



Gerhard Schnell
Bernhard Wiedemann *Hrsg.*

Bussysteme in der Automatisierungs- und Prozesstechnik

Grundlagen, Systeme und Anwendungen
der industriellen Kommunikation

9. Auflage



Springer Vieweg

Bussysteme in der Automatisierungs- und Prozesstechnik

Lizenz zum Wissen.

Sichern Sie sich umfassendes Technikwissen mit Sofortzugriff auf tausende Fachbücher und Fachzeitschriften aus den Bereichen:
Automobiltechnik, Maschinenbau, Energie + Umwelt, E-Technik,
Informatik + IT und Bauwesen.

Exklusiv für Leser von Springer-Fachbüchern: Testen Sie Springer für Professionals 30 Tage unverbindlich. Nutzen Sie dazu im Bestellverlauf Ihren persönlichen Aktionscode **C0005406** auf www.springerprofessional.de/buchaktion/



Jetzt
30 Tage
testen!

Springer für Professionals.

Digitale Fachbibliothek. Themen-Scout. Knowledge-Manager.

- ⌚ Zugriff auf tausende von Fachbüchern und Fachzeitschriften
- ⌚ Selektion, Komprimierung und Verknüpfung relevanter Themen durch Fachredaktionen
- ⌚ Tools zur persönlichen Wissensorganisation und Vernetzung

www.entschieden-intelligenter.de

Springer für Professionals

 Springer

Gerhard Schnell · Bernhard Wiedemann
(Hrsg.)

Bussysteme in der Automatisierungs- und Prozesstechnik

Grundlagen, Systeme und Anwendungen
der industriellen Kommunikation

9., aktualisierte und verbesserte Auflage



Springer Vieweg

Hrsg.

Gerhard Schnell
Stuttgart, Deutschland

Bernhard Wiedemann
Mannheim, Deutschland

ISBN 978-3-658-23687-8
<https://doi.org/10.1007/978-3-658-23688-5>

ISBN 978-3-658-23688-5 (eBook)

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 1994, 1996, 1999, 2000, 2003, 2006, 2008, 2012, 2019

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Vorwort zur 9. Auflage

Gegenüber der 8. Auflage wurde das Kapitel EIB (Europäischer Installationsbus) aktualisiert: Er firmiert jetzt unter „KNX-System für die Haus- und Gebäudesystemtechnik“. Außerdem wurde ein Text über Sercos, den Bus für synchrone Antriebstechnik, neu aufgenommen. Dadurch sind auch neue Autoren hinzugekommen. Bussysteme die unserer Meinung nach keine Marktbedeutung mehr haben, haben wir nicht mehr beschrieben. Da Bussysteme aber langlebige Investitionen sind, haben wir im Teil „Datenblätter“ aus historischen Gründen wenig geändert.

Die Herausgeber danken den neu hinzugekommenen Koautoren für ihre Mithilfe, dieses Standardwerk zu aktualisieren, dem Verlag für seine Mühe bei der Neubearbeitung und hoffen weiterhin auf das Wohlwollen unserer Leser in Hochschulen und Industrie.

Stuttgart/Mannheim
Herbst 2018

Gerhard Schnell
Bernhard Wiedemann

Inhaltsverzeichnis

1	Technische Grundlagen	1
1.1	Netzwerktopologien	1
1.1.1	Zweipunktverbindungen	1
1.1.2	Zweipunktverbindungen mit Multiplexer	2
1.1.3	Bus-Struktur	3
1.1.4	Baumstruktur	6
1.1.5	Ringstruktur	7
1.1.6	Sternstruktur	8
1.2	Kommunikationsmodelle	9
1.2.1	Das ISO/OSI-Referenzmodell	9
1.2.1.1	Allgemeines	9
1.2.1.2	Die physikalische Schicht oder Bitübertragungsschicht	11
1.2.1.3	Die Sicherungsschicht	11
1.2.1.4	Die Netzwerkschicht	12
1.2.1.5	Die Transportschicht	13
1.2.1.6	Die Sitzungsschicht	13
1.2.1.7	Die Darstellungsschicht	14
1.2.1.8	Die Anwendungsschicht	14
1.2.1.9	Dienste für die Kommunikation zwischen den Schichten	15
1.2.1.10	Beispiel: Ablauf einer Kommunikation im OSI-Modell	15
1.2.2	Das TCP/IP-Protokoll	17
1.3	Buszugriffsverfahren	19
1.3.1	Master/Slave-Verfahren	20
1.3.2	Token-Prinzip	22
1.3.3	Token-Passing	24
1.3.4	CSMA	25
1.3.5	CSMA/CA	28
1.3.6	Busarbitration	29
1.4	Datensicherung	30
1.4.1	Einleitung	30
1.4.2	Fehlerarten	31

1.4.3	Einige grundlegende Beziehungen	31
1.4.3.1	Bitfehlerrate	31
1.4.3.2	Wiederholung einer Übertragung	31
1.4.3.3	Restfehlerrate	32
1.4.3.4	Hamming-Distanz	32
1.4.3.5	Telegrammübertragungseffizienz	33
1.4.4	Einige Strategien der Fehlererkennung	34
1.4.4.1	Paritätsbit	34
1.4.4.2	Blocksicherung	35
1.4.4.3	CRC	37
1.4.5	Datenintegritätsklassen	40
1.4.6	Telegrammformate	41
1.4.6.1	Telegramm mit Paritätsbit	41
1.4.6.2	Telegramm mit CRC	42
1.5	Telegrammformate	44
1.5.1	Das HDLC-Protokoll	44
1.5.2	UART	47
1.5.3	PROFIBUS-Norm EN 50 170 Teil 2	48
1.5.4	HART-Protokoll	49
1.5.4.1	Smart-Transmitter	50
1.5.4.2	Busaufbau	51
1.5.4.3	Buszeiten	52
1.5.5	Token-Telegramm	53
1.5.6	Ethernet-Telegramm	53
1.5.6.1	Die Felder des Ethernet-Telegramms	53
1.5.6.2	TCP/IP-Protocol	55
1.6	Binäre Informationsdarstellung	55
1.6.1	NRZ, RZ	56
1.6.2	Bipolar-Kodierung, HDB _n -Kodierung	57
1.6.3	NRZI	58
1.6.4	AFP	59
1.6.5	Manchester-II-Kodierung	59
1.6.6	FSK, ASK, PSK	60
1.7	Übertragungsstandards	60
1.7.1	RS 232-, V.24-Schnittstelle	60
1.7.2	RS 422-Schnittstelle	63
1.7.3	RS 485-Schnittstelle	63
1.7.4	20 mA-Stromschleife	66
1.7.5	IEC 61158-2, FISCO-Modell	67
1.7.6	Ethernet-Übertragungsarten	72
1.7.6.1	10 MBd Ethernet	73
1.7.6.2	100 MBd-Ethernet (Fast Ethernet)	75

1.7.6.3	1000Base Ethernet (Gigabit Ethernet)	76
1.7.6.4	10GigaBit Ethernet	78
1.7.6.5	Infrastruktur-Komponenten	78
1.8	Leitungen und Übertragungsarten	79
1.8.1	Übersicht über die Leitungsarten	79
1.8.2	Paralleldrahtleitung	80
1.8.3	Koaxialleitung	84
1.8.4	Lichtwellenleiter (LWL)	86
1.8.5	Übertragungsarten	87
1.8.5.1	Basisbandübertragung	87
1.8.5.2	Trägerfrequenzübertragung	87
1.8.5.3	Breitbandübertragung	88
1.9	Verbindung von Netzen	89
1.9.1	Repeater	89
1.9.2	Bridges	90
1.9.3	Router	92
1.9.4	Gateways	94
1.10	Feldbusankopplung an Host-Systeme	95
1.10.1	Grundlagen	95
1.10.2	SPS-Ankopplung	96
1.10.2.1	Feldbusfunktionen auf Kommunikations-Baugruppen	96
1.10.2.2	Software-Schnittstelle	97
1.10.2.3	Einheitliche Programmierung mit IEC 1131	98
1.10.3	PC-Ankopplung	99
1.10.3.1	Hardware-Aspekte	99
1.10.3.2	Techniken des Anwenderzugriffs	100
1.10.4	Controller-Ankopplung	104
1.10.5	Ankopplung an höhere Netze über Gateways	104
1.10.6	Host-Zugriffe unter MMS	105
1.11	Buszykluszeiten	106
1.11.1	Deterministische Bussysteme	106
1.11.2	Nichtdeterministische Bussysteme	109
1.12	Sicherheitsbussysteme	110
Literatur	113
2	Netzwerkhierarchien in der Fabrik- und Prozessautomatisierung	117
2.1	Übersicht und Spezifik der Kommunikation in der Automatisierung	117
2.1.1	Fabrikautomatisierung	122
2.1.2	Prozessautomatisierung	123
2.2	Kommunikationsebenen bei Stückgut- und Fließgutprozessen	126
2.2.1	Stückgutprozesse	126
2.2.2	Fließgutprozesse	127

2.3	Managementebene	129
2.4	Prozesselebene	131
2.4.1	ETHERNET für den Industrieeinsatz	131
2.4.2	Lösungsmöglichkeiten mit TCP/IP	134
2.4.3	ETHERNET-Anwendungen in der industriellen Kommunikation	137
2.5	Feldebene	138
2.5.1	Anforderungen an ein Bussystem der Feldebene	138
2.5.2	Industrielösungen für Busse der Feldebene	139
2.6	Sensor-Aktor-Ebene	140
2.6.1	Anforderungen im Sensor-Aktor-Bereich	140
2.6.2	Industrielösungen für Sensor-Aktor-Bussysteme:	142
2.7	Fazit	142
	Literatur	143
3	Feldbusnormung	145
3.1	Internationale Normungsarbeit	145
3.2	Europäische Normungsarbeit	149
	Literatur	150
4	Beispiele ausgeführter Bussysteme	151
4.1	Sensor/Aktor-Busse	151
4.1.1	AS-Interface – Aktor/Sensor-Interface	151
4.1.1.1	Konzept des intelligenten Verkabelungs-Systems	151
4.1.1.2	Der Master	152
4.1.1.3	Netzteil	152
4.1.1.4	Elektromechanik	154
4.1.1.5	Netzwerktopologie	155
4.1.1.6	Slaves	155
4.1.1.7	Bitübertragung	156
4.1.1.8	Buszugriffsverfahren	157
4.1.1.9	A/B- und Single Slaves	158
4.1.1.10	Analogwertübertragung	159
4.1.1.11	Datensicherheit	161
4.1.1.12	Elektromagnetische Verträglichkeit	161
4.1.1.13	AS-Interface im explosionsgefährdeten Bereich	163
4.1.1.14	Neuerungen nach der Spezifikation 3.0	164
4.1.1.15	Safety at work	165
4.1.2	Das KNX-System für die Haus- und Gebäudesystemtechnik	167
4.1.2.1	Einführung	167
4.1.2.2	Netzwerktopologie	168
4.1.2.3	Übertragungsmedien	169
4.1.2.4	KNX OSI Kommunikationsprotokoll	171

4.1.2.5	Netzverwaltung und Adressierung	173
4.1.2.6	Datenformate und Interworking	174
4.1.2.7	Werkzeugsätze und Software Engineering	176
4.1.2.8	Weitere Systemmerkmale	176
4.1.2.9	Spektrum verfügbarer Produkte	177
4.2	Feldbusse	178
4.2.1	Sercos	178
4.2.1.1	Einleitung	178
4.2.1.2	Topologie	179
4.2.1.3	Übertragungsverfahren, Synchronisation und Protokollstruktur	181
4.2.1.4	Implementierung	185
4.2.2	PROFIBUS	186
4.2.2.1	PROFIBUS als „System-Baukasten“	186
4.2.2.2	Übertragungstechnik	189
4.2.2.3	Kommunikation	191
4.2.2.4	Allgemeine Applikationsprofile	197
4.2.2.5	Spezifische Applikationsprofile	199
4.2.2.6	Gerätemanagement	202
4.2.2.7	PROFIBUS Implementierung	202
4.2.2.8	Qualitätssicherung und Zertifizierung	203
4.2.3	Interbus	203
4.2.3.1	Topologie	203
4.2.3.2	Interbus-Protokoll	206
4.2.3.3	Protokollrealisierung	208
4.2.3.4	Anwendungsschnittstelle	211
4.2.4	Modbus-RTU und Modbus-ASCII	212
4.2.5	LON	214
4.2.5.1	LON-Gerät	214
4.2.5.2	LonWorks-Protokoll	218
4.2.5.3	Funktionsprofile für LON-Geräte	226
4.2.5.4	LON-übergreifende Kommunikation	228
4.2.5.5	Netzwerk-Management und Tools	228
4.2.6	CAN-basierende Netzwerke	229
4.2.6.1	Grundlagen	229
4.2.6.2	Physikalische Übertragung	230
4.2.6.3	CAN-Protokoll	231
4.2.6.4	CANopen	234
4.2.6.5	Devicenet	248
4.2.7	FOUNDATION Fieldbus H1	254
4.2.7.1	Übersicht	254

4.2.7.2 Die Feldbusphysik	255
4.2.7.3 Die Kommunikation	260
4.2.8 ControlNet	264
4.2.8.1 Zielanwendungen	266
4.2.8.2 Das ControlNet-Protokoll	266
4.3 Ethernetbasierte Feldbusse	270
4.3.1 Industrial Ethernet – Was ist das eigentlich?	270
4.3.2 Grundlagen des Ethernet	271
4.3.3 Ethernet im Vergleich zu modernen Feldbussystemen	275
4.3.4 PROFINET	281
4.3.4.1 PROFINET im Überblick	283
4.3.4.2 Grundlagen von PROFINET IO	284
4.3.4.3 IRT-Kommunikation bei PROFINET IO	291
4.3.4.4 PROFINET IO-Controller und -Devices	293
4.3.4.5 Conformance Classes (CC)	294
4.3.4.6 Applikationsprofile für PROFINET IO	294
4.3.4.7 Integration von Feldbus-Systemen	296
4.3.4.8 Netzwerkinstallation	296
4.3.4.9 PROFINET IO-Zertifizierung	300
4.3.5 Ethernet/IP	301
4.3.6 Echtzeit-Ethernet: Powerlink	305
4.3.7 Modbus-TCP	314
4.3.8 Echtzeit Ethernet EtherCAT	316
4.4 Peripheriebusse am PC	327
4.4.1 Vergleich USB – Firewire	327
4.4.2 USB	329
4.4.3 Bluetooth	333
Literatur	335
5 Weitverkehrsnetze	339
5.1 ISDN	339
5.2 DSL – Öffentliches Breitbandnetz	343
Literatur	349
6 Installationsbeispiele aus der Bus-Praxis	351
6.1 Verbindung von Feldgeräten über PROFIBUS und OPC mit Anwendersoftware	351
6.1.1 Kurze Einführung in OPC	351
6.1.2 Die Aufgabe: PROFIBUS an Visualisierungssoftware	352
6.1.3 Konfiguration des PROFIBUS	353
6.1.4 Konfiguration des OPC-Servers	354
6.1.5 SCADA-Projekt und OPC-Konfiguration	354

6.2	Prozesssteuerung über das Internet-Netzwerk	358
6.2.1	Die Aufgabe	358
6.2.2	Erstellung der LabView-Applikation	359
6.2.3	Internetanbindung	360
6.2.4	Die Konfiguration des HTTP-Servers	364
6.3	Konfiguration AS-i/Interbus-Gateway an Interbus	365
6.3.1	Aufbau der Bus-Systeme	365
6.3.2	Konfiguration des AS-i	367
6.3.3	Kommunikation des AS-i/Interbus-Gateway mit dem Interbus	367
6.3.4	Die sw-Verknüpfung Interbus/AS-i	369
6.4	Die Verbindung einer SPS mit dem PROFIBUS DP	371
6.4.1	S7-Projekt	373
6.4.2	Konfiguration der S7-Station	373
6.4.3	Kommunikation zwischen CPU und CP	377
6.4.4	Programmbeispiel	378
6.5	Konfiguration AS-i/Ethernet/IP-Gateway an Ethernet/IP	379
6.5.1	Aufbau der Bussysteme	379
6.5.2	Konfiguration des AS-i-Netzwerks und des AS-i/Ethernet/IP-Gateways	380
6.5.3	Kommunikation über Ethernet/IP	381
6.5.4	Die Software-Verknüpfung zwischen AS-i und Ethernet/IP	382
7	Datenblätter	389
7.1	AS-i (Aktor/Sensor-Interface)	389
7.2	KNX (ehemals EIB, European Installation Bus)	390
7.3	Sercos I, II und III	391
7.4	PROFIBUS	392
7.4.1	PROFIBUS-DP	392
7.4.2	PROFIBUS-PA	393
7.5	Interbus	394
7.6	Modbus Plus	395
7.7	Industrial Ethernet	396
7.8	LON (Local Operating Network)	397
7.9	CAN (Controller Area Network)	398
7.10	Foundation Field Bus	399
7.11	Eigensichere Feldbusse	400
7.11.1	PROFIBUS PA	400
7.11.2	PROFIBUS (DP) Ex-i	400
7.12	DeviceNet	401
7.13	ControlNet	402
7.14	EtherNet/IP	403
	Stichwortverzeichnis	405

Mitarbeiterverzeichnis

Die Herausgeber

Prof. Dr. Ing. Gerhard Schnell Ehem. Fachhochschule Frankfurt am Main, Frankfurt am Main, Deutschland

Dipl. Ing. Bernhard Wiedemann Geschäftsführer Bihl und Wiedemann GmbH, Mannheim, Deutschland

Die Koautoren

Dipl. Ing. Roland Bent Phoenix Kontakt, Blomberg, Deutschland

Prof. Dr. Ing. Jürgen Beuschel Fachhochschule für Technik und Wirtschaft, Berlin, Deutschland

Prof. Dr. Ing. Jörg Böttcher b-plus, Deggendorf, Deutschland

Dipl. Ing. Michael Brill Schneider Automation, Seligenstadt, Deutschland

Dr. Ing. Hans Endl Softing GmbH, Haar bei München, Deutschland

Dipl. Phys. Marc Goosens EIB Association, Brüssel, Belgien

Dipl. Ing. Wolfgang Grote Fachhochschule Frankfurt am Main, Frankfurt am Main, Deutschland

Prof. Dr. Ing. Thilo Heimbold Hochschule für Technik, Wirtschaft, Kultur, Leipzig, Deutschland

Dipl. Ing. Andreas Hennecke Pepperl+Fuchs GmbH, Mannheim, Deutschland

Dipl. Ing. Norbert Heinlein Fachhochschule Frankfurt am Main, Frankfurt am Main, Deutschland

Dipl. Ing. Thomas Klatt Pepperl+Fuchs GmbH, Mannheim, Deutschland

Prof. Dr. habil. Werner Kriesel Hochschule für Technik, Wirtschaft, Kultur, Leipzig, Deutschland

Heinz Lux CIO KNX-Association, Brüssel, Belgien

Dipl. Ing. Peter Lutz Sercos International, Süßen, Deutschland

Dipl. Ing. Anton Meindl B&R Industrie-Elektronik, Eggelsberg, Österreich

Martin Rostan Executive Director EtherCat Technology Group, Nürnberg, Deutschland

Dipl. Ing. Reinhard Simon Rockwell Automation, Haan-Gruiten, Deutschland

Dipl. Ing. Alexander Stamm Ehem. Fachhochschule Frankfurt am Main, Frankfurt am Main, Deutschland

Dr. Peter Wenzel Profibus Nutzerorganisation (PNO), Karlsruhe, Deutschland

Dipl. Ing. Holger Zeltwanger CiA, CAN in Automation, Nürnberg, Deutschland



Technische Grundlagen

1

1.1 Netzwerktopologien

Um beliebige Prozesse effektiver gestalten zu können, ist es notwendig, dass die Einheiten, die den Prozess überwachen bzw. steuern, untereinander Informationen austauschen. Dabei ist es unerheblich, ob es sich bei den Überwachungseinrichtungen um technische Geräte wie z. B. Rechner oder SPS-Geräte oder um Menschen handelt.

Verknüpft man Rechner, SPS-Geräte etc. derart miteinander, dass über die entstehenden Verbindungsleitungen Informationen übertragen werden können, entsteht ein Netzwerk. Unter dem Begriff Netzwerktopologie versteht man zum einen die geometrische Anordnung der Teilnehmer im Netzwerk und zum anderen die logische Anordnung der Teilnehmer, unabhängig von der Geometrie.

Im Folgenden soll auf die unterschiedlichen geometrischen Anordnungsmöglichkeiten eingegangen werden.

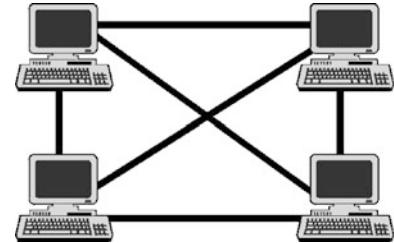
1.1.1 Zweipunktverbindungen

Die einfachste Möglichkeit, Daten auszutauschen besteht darin, genau zwei Kommunikationspartner über eine Leitung miteinander zu verbinden, wie z. B. bei einer Gegensprech-anlage, einem Modem oder der Verbindung zwischen PC und Drucker. Die notwendige Steuerung eines Kommunikationsprozesses ist einfach über Steuer-, Melde- und Taktleitungen zu realisieren (Handshake-Betrieb).

Werden mehrere Teilnehmer mit Zweipunktverbindungen verknüpft, entsteht ein vermaschtes Netz nach Abb. 1.1.

Bei dieser Topologie besteht zwischen zwei kommunizierenden Teilnehmern eine Zweipunktverbindung. Dabei werden bei n Teilnehmern $(n - 1)$ Schnittstellen pro Teilnehmer und $(^n_2)$ Verbindungsleitungen benötigt. Daraus resultiert, dass die Kosten einer solchen Topologie sehr hoch sind.

Abb. 1.1 Prozesskommunikation/Vermaschtes Netz



Im Falle eines Fehlers würde entweder nur ein Teilnehmer oder nur ein Kommunikationskanal ausfallen, und die Diagnose wäre sehr einfach.

1.1.2 Zweipunktverbindungen mit Multiplexer

Soll eine Zweipunktverbindung von mehr als zwei Teilnehmern benutzt werden, müssen Maßnahmen getroffen werden, die eine gegenseitige Signalbeeinflussung und damit eine Zerstörung der Signale verhindern. Eine Möglichkeit, dies zu erreichen, stellt das Zeitmultiplex-Verfahren, eine andere Möglichkeit das Frequenzmultiplexverfahren dar. Wird das Zeitmultiplex-Verfahren angewendet, spricht man von einer Basisbandübertragung, da hier das unmodulierte Signal im Frequenzband von 0 Hz bis zur Grenzfrequenz des Trägermediums zur Verfügung steht.

Bei Verwendung des Frequenzmultiplex-Verfahrens wird ein moduliertes Signal mit einer definierten Bandbreite übertragen.

Das Prinzip des Zeitmultiplexverfahrens ist in Abb. 1.2 dargestellt.

Im Multiplexer (MUX) werden mit Hilfe der Steuersignale a_0 und a_1 die Daten $d_0 \dots d_3$ nacheinander auf die Übertragungsleitung y geschaltet. Damit wird jedem der vier Teilnehmer ein Zeitschlitz zugewiesen, in dem er seine Übertragung vornehmen kann. Der De-

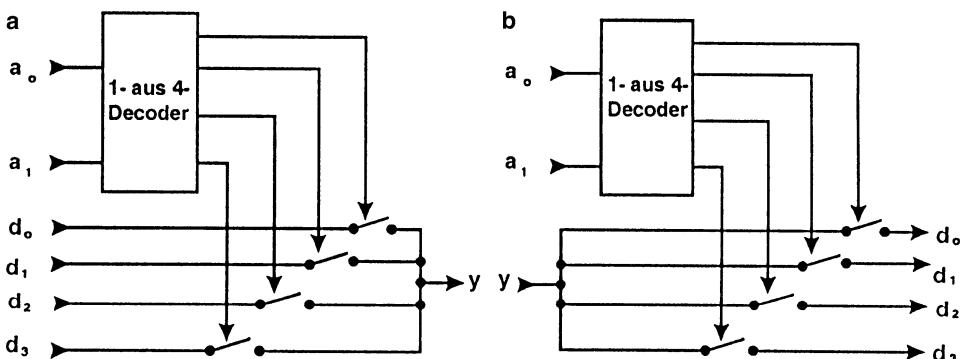


Abb. 1.2 Prinzipielle Funktionsweise eines **a** Multiplexers, **b** Demultiplexers

multiplexer (DEMUX) sorgt mit Hilfe der selbsterzeugten Steuersignale a_0 und a_1 dafür, dass die seriell empfangenen Daten wieder auf die Datenleitungen $d_0 \dots d_3$ geschaltet werden. Um dies problemlos realisieren zu können, müssen beide 1- aus 4-Decoder synchron laufen. Dies wird dadurch erreicht, dass zu Beginn jeder Übertragungsperiode ein Synchronisationssignal über die Datenleitung gesendet wird.

In der oben gewählten Darstellung ist die Anordnung nur für den so genannten Simplexbetrieb geeignet. Darunter versteht man die Nachrichtenübertragung ausschließlich in eine Richtung.

Der Multiplexer und der Demultiplexer unterscheiden sich vom Funktionsprinzip nicht. Um einen bidirektionalen Datenaustausch zu ermöglichen benötigt man nur eine MUX/DEMUX Einrichtung, bei der zwischen „kommender“ und „gehender“ unterscheiden kann. Eine solche Einrichtung ist dann Halbduplex-fähig, dass heißt, dass Daten nacheinander in beide Richtungen übertragen werden können. In der industriellen Praxis wird das Multiplexverfahren sehr häufig in Verbindung mit dem HART-Protokoll (siehe Abschn. 1.5.4) in Form von so genannten HART-Multiplexern eingesetzt. Diese sorgen dafür, dass eine Punkt-zu-Punkt-Verbindung zwischen einer Anzeige und Bedienkomponente und einem HART-fähigen Feldgerät hergestellt wird. Über Hart-Multiplexer können bis zu 7936 Verbindmöglichkeiten verwaltet werden.

Beim Frequenzmultiplex-Verfahren wird der Übertragungskanal in voneinander unabhängige Frequenzbänder mit definierter Bandbreite eingeteilt. Damit besteht die Möglichkeit, mehrere Signale gleichzeitig bidirektional zu übertragen. Diese Vorgehensweise eignet sich zur Vollduplex-Übertragung.

Als Modulationsarten eignen sich Amplituden-, Frequenz- und Phasenmodulation. Der Vorteil liegt in der optimalen Nutzung des Übertragungsmediums. Da die zur Modulation benötigten Baugruppen relativ teuer sind, findet diese Breitbandübertragung ihre Anwendung hauptsächlich in so genannten Weitverkehrsnetzen (*Wide Area Networks, WAN*).

1.1.3 Bus-Struktur

Bei der Bus-Struktur, auch Liniенstruktur genannt, kommunizieren alle Teilnehmer über eine gemeinsame Leitung (Abb. 1.3).

Die Anbindung der Teilnehmer an das Buskabel geschieht über kurze Stichleitungen (Dropkabel). Dadurch wird der Kabelaufwand, verglichen mit dem vermaschten Netz, erheblich reduziert. Jeder Teilnehmer benötigt hier nur noch eine Schnittstelle, um mit einem beliebigen, an den Bus angeschlossenen Teilnehmer kommunizieren zu können. Hier entsteht allerdings das Problem, dass immer nur ein Teilnehmer zu einem bestimmten Zeitpunkt senden darf. Damit werden Regeln notwendig, die das Zugriffsrecht auf den Bus festlegen, so genanntes Buszugriffsverfahren.

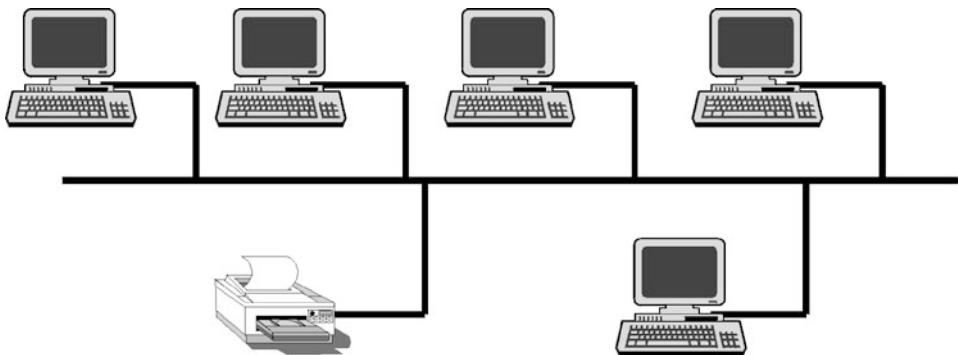


Abb. 1.3 Linienstruktur

Bei Verwendung der Bus-Struktur treten folgende Probleme auf.

1. Da ein beliebiger Datenverkehr gefordert ist, müssen alle Teilnehmer jede Sendung „mithören“. Dadurch wird bei steigender Teilnehmerzahl der Sender immer stärker belastet, da die Empfänger parallel geschaltet sind.
2. Die Übertragungsstrecken für Feldbussysteme liegen häufig in einem Bereich von wenigen hundert Metern, in der Prozessautomatisierung manchmal auch im Kilometerbereich. Damit ist die Leitungslänge nicht mehr vernachlässigbar klein gegenüber der zu übertragenden Wellenlänge. Damit muss die Busleitung an beiden Enden mit ihrem Wellenwiderstand abgeschlossen werden, um Reflexionen auf der Leitung zu vermeiden, die die Signalqualität erheblich beeinflussen könnten. Dieser Abschlusswiderstand belastet ebenfalls den Sender.

Die hier angeführten Gründe haben zur Folge, dass die Teilnehmerzahl an einer Busleitung begrenzt ist. In der Regel sind dies 32 Teilnehmer pro Bussegment. Manche Feldbussysteme verzichten bewusst auf Abschlusswiderstände (z. B. AS-Interface). Dies hat zur Folge, dass Reflexionen an den Leitungsenden auftreten. Um einen negativen Einfluss auf die Signalqualität zu vermeiden, wird die maximale Leitungslänge, wie im Falle AS-Interface, begrenzt. In einem solchen Fall gilt:

$$l \leq \lambda_{\max} \div 10$$

mit l = maximale Leitungslänge, λ maximal auftretende Wellenlänge.

Ein weiteres Problem soll mit Abb. 1.4 verdeutlicht werden.

Ausgehend von einer Leitung mit vernachlässigbarem Induktivitäts- und Leitwertbeleg, stellt diese ein einfaches RC-Glied dar. Dabei sind der Leitungswiderstand R_{Leitung} und die Leitungskapazität C_{Leitung} von der Leitungslänge abhängig.

Erzeugt der Sender zum Zeitpunkt t_0 einen Spannungssprung, so hat die Spannung U_{Last} einen exponentiellen Verlauf. Die Zeitkonstante und damit die Steigung der Funk-

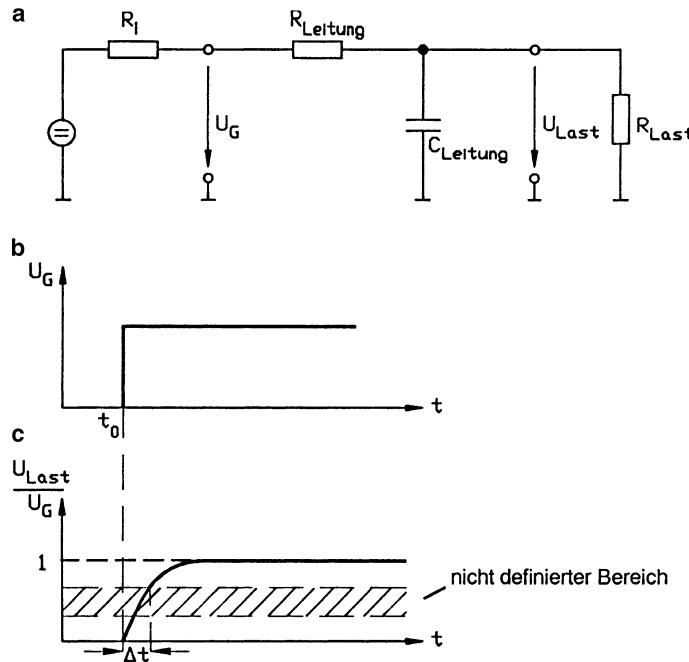


Abb. 1.4 Impulsverzerrung auf einer Leitung: **a** Ersatzschaltbild der Anordnung **b** Ausgangsspannung des Generators **c** Empfängerspannung

tion zum Zeitpunkt t_0 wird durch R_{Last} , R_{Leitung} , R_i und C_{Leitung} bestimmt. Damit der Empfänger eine Änderung des logischen Zustandes akzeptiert, muss die Spannung U_{Last} den nichtdefinierten Bereich komplett durchlaufen. Dazu wird die Zeit Δt benötigt, die von den Kabelkennwerten abhängig ist. Wird die Leitung verlängert, steigen der Widerstands- und Kapazitätswert der Leitung, was zur Folge hat, dass auch Δt größer wird. Ist nun die maximal auftretende Frequenz

$$f_{\max} \geq 1/\Delta t ,$$

hat dies zur Folge, dass die Spannung U_{Last} den nichtdefinierten Bereich nicht mehr komplett durchlaufen kann. Damit kann der Empfänger den Wechsel des logischen Zustandes nicht mehr registrieren.

In der Praxis hat dies zur Konsequenz, dass die maximale Übertragungsrate und die maximale Leitungslänge miteinander verknüpft sind. So lässt z. B. der PROFIBUS bei einer Übertragungsrate von 93,75 kBd eine Leitungslänge von 1200 m zu, während bei einer Übertragungsrate von 500 kBd nur noch 400 m realisierbar sind und bei 12 MBd beträgt die maximale Leitungslänge noch 100 m.

Höhere Übertragungsraten und Leitungslängen sind bei Verwendung von Lichtwellenleitern (LWL) erreichbar. Jedoch ist in diesem Fall die Ankopplung der Teilnehmer an die Busleitung kompliziert und teuer.

1.1.4 Baumstruktur

Bei der Baumstruktur handelt es sich um eine Weiterentwicklung der Linienstruktur. Mit dieser Topologie sind größere Flächen als bei der Bustopologie vernetzbar (Abb. 1.5).

Die Ausführungen bzgl. der maximalen Leitungslänge, der maximalen Teilnehmerzahl und der maximalen Übertragungsrate gelten wie bei der Busstruktur.

Diese Werte können mit so genannten Repeatern vergrößert werden. Bei diesen Elementen handelt es sich um Verstärkerelemente, die bidirektional arbeiten. Bei der Baumstruktur werden sie zur Bildung eines neuen Zweiges verwendet, um z. B. die Übertragungsleitung zu verlängern oder um mehr als die üblicherweise 32 Teilnehmer pro Segment anschließen zu können.

Durch die größeren Leitungslängen ist eine galvanische Trennung der Teilnehmer voneinander notwendig. Diese wird in der Regel im Eingang eines jeden Teilnehmers vorgenommen, wobei der Repeater ein Teilnehmer ist, der jedoch keine Adresse benötigt. Durch die galvanische Trennung werden nur Probleme beseitigt, die aufgrund von Potenzialunterschieden längs der Busleitung und den daraus resultierenden Ausgleichströmen entstehen.

Verwendet man Sender, die einen differentiellen Spannungsausgang besitzen und Empfänger mit Differenzspannungseingang, kann man Störungen aufgrund elektromagnetischer Einkopplungen weitestgehend unterdrücken (Abb. 1.6).

Unter der Voraussetzung, dass es sich bei der Leitung um eine verdrillte Zweidrahtleitung handelt, kann man davon ausgehen, dass sich elektromagnetische Einkopplungen auf beide Leitungen gleichmäßig auswirken. Damit wirkt sich dies nicht mehr auf den

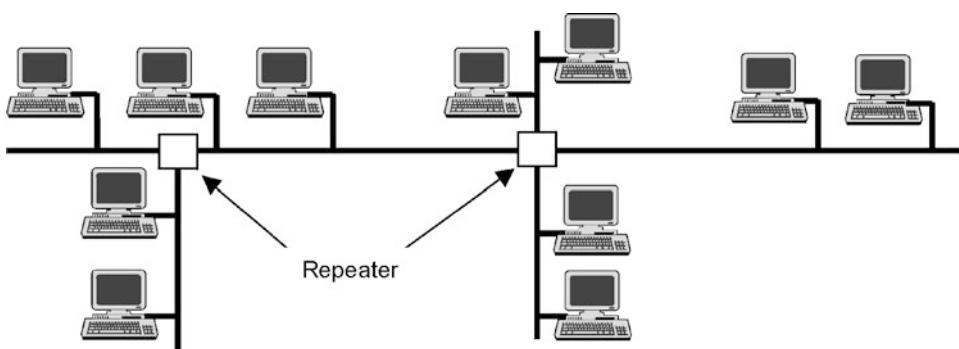


Abb. 1.5 Baumstruktur

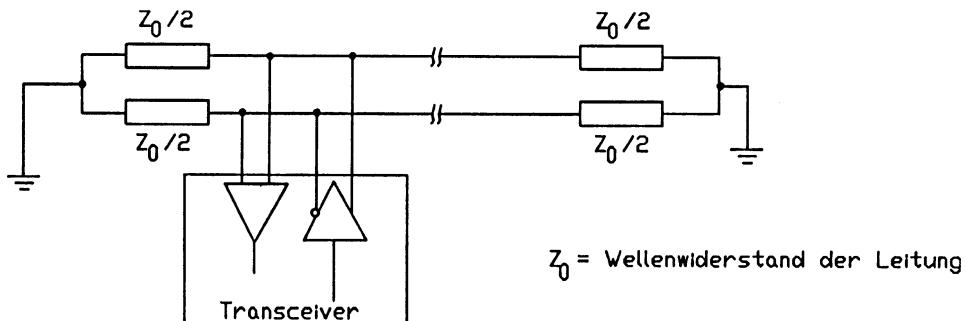


Abb. 1.6 Ankopplung mit Differenzspannungs-Signal

Eingang bzw. Ausgang des Transceivers (Kunstwort aus Transmitter = Sender und Receiver = Empfänger) aus. Eine weitere Verbesserung wird durch die Schirmung der Zweidrahtleitung erreicht. Sollten diese Schutzmaßnahmen nicht ausreichen kommen Lichtwellenleiter (LWL) zum Einsatz.

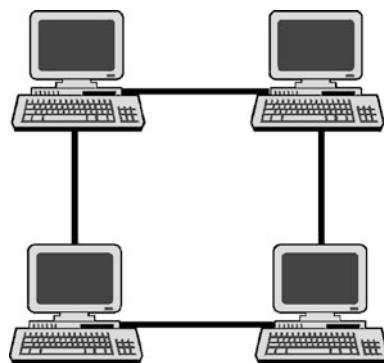
1.1.5 Ringstruktur

Mittels mehrerer Zweipunktverbindungen wird bei dieser Struktur ein physikalischer Ring aufgebaut (Abb. 1.7).

Dabei wird die zu übertragende Information von Teilnehmer zu Teilnehmer weitergebracht. Auch hier muss durch ein Buszugriffsverfahren sichergestellt sein, dass nur ein Teilnehmer zu einem bestimmten Zeitpunkt sendet.

Dadurch, dass die Ringstruktur aus Zweipunktverbindungen aufgebaut ist und jeder Teilnehmer als Repeater wirken kann, können hier relativ große Entfernung überbrückt werden. Diese liegen zwischen zwei Teilnehmern bei Verwendung von LWL im Kilometerbereich, bei gleichzeitig sehr hohen Datenraten. Beispielsweise gestattet das Bus-

Abb. 1.7 Ringstruktur



System Industrial Ethernet (Siemens) einen Ringumfang von 100 km bei einer Übertragungsrate von 100 MBd.

Problematisch ist diese Topologie bei Ausfall eines Teilnehmers bzw. bei Leitungsbruch oder Kurzschluss. Ohne geeignete Gegenmaßnahmen würde dies hier bedeuten, dass das gesamte Netz ausfallen würde.

Wird der Ring redundant ausgelegt, sodass in beide Richtungen übertragen werden kann, können defekte Stellen umgangen werden. Durch geeignete Suchmechanismen können diese lokalisiert und mittels Kurzschlussbrücken aus dem Ring ausgeschlossen werden.

1.1.6 Sternstruktur

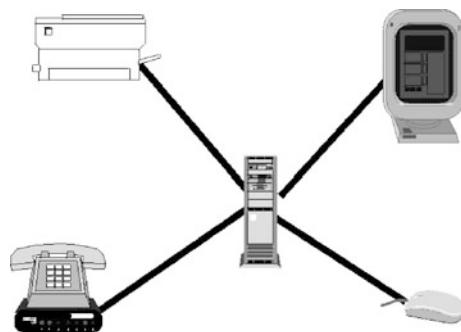
Hier ist die Zentralstation mittels Zweipunktverbindung mit jedem anderen Teilnehmer verbunden (Abb. 1.8).

Es existieren zwei Möglichkeiten, die Zentralstation zu realisieren. Zum einen der so genannte Sternkoppler (Hub), dessen Aufgabe es ist, die Signale ausschließlich vom Sender zum richtigen Empfänger weiterzuleiten. Dabei kann der Hub sowohl passiv sein als auch aktiv, d. h., die empfangenen Signale werden vor der Weiterleitung noch aufbereitet.

Die zweite Möglichkeit ist, in der Zentralstation Intelligenz zu implementieren. Damit könnte diese Station die Steuerung des gesamten Kommunikationsprozesses übernehmen, die im ersten Fall von einem oder allen anderen Teilnehmern vorgenommen werden müsste.

Prinzipiell findet die Kommunikation zwischen zwei Teilnehmern über die Zentralstation statt. Damit stellt diese einen Engpass dar. Ein Ausfall dieser Station hat zur Folge, dass das gesamte Netz ausfällt. Ein klassisches Beispiel für eine Sternstruktur mit aktiver Zentralstation ist der PC. Sämtlicher Datenaustausch zwischen den PC-Komponenten wird über die CPU organisiert; fällt beispielsweise die Maus aus, kann der PC weiterhin verwendet werden, fällt jedoch die CPU aus, ist der PC arbeitsunfähig.

Abb. 1.8 Sternstruktur



1.2 Kommunikationsmodelle

1.2.1 Das ISO/OSI-Referenzmodell

1.2.1.1 Allgemeines

Kommunikation von Rechnern untereinander zum Zwecke des Datenaustausches erfordert vorherige Vereinbarungen darüber, in welcher Art und Weise sie stattfinden soll. Die Betrachtung unterschiedlicher Kommunikationen zeigt, dass die Schemata des Ablaufs sich oftmals ähneln. Die immer stärker wachsende Bedeutung der Kommunikation in der Weltwirtschaft veranlasste in den siebziger Jahren die International Standards Organization (ISO), eine Institution der United Nations Organization (UNO), eine Arbeitsgruppe zu bilden, die sich mit der Standardisierung von Rechnerkommunikationen befasste.

Die Arbeit dieses Komitees führte 1983 zur ISO-Norm 7498 (später auch vom Comité Consultatif International Télégraphique et Téléphonique CCITT als X.200 übernommen), einem Referenzmodell für Rechnerkommunikation mit dem Titel „*Basic Reference Model for Open Systems Interconnection (OSI)*“. Sie beschreibt die Kommunikation von offenen Systemen, d. h. von Systemen, die für diese Art der Kommunikation offen sind. Dies ist nicht mit offener Kommunikation gleichzusetzen.

Das OSI-Referenzmodell teilt die Kommunikation abstrakt in sieben Ebenen (Schichten) mit festgelegter Funktionalität, daher wird das Modell auch als OSI-Schichtenmodell bezeichnet. Jeder Schicht kommt die Übernahme einer speziellen, klar definierten Gruppe von Teilaufgaben in der Kommunikation zu. In jedem der beteiligten Kommunikationspartner sind alle sieben Schichten enthalten. Die Schichten kommunizieren untereinander über genau definierte Schnittstellen, was den Austausch einzelner Schichten ohne Eingriff in die Funktionalität des Gesamtsystems erleichtert. An diesen Schnittstellen stellt jede Schicht Dienste bereit, die von den Nachbarschichten in Anspruch genommen werden können.

Das OSI-Schichtenmodell beschreibt die Kommunikation von Partnerprozessen auf einer abstrakten Ebene. Es sind keine Angaben darüber enthalten, wie die einzelnen Schichten letztendlich implementiert werden sollen. Damit ist die Basis für ein offenes System geschaffen, das durch die Definition der Inhalte der einzelnen Schichten und durch die Festlegung der Schnittstellen auch dann genutzt werden kann, wenn ein Gesamtsystem aus Komponenten mehrerer Hersteller zusammengefügt wird. Die Schichteneinteilung dient der Abstraktion der Kommunikationsprozesse.

Die Aufgliederung der Rechnerkommunikation in sieben Ebenen ist in Abb. 1.9 dargestellt und im folgenden Text beschrieben.

Die Beschreibung der Schichten des ISO-Referenzmodells beginnt bei Schicht 1 und endet mit Schicht 7. Innerhalb des Modells ist eine Zweiteilung vorhanden. Die Schichten 1 bis 4 sind für die Datenübertragung zwischen den Endgeräten zuständig, während die Schichten 5 bis 7 bei der Datenübertragung das Zusammenwirken mit dem Anwendungsprogramm und dem Betriebssystem des verwendeten Rechners koordinieren.

a

Nr.	Bezeichnung	Erläuterungen
7	Anwendungsschicht (Application Layer)	stellt die auf dem Netzwerk basierenden Dienste für die Programme des Endanwenders bereit (Datenübertragung, elektronische Post usw.)
6	Darstellungsschicht (Presentation Layer)	legt die Anwenderdaten-Strukturen fest und konvertiert die Daten, bevor sie zur Sitzungs- bzw. Anwendungsschicht gegeben werden (Formatierung, Verschlüsselung, Zeichensatz)
5	Sitzungsschicht (Session Layer)	definiert eine Schnittstelle für den Auf- und Abbau von Sitzungen, d. h. zur Benutzung der logischen Kanäle des Transportsystems
4	Transportschicht (Transport Layer)	stellt fehlerfreie logische Kanäle für den Datentransport zwischen den Teilnehmern bereit
3	Netzwerkschicht (Network Layer)	transportiert die Daten von der Quelle zum Ziel und legt die Wege der Daten im Netz fest
2	Datenverbindungsschicht (Data Link Layer)	legt die Datenformate für die Übertragung fest und definiert die Zugriffsart zum Netzwerk. Sie wird in die "Zugriffssteuerung für das Medium" (MAC) und die "Logische Ankopplungs-Steuerung" (LLC) unterteilt
1	Physikalische Schicht (Physical Layer)	definiert die elektrischen und mechanischen Eigenschaften der Leitung, Pegeldefinition

b

3. Netzwerkschicht	Netzwerkverwaltung und Netz/Netz-Verwaltung	IEEE 802.1		
2. Datenverbindungsschicht	Logische Verknüpfungssteuerung	IEEE 802.2		
	Mediumszugriff-Steuerung	802.3	802.4	802.5
1. Physikalische Schicht	elektronischer und mechanischer Aufbau	CSMA/CD	Token-Bus	Token-Ring

Abb. 1.9 OSI-Modell (Open Systems Interconnection) von ISO (International Standardization Organisation). **a** Übersicht, **b** die IEEE-Standards der unteren 3 Schichten

Die oberen Schichten (5–7) werden daher auch als Anwendungsschichten, die unteren Schichten (1–4) als Übertragungsschichten oder Transportsystem bezeichnet.

1.2.1.2 Die physikalische Schicht oder Bitübertragungsschicht

Schicht 1 ist die Physikalische Schicht (*Physical Layer*). Sie bestimmt, in welcher Weise die Datenübertragung physikalisch zu erfolgen hat, d. h. die elektrischen und mechanischen Eigenschaften der Übertragung. In Schicht 1 wird vereinbart, wie die Übertragung der einzelnen Bits von statten geht. Dazu gehört die Art der Codierung (*Immediate Return to Zero, No Return to Zero Inverted, No Return to Zero, Manchester, FSK etc.*), der Spannungspegel für die Übertragung, die vereinbarte Zeitdauer für ein einzelnes Bit, die Wahl der Übertragungsleitung und der Endsystemkopplung (Stecker) und die Zuordnung der Anschlüsse (Pinbelegung) für die Übertragung des Bitstroms.

Die physikalische Schicht ist wie jede andere Schicht im System austauschbar, ohne dass die anderen Schichten davon betroffen sind. Die Kommunikation des Gesamtsystems ist unabhängig von der Ausprägung der einzelnen Schicht. Schicht 1 kann also z. B. eine Glasfaserstrecke betreiben, genauso wie eine elektrische Übertragungsstrecke nach RS232-, RS422- oder RS485-Norm oder jede beliebige andere Übertragungsstrecke. Schicht 1 ist nicht das physikalische Medium selbst, sondern derjenige Teil in der Übertragungsdefinition, der die physikalische Strecke definiert.

1.2.1.3 Die Sicherungsschicht

Schicht 2 ist die Sicherungsschicht der Leitungsebene (*Data Link Layer*). Ihre Aufgabe ist der sichere Transport der Daten von einer Station zu einer anderen Station. Sie dient damit der Datensicherung während der physikalischen Übertragung. Die Daten werden so verpackt, dass Übertragungsfehler von den teilnehmenden Stationen erkannt werden können.

Dazu werden die zu übertragenden Daten in Rahmen (*data frames*) eingeteilt, sodass in jedem Rahmen nur eine maximale Anzahl von Bytes enthalten sind. Rahmengrößen im Bereich von einigen hundert Bytes sind üblich. Die Rahmen enthalten außer den Rohdaten zusätzliche Informationen für die Übertragung, die die Sicherungsschicht ihrerseits zu den bereits vorhandenen Daten hinzufügt. Diese Zusatzinformation enthält z. B. eine Prüfsumme und Anfangs- und Endinformationen für den Rahmen. Außerdem kann die Zusatzinformation zur Quittierung von Telegrammen dienen, die bereits vom Kommunikationspartner übertragen wurden. Mit den hierbei verwendeten Mechanismen soll festgestellt werden, ob Rahmen fehlerhaft übertragen wurden oder ob Rahmen auf dem Übertragungsweg verloren gingen. Werden bereits verloren geglaubte Rahmen zum wiederholten Male gesendet, so ist in der Sicherungsschicht dafür Sorge zu tragen, dass sie beim Empfänger nicht dupliziert werden, d. h. dass dieser nicht annimmt, mehrere Rahmen anstatt mehrfach den gleichen Rahmen empfangen zu haben. Die Sicherungsschicht besitzt keine Kenntnis über den Inhalt der Information.

Die Sicherungsschicht stellt der nächsthöheren Ebene 3 einen logischen Kanal zur Verfügung, der ohne Übertragungsfehler funktioniert. Außerdem gleicht die Ebene 2 unterschiedliche Geschwindigkeiten der Datenverarbeitung (Lesen/Schreiben) bei Sender und Empfänger aus und kontrolliert damit den Datenfluss zwischen den beteiligten Stationen und verhindert ein „Überlaufen“ einer Station, falls eine Station schneller sendet, als die empfangende Station Daten weiterverarbeiten kann.

Die Sicherungsschicht wird in der IEEE 802 Norm in zwei Teilen beschrieben, der *Logical Link Control* (LLC) stellt die Dienste zur Kommunikation mit der Ebene 3 und der *Medium Access Control* (MAC) wird zur Anbindung der Schicht 1 benötigt.

