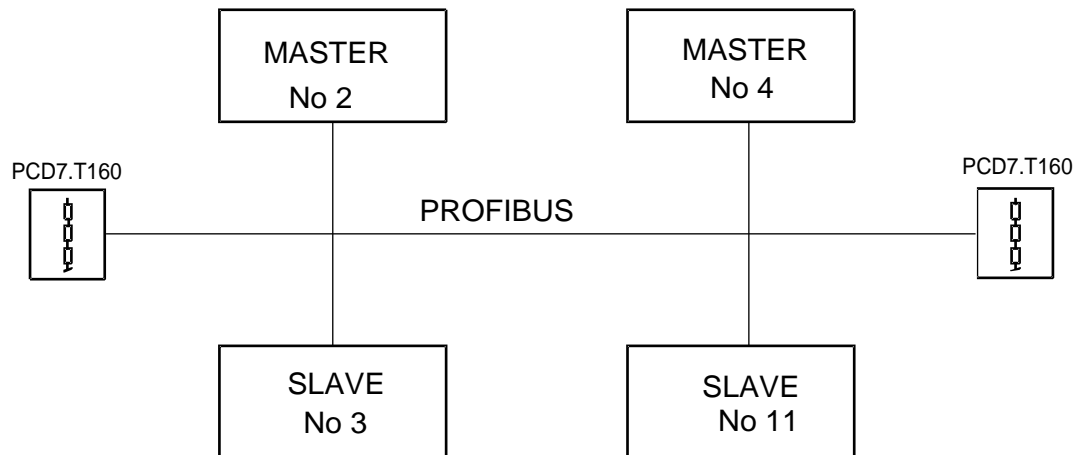
Local station				Remote station				
Object	Data typ/count	Access	Media	System	Object	Data typ/count	Access	Media
100	Octet-String/1	r/w	I 0..7	Slave 3	51	Octet-String/1	r/w	O 16..23
101	Octet-String/1	r/w	O 32..39	Slave 3	50	Octet-String/1	r/w	I 0..7

**Slave 3:**

Local station				Remote station				
Object	Data typ/count	Access	Media	System	Object	Data typ/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 Beispiel 3

#### Netzwerk:



#### Busteilnehmer:

- Master 2: PCD4.M445
- Master 4: PCD4.M445
- Slave 3: Weidmüller Remote I/O mit digitalen Ein/Ausgängen
- Slave 11: Festo Ventil-Insel mit digitalen Ein/Ausgängen

#### Aufgaben:

##### a) Master 2:

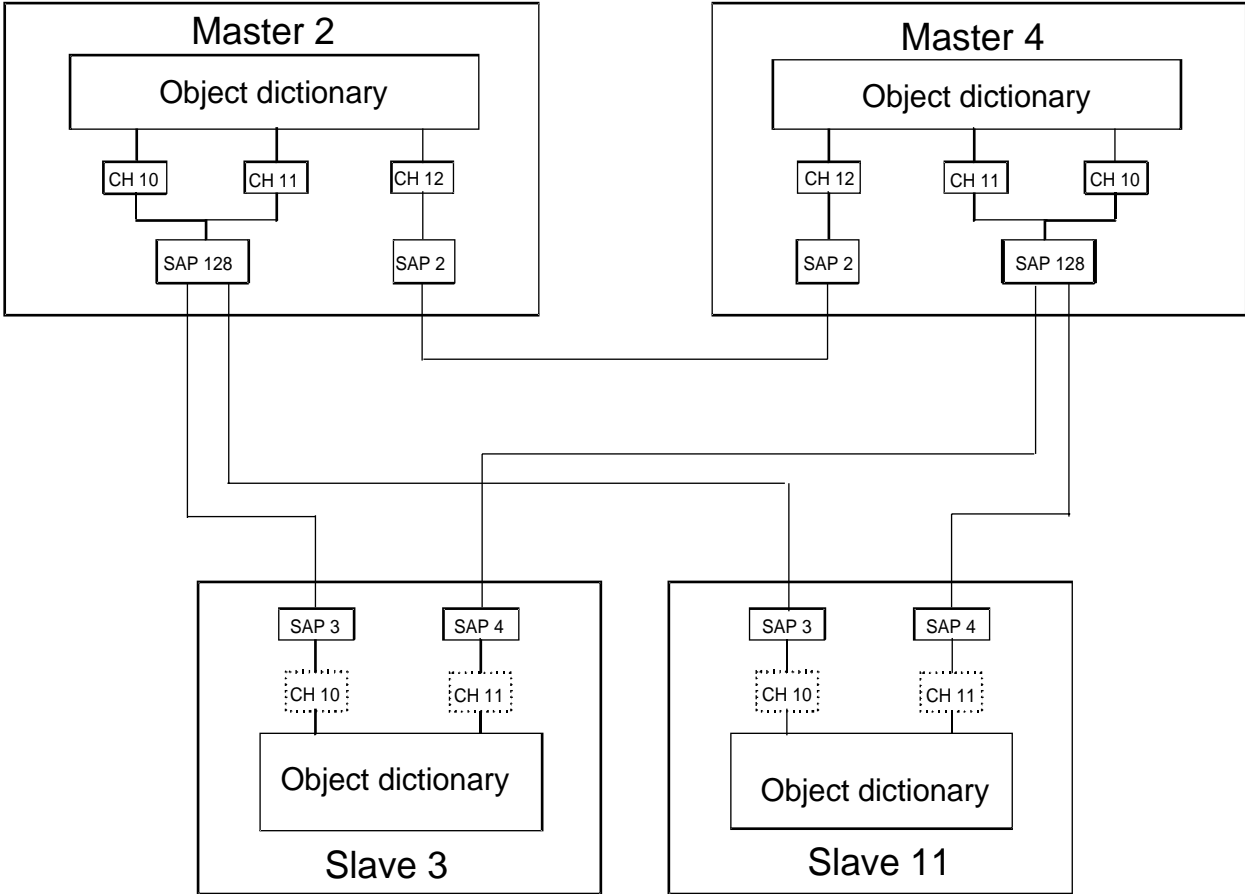
1. Register 150 auf R 2500 bei Master 4 kopieren.
2. Von Slave 3 Eingänge 0..7 lesen und auf Ausgänge 32..39 kopieren.
3. Von Slave 11 Eingänge 0..15 lesen und auf Ausgänge 48..63 kopieren.
4. Eingänge 0..7 auf Ausgänge 8..15 bei Slave 3 kopieren.
5. Öffnen des Kommunikationskanals zu Master 4.

##### b) Master 4:

6. Von Slave 3 Eingänge 0..7 lesen und auf Ausgänge 32..39 kopieren.
7. Von Slave 11 Eingänge 0..15 lesen und auf Ausgänge 48..63 kopieren.
8. Eingänge 0..15 auf Ausgänge 16..31 bei Slave 11 kopieren.



**Kommunikationsbeziehungen:**



**Definition der Kommunikationsverbindungen**

**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 der Objekte**

**Master 2 (PCD4.M445)**

Local station				Remote station				
Object	Data typ/count	Access	Media	System	Object	Data typ/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 typ/count	Access	Media	System	Object	Data typ/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 typ/count	Access	Media	System	Object	Data typ/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 typ/count	Access	Media	System	Object	Data typ/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

Notizen:

## 6. Inbetriebnahme eines PROFIBUS-Netzwerkes

---

Die Inbetriebnahme eines PROFIBUS-Netzwerkes vollzieht sich in zwei Schritten:

- a) Überprüfung und Test der Hardware-Installation (Physical Layer)
- b) Überprüfung und Test der Schicht 7 (Konfiguration und Datenaustausch)

### 6.1 Überprüfung und Test der Hardware-Installation (Physical Layer)

---

Erfahrungsgemäss liegen die Hauptursachen von Kommunikationsproblemen in einer mangel- oder fehlerhaft ausgeführten Hardware-Installation. Aus diesem Grunde muss diesem Teil der Inbetriebnahme eine wichtige Bedeutung beigemessen werden.

Folgende Prüfungen und Tests sollen gewissenhaft durchgeführt werden:

- Anschluss und Verlegung des Buskabels (Klemmen und Stecker, Schirmung, Stichleitungen, Verlegung von Leistungskabeln, usw.)
- Überprüfen der eingestellten Abschlusswiderstände, Repeater, ..
- Prüfen der Busleitung auf Durchgang
- Prüfen der elektrischen Signalpegel

usw.

Eine detaillierte Beschreibung dieser Prüfungen und Tests (auch statische Tests genannt) finden Sie im Handbuch "Installations-Komponenten für RS 485-Netzwerke" (Bestellnummer 26/740 D).

## 6.2 Überprüfung und Test der Schicht 7

---

Bei diesen Tests soll die Netzwerk-Konfiguration und das Anwenderprogramm für den Datenaustausch überprüft werden.

### 6.2.1 Verbindungsaufbau (Initiate)

Bei diesen Tests sollen die mit dem Konfigurator projektierten Kommunikationsbeziehungen (Verbindungskanäle) und deren eingestellte Parameter überprüft werden. Erfahrungsgemäss treten beim Verbindungsaufbau die grössten Probleme auf.

Der Verbindungsaufbau kann wie folgt überprüft werden:

1. Assignierung des Kanals durch manuelle Ausführung des SASI-Befehls. Dies wird im Debugger mit dem Befehl "Instruction..." oder durch Abarbeiten des Anwenderprogrammes in "Trace" (Einzelschrittausführung) erreicht. Dabei ist das Error-Flag zu beobachten. Im Fehlerfall sind folgende Punkte zu überprüfen:
  - Syntax des SASI-Befehls
  - Wurde im Konfigurator die Konfigurationsdatei (\*.CON) mit dem Befehl "Output Station" generiert?
  - Wurde diese Konfigurationsdatei im Anwenderprogramm eingebunden (\$Include \*.CON)?
  - Wurde das Anwenderprogramm assembliert, gelinkt und in die CPU 1 (nur bei PCD4.M445) geladen?
  
2. Überprüfung des Verbindungsaufbaus durch manuelle Ausführung des Befehls "SCON" (Debugger mit "Instruction..." oder "Trace"). Ein erfolgreich ausgeführter Verbindungsaufbau wird durch ein High gesetztes XBSY-Flag angezeigt. Wird das XBSY-Flag auch nach einigen Sekunden nicht High gesetzt, so sind folgende Punkte zu überprüfen:
  - Syntax des SCON-Befehls.
  - Wurde der entsprechende Kanal vorgängig assigniert?
  - Befindet sich die Partnerstation am Netz und ist eingeschaltet?
  - Wurde der Kanal vorher bei der Partner-Station assigniert?
  - Überprüfen der eingestellten und verwendeten Kanal- und SAP-Nummern.
  - Überprüfung der eingestellten Verbindungsaufbau-Überwachungszeit (Control Interval im Menü "Header Channel Definition")
  - Ist Bit 20 (Negative Response) im Diagnose-Register gesetzt?

wenn ja, sind folgende Ursachen möglich

- FMS-PDU-Grösse (max. message length) nicht kompatibel.
- FMS-Eigenschaften (FMS features) nicht unterstützt.
- Fehlerhaftes Passwort.
- Zeitintervall für Verbindungsüberwachung (cyclic control interval) inkompatibel.
- Profil-Nummer inkompatibel.
- Versionsnummer des Objekt-Verzeichnisse inkompatibel.