Ein Beispiel für die Realisierung der Sicherungsschicht folgt an anderer Stelle mit der Beschreibung des HDLC-Protokolls.

1.2.1.4 Die Netzwerkschicht

Während in Schicht 2 die Kommunikation zwischen zwei Stationen betrachtet wurde, gilt in der dritten Schicht, der Netzwerkschicht (*Network Layer*), das gesamte Netzwerk als logische Einheit, das in seiner Gesamtheit bearbeitet wird. Die Aufgaben der Netzwerkschicht sind:

- der Transport von Daten von der Quelle bis zum Ziel, eventuell über Zwischenstationen,
- das Bereitstellen von Schnittstellen zwischen Endsystemen,
- das *Routing*, d. h. die Festlegung des Weges der Daten im Netz und die Wegsteuerung, was statisch oder dynamisch erfolgen kann und
- das Packen und Auspacken von Paketen, die von Schicht 2 verarbeitet werden können.

Die Netzwerkschicht hat dafür zu sorgen, dass Stauungen im unterliegenden Netzwerk vermieden werden, d. h. die Anzahl der gerade im Netz befindlichen Datenpakete muss von ihr kontrolliert werden.

Grundsätzlich werden dabei verbindungsorientierte und verbindungslose Dienste unterschieden. Ist der Dienst verbindungsorientiert, so stellt er dem Benutzer einen virtuellen Kanal zur Verfügung (*Virtual Circuit Service*). Der zugehörige Kommunikationsablauf besteht aus

- dem Verbindungsaufbau,
- dem Datenaustausch und
- dem Verbindungsabbau.

Solche Kommunikationsformen sind einem Telefongespräch vergleichbar, bei dem der Verbindungsaufbau nach Wahl der Teilnehmernummer hergestellt wird, der Datenaustausch durch Sprechen erfolgt und das Einhängen des Hörers den Abbau der Verbindung zur Folge hat.

Verbindungslose Dienste (*Datagram Service*) stellen keine Verbindung zwischen den Kommunikationspartnern her. Die zu übertragenden Datenpakete werden mit der vollständigen Zieladresse versehen ins Netz gegeben und dort weitertransportiert. Sie sind dem Briefverkehr ähnlich, bei dem ebenfalls Datenpakete (Briefe) mit einer vollständigen Zieladresse versehen an den dafür vorgesehenen Punkten (Briefkästen) ins Netz (Postdienst) gegeben werden und ohne Beeinflussung des Transportweges durch den Benutzer vom Netzwerkservice an der Zieladresse abgeliefert werden.

1.2.1.5 Die Transportschicht

Die 4. Ebene im OSI-Referenzmodell ist die Transportschicht (*Transport Layer*). Sie beschreibt die Kommunikation zwischen Prozessen, wie z.B. Programmen in Host-Rechner A und Host-Rechner B, die Daten miteinander austauschen.

Die Transportschicht hat folgende Einzelaufgaben:

- Namensgebung für die Host-Rechner,
- Adressierung der Teilnehmer,
- Aufbau und Abbau der Verbindung (bezüglich des Transports),
- Fehlerbehandlung und Flusskontrolle,
- *Multiplexing* verschiedener Datenströme auf einem Kanal,
- Synchronisation der Hosts,
- Wiederherstellung einer Verbindung bei Fehler im darunterliegenden Netzwerk.
- *Internetworking*.

Die Transportschicht zerlegt die Daten der nächsthöheren Ebenen in transportierbare Einheiten. Sie baut bei verbindungsorientierten Netzwerken die Verbindung zum Kommunikationspartner auf. Je nach gewünschter Eigenschaft wird für jede Transportverbindung eine eigene Netzverbindung, mehrere Netzverbindungen (bei hohem Datendurchsatz) oder für mehrere Transportverbindungen eine einzige Netzwerkverbindung (Sammelverbindung) bereitgestellt. Sammelverbindungen werden meist aus Kostengründen betrieben. Das Vorhandensein einer solchen Sammelverbindung ist für die höheren Schichten transparent.

Zu den höheren Ebenen bestehen so genannte *Service Access Points* mit Name und Adresse (SAP). Je nachdem, welche Dienste der Schicht 4 in Anspruch genommen werden, gibt es unterschiedliche Service-Klassen, die jeweils einen Teil der oben genannten Aufgaben enthalten.

Bei der Aufgabe des Internetworking in einem Gateway-Rechner (Host A an Netz 1 kommuniziert mit Host B an Netz 2 über diesen Gateway-Rechner) ist es Aufgabe der Transportschicht des Gateway-Rechners, die unterschiedlichen Protokolle umzusetzen.

Beim Aufbau der Verbindung wird die Art des Transports festgelegt. Es gibt die Möglichkeit einer Punkt-zu-Punkt-Verbindung (*peer to peer*), in der die Daten in der Reihenfolge ihres Eintreffens übertragen werden, ebenso wie die Paketvermittlung, bei der die Daten ins Netz gegeben werden und die Reihenfolge des Eintreffens beim Gegenüber nicht festgelegt ist. Die Übertragungsarten *Broadcast* bzw. *Multicast* dienen dazu, alle bzw. eine bestimmte Anzahl der angeschlossenen Stationen gleichzeitig mit denselben Nachrichten zu versorgen. Die Ebenen 1–4 bilden gemeinsam das Transportsystem im OSI-Referenzmodell.

1.2.1.6 Die Sitzungsschicht

Die Ebene 5 im OSI-Referenzmodell wird als Sitzungsschicht (*Session Layer*) bezeichnet. Unter einer Sitzung versteht man die Benutzung des Transportsystems, d. h. des fehlerfreien logischen Kanals, den die Transportschicht zur Verfügung stellt.

Dazu werden Dienste zum Aufbau und Abbau von Sitzungen bereitgestellt, so dass einer oder mehrere Prozesse auf das Transportsystem zugreifen können. Die Sitzungsschicht ist normalerweise mit dem Betriebssystem des Rechners verbunden. Sie synchronisiert, falls erforderlich, die kommunizierenden Prozesse, um einen korrekten Datenfluss zu ermöglichen.

Abhängig davon, welche Aktivitäten in den höheren Schichten ausgeführt werden sollen, kann unterschiedlicher Funktionsumfang in der Sitzungsschicht implementiert werden. Im OSI-Modell gibt es die Funktionsmengen:

BCS *Basic Combined Subset* für Verbindungssteuerung und Datenübertragung,

BAS *Basic Activity Subset* für Aktivitätsverwaltung und

BSS *Basic Synchronized Subset* zur Synchronisierung.

Die Sitzungsschicht kann symmetrische Partnerkonstellationen ebenso verwalten wie unsymmetrische Verbindungen (Client-Server-Architektur). Ein Prozedurauftrag auf einem fernen Rechner (*Remote Procedure Call*) wird von der Sitzungsschicht gesteuert.

1.2.1.7 Die Darstellungsschicht

Die Darstellungsschicht (*Presentation Layer*), Schicht 6, stellt Dienste zur Darstellung der übertragenen Daten zur Verfügung. Dies beinhaltet Funktionen

- zum verwendeten Zeichensatz,
- zur Codierung zu übertragender Daten und
- zur Darstellung der Daten auf Bildschirm oder Drucker.

Prozesse in einer Kommunikation tauschen Daten miteinander aus, die einer bestimmten Syntax unterworfen sind und einer festgelegten Semantik dienen. Innerhalb dieses Datenaustausches muss vereinbart werden, wie die Informationsdarstellung während der Nachrichtenübertragung sein soll und welche Art der Darstellung die beiden kommunizierenden Prozesse benutzen.

Übertragene Daten können z. B. in verschiedenen Kodierungen bei EBCDIC- oder ASCII-Terminals oder in unterschiedlichen Dateiformaten vorliegen.

Daher liegen die Aufgaben der Darstellungsschicht auch in der Ver- und Entschlüsselung der Daten (*Data Encryption*) und in der Wahrung der Datensicherheit (*Data Security & Privacy*).

Auch die Komprimierung der Daten zum Zwecke der Verkleinerung der Datenmenge und damit der Zeit- und Kostenersparnis wird von der Darstellungsschicht geleistet.

1.2.1.8 Die Anwendungsschicht

Die oberste Schicht des OSI-Referenzmodells ist Schicht 7, die Anwendungsschicht (*Application Layer*). Sie beinhaltet Funktionen, mit denen der Benutzer auf das Kommunikationssystem zugreifen kann. Der Benutzer ist hierbei in aller Regel nicht der Mensch,

sondern ein Computerprogramm, wie z. B. FTAM (*File Transfer, Access and Management*), ein Programm für Dateiübertragung und Dateizugriff über Rechnergrenzen hinweg.

Die Anwendungsschicht hat Ortstransparenz zu gewährleisten, beispielsweise bei verteilten Datenbanken, wo logisch zusammengehörende Daten physisch auf verschiedenen Rechnern an geographisch unterschiedlichen Orten abgelegt sind. Bei Abfrage über ein Netz darf der Benutzer nichts von den physischen Eigenschaften der Datenbank merken.

1.2.1.9 Dienste für die Kommunikation zwischen den Schichten

Jede Instanz einer OSI-Schicht bietet der darüberliegenden Schicht ihre Dienste an. Beim Datenaustausch zwischen der Schicht N und der Schicht $N + 1$ stellt die Schicht N die erforderlichen Dienste zur Verfügung, sie ist der *service provider*. Schicht $N + 1$ benutzt diese Dienste und ist damit der *service user*.

Die Dienste sind an ausgezeichneten Zugangspunkten verfügbar, den so genannten *service access points* (SAP). Jeder SAP hat eine eindeutige Adresse.

Die Dienste werden in verbindungsunabhängige und verbindungsorientierte Dienste unterschieden (siehe oben).

Für die Abhandlung der Dienstaufgaben stehen Dienstprimitive zur Verfügung. Es sind:

- die Anforderung (*request*),
- die Indikation (*indication*),
- die Antwort (*response*) und
- die Bestätigung (*confirmation*).

Bei bestätigten Diensten sind alle vier Dienstprimitive vorhanden, bei unbestätigten Diensten nur die Anforderung und die Indikation.

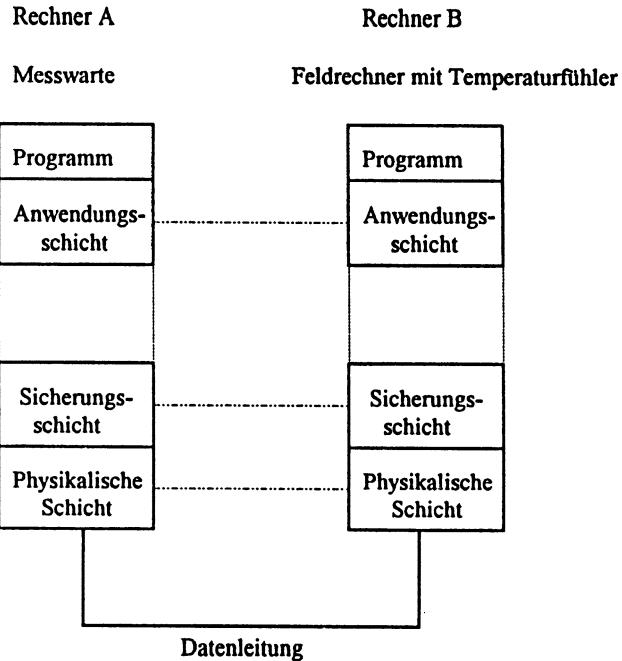
Ein typischer bestätigter Dienst ist der Verbindungsaufbau (*connect*). In der OSI-Notation wird jeweils der Dienst angegeben und durch einen Punkt getrennt die jeweilige Primitive. Bei einem Verbindungsaufbau zwischen zwei Stationen wird zunächst von demjenigen Partner, der die Verbindung eröffnen möchte, ein *connect.request* gesendet. Dies löst beim empfangenden Partner eine *connect.indication* aus, d. h. die Anzeige, dass eine Verbindung aufgebaut werden soll. Der angesprochene Partner antwortet *connect.response* ob er die Verbindung aufnehmen will. Diese Antwort löst beim Initiator der Verbindung die Bestätigung *connect.confirmation* aus.

Weitere Dienste sind die Datenübergabe (*data*) und der Verbindungsabbau (*disconnect*). Sie können ebenfalls als bestätigte oder unbestätigte Dienste eingesetzt werden.

1.2.1.10 Beispiel: Ablauf einer Kommunikation im OSI-Modell

Kommunizieren zwei Rechner A und B im OSI-Referenzmodell, so kommunizieren jeweils gleiche Schichten innerhalb der Kommunikationspartner miteinander. Die physikalische Schicht von Station A steht in direkter Verbindung zur physikalischen Schicht von Station B. Die Netzwerkschichten der beiden Stationen tauschen miteinander Daten aus,

Abb. 1.10 Beispiel einer Kommunikation im OSI-Modell



wobei die darunterliegenden Schichten vollkommen transparent sind für den Datenaustausch von Schicht 3 zu Schicht 3.

Es sind immer alle Schichten des Referenzmodells vorhanden, allerdings können die Schichten auch leer sein, sodass bei einer Kommunikation z. B. nur für die Schichten 1, 2 und 7 Programmcode vorhanden ist, während die Schichten 3 bis 6 leer sind (z. B. Profibus-Definition).

Die Schichten tauschen Daten untereinander über festgelegte, bestätigte oder unbestätigte Dienste aus, wie im vorangegangenen Kapitel beschrieben.

Der Ablauf einer Kommunikation sei an einem Beispiel dargestellt (Abb. 1.10). Der Anwender der Station A, im Beispiel ein Programm zur Verarbeitung von Messwerten, möchte einen neuen Messwert bei einer Außenstation, Station B, abfragen. Die Verbindung zwischen den Stationen A und B ist bereits aufgebaut.

Das Anwendungsprogramm weist als erstes die Anwendungsschicht an, einen Messwert von der Station B zu verlangen. Die Anwendungsschicht bearbeitet diese Direktive und leitet sie an die darunterliegenden Schichten als Daten weiter. Die Darstellungsschicht wandelt die Daten in die für die Übertragung vereinbarte Form und reicht sie an die Sitzungsschicht, welche die Information hinzufügt, aus welcher Sitzung die Anforderung stammt.

Die Daten werden in den Transportkanal gegeben und in der Netzwerkschicht auf den richtigen Weg gelenkt (Routing). Die Sicherungsschicht fügt den Daten Sicherungsinfo-

mation hinzu und gibt das nun in der Länge vergrößerte Telegramm an die physikalische Schicht zur Bitübertragung weiter.

Über die Datenleitung gelangt das Telegramm zur Station B, wo es in der physikalischen Schicht empfangen wird. Von dort wird es an die Sicherungsschicht weitergereicht. Sie überprüft die Korrektheit der Bitübertragung, entfernt die Sicherungsinformation und gibt die restlichen Daten an die nächsthöhere Schicht weiter. Die Anwendungsschicht erkennt aus den Daten des Telegramms die Aufforderung zum Lesen des Messwertes und reicht sie an das Programm zur Bearbeitung weiter.

Der gelesene Messwert nimmt analog zur obengenannten Vorgehensweise den Weg rückwärts durch alle Schichten, bis die Anwendungsschicht in Station A den gelesenen Messwert an das Programm abgeben kann.

1.2.2 Das TCP/IP-Protokoll

Ein allgemein anerkannter Standard für den Datenaustausch in heterogenen Netzen ist das *Transmission Control Protocol* (TCP) im Zusammenhang mit einer speziellen Ausführung der *Internet suite of protocols* (IP), abgekürzt als TCP/IP. Er entstand aus einer Auftragsentwicklung des amerikanischen Verteidigungsministeriums.

TCP/IP ist Teil einiger UNIX-Betriebssysteme (z. B. Berkeley-UNIX, ULTRIX), so dass ein Großteil der unter UNIX betriebenen Rechner ohne zusätzliche Software an TCP/IP-betriebene Netze angeschaltet werden können.

TCP/IP wird sowohl in lokalen Netzen zur Kommunikation verschiedenartiger Rechner untereinander als auch für den Zugang von LAN (*Local Area Networks*) zu WAN (*Wide Area Networks*) eingesetzt.

Betrachtet man das TCP/IP-Protokoll in der Denkweise des OSI-Modells, so ist das Internet-Protokoll (IP) die Netzwerkschicht. Im Regelfall wird ein TCP/IP-Netz auf einem Ethernet (OSI-Schicht 1) betrieben. Die Anbindung der Netzwerkschicht an die Sicherungsschicht (LLC) erfolgt nach der IEEE 802.2-Norm, die Zugriffskontrolle für das Medium (MAC) nach CSMA/CD (*Carrier Sensing Multiple Access/Collision Detection*, IEEE 802.3).

Die Transportschicht wird entweder von TCP (*Transmission Control Protocol*) als verbindungsorientiertem oder von UDP (*User Datagram Protocol*) als verbindungslosem Dienst gebildet. Die Einordnung von TCP/IP in OSI kann auf der theoretischen Ebene vorgenommen werden. Vom praktischen Standpunkt her ist jedoch zu erwähnen, dass TCP/IP-Protokolle älter als der OSI-Standard sind und daher nicht nach dem OSI-Standard programmiert wurden.

Ein Vergleich zwischen OSI-basierten Modellen, TCP/IP und anderen gebräuchlichen Nicht-OSI-Modellen wird in Abb. 1.11 gezeigt.

Das TCP/IP-Transportsystem wird von Applikationsprotokollen wie FIP (*File Transfer Protocol*), TELNET (*Telnet Protocol*) und SMTP (*Simple Mail Transfer Protocol, Electronic Mail*) zur Datenübertragung zwischen Endsystemen benutzt.

OSI	ISO 7498	TCP/IP	Novell	IBM	IBM	DEC
	CCITT X.200			NETBIOS	SNA	DECNET
Anwendungsschicht	FTAM: ISO 8571 ISO 8572 (File Transport Access and Management) JTM: ISO 8831 ISO 8832 (Job Transfer and Manipulation) VTP: ISO 8831 ISO 8832 (Virtual Terminal Protocol) CCITT X.400	Networkfile Server (NFS) Telnet File Transfer Protocol (FTP)	Anwendungsprogramm MS-DOS OS/2	Anwendungsprogramm	End User	Anwendungsprogramm
Darstellungsschicht	ISO 8822 ISO 8823			MS-DOS		
Sitzungsschicht	ISO 8326 ISO 8327		Shell	MS-NET	Data Flow Control	Sitzung
Transportschicht	ISO 8072 ISO 8073	TCP / UDP	Internetwork Packet Exchange (IPX)	NETBIOS	Transmission Control	Netzwerk und Transport
Netzwerkschicht	ISO 8473 CCITT X.25 (Schicht 3)	IP			Path Control	
Sicherungsschicht	CCITT X.25 (LAPB)		IEEE 802.2 / ISO 8802 ----- CSMA/CD (IEEE 802.3)			
Physische Schicht			Ethernet / CCITT X.21			

Abb. 1.11 Einordnung von TCP/IP in das OSI-Schichtenmodell und Vergleich mit anderen Kommunikationsmodellen

Das TCP-Protokoll teilt als verbindungsorientiertes Protokoll die zu übertragenden Daten in Datenblöcke. Beim Start der Übertragung wird die maximale Blockgröße zwischen Sender und Empfänger ausgetauscht.

Von der Netzwerkschicht IP werden diese Datenblöcke mittels eines Datagrammservice versendet, sodass die Reihenfolge der Blöcke im Empfänger durch TCP wiederhergestellt werden muss. Jedem Datenblock geht ein Header voraus, der die Adressen von Quelle und Ziel, die Sequenznummer, Steuerinformationen und eine Checksumme enthält. Die maximale Länge jedes Datagramms beträgt 64 kByte.

Durch die Vergabe von Portnummern für jeden Übertragungsprozess können mehrere Prozesse parallel über ein TCP-Modul auf das Netz zugreifen, ohne dass Daten vertauscht werden. Die Verbindung zweier Prozesse in verschiedenen Rechnern wird am Netz über den *Socket* identifiziert, einer Kombination aus der Internet-Adresse und der Portnummer. Der Kommunikationspartner muss empfangene Telegramme quittieren. Aus Gründen des Datendurchsatzes werden aber mehrere Telegramme ins Netz gesendet, bevor die Ankunft des ersten Telegrammes bestätigt ist. Um bei unterschiedlichen Schreib-/Lesegeschwindigkeiten der Teilnehmer den Datenfluss zu kontrollieren, wird die Anzahl der maximal im Netz verschickten unquittierten Telegramme, die ein Rechner bearbeiten kann, im Header als Fenstergröße mit angegeben.

Die fehlerfreie Übertragung von Telegrammen wird bei Verbindungsauftbau, Datenverkehr und Verbindungsabbau über ein Handshake-Verfahren mit Timeout-Überwachung sichergestellt.

1.3 Buszugriffsverfahren

Fast alle im Abschn. 1.1 besprochenen Topologien setzen voraus, dass zu einem bestimmten Zeitpunkt nur ein Sender auf das gemeinsame Trägermedium zugreift. Im Folgenden sollen verschiedene Möglichkeiten, diesen Buszugriff zu regeln, vorgestellt werden. Dabei unterscheidet man, wie in Abb. 1.12 dargestellt, zwischen kontrollierten und zufälligen Buszugriffsverfahren.

Bei den kontrollierten Buszugriffsverfahren ist der Sender vor dem Sendebeginn eindeutig bestimmt. Damit ist eine Buszuteilung für den jeweiligen Sender notwendig. Diese kann zentral von einer Leitstation (*Master/Slave-Verfahren*) oder dezentral durch mehrere Steuereinheiten (*Tokenbus*, *Tokenring*) vorgenommen werden. Wird der Zeitraum oder die Datenlänge für einen Kommunikationszyklus begrenzt, ist die maximale Zeitspanne, bis die Daten übertragen sind, berechenbar (*Zykluszeit*). Solche Systeme nennt man echtzeitfähig.

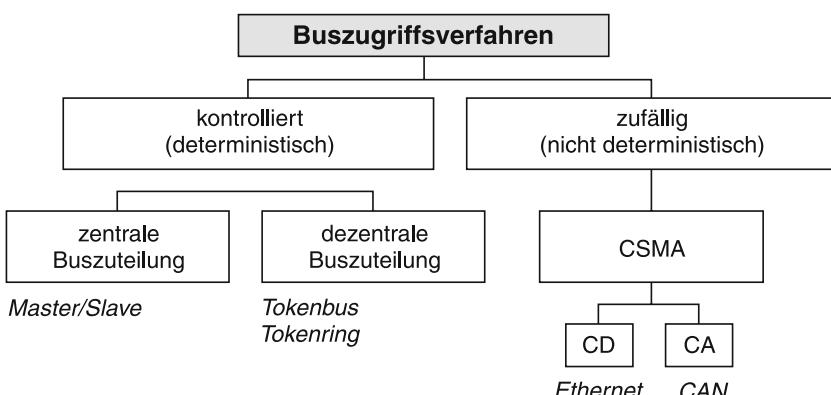


Abb. 1.12 Übersicht Buszugriffsverfahren

Bei den zufälligen Buszugriffsverfahren greifen die sendewilligen Teilnehmer nur bei Bedarf auf das Übertragungsmedium zu (Carrier Sense, CS). Dabei muss gewährleistet sein, dass das Medium nicht anderweitig von einem anderen Teilnehmer belegt ist. Ist dies der Fall, muss die Sendung auf einen späteren Zeitpunkt verschoben werden (Multiple Access, MA). Damit ist eine Bestimmung des maximalen Zeitraumes, in dem eine Information übertragen wird, nicht mehr möglich. Damit sind zufällige Buszugriffsverfahren i. d. R. nicht echtzeitfähig.

1.3.1 Master/Slave-Verfahren

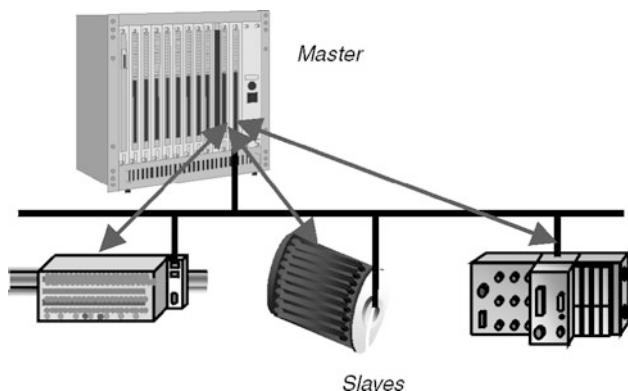
Bei diesem Verfahren stellt die Bussteuereinheit, der so genannte Master, die Verbindung zum passiven Teilnehmer, dem *Slave*, her. Dieser antwortet auf eine Datenanforderung des Masters unmittelbar (*immediate response*) (Abb. 1.13).

Der Master stellt in der Regel zyklisch die Verbindung zu jedem Slave her (*polling*). Damit ist im Master immer ein aktuelles Abbild des zu steuernden Prozesses abgelegt. Prioritäten können dadurch vergeben werden, dass manche Slaves mehrmals innerhalb eines Zyklus abgefragt werden.

Dieses Verfahren hat den Vorteil, dass die Busanschaltung der Slaves sehr einfach und damit kostengünstig ist, da die gesamte benötigte Intelligenz im Master implementiert ist. Problematisch können die Zeiten sein, die benötigt werden, wenn Daten zwischen zwei Slaves ausgetauscht werden müssen. In diesem Fall schickt der Master eine Datenanforderung an den signalgebenden Slave, woraufhin dieser unmittelbar antwortet. Diese Information muss im Master bzw. in der Steuerung verarbeitet und an den empfangenden Slave übertragen werden.

Das bedeutet, dass im Extremfall die Zykluszeit sowohl für die Datenanforderung als auch für die Sendung benötigt wird. Zusätzlich muss die Bearbeitungszeit durch den Master/die Steuerung berücksichtigt werden. Damit liegt die Zeitspanne dieser Datenübertragung unter Umständen um ein Vielfaches über der Zykluszeit.

Abb. 1.13 Master/Slave-Verfahren



Das Problem soll durch folgendes Beispiel verdeutlicht werden: Ein Füllstandssensor erfasst den maximalen Füllstand eines Behältnisses. Unter „Worst-case“-Betrachtung kann es passieren, dass der Master den Slave abfragt, dieser die Nachricht überträgt, dass der maximale Füllstand nicht erreicht ist und genau zu diesem Zeitpunkt sich der Zustand am Sensor ändert. Das bedeutet, dass die Information erst im folgenden Buszyklus übertragen wird. Damit wird im Extremfall eine Buszykluszeit benötigt um Zustandsänderungen im Speicher des Masters zu hinterlegen. In diesem Beispiel sei der Master in einer SPS integriert. Es kann nun passieren, dass zu dem Zeitpunkt, wenn die Zustandsänderung den Master erreicht gerade ein neuer SPS-Zyklus gestartet wurde und das Prozessabbild der Eingänge gerade gelesen wurde. In diesem Fall wird die Information erst im darauf folgenden SPS-Zyklus verarbeitet. Dies bedeutet konkret, dass unter Umständen 2 SPS-Zyklen benötigt werden, um die Nachricht, dass das entsprechende Ventil über welches das Behältnis gefüllt wird, zu schließen ist im Master zu hinterlegen. Anschließend wird ein weiterer Buszyklus benötigt um die Information an das Ventil zu übertragen.

Das eben beschriebene Beispiel zeigt den Unterschied zwischen Zyklus- und Reaktionszeit. Allgemein gilt folgender Zusammenhang:

$$\text{Reaktionszeit}_{\max} = 2 \times \text{Buszykluszeit} + 2 \times \text{SPS-Zykluszeit} .$$

In dieser Gleichung sind z. B. die mechanischen Verzögerungszeiten eines Ventils nicht berücksichtigt und damit noch hinzuzuzählen.

Dieses Buszugriffsverfahren wird beispielsweise bei AS-Interface und bei PROFIBUS DP verwendet. Beim letzteren System gilt dies nur, wenn es sich um ein so genanntes Monomastersystem, d. h. ein System mit nur einem Master, handelt.

In Abb. 1.14 ist ein Verfahren dargestellt, mit dem die Kommunikation zwischen zwei Slaves beschleunigt werden kann. Der Master überträgt an Slave 1 den Befehl „Empfange Daten“. Slave 2 erhält vom Master den Befehl „Sende Daten“, woraufhin dieser mit der Datenübertragung beginnt. Empfängt Slave 1 die Daten inklusive einer „Endmeldung“ korrekt, sendet er wiederum eine „Endmeldung“ an den Master. Dies erfordert von den Slaves eine etwas höhere Intelligenz, was sich direkt auf den Preis auswirkt.

Ein großer Nachteil des Master/Slave-Verfahrens besteht darin, dass bei einem Ausfall des Masters das gesamte Bussystem stillliegt. Auch hier besteht die Möglichkeit, den Master durch einen Slave überwachen zu lassen. Dieser Slave muss damit sämtliche Aufgaben des Masters übernehmen können. Dazu gehören neben der Kommunikationssteuerung die Ausfallüberwachung der Slaves, die Überwachung der Übertragungsqualität und die Fehlerbehandlung.

Mit der Ausfallüberwachung für Slaves soll die Möglichkeit gegeben werden, dass ein defekter Slave inaktiv gesetzt werden kann und aus der Polliste entfernt werden kann. Gleichzeitig muss gewährleistet sein, dass dieser Slave wieder in die Polliste aufgenommen wird, wenn er wieder funktionsfähig ist.

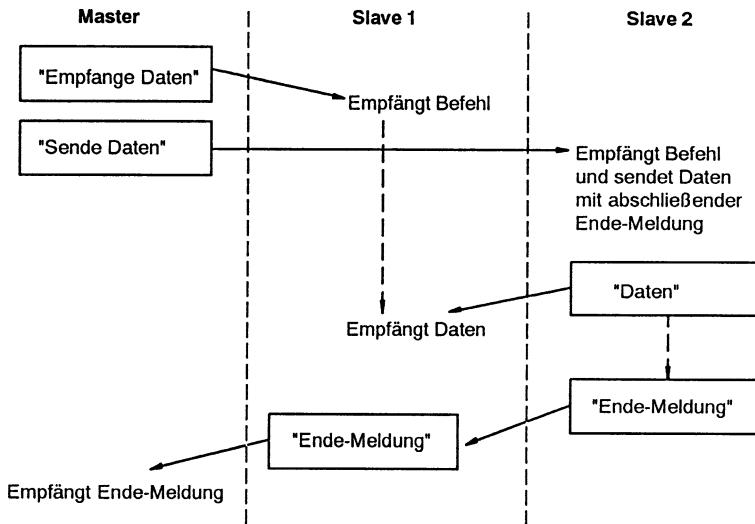


Abb. 1.14 Beschleunigter Datenaustausch zwischen Slave-Stationen

1.3.2 Token-Prinzip

Bei Systemen, die nach dem Token-Prinzip arbeiten, sind alle Teilnehmer in der Lage, die Kommunikationssteuerung zu übernehmen. Die Berechtigung, auf den Bus zugreifen zu dürfen und damit die Kommunikationssteuerung zu übernehmen, wird durch ein spezielles Zeichen oder eine spezielle Nachricht, dem so genannten *Token*, genau einem der Teilnehmer zugeteilt. Hat dieser seine Datenübertragungen abgeschlossen, wird das Token an den nächsten Teilnehmer weitergereicht. Da die Zeitdauer des Token-Besitzes zeitlich limitiert ist, sind auch diese Zugriffsverfahren echtzeitfähig.

Abb. 1.15a zeigt den Token-Bus. Dieser ist in der IEEE-Norm 802.4 spezifiziert. Hier sind alle Teilnehmer an ein gemeinsames Buskabel angeschlossen (Linientopologie). Das Token stellt hier eine spezielle kurze Nachricht dar und wird über die Datenleitung von Teilnehmer zu Teilnehmer weitergereicht. Eine Möglichkeit, dies zu realisieren, ist, das Token an den Teilnehmer mit der nächstniedrigeren Adresse weiterzureichen. Der Teilnehmer mit der niedrigsten Adresse übergibt das Token an den Teilnehmer mit der höchsten Adresse. Daraus entsteht ein logischer Ring.

Prioritäten können einmal durch unterschiedliche maximale Datenmengen oder durch Mehrfachzuteilung des Tokens innerhalb eines Zyklus realisiert werden.

Ein Teilnehmer im logischen Ring muss in der Lage sein, die folgenden Überwachungsfunktionen durchzuführen:

- Überwachung des Tokens: Sollte aufgrund eines Fehlers kein Token oder mehrere Token vorhanden sein, müssen die alten Token gelöscht und ein neues Token erzeugt werden.

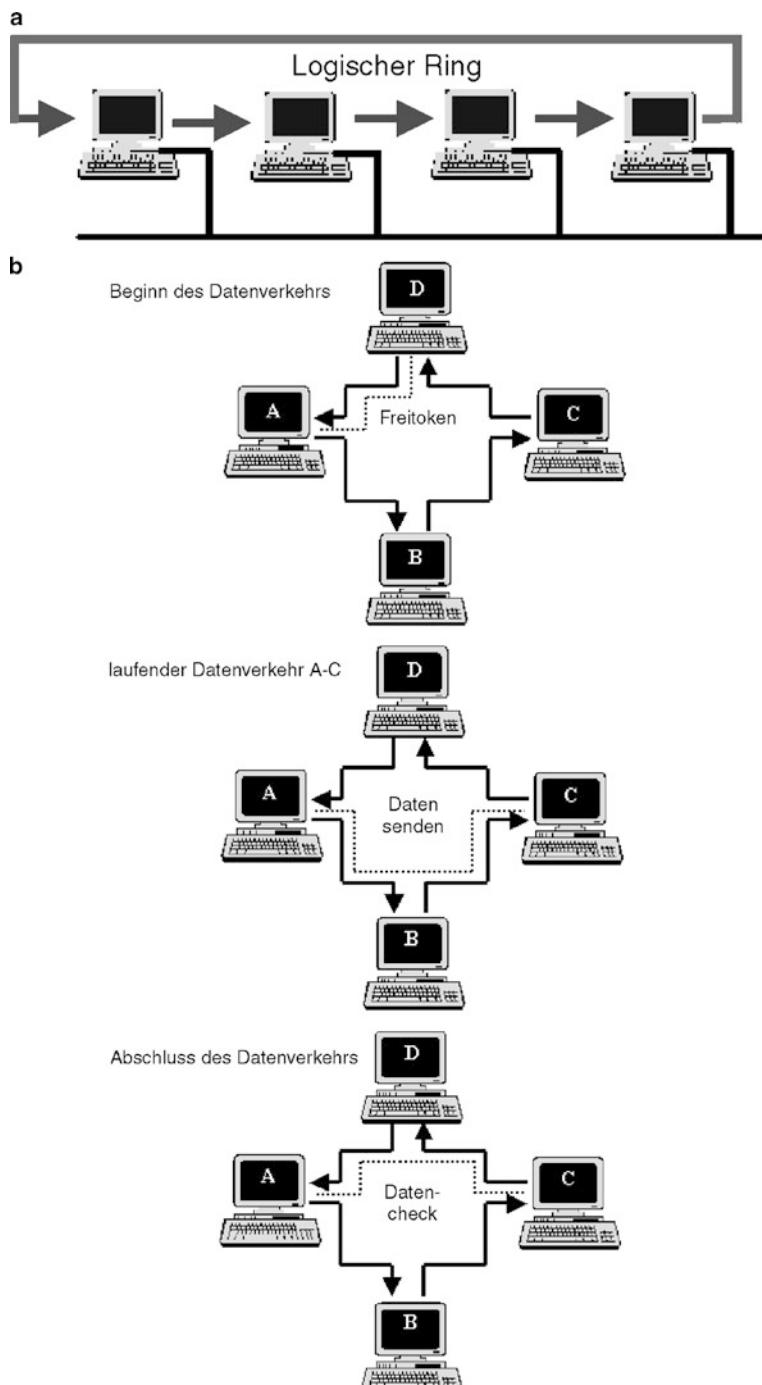


Abb. 1.15 a Token-Bus, b Token-Ring

- Initialisierung nach dem Einschalten: z. B. muss ein definierter Teilnehmer das Token erzeugen.
- Entfernung von Stationen aus dem logischen Ring: Defekte Stationen müssen aus dem logischen Ring entfernt werden können.
- Aufnahme hinzugekommener Stationen in den logischen Ring: Es muss die Möglichkeit bestehen, vorher entfernte Stationen wieder in den logischen Ring aufzunehmen bzw. den logischen Ring zu erweitern.

Das reine Token-Prinzip wird z. B. beim ModbusPlus eingesetzt.

Die Norm IEEE 802.5 beschreibt den Token-Ring, der in Abb. 1.15b dargestellt ist.

Hat eine Station ihre Datenübertragung abgeschlossen, überträgt sie das so genannte Frei-Token zum nächsten Teilnehmer im physikalischen Ring. Dieser wandelt das Frei-Token in ein Belegt-Token um (Änderung von 1 Bit), wenn er Daten zu übertragen hat. Ist dies nicht der Fall, wird das Frei-Token unmittelbar an den nächsten Teilnehmer weitergereicht. Die zu übertragenden Daten werden hinter das Belegt-Token angefügt und zum nächsten Teilnehmer übertragen. Sind die Daten nicht für den nächsten Teilnehmer bestimmt, werden sie unmittelbar weitergeleitet. Haben die Daten den Empfänger erreicht, werden diese in seinen Speicherbereich kopiert. Der Empfänger sendet die empfangenen Daten zum nächsten Teilnehmer, bis diese zum Sender zurückgelangen. Der Sender wandelt das Belegt-Token in ein Frei-Token um, wenn er seine eigene Sendung fehlerfrei empfangen hat und überträgt dieses an die nächste Station.

Problematisch ist dieses System, wenn ein Teilnehmer ausfällt oder es zu einem Leitungsbruch bzw. Kurzschluss kommt. Eine Möglichkeit, dieses Problem in den Griff zu bekommen, liefert die so genannte Doppelringstruktur. Dabei wird der Ring redundant ausgelegt, wobei jeder Teilnehmer jetzt die Möglichkeit hat, sowohl mit seinem Nachfolger als auch mit seinem Vorgänger zu kommunizieren. Diese Kommunikationsmöglichkeiten können dann zyklisch überprüft werden. Damit kann die defekte Stelle lokalisiert und aus dem Ring ausgeschlossen werden.

1.3.3 Token-Passing

Bei dem *Token-Passing*, auch hybrides Zugriffsverfahren genannt, handelt es sich um eine Kombination aus Token-Bus und Master/Slave-Verfahren (Abb. 1.16).

Hierbei befinden sich sowohl aktive Teilnehmer, die die Kommunikationssteuerung des Busses übernehmen können, als auch passive Teilnehmer, die dazu nicht in der Lage sind, an einem Busstrang. Das Token wird unter den aktiven Teilnehmern in einem logischen Ring weitergereicht. Der jeweilige Token-Inhaber kann mit den passiven Teilnehmern im Master/Slave-Verfahren und mit den aktiven Teilnehmern nach dem Token-Prinzip kommunizieren.

Das bedeutet, dass der Token-Inhaber unterscheiden muss, ob er mit einem aktiven oder mit einem passiven Teilnehmer kommuniziert. Diese Unterscheidung muss in der

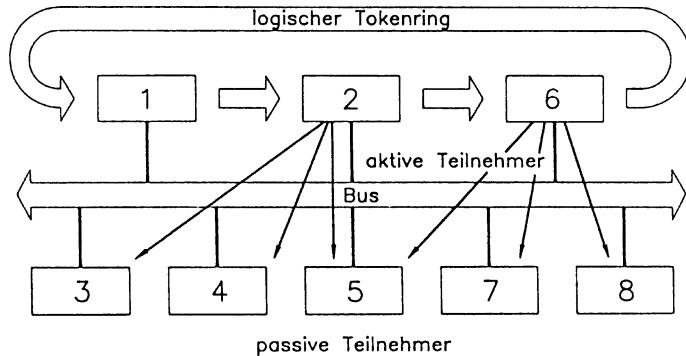


Abb. 1.16 Token-Passing

Projektierungsphase für jeden aktiven Teilnehmer festgelegt werden. Beim PROFIBUS, dessen Kommunikationsmodell auf dem ISO/OSI-7-Schichtenmodell beruht, geschieht dies in der Schicht 2 und ist damit für den Anwender nicht sichtbar.

Dieses Verfahren hat den Vorteil, dass es flexibel ist, was die Anschaltbaugruppen betrifft. Jedoch wirkt sich der höhere Projektierungsaufwand nachteilig aus. Das folgende Beispiel soll zeigen, wo der Vorteil dieses Verfahrens liegt:

An einen Bus sind als aktive Teilnehmer eine SPS und ein PC angeschlossen. Die Aufgabe der SPS besteht darin, zyklisch die Ein- und Ausgangsdaten mit den Feldgeräten (passiven Teilnehmern) auszutauschen. Dafür wird eine ausreichend lange Tokenhaltezeit projektiert. Unter einer Tokenhaltezeit versteht man die Zeitspanne, die der aktive Teilnehmer das Token behalten darf.

Nachdem die Tokenhaltezeit abgelaufen ist, gibt die SPS das Token an den PC. Dessen Aufgabe besteht darin, eine Prozessvisualisierung vorzunehmen, zum anderen soll die Möglichkeit der Umparametrierung im laufenden Betrieb gegeben sein. Dazu hat er die Möglichkeit, alle Ein- und Ausgangsdaten der Feldgeräte zu lesen und azyklisch Parameterwerte an die Teilnehmer zu schicken. Die Tokenhaltezeit des PCs wird üblicherweise kürzer sein als die Tokenhaltezeit der SPS. Kann der PC seine Aufgaben innerhalb der Tokenhaltezeit nicht komplett abarbeiten, merkt er sich die nicht durchgeföhrten Aufgaben und arbeitet diese ab, wenn er das nächste Mal das Token erhält.

Dies ist ein typisches Verfahren, welches der PROFIBUS-DP V1 auch in Verbindung mit dem PROFIBUS PA in der Prozessautomatisierung verwendet. Dort ist der PC die Anzeige- und Bedienkomponente (ABK), von der der Prozess gesteuert und überwacht werden kann.

1.3.4 CSMA

Das Kürzel CSMA steht für *Carrier Sense Multiple Access*. Dabei hört ein sendewilliger Teilnehmer die gemeinsame Busleitung ab (*Carrier Sense*) und sendet, falls diese nicht belegt ist. Sollte die Busleitung durch einen anderen Teilnehmer belegt sein, stellt der

sendewillige Teilnehmer seinen Sendewunsch zurück und versucht, zu einem späteren Zeitpunkt erneut die Daten zu übertragen (*Multiple Access*). Da ein Teilnehmer nur dann auf das Trägermedium zugreift, wenn er Daten übertragen will, kann im Voraus nicht bestimmt werden, welcher Teilnehmer sendet. Damit handelt es sich um ein zufälliges Buszugriffsverfahren. Da auch nicht sichergestellt ist, dass die Busleitung frei ist, wenn ein Sendewunsch besteht, kann auch keine maximale Zeit garantiert werden, innerhalb derer die Daten übertragen werden. Damit ist dieses Verfahren nicht echtzeitfähig.

Es existieren zwei Varianten, den erneuten Buszugriff nach einem gescheiterten Versuch zu regeln:

Bei Variante 1 zieht sich die sendewillige Station für eine zufällig gewählte Zeit zurück und versucht dann den erneuten Zugriff. Ist die Busleitung immer noch oder schon wieder belegt, zieht sich der Teilnehmer erneut für eine zufällig gewählte Zeit zurück, die aber länger als die vorherige Periode ist. Hierbei tritt das Problem auf, dass unter Umständen ein sendewilliger Teilnehmer sehr lange warten muss, bis er ein freies Trägermedium findet. Das Problem wird umso größer, je stärker das Bussystem ausgelastet ist. Durch die zufällig gewählten Wartezeiten kann es zu Zeiträumen kommen, in denen das Trägermedium ungenutzt bleibt. Diese Variante wird z. B. von dem in der Bürokommunikation eingesetzten Ethernet verwendet.

Bei Variante 2, die in der IEEE Norm 802.3 beschrieben ist, hört ein sendewilliger Teilnehmer das Trägermedium ständig ab und sendet sofort, nachdem die laufende Kommunikation abgeschlossen ist. Dadurch entstehen keine Wartezeiten. Hierbei kann es jedoch passieren, dass 2 Sender gleichzeitig mit der Sendung beginnen, da sie beide während der vorherigen Sendung versucht haben, auf den Bus zuzugreifen. Dies hat zur Folge, dass die Sendungen kollidieren und sich gegenseitig zerstören. Ohne Zusatzmaßnahmen würde dies erst erkannt werden, wenn der Empfänger die übertragenen Daten auf Fehler überprüft. Damit ist der Bus für die gesamte Zeit der sich überlagernden Übertragungen belegt und kann nicht anderweitig genutzt werden, die Effizienz sinkt.

Dieses Problem tritt auch bei Variante 1 auf, wenn zwei Teilnehmer quasi gleichzeitig den Bus abhören, ihn für frei befinden und mit der Sendung beginnen. Auch hier gilt, dass mit steigender Busauslastung die Wahrscheinlichkeit einer Kollision steigt. Die Bedeutung von quasi gleichzeitig soll anhand der Abb. 1.17 erläutert werden: Teilnehmer 1 beginnt zum Zeitpunkt t mit seiner Sendung. Das Signal benötigt die Signallaufzeit t_S , bis es beim Teilnehmer n ankommt. Hört der Teilnehmer n die Busleitung im Zeitraum von t bis $t + t_S$ ab, befindet er ihn für frei und beginnt ebenfalls mit der Sendung, und es kommt zur Kollision. Im Extremfall ist dann der Zeitpunkt $t + t_S$ erreicht.

Um die Effizienz zu steigern, empfangen die Teilnehmer die Signale auf der Busleitung, während sie senden. Unterscheiden sich die gesendeten und empfangenen Daten voneinander, ist es zur Kollision gekommen und die Übertragung wird sofort eingestellt. Dieses Verfahren nennt man *Collision Detection*, oder kurz CSMA/CD. Dies bedeutet am Beispiel der Abb. 1.17, dass der Sender n die Kollision nach $t + t_S$ erkennt, jedoch der Sender 1 erst zum Zeitpunkt $t + 2 \cdot t_S$. Daraus lässt sich ableiten, dass die minimale Sendedauer eines Pakets $2 \cdot t_S$ sein muss, um eine sichere Kollisionserkennung zu gewährleisten. In der

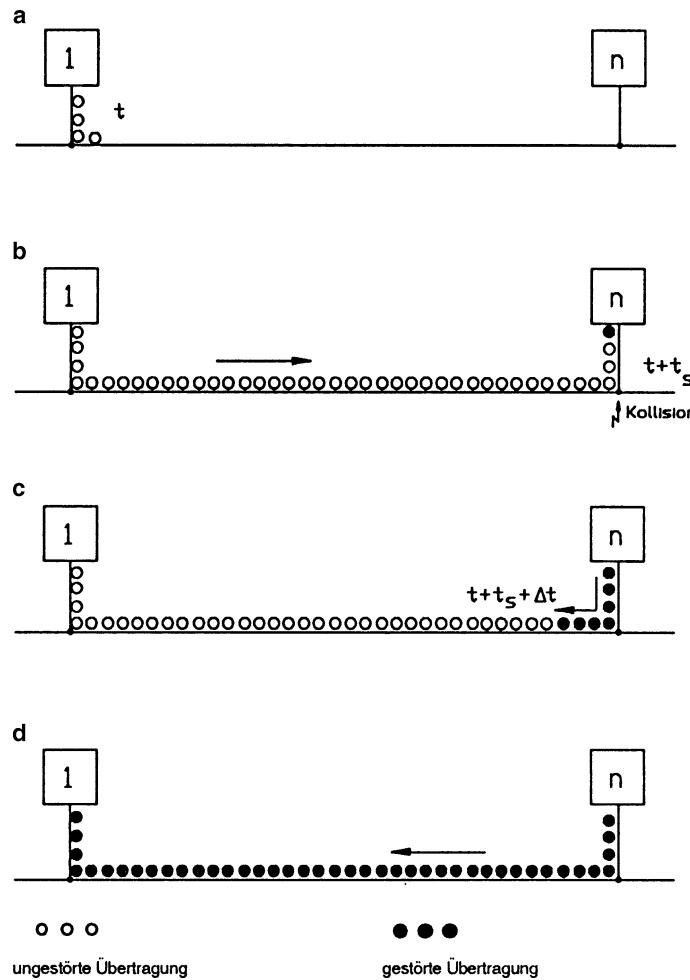


Abb. 1.17 Kollisionserkennung bei CSMA/CD. **a** Sendebeginn zum Zeitpunkt t , **b** Kollision zum Zeitpunkt $t + t_s$ mit $t_s = \text{Signallaufzeit}$, **c** Zustand zum Zeitpunkt $t + t_s + \Delta t$, **d** Zustand zum Zeitpunkt $t + 2t_s$

Praxis bedeutet dies, dass die minimale Paketlänge sowohl von der Datenübertragungsrate (ÜR) als auch von der Leitungslänge abhängt.

Beispiel

Bei $l = 1000 \text{ m}$ ergibt sich eine Signallaufzeit $t_s = 5 \mu\text{s}$ ($v = 0,66 \cdot c$).

Wird mit $\text{ÜR} = 1 \text{ MBd}$ gearbeitet, muss die Information mindestens 10 Bit lang sein, damit eine Kollision sicher erkannt werden kann.

Nach der Kollisionserkennung überträgt der Sender ein kurzes Störsignal (*jam*), mit dem er alle anderen Teilnehmer über die erkannte Kollision informiert. Alle sendewilligen Teilnehmer stellen dann ihre Sendung für eine zufällige Zeitdauer, die einem ganzzahligen Vielfachen der maximalen doppelten Signallaufzeit entspricht, zurück und versuchen dann erneut den Zugriff.

Man kann die beiden Verfahren kombinieren. Dabei würde ein sendewilliger Teilnehmer die Busleitung abhören und, wenn das Medium belegt ist, warten, bis es frei ist und dann mit der Wahrscheinlichkeit p senden. Mit $1 - p$ zieht sich der sendewillige Teilnehmer eine zufällig gewählte Zeit zurück. Dadurch wäre die Gefahr der Kollision geringer als bei Variante 2, jedoch die Zeiten, in denen die Leitung ungenutzt ist, größer.

1.3.5 CSMA/CA

Bei diesem Verfahren hört ein sendewilliger Teilnehmer das Trägermedium wie bei CS-MA/CD ab und beginnt die Übertragung, wenn das Medium frei ist. Ist es belegt, wird die laufende Übertragung abgewartet und unmittelbar im Anschluss daran mit der Sendung begonnen, wobei die Sendung ständig überwacht wird. Sollten zwei Teilnehmer gleichzeitig mit der Sendung beginnen, sind Prioritäten vergeben, so dass sich der Teilnehmer mit der niedrigeren Priorität zurückzieht, d. h. seine Übertragung abbricht, da er nicht mehr seine eigene Sendung empfangen hat. Damit wird eine Kollision vermieden (*Collision Avoidance, CA*).

Ein Telegramm beginnt immer mit der Kennzeichnung des Übertragungsbegins. Diese Information ist für alle sendewilligen Teilnehmer gleich, sodass sich zwei gleichzeitig auf der Übertragungsleitung befindende Sendungen nicht gegenseitig beeinflussen. Es folgt die Kennung des Senders, der so genannte *Identifier*. Definiert man einen logischen Zustand als dominant, z. B. „0“, den anderen als rezessiv, dominiert der Teilnehmer mit der niedrigeren Kennung. Der Teilnehmer mit der höheren Kennung bricht die Sendung ab und versucht, seine Daten im Anschluss an die jetzt laufende Übertragung zu senden.

Voraussetzung für die Funktionsfähigkeit ist hier, dass die Signallaufzeit t_S vernachlässigbar klein gegenüber der Bitzeit t_B ist:

$$t_S = \frac{l}{v} \ll t_B = \frac{1}{UR}$$

l – Leitungslänge

v – Ausbreitungsgeschwindigkeit (siehe Abschn. 1.8.2)

Diese Verfahren ist von seiner Grundkonzeption her nicht deterministisch. Jedoch kann durch entsprechende Software erreicht werden, dass sich ein CSMA/CA basiertes System deterministisch verhält. Ein Beispiel dafür ist das DeviceNet, das auf CAN basiert. CAN nutzt das CSMA/CA Verfahren; DeviceNet setzt den CAN Chip als Kommunikations-

chip ein. Bei DeviceNet kann ein so genannter Poll-I/O-Modus gewählt werden, der eine maximale, berechenbare Zykluszeit garantiert.

1.3.6 Busarbitration

Eine weitere Möglichkeit die Kommunikation auf der Übertragungsleitung zu steuern stellt die Busarbitration dar. Diese wird in unterschiedlichen Varianten z. B. vom FOUNDATION Fieldbus und von WorldFIP verwendet (Abb. 1.18).

Während der Projektionsphase wird festgelegt, welcher Teilnehmer ein bestimmtes Objekt erzeugt (Publisher). Für ein bestimmtes Objekt gibt es genau einen Publisher. Weiterhin wird projektiert, welche Teilnehmer dieses Objekt zu empfangen und zu verarbeiten haben (Subscriber). Es können für ein Objekt mehr als ein Subscriber definiert werden. Die aktive Einheit am Bus, der so genannte Link Active Scheduler, kurz LAS, ruft in definierbaren Zeitabständen das zu übertragende Objekt auf („Compel Data“ CD). Da dieses Telegramm von allen Teilnehmern empfangen wird, wissen die Subscriber, dass das nächste übertragene Telegramm das benötigte Objekt enthält. Der Publisher reagiert auf die Aufforderung mit der unmittelbaren Übertragung des geforderten Objektes.

Der Vorteil dieses Verfahrens besteht darin, dass bei Synchronisationsproblemen kein unnötiger Zeitverzug entsteht. Dies soll am Beispiel der Synchronisation von Antrieben verdeutlicht werden:

In einer Applikation müssen drei Antriebe synchron zueinander starten. Ausgelöst wird dies durch einen Sensor. Wird ein Master/Slave System verwendet, erhält die Steuerung die Information, dass die Antriebe zu starten sind im Zyklus n. Würden jetzt im darauf

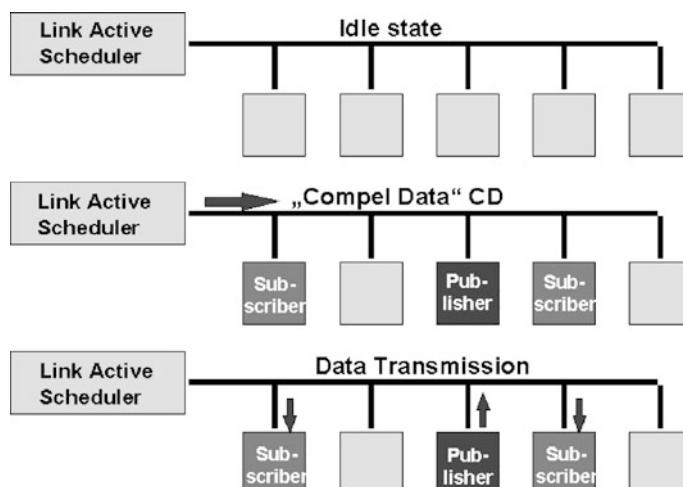


Abb. 1.18 Busarbitration bei FF

folgenden Zyklus die Antriebe aktiviert werden gäbe es einen Zeitverzug, da der Master die Slaves nur nacheinander ansprechen kann. Dieses Problem wird z. B. bei PROFIBUS wie folgt gelöst. Der Master schickt einen „SYNC“-Befehl an die Slaves. Dieser Befehl bedeutet, dass im nächsten Zyklus (Zyklus $n + 1$) Daten kommen, die erst zu verarbeiten sind, wenn dieser Befehl zurückgesetzt wird („UNSYNC“). Bei diesem Befehl handelt es sich um eine so genannte Broadcast-Message, bei der mit einem Telegramm eine Gruppe von Slaves angesprochen werden kann. Im darauf folgenden Zyklus ($n + 2$) werden die Daten an die Antriebe übertragen. Anschließend (Zyklus $n + 3$) wird der UNSYNC-Befehl übertragen, (auch das ist eine Broadcast-Message) und die Antriebe fangen synchron zueinander an zu arbeiten. Das bedeutet, dass mindestens 4 Zyklen benötigt werden, um eine Reaktion der Antriebe auf den aktuellen Prozesszustand zu erreichen.

Beim Busarbitrationsverfahren läuft dies wie folgt: Während der Projektierung wird festgelegt, dass die Antriebe die Information vom Objekt „Schaltzustand des Sensors x“ zu verarbeiten haben, und wenn das Objekt einen bestimmten Wert enthält, dass die Antriebe zu starten haben. Der LAS ruft zyklisch das Objekt „Schaltzustand des Sensors x“ auf (Zyklus n). Unmittelbar danach, im gleichen Zyklus, antwortet der Sensor mit dem Schaltzustand. Die Antriebe empfangen die Nachricht quasi gleichzeitig und verarbeiten die Information sofort. Damit hat das Busarbitrationsverfahren einen Zeitvorteil gegenüber dem Master/Slave-Verfahren. Der Nachteil besteht in dem erhöhten Projektierungsaufwand.

1.4 Datensicherung

1.4.1 Einleitung

Jeder Datenübertragung drohen Störungen durch elektromagnetische Einstreuungen, Rauschen, Potentialdifferenzen, Alterung der Bauteile usw. Störung bedeutet Invertierung von Bits. Gegen Störungen ist man nicht machtlos, im Gegenteil, man geht das Problem von zwei Seiten an:

1. Man vermindert durch technische Vorkehrungen die Wahrscheinlichkeit von Störungen, z. B. durch geschirmte Kabel, Glasfaserkabel, potentialfreie Übertragung.
2. Man überwacht die Nachricht auf Fehler und trifft im Fehlerfall Gegenmaßnahmen verschiedenster Art.

Dieser Abschnitt befasst sich mit der dem Punkt 2 zugrunde liegenden Theorie. Dabei lehnen wir uns in der Denk- und Schreibweise an DIN 19 244, Teil 10 an. Dieser Normentwurf vom März 1988 ist identisch mit dem vom IEC Technical Committee 57 erarbeiteten Entwurf, wird also in absehbarer Zeit international verbindlich sein.

1.4.2 Fehlerarten

Wir betrachten hier und im Folgenden stets transparente Codes (= bitorientierte Codes). Das sind Codes, bei denen jede Bitkombination erlaubt und sinnvoll ist. Man kann dann aus der Bitfolge allein nicht auf einen eventuellen Fehler schließen.

Drei Arten von Fehlern kann man unterscheiden:

- erkennbare und korrigierbare Fehler,
- erkennbare und nicht korrigierbare Fehler,
- nicht erkennbare Fehler.

Sind die Fehler erkannt, so hat man schon halb gewonnen. Man kann dann entweder eine automatische Fehlerkorrektur vornehmen (dies geschieht relativ selten, z. B. bei Satellitenkommunikation), oder man wiederholt einfach die Übertragung, hoffend, dass der Fehler nicht mehr an der gleichen Stelle wie zuvor auftreten wird.

Im Abschnitt „Technik der Fehlerbehandlung“ findet man Beispiele für die oben aufgeführten Fehlerarten.

1.4.3 Einige grundlegende Beziehungen

1.4.3.1 Bitfehlerrate

Die Bitfehlerrate p ist ein Maß für die Störempfindlichkeit des Übertragungskanals:

$$p = \frac{\text{Anzahl der fehlerhaften Bits}}{\text{Gesamtzahl der gesendeten Bits}} . \quad (1.1)$$

Der ungünstigste Wert, den p annehmen kann, ist $p = 0,5$. Jedes zweite Bit ist dann im Mittel gestört, die Nachricht also wertlos (wäre $p = 1$, so brauchte man ja nur alle Bits zu invertieren und hätte eine fehlerfreie Nachricht).

Ein in der Technik mit normalem Aufwand realisierbarer Wert ist

$$p = 10^{-4} .$$

Dies ist eine akzeptable Übertragungsqualität.

Statt der Bitfehlerrate p kann man auch die Wahrscheinlichkeit q des Empfangens unverfälschter Bits angeben:

$$q = 1 - p . \quad (1.2)$$

1.4.3.2 Wiederholung einer Übertragung

Die normale Reaktion auf die erkannte fehlerhafte Übermittlung einer Nachricht ist, im täglichen Leben wie in der Datenübertragungstechnik, dass man eine Wiederholung der Übertragung anfordert. Dies ist bekannt unter der Bezeichnung ARQ (error detection with

automatic request repeat). Zwar bleibt die ursprüngliche Bitfehlerrate p auch bei der zweiten Übertragung dieselbe, aber die Wahrscheinlichkeit p^* der Störung an derselben Stelle nimmt stark ab:

$$p^* = p^a, \quad (1.3)$$

p – Bitfehlerrate ,

a – Anzahl der Übertragungen .

1.4.3.3 Restfehlerrate

Die Restfehlerrate R bezieht sich auf die unerkannten Fehler, die nach Anwendung einer Fehlererkennungsstrategie noch verbleiben:

$$R = \frac{\text{Anzahl der unerkannt fehlerhaften Bitkombinationen}}{\text{Gesamtzahl } n \text{ der gesendeten Bitkombinationen}}. \quad (1.4)$$

Die Restfehlerrate R ist ein Maß für die Datenintegrität, d. h. ein Maß für die Unversehrtheit der Daten.

Aus R lässt sich die mittlere Zeit T zwischen zwei unerkannten Fehlern berechnen:

$$T = \frac{n}{v \cdot R}, \quad (1.5)$$

n – Telegrammlänge in Bit ,

v – Übertragungsgeschwindigkeit in bit/s .

Beispiel

In DIN 19 244 ist ein sehr informatives Beispiel dazu aufgeführt:

Telegramme mit $n = 100$ Bit werden pausenlos mit einer Übertragungsgeschwindigkeit von 1200 Bd gesendet. Der Datenkanal habe eine Bitfehlerrate von $p = 10^{-4}$.

Die sich ergebenden mittleren Zeiten T zwischen zwei Fehlern sind dann nach (1.5):

R nach Abb. 1.20	T nach (1.5)	Typische Anwendung
10^{-6}	1 Tag	sich zyklisch aufdatende Systeme
10^{-10}	26 Jahre	Ereignisgesteuerte Übertragung
10^{-14}	260 000 Jahre	Fernsteuerung

1.4.3.4 Hamming-Distanz

Die Hamming-Distanz d ist ein Maß für die Störfestigkeit eines Codes (genannt nach dem Amerikaner R. W. Hamming).

Ist e die Anzahl der sicher erkennbaren Fehler, so gilt:

$$d = e + 1. \quad (1.6)$$

Beispiel

Ist ein einziger Fehler sicher erkennbar (z. B. bei der Verwendung des Paritätsbits, vgl. den diesbezüglichen Abschn. 1.4.4.1), also $e = 1$, so ist nach (1.6) die Hamming-Distanz

$$d = 2.$$

Dies ist das mindeste, was man in der Datenübertragung an Übertragungssicherheit fordert. Bei Feldbus-Systemen ist $d = 4$ üblich, sehr hohe Sicherheitsbedürfnisse erfüllt $d = 6$.

Man kann die Hamming-Distanz d auch mit der Bitfehlerrate p nach (1.1) und der Restfehlerrate R nach (1.4) verknüpfen (vgl. hierzu auch Abschn. 1.4.5):

$$d = \frac{\lg R(p_1) - \lg R(p_2)}{\lg p_1 - \lg p_2}, \quad p_1 > p_2. \quad (1.7)$$

Beispiel

Zur Bitfehlerrate $p_1 = 10^{-3}$ gehöre $R(p_1) = 10^{-4}$, zur Bitfehlerrate $p_2 = 10^{-4}$ gehöre $R(p_2) = 10^{-6}$.

Der durch diese Kombination beschriebene Übertragungskanal hat dann nach (1.7) die folgende Hamming-Distanz d :

$$d = \frac{\lg 10^{-4} - \lg 10^{-6}}{\lg 10^{-3} - \lg 10^{-4}} \\ d = 2.$$

1.4.3.5 Telegrammübertragungseffizienz

Die Effizienz E einer Datenübertragung ist gegeben durch

$$E = \frac{\text{fehlerfreie Informationsbits}}{\text{Gesamtzahl der übertragenen Bits}}. \quad (1.8)$$

Die Effizienz lässt sich folgendermaßen berechnen:

$$E = \frac{k \cdot q^n}{n}, \quad (1.9)$$

k – Anzahl der Informationsbits pro Telegramm,

q – Wahrscheinlichkeit des Empfangs unverfälschter Bits, siehe (1.2),

n – Gesamtzahl aller Bits pro Telegramm, einschließlich Synchronisations- und Fehlerprüfbits.

Beispiel 1

$k = 8$ Bit,

$n = 11$ Bit (1 Startbit + 8 Datenbits + 1 Paritätsbit + 1 Stoppbit),

$q = 1 - 10^{-3}$.

Nach (1.9) ist dann

$E = 0,72$.

Die Telegrammübertragungseffizienz ist also 72 %.

Beispiel 2

$k = 8$ Bit,

$n = 24$ Bit (8 Startbit + 8 Datenbits + 8 Bit CRC),

$q = 1 - 10^{-3}$.

Nach (1.9) ist dann

$E = 0,32$.

Die Telegrammübertragungseffizienz ist also 32 %.

Bei beiden Beispielen haben wir die Bitfehlerrate p des Übertragungskanals gleich gelassen (nämlich 10^{-3}), ebenfalls die Anzahl der zu übertragenden Informationsbits ($k = 8$). Im Beispiel 1 haben wir 3 Bit zur Verminderung der Restfehlerquote R vorgesehen, in Beispiel 2 dagegen 16 Bit. Man sieht hier und das gilt allgemein:

Die Übertragungseffizienz E und die Restfehlerrate R stehen in direktem Verhältnis zueinander: Je sicherer die Übertragung sein soll, desto schlechter ist die Übertragungseffizienz.

1.4.4 Einige Strategien der Fehlererkennung

Der Grundgedanke ist immer derselbe:

- Man erinnert eine Strategie, die Fehler erkennt.
- Man korrigiert die erkannten Fehler (automatisch oder durch ARQ).
- Man quantifiziert die nicht erkennbaren Restfehler durch Wahrscheinlichkeitsrechnung.

Im Folgenden beschreiben wir einige Strategien der Fehlererkennung.

1.4.4.1 Paritätsbit

Wir senden ein ASCII-Zeichen mit 7 Bit, z. B.

1 0 1 0 1 1 1 .

Wir bilden die Quersumme und stellen fest, diese ist ungerade. Haben Sender und Empfänger untereinander gerade Quersumme vereinbart, so wird ein Paritätsbit $P = 1$ hinzugefügt und übertragen:

$$1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ P = 1, \text{ Parität gerade.}$$

Tritt ein Fehler auf (d. h. 1 Bit wird durch Störung invertiert), so ist die Parität P nicht mehr gerade, der Fehler wird erkannt:

$$1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ P = 1, \text{ Parität ungerade, also Fehler.}$$

Zwei Fehler werden nicht erkannt, weil sie sich gegenseitig kompensieren, genauso wie bei vier, sechs oder acht Fehlern.

Drei, fünf, sieben Fehler werden erkannt, können aber nicht von einem Fehler unterschieden werden.

Sicher erkannt wird $e = 1$ Fehler, deshalb ist nach (1.6) die Hamming-Distanz

$$d = 2.$$

1.4.4.2 Blocksicherung

Wir senden nacheinander 7 ASCII-Zeichen nebst Paritätsbit. Dann folgt ein 8. Oktett P , das die Spaltenparitäten enthält.

Beispiel

1. Fehlerfreies Sendetelegramm (gerade Parität)

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	0	0	1	1
5.	1	1	0	0	1	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

2. Wird 1 Bit gestört, d. h. invertiert, so hat man eine Paritätsverletzung, beispielsweise in der 4. Zeile und der 5. Spalte. Der Fehler ist erkannt und kann korrigiert werden.

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	1	0	1	1
5.	1	1	0	0	1	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

3. Werden 2 Bit gestört, so hat man eine Paritätsverletzung z. B. in Spalte 5 in den Zeilen 4 und 5. Spalte 5 liefert keine Paritätsmeldung. Es sind also zwei Fehler erkennbar, aber keiner lokalisierbar und damit auch nicht korrigierbar.

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	0	0	1	1
5.	1	1	0	0	1	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

4. Nun betrachten wir drei Bitfehler in ungünstiger Kombination:

	1.	2.	3.	4.	5.	6.	7.	P
1.	0	1	0	1	0	1	1	0
2.	1	1	1	0	0	0	1	0
3.	1	0	0	1	0	0	1	1
4.	0	0	1	1	1	1	1	1
5.	1	1	0	0	0	1	1	1
6.	0	0	1	1	0	0	1	1
7.	1	1	0	0	0	1	1	0
P	0	0	1	0	1	1	1	0

Eine Fehlersignalisation erfolgt hier wie gewünscht, allerdings an der falschen Stelle. Eine Korrektur ist nicht nur nicht möglich, sondern sie würde sogar an der falschen Stelle durchgeführt.

Vier Fehler werden nicht mehr erkannt, fünf wieder, sechs nicht usw.

Sicher erkannt werden also $e = \text{drei Fehler}$, deshalb ist nach (1.6) die Hamming-Distanz

$$d = 4 .$$

1.4.4.3 CRC

Aus der üblichen Bezeichnung CRC, *Cyclic Redundancy Check*, ist nicht erkennbar, wie dieses sehr häufig angewendete Verfahren arbeitet. Man fasst die Information, unabhängig von ihrer Länge, als Zahl auf, die durch eine andere, feste Zahl, das so genannte Generatorpolynom G im Sender dividiert wird (Abb. 1.19).

Den Quotienten Q verwirft man, den resultierenden Rest R hängt man an die Information 1 an und sendet den so entstandenen Codevektor IR . Der Empfänger dividiert den Codevektor durch dasselbe Polynom G und erhält bei fehlerfreier Übertragung den Rest $R = 0$.

Die Division erfolgt mit einfachen Regeln $1 + 1 = 0$ (es erfolgt kein Übertrag)

$$0 - 1 = 1$$

$$1 - 1 = 0$$

$$0 + 1 = 1$$

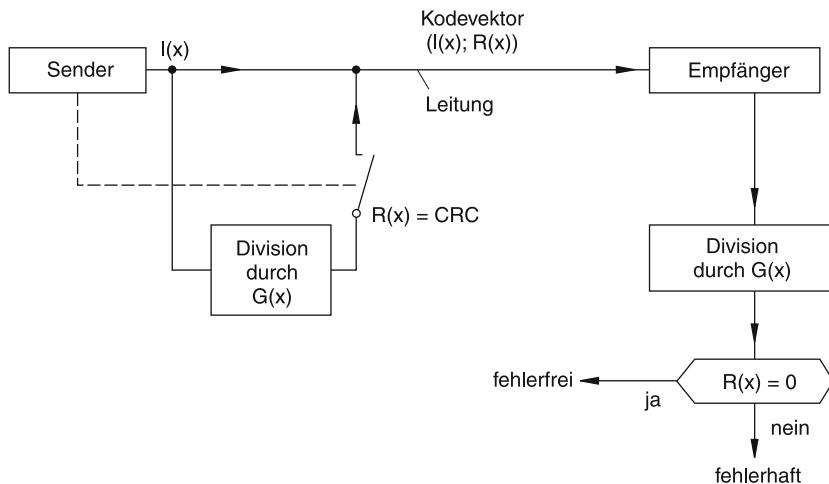


Abb. 1.19 Fehlererkennung mit CRC

Diese Regeln sind hardwaremäßig realisierbar durch eine EXOR-Verknüpfung. Abb. 1.20 zeigt ein Beispiel in dezimal und binär. Die Generatorpolynome G werden auf verschiedene Weise geschrieben:

Als Formel: $G = x^7 + x^6 + x^5 + x^2 + 1$

Als Binärzahl: $G = 11\ 100\ 101$

Als Oktalzahl: $G = 345$

Die Länge des Polynoms ist, neben anderen Eigenschaften, maßgeblich für die erreichbare Hamming-Distanz. Die in DIN 19 244 vorgeschlagenen Polynome haben folgende Form:

Polynomform	HD d	Format	$i = 2$	$i = 3$	$i = 4$
345	4	FT2			
235 546	6	FT3	26	94	304

Die Tabelle vor Abb. 1.20 zeigt für das optimierte Generatorpolynom $G = 235\ 546$, dass nicht alle Fehler-Muster erkannt werden: Für den Fall von sechs zufälligen Fehlerbits ist die Zahl der nicht erkennbaren Fehler bei $i = 2, 3$ und 4 gesendeten Datenoktettts beispielhaft dem Anhang von DIN 19 244 entnommen. Auch wird die Hamming-Distanz $d = 6$ nur bis $i = 16$ aneinanderfolgende Datenoktettts garantiert.

CRC (Cyclic Redundancy Check)	
1. Die Nachricht sei I	
Beispiel dezimal: $I = 14$	binär: $I = 110101$
2. Das Prüfpolynom sei G	
$G = \dots a_4 \cdot x^4 + a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 \cdot x^0$	
d.: $\dots 0 \cdot x + 3 \cdot x^0 = 3$	b.: $\dots 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 \cdot x^0 = 1011$
3. Der Hilfsvektor ist dann $H = 100 \dots 0$	
d.: $H = 10$	b.: $H = 10000$
4. Die Information I wird mit H multipliziert:	
$B = I \cdot H$	
d.: $B = 140$	b.: $B = 1101010000$
5. Das Produkt B wird durch G dividiert:	
$\frac{B}{G} = Q + \frac{R}{G}$	
d.: $\frac{B}{G} = \frac{140}{3} = 46 + \frac{2}{3}$	b.: $\frac{B}{G} = \frac{1101010000}{1011}$
	$= 1111011 + \frac{101}{1011}$
6. Der Rest R wird B hinzugefügt und das Ganze gesendet	
d.: $B + R = 142$	b.: $B + R = 1101010101$
7. Der Empfänger bildet	
$(B - R) : G$	
d.: $(B - R) : G = 138 : 3$	b.: $(B - R) : G$
	$= (B + R) : G = 1101010101 : 1011$
	$= 1111011, R = 0$
8. Es bedeutet $R = 0$ fehlerfreie Übertragung	

Abb. 1.20 CRC (Cyclic Redundancy Check) ($1 + 1 = 0, 0 - 1 = 1$)

1.4.5 Datenintegritätsklassen

Das Diagramm in Abb. 1.21 zeigt ein Achsenkreuz in doppelt logarithmischem Maßstab. Nach rechts ist die durch den Übertragungskanal gegebene Bitfehlerrate p aufgetragen, nach oben die durch irgendeinen Code erzielbare Restfehlerrate R . Jeder sicherheitsorientierte Code wird in diesem Achsenkreuz durch eine Kurve dargestellt.

Diese Kurve repräsentiert ein bestimmtes Maß an Datensicherheit, oder, wie die Norm 19 244 sagt: eine bestimmte Datenintegrität. Die drei in Abb. 1.21 eingezeichneten Kurven I1, I2 und I3 repräsentieren drei Datenintegritätsklassen I1, I2, I3, welche die Norm für verschiedene Sicherheitsbedürfnisse vorschlägt.

Man sieht, dass die Integritätsklasse I1 das niedrigste Maß an Sicherheit, d. h. die niedrigste Integrität bietet und I3 die höchste.

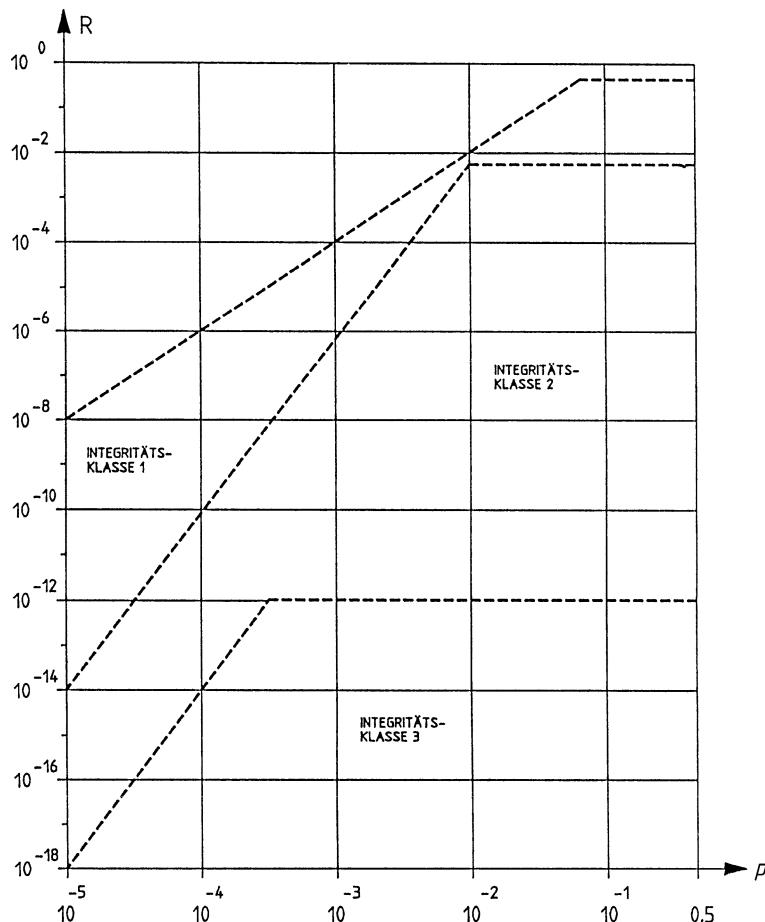


Abb. 1.21 Definition der Integritätsklassen nach DIN 19 244

Die Steigungen der Rampen gehorchen der Beziehung (1.7), was bedeutet, dass die Steigung der Rampe der Hamming-Distanz d einer bestimmten Codierungsmethode entspricht.

Zur Veranschaulichung der drei Integritätsklassen möge der Leser die Tabelle in Abschn. 1.4.3.3 betrachten. Die drei dort gewählten Beispiele sind – von oben nach unten – den drei Integritätsklassen I1, I2, I3 zuzuordnen.

Die Integritätsklassen sind künstliche Gebilde, die es erlauben, das Maß der Datensicherheit eines bestimmten Code auszudrücken. Über den Code selbst sagen sie nichts aus.

1.4.6 Telegrammformate

Eine Datenintegritätsklasse bleibt ein theoretischer Begriff, solange kein Telegrammformat mit seinem spezifischen Datensicherungsverfahren dahinter steht. Im Folgenden werden einige bekannte Telegrammformate daraufhin untersucht.

1.4.6.1 Telegramm mit Paritätsbit

Das folgende Telegrammformat ist weit verbreitet in der Form:

Startbit, 7 Bit-ASCII-Zeichen, Paritätsbit, Stopppbit

und wird als UART-Format bezeichnet (*universal asynchronous receiver/transmitter*).

Das hier untersuchte Format hat allerdings eine leicht unterschiedliche Form:

Startbit 0, 8 Informationsbits, gerades Paritätsbit, Stopppbit 1.

Es ist hardware-kompatibel mit allen PCs und deren Peripherie.

Bezeichnung nach DIN 19 244: (11,8)-Code

Formatklasse nach DIN 19 244: FT 1.1

Hamming-Distanz: $d = 2$

Die Anzahl der unerkannt bleibenden Fehlermuster (= Restfehler) ist

$$A_e = \binom{m}{e}, \quad (1.10)$$

wobei:

$$\begin{aligned} e &= 2, 4, 6, 8, && \text{Anzahl der gleichzeitig auftretenden gestörten Bits,} \\ &&& \text{die nicht erkennbar sind,} \\ m &= 9, && \text{Anzahl der Datenbits, inklusive Paritätsbit} \end{aligned}$$

ist.

Aus (1.10) folgt für den (11,8)-Code:

$$\begin{aligned} A_2 &= 36, \\ A_4 &= 126, \\ A_6 &= 84, \\ A_8 &= 9. \end{aligned}$$

Das heißt beispielsweise, dass es 126 Bitkombinationen gibt, wo $e = 4$ gestörte Bits gleichzeitig vorkommen, die sich gegenseitig aufheben, also die Parität nicht ändern.

Aus der Zahl A der Restfehler und der Bitfehlerrate p bzw. $q = 1 - p$ lässt sich die Restfehlerrate R berechnen:

$$R(p) = \sum_e (A_e \cdot p^e \cdot q^{m-e}) \cdot q^2. \quad (1.11)$$

Im Diagramm in Abb. 1.22 ist (1.11) als oberste Kurve eingetragen [19]. Man erkennt, dass das Telegramm mit Paritätsbit die Integritätsklasse I1 erfüllt. Die Übertragungseffizienz E ist (wie in Abschn. 1.4.3.5 bereits berechnet):

$$E = 72\%.$$

1.4.6.2 Telegramm mit CRC

Dieses Telegrammformat wird insbesondere verwendet, wenn mehrere Datenoktette zusammen hintereinander übertragen werden sollen. Es hat die in Abb. 1.23 gezeigte Form.

Bezeichnung nach DIN 19 244: $(8i + 8, 8i)$ -Code; $i = 1 \dots 15$

Formatklasse nach DIN 19 244: FT2

Hamming-Distanz: $d = 4$

Generatorpolynom: $x^7 + x^6 + x^5 + x^2 + 1$

Das Prüfzeichen wird mit geradem Paritätsbit ergänzt und invertiert.

Für die Anzahl A der unerkannt bleibenden Fehlermuster gilt entsprechend (1.10):

$$A_{n,e} \cong \frac{1}{128} \cdot \binom{n}{e}, \quad (1.12)$$

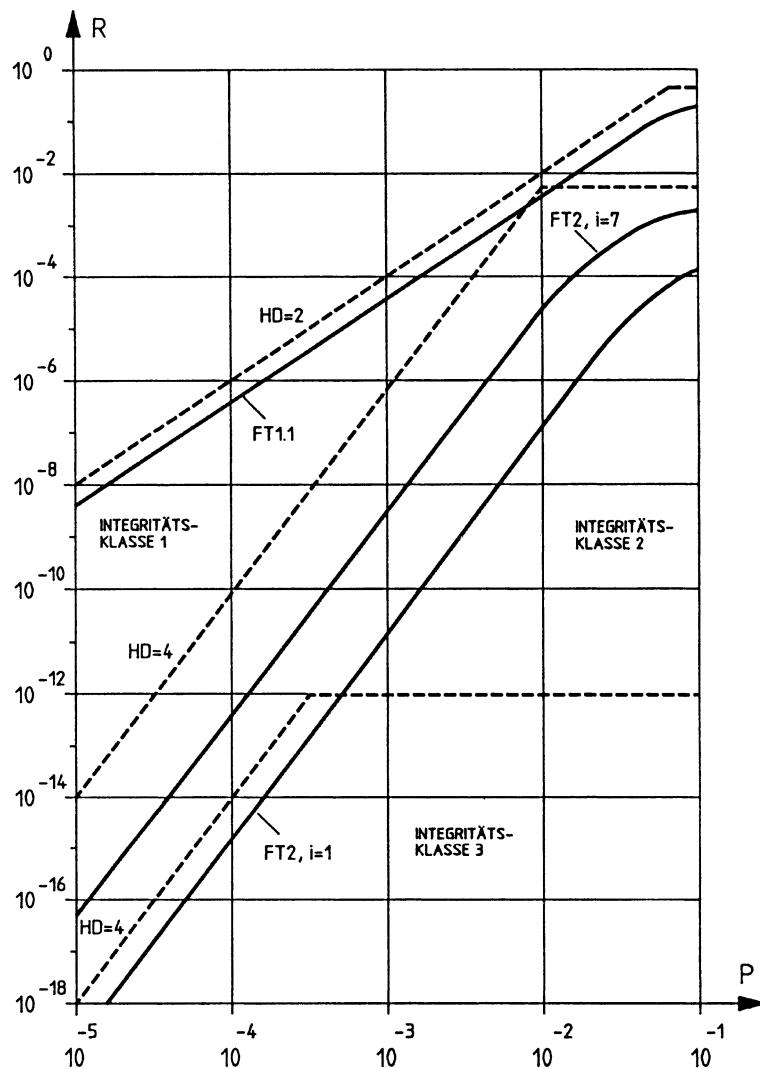
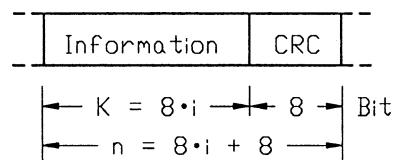


Abb. 1.22 Restfehlerrate R verschiedener Formatklassen (nach DIN 19244)

Abb. 1.23 Telegramm der Formatklasse FT2 mit 8 Bit Prüfzeichen



wobei

$$e = 4, 6, 8, \dots n \text{ und}$$

n – Gesamtzahl der übertragenen Bits ist.

Beispiel

Wir wollen nur 1 Datenoktett übertragen, also $i = 1, n = 16$. Aus (1.12) folgt für die Zahl der unerkannt bleibenden Fehler mit $e = 4$ gleichzeitig gestörten Bit:

$$A_{16,4} = 14,2 .$$

Dies sind fast 10-mal weniger unerkannte Fehler als bei der Paritätsprüfung, ein Zeichen für die Wirksamkeit des CRC.

Aus der Zahl A der Restfehler und der Bitfehlerrate p bzw. $q = 1 - p$ lässt sich mit (1.11) die Restfehlerrate $R(p)$ berechnen. Die Funktion $R(p)$ ist gleichfalls in Abb. 1.22 aufgetragen, und zwar für 2 Fälle:

$$i = 1 \text{ (nur 1 Datenoktett)} \quad \text{und} \quad i = 7 \text{ (7 Datenoktette).}$$

Man erkennt daran, dass das Telegramm mit 8-stelligem Generatorpolynom zur Integritätsklasse I2 gehört.

Die Übertragungseffizienz E ist, wie in Abschn. 1.4.3.5 für $i = 1$ Datenoktett berechnet (8 Startbits, i Datenoktette, 8 CRC-Bits):

$$E_1 = 32 \% .$$

Für 7 Datenoktette ($i = 7$) ergibt (1.9):

$$E_7 = 72 \% .$$

Man erkennt, dass der Code in seiner Form (64, 56)-Code, also für sieben Datenoktette, wesentlich effizienter ist als für ein Oktett.

1.5 Telegrammformate

1.5.1 Das HDLC-Protokoll

Die OSI-Schicht 2 dient der Datenübertragung und -sicherung. Spezielle Übertragungsprotokolle für diese OSI-Schicht bietet ein Satz von Protokollen zur bitseriellen Übertragung von Daten, der von ISO 1987 in seiner endgültigen Fassung genormt wurde. Es sind

die unter dem Namen *High Level Datalink Control Procedures*, kurz HDLC, bekannten Datenübertragungsprotokolle, die weite Verbreitung gefunden haben.

Die HDLC-Datenübertragung läuft synchron oder im Start/Stopp-Betrieb und beinhaltet eine mehrfache Sicherung. Ein Datenpaket enthält jeweils eine Adresse, Kontroll- und Steuerinformation, die zu übertragenden Daten und Prüfinformation.

Werden Bytes übertragen und ist die Übertragungsreihenfolge der einzelnen Bits innerhalb eines Bytes im folgenden Beispiel vom niederwertigsten Bit aufwärts, so ergibt sich für die zu übertragende ASCII-Zeichenfolge „987“ die zugehörige Bitfolge

„10011100 10001100 01111100“.

Ist diese Bitfolge ermittelt, dann werden anschließend die Daten so verändert, dass maximal fünf Bits hintereinander auf logisch Eins gesetzt sind, d. h. dass maximal fünf Einsen aufeinander folgen. Nach jeder fünften Eins in Folge wird deswegen eine Null eingefügt, die der Empfänger später wieder herausfiltert. Diese Vorgehensweise bezeichnet man als *Bit Stuffing*. Für das Beispiel von oben sieht dies so aus:

„10011100 10001100 011111 **0** 00“.

Zu Beginn und zum Abschluss jedes HDLC-Rahmens werden Anfangs- und Endekennungen gesendet. Sie bestehen jeweils aus einer Folge von sechs Einsen eingebettet in Nullen, hexadezimal „7E“, binär „01111100“. Dieses so genannte *Flag-Byte* kann, je nach Vereinbarung und Art der Übertragung ein- oder mehrmals vor und nach dem Datenpaket gesendet werden. Bei einigen Datenübertragungen werden permanent Flag-Bytes über die Leitung gesendet, solange kein Datenverkehr stattfindet, bei anderen wird eine vorher festgelegte Anzahl von Flag-Bytes vor jedem neuen Telegramm gesendet, sodass der Empfänger sich darauf synchronisieren kann.

Ein HDLC-Rahmen besteht konsekutiv aus einem führenden Flag-Byte (01111100), der Adresse einer Datenstation (8 Bits), einer Kontrollinformation (8 Bits), den Informationsbits (nicht festgelegte Anzahl von Bits), einer Prüfsumme (*Frame Check Sequence FCS*, 16 oder 32 Bits) und zum Abschluss wiederum einem Flag-Byte (01111100).

Flag	Adresse	Kontrolle	Information	FCS	FLAG
01111100	8 Bits	8 Bits	<i>n</i> Bits	16 o. 32 Bits	01111100

Informationsbits sind nicht in allen Telegrammen enthalten, alle anderen Telegrammteile sind obligatorisch. Bei allen genannten Bestandteilen des HDLC-Rahmens außer den Informationsbits werden die Bits in der Reihenfolge vom niederwertigen zum höchswertigen Bit übertragen. Für die Informationsbits ist keine Reihenfolge festgelegt. Die Anzahl der Informationsbits ist weder vorgegeben, noch muss ihre Anzahl durch acht teilbar sein, d. h., sie kann vollkommen willkürlich vom Benutzer gewählt werden.

Der Aufbau des Kontrollfeldes ist in der HDLC-Norm festgelegt. Es werden drei Arten von Telegrammen mit den zugehörigen Kommandos und Antworten unterschieden. Es existieren Telegramme

- für den Informationstransfer (I-Format),
- für die Überwachung des Datenverkehrs (S-Format) und
- solche mit nicht nummerierten Formaten (U-Format).

Der Aufbau der Kontrollfelder der drei verschiedenen Formate ist nachfolgend dargestellt.

	1	2	3	4	5	6	7	8
I-Format	0	N (S)			P/F	N (R)		
S-Format	1	0	S	S	P/F		N (R)	
U-Format	1	1	M	M	P/F	M	M	M

Die in der Darstellung benutzten Kürzel bedeuten

N(S) Sendefolgezähler	N(R) Empfangsfolgezähler
P/F Poll- oder Endebit	S, M Funktionsbits.

HDLC-Telegramme im I- oder S-Format enthalten Zähler, die der Datensicherung dienen. Der Sendefolgezähler N(S) wird mit jedem ausgesendeten Telegramm inkrementiert (Modulo 8) und gibt an, wie viele Telegramme die sendende Station bereits ausgesendet hat. Der Empfangsfolgezähler N(R) zeigt an, auf welches Telegramm der Empfänger gerade wartet (Modulo 8) und er bestätigt den Eingang aller Telegramme mit niedrigerer Empfangsfolgenummer.

Die Sende- und Empfangsfolgenummern werden von den in den Kommunikationspartnern verwalteten Sende- und Empfangsstatusvariablen abgeleitet.

Mit der HDLC-Norm werden drei Übertragungsklassen definiert. Es sind

- der Normal Response Mode (NRM),
- der Asynchronous Response Mode (ARM) und
- der Asynchronous Balanced Mode (ABM).

Der Normal Response Mode (NRM) beschreibt einen Master-Slave-Betrieb, bei der eine Primary Station (Master) mit einer Secondary Station (Slave) kommuniziert. Die Secondary Station darf nicht spontan, sondern nur nach Aufforderung durch die Primary Station senden.

Im Asynchronous Response Mode (ARM) darf die Secondary Station spontan Telegramme verschicken, ohne zuvor die Erlaubnis der Primary Station abzuwarten. Übertragen werden dabei vor allem Informationsfelder (I-Frames) und Information über Statusänderungen.

Im Asynchronous Balanced Mode (ABM) sind beide Stationen gleichberechtigt und enthalten beide jeweils alle Funktionen der Primary und Secondary Station und werden daher als Combined Station bezeichnet.

Das HDLC-Protokoll beinhaltet Befehle für den Verbindungsaufbau, den Übertragungsablauf und den Verbindungsabbau. Am Beispiel des Normal Response Mode sind nun die Kommandos und Antworten des Basisfunktionsrepertoires dargestellt.

Basiskommandos sind:

- I Senden eines Informationsframes,
- RR Nachricht, dass Empfangskanal bereit ist (Receiver Ready),
- RNR Nachricht, dass Empfangskanal nicht bereit ist (Receiver Not Ready),
- SNRM Aufforderung zum Verbindungsaufbau (Set Normal Response Mode) und
- DISC Aufforderung zum Verbindungsabbau (Disconnect).

Die möglichen Antworten der Secondary Station darauf sind:

- I Senden eines Informationsframes,
- RR Nachricht, dass Empfangskanal bereit ist (Receiver Ready),
- RNR Nachricht, dass Empfangskanal nicht bereit ist (Receiver Not Ready),
- DM Verbindung unterbrochen (Disconnected Mode),
- UA Antwort bei erfolgreichem Verbindungsaufbau (Unnumbered Acknowledgement) und
- FRMR Zurückweisen eines unverstandenen HDLC-Rahmens (Frame Error).

Die typischen Übertragungsgeschwindigkeiten in technischen Anwendungen reichen von 9,6 kbit/s bis zu 2 Mbit/s.

Das HDLC-Protokoll beinhaltet eine ganze Reihe von Sicherheiten für die Übertragung. Zunächst bietet das oben erwähnte Bit-Stuffing eine Sicherheit bei der physikalischen Übertragung, fehlerhafte Rahmen werden bereits beim Empfang erkannt und können verworfen werden. Die Sende- und Empfangsfolgezähler ermöglichen es, Sender- und Empfänger einen Datenverlust oder einen Fehler in der Übertragungsstrecke zu erkennen. Zusätzlich dazu beinhalten HDLC-Übertragungsstrecken eine Zeitüberwachung. Ist ein Telegramm innerhalb einer vorgegebenen Zeit, die von der Übertragungsgeschwindigkeit abhängt nicht bestätigt, so gilt das Telegramm als verloren und eine vorprogrammierte Fehlerbehandlung wird durchgeführt.

SDLC ist ein von IBM genormtes Protokoll, das sich eng an HDLC in der synchronen Betriebsart anlehnt.

1.5.2 UART

Einer der häufigsten Übertragungsbausteine im Bereich der Microcomputer-Hardware ist der *Universal Asynchronous Receiver and Transmitter* (UART). Er dient einer bitseriellen Übertragung von Daten über eine Übertragungsstrecke und der Umsetzung der Datenwörter zum und vom parallelen Rechnerbus. Wie der Name bereits ausdrückt, handelt es sich

um eine asynchrone Datenübertragung, d. h. es ist kein Synchronsignal für Sender und Empfänger vorhanden. Die Übertragung ist zeichenorientiert, sodass die Synchronisation bei jedem Zeichen aufs Neue ausgeführt werden muss.

Der Aufbau eines UART-Zeichens ist in DIN 66 022/66 203 beschrieben. Es besteht aus elf Bits und beginnt mit einem Startbit, das logisch Null ist. Es endet mit einem Stopppbit, das immer eine logische Eins enthält.

Startbit	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parität	Stopppbit
	LSB							MSB		

Auf das Startbit folgt die zu übermittelnde Information als Datenbits. Nach den Datenbits und vor dem Stopppbit wird ein so genanntes Paritätsbit gesendet. Ist gerade Parität vereinbart, so ergänzt das Paritätsbit die Quersumme auf einen geraden Wert, ist ungerade Parität vereinbart, so ergänzt das Paritätsbit die Quersumme auf einen ungeraden Wert. Die Zahl der Datenbits kann auf sieben oder acht eingestellt werden. Das niederwertige Bit (LSB) wird immer direkt nach dem Startbit, das höchstwertige Bit (MSB) als letztes Datenbit gesendet.

Mögliche Übertragungsarten sind simplex, halbduplex und vollduplex. Bei der Übertragungsart simplex können nur in einer Richtung Daten übertragen werden. Bei Strecken, die halbduplex arbeiten, können Daten in beiden Richtungen übertragen werden, jedoch nicht in beiden Richtungen gleichzeitig. Arbeitet die Datenübertragung vollduplex, so ist eine asynchrone Übertragung von Daten in beide Richtungen jederzeit möglich.

Der Sender gibt seine Zeichen in einem vorher in beiden Stationen vereinbarten Zeitstrahl auf die Übertragungsstrecke. Beim Empfänger ankommende Zeichen werden aufgrund der eingestellten Datenübertragungsgeschwindigkeit und der Struktur des eingehenden Rahmens beurteilt.

Alle Parameter der Datenübertragung müssen auf beiden Seiten der Übertragungsstrecke voreingestellt sein. Die Übertragungsgeschwindigkeit ist in Stufen einstellbar. Üblich sind 150 bit/s, 300 bit/s, 600 bit/s, 1200 bit/s, 2400 bit/s, 4800 bit/s, 9600 bit/s, 19 200 bit/s, 38 400 bit/s und 64 kbit/s.

Die maximale Länge der Übertragungsstrecke hängt vor allem von Hardwarekriterien und von der Übertragungsgeschwindigkeit ab.

1.5.3 PROFIBUS-Norm EN 50 170 Teil 2

Die Datenübertragung des PROFIBUS bedient sich der im vorhergehenden Abschnitt beschriebenen UART-Zeichen. Es werden jedoch nicht einzelne Zeichen, sondern Zeichentelegramme, bestehend aus Zeichenketten, übertragen.

Die Telegramme bei PROFIBUS-Übertragungen können reine Informationstelegramme ohne Daten oder Telegramme mit Daten sein.

Informationstelegramme ohne Daten haben eine feste Länge von 6 Byte.

Startzeichen	Zieladresse	Quelladresse	Kontrollbytes	FCS	Endzeichen
--------------	-------------	--------------	---------------	-----	------------

Sie beginnen mit einem Startzeichen, das den Befehlscode des Telegrammformats enthält. Danach folgen die Ziel- und die Quelladresse zur Identifikation der Empfangs- und Sendestationen. Eine anschließend gesendete Prüfsumme (*Frame Check Sequence FCS*) dient der Datensicherung. Zum Abschluss des Telegramms folgt ein Endezeichen.

Telegramme mit Daten gibt es zum einen mit festgelegter Datenlänge und damit auch festgelegter Gesamtlänge und mit variabler Datenlänge. Telegramme mit Daten fester Länge haben folgenden Aufbau:

Startzeichen	Zieladresse	Quelladresse	Kontrollbytes	8 Datenbytes	FCS	Endzeichen
--------------	-------------	--------------	---------------	--------------	-----	------------

Da die Datenlänge fest vorgegeben ist, wird keine zusätzliche Längenangabe benötigt. Bei Telegrammen, deren Länge nicht im Voraus feststeht, muss sie im Telegramm mit übertragen werden. Die maximale Länge eines PROFIBUS-Telegramms ist 255 Byte, d. h., die maximale Anzahl der Informationsbytes ist $n = 244$:

1. Startzeichen	Länge	(Länge)	2. Startzeichen	Zieladresse	Quelladresse	...
...	Kontrollbytes	n Daten		FCS		Endzeichen

Das zweimalige Senden eines Startzeichens bei dieser Art von Telegramm dient der Übertragungssicherheit. Auch die Längenangabe wird aus Gründen der Datensicherheit wiederholt.

Achtung: Auf der physikalischen Schicht wird jedes Byte im UART-Rahmen gesendet: 1 Startbit, 8 Datenbits, gerade Parität, 1 Stopbit.

1.5.4 HART-Protokoll

Im Bereich der chemischen, petrochemischen und verfahrenstechnischen Industrie tritt sehr häufig das Problem auf, dass Signale in explosionsgefährdeten Bereichen erfasst werden müssen. Dabei darf auch im Fehlerfall, z. B. bei Leitungskurzschluss, das umgebende Medium nicht zur Zündung gebracht werden. Dies wird bei der Zündschutzart „Eigensicherheit“ durch Limitierung der Energie erreicht.

Diese Zündschutzart ist die einzige, die in Bereichen mit ständiger Anwesenheit von explosiven Materialien (z. B. Benzintank) verwendet werden darf. Damit ist erklärt, warum die oben genannten Industriezweige eigensichere Bussysteme fordern. Hier konkurrieren zurzeit zwei Systeme, der PROFIBUS-PA und der FOUNDATION™

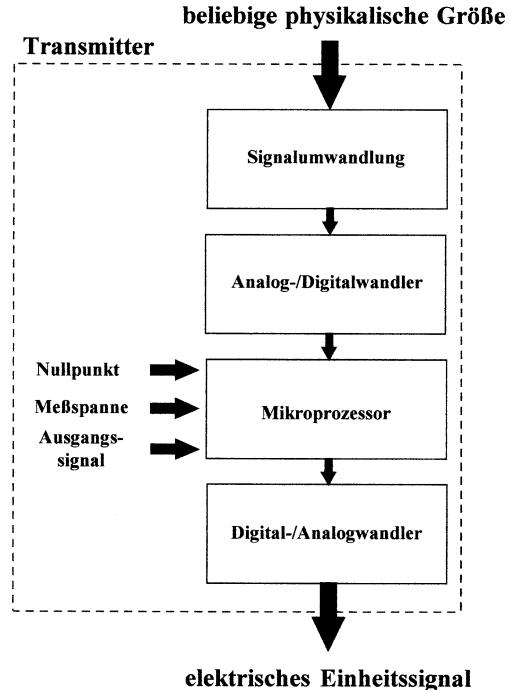
Fieldbus. Problematisch ist zur Zeit, dass nicht alle prozesstechnischen Feldgeräte eine Busschnittstelle aufweisen. Weiterhin ist die Zahl der Busteilnehmer aufgrund der Anforderungen des Explosionsschutzes stark eingeschränkt. Deshalb werden in vielen Anwendungen so genannte Remote-I/O-Systeme eingesetzt. Dabei handelt es sich um dezentrale Ein-/Ausgabeeinheiten, die mit den Feldgeräten parallel verbunden sind und eine serielle busfähige Schnittstelle zur Kommunikation mit der Steuerung besitzen. In der Vergangenheit, als keine eigensicheren Bussysteme verfügbar waren, hat sich das HART-Protokoll (Highway Adressable Remote Transducer) als Quasistandard zur Kommunikation zwischen einer Anzeige- und Bedienkomponente (ABK) und intelligenten Feldgeräten durchgesetzt. Der Nachteil der Remote-I/O-Systeme besteht darin, dass keine direkte Kommunikation zwischen der ABK oder der Steuerung und dem Feldgerät stattfinden kann. Deshalb fordern Anwender prozesstechnische Busse, dass das HART-Protokoll über den Bus übertragen werden kann.