→ die vorangehend aufgeführten Punkte sind sowohl bei der eigenen wie auch bei der Partner-Station zu überprüfen (siehe dazu auch Kapitel 4.4.2 in diesem Handbuch).

### 6.2.2 Verbindungsabbau (Abort)

Verbindungen können sowohl vom Client, als auch vom Server unter Verwendung des Befehls "SCON" abgebaut werden. In besonderen Fehlersituationen wird der Verbindungsabbau auch durch die Protokollsoftware durchgeführt.

Folgende Fehlersituation führt unter anderen zu einem automatischen Verbindungsabbau:

- Ablauf der Verbindungsüberwachungszeit (ACI).

### 6.2.3 Überprüfung des Datenaustausches

Nach dem Verbindungsaufbau ist es in der Regel der Variablenzugriff, der zu den meisten Problemen führt. Oft sind Fehler im Anwenderprogramm oder in der Konfiguration die Ursache dafür.

Zur Überprüfung empfiehlt es sich, alle geplanten Variablenzugriffe im Debugger mit dem Befehl STXM resp. SRXM, ... oder durch Abarbeiten des Anwenderprogrammes in "Trace" (Einzelschrittausführung) manuell zu testen. Dabei sind die Diagnose-Elemente des verwendeten Kommunikationskanal zu beobachten. Für die möglichen Ursachen bei einer fehlerhaften Übertragung siehe auch Kapitel 5.1.1 "SASI-Diagnose" in diesem Handbuch.

### 6.2.4 Optimierung der Übertragungszeiten

Optimale Reaktions- und Übertragungszeiten werden durch Anpassen respektive Optimieren der Bus-Parameter erreicht. Dies ist allerdings keine triviale Angelegenheit und erfordert vom Anwender Fingerspitzengefühl und sehr gute PROFIBUS-Kenntnisse. Wenn also keine speziellen Anforderungen an die Übertragungszeiten bestehen, empfiehlt es sich, die Bus-Parameter auf den Standardwerten zu belassen. Diese sind so gewählt, dass eine Kommunikation mit den unterschiedlichsten Gerätetypen möglich sein sollte.

Sollte es trotzdem einmal notwendig sein, die Parameter anzupassen, so müssen unbedingt die Regeln und Empfehlungen in Kapitel 4.4.1 "Definition der Bus-Parameter" befolgt werden.

Im Anhang B finden Sie Beispiele, die den Einfluss der einzelnen Parameter auf die Reaktions- und Übertragungszeiten deutlich machen.

### 6.2.5 Verwendung von Busmonitoren

Führen die in den vorangehenden Kapiteln beschriebenen Prüfungen und Tests nicht zum gewünschten Erfolg, so ist gezwungenermassen ein Busmonitor für weiterführende Tests und Analysen einzusetzen.

Ein Busmonitor ist ein Testwerkzeug für die Inbetriebnahme, die Instandhaltung und die Diagnose von PROFIBUS-Netzwerken. Als passives Werkzeug beeinflusst dieser den Bus in keiner Weise und benötigt weder eine Stationsadresse, noch muss dieser während der Projektierung berücksichtigt werden.

Im Online-Betrieb zeichnet der Busmonitor den Telegrammverkehr auf, zeigt wahlweise die "Life-List" der am Bus angeschlossenen Stationen oder einige Bus-Charakteristiken an. Im Offline-Betrieb können die aufgezeichneten Daten ausgewertet und eine Telegrammanalyse auf Schicht 2 bzw. 7 des Kommunikationsmodells durchgeführt werden.



**Busmonitor der Firma TMG I-Tec (Deutschland)**

Der Bus-Monitor besteht aus einer MS-DOS-Software und einer PC-PROFIBUS-Steckkarte, welche die physikalische Verbindung zum Feldbus herstellt.

Mit dem Monitor können FMS- und DP-Telegramme mit Zeitstempel bis zu Baudraten von 500 kbps aufgezeichnet und analysiert werden. Umfangreiche und einstellbare Filterfunktionen ermöglichen eine detaillierte Fehlersuche und Fehleranalyse.

Mehr Informationen und Adressen können im elektronischen PROFIBUS-Produktkatalog (erhältlich bei der PROFIBUS-Nutzerorganisation) gefunden werden..

**Arbeiten mit dem Busmonitor**

Der Einsatz eines Busmonitors erfordert vom Anwender leider immer noch sehr weitgehende und tiefe PROFIBUS-Kenntnisse. Der Anwender muss mindestens in der Lage sein, die aufgezeichneten FMS-Telegramme zu interpretieren und zu verstehen, um eine Analyse und Fehlersuche durchführen zu können. Das heisst, er muss z.B. die Codierung der FMS-Dienste sowie die Context-Management-Dienste interpretieren können. Im Anhang C können Informationen über FMS-Dienste und deren Codierung sowie eine Beschreibung der Context-Management-Dienste mit den möglichen Fehler-Codes gefunden werden.

Notizen:

## Anhang A: Objekt-Datentypen

---

- **Boolean**

Wertebereich: 0 = FALSE  
1 = TRUE

Kopie von PCD-Medien → Sende-Telegramm

- Der Wert "False" wird im Telegramm als 00000000 übertragen.
- Der Wert "True" wird im Telegramm als 11111111 übertragen.

Medien-Wert (F/I/O):	False	True
	↓	↓
Sende-Telegramm:	00000000	11111111

Kopie von Empfangs-Telegramm → PCD-Medien

Empfangs-Telegramm:	xxxxxxx0	xxxxxxx1
	↓	↓
Medien-Wert (F/I/O):	False	True

Beim Empfang von einem Boolean-Telegramm hat nur das LSBit Bedeutung für den Medien-Wert.

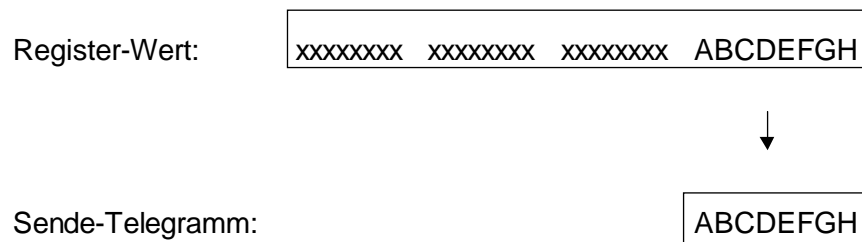
- **Integer-8/16/32**

Wertebereich: ....

Kopie von PCD-Medien (Register) → Sende-Telegramm

Bei den Datentypen Integer-8 und -16 werden die höherwertigen Bit vom Register ignoriert.

Beispiel: Integer-8



Kopie Empfangstelegramm → PCD-Medien (Register)

Bei den Datentypen Integer-8 und -16 werden die höherwertigen Bit im Register mit dem gleichen Wert wie das Vorzeichen-Bit geschrieben.

Beispiel: Integer-8



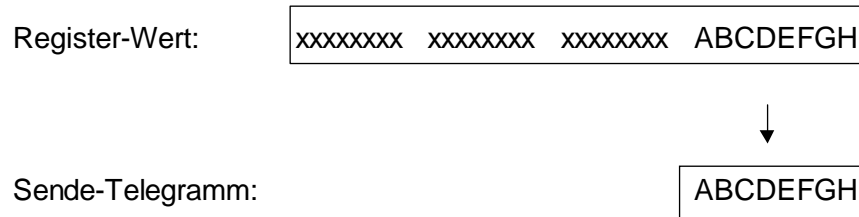
• **Unsigned-8/16/32**

Wertebereich: ....

Kopie von PCD-Medien (Register) → Sende-Telegramm.

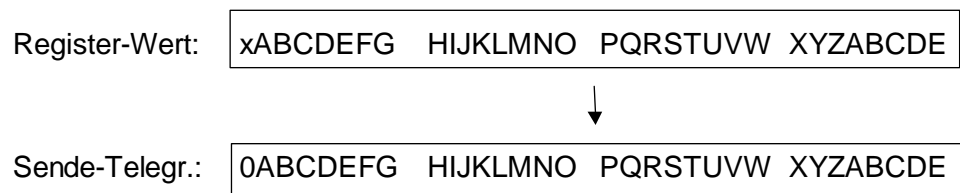
Bei den Datentypen Unsigned-8 und -16 werden die höherwertigen Bit vom Register ignoriert.

Beispiel: Unsigned-8



Beispiel: Unsigned-32

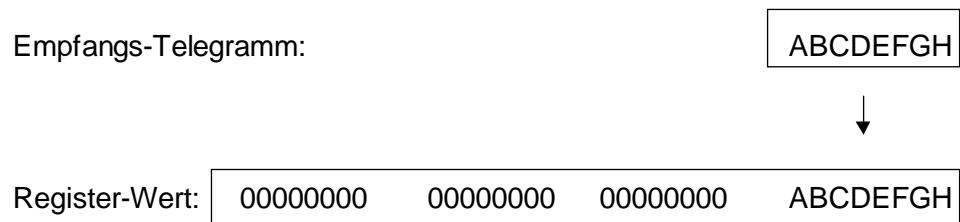
Das MSBit 32 wird im Sende-Telegramm = 0 gesetzt.  
(Bit 32 entspricht dem Vorzeichen im der PCD-Register)



Kopie Empfangstelegramm → PCD-Medien (Register)

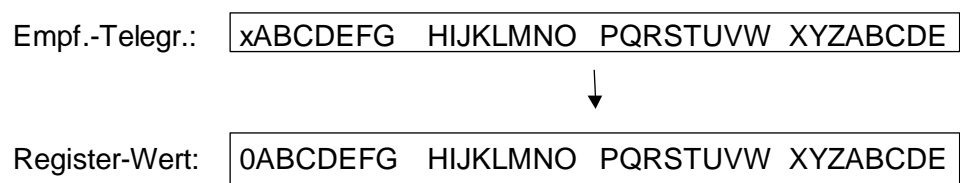
Bei den Datentypen Unsigned-8 und -16 werden die nichtbeschriebenen höherwertigen Bit im Register Null gesetzt.

Beispiel: Integer-8



Beispiel: Unsigned-32

Das Vorzeichen-Bit 32 wird im Register = 0 gesetzt.



- **Floating Point**

Wertebereich: ....

Kopie von PCD-Medien (Register) → Sende-Telegramm

Beim Kopieren wird der Wert vom Motorola-Format in das IEEE-Format umgewandelt.

Register-Wert:

Floating point im Motorola-Format

↓ Umwandlung Motorola- → IEEE-Format

Sende-Telegramm:

Floating point im IEEE-Format

Kopie vom Empfangs-Telegramm → PCD-Medien (Register)

Beim Kopieren wird der Wert vom IEEE-Format in das Motorola-Format umgewandelt.