1.5.4.1 Smart-Transmitter

In der Prozessautomatisierung werden sehr häufig so genannte Transmitter eingesetzt. Diese Transmitter wandeln eine nichtelektrische physikalische Größe in ein elektrisches Einheitssignal, in der Regel 0/4 bis 20 mA, um (Abb. 1.24).

Um diese Aufgabe wahrzunehmen, wird das Eingangssignal analog/digital gewandelt und von einem Mikroprozessor weiterverarbeitet. Hier müssen Kennwerte, wie z. B.

Abb. 1.24 Funktionsprinzip
Transmitter



Messbereichseinstellung, Nullpunkt Korrektur etc., eingestellt werden. Da die Kennwerte digital verarbeitet werden, ist es sinnvoll, die Einstellungen ebenfalls digital per Datenübertragung vorzunehmen. SMART-Systeme gestatten es, dem analogen Messsignal ein digitales Signal zu überlagern, sodass eine Kommunikation mit Handheld-Terminals, PCs oder Prozessleitsystemen stattfinden kann. Der Vorteil besteht darin, dass eine Änderung der Einstellung ferngesteuert vorgenommen werden kann.

Bei dieser Art der Kommunikation hat sich das HART-Protokoll als Quasistandard durchgesetzt. Es wird in erster Linie dazu verwendet, um z. B. Daten für die Qualitätsüberwachung, Parameterwerte oder Diagnosedaten zu übertragen.

1.5.4.2 Busaufbau

Es handelt sich dabei um ein Master/Slave-Verfahren, wobei bis zu 2 Master angeschlossen werden können. Als Übertragungsverfahren wird das Frequency-Shift-Keying-Verfahren benutzt, wobei logisch 0 durch ein sinusförmiges 2200 Hz-Signal dargestellt wird. Logisch 1 wird durch ein 1200 Hz-Signal repräsentiert. Die Übertragungsrate beträgt 1200 bit/s.

Das HART-Protokoll lässt sowohl eine Punkt-zu-Punkt- als auch eine Multidrop-Verbindung zu. Im zweiten Fall ist eine parallele Übertragung von analogem Messsignal und digitaler Information nicht möglich. Diese Option wird relativ selten verwendet, da aufgrund der geringen Übertragungsrate die Zykluszeit des Systems relativ hoch ist.

Viel häufiger wird die Punkt-zu-Punkt-Verbindung eingesetzt, wo mit Hilfe eines HART-Multiplexers Kommunikationskanäle zu mehreren Teilnehmern nacheinander aufgebaut werden können. Die Messsignalübertragung zum Prozessleitsystem bleibt von dieser Signalübertragung unberührt. Das Telegrammformat wird in Abb. 1.25 dargestellt.

Die Präambel dient der Synchronisation zwischen Master und Slave. Nach dem Startzeichen folgt die Adresse des angesprochenen Slaves. Hier muss zwischen dem Short-Frame-Format und dem Long-Frame-Format unterschieden werden. Welche Übertragungsform momentan verwendet wird, wird dem Slave mit dem Startzeichen mitgeteilt.

Bei dem älteren Short-Frame-Format werden von dem Adressbyte 4 Bit für die Slaveadresse reserviert. Damit sind im Multidrop-Verfahren jedoch nur 15 Teilnehmer adressierbar, da die Adresse 0 für die Punkt-zu-Punkt-Verbindung reserviert ist. Mit den noch verbleibenden 4 Bits wird z. B. eine Masteradressierung durchgeführt.

Beim neueren und heute verwendeten Long-Frame-Format stehen 38 Bit zur Adressierung der Teilnehmer zur Verfügung. Die restlichen 2 Bits sind Kontrollbits. Die Adressbits sind so strukturiert, dass der Gerätehersteller, der Gerätetyp und eine Geräteidentifikationsnummer hinterlegt werden. Damit besitzt jedes Gerät eine eindeutige und durch einen Anwender nicht veränderbare Adresse. Diese eindeutige Adressierung gestattet die Erstellung einer Datenbank mit gerätespezifischen Daten von Feldgeräten unterschiedlicher Anbieter. Der Anwender kann dadurch Standardeinstellungen durch Übertragen der Datenbankwerte an das Feldgerät vornehmen.

Mit dem Bytezähler wird die Länge des Telegrammes übertragen. Das folgende Statusbyte ist nur in der Slaveantwort enthalten und beschreibt den aktuellen Zustand des

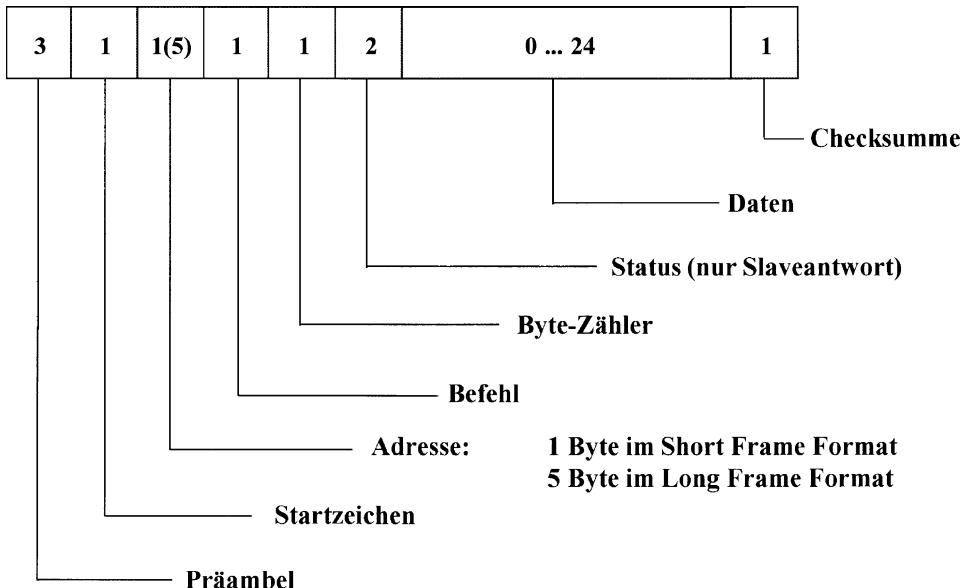


Abb. 1.25 Hart-Telegrammformat

Slaves. Die zu übertragende Information kann bis zu 24 Byte lang sein. Gesichert wird das Telegramm durch eine Checksumme.

1.5.4.3 Buszeiten

Dass das HART-Protokoll nicht für schnelle Prozesse geeignet ist, soll das folgende Beispiel verdeutlichen.

Bei einer Anlage mit 15 Slaves, die jeweils 2 Byte Daten übertragen, ist der Masteraufruf 12 Byte lang (3 Byte Präambel, 1 Byte Startzeichen, 5 Byte Adresse, 1 Byte Befehl, 1 Byte Bytezähler, 1 Byte Prüfsumme). Die Slaveantwort besteht aus insgesamt 16 Byte, da zusätzlich zu den oben genannten Bytes noch die Daten (2 Byte) und der Status (2 Byte) hinzukommen. Das bedeutet, dass der Austausch von 2 Byte Daten insgesamt 28 Byte Datenübertragung umfasst. Die einzelnen Bytes werden asynchron als UART-Zeichen gesendet. Damit besteht 1 Byte aus 11 Bit. Für den Datenaustausch müssen demnach 308 Bit übertragen werden, was bei einer Übertragungsrate von 1200 bit/s etwa 256,6 ms in Anspruch nimmt. Damit ergibt sich für die Gesamtzykluszeit bei 15 Slaves ein Wert von ca. 3,85 s.

Wie schon erwähnt, wird das HART-Protokoll sehr häufig in Verbindung mit speziellen HART-Multiplexern verwendet. Solche Multiplexer werden z. B. von den Firmen Pepperl+Fuchs, Elcon und MTL angeboten und gestatten es, bis zu 7936 HART-fähige Geräte zu verwalten. In der maximalen Ausbaustufe würde die Zykluszeit bei obigem Beispiel ca. 34 Minuten betragen, ohne dass dabei die Zeiten, die der HART-Multiplexer zur Verarbeitung benötigt, berücksichtigt sind. Bei zunehmendem Datenumfang steigt die Zykluszeit entsprechend an.

1.5.5 Token-Telegramm

Im physikalischen oder virtuellen Token-Ring rotiert gleichmäßig das Frei-Token. Es besteht aus drei Byte:

Startmarke	Zugriffskontrolle	Endmarke
------------	-------------------	----------

Will einer der gleichberechtigten Teilnehmer eine Nachricht absetzen, so fängt er das Frei-Token ein, wandelt es in ein Belegt-Token um und ergänzt es mit seiner adressierten Nachricht:

Startmarke	Zugriffs-kontrolle	Block-kontr.	Ziel-adr.	Quell-adr.	Infor-mation	Block-Prüfg.	Ende-marke	Block-status
1	1	1	6	6	...	4	1	1 Byte

Dieses Protokoll läuft zur Kontrolle mit vertauschter Ziel- und Quellenadresse zum Sender (Quelle) zurück. Dieser erzeugt bei fehlerfreier Übertragung wieder ein Frei-Token und gibt es weiter.

1.5.6 Ethernet-Telegramm

Ethernet ist das wohl älteste elektronische Informations-Netzwerk (1976, Metcalfe). Es wurde schließlich unter IEEE 802.3 (Institute of Electric and Electronic Engineers) genormt. Diese Norm wurde später von der ISO (International Standardisation Organisation) übernommen.

Ethernet ist weltweit verbreitet zur Vernetzung von Bürorechnern. Die Hardware dazu (Bridges, Gateways, Hubs ...) ist preiswert. Weder das stochastische Zugriffsverfahren (CSMA/CD) noch die große Datenmenge pro Telegramm (bis 1500 Byte) machen das Ethernet primär zu einem für die Automatisierungstechnik geeigneten Bus. Dennoch findet man ihn auch dort (vgl. Abschn. 4.3).

Trotz stetiger Weiterentwicklung hat das IEEE darauf geachtet, dass modifizierte Telegrammformate stets rückwärtskompatibel sind (Abb. 1.26). Die zugehörige Legende ist anschließend abgedruckt. Ohne die beiden Ergänzungen SNAP und Tagged MAC ist es das ursprüngliche, immer noch verwendete DIX-Format (DEC, Intel, Xerox).

Die Daten des Frames werden Byteweise übertragen, jeweils mit dem niederwertigsten Bit zuerst. Die Bytes innerhalb von breiteren Datenobjekten (2 Byte, 4 Byte) werden mit dem höchstwertigen Byte zuerst übertragen.

1.5.6.1 Die Felder des Ethernet-Telegrams

P Präambel (7 Byte). Diente zur Synchronisierung zwischen Sender und Empfänger. Bitmuster AAH = 10101010 1010 ... Heute nur noch aus Kompatibilitätsgründen vorhanden.

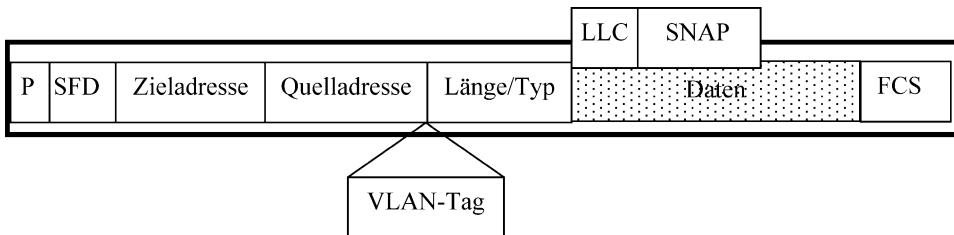


Abb. 1.26 Ethernet-Telegrammformat nach IEEE 802.3. Varianten: SNAP (802.3.3.1a), Tagged MAC Frame mit VLAN-Tag (802.3.3.1.b)

SFD	Start Frame Delimiter (1 Byte). Bitmuster D5h = 10101011 (lies 11011110).
Zieladresse	Ziel-MAC (Medium Access Control, 6 Byte). Adresse des Zielrechners.
Quelladresse	Quell-MAC (6 Byte). Senderadresse.
Länge/Typ	(2 Byte). Gibt die Länge des Datenblocks an. Heute wird stattdessen oft der Ethertype angegeben. Bei $n \leq 1500$ bedeutet n die Datenlänge, bei $n > 1500$ den Ethertype.
Ehtertype	beschreibt das Format, bzw. das Protokoll zur Interpretation des Datenblocks, z. B. 08 00h = Ethernet-Protokoll, Version 4. Die Ethertypes werden von der IEEE verwaltet.
Daten	Länge zwischen 64 Byte und 1500 Byte. Ist die Datenlänge $n < 64$ Byte, wird mit Bits aufgefüllt (Padding, wg. CSMA/CD).
FCS	Frame Check Sequence (4 Byte). Es wird eine CRC-Prüfzahl berechnet und übertragen. Der Empfänger macht die Gegenrechnung. Die Prüfung erstreckt sich über das gesamte Telegramm, ausgenommen Präambel und SFD.

Beim SNAP-Frame nach IEEE 802.3.3.1a wird im Datenblock nochmals die Ethertype-Information gegeben, damit übergeordnete ISO-OSI-Schichten korrekt arbeiten können.

- LLC** Logic Link Control (3 Byte) – bezieht sich auf den Data Link Layer (Schicht 2 des OSI-Modells). Im LLC-Block bedeuten:
 - DSAP** Destination Service Access Point (1 Byte). Zeigt auf den Pufferspeicher im Empfänger. Bitmuster AAh.
 - SSAP** Source Service Access Point (1 Byte). Zeigt auf den Pufferspeicher der Quelle. Bitmuster AAh.
 - C** Controlbyte. Bitmuster 03h.
- SNAP** Subnetwork Access Protocol (5 Byte).
 - Byte 1–3: Herstellercode, oder Bitmuster 00 00 00h.
 - Byte 4–5: Ethertype, z. B. 08 00h.

Beim Tagged MAC-Frame nach IEEE 802.3.3.1b (etikettiertes Medium Access Control Telegrammformat) wird ein VLAN-Tag eingeschoben. Damit kann ein virtuelles Netz adressiert werden ($n \leq 4096$ Tln.).

VLAN-Tag Virtuelles Local Area Network-Etikett (4 Byte).

Optionales Feld, das die Unterteilung des Netzes in logische Sub-Netze (Virtuelle LANs) ermöglicht. Die ersten 2 Byte (TPID – Tag Prefix Identifier) haben den festen Wert 0x8100, der das Vorhandensein des VLAN Tags anzeigt. Das VLAN Tag Feld enthält außerdem den TCI (Tag Control Identifier, 2 Byte). Darin 3 Bit Prioritätskennzeichnung, mit der geeignete Switches die Telegrammreihenfolgen entsprechend der Priorität verändern können. Weitere 12 Bit sind für maximal 4096 VLAN-Adressen vorgesehen.

1.5.6.2 TCP/IP-Protocol

Der in obiger Tabelle angegebene Datenblock 64...1500Byte beinhaltet noch einen 20 Byte großen Header des IP (Internet Protocol). Darin u. a.: Flags, Lifetime (< 256 s, damit ein Telegramm nicht ewig im Ethernet zirkuliert), Netzwerkadresse, Teilnehmeradresse, Header Check Sequence.

Die verbleibenden 1480 Byte Daten enthalten den 20 Byte großen Header des TCP (Transmission Control Protocol). Darin enthalten u. a.: Flags, Source Port (Senderadresse des Anwenderprogramms, Destination Port (Empfängeradresse des Anwenderprogramms), TCP Check Sequence.

Die verbleibenden 1460 Datenbyte können noch einen Anwender Header (UDP, User Data Protocol) enthalten, der allerdings nur 7 Byte lang ist.

Man erkennt aus Obigem, dass Ethernet ein System für große und komplexe Netzwerke ist mit großem, schnellem Datendurchsatz (primär Bürokommunikation).

1.6 Binäre Informationsdarstellung

Unabhängig vom Telegrammformat müssen die digitalen Zustände codiert werden. Die digitale Information kann in der Amplitude, in der Flanke, in der Phase und in der Frequenz enthalten sein.

Wenn eine synchrone Datenübertragung gewählt wurde, wie z. B. bei den Telegrammformaten HDLC und SDLC, ist es sinnvoll, eine Kodierung zu verwenden, die die Taktinformation enthält. Damit würde eine zusätzliche Takteleitung entfallen.

Ist die Kodierung frei von Gleichanteilen, ist eine Datenübertragung über Energieversorgungsleitung möglich, sofern diese mit Gleichgrößen vorgenommen wird.

Hinsichtlich dieser Kriterien sollen die folgenden Möglichkeiten der binären Informationsdarstellung untersucht werden.

1.6.1 NRZ, RZ

Das am häufigsten eingesetzte Modulationsverfahren ist das NRZ-Verfahren (Non Return to Zero), das in Abb. 1.27a beschrieben ist.

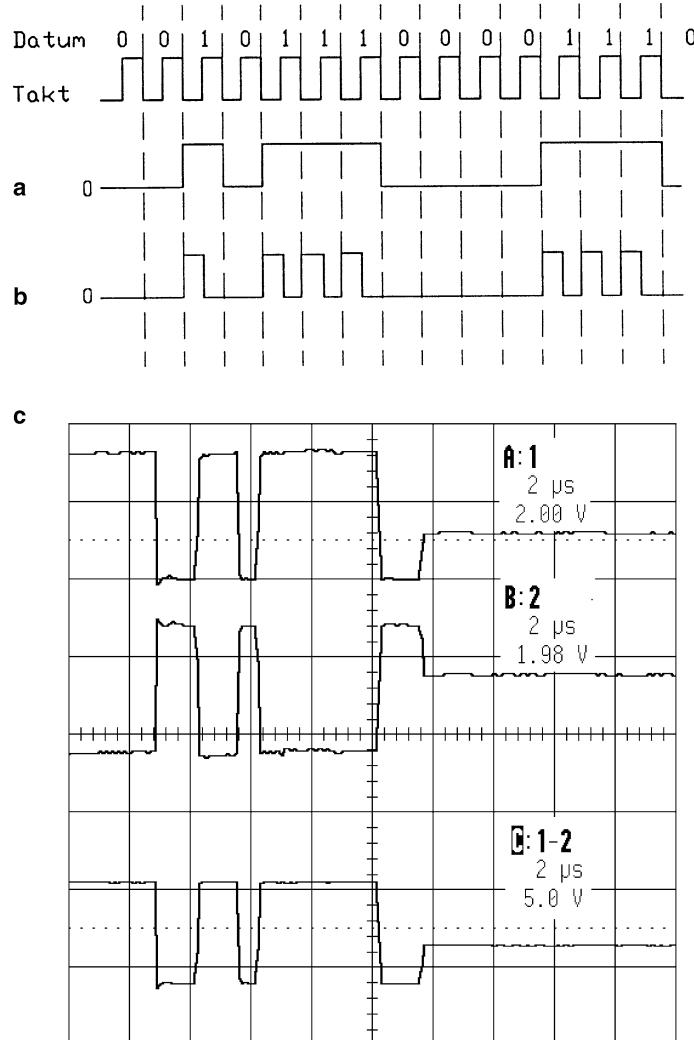


Abb. 1.27 Gleichstrombehaftete Kodierungen, ohne Takt. **a** NRZ, **b** RZ, **c** Oszillosgramm einer NRZ-Kodierung bei Profibus DP. *Oben: Bus-A, Mitte: Bus-B, unten: C = Differenzsignal. Profibus-typisch der Passiv-Pegel von –1 V des Terminierungsnetzwerkes beim Abschalten des Senders*

Die Signale logisch „0“ und „1“ werden durch unterschiedliche Amplituden dargestellt, wobei während des Taktes keine Potentialveränderung auftritt. Dabei sind folgende Darstellungsmöglichkeiten für die logischen Zustände denkbar:

- Masse und positives Potential,
- negatives Potential und Masse,
- negatives und positives Potential gleichen Betrages.

Für das NRZ-Verfahren gilt allgemein, dass die Taktinformation nicht enthalten ist und es nicht frei von Gleichanteilen ist, auch wenn mit positiven und negativen Potentialen gearbeitet wird. Dieses Verfahren wird z. B. bei PROFIBUS-DP angewendet. Es muss dabei berücksichtigt werden, dass ein rechteckförmiges Signal mit einer Frequenz übertragen wird, die der halben Übertragungsrate entspricht (z. B. 6 MHz bei PROFIBUS mit einer Übertragungsrate von 12 MHz bei 0–1 Folgen). Durch die Oberwellen treten aber weit höhere Frequenzen auf. Deshalb ist zu gewährleisten, dass die Übertragungsleitung immer ausreichend geschirmt ist oder dass gleich eine LWL-Übertragung gewählt wird.

Das RZ-Verfahren (Return to Zero) gemäß Abb. 1.27b unterscheidet sich vom NRZ-Verfahren nur dadurch, dass der Pegel, wenn eine „1“ gesendet wird, während des Taktes in seinen Ausgangszustand zurückkehrt. Auch hier gilt, dass die Taktinformation nicht enthalten ist, da bei langen „0“-Folgen keine Impulse übertragen werden.

1.6.2 Bipolar-Kodierung, HDB_n-Kodierung

Um die Informationsdarstellung des NRZ-Formates gleichanteilfrei zu gestalten, kann man die logische „1“ mit Potentialen abwechselnder Polarität darstellen. Es ergibt sich daraus eine Kodierung nach Abb. 1.28a.

Durch die alternierenden Polaritäten ist diese Kodierung frei von Gleichanteilen, jedoch ist die Taktinformation nicht enthalten.

Das Einfügen der Taktinformation kann dadurch erreicht werden, dass nach n aufeinander folgenden logischen „0“ ein Verletzungsimpuls gesendet wird. Damit erhält man das HDB_n-Format (*High Density Bipolar Format*). In Abb. 1.28b ist das HDB₂-Verfahren dargestellt.

Der Verletzungsimpuls wird dadurch gekennzeichnet, dass er die gleiche Polarität wie die zuletzt gesendete „1“ aufweist. Damit ist gewährleistet, dass spätestens nach n Takten ein Impuls gesendet wird und somit eine Taktinformation reproduziert werden kann. Es kann nun passieren, dass die Verletzungsimpulse alle die gleiche Polarität aufweisen. Damit ist die Kodierung nicht mehr gleichanteilfrei.

Fügt man so genannte Ausgleichsimpulse ein, kann auch die Gleichanteilfreiheit wieder hergestellt werden. Wird nach dem letzten Verletzungsimpuls eine gerade Anzahl von „1“ gesendet und ein neuer Verletzungsimpuls notwendig, wird ein Ausgleichsimpuls mit umgekehrter Polarität zur letzten „1“ gesendet und ein Verletzungsimpuls mit gleicher

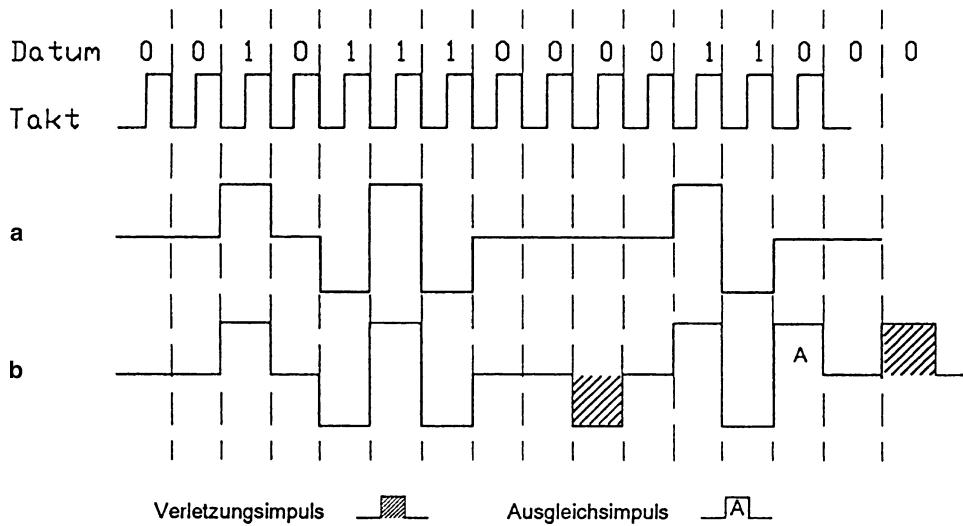


Abb. 1.28 Gleichstromfreie Kodierungen. **a** Bipolar-Kodierung, ohne Takt, **b** HDB₂-Kodierung, mit Takt

Polarität wie der Ausgleichsimpuls. Dabei muss der Abstand zwischen Ausgleichs- und Verletzungsimpuls kleiner als n Takte sein. Damit ist ein Ausgleichsimpuls eindeutig gekennzeichnet.

1.6.3 NRZI

NRZI bedeutet *Non Return to Zero Insert*. Bei diesem Verfahren wird die logische „0“ dadurch dargestellt, dass zu Beginn des Taktes die Polarität wechselt. Dieser Polaritätswechsel bleibt bei logisch „1“ aus (Abb. 1.29).

Diese Kodierung ist nicht gleichanteilfrei. Wenn, wie z. B. beim Bitbus, nach einer definierten Anzahl von gesendeten „1“ automatisch vom Sender eine „0“ eingefügt wird, ist

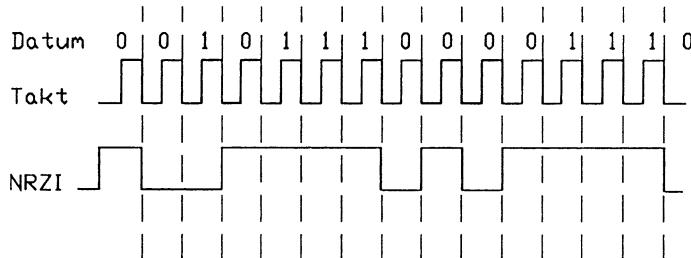


Abb. 1.29 Gleichstrombehaftete Kodierung NRZI

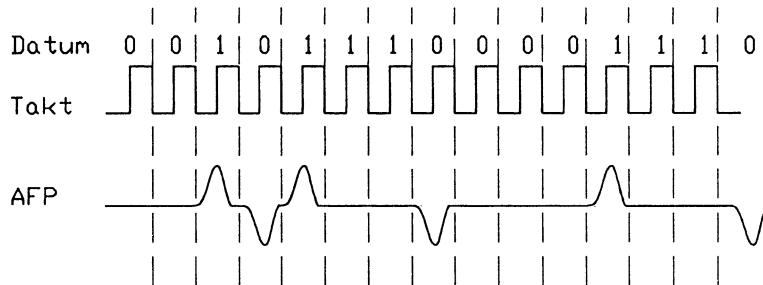


Abb. 1.30 Alternierendes Flankenpulsverfahren AFP

die Taktinformation enthalten. Der Empfänger muss diese zusätzliche „0“ wieder entfernen, um die eigentliche Information zurückzugewinnen.

1.6.4 AFP

Das Alternierende-Flanken-Puls-Verfahren stellt einen Wechsel des logischen Zustandes durch einen Impuls dar. Wenn, wie in Abb. 1.30, die Polarität der Impulse abwechselt, ist das Signal frei von Gleichanteilen. Eine Taktinformation ist, wenn lange Folgen des gleichen logischen Zustandes vorkommen, nicht enthalten. Der große Vorteil dieser Kodierung liegt in der geringen Störabstrahlung, wenn man als Impulsform einen \sin^2 -Impuls verwendet.

1.6.5 Manchester-II-Kodierung

Bei der Manchester-II-Kodierung ist die Information in der Phasenlage des Signales enthalten, d. h. tritt in der Taktmitte eine positive Flanke auf, handelt es sich um eine „0“, ist die Flanke negativ, handelt es sich um eine „1“ (Abb. 1.31).

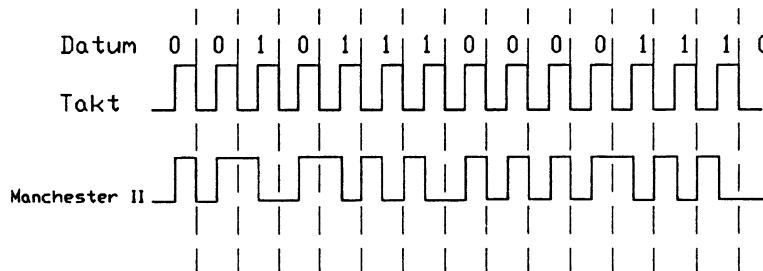


Abb. 1.31 Manchester-II-Kodierung

Wird ein bipolares Signal verwendet, ist dieses frei von Gleichanteilen. Dadurch, dass in der Taktmitte immer eine Flanke auftreten muss, ist die Taktinformation ebenfalls enthalten. Problematisch ist, dass die maximal auftretende Frequenz gleich der Taktfrequenz bzw. doppelt so hoch wie die maximal auftretende Frequenz beim NRZ-Verfahren ist. Dadurch ist die Störabstrahlung relativ hoch.

Die Manchester-II-Kodierung wird bei vielen synchron arbeitenden Bussystemen verwendet (z. B. PROFIBUS-PA, der so genannte H1-Bus des FOUNDATION Fieldbus). AS-Interface verwendet eine Mischung aus dem Manchester-II- und dem AFP-Verfahren. Da AS-Interface ohne Schrimung arbeitet, muss auf eine geringe Störabstrahlung geachtet werden. Beim AS-Interface-Signal werden alle Flanken des Manchester-II-Signals mittels eines \sin^2 -Impulses übertragen. Dadurch wird erreicht, dass die Taktinformation mit übertragen wird, dass das Signal frei von Gleichanteilen ist und es praktisch keine Oberwellen erzeugt.

1.6.6 FSK, ASK, PSK

Die bisher behandelten Kodierungsmöglichkeiten verwenden, bis auf das AFP-Verfahren, Rechtecksignale. Bei den hier angesprochenen Verfahren werden sinusförmige Signale verwendet. Dies hat den Vorteil, dass die Signale keine Oberwellen haben und damit in ihrer Bandbreite begrenzt sind.

Bei dem FSK-Verfahren (*Frequency Shift Keying*) werden „0“ und „1“ durch zwei unterschiedliche Frequenzen dargestellt. Diese Übertragungsart wird auch Carrierbandübertragung genannt.

Das ASK-Verfahren (*Amplitude Shift Keying*) unterscheidet sich vom FSK-Verfahren dadurch, dass einer der beiden logischen Zustände durch die Frequenz $f = 0 \text{ Hz}$ dargestellt wird. Das bedeutet, dass ein logischer Zustand durch eine vorhandene, der andere logische Zustand durch eine fehlende Amplitude dargestellt wird.

Beim PSK-Verfahren (*Phase Shift Keying*) wird eine „1“ durch einen Phasensprung am Taktbeginn und eine „0“ durch einen fehlenden Phasensprung dargestellt.

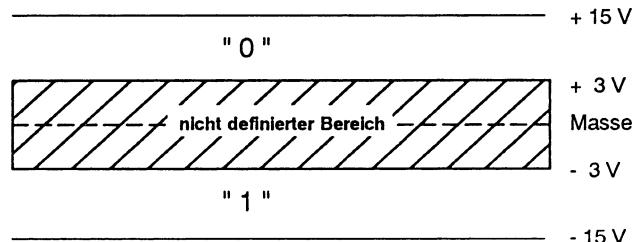
Durch das sinusförmige Signal sind alle drei Möglichkeiten gleichanteilfrei. Die Taktinformation ist beim FSK- und PSK-Verfahren enthalten, da es einen Zusammenhang zwischen Übertragungsfrequenz und Takt geben muss. Beim ASK-Verfahren geht die Taktinformation verloren, wenn lange Folgen entstehen, in denen kein Signal gesendet wird.

1.7 Übertragungsstandards

1.7.1 RS 232-, V.24-Schnittstelle

Für diese Schnittstelle existieren u. a. eine amerikanische (RS 232 C = EIA 232), eine internationale (CCITT V.24) und eine deutsche Norm (DIN 66 020). Dabei steht RS für Recommended Standard und CCITT für Comité Consultatif Télégraphique et Téléphonique.

Abb. 1.32 Spannungspegel der RS 232-Schnittstelle



Bei der RS 232-Schnittstelle handelt es sich um eine erdunsymmetrische Schnittstelle für die Kommunikation zwischen zwei Datenendgeräten (DEE) oder zwischen einem Datenendgerät und einer Datenübertragungseinrichtung (DÜE). Unter einem DEE versteht man z. B. einen PC, Drucker etc., während ein Modem ein Beispiel für ein DÜE ist. Im Folgenden soll nur noch die Kommunikation zwischen zwei DEE betrachtet werden.

Die RS 232C ist ausschließlich für Punkt-zu-Punkt-Verbindungen geeignet. Erdunsymmetrisch heißt, dass der Signalpegel zwischen der Datenleitung und Masse gemessen wird und damit nicht symmetrisch zur Masse sein kann.

Die Pegel sind wie folgt definiert (Abb. 1.32):

$$\text{Logisch „0“: } 3 \text{ V} < U < 15 \text{ V}$$

$$\text{Logisch „1“: } -15 \text{ V} < U < -3 \text{ V}$$

Daraus kann man sehen, dass der Bereich zwischen -3 V und 3 V nicht definiert ist. Dieser Bereich muss bei einem Wechsel des logischen Zustandes so schnell wie möglich durchlaufen werden. Hierzu wird in der DIN 66 259 Teil 1 festgelegt, dass die Flankensteilheit minimal 6 V/ms oder 3% der Schrittdauer betragen darf (es zählt immer der kleinere Wert). Damit sind die Kapazitäten des Senders, des Empfängers und der Leitung begrenzt.

Wie schon erwähnt, hat die Leitungslänge einen Einfluss auf die maximale Übertragungsrate. Dazu werden folgende Richtwerte angegeben:

Übertragungsrate	Leitungslänge
1200 Bd	900 m
19 200 Bd	50 m

Die genauen Werte sind der DIN 66 259 Teil 2 zu entnehmen.

Die RS 232C-Schnittstelle ist vollduplexfähig und man kann verschiedene Übertragungsverfahren mit ihr realisieren.

Im einfachsten Fall werden die Leitungen TxD (*Transmit Data*) und RxD (*Receive Data*) des Senders und Empfängers gekreuzt und zusätzlich eine Masseverbindung (GND) hergestellt. Bei dieser Minimalkonfiguration muss die Software die Sicherheit der elektrischen Datenübertragung gewährleisten (Software-Handshake).

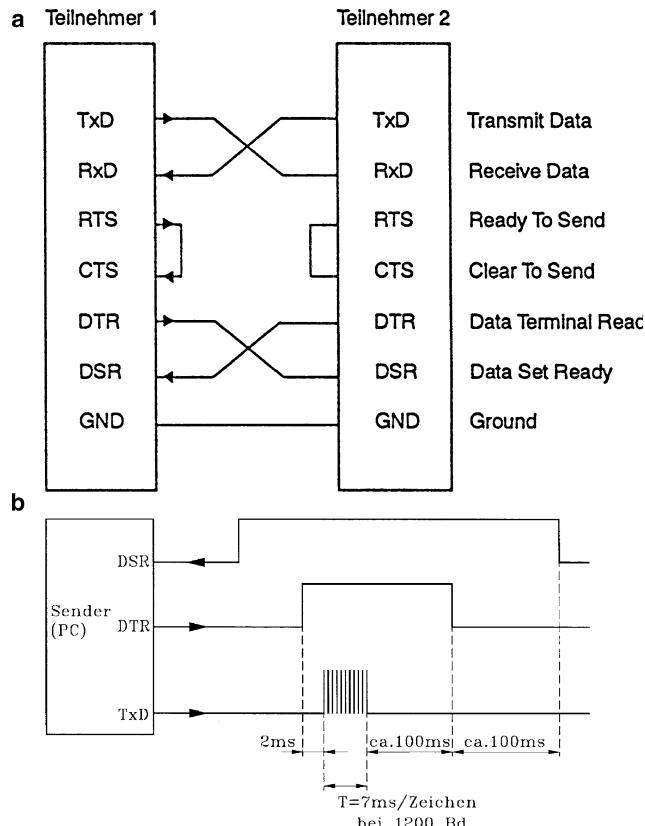


Abb. 1.33 Schaltungsbeispiel EIA232-Schnittstelle. **a** mögliche Verbindungen, **b** Zeitdiagramm

Diese Sicherheit kann durch die Schnittstelle selbst realisiert werden, wenn der so genannte Handshake-Betrieb gewählt wird (siehe Abb. 1.33). Dazu existieren die Meldeleitungen

CTS (*Clear To Send*) bzw.

DSR (*Data Set Ready*)

und die Steuerleitungen

RTS (*Ready To Send*) bzw.

DTR (*Data Terminal Ready*).

Die Meldeleitung des Senders signalisiert, dass Daten übertragen werden sollen. Dieses Signal wird auf die Steuerleitung des Empfängers gelegt, der damit empfangsbereit ist.

Ein Beispiel der möglichen Beschaltungen zeigt Abb. 1.33a. Es muss bei der Verwendung der Handshake-Leitungen DTR und DSR darauf geachtet werden, dass die Leitungen RTS und CTS gebrückt sind. Bei den in Abb. 1.33b angegebenen Zeiten handelt es sich um gemessene Werte an einer Applikation.

Ein Vorteil dieser Schnittstelle liegt darin, dass sie mit geringen Sendeleistungen auskommt. Dies resultiert daraus, dass der Empfänger einen Eingangswiderstand im Bereich von 3 bis 7 kΩ aufweisen muss.

Die Leerlaufspannung des Senders soll kleiner als 25 V und der Kurzschlussstrom kleiner als 0,5 A sein. Daraus ergibt sich ein minimaler Innenwiderstand des Senders von 50 Ω.

1.7.2 RS 422-Schnittstelle

Die RS 422 ist eine erdsymmetrische Schnittstelle für Punkt-zu-Punkt-Verbindungen. Die logischen Zustände werden durch eine Differenzspannung zwischen zwei Leitungen dargestellt. Bei Verwendung einer zweiadriigen Leitung ist die Schnittstelle halbduplexfähig, bei Verwendung von vieradriigen Verbindungsleitungen vollduplexfähig. Beschrieben wird die RS 422-Schnittstelle in der EIA 422, der CCITT V 1 1 und der DIN 66 259 Teil 4. Die elektrischen Spezifikationen sind identisch mit der RS 485 und werden dort beschrieben. Das gleiche gilt für die Übertragungsraten und zulässigen Leitungslängen.

1.7.3 RS 485-Schnittstelle

Der wichtigste Unterschied zwischen der RS 485- und der RS 422-Schnittstelle ist, dass die RS 485 für Mehrpunktverbindungen geeignet ist. Damit müssen zusätzlich zu den Spezifikationen, die auch für die RS 422 gelten, Maßnahmen festgelegt werden, die den Konkurrenzbetrieb mehrerer Sender berücksichtigen.

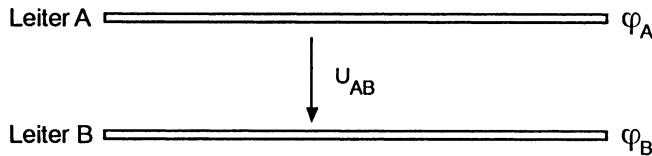
Die RS 485 wird in der EIA 485 und in der ISO 8482 beschrieben. Beide Normen unterscheiden sich in einigen Punkten voneinander, was im Folgenden an entsprechender Stelle aufgezeigt werden soll.

Bei den Festlegungen wird unterschieden zwischen den Sender- und Empfängerspezifikationen. Dabei muss ein Sender bei maximaler Belastung eine Differenzspannung von $1,5 \text{ V} \leq U_{AB} \leq 5 \text{ V}$ liefern. Der Zustand logisch „0“ wird durch eine positive Spannung U_{AB} und logisch „1“ durch eine negative Spannung dargestellt (Abb. 1.34).

Wie Abb. 1.34 zeigt, würde eine induktiv oder kapazitiv auf beide Leiter A und B eingestraute Störspannung U_S nicht beeinflussen:

$$\begin{aligned} U_{AB} &= \varphi_A + U_S - (\varphi_B + U_S) \\ &= \varphi_A - \varphi_B . \end{aligned}$$

Dies begründet die Störfestigkeit der Übertragung nach RS 485 (und RS 422).



	Sender	Empfänger	
		EIA 485	ISO 8482
"0"	$1,5 \text{ V} \leq U_{AB} \leq 5 \text{ V}$	$U_{AB} > 0,2 \text{ V}$	$U_{AB} > 0,3 \text{ V}$
"1"	$-5 \text{ V} \leq U_{AB} \leq -1,5 \text{ V}$	$U_{AB} < -0,2 \text{ V}$	$U_{AB} < -0,3 \text{ V}$

Abb. 1.34 Spannungspegel der RS 485-Schnittstelle

Volle Belastung des Senders bedeutet, dass der Sender mit 54Ω abgeschlossen wurde. Dieser Wert resultiert aus den notwendigen Abschlusswiderständen der Leitung und der Parallelschaltung von 32 Teilnehmern. Ein Teilnehmer, in der Norm Unit load (U1) genannt, ist über eine Strom-Spannungskennlinie definiert. Weist ein angeschaltetes Gerät eine von der Norm abweichende Kennlinie auf, ist der berechenbare Wert U1 von 1 verschieden. Damit kann man erreichen, dass mehr als 32 Geräte an die RS 485 angeschlossen werden können.

Die Leerlaufspannung des Senders soll kleiner als 6 V sein. Der Kurzschlussstrom ist mit maximal 250 mA relativ hoch angesetzt, weil bei gleichzeitigem Senden zweier Teilnehmer die ohmsche Belastung jedes Senders weit unterhalb von 54Ω liegt.

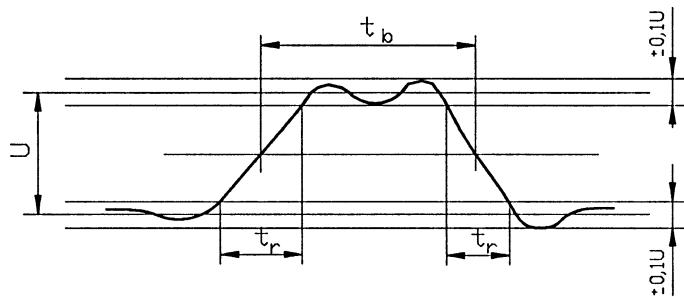
Die ISO 8482 empfiehlt als Kabeltyp eine verdrillte Zweidrahtleitung, während die EIA 485 über den Kabeltyp keine Angaben macht. Die Leitungen müssen an beiden Enden mit ihrem Wellenwiderstand abgeschlossen werden, was in der Praxis die größte Belastung für den Sender darstellt.

Die maximale Leitungslänge wird von der ISO 8482 mit 500 m angegeben, wobei Stichleitungen bis zu einer Länge von 5 m zulässig sind. In der EIA 485 wird folgende Gleichung für die Berechnung des maximalen Leitungsschleifenwiderstandes angegeben:

$$R_{\text{Leitung}} = \frac{R_W \cdot (1,5 \text{ V} - U_{\min})}{U_{\min}} .$$

Dabei ist R_W der Wellenwiderstand der Leitung und U_{\min} die im worst case am Empfänger anliegende minimale Signalspannung. Die Stichleitungen sollen so kurz wie möglich sein.

Die Empfänger müssen so ausgelegt sein, dass Spannungen von $\pm 0,3 \text{ V}$ (ISO 8482), bzw. $\pm 0,2 \text{ V}$ (EIA 485) als Signalspannung noch erfasst werden können. Das Potential eines einzelnen Leiters bezogen auf Masse muss in dem Intervall von -7 V bis 12 V liegen. Die Signalpegel der ISO 4882 orientieren sich an der RS 422-Schnittstelle.



- t_r = Anstiegszeit
- t_b = Zeitintervall für die Übertragung eines Bit
- $t_s \leq 0,3 \cdot t_b$
- U = Differenzspannung zwischen den Datenleitungen

Abb. 1.35 Signalmuster bei RS 485 (Generator)

Die ISO 8482 empfiehlt eine maximale Übertragungsrate von 1 Mbaud, die EIA 485 lässt 10 MBd zu. Die Abhängigkeiten zwischen Leitungslänge und Übertragungsrate lassen sich aus der Signalform, die in Abb. 1.35 dargestellt ist, ableiten.

Ist die Bedingung $t_s \leq 0,3 \cdot t_b$ wegen der Leitungsinduktivitäten, -kapazitäten und der Eingangsimpedanzen der Teilnehmer nicht realisierbar, ist die Übertragungsrate zu reduzieren.

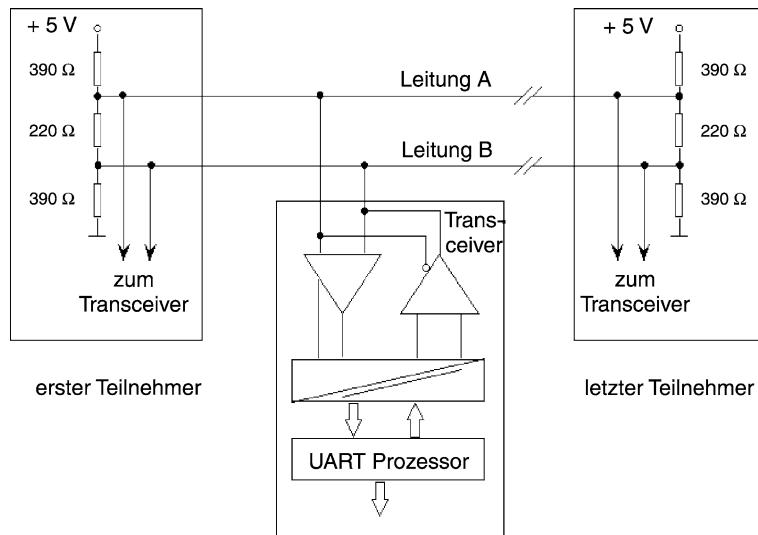


Abb. 1.36 PROFIBUS-Anschaltung

Für die Übertragungsraten gelten folgende Richtwerte:

Übertragungsraten	Leitungslänge
93,75 kBd	1200 m
500 kBd	400 m

In der Praxis kann z. B. der PROFIBUS mit einer RS 485-Schnittstelle betrieben werden. Eine entsprechende Anschaltung zeigt Abb. 1.36.

1.7.4 20 mA-Stromschleife

Diese Schnittstelle ist in der DIN 66 258 Teil 1 beschrieben und ist nur für Zweipunktverbindungen ausgelegt. Die logischen Zustände sind wie folgt definiert:

Logisch „0“: $0 \text{ mA} \leq I \leq 3 \text{ mA}$
 Logisch „1“: $14 \text{ mA} \leq I \leq 20 \text{ mA}$

Die Toleranzen sind nicht zwingend vorgeschrieben und können vom Anwender auch anders gewählt werden.

Es ist festgelegt, dass nur einer der beiden Teilnehmer Strom in die Datenleitung einspeisen darf. Als maximale Leitungslänge werden 1000 m empfohlen. Die Datenübertragungsrate liegt bei maximal 9600 Bd. Das Zeichenformat ist gemäß Abb. 1.37 festgelegt.

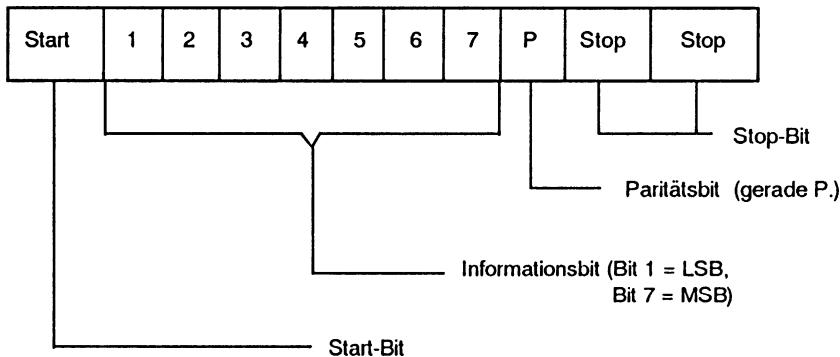


Abb. 1.37 Telegrammformat der 20-mA-Schnittstelle

1.7.5 IEC 61158-2, FISCO-Modell

Die IEC 61158 beschreibt eine Vielzahl unterschiedlicher Bussysteme (u. a. PROFIBUS, FOUNDATION Fieldbus, InterBus, ...). Im Abschnitt 2 (IEC 61158-2) werden dafür verschiedene physikalische Schnittstellen definiert. Die folgenden Tabellen zeigen eine Auswahl möglicher Optionen:

H1-Bus			
Übertragungsrate	31,25 kBd	31,25 kBd	31,25 kBd
Schnittstelle	Spannung	Spannung	Spannung
Topologie	Linie/Baum	Linie/Baum	Linie/Baum
Busspeisung	Nein	Ja	Ja
Eigensicher	Nein	Ja	Nein
Max. Teilnehmerzahl	32	32	32
Max. Leitungslänge	1900 m	1900 m	1900 m
Max. Stichleitungslänge	120 m	120 m	120 m

H2-Bus			
Übertragungsrate	1 MBd	1 MBd	1 MBd
Schnittstelle	Spannung	Strom	Spannung
Topologie	Linie	Linie	Linie
Busspeisung	Nein	Ja	Nein
Eigensicher	Nein	Ja	Nein
Max. Teilnehmerzahl	32	32	32
Max. Leitungslänge	750 m	750 m	750 m

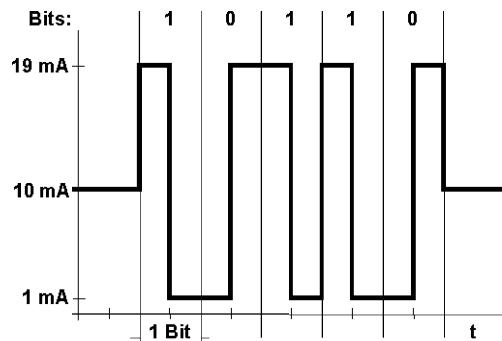
Weiterhin wird in der IEC 61158-2 eine Schnittstelle mit einer Übertragungsrate von 2,5 MBd beschrieben. Im Folgenden wird auf die Schnittstelle eingegangen, die in der H1-Bus-Tabelle in der mittleren Spalte beschrieben ist.

Es handelt es sich hierbei um eine Physik, die sowohl für eigensichere als auch nicht eigensichere Anwendungen im Bereich der Prozessautomatisierung zum Einsatz kommt. Ob diese Schnittstelle eigensicher ist oder nicht, hängt vom verwendeten Speisegerät ab. Real zum Einsatz kommt diese Schnittstelle beim PROFIBUS-PA, dort wird das Speisegerät Segmentkoppler genannt (siehe Abschn. 1.9.2) und beim FOUNDATION Fieldbus (FF) als so genannter H1-Bus.

Grundlage dieser Physik ist die Manchester II-Modulation (siehe Abschn. 1.6.5). Das Datensignal ist spannungsmoduliert. Das modulierte Spannungssignal (450 mV_S) wird über einem 50Ω Widerstand abgegriffen, sodass sich die allgemein bekannte Darstellungsart mit einem Signalstrom von $\pm 9 \text{ mA}$ ergibt (Abb. 1.38).

Über diese Schnittstelle werden die Teilnehmer, parallel zur Datenübertragung, mit Energie versorgt. Eine zusätzliche Speisung der Feldgeräte für den Fall, dass die Energieversorgung über die Busleitung nicht ausreicht, ist möglich.

Abb. 1.38 Signal gemäß IEC 61158-2



Jeder Teilnehmer soll einen Versorgungsstrom $I \geq 10\text{ mA}$ aus der Übertragungsleitung entnehmen. Dieser dient der Fernspeisung der Feldgeräte. In der Praxis benötigen die Feldgeräte häufig einen Versorgungsstrom, der größer als 10 mA ist. Dies hat zu Folge, dass die Anzahl der betreibbaren Feldgeräte an einem Segment sinkt. Die IEC 61158-2 sieht vor, dass 32 Teilnehmer an einem Segment betrieben werden können. Dies gilt allerdings nicht, wenn die Schnittstelle eigensicher betrieben wird. Für eigensichere Anwendungen können aus Gründen des Explosionsschutzes maximal 10 Teilnehmer angeschlossen werden. Werden mehrere Teilnehmer an der Busleitung betrieben, addieren sich die Versorgungsströme und stellen die Gleichstromkomponente dar. Diesem Gleichstrom ist das Datensignal in Form eines des oben erläuterten Wechselstroms mit einer Amplitude von $\pm 9\text{ mA}$ überlagert.

Die Übertragungsrate beträgt $31,25\text{ kBd}$. Dies ist für die meisten Anwendungen der Prozessautomation ausreichend schnell. Für nicht Ex-Anwendungen beträgt die maximale Leitungslänge 1900 m . Auch hier müssen bei Ex-Anwendungen aufgrund des Explosionsschutzes durch Eigensicherheit Reduzierungen in Kauf genommen werden (maximal 1000 m für EEx ia IIC Anwendungen).

Weiterhin legt die IEC 61158-2 fest, dass die minimale Eingangsspannung eines Feldgerätes 9 V betragen muss. Dies führt zu folgendem Problem (Abb. 1.39):

Das Speisegerät stellt hinsichtlich der Energieversorgung eine Spannungsquelle dar. Diese Quelle liefert, je nach Typ, zwischen $12,6\text{ V}$ und 32 V . Der Strom I_1 ist der Versor-

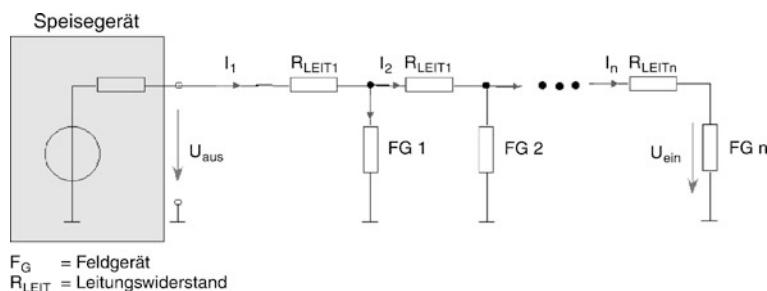


Abb. 1.39 Leitungslängenproblematik

gungsstrom für alle Feldgeräte (FG). Der Versorgungsstrom über die Busleitung reduziert sich nach jedem Teilnehmer um den Stromwert, den der Teilnehmer benötigt. Es gilt:

$$I_2 = I_1 - I_{\text{FG}1}$$

Die Leitungswiderstände verursachen einen Spannungsabfall, der speziell dann, wenn der erste Teilnehmer weit vom Speisegerät entfernt ist, dazu führen kann, dass die Eingangsspannung am letzten Feldgerät die geforderten 9 V nicht mehr erreicht. Eine Hilfe zur Bestimmung der maximalen Leitungslänge wird später gegeben, wenn das FISCO-Modell besprochen wurde.

Gemäß IEC 61158-2 ist die Übertragungsleitung mit einem resistiven Busabschluss zu versehen, der eine kapazitive DC-Entkopplung enthält. Dieser Busabschluss ist in den Speisegeräten, die immer den Anfang einer solchen Schnittstelle darstellen, integriert. Für den Abschluss im Feld stehen vorkonfektionierte Lösungen zur Verfügung.

Weiterhin werden folgende Kabelkennwerte empfohlen.

	Typ A	Typ B
Kabelstruktur	Paarweise verdrillt und geschirmt	Ein oder mehrere verdrillte Kabel, gemeinsamer Schirm
Leitungsquerschnitt	0,8 mm ²	0,32 mm ²
Schleifenwiderstand (DC)	44 Ω/km	112 Ω/km
Impedanz bei 31,25 kHz	100 Ω ± 20 %	100 Ω ± 30 %
Dämpfung bei 39 kHz	3 dB/km	5 dB/Km
Kapazitive Unsymmetrie	2 nF/km	2 nF/km
Abdeckungsgrad des Schirms	90 %	–
Max. Leitungslänge	1900 m	1200 m

Da in der Prozessautomation das Problem besteht, dass Feldbusssysteme in explosionsgefährdeten Bereichen eingesetzt werden sollen, wurde die Physikalisch Technische Bundesanstalt (PTB) damit beauftragt Untersuchungen zur Eigensicherheit von Feldbusssystemen durchzuführen. Die Ergebnisse dieser Untersuchung sind im PTB-Bericht „PTB-W-53“ veröffentlicht. Ergebnis dieser Untersuchung war das FISCO-Modell.

Dieses FISCO-Modell basiert auf der eben besprochenen Physik gemäß IEC 61158-2.

Das Problem bei Feldbusssystemen besteht darin, dass an ein Übertragungsmedium mehr als ein Teilnehmer angeschlossen wird. Dadurch wird der so genannte Nachweis der Eigensicherheit, der für jeden eigensicheren Stromkreis zu erstellen ist, relativ aufwändig und kompliziert. So müssten beispielsweise alle denkbaren Zusammenschaltungen von Induktivitäten und Kapazitäten betrachtet werden.

Die Basis für die Untersuchung waren die folgenden Bedingungen:

- Alle anderen Geräte (die Feldgeräte) verhalten sich als passive Stromsenken
- Jeder Teilnehmer nimmt einen konstanten Grundstrom (Gleichstrom) auf
- Es gibt an einem eigensicheren Feldbus genau ein aktives Gerät (das Speisegerät)

- Die wirksamen inneren Induktivitäten und Kapazitäten sind vernachlässigbar klein
- Leitungsabschlüsse an den beiden Enden der Hauptleitung

Auf Grundlage dieser Basis wurde das Zündverhalten von unterschiedlichen Leitungen/Leitungslängen mit unterschiedlichen Prüfstromkreisen (EEx ia IIC, EEx ib IIC/IIB) und der Einfluss der Speisespannung untersucht. Das Ergebnis dieser Untersuchung ist das FISCO-Modell.

Zusammengefasst lässt sich sagen, dass ein Feldbussystem eigensicher ist, wenn *alle* Komponenten gemäß dem FISCO-Modell zertifiziert sind. Dies bedeutet im Einzelnen:

- a) Grundsätzliche Forderungen der Eigensicherheit
 - Für jedes Feldgerät muss die maximale zulässige Eingangsspannung U_i größer oder gleich sein als die maximale Ausgangsspannung U_0 des Speisegerätes. ($U_i \geq U_0$)
 - Für jedes Feldgerät muss der maximale zulässige Eingangsstrom I_i größer oder gleich sein als der maximale Ausgangsstrom I_0 des Speisegerätes ($I_i \geq I_0$)
 - Für jedes Feldgerät muss die maximale zulässige Eingangsleistung P_i größer sein als die maximale Ausgangsleistung P des Speisegerätes ($P_i P_0$).
- b) Zusätzliche Forderungen gemäß FISCO-Modell
 - Die inneren Induktivitäten und Kapazitäten der Feldgeräte müssen vernachlässigbar klein sein, d. h. $L_i \leq 10 \mu\text{H}$; $C_i \leq 5 \text{nF}$.
 - Die Kabelparameter entsprechen denen im PTB-Bericht-W53 veröffentlichten Werten. Hierbei ist zu beachten, dass nur mit dem Kabeltyp A die maximale Leitungslänge von 1000 m erreicht werden kann.

Für die Speisegeräte gelten für den Normalbetrieb Beschränkungen hinsichtlich der Ausgangsspannung bzw. des Ausgangsstroms. Diese Werte sind nicht zu verwechseln mit den oben genannten Maximalwerten. So liefert z. B. der Segmentkoppler für den PROFIBUS-PA mit eigensicherer Schnittstelle von Pepperl+Fuchs im Normalbetrieb eine Ausgangsspannung zwischen 12,6 V ... 13,4 V und einen Ausgangsstrom von 110 mA.

Dies führt bei eigensicheren Anwendungen dazu, dass die Leitungslängenproblematik sich dadurch verschärft, dass die Ausgangsspannung klein ist. So darf der maximale Spannungsabfall über der Übertragungsleitung unter „Worst-case“-Bedingungen 3,6 V betragen

Das FISCO-Modell empfiehlt, ebenfalls dafür Sorge zu tragen, dass ein fehlerhafter Teilnehmer den Bus nicht dadurch lahm legt, dass er einen zu hohen Strom aus der Übertragungsleitung zieht. Dazu wurde die so genannte Fault Disconnection Electronic (FDE) eingeführt. Dabei handelt es sich um eine elektronische Strombegrenzung, die gewährleistet, dass der aus der Übertragungsleitung entnommene Strom einen bestimmten Wert (I_{FDE} maximal 9 mA zusätzlich zum Versorgungsstrom) nicht überschreitet.

Durch die Strombegrenzung reduziert sich die Anzahl der anschließbaren Teilnehmer. Dies soll an einem Beispiel verdeutlicht werden:

Ein PROFIBUS-PA Segmentkoppler liefert z. B. einen maximalen Ausgangsstrom von $I_{\text{ges}} = 110 \text{ mA}$. Die Gesamtstromaufnahme des Feldbusses bestimmt sich wie folgt:

$$I_{\text{ges}} = I_{\text{signal}} + I_{\text{FDE}} + \sum_{i=1}^n I_i ,$$

woraus folgt:

$$\sum_{i=1}^n I_i = I_{\text{ges}} - I_{\text{signal}} - I_{\text{FDE}} = (110 - 9 - 9) \text{ mA} = 92 \text{ mA} .$$

Das bedeutet, wenn ein Segmentkoppler 110 mA liefert, stehen für die Versorgung der Teilnehmer nur 92 mA zur Verfügung. Da viele Busteilnehmer mit einem I_{FDE} kleiner als 9 mA arbeiten, kann dieser Wert auch etwas höher sein. Da jeder Teilnehmer einen Versorgungsstrom von mindestens 10 mA aus der Busleitung entnehmen muss, können in diesem Beispiel 9 Teilnehmer an diesem eigensicheren Strang betrieben werden. Da in der Praxis viele feldbusfähige Geräte einen Strom entnehmen, der größer als 10 mA ist, reduziert sich dadurch die Teilnehmerzahl nochmals. Weiterhin kommt es in der Praxis häufig vor, dass die Anbieter von Segmentkopplern nur den zur Verfügung stehenden Versorgungsstrom I_i angeben. Das bedeutet, dass man die Größen I_{Signal} und I_{FDE} für die Bestimmung der maximalen Teilnehmerzahl nicht zu berücksichtigen braucht. Ein typischer Wert für die maximale Anzahl von Teilnehmern ist 10.

Als Letztes muss man sich Gedanken darüber machen, wie lang die Busleitung tatsächlich sein darf. Wie schon erwähnt stellt hier der Spannungsabfall entlang der Übertragungsleitung ein Problem dar. Um das Problem zu verdeutlichen wird im folgenden Beispiel eine Punkt-zu-Punkt-Verbindung zwischen Segmentkoppler und Feldgerät betrachtet. Es gilt dann gemäß Abb. 1.39 (Leitungslängenproblematik):

Aus

$$\begin{aligned} U_{\text{Leitung}} &= U_{\text{aus}} - U_{\text{ein}} \\ R_{\text{Leitung}} &= (U_{\text{aus}} - U_{\text{ein}}) \div I_1 \\ R' \cdot l_{\max} &= (U_{\text{aus}} - U_{\text{ein}}) \div I_1 \quad \text{folgt:} \\ l_{\max} &= (U_{\text{aus}} - U_{\text{ein}}) \div (i_1 \cdot R') . \end{aligned}$$

Dabei ist

- U_{Leitung} = Spannungsabfall entlang der Leitung,
- U_{aus} = Ausgangsspannung des Segmentkopplers,
- U_{ein} = Eingangsspannung des Feldgerätes,
- I_1 = Grundstrom des Feldgerätes,
- R' = Widerstandsbelag der Übertragungsleitung [Ω/km],
- l_{\max} = maximale Leitungslänge.

Wird nun ein Kabel gemäß IEC 61158-2 mit einem Widerstandsbelag von $44 \Omega/\text{km}$ verwendet, die Ausgangsspannung des Segmentkopplers beträgt unter „Worst-case“-Bedingungen 12,6 V und der Grundstrom des Feldgerätes beträgt 10 mA, ergibt sich aus der obigen Gleichung eine maximale Leitungslänge von 8,18 km, was aufgrund der Forderungen hinsichtlich des Explosionsschutzes nicht zulässig ist. Ist jedoch die Grundstromaufnahme aufgrund mehrerer Teilnehmer in einem räumlich sehr engen Raum 100 mA, reduziert sich die Leitungslänge auf 818 m. Wird ein Kabel mit einem höheren Widerstandsbelag R' verwendet, verursacht dies ebenfalls eine Verkürzung der Leitungslänge l_{\max} .

Bei dieser ganzen Betrachtung muss allerdings Folgendes berücksichtigt werden:

- Das FISCO-Modell hat durch Zündversuche nach EN 50020 (Explosionsschutz durch Eigensicherheit) festgestellt, dass das Kabel keinen Einfluss auf das Zündverhalten hat, wenn die entsprechenden Parameter eingehalten werden.
- Sind alle Teilnehmer, d. h. das Speisegerät und alle Feldgeräte, eines eigensicheren Bussystems gemäß FISCO zertifiziert, vereinfacht sich der Nachweis der Eigensicherheit dadurch, dass die am Bus vorhandenen Induktivitäten und Kapazitäten nicht betrachtet zu werden brauchen. Der Nachweis der Eigensicherheit reduziert sich auf die Betrachtung der Spannungen, Ströme und Leistungen.
- Durch die im FISCO-Modell vorgenommenen Begrenzungen hinsichtlich der Eingangsinduktivitäten und -kapazitäten wird erreicht, dass an einem eigensicheren Bussystem bis zu 10 Teilnehmer betrieben werden können. Dies wäre bei einer „normalen“ Betrachtung nicht möglich. In diesem Fall wäre die maximale Teilnehmerzahl auf 4 beschränkt.

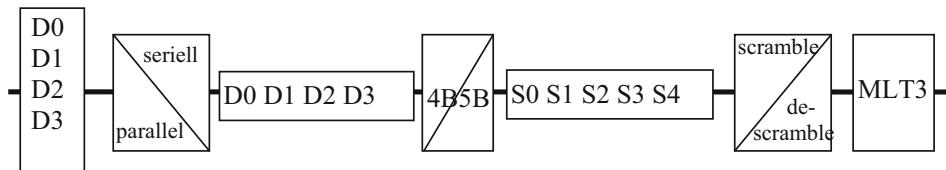
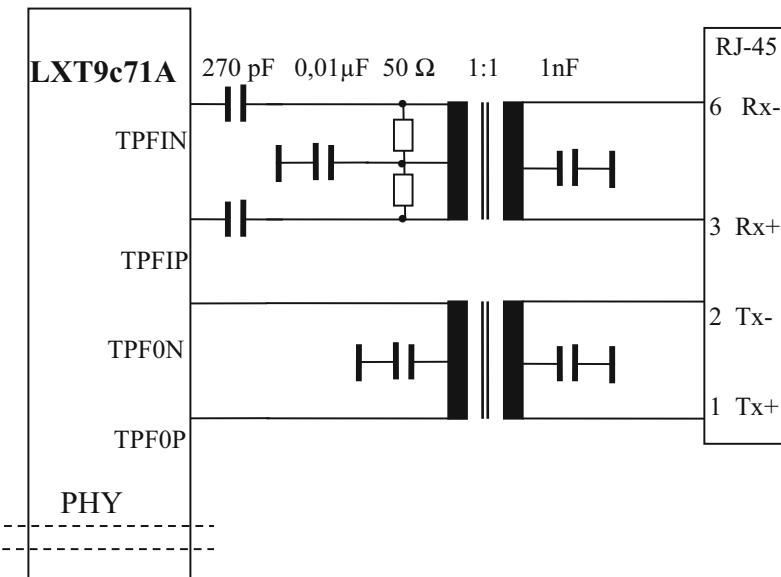
1.7.6 Ethernet-Übertragungsarten

Im Gegensatz zu klassischen Feldbussen, die z. B. RS 485 mit unterschiedlichen Baudaten benutzen, hängt die Baudate von Ethernet direkt an der verwendeten Übertragungsphysik. Es gibt zahlreiche Ethernet-Übertragungsarten mit Baudaten von 10 MBd bis 10 GBd, jeweils über optische und elektrische Leitungen.

Die physikalische Schnittstelle wird in der Regel mit einem eigenen IC realisiert, dem so genannten PHY, auch wenn es spezielle Ethernet-Controller gibt, die den PHY integrieren (Abb. 1.40 und 1.41). Im industriellen Umfeld sind zurzeit vor allem zwei Übertragungsarten gebräuchlich, nämlich 10Base-T mit 10 MBd und 100Base-Tx (Fast Ethernet) mit 100 MBd. Übliche PHYs beherrschen beides.

Gemeinsame Eigenschaften der Ethernet-Familie:

- Punkt-zu-Punkt-Verbindung,
- maximal 100 m Entfernung zwischen 2 Stationen,
- 2 symmetrische, verdrillte Leiterpaare zur Übertragung, Wellenwiderstand 100Ω ,

**Abb. 1.40** Blockbild eines PHY-Bausteins**Abb. 1.41** Ethernet-Anschaltung eines PHY-Bausteins

- ein Leitungspaar zum Senden und eins zum Empfangen, über Kreuz geschaltet,
- galvanische Trennung über Transformatoren sowohl auf Empfänger- als auch Senderseite (vgl. obige Skizze),
- Vollduplexbetrieb mit Switches möglich.

1.7.6.1 10 MBd Ethernet

Die Bitfolge wird Manchester-codiert, wodurch das Signal gleichspannungsfrei ist und Taktrückgewinnung im Empfänger möglich ist. Die Grundfrequenz ist wegen der Codierung 20 MHz.

10Base5 (IEEE802.3, Abschnitt 8)

Damit hat es angefangen, heute praktisch ausgestorben: Ein fingerdickes, wenig flexibles Koaxkabel (yellow cable, RG8) verband die Teilnehmer. Der Anschluss eines Teilnehmers erfolgte durch Anbohren der Leitung. Abschlusswiderstand 50 Ω.

- Leitungslänge je Segment: < 500 m, Gesamtlänge mit Repeater: < 2,5 km.
- Maximal 100 Teilnehmer je Segment.

10Base2 (IEEE 802.3, Abschnitt 10)

Lange Zeit der dominierende Standard: Das aus der Rundfunktechnik bekannte, dünne Koaxkabel (Cheapernet, RG58) verbindet die Teilnehmer. Jeder Teilnehmer benutzt ein BNC-T-Stück zur Ankopplung. Abschlusswiderstand 50Ω .

- Leitungslänge je Segment: < 185 m, mit Repeater $< 5 \times 185$ m.
- Maximal 30 Teilnehmer pro Segment.

10BaseT (IEEE 802.3 Abschnitt 14)

Diese heute übliche 10 MBd-Variante verwendet eine 4-adrige, verdrillte Leitung (Twisted Pair UTP, CAT 3 oder 5, siehe unten). Verwendet werden preiswerte, in der Telefontechnik übliche R-J45-Stecker/Buchsen (siehe unten).

Ein Hub (Radnabe!) sitzt in der Mitte mit Port für jeden Teilnehmer.

- Leitungslänge je Segment: < 100 m. Teilnehmer je nach Hub.

Twisted Pair Verdrillte Leitungen hat man in der Telefontechnik schon immer verwendet, um Störausstrahlungen zu verhindern. Man unterscheidet:

- UTP (unshielded twisted pair): Zwei ungeschirmte verdrillte Adernpaare in einem Mantel.
- STP (Shielded Twisted Pair): Gruppe der geschirmten Leitungen, darunter:
 - S/UTP (Shielded UTP): Die UTP-Paare sind insgesamt von einem Schirmmantel umgeben.
 - FTP (foiled twisted pair): Jedes Adernpaar ist von einer Schirmung umgeben.

CAT xx Man kann die Leitungen auch nach Anforderungskategorien 1–7 unterscheiden. Hier ist relevant:

- CAT 3: Dies ist eine UTP-Leitung. Sie kann bis 16 MHz verwendet werden, ist also tauglich für 10BaseT und ISDN.
- CAT 5: Heute überwiegend installiert. Sie kann bis 100 MHz verwendet werden. Wird auch bei 1000BaseT-Ethernet angewandt.
- CAT 7: Dies ist eine S/UTP-Leitung. Sie kann bis 600 MHz verwendet werden. Wird bei 10 GigaBd-Ethernet angewandt.

RJ-XX (Registered Jack) ist der genormte amerikanische Telefonstecker, auch Western-Stecker genannt. Hat sich auch in Deutschland durchgesetzt. Der Stecker RJ-45 hat acht Kontaktpositionen P und acht belegte Kontakte C und heißt deshalb korrekt 8P8C-Stecker. Die Kontaktbelegung bei Ethernet zeigt folgende Zuordnung:

P =		1	2	3	4	5	6	7	8
10BaseT, 100BaseT:	C =	Tx+	Tx-	Rx+			Rx-		
1000BaseT:	C =	D1+	D1-	D2+	D3+	D3-	D2-	D4+	D4-

Die Leitungen werden signalmäßig nicht gekreuzt, d. h., das Tx+ des Teilnehmers kommt an den Eingang Rx+ des Hub.

Es ist auch eine Glasfaser-Version des 10MBd-Ethernet genormt:

10BaseFL (IEEE 802.3, Abschnitt 18)

FL steht für Fiber optic Link. Die Verwendung von Glasfasern hat bei 10 MBd-Ethernet im Allgemeinen keine technischen Vorteile gegenüber UTP, es sei denn, extrem hohe Störsicherheit ist gefordert.

1.7.6.2 100 MBd-Ethernet (Fast Ethernet)

100Base-TX (IEEE 802.3, Abschnitt 25)

Diese heute übliche 100 MBd-Variante verwendet eine 4-adige, verdrillte ungeschirmte Leitung UTP (CAT 5) mit je einem verdrillten Adernpaar je Richtung, wie 10BaseT.

Als Stecker findet wie dort der RJ-45 Anwendung.

- Die Leitungslänge ist maximal 100 m.

Die Codierung der Bits erfolgt mit dem 4B5B-Verfahren (siehe unten), auf der Leitung wird spannungsmäßig mit MLT-3 codiert (siehe unten).

Bei der *4B5B-Codierung* wird jede Datenbytehälfte (nibble) durch Hinzufügen eines Bits auf 5 Bit erweitert (z. B. durch Einschieben einer „1“ zwischen letztes und vorletztes Bit.) Es darf kein Codewort mit mehr als einer führenden „0“ und mehr als zwei abschließenden „0“ geben. Damit wird trotz der Möglichkeit zur Taktrückgewinnung die Redundanz von 50 % bei Manchester auf 20 % gesenkt.

Auf der Kupferleitung wird dieser 4B5B-Bitstrom noch spannungscodiert mit *MLT-3* (*Multi Level Transmission encoding with 3 levels*): Positive Spannung (Symbol +), Nullpegel „0“, negative Spannung (Symbol -). Bei einer logischen „1“ im Datenstrom erfolgt ein Spannungszustandswechsel, bei einer logischen „0“ bleibt der Signalpegel gleich. Ein Codierungsbeispiel ist nachfolgend gezeigt. Man beachte dabei, dass auch eine Nullfolge Pegelwechsel erzeugt.

Datenstrom	0 1 1 1	0 1 0 0	0 0 1 0	0 0 0 0	0 0 0 0
4B5B-Code	0 1 1 1 1	0 1 0 1 0	1 0 1 0 0	1 1 1 1 0	1 1 1 1 0
MLT-3 Pegel	0	0 + 0 - 0	0 - - 0 0	- 0 0 0	- 0 - 0 0

Das nachfolgende Oszilloskopogramm (Abb. 1.42) zeigt die Signale bei 100Base TX Ethernet.

Man erkennt die dreiwertige Codierung MLT-3.

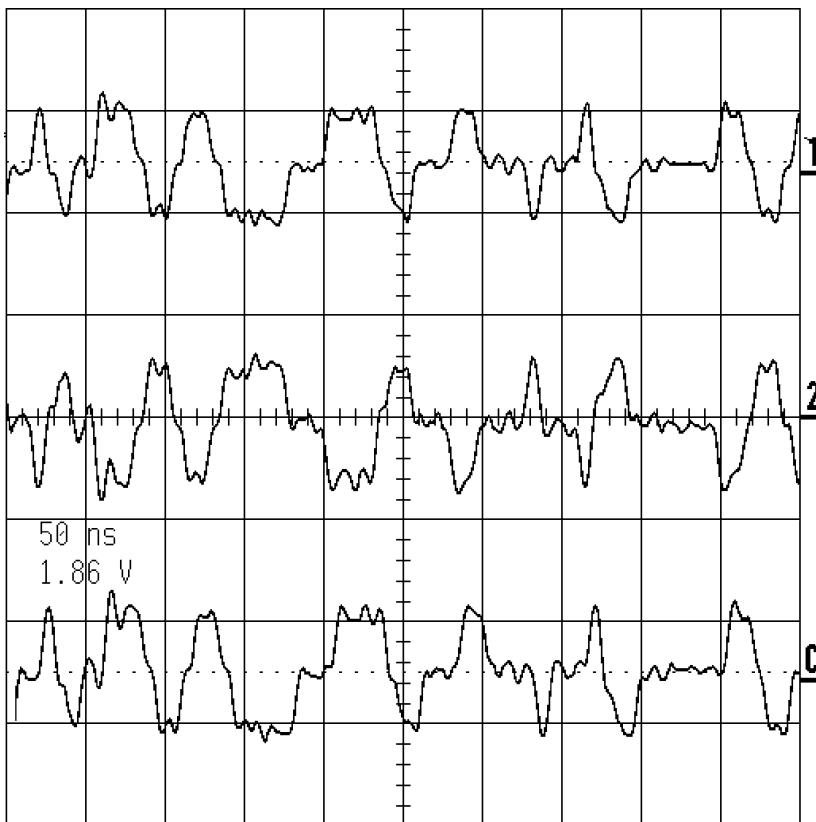


Abb. 1.42 Oszilloskopbilder der Signale bei 100Base-Tx Ethernet. 1: Tx+, 2: Tx-, C: Differenzsignal Tx+ – Tx-

100Base-FX (IEEE 802.3, Abschnitt 26)

Die Übertragung erfolgt über Multimode Glasfaser im 4B5B-Code.

- Segmentlänge maximal 400 m, mit Repeater bis 2 km.

1.7.6.3 1000Base Ethernet (Gigabit Ethernet)

Auch hier existiert eine Variante mit Kupferleitung und eine mit Lichtwellenleiter.

1000Base-T (IEEE 802.3, Abschnitt 40)

Der Datenstrom wird in vier Teilströme unterteilt, die über die vier Adernpaare der Leitung gleichzeitig gesendet und empfangen werden. Mit Pulsamplitudenmodulation (PAM 5, siehe unten) wird pro Schritt 1 Byte übertragen.

- Leitung: UTP (CAT 5e), Leitungslänge maximal 100 m.

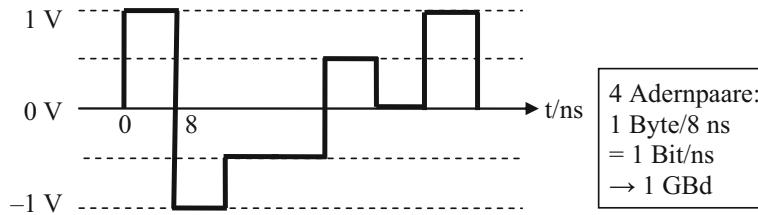
1000Base-CX (IEEE 802.3, Abschnitt 39)

Man verwendet zwei Adernpaare einer STP-Kupferleitung mit Wellenwiderstand 150Ω . Anschluss über RJ-45-Stecker (P8C).

- Leitungslänge < 25 m, unter Anwendung der Sternkopplung mit Hub.

Eine CAT-5-Leitung hat eine obere Grenzfrequenz von etwa 125 MHz. Das erlaubt bei klassischer 0/1-Codierung die Übertragung von maximal 1 Bit/8 ns. Gewünscht sind aber 1 Byte/8 ns. Man löst das Problem mit zwei Tricks:

- Man zerlegt die Daten in 4-Byte-Blöcke. Diese 4 seriellen Bytes sendet man dann zeitgleich auf 4 parallelen, verdrillten Kupferadern. Damit hat man den Faktor 4 gewonnen. Man braucht aber eine 8-fach höhere Übertragungsrate.
- Man geht von der 0/1-Codierung auf die *PAM 5-Codierung* über. Es gibt dabei 5 Spannungsspeicher, wie die Skizze zeigt. Man hat dann bei 4 Adernpaaren insgesamt $5^4 = 625$ Kombinationsmöglichkeiten 0/1.



Ein Byte benötigt aber nur $2^8 = 256$ Kombinationsmöglichkeiten. Man hat also noch ein Bit/Byte frei. Dieses verwendet man für eine Vorwärtsfehlerkorrektur mit Hilfe der *Trellis-Code-Modulation* (trellis – engl., Gitter). (Bei Bussystemen wird normalerweise eine nachträgliche Fehlererkennung durchgeführt. Erkennt der Empfänger dabei einen Fehler (z. B. über CRC), so fordert er eine Wiederholung der Sendung an. Bei der Vorwärtsfehlerkorrektur sendet man redundant, sodass der Empfänger den Fehler nicht nur erkennen, sondern auch gleich nachträglich korrigieren kann. Das ist aus Zeitgründen manchmal erforderlich (CDs, Satellitenfunk)).

1000Base-LX, SX (IEEE 802.3, Abschnitt 38)

Diese beiden Verfahren mit Lichtwellenleitern unterscheiden sich nur in der Wellenlänge des verwendeten Infrarot-Lasers.

- SX: 850 nm, LX 1310 nm.
- LWL-Länge $2 \text{ m} < l < 200 \text{ m}$ (500 m) bei Multimode-LWL, $l < 5 \text{ km}$ bei Monomode-LWL.

1.7.6.4 10GigaBit Ethernet

10GBASE-T (IEEE 802.3an)

Mit den verwendeten Leitungen U/UTP (CAT 6) ist etwa 50 m Leitungslänge (Stern) zu erreichen, mit U/UTP (CAT 5) etwa 22 m. Auch hier wird der Datenstrom in viermal 2,5 MBd aufgeteilt und jeweils über ein Adernpaar übertragen und am Ende wieder zusammengesetzt (vgl. GigaBd-Ethernet).

Trotz aller nachrichtentechnischer Sorgfalt ist zur Erzielung eines ausreichenden Störabstandes hier DSP (Digital Signal Processing) erforderlich.

10GBase-LX4, -SR, -LR, -ER, SW, -LW, -EW (IEEE 802.3ae)

Mit diesen Lichtwellenleitervarianten sollen Reichweiten zwischen 240 m und 10 km realisiert werden. Es bleibt abzuwarten, welche Varianten sich durchsetzen werden. Für die industriellen Netze spielen sie keine Rolle.

1.7.6.5 Infrastruktur-Komponenten

Da Ethernet 10Base-T und 100Base-TX jeweils als Punkt-zu-Punkt-Verbindung aufgebaut ist, sind Infrastruktur-Komponenten nötig, um ein Netz mit vielen Teilnehmern aufzubauen. Für Ethernet als industrielles Bussystem sind zwei Infrastruktur-Komponenten wichtig zu unterscheiden, Hubs und Switches.

Hub

Hubs sind einfache Repeater mit mehreren Ports. Der Hub überträgt je an einem Port empfangene Nachricht an alle anderen Ports. Dadurch können alle am Hub angeschlossenen Geräte miteinander kommunizieren. Kaskadiert man mehrere Hubs, kann man prinzipiell beliebig große Netze aufbauen. Aus der einfachen Funktion ergeben sich einige besondere Eigenschaften von Hubs:

- Die Baudrate an allen Ports ist gleich.
- Kollisionen können zwischen beliebigen Ports und zwischen verschiedenen Hubs auftreten.
- Jeder Hub verursachte eine kurze Verzögerung, typischerweise ca. 700 ns bei 10Base-T.
- Da über das gesamte, durch Hubs verbundene Netzwerk Kollisionen entstehen können und durch das CSMA/CD abgefangen werden müssen, ist die Anzahl der Hubs zwischen zwei Stationen begrenzt, um Kollisionen sicher auch für die kürzesten Telegramme zu erkennen. Daher sind bei 10Base-T nur 4 Hubs zwischen zwei beliebigen Stationen im Netz erlaubt, bei 100Base-T nur 3.

Switch

Ein Switch ist ein intelligenterer Hub. Switches sind in der Lage, ankommende Telegramme zwischenzuspeichern und dann zu verzögern, z. B. um Kollisionen zu vermeiden, und dann an die einzelnen Ports weiterzugeben. Außerdem lernen Switches das Netzwerk ein und können daher empfangene Telegramme gezielt an das Segment des Empfängers

schicken, aber nicht an die anderen Segmente. Dadurch kann je nach Topologie die Netzwerklast stark reduziert werden.

- Switches können verschiedenen Baudaten und Physiken an ihren Ports vereinigen.
- Kollisionen treten immer nur in einem Segment auf.
- Beliebig viele Switches können kaskadiert werden.
- Die Verzögerungszeit eines Switches ist länger, da zumindest die Ziel-MA Adresse gelesen werden muss, ehe das Telegramm an den entsprechenden Port weitergeleitet werden kann.
- Ohne zusätzliche Maßnahmen zur Regulierung des Telegrammverkehrs können durch Switches erhebliche Telegramm-Verzögerungen auftreten, da ein Telegramm auf seinem Weg durch die Segmente unter Umständen in jedem Segment andere Telegramme abwarten muss. Dies kann wegen des fehlenden Determinismus eine Einschränkung der Echtzeiteignung sein. Durch geeignete Maßnahmen lässt sich das Problem aber begrenzen.

Im Büro-Umfeld sind Hubs fast nicht mehr anzutreffen und auch im industriellen Umfeld dürften alle moderneren Ethernet-Installationen wegen der erheblichen Vorteile für die Topologie als „Switched Ethernet“ ausgeführt sein. Es ist aber wichtig, die Unterschiede zwischen Switches und Hubs zu kennen und um die potentiellen Probleme von Switches zu wissen.

1.8 Leitungen und Übertragungsarten

1.8.1 Übersicht über die Leitungsarten

Im Bereich der Feldbusse werden normalerweise Leitungen aus Kupfer verlegt. Einen Überblick gibt die nachstehende Tabelle.

	Kupferleiter			
	verdrillt, unabgeschirmt	verdrillt, abgeschirmt	Koaxkabel	
			UTP	STP
Z/Ohm	100–120	100–120	50	50
Bezeichnungen	IBM Type 3, VDE YR	IBM Typ 1, IBM Typ 2 VDE YCY VDE Y(St)Y	10base5	10base2 Cheaper-net

Lichtwellenleiter finden nur Anwendung bei extrem elektromagnetisch verseuchter Umgebung, da ihre Verlegung aufwändiger ist (vgl. Abschn. 1.8.4). Für die üblichen Feldbusse mit RS 485-Übertragung verwendet man meist verdrillte, abgeschirmte Kupferleiterpaare (*shielded twisted pair*), bei kurzen Entferungen auch ungeschirmte, verdrillte Kupferpaare (*unshielded twisted pair*). Die Übertragungsraten liegen stets unter 1 MBd.

Bei höheren Ansprüchen an Störsicherheit und Datenübertragungsrate (Token-Ring, -Bus, CSMA/CD) wählt man Koaxkabel.

1.8.2 Paralleldrahtleitung

Diese bei Automatisierungs-Bussen meistens verwendete Leiterart ist in Abb. 1.43a in ihrer geometrischen und in Abb. 1.43b in ihrer elektrischen Darstellung gezeigt.

Charakteristische dynamische Größe ist der Wellenwiderstand Z :

$$Z = \frac{U_1}{I_1} = \sqrt{\frac{\Delta L}{\Delta C}} .$$

Sein Wert liegt hier typisch bei 120Ω . Er ist insofern von praktischer Bedeutung, als eine Leitung mit ihrem Wellenwiderstand abgeschlossen sein muss, wenn man Reflexionen der Signale am Leitungsende vermeiden möchte. Dies kommt aber erst bei Übertragungsraten von über 100 kBd störend zum tragen. Abb. 1.44a zeigt den Abschlusswiderstand bei einer unsymmetrischen Leitung (z. B. für RS 232). In Abb. 1.44b ist der Abschluss für eine symmetrische Leitung (z. B. RS 485) gezeigt.

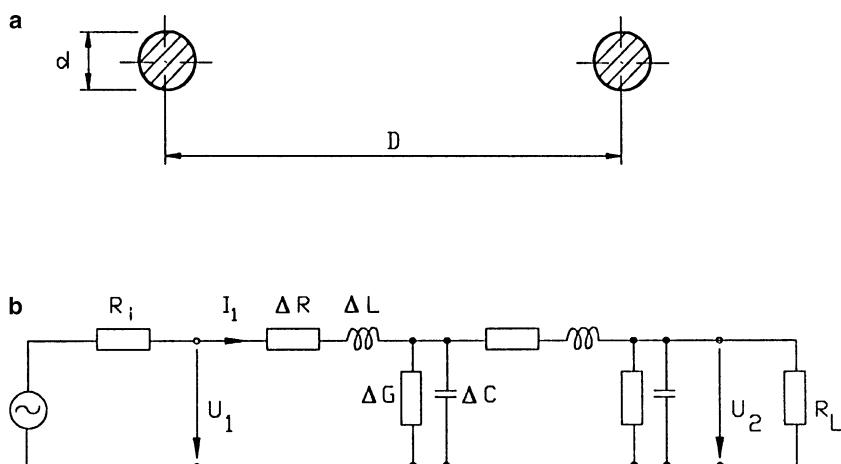
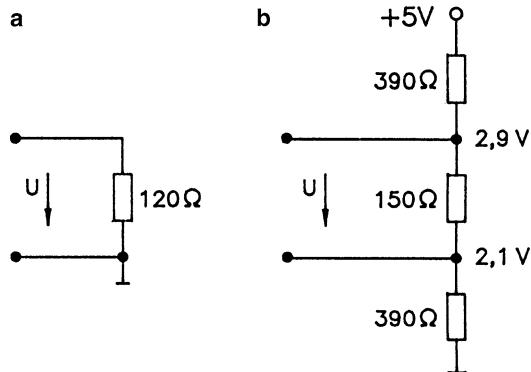


Abb. 1.43 Paralleldrahtleitung. **a** Geometrie, **b** Ersatzschaltung

Abb. 1.44 Abschlusswiderstand. **a** unsymmetrisch, **b** symmetrisch



Der Wellenwiderstand ist durch die Geometrie der Leitung bestimmt:

$$Z = \frac{120 \Omega}{\sqrt{\varepsilon_r}} \cdot \ln \frac{D}{d}, \quad (1.13)$$

ε_r – Dielektrizitätskonstante der Isolation.

Von Interesse ist der so genannte Verkürzungsfaktor V . Er gibt an, um wie viel die Ausbreitungsgeschwindigkeit der Nachricht auf der isolierten Leitung (v) langsamer ist als die auf der Freileitung (c):

$$V = \frac{v}{c} = \frac{1}{\sqrt{\varepsilon_r}}. \quad (1.14)$$

Beispiel

Ist die Isolation aus Polyäthylen PE mit $\varepsilon_r = 2,3$, so ist nach (1.14):

$$v = 0,66 \cdot c.$$

Für eine typische verdrillte, abgeschirmte Zweidrahtleitung werden die charakteristischen Werte unten angegeben:

VDE Y(St)Y 2 × 0,28	
abgeschirmt, Isolation: PVC	
A in mm ²	2 × 0,28
Außendurchmesser in mm	5
Aderdurchmesser in mm	0,6
ΔC in pF/m	120
ΔR in Ω/m	0,13
ΔG in mS/m	10
Dämpfung (800 Hz) in dB/km	1,7
Z in Ω	120

ΔR ist der Schleifenwiderstand gemäß Abb. 1.43, umfasst also Hin- und Rückleitung. Manchmal findet man die Querschnitte auch nach amerikanischen AWG-Werten angegeben (*American Wire Gauge*). Die folgende Drahttabelle zeigt die Zusammenhänge.

AWG	28	26	24	22	20
A/mm ²	0,08	0,13	0,2	0,32	0,5
d/mm	0,32	0,40	0,51	0,64	0,8
ΔR in Ω/m	0,436	0,28	0,178	0,106	0,07

Sehr wichtig ist das dynamische Verhalten der Leitung, die ja einen komplexen Vierpol darstellt (Abb. 1.43b). Ist die Leitung mit einem Wellenwiderstand Z abgeschlossen, so ergeben sich die zulässigen Leitungslängen aus dem Diagramm in Abb. 1.45.

Dieses Diagramm sei mit folgendem Beispiel erläutert:

Hat der treibende Generator den Innenwiderstand $R_i = 100 \Omega$, arbeitet er auf die Last $R_l = 100 \Omega$, vernachlässigt man den Induktivitätsbelag der Leitung und nimmt man den Kapazitätsbelag zu $\Delta C = 52,5 \text{ pF/m}$, den Widerstandsbelag zu $\Delta R = 0,178 \Omega/\text{m}$ (vgl. obige Tabelle) an, so ist die Zeitkonstante τ der Leitung bei $l = 100 \text{ m}$

$$\tau = \Delta C \cdot l \cdot [(R_i + \Delta R \cdot l)/R_l] = 0,386 \mu\text{s} .$$

Für die maximale Bitrate erhält man

$$\begin{aligned} BR_{\max} &= \frac{0,5}{\tau} \\ &= 1,29 \text{ MBd} . \end{aligned}$$

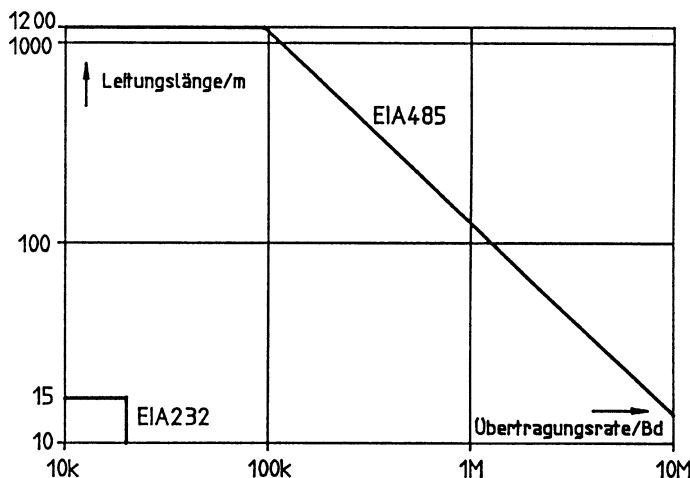


Abb. 1.45 Leitungslänge und Übertragungsrate für AWG 24 und 52,5 pF/m (EIA RS 422)

Berechnungsbeispiel für die maximale Leitungslänge

Sender S und Empfänger E sind die Leitungslänge l_{\max} voneinander entfernt (Abb. 1.46a). Die elektrische Ersatzschaltung zeigt Abb. 1.46b.

Der Sender liefert U_S , davon soll U_0 beim Empfänger ankommen.

Dann gilt für den maximal zulässigen Schleifenwiderstand R_S der Leitung:

$$R_S = \frac{Z \cdot (U_S - U_0)}{U_0} . \quad (1.15)$$

Die zulässige Leitungslänge l_{\max} ist dann

$$l_{\max} = \frac{R_S}{\Delta R} . \quad (1.16)$$

Es sei $Z = 100 \Omega$, $U_S = 6 \text{ V}$, $U_0 = 2 \text{ V}$. Damit nach (1.15):

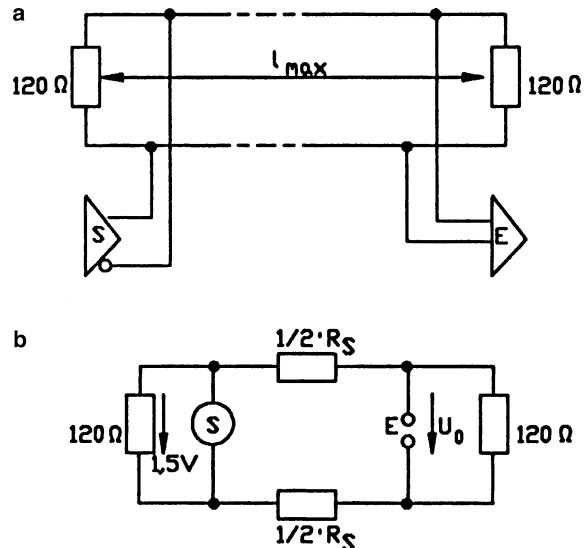
$$R_S = 200 \Omega .$$

Wählt man eine Leitung mit Drahtquerschnitt $A = 0,2 \text{ mm}^2$, so ist mit (1.16) und der Drahttabelle

$$l_{\max} = \frac{200 \Omega}{0,178 \Omega/\text{m}} = 1124 \text{ m} .$$

Länger als 1124 m darf also die Leitung in diesem Fall nicht sein. Aus dem Diagramm in Abb. 1.45 liest man ab, dass dies für alle $\text{BR} < 90 \text{ kBd}$ gilt. Für $\text{BR} > 90 \text{ kBd}$ ist die dynamische Berechnung zusätzlich durchzuführen.

Abb. 1.46 RS 422. a Geometrie, b Ersatzschaltung



1.8.3 Koaxialleitung

Diese vor allem bei Bussen mit hoher Übertragungsrate verwendete Leiterart ist in Abb. 1.47 gezeigt.

Der Wellenwiderstand Z ist wie in Abschn. 1.8.1 definiert. Sein Wert ist normalerweise 50Ω . Er hängt mit der Geometrie wie folgt zusammen:

$$Z = \frac{60 \Omega}{\sqrt{\varepsilon_r}} \cdot \ln \frac{D}{d} . \quad (1.17)$$

Der Verkürzungsfaktor V ist wie in (1.14) definiert.

Für eine typische Koaxialleitung sind die charakteristischen Werte unten angegeben.

RG 58	
A (Innenleiter) in mm	19 × 0,18
Außendurchmesser in mm	4,95
ΔC in pF/m	101
ΔR in Ω/m (R_i/R_a)	0,039/0,0135
Dielektrikum	PE
Z in Ω	50

Die Dämpfung ist in Abb. 1.48 gezeigt.

Zahlenbeispiel

Die maximal zulässige Dämpfung zwischen der sendenden Datenendeinrichtung (DEE) und der entferntesten DEE sei $a_{zul} = 11,5 \text{ dB}$.

Die verwendete Koaxleitung habe eine Dämpfung von $a_K = 5,6 \text{ dB}/100 \text{ m}$ bei $f = 10 \text{ MHz}$ (vgl. Abb. 1.48).

Dann ist die maximale Leitungslänge

$$l_{\max} = \frac{a_{zul}}{a_K} = 203 \text{ m} .$$

Abb. 1.47 Koaxkabel

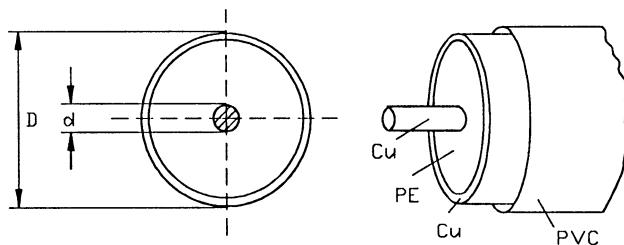
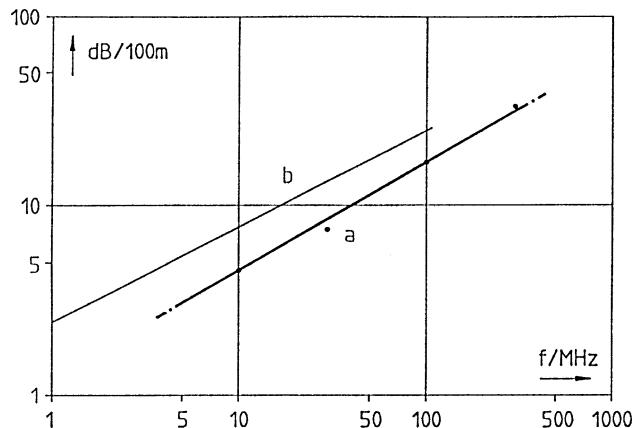


Abb. 1.48 Leitungsdämpfungen a_K . a Koaxkabel, b STP-Kabel



Von dieser Länge sind die Anschlussleitungslängen l_a abzuziehen sowie die der Anschlussdämpfung a_T entsprechende Leitungslänge $l_T = a_T/a_K$, wodurch sich die tatsächlich zulässige Leitungslänge l_{zul} ergibt (Abb. 1.49):

$$l_{\text{zul}} = l_{\text{max}} - \frac{a_T}{a_K} \cdot n - 2 \cdot l_a \cdot n , \quad (1.18)$$

n – Anzahl der anzuschließenden DEE.

Zahlenbeispiel

$$n = 10, l_a = 3 \text{ m}, a_T = 0,1 \text{ dB}.$$

Aus (1.18):

$$l_{\text{zul}} = 203 \text{ m} - \frac{0,1 \text{ m}}{0,056} \cdot 10 - 2 \cdot 3 \text{ m} \cdot 10 = 125 \text{ m} .$$

So lang darf hier die Leitung zwischen den beiden Abschlusswiderständen bzw. zwischen 2 Repeatern maximal sein.