Empfangs-Telegr.:

Floating point im IEEE-Format

↓ Umwandlung IEEE- --> Motorola-Format

Register-Wert:

Floating point im Motorola-Format

• **OCTET-String**

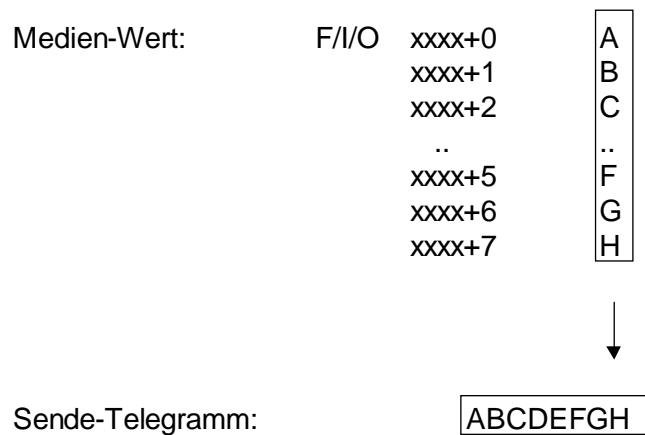
Wertebereich: .....

Kopie von PCD-Medien → Sende-Telegramm:

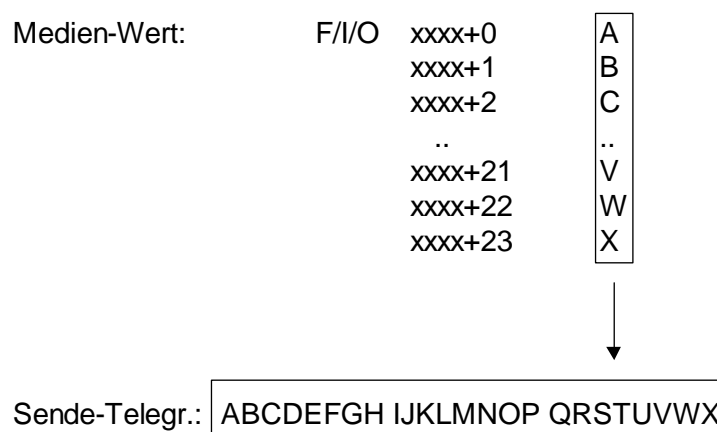
Es wird mindestens 1 String à 8 Bit (1 Byte) oder eine Anzahl n String à 8 Bit (n Byte) übertragen.

Dabei wird das Medien-Element mit der tiefsten Adresse als MSBit im Telegramm übertragen. Das Element mit der höchsten Adresse wird als LSBit übertragen. Dies entspricht dem Profil Sensorik-Aktuatorik.

Beispiel: Octet string (1)



Beispiel: Octet string (3)

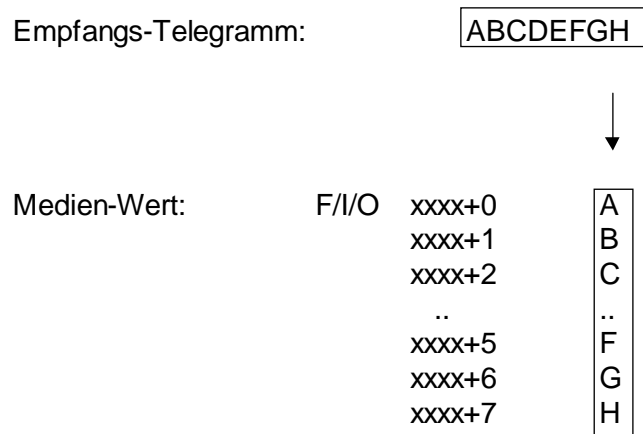


Kopie von Empfangs-Telegramm → PCD-Medien:

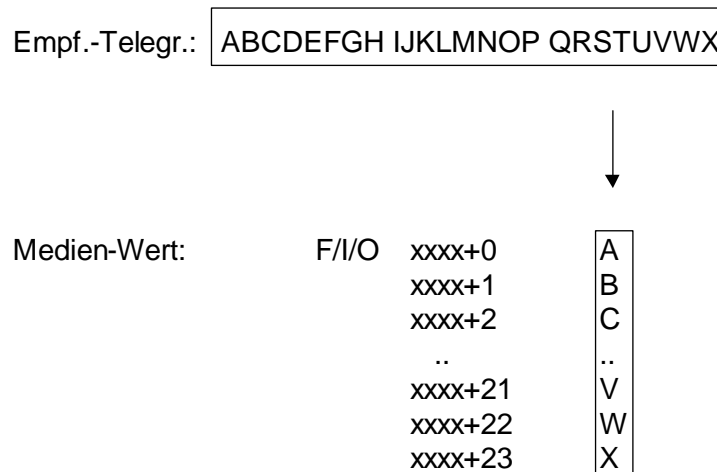
Es wird mindestens 1 String à 8 Bit (1 Byte) oder eine Anzahl n String à 8 Bit (n Byte) übertragen.

Dabei wird das MSBit vom Telegramm auf die tiefste Medienadresse kopiert. Das LSBit wird auf die höchste Medienadresse kopiert. Dies entspricht dem Profil Sensorik-Aktuatorik.

Beispiel: Octet string (1)



Beispiel: Octet string (3)





• **BIT-String**

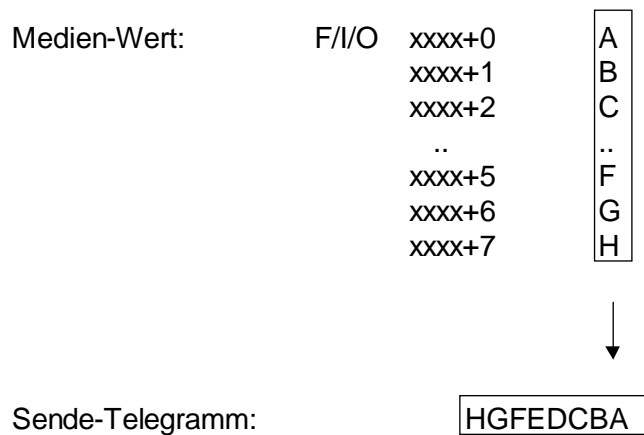
Wertebereich: .....

Kopie von PCD-Medien → Sende-Telegramm:

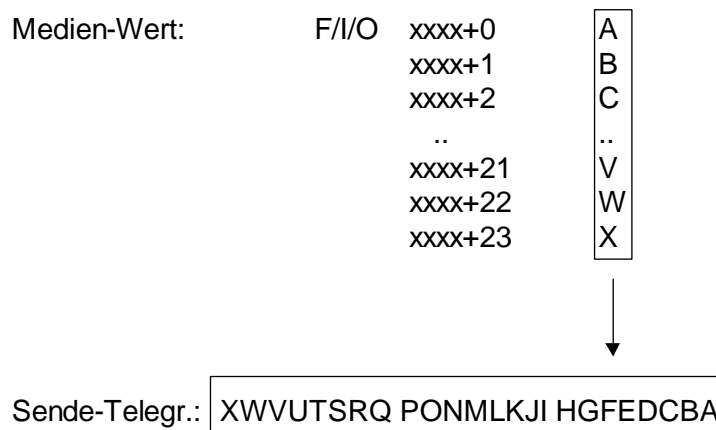
Es wird mindestens 1 String à 8 Bit (1 Byte) oder eine Anzahl n String à 8 Bit (n Byte) übertragen.

Dabei wird das Medien-Element mit der tiefsten Adresse als LSBit im Telegramm übertragen. Das Element mit der höchsten Adresse wird als MSBit übertragen. Dies entspricht der Norm 19245 Teil 2.

Beispiel: Bit string (1)



Beispiel: Bit string (3)

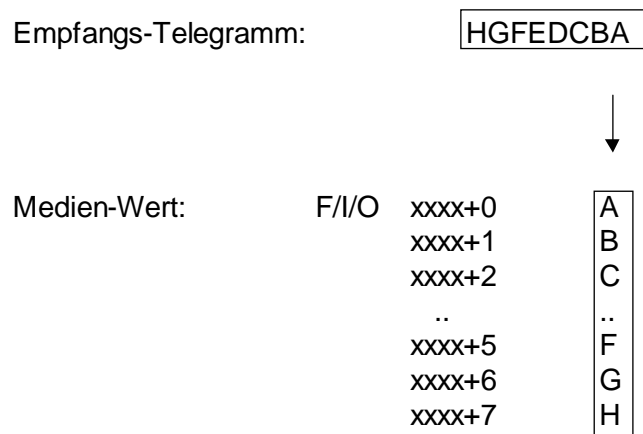


Kopie von Empfangs-Telegramm → PCD-Medien:

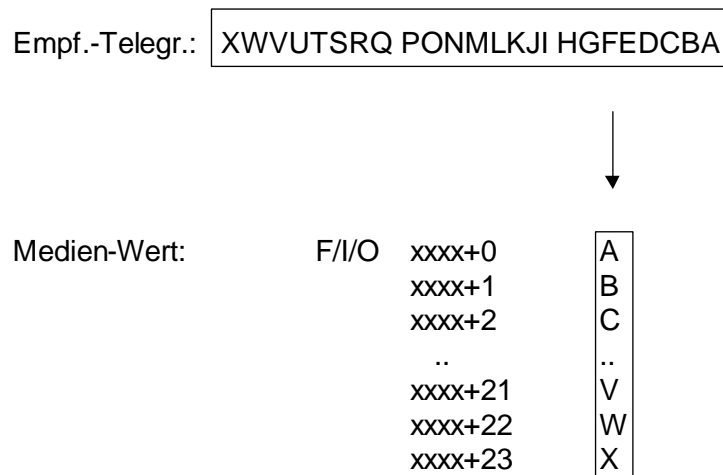
Es wird mindestens 1 String à 8 Bit (1 Byte) oder eine Anzahl n String à 8 Bit (n Byte) übertragen.

Dabei wird das MSBit vom Telegramm auf die höchste Medienadresse kopiert. Das LSBit wird auf die tiefste Medienadresse kopiert. Dies entspricht der Norm 19245 Teil 2.

Beispiel: Bit string (1)



Beispiel: Bit string (3)

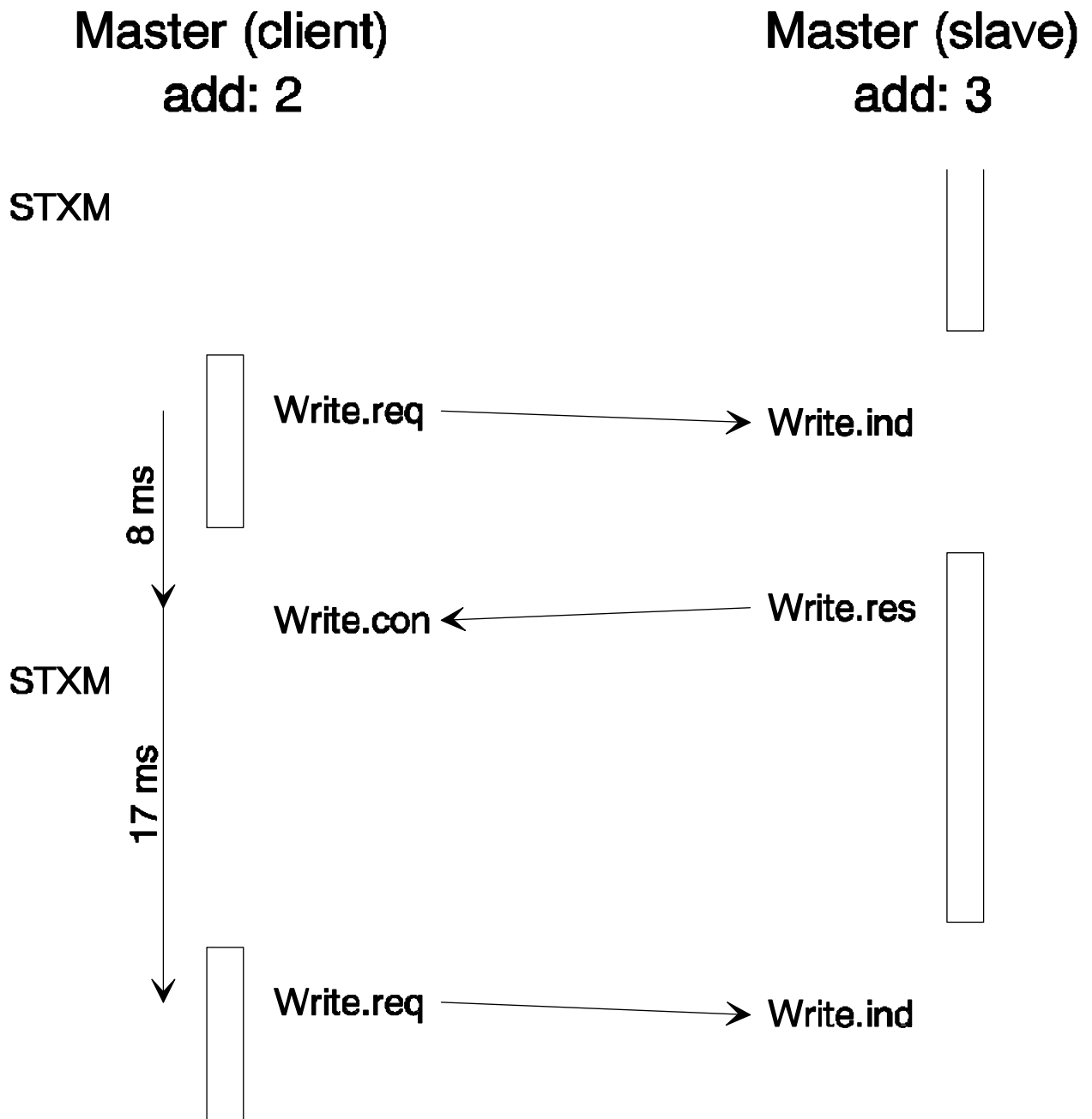


# Anhang B: Einfluss der Bus-Parameter auf die Übertragungszeiten

---

## Master -> Master

default parameters



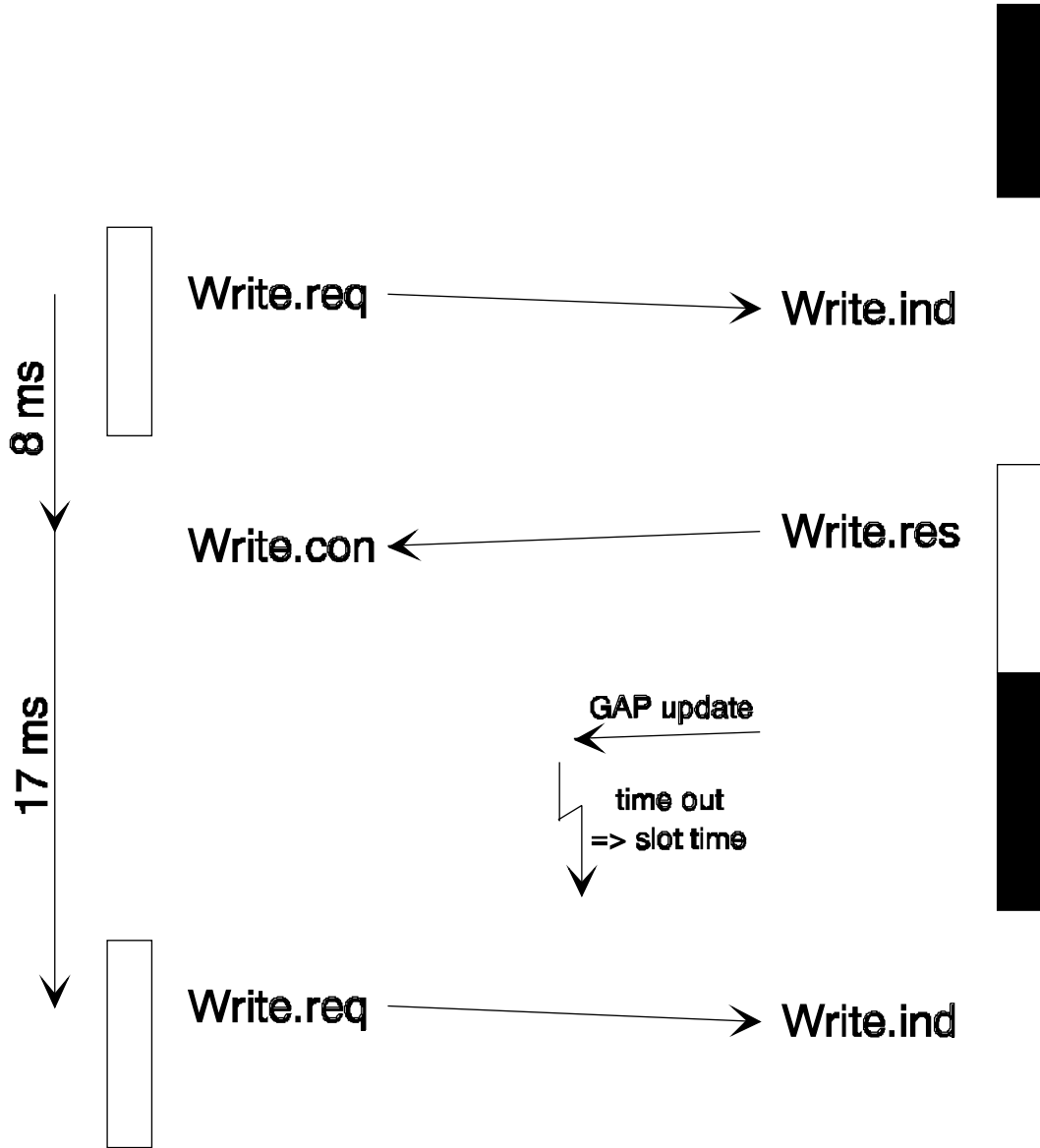
# Master -> Master

default parameters

Master (client)  
add: 2

STXM

STXM



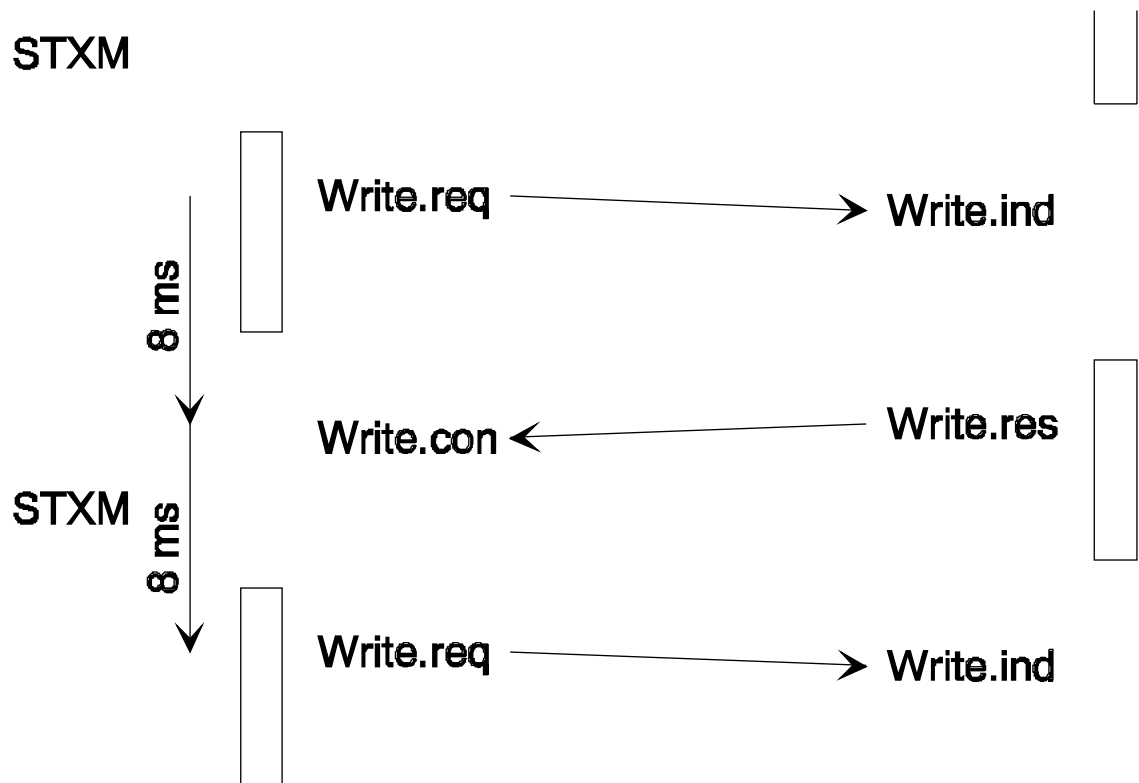
**Master -> Master**

highest station add: 3

gap update: 99

**Master (client)  
add: 2**

**Master (server)  
add: 3**