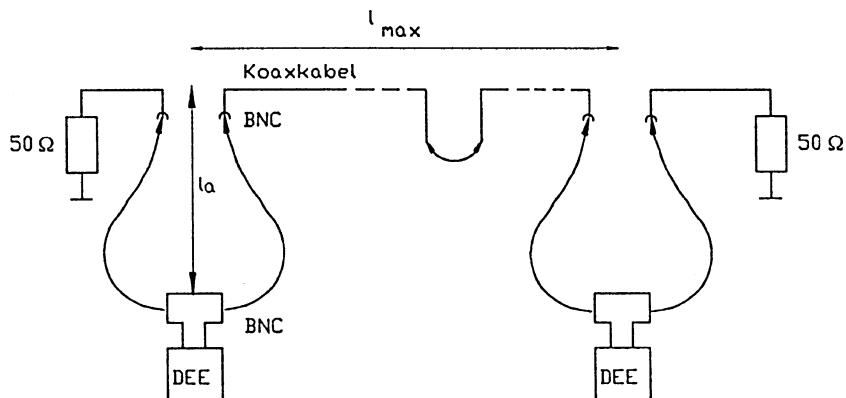


Abb. 1.49 Koaxverkabelung

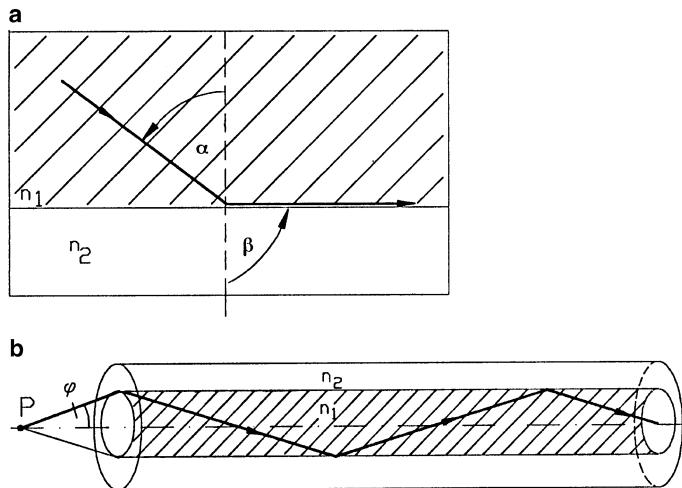


Abb. 1.50 a Brechung bei $n_1 > n_2$, b Lichtwellenleiter

1.8.4 Lichtwellenleiter (LWL)

Im Bereich der Automatisierungstechnik werden Busse mit LWL nur dann eingesetzt, wenn elektromagnetische Störungen mit anderen Mitteln nicht in den Griff zu bekommen sind oder die Datenraten sehr hoch sind. Dies ist selten. Deshalb beschränken wir uns hier auf das Notwendigste.

Grundprinzip: Trifft ein Lichtstrahl auf die Grenzfläche zwischen zwei Gläsern mit verschiedenen Brechungsindizes n_1 und n_2 (vgl. Abb. 1.50a), so wird er total reflektiert, wenn $\beta \geq 90^\circ$ ist. Damit folgt aus dem Snellius'schen Brechungsgesetz:

$$\alpha \geq \arctan \frac{n_2}{n_1} . \quad (1.19)$$

Man wählt meist n_1 um etwa 1 % größer als n_2 .

Führt man die beiden Gläser in Form eines dünnen Lichtwellenleiters aus, so ist der Lichtstrahl darin „gefangen“, kann also geführt werden (Abb. 1.50b).

Aus dem Akzeptanzwinkel φ folgt die so genannte numerische Apertur des LWL:

$$\sin \varphi = \sqrt{n_1^2 - n_2^2} . \quad (1.20)$$

Sie ist wichtig bei der Dimensionierung der Lichtquelle bzw. Optik (LED, Laser) in P. Die nachfolgende Tabelle gibt Auskunft über die Ausführungsformen der LWL.

	Multimode-LWL		Monomode-LWL
	Stufenprofil	Gradientenprofil	
Verlauf des Brechungsindex			
Innen Ø Außen Ø	50 µm 125 µm	125 µm	9 µm 125 µm
Dämpfung 1 dB/100 m bei ...	100 MHz	1 GHz	100 GHz

1.8.5 Übertragungsarten

1.8.5.1 Basisbandübertragung

Dies ist die einfachste und für Bus-Systeme in der unteren Ebene der Automatisierungs-technik allgemein angewandte Übertragungsart.

- Die logische 0 entspricht dem Spannungspegel x , bzw. der Polarität A-B;
- die logische 1 entspricht dem Spannungspegel y , bzw. der Polarität B-A.

Man arbeitet mit verschiedenen Pegeln, z. B.:

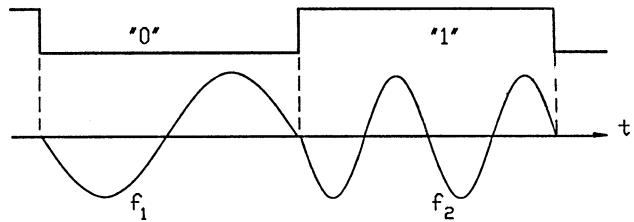
	1	0	
TTL	5 V	0 V	gegen Masse
RS 232	$-15 \text{ V} < U < -3 \text{ V}$	$+3 \text{ V} < U < +15 \text{ V}$	gegen Masse
RS 485	$-6 \text{ V} < U_{AB} < -1,5 \text{ V}$	$1,5 \text{ V} < U_{AB} < 6 \text{ V}$	Leitung A gegen Leitung B

1.8.5.2 Trägerfrequenzübertragung

Bei dieser Übertragungsart wird der logischen 0 die Frequenz f_1 und der logischen 1 die Frequenz f_2 zugeordnet (Frequenzmodulation) (Abb. 1.51).

Bei Übertragungslängen $> 1200 \text{ m}$ über Telefonleitungen o. Ä. wird oft Trägerfrequenzübertragung in Form von FSK (Frequency Shift Keying) verwendet. Bei Vollduplex-

Abb. 1.51 Trägerfrequenzübertragung



betrieb befindet sich ein Modem (Modulator/Demodulator) auf jeder Seite der Leitung. Eine übliche Kodierung ist:

Anfrage:	“1” – 1270 Hz	Antwort:	“1” – 2225 Hz
	“0” – 1070 Hz		“0” – 2025 Hz

Die Übertragungsrate ist niedrig, typisch 1200 Bd.

1.8.5.3 Breitbandübertragung

Die Breitbandübertragung gestattet die Übertragung vieler Kanäle auf einem Übertragungsmedium, inklusive Video, Daten, Sprache.

Anwendung findet diese aufwändige Technik in der obersten Hierarchieebene der Automatisierungstechnik als „Backbone-Netz“.

Charakteristisch ist hierbei, dass die Übertragung unidirektional erfolgt, d. h., der Sender sendet auf einem 6 MHz breiten Kanal im Vorwärtsband in Richtung Frequenzumsetzer (Abb. 1.52). Dort wird die Nachricht auf das Rückwärtsfrequenzband umgesetzt und zum Empfänger geschickt.

Das Vorwärtsband umfasst 30 Kanäle, das Rückwärtsband 39 Kanäle, es können also 30 Kommunikationen zeitgleich laufen.

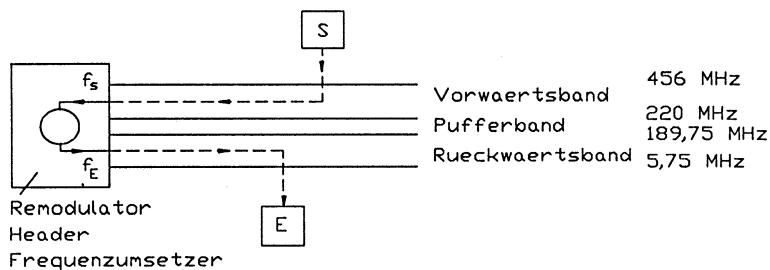


Abb. 1.52 Breitbandübertragung

1.9 Verbindung von Netzen

1.9.1 Repeater

Das Betreiben von Netzen erfordert die Übertragung der Daten zwischen den Teilnehmern. Die Teilnehmer sind physikalisch an das Übertragungssystem angeschlossen und kommunizieren über eine elektrische Schnittstelle miteinander. Die für eine Übertragung zulässige Entfernung ist aber in jedem Fall limitiert. Soll die gesteckte Grenze überschritten werden, so muss das Signal verstärkt werden. Ein anderes Problem stellt sich, wenn ein Bussystem mehr Teilnehmer verwalten (adressieren) kann, als an die elektrische Schnittstelle angeschlossen werden dürfen. In beiden Fällen ist das Datensignal zu verstärken. Diese Aufgabe wird von so genannten Repeater übernommen.

Der Repeater kopiert die auf der Leitung empfangenen Bits auf die jeweils andere Seite und verstärkt sie dabei. Dadurch wird das Netzwerk in zwei oder mehrere Teile gegliedert. Legt man für Repeater das OSI 7-Schichten-Modell zugrunde, verbindet ein Repeater zwei Endsysteme auf der Schicht 1 (Abb. 1.53).

Die Bearbeitung der Daten durch den Repeater erfolgt lediglich durch die eingesetzte Hardware. Die Teilnetzwerke müssen daher von gleicher Art sein (z. B. PROFIBUS-DP Repeater oder AS-Interface Repeater). Bei der Verstärkung werden die Daten nicht verändert, da der Repeater sich zwar in Bezug auf das physikalische Signal aktiv verhält, auf der

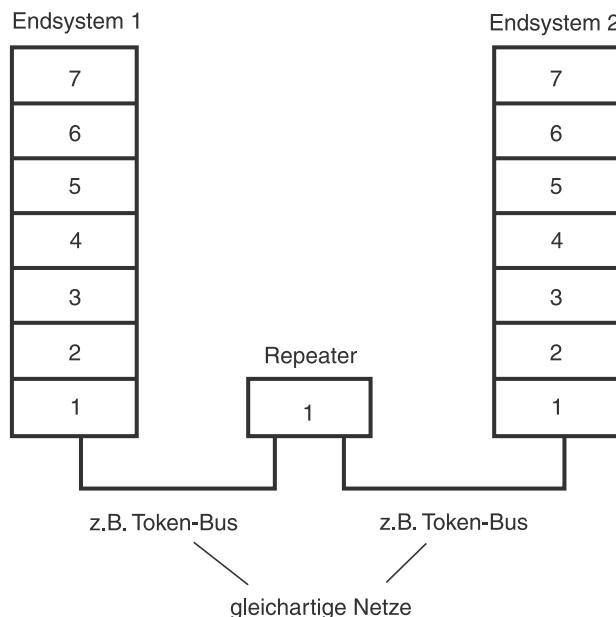


Abb. 1.53 Eine Verstärkerstation (Repeater) innerhalb eines Netzwerks

anderen Seite aber keine Software beinhaltet und damit in Bezug auf die Dateninhalte passiv reagiert. Das bedeutet, dass sie i. d. R. keiner Konfiguration bedürfen. Aus Sicht der Kommunikationsteilnehmer ist der Repeater unsichtbar, jedoch verursacht er eine Zeitverzögerung, was sich z. B. bei Master/Slave-Systemen nachteilig auswirken kann. So ist z. B. die Anzahl der in Serie geschalteten Repeater begrenzt. Der Repeater benötigt zum empfangen, aufbereiten und senden des Telegramms Zeit. Bei einem Master/Slave-Prinzip erkennt der Master einen defekten Slave u. a. daran, dass die Slaveantwort nicht innerhalb eines bestimmten Zeitintervalls erfolgt. Werden nun zu viele Repeater in Reihe geschaltet kann es passieren, dass die, durch die Repeater verursachte Verzögerung so groß ist, dass der Slave nicht mehr rechtzeitig antworten kann. Bei AS-Interface dürfen z. B. maximal 2 Repeater, bei PROFIBUS in Abhängigkeit der Übertragungsrate 4 bzw. 7 Repeater in Reihe geschaltet werden.

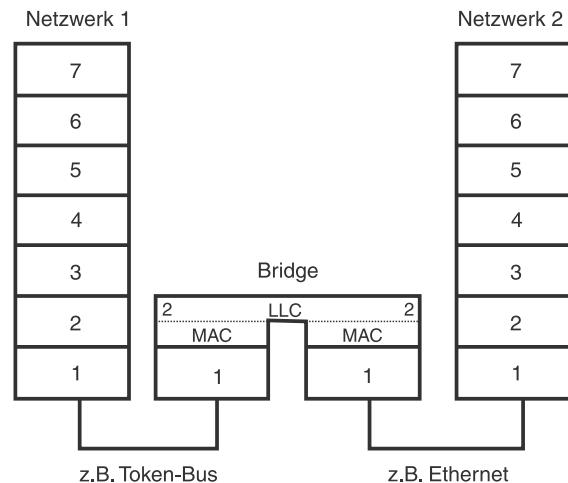
Repeater für Lichtwellenleiter wandeln das eingehende optische Signal in ein elektrisches Signal (Photodiode), verstärken es und senden es wieder als Lichtsignal (mittels LED) in die Leiterstrecke.

In Repeatern werden keine Kontrollen bezüglich der Korrektheit der ankommenden Daten durchgeführt. Die ankommenden Datenbits werden 1 : 1 auf den Ausgang kopiert.

1.9.2 Bridges

Bridges werden zur Verbindung zweier oder mehrerer lokaler Netze eingesetzt (Abb. 1.54). Beide Netze müssen auf der oberen Teilebene der Sicherungsschicht (*Logical Link Control* der OSI-Schicht 2) mit denselben Protokollen arbeiten. Die Übertragungsmedien können unterschiedlich sein, was natürlich auch unterschiedliche Protokolle für den Zugriff auf die Übertragungsmedien (*Medium Access Control* der OSI-Schicht 2) zufolge hat.

Abb. 1.54 Bridge zwischen zwei Netzwerken. LLC: Logical Link Control, MAC: Medium Access Control



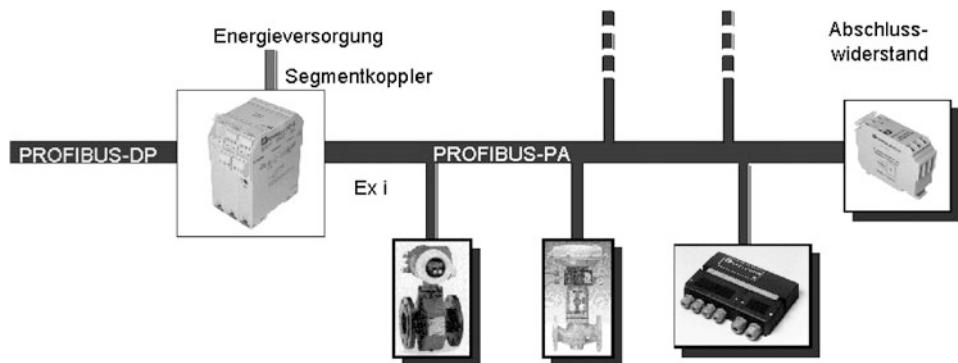


Abb. 1.55 Beispiel für eine Bridge (Segmentkoppler)

Bridges werden hauptsächlich dann eingesetzt, wenn lokale Netzwerke unterschiedlicher Topologien, Übertragungsarten und/oder unterschiedlicher physikalischer Schnittstellen zu verbinden sind oder wenn durch spezielle Anwendungen bestimmte Strukturen an bestehende Netze angebunden werden sollen. Die Benutzung der Bridge ist für den Benutzer normalerweise transparent, d. h. er erkennt nicht, dass eine Bridge im Netz vorhanden ist und kann mit seinem üblichen Befehlssatz alle ihm zugänglichen Stationen im Netz erreichen.

Die Bridge enthält eine Adresstabelle, aus der hervorgeht, welche Stationen der beteiligten Subnetze von ihr aus angesprochen werden können. Der Datenverkehr im Netz kann über die Bridge nur diejenigen Stationen erreichen, die in ihrer Adresstabelle angegeben sind. Dadurch übt die Bridge im Netz eine Filterfunktion aus. Sie leitet nur solche Datenpakete in ein Netzwerk weiter, die auch an einen Empfänger adressiert sind, der in diesem Netzwerk liegt und der in ihrer Tabelle vorhanden ist. Dies führt zu einer Lastentkopplung in den Teilbereichen des Gesamtnetzes, die über Bridges gekoppelt sind, da bestimmte Telegramme nur innerhalb des Teilnetzes übertragen werden.

Da Bridges auf der OSI-Schicht 2 arbeiten, können sie auch zur Handhabung unterschiedlicher Übertragungsgeschwindigkeiten in den Teilnetzen eingesetzt werden (Abb. 1.55).

Die Aufgaben der Bridge beziehen bei manchen Ausführungen nur auf den unteren Teil der Sicherungsschicht, auf die Zugriffskontrolle des Mediums (MAC), die logische Anbindung (*Logical Link Control LLC*) an die übrigen Schichten bleibt dabei unberührt. Dies soll durch das obige Beispiel verdeutlicht werden. In diesem Beispiel ist der Segmentkoppler die Bridge zwischen PROFIBUS-DP und PROFIBUS-PA. Die Umsetzungsaufgaben dieser Bridge werden in der folgenden Tabelle aufgelistet:

	PROFIBUS-DP	PROFIBUS-PA
Elektrische Schnittstelle	RS485	IEC 61158-2
Übertragungsart	Asynchron	Synchron
Übertragungsrate	9,6 kBd ... 12 MBd	31,25 kBd
Zugriffsverfahren	Token passing	Token passing
Datensicherungsverfahren	Blocksicherung	CRC-Check

Die Tabelle zeigt, dass sich PROFIBUS-DP und PROFIBUS-PA in allen Punkten der Schicht 1 und 2 unterscheiden, mit Ausnahme des Zugriffsverfahrens. Der Segmentkoppler übernimmt die „Bridge-Funktionalität“ und passt die unterschiedlichen Systeme so an, dass aus Sicht der Steuerung, ein PROFIBUS-DP-Teilnehmer, sich alle PROFIBUS-PA-Teilnehmer verhalten wie ein „normaler“ PROFIBUS-DP-Teilnehmer. Der Segmentkoppler arbeitet transparent. Dies hat wieder den Vorteil, genau wie beim Repeater, dass dieser nicht konfiguriert zu werden braucht und man von der Steuerung einen direkten Zugriff auf jeden PROFIBUS-PA-Teilnehmer hat.

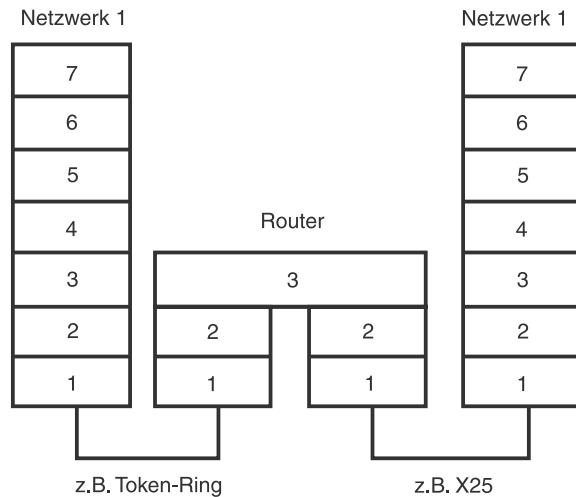
Bridges werden teilweise auch mit „intelligenter“ Software zum Netzmanagement angeboten. Sie erkennen Schleifen in größeren Netzwerken und nutzen die vorgefundene Redundanz beim Ausfall von Teilnetzen. Auch eine Steuerung des Datenverkehrs über unterschiedliche Wege zur optimalen Auslastung des Netzes ist möglich.

1.9.3 Router

Datenpakete, die ihren Weg von einem Netzwerk zum anderen Netzwerk nehmen, müssen auf ihrer Reise geführt werden. Die Bestimmung des Weges der Nachricht im Netzsystem wird als Routing bezeichnet, die dafür zuständigen Einheiten zur Leitwegebestimmung heißen Router. Der Router arbeitet auf Schicht 3 (Netzwerkschicht) des OSI-Modells (Abb. 1.56).

Für die Bestimmung des Leitweges in einem Netz oder Subnetz existieren eine Reihe adaptiver und nichtadaptiver Algorithmen, die jeweils auf ein bestimmtes Kriterium der Wegwahl besonderes Gewicht legen. Gängige Konzepte sind die des kürzesten Pfades (in Bezug auf die geographische Entfernung oder die Anzahl der Zwischenstationen oder die mittlere Übermittlungsgeschwindigkeit eines Paketes), der Mehrfach-Leitwegbestimmung (Verteilung der Datenpakete auf mehrere Wege zur Erhöhung des Durchsatzes), der zentralisierten Leitwegbestimmung (von einem Steuerzentrum aus), der dezentralen Leitwegbestimmung (an jedem Knotenpunkt) oder der Leitwegbestimmung nach Optimalitätsprinzipien (Auswertung von Bäumen).

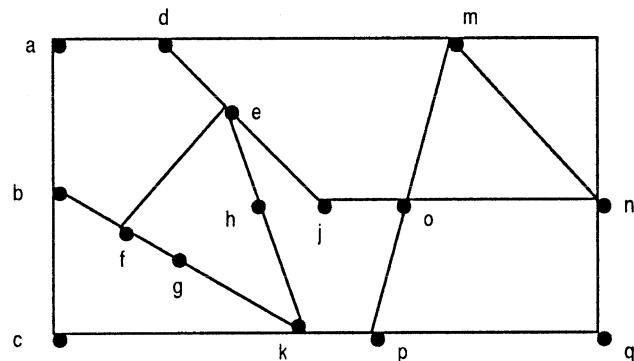
Die Bestimmung des Leitweges wird normalerweise in der Netzwerkschicht vorgenommen. Bei verbindungslosen Diensten (*Datagram*) wird für jedes Paket einzeln bestimmt, welchen Weg es durch das Netz nehmen soll. Wird eine virtuelle Verbindung benutzt, so wird zum Zeitpunkt des Verbindungsbaus festgelegt, welcher Leitweg für die gesamte Verbindungsduer gewählt wird.

Abb. 1.56 Router

Der Router bestimmt den Weg, den das Datenpaket zu nehmen hat nach einem oben genannten oder einem anderen Algorithmus.

In Abb. 1.57 ist ein Beispiel für ein Netzwerk gegeben. Die angegebenen Knoten können Netzstationen oder selbst wiederum Netze sein. Die Wahl des Weges von Knoten „a“ nach „k“ kann über „b“ oder „d“ erfolgen. Dort gibt es wiederum je zwei Möglichkeiten des Weitertransports. Die Fortsetzung des Weges über „c“ führt zur Lösung mit den wenigsten Zwischenstationen, während die Wege über „d-e-h“ und „b-f-g“ geographisch kürzer sind. Möglicherweise sind aber alle diese Wege überlastet und die Datenpakete kommen über „d-m-o-p“ schneller bei „k“ an als über die anderen Wege.

Damit die Datenpakete den Weg durch das Netz finden können, muss der Router sie so verändern, dass die gewünschte Wegsteuerung eingehalten werden kann. Der Router bearbeitet dabei die Ziel- und Quelladressen für die Netzwerkschicht, bevor er die Pakete weiterleitet.

Abb. 1.57 Netzbeispiel zur Leitwegbestimmung

Da Router wesentlich komplexere Aufgaben bewältigen müssen als Bridges, bieten sie eine geringere Arbeitsgeschwindigkeit. Neuere Entwicklungen von Routern mit hohen Durchsatzgeschwindigkeiten bei etwas verringertem Funktionsumfang und die wachsende Komplexität von Bridge-Algorithmen haben eine Klasse zwischen Router und Bridge entstehen lassen. Es sind die so genannten Brouter oder Routing Bridges.

Es gilt, in jedem Einzelfall zu prüfen, ob bei den gegebenen Anforderungen eine Bridge, ein Router oder eine Routing Bridge einzusetzen ist.

1.9.4 Gateways

Ein Gateway dient der Kopplung von Netzwerken mit unterschiedlicher Architektur (Abb. 1.58). In der Denkweise des OSI-Modells ist die Aufgabe des Gateways die Übersetzung der Kommunikationsprotokolle für alle erforderlichen Schichten. In der Automatisierungstechnik werden Gateways sehr häufig als Kopplung zwischen Feldbusystemen unterschiedlicher Funktionalität eingesetzt. So wird z. B. AS-Interface eingesetzt, um binäre Sensoren und Aktoren zu vernetzen. InterBus oder PROFIBUS-DP wird verwendet, um komplexere Sensoren, Aktoren, Bedienterminals, ... mit der Steuerung zu verbinden. Um die Signale, die über AS-Interface übertragen werden, der Steuerung zur Verfügung zu stellen ist ein AS-i/PROFIBUS-DP Gateway notwendig. Dieses Gateway verhält sich auf der AS-Interface Seite wie ein AS-i Master und auf der PROFIBUS-DP Seite wie ein PROFIBUS-DP Slave, es koppelt zwei vollkommen unterschiedliche Systeme miteinander.

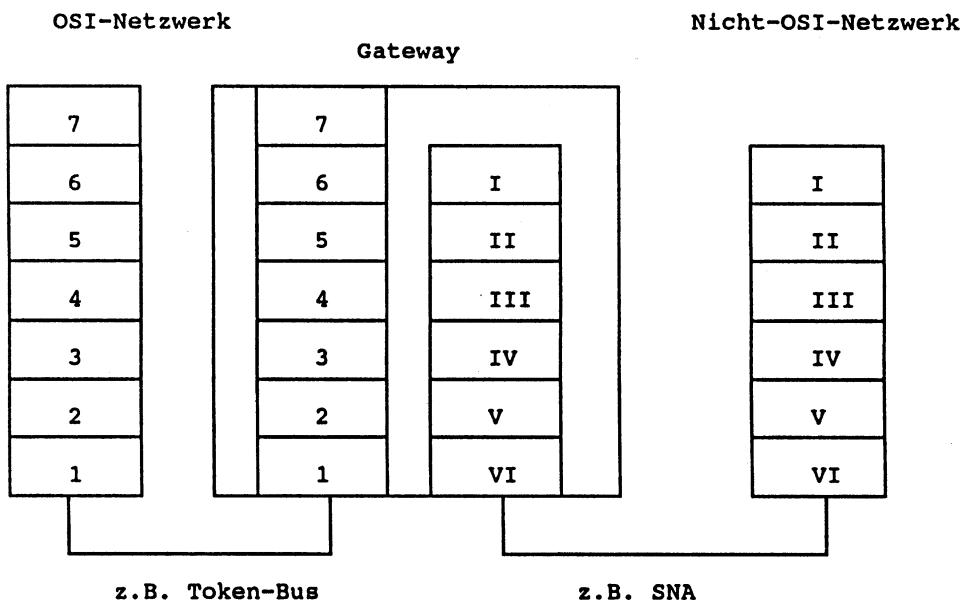


Abb. 1.58 Gateway

Diese Gateways ermöglichen es, dass reale Anlagen strukturiert werden können. Das bedeutet, dass in einer Anlage für ein Problem A, für das das Bussystem X am besten geeignet ist, das Bussystem X eingesetzt wird. Für ein anderes Problem B, für das das Bussystem Y am besten geeignet ist, wird dieses System verwendet. Gekoppelt werden die beiden Systeme über Gateways.

Gateways ermöglichen auch den Übergang zwischen Netzen der OSI-Welt und der übrigen Netzwerk Welt (z. B. SNA). Heute ist sehr häufig in der Diskussion, das Industrial Ethernet auch für die Automatisierungstechnik zu verwenden, d. h. in der unteren Feldebene. Gateways zwischen Feldbus systemen (z. B. InterBus, PROFIBUS, ...) stellen dann die Kopplung zwischen Systemen nach dem OSI-Modell und nach dem TCP/IP-Modell (Ethernet) zur Verfügung.

Bei Verbindung zwischen zwei oder mehreren Netzen (OSI) ist es oftmals der Fall, dass das Gateway nur bis zur Transportebene arbeiten muss. Die Kopplung zwischen einem OSI- und einem Nicht-OSI-Netzwerk kann dagegen eine weitergehende Behandlung erfordern.

Kann ein beliebiger Hostrechner an einem Netz einen beliebigen anderen, ihm bekannten Hostrechner am anderen Netz erreichen, so spricht man von einem vollständigen Gateway. Ist das Gateway nur von einem Netzknoten zu einem anderen Netzknoten realisiert, so wird es als halbes Gateway bezeichnet.

Sollen Rechner in mehreren Netzwerken miteinander kommunizieren können, so verwendet man als Gateway meist ein Netzwerk, zu dem alle anderen Netze Zugang haben. Ein solches Gateway-Netzwerk bezeichnet man auch als Internet.

Das Gateway hat die Aufgabe, die Adressen in den verschiedenen Netzen umzusetzen, Paket- und Darstellungsformate umzuwandeln, verbindungslose und verbindungsorientierte Dienste bereitzustellen und bei Gateway-Netzwerken die Wegwahl zwischen den Netzen zu verwalten.

Im OSI-Schichtenmodell ist das Gateway auf Schicht 4 zu finden. Falls es die Aufgabe erforderlich macht, werden auch höhere Schichten in das Gateway mit einbezogen.

1.10 Feldbusankopplung an Host-Systeme

1.10.1 Grundlagen

Prinzipbedingt benötigen alle Feldbus systeme die Gesamtfunktion steuernde und überwachende Geräte, die man als Hosts (oder Host-Rechner, Host-Systeme) bezeichnet. Bisweilen wird auch der Begriff „Master“ verwendet, um zu kennzeichnen, dass dieses Gerät übergeordnete Aufgaben im Sinne einer Befehlszentrale des vernetzten Systems wahrnimmt. In diesem Kontext ist „Master“ nicht mit der bei der Definition von Buszugriffsverfahren verwendeten Terminologie „Master-Slave“ identisch. Busknoten, die bezüglich des Buszugriffsverfahrens Master darstellen, müssen nicht notwendigerweise auch Master im Sinne eines Host-Systems sein. Beispielsweise sind bei CSMA/CA-Verfahren (z. B. CAN,

LON) alle Busknoten Busmaster – also gleichberechtigt –, nur einer oder einige wenige davon werden aber bei üblichen Installationen auch die Rolle eines Hosts übernehmen. Im Folgenden sollen die prinzipiellen Möglichkeiten der Ankopplung von Feldbussen an Host-Systeme sowohl unter Hardware- als auch Softwaregesichtspunkten kurz aufgezeigt werden. Auf eine zu detaillierte Darstellung im Einzelfall wird aber bewusst verzichtet, da diese sehr herstellerspezifisch ausfällt.

1.10.2 SPS-Ankopplung

Speicherprogrammierbare Steuerungen (SPS) stellen insbesondere in der Fertigungstechnik den größten Anteil der Host-Systeme dar. Deren Techniken der Feldbusanbindung sind seit längerer Zeit eingeführt und bei praktisch allen SPS-Herstellern serienmäßig im Programm. Üblicherweise werden SPS über zusätzliche, so genannte Kommunikations-Baugruppen, an den jeweiligen Feldbus angekoppelt.

1.10.2.1 Feldbusfunktionen auf Kommunikations-Baugruppen

Alle feldbusrelevanten Funktionen werden dabei immer in der Baugruppen-Elektronik ausgeführt:

- **Funktionen der Schicht 1 (Physikalische Busankopplung)**

Die eigentlichen Sende- und Empfangssignale werden über entsprechende Verstärkerbausteine, kombiniert mit einem Netzwerk passiver Bauteile (Widerstände, Dioden etc.), geführt. Je nach Busspezifikation kommen dabei galvanische Trennungen zur Anwendung, die beispielsweise aus Optokopplern und DC/DC-Wandlern für die Spannungsversorgung der busseitigen Elektronik bestehen. Kostenmäßig ist die Implementierung der Schicht 1 nur in kleinerem Maße von dem Feldbus Typ abhängig, wesentlicher Kostenfaktor ist vielmehr meist die Frage, ob eine galvanische Trennung implementiert werden muss oder nicht. Im industriellen Umfeld wird dies allerdings in der Regel empfohlen.

- **Funktionen der Schicht 2 (Zugriffsverfahren, Datenrahmen, Datensicherung)**

Typischerweise werden diese Funktionen von auf der Baugruppe vorhandenen Prozessoren wahrgenommen. Die Prozessoren sind dabei mit einer seriellen Schnittstelle ausgestattet, über die sie die zu sendenden bzw. zu empfangenden Bitfolgen mit den Blöcken zur physikalischen Busankopplung austauschen. Als Prozessoren kommen meist Standard-Typen, wie sie sonst für allgemeine Applikationen auch eingesetzt werden, zur Anwendung (z. B. 8051, 80 166).

- **Funktionen der Schicht 7 (Anwenderzugriff)**

Die Funktionen werden ebenfalls vom Baugruppen-Prozessor übernommen. Konkret handelt es sich dabei in der Regel um die Umsetzung der in den jeweiligen Feldbusspezifikationen definierten Dienste (z. B. Lesen, Schreiben, ...) und Objekte in

entsprechende Aktionen, die letztlich zur Aussendung und zum Empfang von Datenrahmen über die Schicht 2 führen.

Abweichend von obiger Darstellung existieren noch einzelne Produkte, welche Funktionen der Schicht 7 nicht vollständig auf der Baugruppe implementieren, sodass das Anwendungsprogramm noch um entsprechenden Code erweitert werden muss. Da die üblichen Verfahren der SPS-Programmierung eine „feldbusnahe“ Programmierung aber nur sehr schwer erlauben, besteht ein eindeutiger Trend zu voll-integrierten Baugruppen.

1.10.2.2 Software-Schnittstelle

Softwaremäßig werden meist folgende Elemente herangezogen:

- Funktionsbausteine (Siemens: „Hantierungsbausteine“) mit dem eigentlichen Programmcode zur Initialisierung der Feldbuschnittstelle sowie zum Datenaustausch,
- Datenbausteine zur Ablage von Konfigurationsparametern (wie Geräteadressen etc.) und zur Zwischenablage für nicht-zyklische Kommunikationsdaten,
- E/A-Bereich der SPS mit Teilbereichen, in denen zyklisch abgefragte Empfangsdaten bzw. zyklisch auszugebende Sendedaten abgelegt sind.

Je nach Hersteller sind dabei sowohl zyklische als auch azyklische Datentransfers möglich. Bei reinen E/A-Zubringerbussen (zum Beispiel Interbus-S, ASI) wird der zyklische Datentransfer bevorzugt, da der SPS-Programmierer dann ohne nähere Kenntnis der Feldbusstechnik seine Daten wie bisher direkt aus einem Prozessabbild im E-/A-Bereich entnehmen kann.

Als Beispiel für eine Feldbus-/Hostankopplung sei an dieser Stelle die Baugruppe CP443-5 Extended genannt, die eine SIMATIC S7-400 an den Profibus ankoppeln kann (Abb. 1.59). Dabei können, je nach Ausstattung der beteiligten Busteilnehmer, verschiedene Busprotokolle, die auf dem Profibusstandard (EN 50170, Volume 2) basieren, eingesetzt werden. Im Einzelnen handelt es sich dabei um Profibus DP, FDL, FMS bzw. die SIEMENS-eigene Definition der Schicht 7, die S7-Funktionen.

Bei der Kopplung mit einem Profibus-DP-Netz werden auf Basis der genormten Kommunikationsdienste Daten über Aufruf- und Quittungstelegramme ausgetauscht. Der nach dem Zugriffsverfahren „Token Passing mit unterlagertem Master-Slave“ gerade aktive Master führt mittels dieser Dienste den zyklischen Datentransfer (polling) mit den ihm zugeteilten passiven Busslaves durch. Dabei verwendet der Master beispielsweise den SRD-Dienst (Send and Request Data with Reply) oder den Dienst SDN (Send Data with no Acknowledge), um mit einem Slave zu kommunizieren. Bei der Kommunikation über den SRD-Dienst enthält die Quittung die Antwortinformation des Slaves.

Intern tauscht das AG die Informationen mit dem Kommunikationsprozessor über SFCs (Systemfunktionen) aus. Über die SFC7 kann durch einen intelligenten Slave beim DP-Master ein Alarm ausgelöst werden. Ein aufgetretener Alarm kann über den OB40 ausgewertet werden. Die Funktion SFC13 ermöglicht das Auslesen der Diagnosedaten

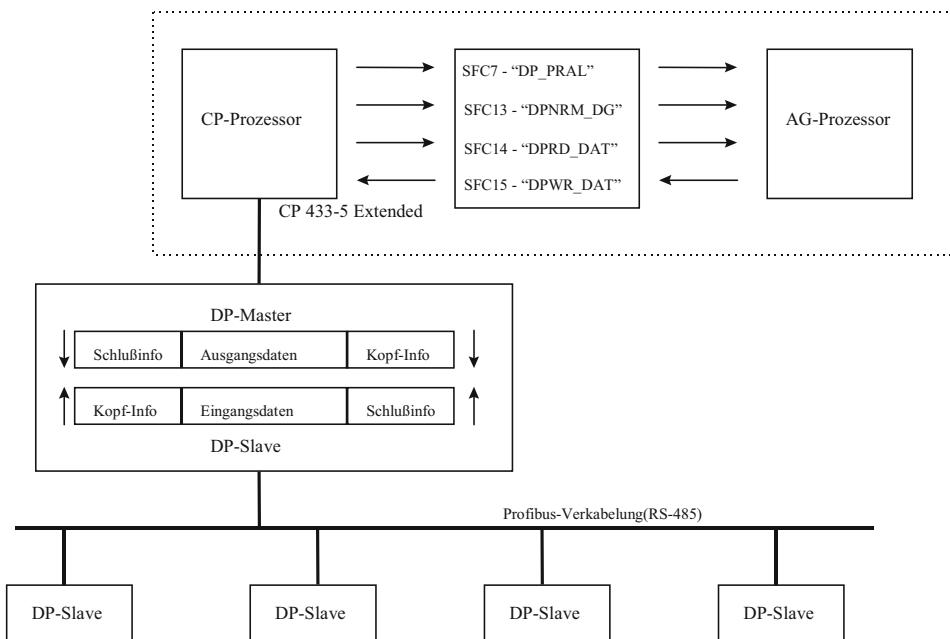


Abb. 1.59 Ankopplung einer SPS an den Profibus am Beispiel SIMATIC S7

eines Normslaves. Die Daten werden dabei in einen durch RECORD aufgespannten Zielbereich geschrieben. Mit SFC14 können Daten bzw. Variablen von einem passiven Busteilnehmer gelesen werden. Der Zielbereich für die Daten ist ebenfalls ein durch RECORD definierter Bereich. Auf einen Slave geschrieben wird über die Funktion SFC15. Die Daten für die Übertragung müssen zuvor im Quellbereich, der durch RECORD aufgespannt wird, abgelegt werden.

Die jeweils gewünschte Funktion wird durch Ihren Aufruf eingeleitet. Dabei werden Empfangsdaten direkt aus dem E/A-Bereich der Peripherie gelesen, Sendedaten werden direkt in den E/A-Bereich der Peripherie geschrieben. Die Einstellung der Busparameter, die bei jedem Teilnehmer identisch sein müssen, erfolgt bei dem CP433-5 Extended über die Projektierung der Baugruppe.

1.10.2.3 Einheitliche Programmierung mit IEC 1131

Zur Vereinheitlichung der Programmierung von SPS unabhängig vom konkreten Gerätehersteller arbeitet IEC an der Normenreihe IEC 1131-X. Der Teil IEC 1131-3 wurde 1993 als Norm veröffentlicht und kurz darauf unter EN 61131-3 von der CENELEC als europäische Norm, die damit automatisch auch als DIN-Norm erschienen ist, übernommen. Dieser Teil definiert die Programmiersprache selbst, wobei die prinzipiellen Programmierarten Ablaufsprache (AS), Anweisungsliste (AWL), Kontaktplan (KOP), Funktionsbaustein-Sprache (FBS) und eine den Hochsprachen der PCs sehr ähnliche

Sprache, der so genannte Strukturierte Text (ST) als gleichberechtigt nebeneinander verwendbar aufgeführt werden. Weitere Elemente moderner Hochsprachen, wie die Möglichkeit der Task-Deklaration, wurden ebenfalls aufgenommen.

Für den Feldbusbereich relevant ist der Teil IEC 1131-5, der die Sprachelemente zur Kommunikation vereinheitlicht. Unter Kommunikation wird in diesem Normteil sowohl der Datentransfer innerhalb eines Programms als auch zwischen mehreren gleichzeitig ablaufenden Programmen und zwischen verschiedenen, an einem Bus angeschlossenen Geräten verstanden. Für jeden Fall stehen entsprechende Funktionsbausteine zur Verfügung.

Alle SPS-Hersteller bieten die in IEC 1131-X definierten Sprachelemente an. Die meisten von ihnen haben sich dazu in der Nutzerorganisation PLCopen zusammengeschlossen, um am Markt die Durchsetzung dieser Norm zu unterstützen.

1.10.3 PC-Ankopplung

Aufgrund des steigenden Bedarfes nach höheren Verarbeitungsfunktionen im Prozessbereich wie Visualisierung, Datenauswertung, Datenspeicherung etc. werden in stark ansteigendem Maße PCs auch im Feldbusbereich eingesetzt. Neuere Trends wie die Integration von SPS-Funktionen in PCs über entsprechende Einsteckkarten bzw. Softwaremodule (oder umgekehrt) brachten die Entwicklung einer großen Anzahl von Produkten zur Ankopplung eines PCs an den Feldbus.

1.10.3.1 Hardware-Aspekte

Analog den Techniken bei SPS haben sich im PC-Bereich ähnliche Varianten herausgebildet:

- Ankopplung über Einsteckkarten

Die Karten weisen ähnliche Funktionalitäten wie die entsprechenden Baugruppen der SPS auf. Während allerdings bei SPS-Baugruppen die Schicht 7-Funktionalität meist bereits enthalten ist (vgl. oben), gibt es bei PC-Einsteckkarten in etwa eine gleichmäßige Aufteilung der verfügbaren Produkte auf solche, die keinerlei Schicht 7-Funktionen enthalten und solche, die Teilmängel bzw. den gesamten Umfang der Schicht 7 implementiert haben. Beiden gemeinsam ist die Tatsache, dass ein auf der Einsteckkarte vorhandener Mikroprozessor die Aufgaben der Schicht 2 übernimmt. Bezüglich der internen Ankopplung der Karte an das Prozessorsystem des PCs sind drei Varianten – häufig auch innerhalb einer Karte realisiert – möglich: Reines Polling durch E/A-Abfragen des Prozessors, interruptgesteuerter Betrieb, DMA-Betrieb.

- Ankopplung über serielle/parallele Schnittstelle

Technisch sehen diese Varianten genauso wie die entsprechenden bei der SPS beschrieben aus (vgl. oben). Starke Verbreitung findet diese Technik vor allem im Laptop-Bereich, da hier in der Regel keine Erweiterungsplätze für Einsteckkarten

vorhanden sind. Die erhältlichen Koppelmodule sind meist in einem sehr kompakten Steckergehäuse untergebracht, das sich direkt auf den Stecker der entsprechenden PC-Schnittstelle aufstecken lässt. Da die serielle Schnittstelle eine maximale Datenrate von 115 000 Bd übertragen kann, sind in den externen Adapters meist Ringspeicher eingebaut, welche die Pufferung der Daten übernehmen.

- Ankopplung über PCMCIA-Karte

Ebenfalls im Laptop- (bzw. Sub-Laptop-) Bereich angesiedelt sind Einsteckkarten auf PCMCIA-Basis. Diese sind bislang nur vereinzelt für Feldbusse erhältlich und noch vergleichsweise teuer. Der interne Aufbau entspricht dem konventioneller Einsteckkarten, naturgemäß müssen aber entsprechend flache Sonderbauformen der Bauteile auf der Platine verwendet werden.

1.10.3.2 Techniken des Anwenderzugriffs

Die Art und Weise, wie der Anwender auf obige Feldbus-Ankoppelhardware zugreift, hängt sehr von dem verwendeten Betriebssystem ab. Aus diesem Grunde werden nachfolgend die am häufigsten eingesetzten Systeme erklärt.

DOS-basierte Systeme

DOS-basierte PC-Systeme erlauben stets nur die Ausführung eines Programmes zur selben Zeit. Deshalb weisen auch die verfügbaren Feldbus-Ankoppel-Mechanismen einen „linearen“ Charakter auf. Prinzipiell können zwei Arten des Anwenderzugriffs unterschieden werden:

Beim Programm-impliziten Verfahren enthält der Programmcode des Anwendungsprogramms direkt entsprechende Sequenzen, um die mit dem Feldbus in Verbindung stehenden Hardware-Komponenten des PCs anzusprechen. Dieses Verfahren muss jeweils anwendungsprogrammspezifisch implementiert werden. Zur Unterstützung des Anwendungsprogrammierers liefern die Hardware-Hersteller linkfähige Objektcode-Dateien mit, die dokumentierte Prozeduren (Initialisieren, Lesen, Schreiben, ...) enthalten. Diese Dateien sind meist konform zu den gebräuchlichsten Entwicklungs-Tools für PCs wie Borland-C, Borland-Pascal, Microsoft-C etc. Je nach Funktionalität der zugrunde liegenden Hardware übernehmen diese Treiber die Funktionen der Schicht 7. Die zur Verfügung gestellten Hochsprachen-Prozeduren sind aber je nach Hersteller nicht unbedingt identisch mit den in der Feldbuspezifikation aufgeführten Schicht 7-Diensten, da öfter im Sinne eines möglichst einfachen Anwenderzugriffs nur „komprimierte“ Dienste angeboten werden.

Demgegenüber erlaubt das Verfahren des speicherresidenten Treibers die Entwicklung von Software-Modulen zur Feldbusankopplung, die von mehreren dafür ausgelegten Programmen angesprochen werden können. DOS-bedingt kann aber stets nur ein Programm aktiv arbeiten. Der eigentliche Programmcode zur Ansteuerung der Feldbus-Hardware ist dabei immer analog zu anderen Treibern (wie dem Maustreiber) im Arbeitsspeicher des PCs geladen. Der Aufruf entsprechender Prozeduren des Treibers erfolgt über einen so genannten Software-Interrupt des Anwendungsprogrammes. Dieser Interrupt kann von allen gängigen Hochsprachen-Programmen sehr leicht generiert werden. In entsprechenden Re-

gistern des PC-Prozessors werden die Parameter und Daten übergeben. Der Anwendungsprogrammierer ist dabei unabhängig von der konkreten Entwicklungsumgebung, die der Entwickler des Treibers zur Verfügung hatte. Er muss lediglich die Aufrufkonventionen kennen. Auch diese Art von Treibern wird von den meisten Hardware-Herstellern mitgeliefert. Dem Vorteil der größeren Offenheit des Systems stehen jedoch Geschwindigkeits-Einbußen gegenüber.

Ein genereller Nachteil aller DOS-basierten Systeme liegt darin, dass keine standardisierten Schnittstellen zwischen Treibern und Anwendungsprogrammen existieren. Ein Projektierer, der ein bestimmtes Anwendungsprogramm – beispielsweise ein Visualisierungsprogramm eines bestimmten Herstellers – mit einem ebenfalls vorgegebenen Feldbus verbinden will, ist darauf angewiesen, dass der Hersteller des Anwendungsprogramms gerade für diesen Feldbus (und bestimmte Feldbus-Hardwareankopplungen) einen Treiber mit anbietet. Kundenspezifische Treiberentwicklungen werden typischerweise erst zu Preisen ab ca. 7000,- € angeboten.

Windows-basierte Systeme

Zunehmend wird auch in der Automatisierungsindustrie Windows als Betriebssystem eingesetzt. Windows als hardwareabstraktes Betriebssystem, die Forderung der Industrie nach einer immer kürzer werdenden Zeit „Time to market“ und immer schneller werdende Innovationszyklen machen den Einsatz von Treibern mit offenen und standardisierten Schnittstellen notwendig (Abb. 1.60). Mit modernen Visualisierungstools wie InTouch, LabView oder Hochsprachen wie VisualBasic und Delphi kann auch ein nicht allzu versierter Programmierer in kürzester Zeit eine Applikation auf Basis fertiger Treiber erstellen.

Bei diesen Client-/Serverstrukturen kommen für die Interprozesskommunikation Mechanismen wie DLLs (*Dynamic Link Library*), DDE (*Dynamic Data Exchange*), OLE (*Object Linking and Embedding*) und OPC (*OLE for ProcessControl*) zum Einsatz.

Eine DLL ist eine Bibliothek von Funktionen die unter Windows von verschiedenen Programmen dynamisch gebunden werden kann. Sie wird meist bei geschwindigkeitsrelevanten Zugriffen und schnellen Algorithmen oder als Interface zu Gerätetreibern eingesetzt. Die Definition der Schnittstelle wird dabei durch den Entwickler vorgegeben. Mit den oben genannten objektorientierten Programmiertools ist es mit wenigen Zeilen Programmcode möglich, eine DLL einzubinden. Bei DDE, OLE und OPC ist das Interface fest definiert.

DDE wird ab der Windowsversion 3.x zur Interprozesskommunikation eingesetzt. Der Zugriff wird dabei über drei Elemente genau fixiert. Dies sind die Elemente Service, Topic und Item. Der Informationsaustausch wird über Nachrichten getätig, was dem Versenden von Post ähnlich ist. Vergleicht man „Service“ mit einer Stadt, so entspricht „Topic“ einer Straße und „Item“ einem Haus.

Mit den Versionen von Windows 95/NT wird der Einsatz von OLE immer weiter forciert. Für OLE wird oft auch der Begriff ActiveX verwendet. Bei diesem Zugriffsverfahren ist es neben der einfachen Verbindung (*link*) möglich, Sourcecode einer Anwendung in

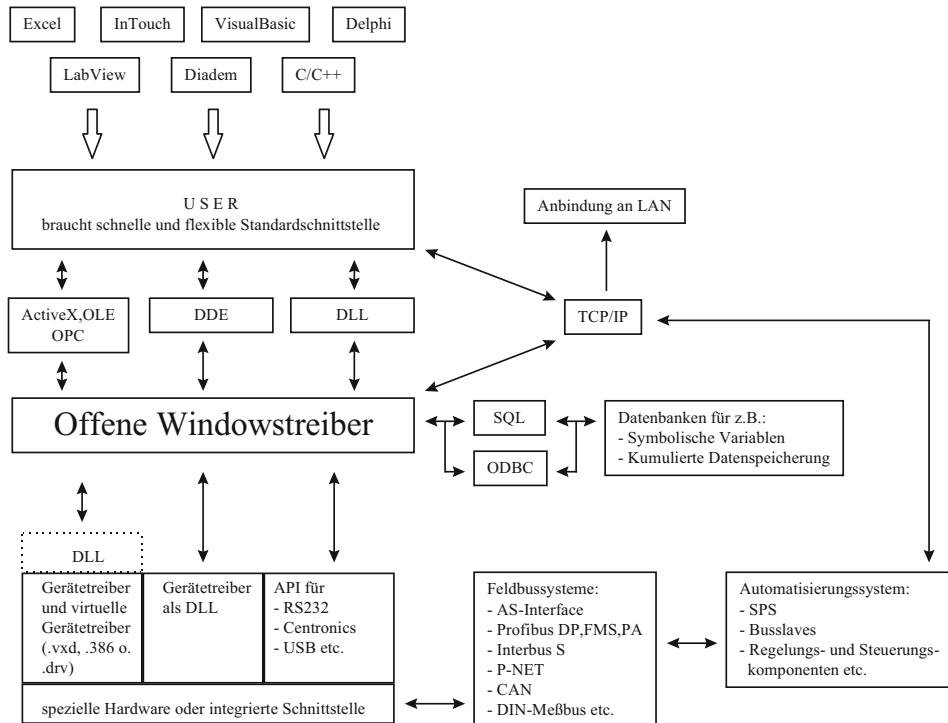


Abb. 1.60 Kommunikationstopologie unter Windows

eine andere einzubetten (*embedding*). Der Informationspfad wird dabei über folgende Elemente fixiert:

Der Ort der Information wird über den Namen der Applikation und die zu ihr gehörige Objektklasse spezifiziert. Diese beiden Parameter werden in der Windowsregistry eingetragen, auf die der OLE-Prozess zugreift. Innerhalb der Objektklasse stehen verschiedene Properties (Eigenschaften) und Methoden (Funktionen) zur Verfügung, auf die die aufrufende Anwendung dann zugreifen kann.