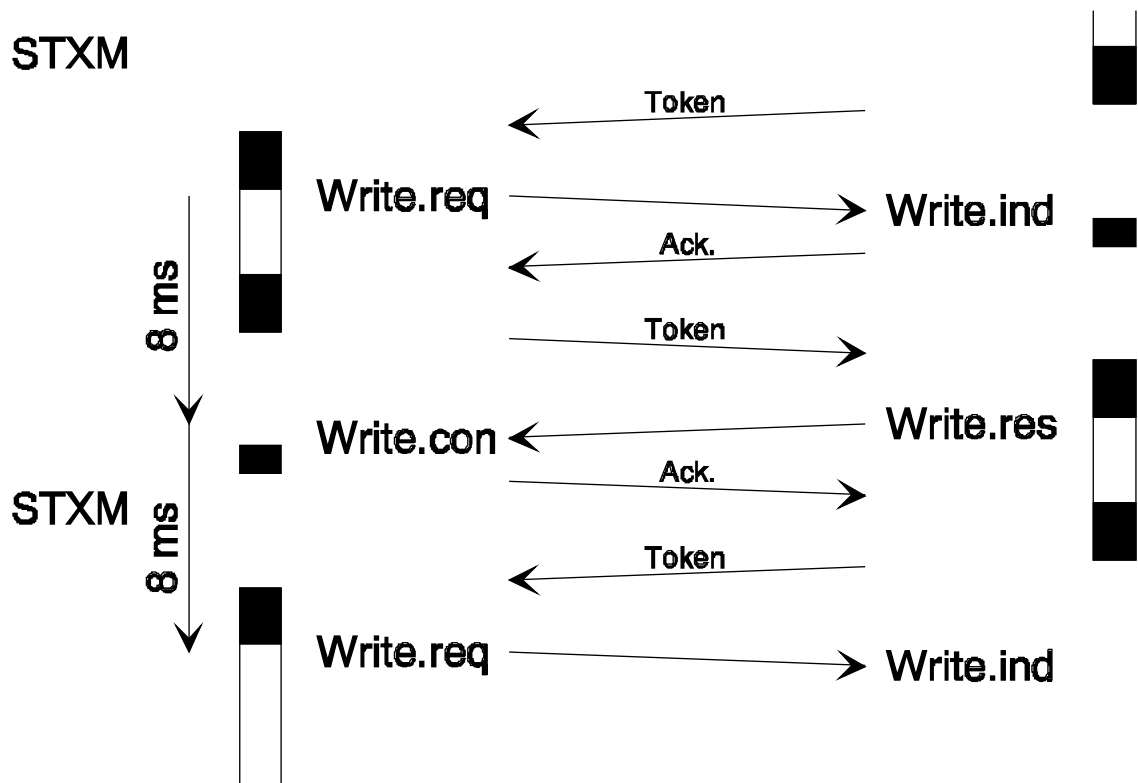


**Master -> Master**

highest station add: 3  
gap update: 99

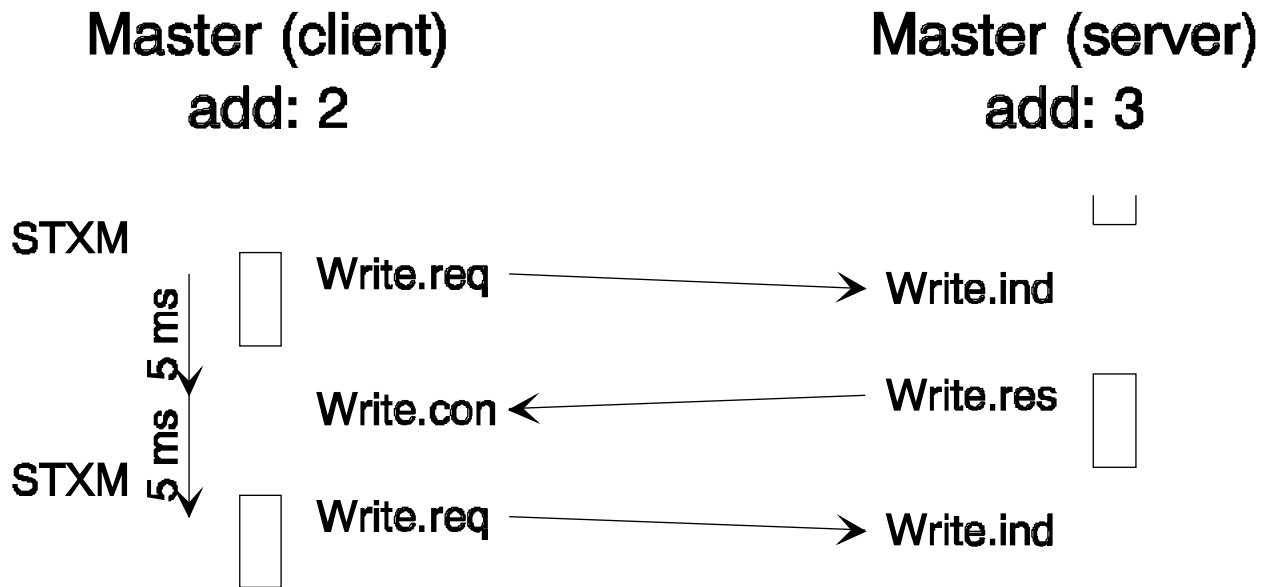
**Master (client)  
add: 2**

**Master (server)  
add: 3**



**Master -> Master**

highest station add: 3  
gap update: 99  
TSDR min: 50  
TSDR max: 100  
(slot time: 200)

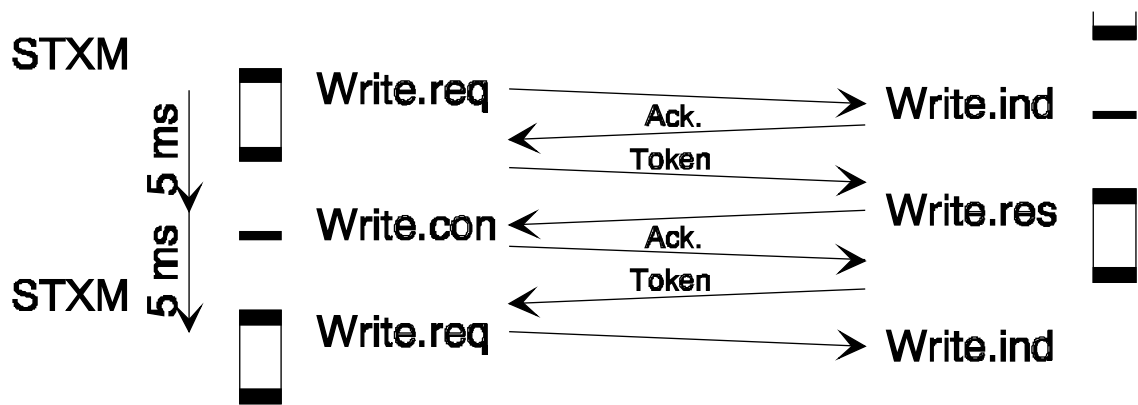


**Master -> Master**

highest station add: 3  
 gap update: 99  
 TSDR min: 50  
 TSDR max: 100  
 (slot time: 200)

**Master (client)  
 add: 2**

**Master (server)  
 add: 3**



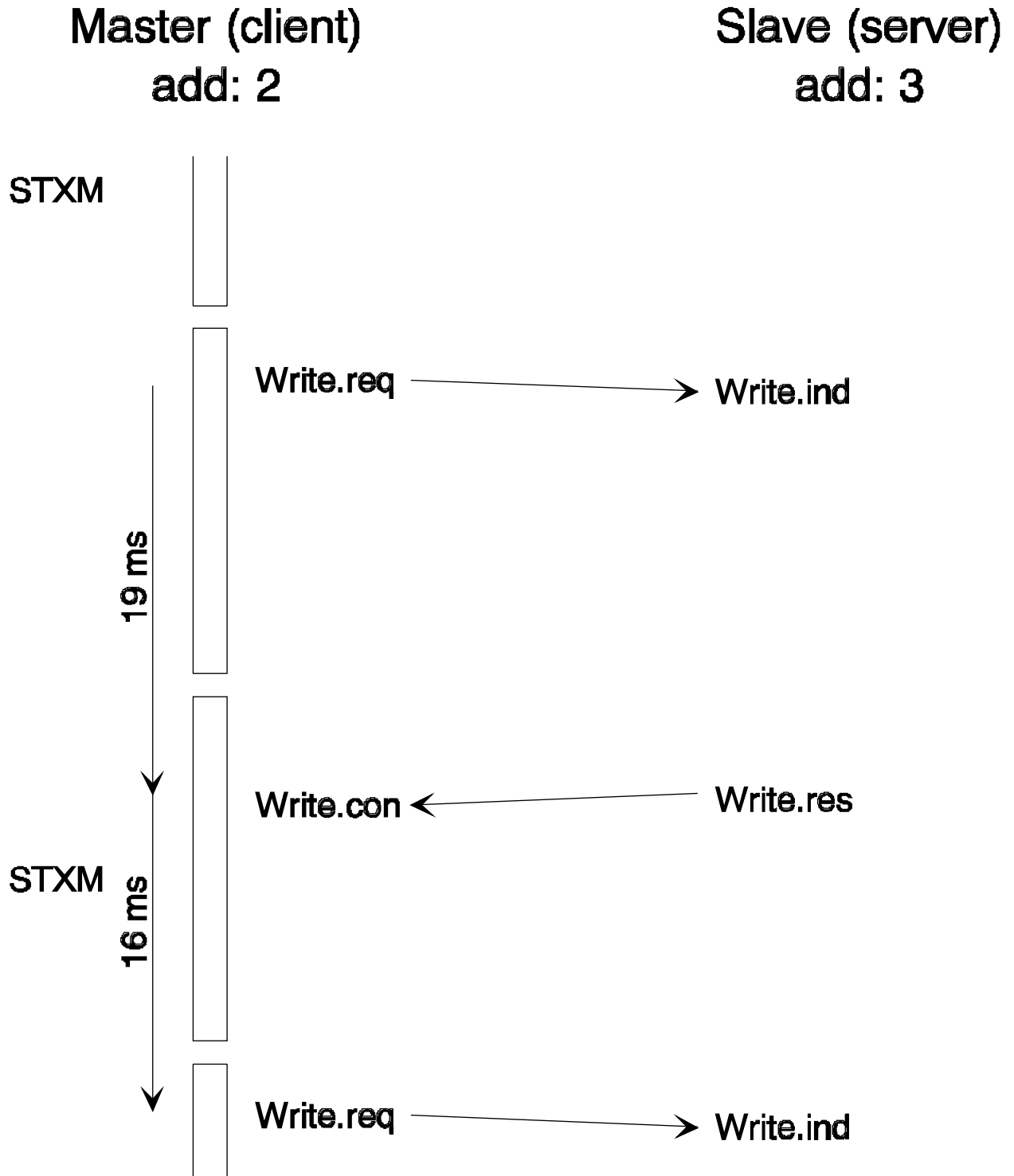
■ TSDR max

— TSDR min



**Master -> Slave**

default parameters

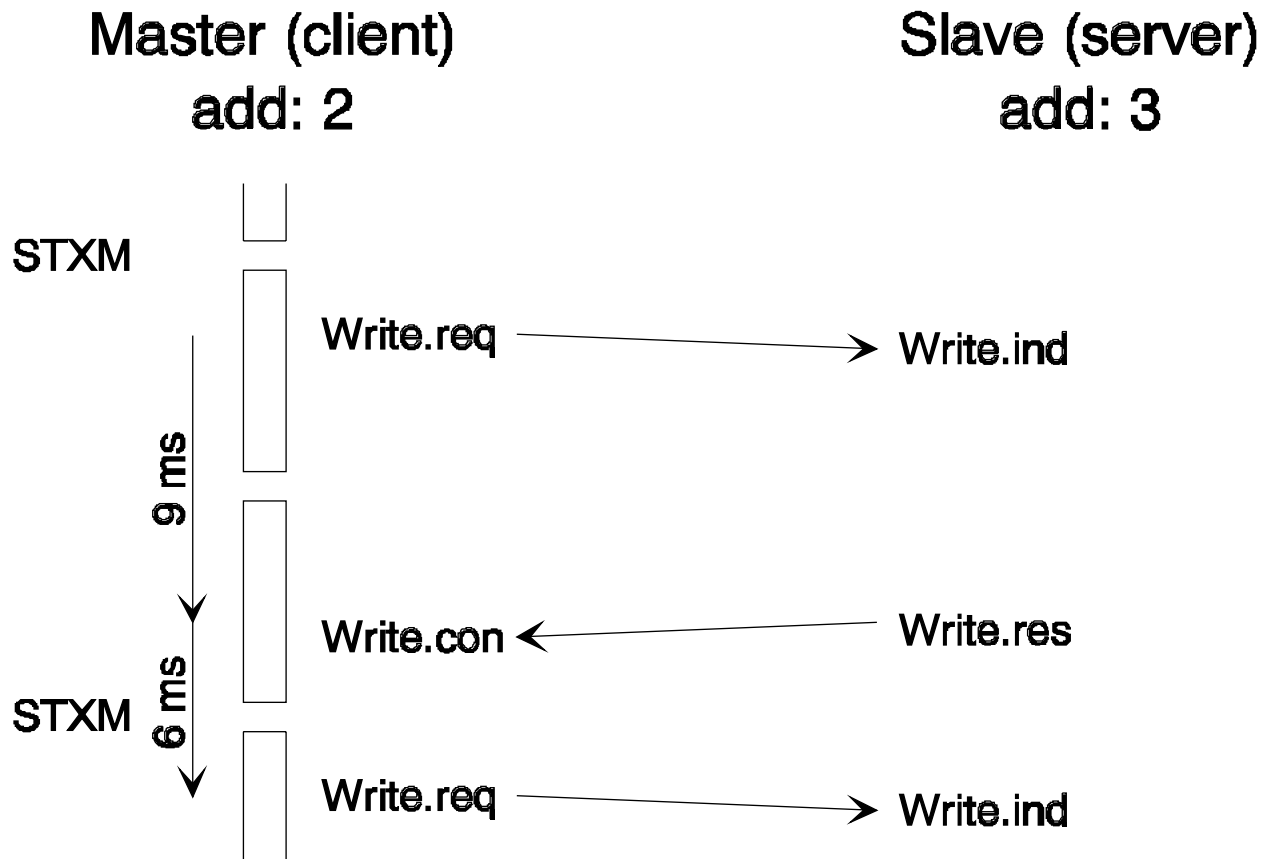




**Master -> Slave**

Highest station add: 2

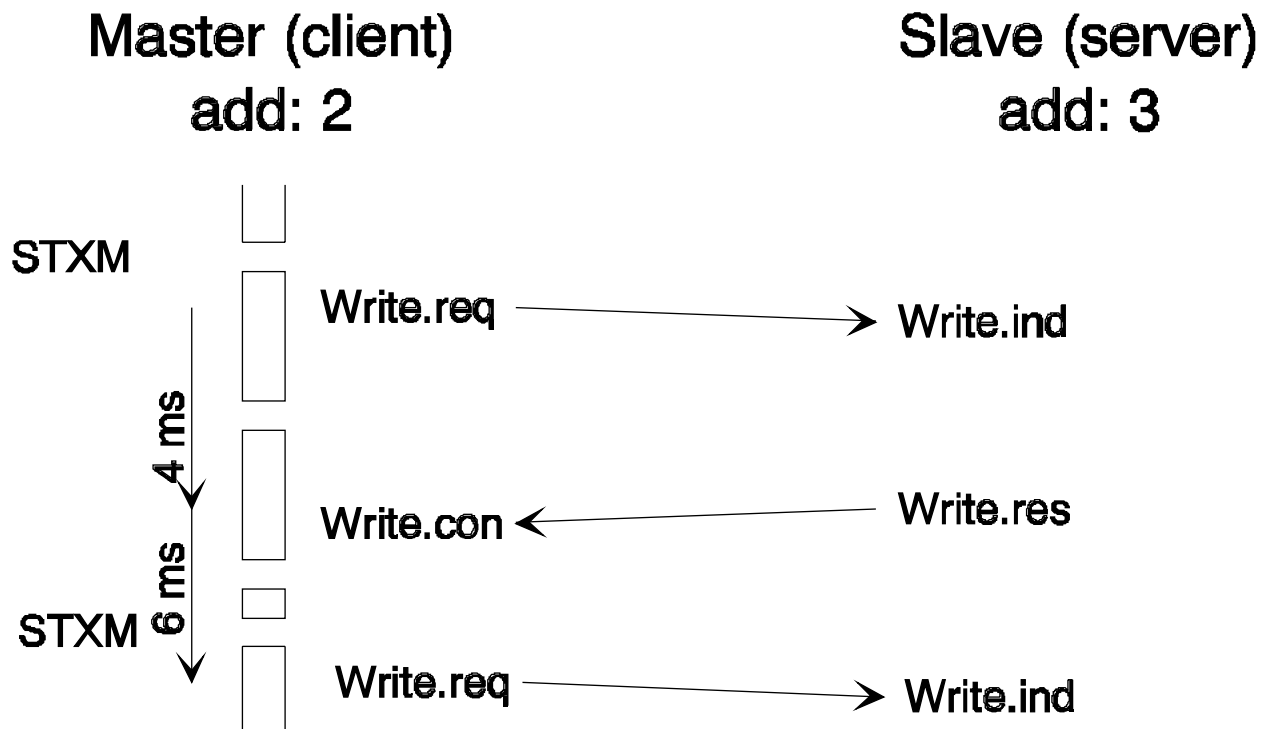
gap update: 99





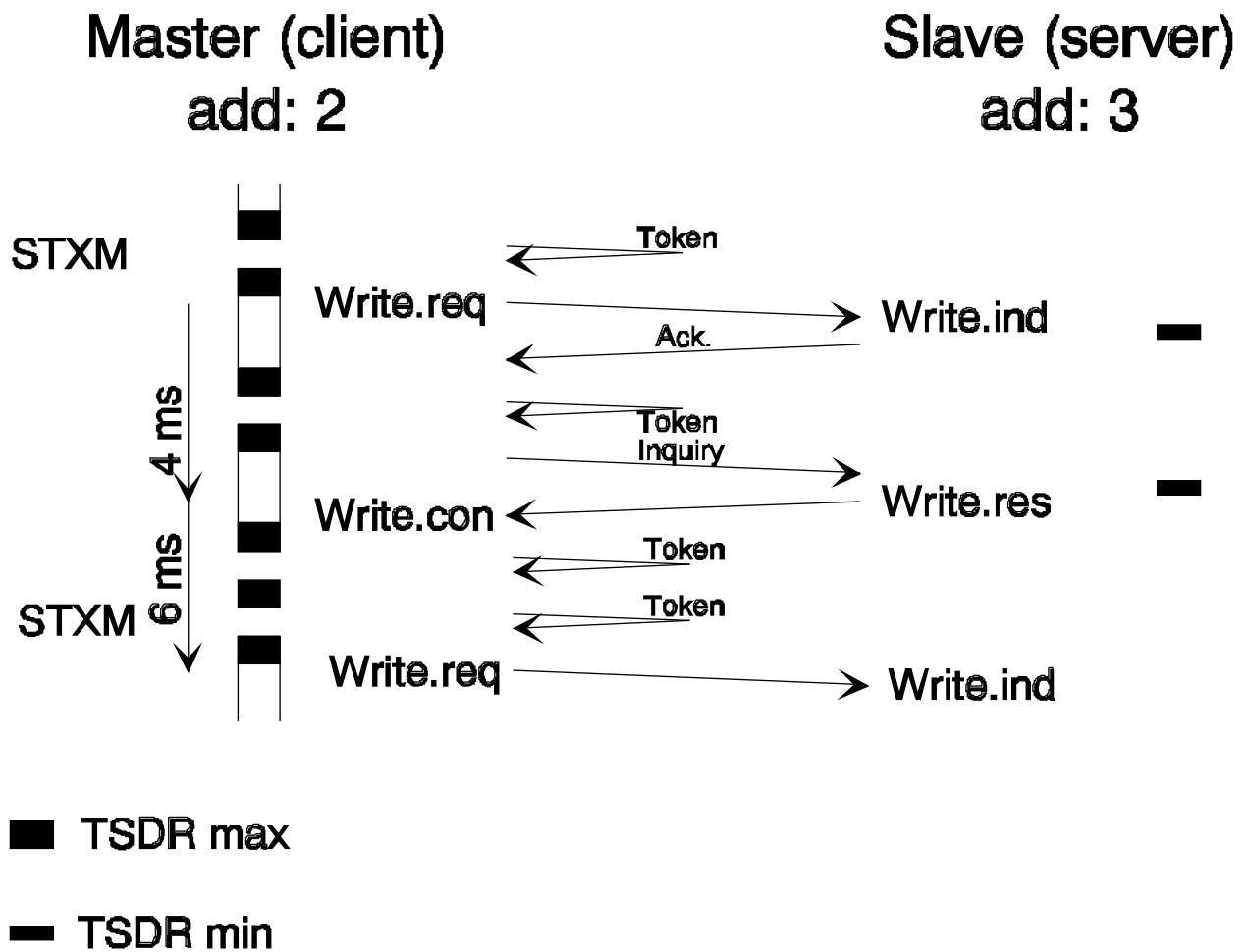
**Master -> Slave**

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)



# Anhang C: PROFIBUS-Dienste

• PROFIBUS-Dienste

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

**Cli:** Client (Dienstanforderer)      **Serv:** Server (Dienstbringer)  
**cf d:** Confirmed-Service              **unc:** Unconfirmed-Service  
**M:** Mandatory (Pflicht-Dienst)      **O:** Optional

• **Codierung der Dienste**

Die PROFIBUS-Norm sieht für die FMS-Dienste untenstehende Codierung vor.

Es wird unterschieden zwischen:

**Request und Confirmation:** Dienste, die eine Station als Client zur Verfügung stellt.

**Indication und Response:** Dienste, die eine Station als Server zur Verfügung stellt.

**Indication and Response**

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

Bits

**Request and Confirmation**

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

Bits

Beispiele:

- Codierung der Dienste für eine SAIA® PCD bei einer Master-Master-Verbindung:

00 30 00 80 30 00

- Codierung der Dienste für eine SAIA® PCD bei einer Master-Slave-Verbindung:

00 30 00 00 00 00  
Master-Station

00 00 00 80 30 00  
Slave-Station



• **Context -Management-Dienste**

Der Context umfasst sämtliche Absprachen über eine Kommunikationsbeziehung. Die Context-Management-Dienste werden zur Initialisierung und zum Freigeben einer Kommunikationsverbindung benötigt. Unzulässige Dienste werden abgewiesen.

**1. Initiate**

Mit diesem Dienst wird eine Verbindung zwischen zwei Kommunikationspartnern initialisiert. Diese Kommunikationsbeziehung muss vor der Ausführung des Initiate-Diensts bereits in der Kommunikations-Beziehungsliste (KBL) eingetragen sein. Bei der Initialisierung werden Vereinbarungen über die PDU -Länge (Nachrichten-Länge) getroffen und diejenigen Dienste genannt, die über diese Verbindung aufgerufen werden können. Ausserdem werden die aktuellen Versionsnummern des Objektverzeichnisses übertragen.

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

\*) Parameters vom Dienstanforderer  
 \*\*) Parameters vom Dienstbringer

Die ersten sechs Parameter des Initiate-Dienstes müssen an der Schnittstelle ALI/FMS vom Dienstanforderer bereitgestellt werden. Die übrigen werden durch das FMS (Fieldbus Message Specification) generiert.

**Version-OV**

Versionskennung des Objektverzeichnisses.

**Profile-Name**

Angabe des Profil-Namens vom Typ Visible-String.

**Access-Protection-Supported**

Gibt an, ob Zugriffsschutz unterstützt wird.

**Password**

Dieser Parameter enthält das Password, das auf dieser Kommunikationsbeziehung für alle Zugriffe auf Objekte des Partners gilt.

**Access-Groups**

Enthält eine Zuordnung zu bestimmten Zugriffsgruppen. Diese Zuordnung gilt auf dieser Kommunikationsbeziehung für alle Zugriffe auf Objekte des Kommunikationspartners.