Ein Beispiel für eine solche Anwendung stellt der externe ActiveX-Server „S7_NET“ der Firma b-plus dar (Abb. 1.61). Mit dieser Software ist es möglich, über nur vier Properties eine Kommunikation mit einer SIMATIC S7-300/400 aufzubauen. Wahlweise kann der Informationsaustausch über Profibus oder Industrial Ethernet stattfinden.

Der Anwender greift in diesem Fall auf eine sehr einfache Schnittstelle zu und braucht sich nicht mit der zum Teil komplexen Bustopologie auseinanderzusetzen.

Neben externen Servern gibt es auch interne Server. Ein Beispiel für einen solchen Server wäre ein Controlpanel, das als fertiges Modul in eine Anwendung eingebunden wird. In diesem Zusammenhang wird auch vom Begriff der Komponente gesprochen.

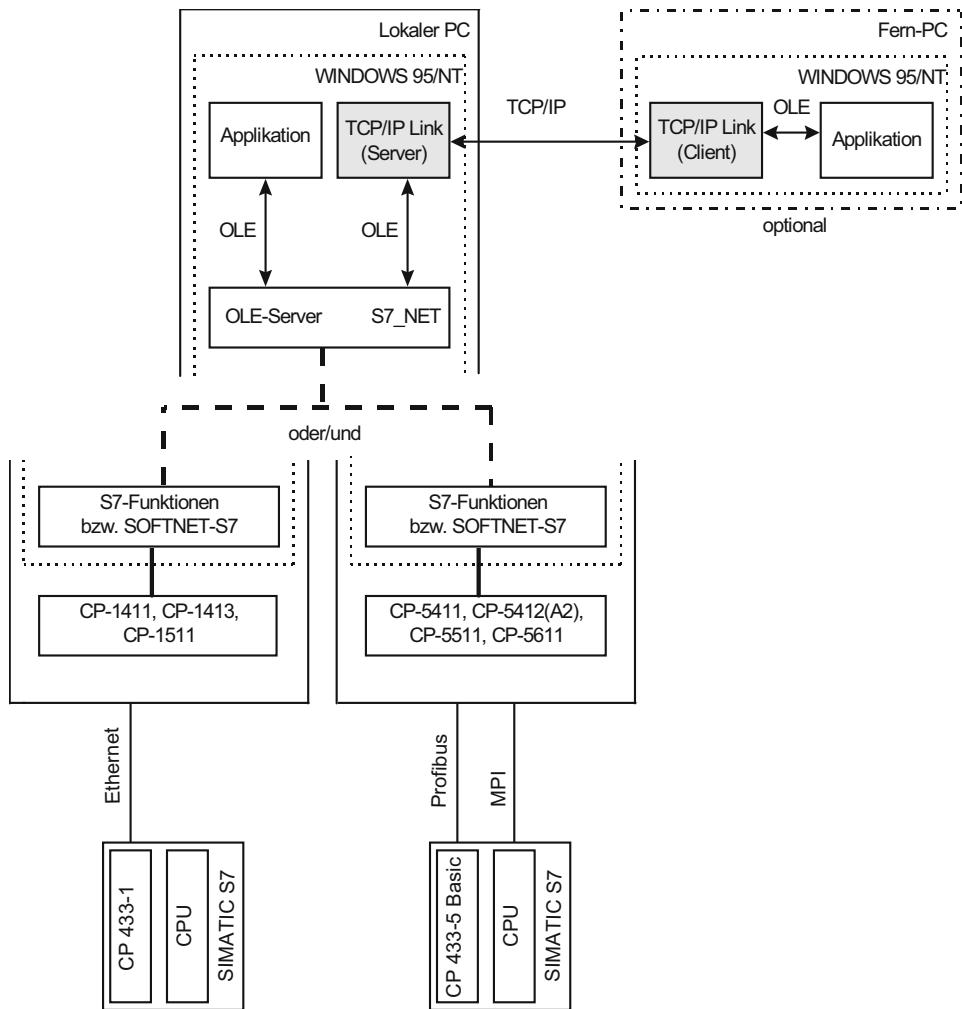


Abb. 1.61 OLE-Treiber am Beispiel SIMATIC S7 PC-Anbindung

Die OPC-Foundation hat einen neuen Standard für die OLE/COMbasierende Interprozesskommunikation definiert. Dabei umfasst ein Server als Container mehrere Gruppen von Informationseinheiten. In jeder Gruppe befinden sich Items, welche die tatsächliche Information bereitstellen. Ein oder mehrere Clients können auf das „Specific OPC Custom Interface“ oder auf das optionale „OPC Automation Interface“ zugreifen, um Daten vom Server zu bekommen. Der Client generiert eine Referenz auf das Serverobjekt. Das Gruppenobjekt wickelt den Informationsaustausch zwischen der Kommunikationsschnittstelle und den Items ab. Das Itemobjekt hält Informationen über die aktuellen Parameterwerte, Status-, Zeitstempelinformationen etc. bereit.

Der OPC-Standard ist nach den Kriterien der Offenheit und entsprechend der objekt-orientierten Programmierung aufgebaut. Der Benutzer erhält eine leicht implementierbare Schnittstelle zum Server. Auf diese Weise folgt der Standard dem Trend der Betriebssysteme, den Hardwarezugriff vom User logisch zu trennen.

1.10.4 Controller-Ankopplung

Insbesondere für Anwendungen, bei denen eine Vor-Ort-Bedienung sowie Signalverarbeitungsfunktionen im kleineren Umfang notwendig sind, bieten mehrere Hersteller kompakte „Computer“ an, die sich durch folgende Merkmale auszeichnen:

- kompaktes, robustes Gehäuse mit Typ IP 65
- 1 bis 2 Feldbusanschlüsse
- LC-Display, teilweise grafikfähig
- Folientastatur
- programmierbar mit PCs mit in der Regel spezieller Programmiersprache

Im Unterschied zu reinen Anzeigegeräten, die in steigender Stückzahl ebenfalls von vielen Herstellern für den direkten Feldbusanschluss schon angeboten werden, weisen solche Geräte echte Host-Funktionen auf. Sowohl unter Hardware- als auch Software-Aspekten sind die Feldbusankopplungen hierbei jedoch nicht offen gestaltet. In der Regel arbeiten die Geräte intern mit 8- bis 32-Bit-Prozessoren, die je nach Feldbus die eigentlichen Kommunikationsprotokolle der Schicht 2 selbst (typisch für zum Beispiel P-NET) oder über vorgeschaltete Controller-ICs (typisch für zum Beispiel CAN) ausführen.

1.10.5 Ankopplung an höhere Netze über Gateways

Da Gateways in einem anderen Kapitel bereits in ihrer grundlegenden Funktion detaillierter beschrieben werden, sollen hier nur kurz einige ergänzende Hinweise unter dem Gesichtspunkt des Anwenderzugriffs gegeben werden. Gateways sind insofern als Host-Systeme zu betrachten, als sie – zumindest was einen ihrer Feldbusanschlüsse betrifft – als Busmaster operieren. Die wenigen derzeit erhältlichen reinen Gateways sind meist so ausgelegt, dass sie einen Slave-Anschluss für einen übergeordneten Bus höherer Komplexität (zum Beispiel Profibus) aufweisen, während der andere als Busmaster für einen einfacheren, mehr im Sensor-/Aktorbereich angesiedelten Bus (zum Beispiel CAN) fungiert. Intern werden leistungsfähigere Prozessoren mit 16 oder 32 Bit Wortbreite benutzt, welche den Zeitbedingungen beider Busseiten genügen müssen.

Diese Gateways sind als „black boxes“ zu sehen, die über keinerlei Bedienungs- und Konfigurationsmöglichkeiten verfügen müssen.

In der industriellen Praxis viel häufiger anzutreffen sind derzeit Gateways auf PC-Basis, die jeweils eine Einsteckkarte für jede Busseite besitzen. Als Koppelsoftware dienen bei zeitkritischen Anwendungen applikationsspezifische Programme auf der Basis von Echtzeitbetriebssystemen. Ebenfalls gängige Praxis ist es, bei zeitkritischen Anwendungen, offene Windowstreiber mit OLE oder OPC-Schnittstellen einzusetzen. Diese Server geben auf dem lokalen Rechner die Daten der Hardware über OLE direkt an die Applikation weiter. Wahlweise besteht auch die Möglichkeit, die Daten über LAN an einen Remote-PC weiterzugeben. Typischerweise werden hier Protokolle wie TCP/IP oder IPX/SPX eingesetzt.

1.10.6 Host-Zugriffe unter MMS

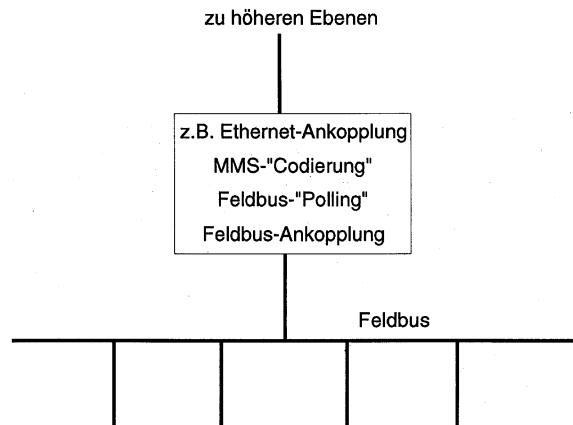
MMS (*Manufacturing Message Specification*) ist eine unter ISO/IEC 9605 bzw. DIN 66 306 standardisierte Spezifikation einheitlicher Dienste und Objekte für die Schicht 7 (*Application Layer*) von vor allem für die Fertigungstechnik tauglichen Bussystemen. Ursprünglich wurde MMS nicht für den Bereich der Feldbusse entwickelt, sondern stellte vielmehr einen wesentlichen Bestandteil von MAP (*Manufacturing Automation Protocol*) einem ursprünglich von General Motors eingeführten Buskonzept für die Vernetzung auf höheren Fabrikebenen, dar. Da MMS allerdings lange Zeit der einzige nicht ausschließlich für spezielle Applikationen entwickelte Schicht 7-Standard war, übernahmen mehrere Feldbussysteme Teilfunktionen von MMS (*subsets*). Beispiele sind FMS (*Fieldbus Message Specification*) von Profibus, PMS (Process Message Specification) von Interbus-S und RAC (*Remote Access Control*) des Bitbus.

MMS definiert im Wesentlichen:

- über 80 allgemeine Dienste für die Buskommunikation (u. a. für Verbindungssteuerung, Variablenzugriff, Ereignissesteuerung, Programm-Management etc.),
- Objekte als durch den Programmierer festzulegende abstrakte Modelle von realen Objekten (zum Beispiel Maschine, Roboter etc.),
- ein Protokoll, welches Datenformate zur Übermittlung der Parameter eines Dienstes definiert,
- einen Client-Server-Mechanismus, nach dem immer ein – durchaus wechselndes – Busgerät (Client) die Ausführung eines Dienstes bei einem anderen Gerät (Server) anfordert.

Teil 1 und Teil 2 von MMS stellen dabei den wesentlichen Kern der Spezifikation dar. Sie sind unabhängig von speziellen Applikationen. In weiteren Teilen sind so genannte *Companion Standards* mit erweiterten Festlegungen für Robot Control (Teil 3; Status IS), Numeric Control (Teil 4; Status IS), Programmable Controller (Teil 5; Status CD) und Process Control (Teil 6; Status IS) enthalten. Weitere unter IEC 870-6-XXX genormte

Abb. 1.62 MMS bei der Ankopplung von Feldbussen an Host-Systeme



Companion Standards für MMS betreffen die Kommunikation zwischen Einrichtungen der öffentlichen Versorgungsunternehmen.

Neben der oben erwähnten Tatsache, dass einige Feldbussysteme Untermengen aus MMS, teilweise auch mit anderen Namen, in ihre Schicht 7-Spezifikation übernommen haben, besteht ein deutlicher Trend, bei der Ankopplung kompletter Feldbusnetze an höhere Ebenen MMS zentral in den Hosts der Feldbusnetze zu unterstützen. Wie in Abb. 1.62 dargestellt, findet dabei innerhalb des Hosts eine „Codierung“ der über den Feldbus transferierten Prozessdaten in die MMS-Welt statt. Im Gegensatz zum einzelnen Feldgerät ist der Aufwand, MMS komplett oder in einer größeren Teilmenge in Hosts zu implementieren, wirtschaftlich meist akzeptabel.

1.11 Buszykluszeiten

Ein wichtiges Kriterium zur Beurteilung eines Bussystems ist – technisch und werbemäßig gesehen – die Buszykluszeit, also die Zeit, die ein Teilnehmer (slave) warten muss, bis er wieder „dran“ ist. Die Zykluszeit hängt von der Übertragungsrate des Busses, von seiner Telegrammstruktur und auch vom Anwendungsfall ab.

1.11.1 Deterministische Bussysteme

Im Folgenden werden für drei gängige, aber technisch unterschiedliche Bussysteme deterministischer Art Vergleichswerte berechnet.

Gegeben seien 30 Slaves. Im Falle 1 liefere jeder Slave 1 Byte Ausgangsdaten, im Falle 2 dagegen 10 Byte.

Für den Profibus (Abschn. 4.2.2) berechnen wir die Zykluszeit nach den Angaben in DIN 19 245 T3, für den Interbus S (Abschn. 4.2.3) nach DIN E 19 258 und für AS-I nach Abschn. 4.1.1.

Für die Zykluszeit T des Profibus gilt vereinfachend:

$$T = (476 + s \cdot (158 + d \cdot 11)) \cdot t , \quad (1.21)$$

wobei

s die Anzahl des Slaves,

d die Anzahl der Bytes/Slave und

t die Bitzeit ist.

Der einmalige Overhead ist 476 Bit, der Overhead pro Slave-Ausgabetelegramm 158 Bit und jedes Datenbyte hat noch drei Rahmenbits.

Für $s = 30$ Slaves und $d = 1$ Datenbyte pro Slave ergibt (1.21)

$$T = 5564 \cdot t .$$

Der normale Profibus arbeitet mit einer Übertragungsrate von $\text{ÜR} = 1,5 \text{ MBd}$, also $t = 0,67 \mu\text{s}$. Damit wird

$$T_{1,5} = 3,7 \text{ ms} .$$

Der schnelle Profibus arbeitet mit 12 MBd , wodurch sich

$$T_{12} = 0,46 \text{ ms}$$

ergibt.

Für $d = 10$ Datenbyte pro Slave ergeben sich entsprechende Zykluszeiten:

$$T_{1,5} = 5,7 \text{ ms} ,$$

$$T_{12} = 0,71 \text{ ms} .$$

Man erkennt, dass die Anzahl der Daten pro Slave nicht stark in die Zykluszeiten eingeht.

Für die Zykluszeit des Interbus S gilt eine ähnliche Beziehung:

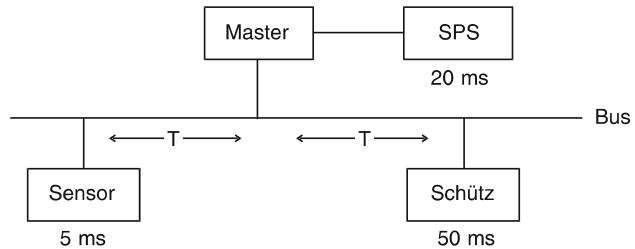
$$T = (78 + s \cdot (28 + d \cdot 13)) \cdot t . \quad (1.22)$$

Der einmalige Overhead ist 78 Bit, der Overhead pro Slave 28 Bit und jedes Datenbyte ist von 5 Bit eingerahmt.

Die ÜR des Interbus S ist $0,5 \text{ MBd}$, also $t = 2 \mu\text{s}$. Damit folgt aus (1.22) für $s = 30$ Slaves und $d = 1$ Datenbyte pro Slave

$$T_{0,5} = 2,6 \text{ ms} .$$

Abb. 1.63 Steuerungsbeispiel mit Bus



Für 10 Datenbyte pro Slave ergibt sich entsprechend

$$T_{0,5} = 9,6 \text{ ms} .$$

Für die Zykluszeit des ASI gilt

$$T = (s + 0,94) \cdot 25,5 \cdot t . \quad (1.23)$$

Da der ASI pro Slave und Zyklus unveränderlich 4 Bit transportiert, entfällt in (1.23) die Angabe von d . Die ÜR ist 0,167 MBd, also $t = 6 \mu\text{s}$. Damit aus (1.23) für 30 Slaves und 0,5 Byte Daten:

$$T_{0,17} = 4,8 \text{ ms} .$$

Man kann daraus folgern, dass für kurze Nachrichten die Busse etwa gleich schnell sind, während für lange Nachrichten der Profibus schneller als Interbus S ist.

Fraglich ist allerdings, ob in einem Steuerkreis die Buszykluszeit überhaupt die kritische Größe ist. Nehmen wir zur Beantwortung dieser Frage folgenden typischen Fall an (Abb. 1.63): Ein Sensor als Slave sendet ein 4-Bit-Telegramm an eine SPS, die daraufhin sich entschließt, über den Bus ein Schütz zu betätigen. Die Zeitbilanz sieht dann etwa folgendermaßen aus [70]:

- Sensorzeit (incl. eingebaute Verzögerung) = 5 ms.
- Buszeit Profibus 1,5 = $2 \cdot 3,7 \text{ ms} = 7,4 \text{ ms}$,
- Profibus 12 = $2 \cdot 0,46 \text{ ms} = 0,92 \text{ ms}$,
- Interbus S = $2 \cdot 2,6 \text{ ms} = 5,2 \text{ ms}$,
- AS-I = $2 \cdot 4,8 \text{ ms} = 9,6 \text{ ms}$.
- SPS-Programmausführungszeit, z. B. = 20 ms.
- Schützanzugverzögerung, z. B. = 50 ms.

Setzt man nun die Buszeit $2 \cdot T$ ins Verhältnis zur gesamten Steuerzeit, so erhält man Werte zwischen 1,6 % (Profibus 12) und 11,3 % (ASI).

Man erkennt daraus, dass die Buszeit also wohl nur in besonders zeitkritischen Fällen die ausschlaggebende Größe sein wird.

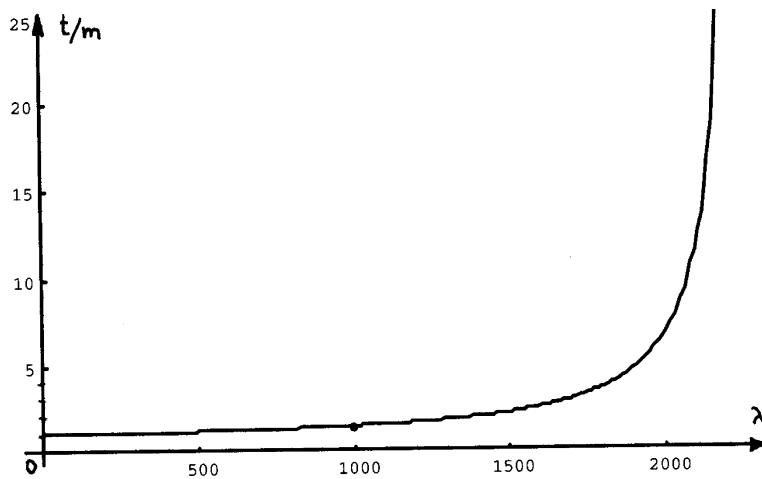


Abb. 1.64 Normierte Meldungsverzögerung t/m als Funktion der Meldungshäufigkeit λ für Telegrammdauer $m = 225,5 \mu\text{s}$

1.11.2 Nichtdeterministische Bussysteme

Das vom Zugriffsverfahren her prinzipiell nicht echtzeitfähige Ethernet (TCP/IP) dringt „von oben her“ (netzwerk hierarchisch gesehen) in den Bereich der klassischen Bussysteme der Automatisierungstechnik ein. Im Folgenden betrachten wir das Zeitverhalten des Ethernet etwas genauer (alle Werte bezogen auf die Bitzeit $0,1 \mu\text{s}$ (10 MBd)).

1. Simulation der mittleren Meldungsverzögerung bei 2 Stationen.

Beispiel

$$\begin{array}{r}
 256 \quad \text{Datenbytes} \\
 + \quad 26 \quad \text{Bytes Overhead} \\
 \hline
 = \quad 282 \quad \text{Bytes Meldungslänge}
 \end{array}$$

Meldungsdauer $m = 282 \cdot 8 \cdot 0,1 \mu\text{s} = 225,6 \mu\text{s}$.

Maximale Meldungsr率e $\lambda = 10 \text{ MBit/s: } (282 \cdot 8 \text{ Bit}) = 4433 \text{ Meldungen/s.}$

Für diesen Fall zeigt das Diagramm in Abb. 1.64 die Abhängigkeit der normierten systembedingten Verzögerung t/m von der Meldungsr率e λ .

Die Zunahme der Meldeverzögerung bei steigender Meldehäufigkeit ist offensichtlich. Man bemerkt aber auch, dass eine Lastbegrenzung die Verzögerung reduziert:

Lesebeispiel

Gewählte Lastbegrenzung auf 10 %: Meldungsr率e 443 Meldungen/s.

Das Diagramm liefert $t/m = 1,2$, also Verzögerung $t = 270 \mu\text{s}$.

Eine mittlere Meldungsverzögerung von 0,27 ms erscheint völlig akzeptabel, selbst für „harte Echtzeit“-Anforderungen. Aber 2 Stationen sind nicht die Realität, und wichtiger als der Mittelwert ist der „worst case“.

2. Mittlere und maximale Meldungsverzögerung bei 24 Stationen Für diese Simulation wurde die Meldungslänge zwischen 8 und 29 000 Nutzdatenbits variiert mit Mittelwert 2050 Bit (= 256 Byte, wie oben).

Bei 10 % Auslastung ergibt die Simulationsrechnung die mittlere Meldeverzögerung $t_{\text{mittel}} = 270 \mu\text{s}$, praktisch dasselbe wie oben. Das zeigt, dass die Stationszahl in den Mittelwert von t kaum eingeht.

Die maximale Meldeverzögerung errechnet sich bei 10 % Auslastung zu $t_{\text{max}} = 8,5 \text{ ms}$. Das ist schon ein Wert, der bei harter Echtzeit nicht zu vernachlässigen ist.

Wie bereits betont, hängt t von der Auslastung ab. Die nachfolgende Tabelle zeigt dies für unser Beispiel mit Meldungen von 8 bis 29 000 Bit Länge, Mittelwert 2048 Bit.

Lastbegrenzung auf	Meldungen/s	maximale Meldungsverzögerung
50 %	2216	294 ms
28 %	443	172 ms
10 %	443	8,5 ms
5 %	221	5,9 ms
1 %	44	3,3 ms

Man findet unter Google: „CSMACD Simulation“ interessante Animationen des CS-MA/CD-Vorgangs, z. B.:

- www.inf.fu-berlin.de/lehre/SS00/19540-V/projekte/bunge_schulz/csmacd.html.

1.12 Sicherheitsbussysteme

Das Hauptaugenmerk der Steuerung und der daran angeschlossenen Bussysteme liegt auf der schnellen und präzisen Steuerung der angeschlossenen Maschine. Es liegt im Interesse des Maschinenherstellers, Fehler in der Technik, z. B. durch defekte Kabel, defekte Sensoren etc., zu erkennen und zu beherrschen, ohne das es zu Schäden an Werkstücken oder der Maschine kommt. Wenn aber eine Fehlfunktion der Steuerung (inklusive Sensoren, Aktoren, Steuerung und Bussystem) für Menschen gefährlich werden kann, gelten andere technische Regeln, die eine geprüfte Strategie zur Beherrschung von Fehlern beinhalten müssen.

Typischerweise wird eine Maschine dabei so gestaltet, dass es eine Steuerung für die Automatisierung des gewünschten Prozesses gibt und davon unabhängige Sicherheitstechnik. Die Sicherheitstechnik stellt im einfachsten Falle sicher, dass während die Maschi-

ne arbeitet, keine Menschen im Gefahrenbereich anwesend sind. Das kann z. B. durch Schutztüren oder Sicherheitslichtschranken geschehen.

Nach europäischem Recht („Maschinenrichtlinie“) muss der Maschinenhersteller dazu eine so genannte Gefahrenanalyse durchführen und entsprechend dem Ergebnis der Analyse die Maschine durch entsprechend zugelassene Sicherheitsausrüstung absichern. Stark vereinfacht dargestellt wird die sicherheitstechnische Qualität der Sicherheitsausrüstung je nach angewandter Norm durch eine „Anforderungsklasse“ (AK), „Safety Integrity Level“ (SIL) oder „Performance Level“ (PL) charakterisiert. Je höher die Einstufung, desto unwahrscheinlicher gefährliche Fehler, was vom Hersteller der Ausrüstung in aufwändigen statistischen Verfahren nachgewiesen wird. Beispielsweise muss für SIL3 die Wahrscheinlichkeit eines gefährlichen Versagens weit niedriger als einmal in 1000 Betriebsjahren liegen.

Diese nachgewiesene Zuverlässigkeit wird konventionell durch Redundanz und Selbsttests sowie die gezielte Verwendung von sehr gut bewährten Schaltungsprinzipien erreicht. Lange Zeit hat diese bewusst konservative Haltung den Einsatz von Bussystemen ausgeschlossen. Seit einigen Jahren aber haben sich die Haltung der Prüfstellen und damit verbunden auch die Normenlage so verändert, dass auch Sicherheitseinrichtungen über Bussystem kommunizieren können. Eine Darstellung dieser Technologien würde den Rahmen dieses Buches sprengen, daher sollen hier nur einige Aspekte angerissen werden.

Für fast jedes Standard Feldbus System gibt es mittlerweile eine Erweiterung, die es für den Einsatz in Sicherheitsanwendungen, üblicherweise bis SIL3, ertüchtigt. Die spezielle Ausprägung ist dabei jeweils unterschiedlich, es gibt aber gemeinsame Grundlagen.

Der Einsatz von Bustechnik führt dazu, dass neue Fehlerquellen existieren, die es bei konventioneller Verdrahtung nicht gibt. Diese Fehlerquellen müssen durch geeignete Maßnahmen beherrscht werden. Folgende Bussystem-typische Fehlerquellen werden angenommen:

- Durch einen Fehler sendet ein Busteilnehmer alte Nachrichten.
- Ein Busteilnehmer sendet eine Nachricht nicht.
- Ein Busteilnehmer fügt eine Nachricht ein.
- Auf dem Bus werden Nachrichten in ihrer Reihenfolge verändert (z. B. in Ethernet Switches).
- Nachrichten auf dem Bus werden verfälscht.
- Eine sicherheitsrelevante Nachricht wird verzögert.
- Durch einen Fehler werden sicherheitsrelevant und nicht sicherheitsrelevant Daten vermischt.

Diese Fehler lassen sich beherrschen durch Maßnahmen wie

- laufende Nummern für jede Nachricht,
- zeitliche Erwartungen,
- Sicherungsverfahren.

Allerdings muss in der Sicherheitstechnik die Wirksamkeit der Maßnahmen statistisch nachgewiesen werden. Dabei hat die Charakteristik des einzelnen Bussystems (Telegrammlänge, Sicherungsmaßnahmen, Geschwindigkeit etc.) sehr starken Einfluss auf das Gesamtergebnis. Das Design einer sicherheitstechnischen Erweiterung für ein bestimmtes Bussystem sowie der rechnerische Nachweise seiner Wirksamkeit sind extrem aufwändig. Für fast alle Sicherheitsbussysteme wurde von der Interessengruppe zunächst die Technik erarbeitet und als Konzept zugelassen. Für jedes einzelne Gerät ist dann nur noch nachzuweisen, dass die im Konzept beschriebene Technik korrekt umgesetzt wurde, um die angestrebte Zuverlässigkeit zu erreichen.

Die meisten sicheren Feldbusse haben nur eikanalige Feldbus-Schnittstellen, um die vorhandenen Feldbus ASICs nutzen zu können. Die Sicherheitsfunktionen werden dabei in der Schicht 7 realisiert, der Feldbus wird als „schwarzer Kanal“ ohne eigene Sicherheitsfunktion betrachtet.

Die meisten sicheren Feldbusse ermöglichen die Kommunikation von nicht-sicheren und sicheren Teilnehmern auf dem gleichen Medium. Das ist hinsichtlich des Projektierungs- und Verkabelungs-Aufwands sehr vorteilhaft, muss aber im Konzept des Feldbusses berücksichtigt sein, da ja auch die unsicheren Teilnehmer Fehler erzeugen können, die die Sicherheit nicht gefährden dürfen.

<i>Sicherheit gegen:</i> Verfälschung, Verlust, falscher Adressierung der Telegramme. Beispiele: Notaus-Taster, Sicherheitslichtschranke	Explosion in explosionsgefährdeten Bereichen. Gasverteiler, Tankanlage
<i>Kategorien:</i> Sicherheits-Kategorien 1...4 z. B. Kat. 3: Schutz bei 1 Fehler	Ex-Zonen 0...2 z. B. Zone 0: Immer Gefahr
<i>Abhilfe:</i> Numerierung, Passwort, Watchdog	Schutzarten q (Sandfüllung) e, m (Verguss), i (Eigensicherheit)
<i>Normen:</i> EN 954-1, DIN VDE 0801 u. a.	EN 500154, DIN EN 60019-10 u. a.
<i>Zertifizierung:</i> TÜV, BIA u. a.	TÜV, PTB u. a.
<i>Sicherheitsbusse:</i> AS-i Safety at Work Profisafe auf Profibus Profisafe auf Profinet INTERBUS-Safety u. a.	Profinet PA, AS-I ex, u. a.

Bei sicheren Bussystemen gibt es einige besondere Komponenten, die es in den Standard-Bussystemen nicht gibt:

- Sichere Steuerung (SPS): Sie verarbeitet die Daten der sicheren Sensoren und schaltet die sicheren Aktoren. Arbeitet häufig auch die sichere Protokollerweiterung des jeweiligen Feldbusses ab, sodass die sichere Steuerung über einen nicht-sicheren Feldbus-Master kommunizieren kann.
- Sichere Master: Hier ist die Master-Baugruppe sicher ausgeführt. Ist üblicherweise eng mit der Sicherheitssteuerung gekoppelt.
- Sicherheitsmonitor: Eigenständige sichere Einheit, die die Kommunikation auf dem Feldbus überwacht. Unabhängig von Master und Slaves. Um eine gefährliche Bewegung der Maschine zuzulassen (z. B. Anlaufen einer Presse), muss der Sicherheitsmonitor zustimmen.

In der vorstehenden Tabelle sind die obigen Darlegungen in der linken Spalte zusammengefasst. In der rechten Spalte soll demgegenüber der ganz anders gelagerte Aspekt der Sicherheit in explosionsgefährdeten Anlagen dargestellt werden.

Für Busse ist nur die Schutzart „i“ interessant: Man begrenzt dabei die im Bussystem gespeicherte elektrische Energie auf einen Wert, der selbst bei einem Kurzschluss noch nicht imstande ist, einen zündfähigen Funken zu erzeugen (i – intrinsic safety). Dies bedeutet eine maximale Spannung kleiner 30 V und einen maximalen Strom kleiner 150 mA im Bus (incl. der Stromversorgung von Sensoren/Aktoren).

Literatur

Literatur zu Abschn. 1.1

1. Bernhard Walke: Datenkommunikation 1, Teil 1: Verteilte Systeme, ISO/OSI Architekturmodell und Bitübertragungsschicht, Hüthig-Verlag, Heidelberg 1987
2. Bernhard Walke: Datenkommunikation 1, Teil 2: Sicherungsprotokolle für die Rechner-Rechner-Kommunikation, Lokale Netze und ISDN Nebenstellenanlagen, Hüthig-Verlag, Heidelberg 1987
3. Halling (Hrsg.): Serielle Busse. Neue Technologien, Standards, Einsatzgebiete, VDE-Verlag GmbH, Offenbach 1987, I Technische Grundlagen
4. W. Kriesel, T. Heimböld, D. Telschow: Bustechnologien für die Automation, Hüthig GmbH, Heidelberg 1998

Literatur zu Abschn. 1.2

5. L. T. Gorys: TCP/IP, Hüthig, Heidelberg 1991
6. ISO International Standard 7809–1984(E): Information processing systems – Data communication – High-level data link control procedures – Consolidation of classes of procedures. First Edition International Organization for Standardization, New York 1984
7. ISO International Standard 7809:1984/Add.1:1987(E)/Add.2:1987(E): Information processing systems – Data communication – High-level data link control procedures – Consolidation of

- classes of procedures, Addendum 1 International Organization for Standardization, New York 1987
8. ISO International Standard 4335:1987(E): Information processing systems – Data communication – High-level data link control elements of procedures, Third Edition International Organization for Standardization, New York 1987
 9. M. T. Rose: The Open Book: A Practical Perspective on OSI+, Prentice-Hall, Englewood Cliffs, New Jersey, 1990
 10. F. Furrer: Ethernet TCP/IP für die Industrieautomation, Heidelberg 1998

Literatur zu Abschn. 1.3

11. Bernhard Walke: Datenkommunikation 1, Teil 1: Verteilte Systeme, ISO/OSI Architekturmodell und Bitübertragungsschicht, Hüthig-Verlag, Heidelberg 1987
12. Bernhard Walke: Datenkommunikation 1, Teil 2: Sicherungsprotokolle für die Rechner-Rechner-Kommunikation, Lokale Netze und ISDN Nebenstellenanlagen, Hüthig-Verlag, Heidelberg 1987
13. Halling (Hrsg.): Serielle Busse. Neue Technologien, Standards, Einsatzgebiete, VDE-Verlag GmbH, Offenbach 1987, 1 Technische Grundlagen
14. W. Kriesel, T. Heimböld, D. Telschow: Bustechnologien für die Automation, Hüthig GmbH, Heidelberg 1998
15. PROFIBUS Nutzerorganisation (Hrsg.), PROFIBUS, Technische Kurzbeschreibung; Karlsruhe 1999
16. Manfred Popp: PROFIBUS-DP, Grundlagen, Tips und Tricks für Anwender, Hüthig-Verlag, Heidelberg 1998
17. Wolfhard Lawrenz (Hrsg.): CAN, Controller Area Network, Grundlagen und Praxis; Hüthig-Verlag, Heidelberg 1997
18. Fisher-Rosemount (Hrsg.): Feldbus, Technische Kurzbeschreibung FOUNDATION Fieldbus Technologie 1997

Literatur zu Abschn. 1.4

19. DIN 19 244, Teil 10 (IEC57(Sec)67 u. IEC57(C0)40): Fernwirkeinrichtungen und Fernwirksysteme, Telegrammformate, März 1988

Literatur zu Abschn. 1.5

20. K. Bender (Hrsg.): PROFIBUS, Hanser, München, Wien 1990
21. ISO International Standard 3309:1991(E): Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures-Frame structures, Fourth Edition International Organization for Standardization, New York 1991
22. H. Kropp: Lokale Netzwerke, WEKA Fachverlage, Augsburg 1992
23. A. S. Tanenbaum: Computer-Netzwerke, Wolfram's Fachverlage, Attenkirchen 1992
24. ISO International Standard 7809–1984(E): Information processing systems – Data communication – High-level data link control procedures – Consolidation of classes of procedures. First Edition International Organization for Standardization, New York 1984
25. ISO International Standard 7809: 1984/Add. 1: 1987(E)/Add. 2: 1987(E): Information processing systems – Data communication – High-level data link control procedures – Consolidation of classes of procedures, Addendum 1 International Organization for Standardization, New York 1987

26. ISO International Standard 4335:1987(E): Information processing systems – Data communication – High-level data link control elements of procedures, Third Edition International Organization for Standardization, New York 1987
27. Romily Bowden: An overview of the HART-Protocol, Rosemount AG, 1991
28. HART Communication Foundation: HART Feld-Kommunikationsprotokoll, Veenendaal (NL) 1994
29. Armin Preuss: Smart/HART Kommunikation für Prozessmessgeräte, in: Technisches Messen 59 (1992) S. 361–366
30. www.de.Wikipedia.org

Literatur zu Abschn. 1.6

31. Bernhard Walke: Datenkommunikation 1, Teil 1: Verteilte Systeme, ISO/OSI-Architekturmodell und Bitübertragungsschicht, Hüthig-Verlag, Heidelberg, 1987
32. Bernhard Walke: Datenkommunikation 1, Teil 2: Sicherungsprotokolle für die Rechner Kommunikation. Lokale Netze und ISDN-Nebenstellenanlagen, Hüthig-Verlag, Heidelberg, 1987
33. Halling (Hrsg.): Serielle Busse. Neue Technologien, Standards, Einsatzgebiete, VDE-Verlag GmbH, Offenbach, 1987
34. Prof. Dr. H. D. vom Stein: Vorlesungsskriptum Allgemeine Nachrichtentechnik, Universität der Bundeswehr Hamburg, 1983

Literatur zu Abschn. 1.7

35. DIN 66 258 Teil 1: Schnittstellen und Steuerungsverfahren für die Datenübermittlung, Beuth Verlag GmbH, Berlin 1983
36. DIN 66 259 Teil 1: Elektrische Eigenschaften der Schnittstellenleitungen, Doppelstrom, unsymmetrisch bis zu 20 kbit/s, Beuth Verlag GmbH, Berlin 1981
37. DIN 66 259 Teil 2: Elektrische Eigenschaften der Schnittstellenleitungen, Doppelstrom, unsymmetrisch, bis 100 kbit/s, Beuth-Verlag GmbH, Berlin 1983
38. DIN 66 259 Teil 4: Elektrische Eigenschaften der Schnittstellenleitungen, Doppelstrom, symmetrisch für Mehrpunktverbindungen, Entwurf, Beuth Verlag, Berlin 1989
39. ISO 8482: Information processing systems – Data communication – Twisted pair multipoint interconnections. International Organization for Standardization, 1987
40. EIA 485: Standard for electrical characteristics of Generators and Receivers for use in balanced digital multipoint systems, Electronic Industries Association, 1983
41. Bresch, Güttler, Patzke: Übertragungssicherheit bei Feldbussen, Teil 1. Elektronik 15/1991
42. PTB-Bericht PTB-W-53: Untersuchungen zur Eigensicherheit bei Feldbus-Systemen, Physikalisch Technische Bundesanstalt, Braunschweig 1993
43. Ulrich Johannsmeyer: FISCO-Concept for I.S. Fieldbus Systems, PTB Braunschweig, Draft 6/99
44. Günter Pinkowski: Die Planung von PROFIBUS-PA-Netzen, atp 10/98, R. Oldenbourg Verlag
45. Frank Thorn: Ohne die Eigensicherheit zu berechnen, cav 12/99, Konradin Verlag
46. Fisher Rosemount Systems: Fieldbus, Technische Kurzbeschreibung, FOUNDATION Fieldbus Technologie, 1997
47. Wikipedia Ethernet, 5-PAM, Trellis-Modulation

Literatur zu Abschn. 1.8

48. VDE 0815/4.81: Bestimmungen für Installationsleitungen für Fernmeldeanlagen
49. Stoll, D.: Einführung in die Nachrichtentechnik, Berlin, 1979
50. EIA 485, Electronic Industries Association, Washington USA, 1983
51. Timmermann, C.-Ch.: Lichtwellenleiterkomponenten und -systeme, Braunschweig, 1984

Literatur zu Abschn. 1.9

- 52. H. Kropp: Lokale Netzwerke, WEKA Fachverlage, Augsburg 1992
- 53. A.S. Tanenbaum: Computer-Netzwerke, Wolfram's Fachverlage, Attenkirchen 1992

Literatur zu Abschn. 1.10

- 54. Produktinformationen unter anderem der Firmen Allen Bradley, b-plus, Berthel, Gesytec, Honeywell, i+me, ISK, MFP, Phoenix Contact, Process Data, Rosemount, Selectron, Siemens, Softing, GTM i-tec
- 55. Manufacturing Message Specification (MMS), Teile 1–6. ISO/IEC 9506/1–6 bzw. DIN 66 306/1–6
- 56. Schwarz, K: Manufacturing Message Specification (MMS). atp, Heft 7/1991, S. 369ff.
- 57. Himmelsdorfer, E.: Anbindung der Kommunikation in die PC- und SPS-Welt anhand praktischer Beispiele. In: VDI-Bericht Nr. 1123 „Vernetzung durch industrielle Kommunikation“, VDI Verlag, Düsseldorf 1994
- 58. Böttcher, J.: Feldbusse als Datenverbindung in der SPS- und IPC-Welt. Tagungsunterlagen zum Seminar „SPS und Industrie-PC“ des OTTI Regensburg, 02.03.1995
- 59. b-plus Messsysteme: Handbuch zur Software „Treibergerüst für einen OLE-Server unter Windows 95/NT“
- 60. b-plus Messsysteme: Kompendium „Das DK 3964 R-Protokoll“
- 61. b-plus Messsysteme: Studie „Der PC-Zugriff auf die S7 über MPI, Profibus und Industrial Ethernet“
- 62. Ben Ezzell/Jim Blaney: Das Programmierbuch Windows 95/NT4, Sybex Verlag 1997, S. 1161 f.
- 63. PROFIBUS Nutzerorganisation e.V. – PROFIBUS Technische Kurzbeschreibung, April 1997
- 64. SIEMENS – SIMATIC Software – System und Standardfunktionen, 1996, Kapitel 15
- 65. SIEMENS – SINEC DP-Programmierschnittstelle, 1996
- 66. OPC Foundation – OLE for Process Control – Data Access Standard Version 1.0A, 11. September 1997

Literatur zu Abschn. 1.11

- 67. Jünger, B.: Profibus contra Interbus S, in: ELEKTRONIK 21/1994
- 68. Boller, A.: Profibus DP mit 12 Mbit/s, in: industrie+elektronik, Nr. 4, 1995
- 69. Furrer, Frank: Ethernet-TCP/IP für die Industrieautomation. Heidelberg 1998
- 70. Schwartz, Mischa: Telecommunication Networks-Protocols, Modeling and Analysis. Addison-Wesley, Reading,, MA, USA, 1988
- 71. Gburzynski, Paweł: Protocol Design for Local and Metropolitan Area Networks. Prentice Hall, Englewood Cliffs, N.J., USA, 1996



Netzwerkhierarchien in der Fabrik- und Prozessautomatisierung

2

2.1 Übersicht und Spezifik der Kommunikation in der Automatisierung

Seit vielen Jahren vollzieht sich in der Automatisierungstechnik ein deutlicher Wandel bei den realisierten Kommunikationsstrukturen zwischen den einzelnen Automatisierungsgeräten. Bedingt durch die Entwicklung auf dem Gebiet der Mikro- und Optoelektronik entstanden auch neuartige Strukturen auf der Basis serieller Kommunikation, die Automatisierungssysteme in verschiedene hierarchische Ebenen gliedern (Abb. 2.1). Diese Entwicklung schließt erweiterte Einsatzmöglichkeiten sowie zusätzliche Funktionen von Automatisierungsgeräten ein. Durch die Implementierung von Mikrocontrollern in intelligente Automatisierungsgeräte kann die Funktionalität beträchtlich erweitert und die Zuverlässigkeit mittels einer Anzahl von Eigen- und Diagnosefunktionen verbessert werden.

Am Anfang dieser Entwicklung wurden zunächst linienförmige Bussysteme konzipiert, an die direkt buskoppelbare sowie intelligente Mess- und Stelleinrichtungen angeschlossen werden sollten. Bald fanden auch weitere Strukturen, wie z. B. die Ringstruktur, Interesse für die Datenübertragung zwischen den einzelnen Automatisierungsgeräten. Solche Strukturen spielen bei erhöhten Zuverlässigkeitsanforderungen eine bedeutende Rolle. Mit der Entwicklung entsprechender Geräte wurde es möglich, auch Kommunikationssysteme unterschiedlicher Leistungsbereiche miteinander zu verbinden. So erfolgte und erfolgt zusehends die Kopplung und Vermischung zwischen dem Anlagen- und Officebereich. In Abb. 2.2 ist ein solches hierarchisches Kommunikationsnetz schematisch dargestellt. Die unterschiedlichen Anforderungen an die verschiedenen Systemebenen wird beispielgebend durch die Darstellung von Reaktionszeit, Datenmenge und Übertragungshäufigkeit verdeutlicht.

Serielle Kommunikationssysteme finden seit Jahren in der Feldebene zunehmende Akzeptanz. An diese Feldbusebene, die als eigentliches Bindeglied zwischen den prozessnahen Einrichtungen der Sensor/Aktorebene und der Steuerungsebene nach Abb. 2.1 anzun-

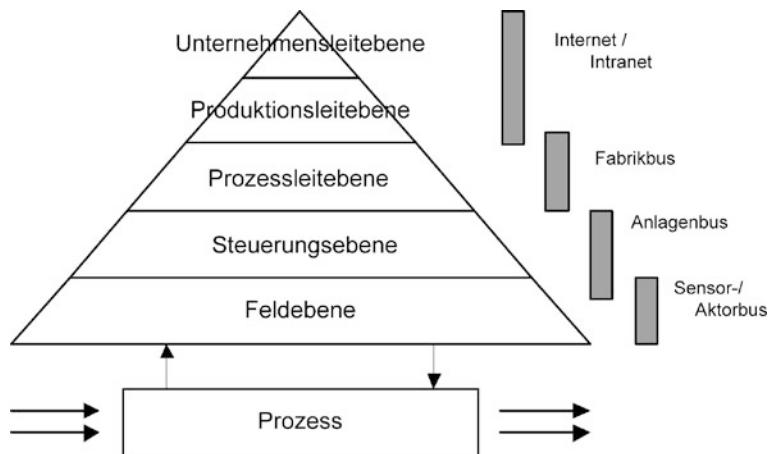


Abb. 2.1 Ebenenmodell zur Einordnung von Bussystemen in Automatisierungsstrukturen

sehen ist, werden besondere Anforderungen an das eingesetzte Kommunikationssystem gestellt. Zu diesen Anforderungen zählen:

- Echtzeitfähigkeit (determinierte Zugriffsverfahren bevorzugt)
- hohe Übertragungsgeschwindigkeit
- Zuverlässigkeit, ggf. Fehlertoleranz
- geringe Störempfindlichkeit (EMV)
- flächendeckende Topologie
- Flexibilität

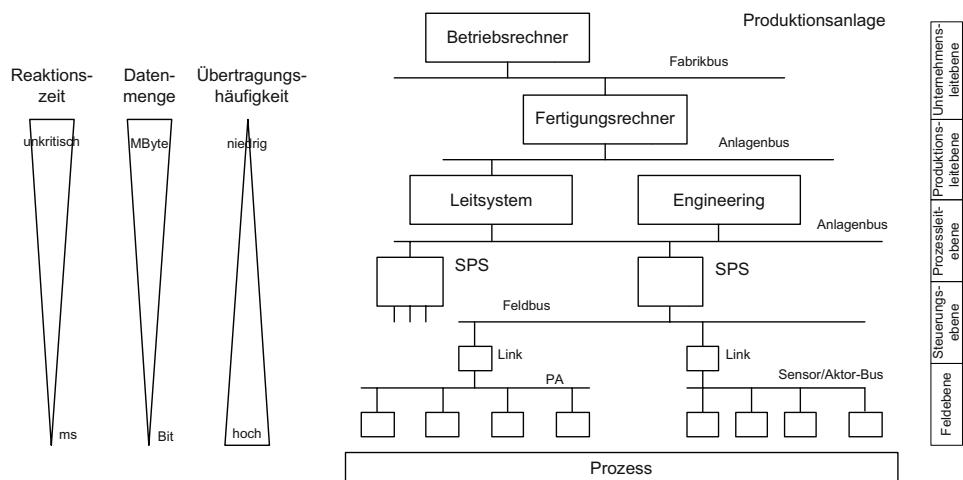


Abb. 2.2 Hierarchisches Kommunikationsnetz (Prozess- und Fertigungsautomatisierung)

- Installations- und Montagetechnik (Sensor/Aktor-Bereich)
- Wirtschaftlichkeit (Low-cost-Lösungen).

Aus den an die Kommunikation im prozessnahen Bereich gestellten spezifischen Forderungen resultiert eine hierarchische Kommunikationsstruktur in Form eines Mehrebenennetzes mit Sensor/Aktor-, Anlagen- und Fabrikbus. Die Vorteile eines solchen Feldbusnetzes in Mehrebenenstruktur liegen vor allem in der hohen Flexibilität. Dazu zählt eine weitgehend frei wählbare Topologie, eine projektierbare Zuverlässigkeit durch gestufte Redundanz und Fehlertoleranz, wählbare Reaktionszeit und damit ein anpassbares Echtzeitverhalten.

Vor allem in der untersten Automatisierungsebene, wo noch ein hoher Grad an herkömmlichen Übertragungsstrukturen zu finden ist, entsteht bei der Errichtung von Anlagen ein hoher Aufwand für Verkabelung. Durch das ständige Sinken der Bauelementekosten ist in den letzten Jahren verstärkt daran gearbeitet worden, kostengünstige Lösungen zur busförmigen Ankopplung von Sensoren und Aktoren zu finden. Derzeit auf dem Markt verfügbare Systeme sind spezifisch für unterschiedliche Anwendungen nutzbar. Besonders hohe Einsparpotentiale liegen in einer Verbesserung der Installations- und Montagetechnik, sodass insbesondere die aufwändigen Arbeiten für Abisolieren, Anlöten von Steckverbindungen, Setzen von Verteilerkästen usw. auf Baustellen entfallen. Ein weiterer wirtschaftlicher Aspekt, der oft schwer zu quantifizieren ist, ist die Diagnosefähigkeit. Da über die Lebensdauer einer industriellen Anlage Stillstandszeiten bei weitem höhere Kosten verursachen können als die Materialpreise oder Montagekosten bei Installation der Anlage, liegen in diagnosefähige Sensoren und Aktoren erhebliche Einsparpotentiale.

Der Austausch von Daten, unabhängig davon, ob es sich dabei um Messergebnisse, Systemzustände oder andere Informationen handelt, wird auf den verschiedenen Ebenen in einer sehr unterschiedlichen Art durchgeführt. Schon innerhalb der einzelnen Automatisierungsgeräte werden Daten ausgetauscht. Hier finden allerdings *parallele Verdrahtung* den Vorrang. Bedingt durch die Anforderungen (kurze Entfernung und sehr hohe Übertragungskapazitäten) ist diese Übertragungsart priorisiert. Bei dem Informationsaustausch zwischen den einzelnen Automatisierungsgeräten kommen *serielle Übertragungssysteme* zum Einsatz, die herkömmliche Übertragungsstrukturen ersetzen. Diese Übertragungssysteme reduzieren den Aufwand an Verkabelung erheblich und erhöhen, durch Implementierung von Automatisierungsgeräten mit Intelligenz, die Leistungsfähigkeit solcher Systeme (Abb. 2.3).

Für die nachfolgende Betrachtung ist besonders auch der Feldbereich relevant. In diesem Bereich werden die einzelnen Automatisierungsgeräte für verschiedene Aufgaben aus dem Automatisierungsbereich miteinander verbunden. Zur Kommunikation zwischen den einzelnen Teilnehmern kommen unterschiedliche Topologien zum Einsatz. Dabei handelt es sich um Sternstrukturen, Linienstrukturen und Ringstrukturen, wobei die beiden letztgenannten aus Zuverlässigkeitsgründen teilweise redundant ausgelegt werden. Mischformen aus den einzelnen Strukturen treten ebenfalls in der Praxis auf.

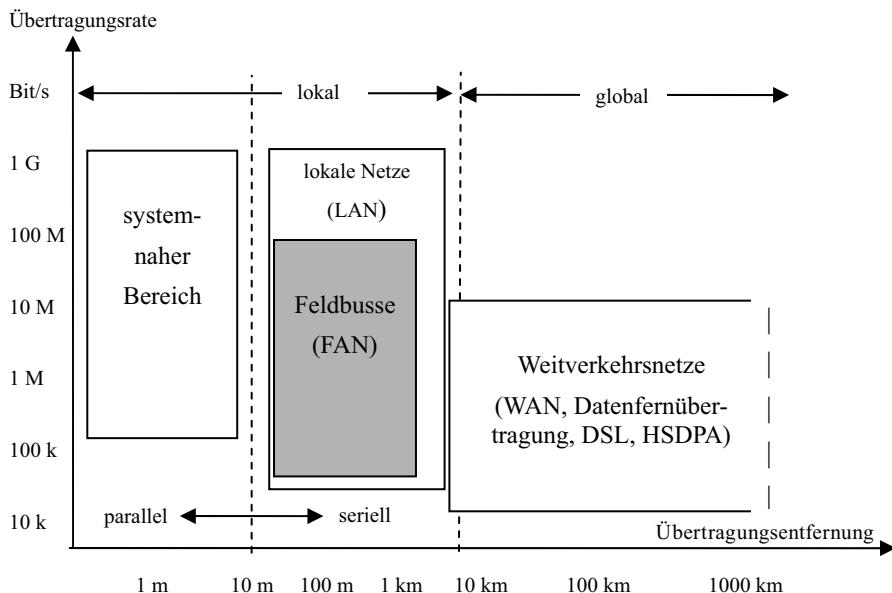
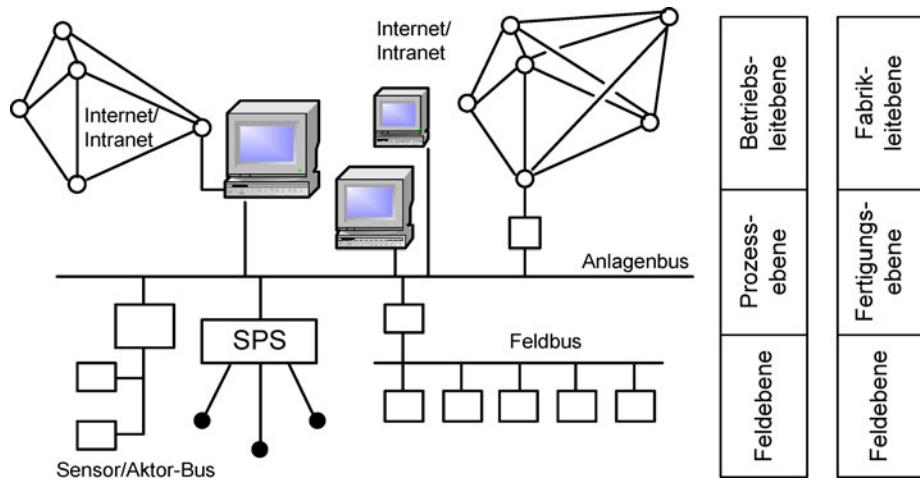


Abb. 2.3 Einordnung von Bussen und Netzen in die Parameterbereiche der Kommunikationstechnik

Die Anforderungen an ein Kommunikationssystem innerhalb des Automatisierungsreiches sind also sehr differenziert. Sie sind insbesondere auch stark abhängig vom jeweiligen Projekt (Leistungsfähigkeit, Einsatzumgebung, Kosten) und damit vom zukünftigen Einsatzbereich. Abb. 2.4 zeigt die allgemeinen Anforderungen an Feldbussysteme für den Einsatz in der Automatisierungstechnik.

Strukturelle Anforderungen	Funktionelle Anforderungen	Betreiberforderungen sowie Wirtschaftlichkeit
<ul style="list-style-type: none"> - Topographie/Topologie flächendeckend - Zuverlässigkeit der Übertragungswege - Hilfsenergieversorgung (z. B. getrennt oder gemeinsam mit Daten über die Busleitung) - Teilnehmeranzahl - Entfernungsbereiche - Durchgängigkeit 	<ul style="list-style-type: none"> - Zeitverhalten - Geschwindigkeit - Zugriffsverfahren - Kommunikationsdienste - Sicherheit der Informationsübertragung - Fehlerbehandlung (Diagnose, Fehlertoleranz) - Kommunikationsprofile - Geräteprofile 	<ul style="list-style-type: none"> - Betriebsbedingungen - Mehraufwand durch besondere Sicherheitsanforderungen (z. B. Ex-Schutz) - Standardisierung/Normung - mehrere Anbieter (Hardware, Software und Servicewerkzeuge) - Kosten des Übertragungsmediums - wirtschaftlicher Betrieb

Abb. 2.4 Anforderungen an ein Feldbussystem

**Abb. 2.5** Vereinfachte Darstellung der Kommunikationsebenen

Eine engere Struktur der industriellen Kommunikation wird durch Abb. 2.5 weiter verdeutlicht. Dabei ist zu erkennen, dass die in Abb. 2.1 dargestellten fünf Funktionsebenen eigentlich durch drei Haupt-Kommunikationsebenen realisiert werden.

Bedingt durch die verschiedenen Anwendungen in den einzelnen Ebenen, entstehen zwangsläufig differenzierte Ansprüche an die Kommunikationssysteme. Abb. 2.6 verdeutlicht die unterschiedlichen Anforderungen der prozessnahen Sensor/Aktor-Ebene zur Leitebene (Systemebene).

Verstärkt soll in den nachfolgenden Ausführungen auf die Anwendungsbereiche Fabrik- und Prozessautomatisierung eingegangen werden.

Kriterium	Anlagenbus	Sensor-Aktor-Bus
Ausdehnung des Netzwerks	ca. 1000 m	ca. 100 m
Anzahl Teilnehmer pro Netzwerk	mittel (ca. 10)	hoch (ca. 100)
Abtastzeit	ca. 10 ms bis 10 s	ca. 1 ms bis 1 s
Datenmenge pro Übertragung	8 bis einige 100 Byte	0,5 bis 8 Byte
Busstruktur	Linie (Master/Slave, Multimaster)	Linie (Master/Slave)
Zugriffsverfahren	kontrolliert (ggf. zufällig)	Polling (kontrolliert)
Übertragung auf gleicher Ebene	ja	selten
Kosten des Übertragungsmediums	niedrig	sehr niedrig
Anschlusskosten pro Busteilnehmer	ca. 100 bis 1000 €	ca. 5 bis 100 €

Abb. 2.6 Abgrenzung zwischen Bussen für den Systembereich und Sensor/Aktor-Bereich

2.1.1 Fabrikanomatisierung

Die Fabrikanomatisierung stellt einen der umfangreichsten Bereiche für die Anwendung von Feldbusssystemen dar. Die Netzwerkausdehnung liegt in der Regel unter 500 m. Im Hinblick auf die Zeitanforderungen der Kommunikation ist zu unterscheiden zwischen zeitkritischen hochdynamischen Regelungsvorgängen, die ggf. eine äquidistante Signalabtastung im 0,1 ms-Bereich benötigen, Steuerungs- und Meldeaufgaben sowie Überwachungs- oder Sicherungsfunktionen mit unterschiedlichen Zeitanforderungen.

Hier wird es häufig sinnvoll sein, unterschiedliche Bussegmente und Bussysteme einzusetzen, um durch Kombination eine wirtschaftlich optimale Lösung zu realisieren und

Anwendungs-ebene	Aufgaben/ Funktionen	Typische Geräte	Charakteristische Anforderungen	Geeignete Kommunikations-systeme
Fabrikleitebene	- Auftragsverwaltung - Konstruktion - PPS - Bestandsführung - Nachdisposition - Istdatenauswertung	- Workstations - Server - mittlere Datentechnik	- Reaktionszeiten < 10 s - große Datensätze - kommunikationsfähig mit unterlagerter Ebene - WAN-Anwendungen im Firmenverbund	- INTERNET, INTRANET, LAN
Fertigungs-leitebene (Zellebene)	- übergeordnete Steuerung von Fertigungs-zellen - Istdatenerfassung - Anlagenvisualisierung - Datenarchivierung - Stammdaten-verwaltung - Kommunikation mit Host-Rechner	- Industrieterminals - Scanner - PCs - Industrie-PCs - Server, Netzwerk-anbindung	- Reaktionszeiten bis < 100 ms - zustandsabhängige Kommunikation zwischen gleichberechtigten Teilnehmern - mittlere Datensätze wechselnder Länge - feste Teilnehmer-configuration, zusätzlich wechselnde Teilnehmer - gleichberechtigter Buszugriff für alle Teilnehmer, Querverkehr - Kommunikation mit über-/ unterlagerten Ebenen	- Anlagenbus, Fabrikbus - Lokale Netze LAN - Gateways für den Verbund unterschiedlicher Einernetze
Maschinen- und Anla-gen-ebene (Feldebene)	- Erfassen von analo-gen und digitalen Signalen - Steuerung und Reg-lung von Einzelma-schinen, Transportan-lagen, - Mess- und Identifika-tionseinrichtungen	- binäre Sensoren und Aktoren - analoge Sensoren und Aktoren - Antriebssteuerungen, Antriebstechnik - Bedien- und Anzeige-geräte - SPS/CNC/RC - Regler (unterlagert)	- Reaktionszeiten < 10 ms - zyklische Übertragung, Abtastung - kurze Datensätze - meist feste Teilnehmerkonfiguration - störsichere Übertragung, ca. 500 m - dezentraler Buszugriff - Kommunikation mit überlagerter Ebene	- Sensor-Aktor-Bussysteme - Feldbusse - hochzu-verlässige Feldbusse

Abb. 2.7 Anforderungen für die Kommunikation in der Fabrikanomatisierung

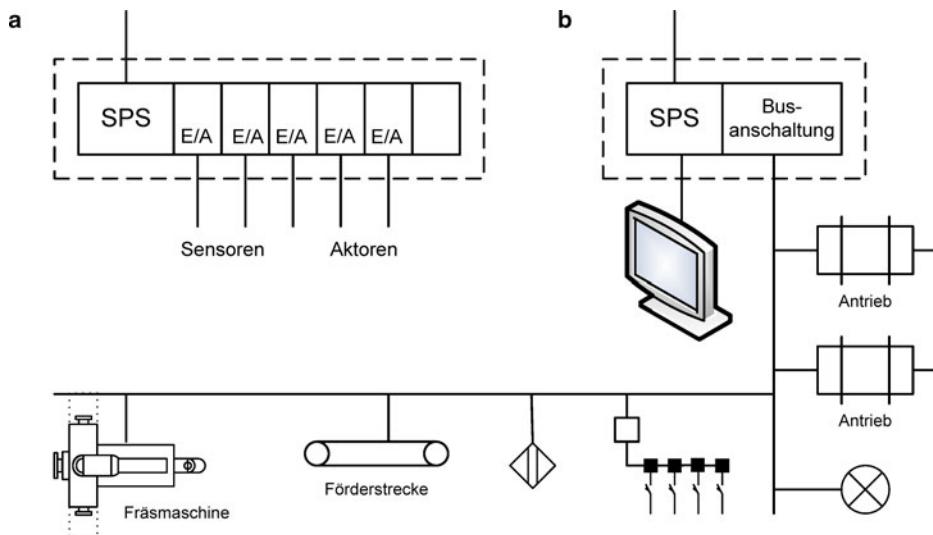


Abb. 2.8 Kommunikationsstrukturen in der Fertigungsautomatisierung. **a** zentrale Lösung, ohne Feldbus, **b** dezentrale Intelligenz, dezentrale E/A-Anschaltung, Linienbusstruktur

auch den unterschiedlichen Zeitanforderungen gerecht zu werden. Durch die Aufteilung der Aufgabenbereiche und Anforderungen gemäß Abb. 2.7 in drei Funktionsebenen mit ihren typischen Geräten und charakteristischen Anforderungen für die Kommunikation wird erreicht, dass die Feldbusauswahl sinnvoll eingeengt wird, indem zunächst die prinzipiell geeigneten Kommunikationssysteme umschrieben werden.

Zur Verdeutlichung der verschiedenen Realisierungsmöglichkeiten verteilter Automatisierungsaufgaben folgen mit Abb. 2.8 zwei Beispiele aus der Fertigungstechnik. Dabei zeigen diese Beispiele, welche Auswirkungen verschiedene Anwendungsstrukturen haben (Topologie, dezentrale/zentrale Intelligenz). Ausgehend von der zentralen Lösung (Anlage mit zentraler Steuerung) wird eine dezentrale Anordnung mit binären Sensoren, Aktoren, busfähigen Feldgeräten und komplexen Baugruppen dargestellt.

2.1.2 Prozessautomatisierung

Die Bezeichnung Prozessautomatisierung steht im Folgenden für die Anlagenautomatisierung von chemischen, verfahrenstechnischen und energietechnischen Produktionsbetrieben mit Mitteln der Prozessleittechnik zur Überwachung und Führung des technologischen Prozesses. Abhängig vom zu automatisierenden Prozess (kontinuierlich, diskontinuierlich bzw. Mischformen) variieren auch die Feldinstrumentierungen und damit die Anforderungen an die Kommunikation in diesem Bereich. Nach dem Ebenenmodell wird die Prozessautomatisierung in eine Feld-, Prozessleit- und Betriebsleitebene unterteilt, wobei diese Ebenen systemtechnisch weitgehend selbstständig aufgebaut sind.

Anwendungs-ebene	Aufgaben/Funktionen	Typische Geräte	Charakteristische Anforderungen	Geeignete Kommunikations-systeme
Betriebs-leitebene	<ul style="list-style-type: none"> - Produktionsdatenverarbeitung - Produktionsplanung, Logistik und Anlagenoptimierung - mittel- und langfristige Datenarchivierung - Datenverknüpfung Statistik 	<ul style="list-style-type: none"> - Mainframes als Leitrechner - leistungsfähige Workstations 	<ul style="list-style-type: none"> - Kommunikation mit dem Leitrechner - hoher Datendurchsatz - Zeitverhalten min bis h - Fremdsystemankopplung 	<ul style="list-style-type: none"> - werksweite Netze - LAN-Anbindung - INTERNET - INTRANET
Prozess-leitebene	<ul style="list-style-type: none"> - Prozessdatenverarbeitung - Visualisierung und Bedienung - Koordination der systeminternen Kommunikation (verteiltes System) - Systemankopplung an Betriebsleitrechner, SPS, PC u. a. 	<ul style="list-style-type: none"> - industrietaugliche Prozessstationen - Anzeige- und Bedienstationen, z. B. Workstations - redundante Prozessstationen 	<ul style="list-style-type: none"> - Bewältigung mittlerer bis hoher Datenmengen - Zeitverhalten s bis min - Kommunikation mit überlagerten und unterlagerten Systemen 	<ul style="list-style-type: none"> - systeminterne Busse - Kommunikationsmöglichkeit mit Gateways oder Routern zu Lokalen Netzen LAN - nachrichtenorientierte Feldbusse
Feldebene (Sensor-Aktor)	<ul style="list-style-type: none"> - Erfassung von analogen und digitalen Signalen - Datenverarbeitung in den E/A-Komponenten - Verteilung der Daten über systeminterne Bussysteme - Steuerung und Regelung von Teilprozessen 	<ul style="list-style-type: none"> - analoge Sensoren und Aktoren - binäre Sensoren und Aktoren - Messumformer - Antriebe mit intelligenten Bausteinen - Analysegeräte - Waagensysteme - SPS, Regler - Bedien- und Anzeigegeräte 	<ul style="list-style-type: none"> - sichere Übertragungseigenschaften - prozessspezifische Übertragungszyklen - standardisierte Softwareschnittstellen und Geräteprofile - Eigensicherheit - Schutz vor Einflüssen aus dem Prozess - Teilnehmeranzahl entsprechend den Anlagen(teil)bereichen (eine 100) - herstellerunabhängiger Einsatz von Feldgeräten - Echtzeitfähigkeit - Kommunikation mit überlagerten Systemen - kurze bis mittlere Datensätze, variable Länge - Anschalten, Entfernen sowie Umparametrieren von Teilnehmern bei laufendem Busbetrieb 	<ul style="list-style-type: none"> - Sensor-Aktor-Bussysteme - geeignete Feldbusssysteme für explosionsgefährdete Bereiche - datenorientierte Feldbusssysteme - Gateways zu hersteller-spezifischen Bussystemen und Punkt-zu-Punkt-Ankopplungen

Abb. 2.9 Anforderungen für die Kommunikation in der Prozessautomatisierung

Im Feldbereich werden die Prozessdaten in Prozessstationen zusammengefasst und untereinander zu Knoten vernetzt, wobei die räumliche Ausdehnung der Anlage und die Datenmenge die Anzahl der Knoten bestimmt. Für die Prozessdatenerfassung kommen neben Sensoren und Aktoren auch intelligente Systeme wie Waagensteuerungen, Probenehmer- und Analyseeinrichtungen sowie Steuerungen von Verpackungs- und Transporteinrichtungen zum Einsatz (hierzu vgl. auch Abschn. 2.1.1 Fertigungsautomatisierung).

Die Explosionssicherheit, insbesondere Eigensicherheit (Ex i), einer Feldbusausführung ist für den Einsatz in der Prozessindustrie oft eine notwendige Bedingung (sog. K. o.-Kriterium), ebenso die Übertragung der Hilfsenergie direkt über die Datenleitung (Fernspeisung) bzw. über ein zusätzliches Adernpaar im Buskabel.

Die Busteilnehmer variieren stark in ihrer Anzahl, und das Netz kann sich von einigen 100 m bis zu 10 km je nach Anlagengröße ausdehnen. Die zeitlichen Anforderungen reichen von einigen ms bis zu Minuten, wobei diese Grenzen einerseits vom Prozess selbst und andererseits von Sicherheitsanforderungen (Abschaltungen) bestimmt werden.

In der Prozessautomatisierung muss aber auch die Möglichkeit des Einbindens von analogen Punkt-zu-Punkt-Verbindungen gegeben sein (z. B. 4–20 mA).

Durch die Aufteilung von Abb. 2.9 in drei Funktionsebenen mit den jeweiligen typischen Geräten und charakteristischen Anforderungen an die Kommunikation in der Prozessautomatisierung wird die Feldbusauswahl im Sinne einer Vorauswahl sinnvoll unterstützt.

Je nach gewähltem Konzept eines Prozessleitsystems übernimmt entweder ein Prozessbus die gesamte Kommunikation zwischen Prozessstationen und Leitstationen oder nur

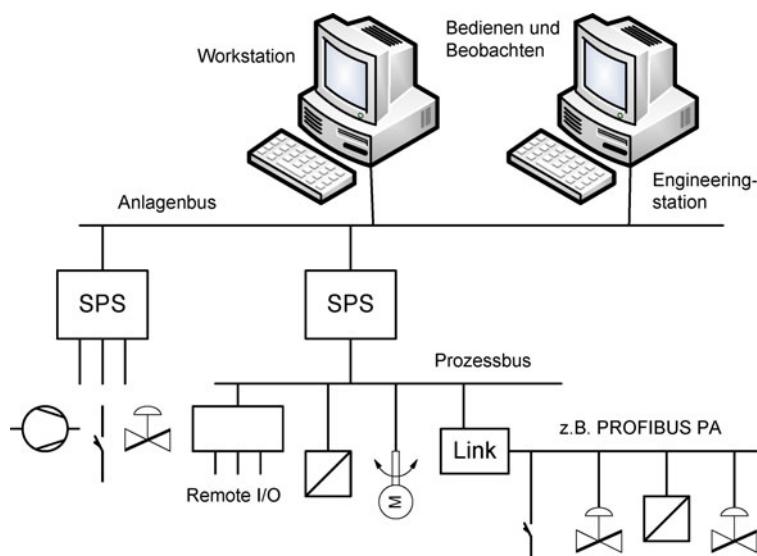


Abb. 2.10 Kommunikationsbeispiel aus der Prozessautomatisierung (Prozesseleitsystem einer verfahrenstechnischen Anlage)

die Kommunikation zwischen den Prozessstationen. Im zweiten Fall sorgt ein getrennt aufgebauter Systembus für die Kommunikation zwischen den Leitrechnern. Hierzu ist in Abb. 2.10 ein Beispiel dargestellt.

2.2 Kommunikationsebenen bei Stückgut- und Fließgutprozessen

Die Anforderungen an die Kommunikationssysteme unterscheiden sich auf jeder Ebene hinsichtlich der zu übertragenden Datenmenge, der erforderlichen Antwortzeit bei der Abfrage für die einzelnen Steuerungsaufgaben jeder Ebene (Datenresponsezeit) sowie hinsichtlich der Abfragehäufigkeit für die Einhaltung der Echtzeitbedingungen einer Steuerung bzw. des Mensch-Maschine-Interface (Abtastzeit). Hierbei steigen die Anforderungen an die zu übertragende Datenmenge von der prozessnahen Vor-Ort-Ebene in Richtung der Managementebene, während dagegen die Zeitforderungen abnehmen (Abb. 2.11).

Der ursprüngliche Ansatz der Industrie, wonach für jede Ebene spezielle sowie firmenspezifische Lösungen bereitgestellt wurden, die ihrerseits nicht kompatibel waren, hat sich auf dem Markt als nicht zukunftssicher erwiesen. Vielmehr hat sich aus langjährigen Erfahrungen von Anwendung und Normung eine typische Struktur gemäß Abb. 2.1 herausgebildet, die relativ selbständig und somit auch getrennt funktionsfähig sind. Die Busse dieser Ebenen sind überwiegend genormt und in Abb. 2.12 in einer Auswahl dargestellt.

2.2.1 Stückgutprozesse

Die Klasse der Stückgutprozesse ist dadurch gekennzeichnet, dass die Gegenstände (Objekte) der Prozesse abzählbare Stückgüter sind; d. h., ihr physikalischer Aggregatzustand ist *fest*, mindestens aber weisen sie eine *feste Oberfläche* auf (z. B. auch Flaschen, Fässer). Bei der Automatisierung derartiger Stückgutprozesse dominieren geometrische Größen

Kommunikationsebene	Nettodata pro Übertragung	Antwortzeit bei Abfrage	Abfragehäufigkeit (Abtastzeit)
Management	Mbyte (1 000 000 Byte)	Stunden/Tag (1 000...100 000 s)	Schicht/Tag (10 000...100 000 s)
Prozess-/Fertigungsleitung	kByte (1 000 Byte)	Sekunden (1...10 s)	Minuten/Stunden (100...1 000 s)
Prozessführung	10 Byte	100 ms (0,1 s)	Sekunden (1...10 s)
Objektnah: Steuerung/Regelung	4...16 bit (0,5...2 Byte)	10 ms (0,01 s)	100 ms (0,1 s)
Vor Ort: Sensoren/Aktoren	4...16 bit (0,5...2 Byte)	5...10 ms (0,005...0,01 s)	5...10 ms (0,005...0,01 s)

Abb. 2.11 Anforderungsbereiche für Kommunikationssysteme auf mehreren Ebenen

System	Norm
INTERBUS	IEC 61784-1, CPF 6
PROFIBUS	IEC 61784-1, CPF 3
WorldFIP	IEC 61784-1, CPF 5
Foundation Fieldbus	IEC 61784-1, CPF 1
ControlNet	IEC 61784-1, CPF 2/1
DeviceNet	IEC 62026-3
CANopen	EN 50325-4
AS-Interface	IEC 62026-3, EN 50295

Abb. 2.12 Auswahl an Bussystemen und deren CENELEC-Normen

wie Wege, Geschwindigkeiten, Winkel, Drehzahlen, Stückzahlen sowie Oberflächenbearbeitung, Transfer- und Verpackungsprozesse. Typisch ist hierfür, dass diese Prozesse sehr schnell ablaufen und aufgrund dieser hohen Dynamik entsprechend schnelle Automatisierungsmittel für den Echtzeitbetrieb erfordern, wie schnelle Sensoren und Aktoren, schnelle Speicherprogrammierbare Steuerungen (SPSen mit Zykluszeiten von ca. 10 ms). Daher muss die Bereitstellung der überwiegend binären Sensor- und Aktorsignale im Echtzeitbetrieb etwa alle 5 bis 10 ms erfolgen. Und wenn hierzu ein *Sensor-Aktor-Bus* als Zubringer für entsprechende SPSEN entworfen oder ausgewählt werden soll, so muss er über derartig kurze Reaktionszeiten verfügen. Dabei liegt die typische Übertragungsentfernung in einer Größenordnung von ca. 100 m.

Weiterhin ist typisch, dass automatisierte Stückgutprozesse wegen ihrer Vielzahl abzählbarer Gegenstände (Körper) zugleich auch eine Vielzahl binärer Signale aufweisen, sodass eine *Vielzahl binärer Sensoren und Aktoren* an einen Sensor-Aktor-Bus anzukoppeln ist (Größenordnung 100 bis 200 pro Busstrang). Die zusätzliche Buskopplung muss natürlich vom Aufwand her in einem bestimmten Verhältnis zu den Grundkosten derartiger Low-cost-Sensoren/Aktoren bleiben.

Soll also bei der Installation und Montage auf das aufwändige Abisolieren, Löten und Prüfen der Signalleitungen verzichtet werden, so führt dies auf die moderne *Schneidklemmtechnik* oder *Durchdringungsstechnik*, die ihrerseits aber keine Schirmung und keine Verdrillung des Kabels erlaubt. Daher muss die im Sensor-Aktor-Bereich unerlässliche Elektromagnetische Verträglichkeit (EMV) durch unkonventionelle elektronische und informationstechnische Mittel gesichert werden. In deren Folge verliert z. B. die Hamming-Distanz, als das bislang wichtigste Beurteilungskriterium für die Störfestigkeit von Feldbussystemen, ihre klassische Bedeutung vollständig.

2.2.2 Fließgutprozesse

Die Klasse der Fließgutprozesse ist dadurch gekennzeichnet, dass als Gegenstände (Objekte) der Prozesse entweder Gase, Dämpfe, Stäube oder Flüssigkeiten auftreten; d. h.,

ihr physikalischer Aggregatzustand ist *gasförmig* oder *flüssig*. Sie weisen also keine feste Oberfläche auf und müssen folglich in Rohrleitungen transportiert und in Behältern aufbewahrt werden, sodass sie insbesondere für den Anlagenfahrer nicht sichtbar sind. Der Zustand von Fließgütern muss daher bei der Automatisierung durch Messung von Temperatur, Druck, Differenzdruck, Durchfluss, Füllstand sowie Dichte, chemische Analysenwerte u. a. erfasst werden. Typisch ist hierfür weiterhin, dass diese Prozesse mittelschnell bis relativ langsam ablaufen und aufgrund dieser geringen Dynamik nur relativ langsame Automatisierungsmittel erfordern, wie z. B. Prozessleitsysteme. Daher genügt es, die überwiegend analogen Sensor- und Aktorsignale im Echtzeitbetrieb für Fließgutprozesse etwa alle 300 ms bereitzustellen (drei Abtastungen pro Sekunde). Eine genauere Zuordnung der Abtastzeiten zu den einzelnen Prozessgrößen gibt die Übersicht nach Abb. 2.13b. Ein hierfür auszuwählender *Sensor-Aktor-Bus* kann also mittelschnell bis relativ langsam abtasten, muss allerdings pro Abtastung eine größere Informationsmenge übertragen, die sich aus den digitalisierten Analogwerten ergibt (10 bis 14 bit). Dabei ist die Übertra-

a

Messgrößen	in Stückgut- prozessen	in Fließgut- prozessen	in Fließgutprozessen der Chemie
- Stückzahl (Zählgrößen)	25		
- Geometrische Größen (Länge, Weg, Winkel, Geschwindigkeit, Drehzahl, Beschleunigung)	25		
- Zeit	15	4	1
- Menge (Masse, Volumen)	5	5	5
- Temperatur	8	50	43
- Druck		10	20
- Durchfluss		15	17
- Füllstand		6	6
- Analysenwert (Stoffeigenschaft)		4	4
- Sonstige	22	6	4
Gesamt	100%	100%	100%

b

	Temperatur (T)	Druck (P)	Durchfluss (F)	Füllstand (L)	Analysen- wert (Q)	Sonstige
Mittlere Abtastzeit	100 s	1 s	1 s	10 s	100 s	0,5
Mittlere Abtastrate in 1/s	0,01	1	1	0,1	0,01	2
Häufigkeit	50%	10%	15%	6%	4%	15%

Abb. 2.13 Typische Unterschiede bei Messgrößen für Fließgut- und Stückgutprozesse. **a** Häufigkeitsverteilung für beide Prozessklassen (in % der Gesamtressortenanzahl), **b** erforderliche mittlere Abtastzeiten für Fließgutprozesse

gungsentfernung aber relativ groß und liegt im Bereich 100 bis 1000 m (auch 5 km), da Fließgutprozesse nicht selten in räumlich ausgedehnten (Freiluft-)Anlagen ablaufen.

Die Anzahl analoger Sensoren und Akten als Teilnehmer an einem Busstrang ist deutlich geringer und liegt in der Größenordnung 10 bis 30. Aufgrund des höheren Preisniveaus gegenüber binären Geräten liegen die tolerierbaren Buskoppelkosten mit maximal 25 % natürlich deutlich über dem Low-cost-Bereich.

Kostentreibend und im Sinne eines K. o.-Kriteriums wirkt hier die Forderung nach explosionsgeschützten Bussystemen, sofern der Anwendungsfall es verlangt; dies führt auf die weitgehend eigenständige Klasse der explosionsgeschützten Busse. Explosionssicherheit ist also ein zusätzliches Anwendungsfeld, das überwiegend nur in einer Unterkategorie der Fließgutprozesse mit bestimmten Gasen, Dämpfen und Stäuben auftritt, die zu Explosionen neigen.

Die vergleichende Gegenüberstellung von wesentlichen Messgrößen für Fließgutprozesse und Stückgutprozesse nach Abb. 2.13a macht zugleich deutlich, dass sich beide Prozessklassen nur wenig überschneiden, sich weitgehend komplementär darstellen und somit gut abgrenzbare Klassen bilden. Ihre gleichzeitige Relevanz für die Einteilung in Kommunikationsklassen haben die vorangestellten Erläuterungen bereits prinzipiell gezeigt.

Gleichzeitig muss aber darauf hingewiesen werden, dass reale Prozesse in der Praxis sehr häufig eine Kombination aus beiden Grundklassen aufweisen, z. B. *Fließgutprozesse mit nachgeschalteten Stückgutstufen* für Abfüllung, Verpackung und Transport des erzeugten Produkts, ggf. eingebettet in eine technologisch erforderliche Gebäudeautomatisierung. Demgemäß ergibt sich in komplexen Anlagen auch eine Mischung von Kommunikationssystemen, sowohl in ihrer charakteristischen Zuordnung zu den vorliegenden Prozessklassen (Einsatzklassen) als auch hinsichtlich der hierbei intern auftretenden Strukturen der Kommunikationsebenen.

2.3 Managementebene

Die mangelnde Integrationsfähigkeit zur Verfügung stehender Automatisierungskomponenten führte zu Beginn der 80er Jahre zur Entwicklung von offenen Übertragungsprotokollen, wobei eine nationale oder internationale Standardisierung angestrebt wurde. Ein wesentlicher Beitrag zur Standardisierung im Bereich der offenen Kommunikation wurde mit der Standardisierung des *Technical and Office Protocol (TOP)* und des *Manufacturing Automation Protocol (MAP)* geleistet. Das Technical and Office Protocol ist für den Bürobereich und das Manufacturing Automation Protocol für den Automatisierungsbereich vorgesehen. Die Spezifikation von MAP geht auf eine Initiative von General Motors im Jahre 1982 zurück und war eng gebunden an den Standardisierungsprozess der International Organization for Standardization (ISO). Wenig später startete der amerikanische Flugzeugkonzern Boeing die Arbeiten zur Spezifikation des TOP. Einen vorläufigen Abschluss der Arbeiten bildete die im Jahre 1987 bereitgestellte Version 3.0

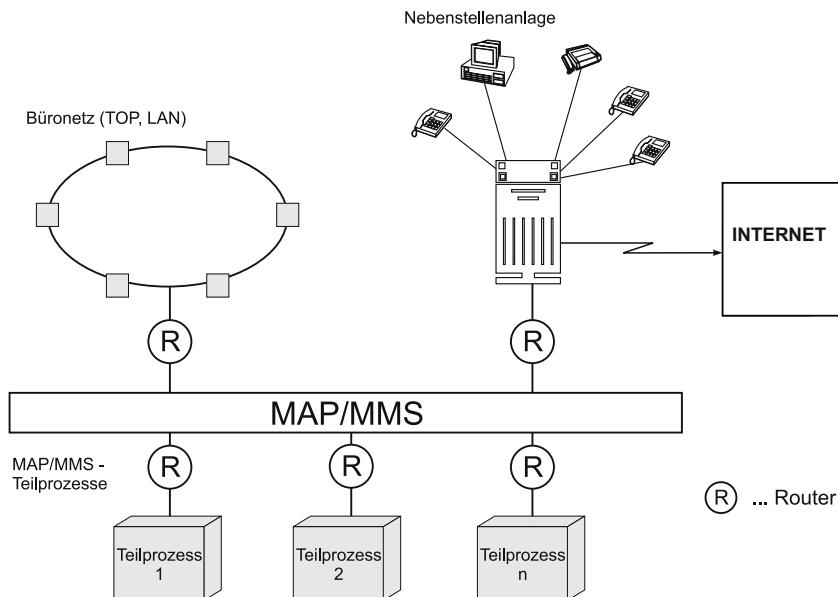


Abb. 2.14 MAP/MMS-Kommunikationsstrukturen

von MAP/TOP. Diese Version stellt erstmalig für die Entwickler von Automatisierungskomponenten eine einheitliche Schnittstelle bereit (Abb. 2.14).

Sie ermöglicht die Kommunikation zwischen Automatisierungsendgeräten verschiedener Herkunft, wie z. B. speicherprogrammierbare Steuerungen, Werkzeugmaschinensteuerungen, Robotersteuerungen oder Bedien- und Überwachungseinrichtungen. Die Spezifikation basiert auf dem ISO-OSI Referenzmodell und beruht auf einer Anzahl von OSI-Protokollen für die wichtigsten Anwendungsbereiche.

Mit der Definition der *Manufacturing Message Specification (MMS)* wird eine objektorientierte Dienstschnittstelle für die Kommunikation mit Automatisierungsgeräten genormt. Dieser internationale Standard (ISO 9506) definiert die Kommunikationsdienste zum Strukturieren, den Betrieb und die Modifikation verteilter Automatisierungssysteme. Dem MMS ist ein Client-Server-Modell zugrunde gelegt, in dem die von außen sichtbare Struktur und das Verhalten der jeweiligen Automatisierungskomponente definiert wird. Voraussetzung ist, dass diese Komponente als Server fungiert. Die zur Verfügung stehenden Dienste ermöglichen ein Erzeugen, Beobachten, Dokumentieren, Manipulieren, Verknüpfen und Löschen von MMS-Objekten in einem heterogenen Automatisierungsverbund. Durch die MMS werden damit standardisierte Eigenschaften von Automatisierungskomponenten festgelegt. Während bei MMS die Übertragung messageorientiert (nachrichtenorientiert) durchgeführt wird, erfolgt in der Feldebene eine datenorientierte Übertragung.

2.4 Prozessleitebene

Mit ersten Überlegungen zum Einsatz von Bussystemen für die Datenübertragung in der Automatisierungstechnik wurden vorwiegend Systemarchitekturen mit linienförmigem Feldbus konzipiert (Abb. 2.15a). An dieses System wurde eine zentrale Instanz (z. B. Leiteinrichtung) und die verschiedenen Mikroprozessrechner in prozessnahen Komponenten PNK angeschlossen. Die einzelnen Feldgeräte sind dann über die Rangierverteiler sternförmig mit dem jeweiligen Mikroprozessrechner verbunden, in dem die eigentliche Signalverarbeitung abläuft. Im Anschluss an diese Entwicklungsstufe erfolgte eine Kopplung des bitseriellen Feldbussystems mit einem *Local Area Network* (LAN) und damit die Möglichkeit des Datenaustausches mit modularen Wartenrechnern. Um eine übergeordnete Informationsverarbeitung auch über größere Entfernung zu gewährleisten, wurde die Kopplung mehrerer Datenübertragungssysteme über ein *Wide Area Network* (WAN) vorgenommen, vgl. Abb. 2.15b.

Mit der Weiterentwicklung der Mess- und Stelleinrichtungen entstanden auch direkt buskoppelbare Feldgeräte. Diese intelligenten Mess-, Stell- und Leiteinrichtungen können nun direkt an ein serielles Datenübertragungssystem angeschlossen werden (Abb. 2.15b). Die einzelnen Busstrukturen werden nun nicht mehr allein aus Linienstrukturen gebildet. Es kommen auch Ringstrukturen bzw. gemischte Varianten zur Anwendung, die ein *Field Area Network* (FAN) bilden. Bei Kompaktleitsystemen nach Abb. 2.15c werden insbesondere Standardkomponenten genutzt, z. B. SPSEN, PCs sowie Industrial-ETHERNET als LAN, CAN und Modnet/Modbus als FAN.

Die Kopplung des bitseriellen Prozessbusses mit einem LAN erfolgt über eine spezielle Einrichtung, das so genannte Gateway. Ein LAN ist also ein Verbund räumlich verteilter Rechner mit dem Ziel des Datenaustausches und der Zusammenarbeit. In Abb. 2.16 ist eine Variante eines LAN dargestellt, basierend auf Personalcomputern und Work-Stations.

Mehrere Personalcomputer bzw. Work-Stations sind in Linienstruktur mit der Server-Station verbunden. Die Server-Station stellt die zentrale Instanz des Datenübertragungssystems dar. Über das oben genannte Gateway erfolgt der mögliche Datenaustausch mit einem Prozess- oder Feldbussystem, um z. B. Ergebnisse aus dem Konstruktionsbereich (CAD) direkt in die Fertigungssteuerung einspeisen zu können. Die verschiedenen LAN können in vielerlei Hinsicht durch ihre Eigenschaften und Parameter spezifiziert werden, u. a. Übertragungsrate, Übertragungsstruktur, Zugriffsverfahren, Übertragungsprotokoll, bereitgestellte Dienste und die passive oder aktive Kopplung der Teilnehmer.

2.4.1 ETHERNET für den Industrie-einsatz

Die amerikanische Firma XEROX brachte Mitte der 70er Jahre ein lokales Netzwerk mit der Bezeichnung ETHERNET auf den Markt. Mit einem bis zu 1000 m langen Koaxialkabel konnten über 100 Teilnehmer (Rechner, Drucker u. Ä.) miteinander verbunden werden und gleichberechtigt kommunizieren. Der Zugriff einzelner Teilnehmer auf das gemeinsa-

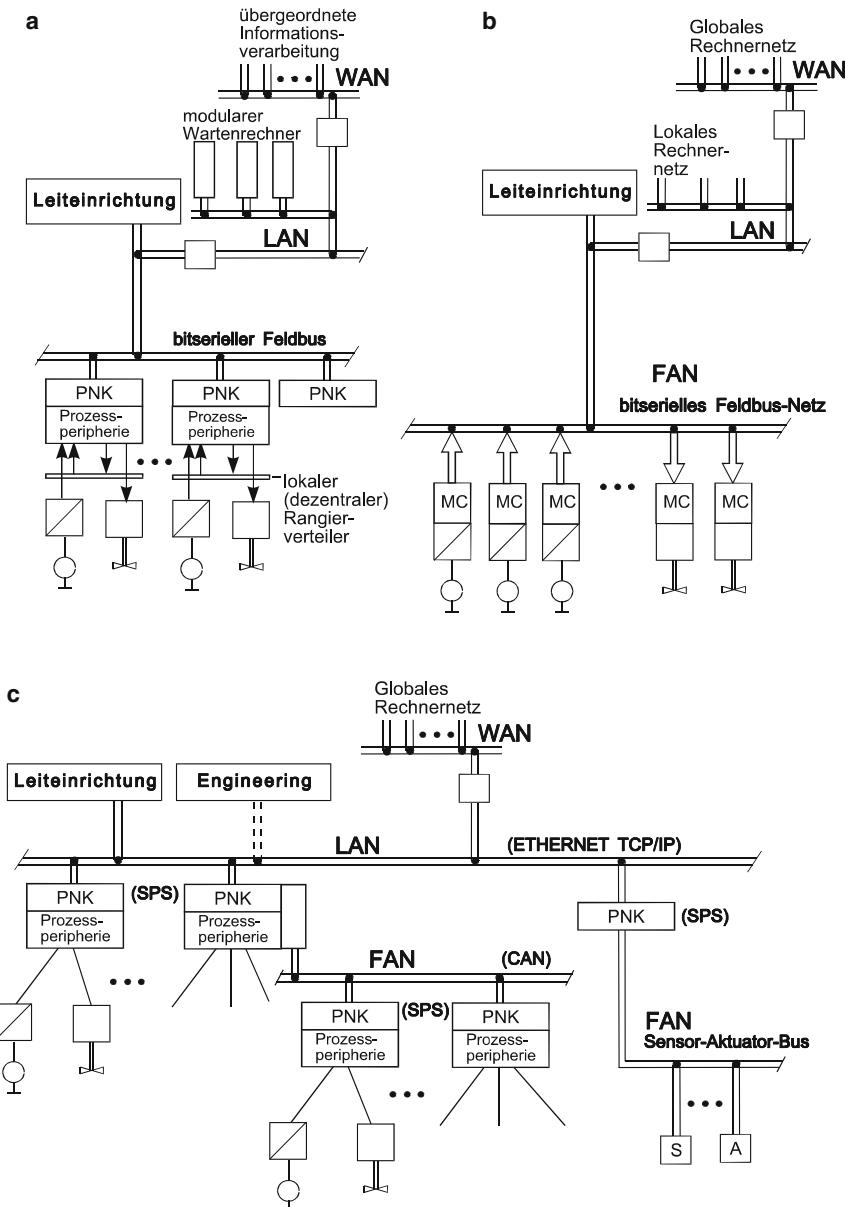
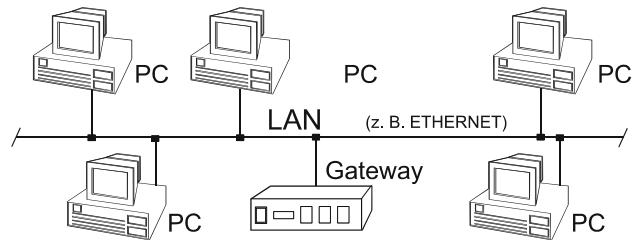


Abb. 2.15 Automatisierungssysteme in drei Generationen mit Anschluss an Kommunikationssysteme. **a** mit Mikroprozessrechnern, **b** mit intelligenten, busgekoppelten Mess-, Stell- und Leiteinrichtungen auf der Basis von Mikrocontrollern, **c** Kompakteitsysteme.

PNK – Prozessnahe Komponenten, WAN – Wide Area Network, MC – Mikrocontroller, LAN – Local Area Network, FAN – Field Area Network

Abb. 2.16 Struktur eines Local Area Network (LAN)



me Koaxialkabel (Buskabel) wird nach dem zufälligen Buszugriffsverfahren CSMA/CD gesteuert. Neben diesem Busprinzip mit *Liniенstruktur* sind auch *Sternstrukturen* mit verdrillten Leitungen (Twisted Pair) bzw. Lichtwellenleitern (LWL) möglich. Auch drahtlose Übertragung (Wireless LAN) sind im Rahmen der Norm IEEE 802.11 vorgesehen. Als zukunftsweisende LAN-Technologie ist das Industrial ETHERNET mit einer Übertragungsrate von 100 Mbit/s anzusehen.

Der Einsatz von Lokalen Netzwerken für die industrielle Kommunikation ist an wesentliche *Voraussetzungen* gebunden:

- hohe Zuverlässigkeit der Informationsübertragung unter Industriebedingungen,
- Erfüllung von Echtzeitforderungen.

Beide Voraussetzungen werden zunächst vom ETHERNET nicht erfüllt. Fehlerhaft übertragene Telegramme werden durchaus vom Empfänger erkannt und nicht weitergeleitet. Trotzdem bleibt ETHERNET *unzuverlässig*, weil der Sender keinerlei Rückinformation über den Empfang seines Telegramms erhält und im Störungsfall also auch keine Wiederholung dieser Übertragung veranlasst. Die Übertragungszeiten nehmen auf Grund des zufälligen Verfahrens für den Buszugriff (CSMA/CD) ohnehin mit der steigenden Busbelastung stark zu, weil hierdurch die Kollisionsgefahr wächst, sodass *kein Echtzeitverhalten* garantiert ist (Abb. 2.17).

Die oben geforderten Voraussetzungen konnten jedoch dadurch erfüllt werden, dass ETHERNET zusätzlich zur Norm IEEE 802.3 mit einer übergeordneten Kommunikationssoftware ausgerüstet wird, z. B. TCP/IP (Transmission Control Protocol/ Internet Protocol), vgl. hierzu Abschn. 1.2.2. Diese Ergänzungen sind für die industrielle Kommunikation besonders gut geeignet, weil sie die Datenübertragung durch Fehlererkennung und Fehlerkorrektur sichert (z. B. durch Wiederholung), die Flusssteuerung hinsichtlich Vollständigkeit und Korrektheit in der Reihenfolge der Telegramme übernimmt, eine Schnittstelle zur Anwendungssoftware bildet und Verwaltungsdienste ausführt sowie bei den Teilnehmern bestimmte Beschränkungen hinsichtlich Häufigkeit und Dauer ihres Buszugriffs erzwingt, um die Busbelastung gering zu halten und somit das Echtzeitverhalten zu verbessern (siehe Abb. 2.17).

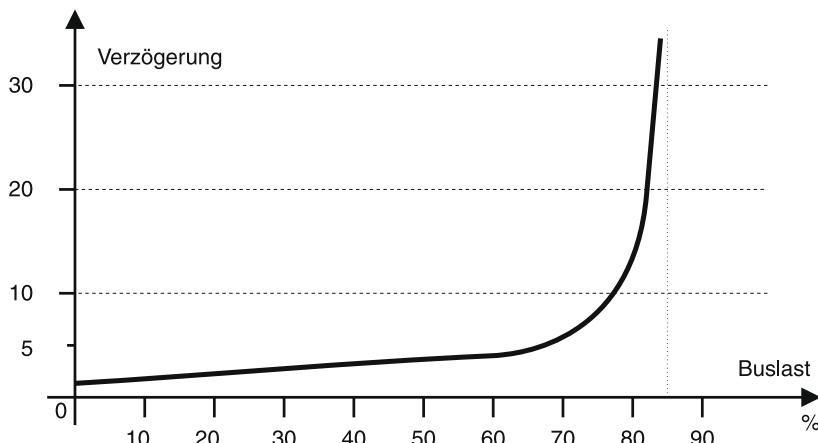


Abb. 2.17 Relative Zeitverzögerung als Folge der Busbelastung beim ETHERNET

2.4.2 Lösungsmöglichkeiten mit TCP/IP

Die Kommunikationssoftware TCP/IP ist dem ETHERNET übergeordnet und bildet zusammen mit diesem eine typische Schichtenstruktur. Ein Vergleich dieses Schichtenaufbaus mit den 7 Schichten des ISO/OSI-Referenzmodells nach Abschn. 1.2.1 zeigt eine gute Übereinstimmung des ETHERNET mit den Schichten 1 bis 4, wobei TCP/IP den Schichten 3 und 4 zuzuordnen ist. Dies ist bemerkenswert, wenn man bedenkt, dass TCP/IP bereits deutlich vor dem ISO/OSI-Modell geschaffen wurde, das seinerseits auf das Jahr 1983 zurückgeht.

Internet Protocol IP

Die Aufgabe des Internet-Protokolls besteht darin, die Telegramme von einem Teilnehmer zu einem oder mehreren anderen Teilnehmern zu übertragen. Dies erfolgt sowohl innerhalb eines Netzwerkes als auch zwischen verschiedenen Netzwerken (daher auch die Bezeichnung Internet). Diese verschiedenen Netzwerke sind über Router verbunden, die auch in der Lage sind, verschiedene Übertragungswege (Pfade) durch einen Netzwerkverbund auszuwählen, sodass Überlastungen oder Störungen einzelner Netze umgangen werden.

Im Einzelnen erfüllt das Internet Protokoll folgende *Funktionen*:

- Übermittlung von Telegrammen (Datagrammen) vom Sender zu einem oder mehreren Empfängern
- Adressenverwalter (Adress-Management)
- Telegrammaufteilung (Segmentierung)
- Pfadsuche (Routing)
- Netzwerk-Kontrollfunktionen (Fehlererkennung).

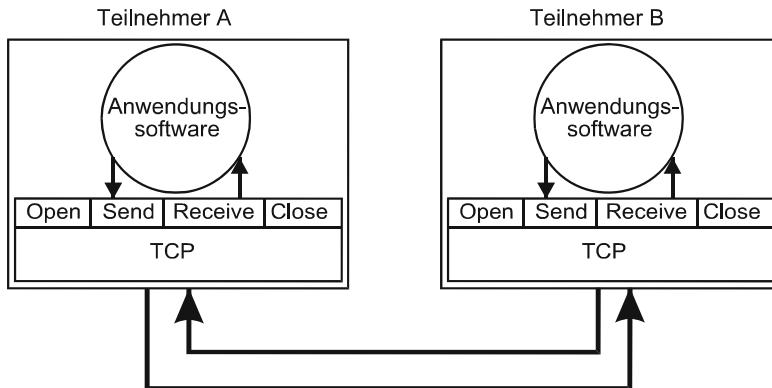


Abb. 2.18 Modell eines Übertragungssystems mit TCP

Als Folge dieses Routings kann es allerdings auftreten, dass sich einzelne Telegramme wegen unterschiedlich langer Wegstrecken überholen und somit in anderer Reihenfolge (Sequenz) beim Empfänger eintreffen.

Insgesamt wird aber das *unzuverlässige Übertragungsverhalten* des ETHERNET auch durch das Internet Protokoll noch nicht beseitigt. Das IP kann weder die Reihenfolge der Telegramme beim Empfänger noch deren sicheren Empfang garantieren (Telegrammverlust oder Verdopplung). Ein solches Kommunikationsverhalten wäre aber für die Vernetzung von industriellen Steuerungssystemen gänzlich ungeeignet, sodass eine weitere Softwareschicht erforderlich ist (TCP).

Transmission Control Protocol TCP

Die Protokollsoftware zur Übertragungssteuerung TCP übernimmt die Aufgabe, eine *zuverlässige Telegrammübertragung* zu sichern, d. h. fehlerfrei, sequenzgerecht und vollständig. Dazu organisiert TCP einen *Vollduplex-Betrieb* zwischen der Anwendungssoftware in verschiedenen Teilnehmern, wie dies in Abb. 2.18 ersichtlich ist.

Somit sind insbesondere auch *Rückinformationen* vom Empfänger zum Sender möglich, um den Telegramempfang zu bestätigen oder Fehler mitzuteilen (Quittungsbetrieb).

Im Einzelnen erfüllt die TCP-Software folgende *Funktionen*:

- zuverlässige Telegrammübertragung
- Vollduplex-Datenstrom zwischen den Teilnehmern
- Aufbau und Abbau von Verbindungen zwischen den kommunikationswilligen Teilnehmern (verbundungsorientiertes Protokoll)
- Überwachung bestehender Verbindungen (z. B. unerwarteter Abbruch, Stau im Netzwerk u. a.; Meldungen an Anwendungssoftware)
- Zwischenspeicherung und Aufbereitung der Datenblöcke (von der Anwendungssoftware an das TCP