**Max-PDU -Sending-High-Prio**

Maximal mögliche Länge der FMS -PDU in hochpriorer Sendrichtung.

**Max-PDU -Sending-Low-Prio**

Maximal mögliche Länge der FMS-PDU in niederpriorer Sendrichtung.

**Max-PDU -Receiving-High-Prio**

Maximal mögliche Länge der FMS-PDU in hochpriorer Empfangsrichtung.

**Max-PDU -Receiving-Low -Prio**

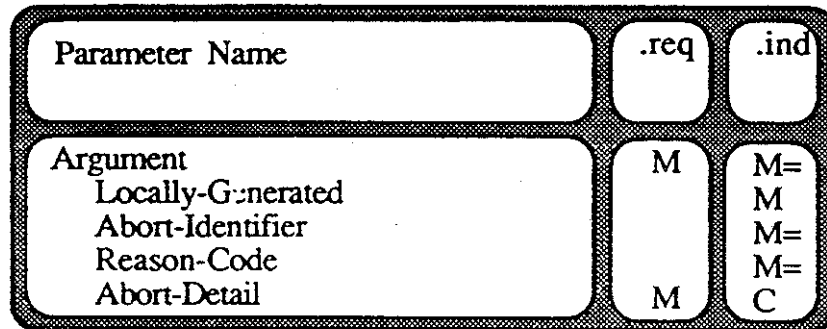
Maximal mögliche Länge der FMS-PDU in niederpriorer Empfangsrichtung.

**FMS-Services -Supported**

Dieser Parameter gibt Auskunft darüber, welche Dienste vom Absender dieser Dienstprimitive ausgeführt werden können.

## 2. Abort

Mit diesem Dienst wird eine bestehende Kommunikationsbeziehung zwischen zwei Kommunikationspartnern freigegeben. Die Verbindung kann sowohl vom Dienstanforderer (Client) als auch vom Diensterbringer (Server) gelöst werden.



### Locally-Generated

Dieser Parameter zeigt an, ob der Abbruch lokal oder vom Kommunikationspartner ausgelöst wurde.

### Abort-Identifier

Dieser Parameter zeigt an, in welcher Schicht die Ursache (0 - USER, 1 - FMS, 2 - LLI, 3 - FDL) für den Verbindungsabbruch erkannt wurde.

### Reason-Code

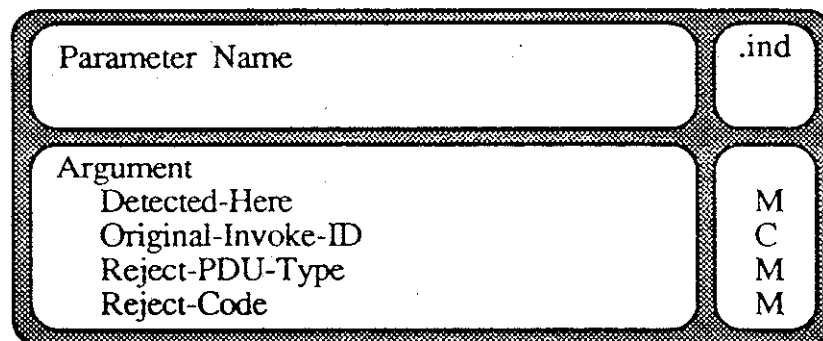
Mit diesem Parameter wird der Grund für den Abbruch angegeben.

### Abort-Detail

Dieser Parameter vom Typ Octet-String enthält zusätzliche Informationen über den Abbruchgrund. Bei Anwendungsprozess-Fehlern ist die Bedeutung durch das Profil festgelegt.

### 3. Reject

Mit dem Reject-Dienst weist die FMS-Schicht eine unzulässige PDU (Nachricht) ab. Der Reject-Dienst wird immer dann generiert, wenn der vom Dienstanforderer erteilte Dienstauftrag vom virtuellen Feldgerät oder dem Kommunikationsprozess überhaupt nicht behandelt werden kann. Zum Beispiel sind unter unzulässigen Nachrichten solche zu verstehen, die eine unerlaubte oder nicht ausführbare Dienstanforderung enthalten. Ein weiterer Grund für die Generierung eines Rejects kann eine zu lange Nachricht sein, die nicht mehr in die vorgesehenen Puffer passt. Eine fehlerhafte Ausführung des Lesediensts aufgrund von zum Beispiel falscher Angabe des Index, führt lediglich zu einer negativen Auftragsquittung (negative Response).



#### Detected -Here

Dieser Parameter gibt an, ob der Fehler noch im eigenen Gerät oder beim Kommunikationspartner erkannt wurde.

#### Original-Invoke-ID

Die Auftragsnummer der zurückgewiesenen PDU.

#### Reject-PDU-Type

Dieser Parameter gibt an, von welchem Typ die zurückgewiesene PDU ist.

Es werden folgende Typen unterschieden:

- |   |                           |
|---|---------------------------|
| 1 | Confirmed-Request-PDU     |
| 2 | Confirmed-Response-PDU    |
| 3 | Unconfirmed-PDU           |
| 4 | nicht erkannter PDU - Typ |

#### Reject-Code

Grund für die Zurückweisung der PDU.

#### 4. Fehler-Codes

##### FMS-Reject-Codes

0	anderer Fehler
1	Invoke-ID existiert schon
2	Max. Anzahl paralleler unbestätigter Dienste überschritten
3	Service nicht unterstützt (verbindungsorientierte KR)
4	Service nicht unterstützt (verbindunglose KR)
5	PDU-Länge überschritten
6	unzulässige Service-Primitive
41h	Max. Anzahl paralleler unbestätigter Dienste überschritten

##### FMS-Reject-PDU-Types

1	Confirmed Request PDU
2	Confirmed Response PDU
3	Unconfirmed PDU
4	nicht erkannter PDU-Typ

##### Initiate-Error-Codes (User)

0	amderer Fehler
3	Version-OV inkompatibel
4	FMS-User lehnt Initiate ab
5	Passwort-Error
6	Profile-Number

##### Initiate-Error-Codes (FMS und FMA7)

1	maximale PDU-Länge überschritten
2	Features not Supported

**FMS-Abort-Reason-Codes**

- 0 FMS-KBL-Fehler
- 1 Benutzer-Fehler
- 2 FMS-PDU-Fehler
- 3 Unzulässige LLI-Service-Primitive
- 4 LLI-Fehler
- 5 PDU-Länge zu gross
- 6 Service nicht unterstützt
- 7 Invoke-ID existiert nicht
- 8 Max. Anzahl paralleler Dienste überschritten
- 9 Zustandskonflikt im FMS
- 10 Service-Fehler (Response stimmt nicht mit Request überein)
- 11 Invoke-ID existiert schon
- 12 FMS-Disable: FMA7 lädt die KBL

**Abort-Detail bei Benutzer-Fehler**

- 1 KR ist nicht definiert
- 2 Illegaler Service-Code
- 3 Optionaler String mit Länge 0
- 4 Namensadressierung mit unbekanntem Zugriffs-Code
- 6 Liste ist leer
- 7 Fehler in Integer-Variable
- 8 Illegale Error-Class
- 9 Illegaler PI-State
- 10 Illegale Typenbeschreibung
- 11 Verbindung ist nicht aufgebaut
- 12 Initiate-Response: Verbindung ist nicht im Aufbaustadium
- 13 Initiate-Request: Verbindung ist nicht im Zustand CLOSED
- 14 Illegaler LSAP
- 15 Characteristic Length < > 40 (nur bei Management)

**FMA7-Abort-Reason-Codes**

- 0 FMA7-KBL-Fehler
- 1 Benutzer-Fehler
- 2 FMA7-PDU-Fehler
- 3 Unzulässige LLI-Service-Primitive
- 4 LLI-Fehler
- 5 PDU-Länge zu gross
- 6 Service nicht unterstützt
- 7 Response-Fehler (Request existiert nicht)
- 8 Max. Anzahl paralleler Dienste überschritten
- 9 Zustandskonflikt im FMA7
- 10 Service-Fehler (Response stimmt nicht mit Request überein)

**Instanz-Identifizier**

0	User
1	LLI-User (FMS bzw. FMA7)
2	LLI
3	FDL/PHY

**User-Abort-Reason-Codes**

0	Disconnect
1	Version-OV inkompatibel
2	Passwort-Error
3	Profile-Number inkompatibel
4	Limited Service permitted
5	Put-OV Loading-Interacting

**LLI-Abort-Reason-Codes**

0	LLI-LLI-Kontextprüfung negativ
1	unzulässige LLI-PCD in der Aufbauphase empfangen
2	unzulässige LLI-PDU in der Datentransferphase empfangen
3	unbekannte oder fehlerhafte LLI-PDU empfangen
4	DAT_ACK_PDU empfangen und SAC = 0
5	Anzahl paralleler Dienste überschritten (Indication vom FDL)
6	Invoke-ID unbekannt
7	Prioritätsfehler
8	lokaler Fehler beim Remote Teilnehmer
9	Timer 1 abgelaufen (Verbindungsaufbauüberwachung)
10	Timer 3 abgelaufen (Verbindungsüberwachung)
11	ETimer abgelaufen
12	Fehler bei LSAP-Aktivierung (AD enthält Status)
13	unzulässige FDL-Primitive in der Aufbauphase
14	unzulässige FDL-Primitive in der Datentransferphase
15	unbekannte FDL-Primitive
16	unbekannte LLI-Primitive
17	unzulässige LLI-Primitive in der Aufbauphase
18	unzulässige LLI-Primitive in der Datentransferphase
19	KBL-Eintrag nicht ok.
20	Konfliktfall in der Verbindungsaufbauphase
21	Ablauffehler bei zyklischem Datenverkehr
22	Anzahl paralleler Dienste überschritten (Request vom FMS)
23	KBL wird vom FMA7 geladen, LLI disabled
24	Confirm / Indication-Mode Fehler
25	unzulässige FMA 1/2-Primitive empfangen
26	unzulässige FMS-Primitive auf zyklischer Verbindung
27	FMS-PDU-Länge auf zyklischer Verbindung überschritten

Notizen:



## Anhang D: Abkürzungen und Begriffe

---

**Abort**

Dienst zum Abbau (Schliessen) eines logischen Kommunikationskanals.

→ Dienst, Initiate

**Array**

Datentyp. Feld von strukturierten Variablen. Aneinanderreihung von Einfachvariablen des gleichen Datentyps

→ Record

**Boolean**

Datentyp. Kann die Werte FALSE (0) oder TRUE (255) annehmen.

**Broadcast**

Eine Nachricht, die gleichzeitig an alle Teilnehmer im Netz gesendet wird. Wird von SAIA-Geräten nicht unterstützt.

→ Multicast

**Client**

Dienstanforderer (in der Regel der Master). Im Sinne der Kommunikation ein Anwendungsprozess, der Aufträge erteilt, um die Funktion (Objekte) eines fernen Anwendungsprozesses mit Hilfe von Diensten zu nutzen.

→ Server

**Confirmation**

Bestätigung des Erhalts einer Dienstanforderung beim lokalen Diensterbringer auf einen zuvor gestarteten Request.

→ Dienstprimitive, Request, Indication

**Context-Management**

Verwaltungsdienste (Initiate, Abort, Reject)

→ Dienst

**Default-Wert**

Standardwert, der dann verwendet wird, wenn kein anderer Wert explizit eingegeben wird.

**Dienst, Service**

Eine auf Objekte gerichtete Operation; definiert durch die DIN 19245 -Teil 1&2.

Zum Beispiel ist ein Lese- oder Schreibauftrag an einen Master oder Slave ein Dienst.

Es wird unterschieden zwischen:

- Bestätigte- und unbestätigte Dienste (confirmed/unconfirmed)

Bei bestätigten Diensten wird die Auftragsausführung mit einer Quittung des Dienstbringers (Server) versehen.

Bei unbestätigten Diensten erfolgt keine Quittierung der Auftragsausführung

Unbestätigte Dienste (Multi- oder Broadcast-Telegramme) werden von SAIA-Geräten nicht unterstützt.

- Parallele Dienste.

Von parallelen Diensten auf einem Kommunikationskanal spricht man, wenn bei einem Server zu einem Zeitpunkt mehrere noch nicht ausgeführte Aufträge von einem Client vorliegen, oder wenn ein Client zu einem Zeitpunkt mehrere Aufträge bei einem Server ausgelöst hat, die noch nicht ausgeführt wurden. Sogenannte Auftragskennungen (Invoke-Identification) ermöglichen eine Unterscheidung der einzelnen Aufträge.

Eine SAIA<sup>o</sup>PCD kann nur als Server parallele Dienste verarbeiten.

### **Dienstprimitive**

Ein abstraktes Element, welches die Verbindung zwischen Benutzer und dem Dienst einer darunterliegenden Schicht ermöglicht. Eine Anforderung zur Ausführung eines Dienstes wird als Request (.req) erteilt und als Indication (.ind) bearbeitet. Bestätigte Dienste werden mit einer Response (.res) beantwortet, die als Confirmation (.con) beim Dienstanforderer empfangen wird.

→ Request, Indication, response, Confirmation

### **Dienstzugangspunkt, SAP, Service Access Point (2.62, 128)**

Ein SAP ist die logische Schnittstelle zwischen der Anwenderschicht (FMS-Ebene) und der darunterliegenden FDL-Schicht (Field Data Link).

Man unterscheidet zwischen

LSAP (Local Service Access Point) = SAP der eigenen Station

und

RSAP (Remote Service Access Point) = SAP der Partnerstation

Daten werden über diese logischen Schnittstellen ausgetauscht.

Ein SAP enthält auch Informationen über die Verbindungsart:

- Master/Slave
- Zyklisch/Azyklisch
- usw.

Es gelten folgende Regeln:

- Poll SAP = gemeinsamer SAP für alle Slavestationen (Default Poll SAP =128)
- Bei einer definierten Master-Master-Verbindung kann der gleiche SAP nur einmal für eine einzige Verbindung (Kanal) verwendet werden.
- Für eine offene Verbindung gilt sinngemäss die gleiche Regel wie für den Kommunikationskanal

### **DP**

Dezentrale Peripherie

### **Event-Management-Dienste**

Dienste, die ereignisabhängige Meldungen (Objektyp: Event) bearbeiten. Nicht unterstützt von SAIA-Geräten.

### **FDL, Fieldbus Data Link**

Bezeichnung für die Datensicherungsschicht (2) bei PROFIBUS

### **FMS, Fieldbus Message Specification**

Definiert die Applikationsdienste bei PROFIBUS

### **GAP**

Der Adressbereich zwischen der eigenen Teilnehmeradresse eines aktiven Teilnehmers bis zum nächsten aktiven Teilnehmer im logischen Ring.

### **Identify**

Dienst zur Anforderung der Geräteerkennung eines anderen Teilnehmers. Wird von SAIA-Geräten nur als Server unterstützt.

### **Indication**

Anzeige einer eingetroffenen Dienstanforderung beim partnerseitigen Diensterbringer (Server).

→ Dienstprimitive, Confirmation

### **KBL, Kommunikationsbeziehungsliste**

Liste, die Daten der projektierten Verbindungen enthält.

→ Kommunikationsreferenz, Verbindung

**Kommunikationskanal**

Virtueller Übertragungsweg für eine virtuelle Verbindung zwischen zwei Stationen.

Ein Kanal enthält folgende Informationen:

- Adresse der Partnerstation
- Die verwendeten **Service Access Points** (LSAP und RSAP)
- Verbindungstyp (Zyklisch, azyklisch, usw.)
- Liste der erlaubten Dienste (z.B. Read/Write, usw.)
- Die Grösse der benötigten Datenpuffer (PDU-Länge)

Für den Kommunikationskanal gelten folgende Regeln:

- Derselbe Kanal kann nur für eine definierte Verbindung verwendet werden.
- Bei einer offenen Verbindung, kann ein Kanal für Verbindungen mit mehreren Partnergeräten verwendet werden (wird oft bei nichtkonfigurierbaren Slaves angewandt).
- Für dieselbe Station können bis zu 90 (10 bei PCD7.F700) Kanäle definiert und parallel betrieben werden.

**KR, Kommunikationsreferenz, Kommunikationskanal-Nummer**

Kennzahl zur logischen Adressierung (Identifizierung) von Verbindungen.

**Kommunikationsobjekt**

Datenblock für die Kommunikation. PROFIBUS sieht eine objektorientierte Arbeitsweise vor, damit auch die unterschiedlichsten Geräte untereinander Daten austauschen können.

Bevor Daten ausgetauscht werden können, müssen dem System die Kommunikationsobjekte und deren Bedeutung bzw. Struktur bekannt gegeben werden. Die Definition der Objekte geschieht für alle Busteilnehmer im Konfigurator.

Eine Station führt jeweils eine vollständige oder teilweise Kopie der Objekte ihrer Partnerstation.

Es gelten folgende Regeln:

- Die Objekte einer Station sind grundsätzlich (Zugriffsrechte können definiert werden) allen Busteilnehmern via den Kommunikationskanälen zugänglich.
- Bei der Übertragung von Objekten müssen das Source und Quellobjekt kompatible Datenstrukturen und -Formate aufweisen.

→ Objektverzeichnis

**Lower Layer Interface, LLI**

Ist ein Teil der Anwendungsschicht 7 bei PROFIBUS-FMS.

**Master**

Aktiver Teilnehmer, der am Token-Passing beteiligt ist.

→ Slave, Token-Passing

**MMAC**

Master-Master Verbindung für azyklischen Datenverkehr.

**MSAC**

Master-Slave Verbindung für azyklischen Datenverkehr.

**MSAC\_SI**

Master-Slave Verbindung für azyklischen Datenverkehr mit Slave-Initiative.

Nicht unterstützt von SAIA-Geräten.

**MSCY**

Master-Slave Verbindung für zyklischen Datenverkehr.

**MSCY\_SI**

Master-Slave Verbindung für zyklischen Datenverkehr mit Slave-Initiative.

Nicht unterstützt von SAIA-Geräten.

**Multicast**

Eine Nachricht, die gleichzeitig an mehrere Teilnehmer im Netz gesendet wird.

Nicht unterstützt von SAIA-Geräten.

→ Broadcast

**Objektverzeichnis, OV**

Verzeichnis aller erreichbaren (definierten) Kommunikationsobjekte. Es enthält z.B. Angaben zur Struktur, Datentyp, Zugriffsrechten und Speicheradresse der Objekte.

**PA**

Prozess Automation

**Password**

Dezimalzahl, Zugriffsschutz für ein Objekt oder eine Verbindung.

**PNO**

PROFIBUS-Nutzerorganisation

**Profile**

Auswahl von Funktionen und Leistungsmerkmalen innerhalb der Grenzen der Gesamtspezifikation der PROFIBUS-Norm.

**RAC**

Receive Acknowledged request Counter.  
Maximal zulässige Anzahl der unbestätigten, parallelen Dienste als Empfänger.

**RCC**

Receive Confirmed request Counter.  
Gibt an, wieviele bestätigte Dienste (Anfragen) der Teilnehmer als Server auf diesem Kanal maximal annehmen kann, ohne eine Antwort zu senden.

**Read**

Dienst zum Lesen eines Objekts eines anderen Teilnehmers.  
→ Dienst, Write

**Record**

Datentyp. Aneinanderreihung von Einfachvariablen unterschiedlichen Datentyps.  
Nicht unterstützt von SAIA-Geräten.  
→ Array

**Reject**

Dienst, mit dem eine ungültige Dienstanforderung zurückgewiesen wird.  
→ Dienst

**Repeater**

Bidirektionaler Leitungsverstärker

**Request**

Dienstanforderung des lokalen Benutzers.  
→ Dienstprimitive, Response

**Response**

Antwort des partnerseitigen Diensterbingers (Server) auf eine Indication.  
→ Dienstprimitive, Indication, Request

**SAC**

Send Acknowledged request Counter.

Maximal zulässige Anzahl der unbestätigten, parallelen Dienste als Sender.

**SCC**

Send Confirmed request Counter.

Gibt an, wieviele ausstehende Antworten der Teilnehmer als Client auf diesem Kanal maximal verarbeiten kann.

**Server**

Diensterbringer (Master oder Slave). Ein auftragsausführender Anwendungsprozess, der die Funktionalität (Objekte) seines virtuellen Feldgerätes einem Client zur Verfügung stellt.

→ Client

**Service**

→ Dienst

**Service Access Point, SAP**

→ Dienstzugangspunkt

**Slave**

Passiver Teilnehmer, der nicht am Token-Passing beteiligt ist und somit nur nach Aufforderung durch einen Master senden darf.

→ Master

**Target-Rotation-Time**

Token-Umlaufzeit. Als Sollwert, ein vom Anwender projektiertes Bus-Parameter (TTR). Als Istwert, die tatsächlich gemessene Zeit, während der alle aktiven Teilnehmer den Token einmal besitzen haben.

→ Token, Token-Passing

**Token**

Ein Telegramm mit vordefinierter Bitfolge, durch das die Kontrolle über das Bussystem von einer Station zur nächsten weitergegeben wird.

→ Token-Passing

**Token-Passing**

Buszugriffssteuerung, bei der durch das Token-Telegramm die Buszugriffsberechtigung, von Master zu Master weitergegeben wird. Nur derjenige Teilnehmer (Master) darf am Bus agieren, der den Token besitzt.

→ Token, Master, Slave

**Verbindung, verbindungsorientierte Kommunikationsbeziehung**

Spezifikation eines logischen Kommunikationskanals zwischen zwei Teilnehmern (Punkt-zu-Punkt-Verbindung).

**Verbindungsaufbau**

→ Initiate

**Verbindungsüberwachung**

Laufende Kontrolle der Funktionsfähigkeit einer logischen Verbindung

**VFD, Virtual Field Device**

Abstraktes Modell zur Beschreibung der Daten und des Verhaltens eines Automatisierungssystems aus Sicht eines Kommunikationspartners.

**Write**

Dienst zum Beschreiben eines Objekts eines anderen Teilnehmers.

→ Dienst, Read



Absender:

Firma  
Abteilung  
Name  
Adresse

Tel.

Datum

An:

SAIA-Burgess Electronics AG  
Bahnhofstrasse 18  
CH-3280 Murten (Schweiz)  
<http://www.saia-burgess.com>

GB: Electronic Controllers

PROFIBUS-FMS mit SAIA® PCD

Falls Sie Vorschläge zu SAIA® PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Ihre Vorschläge: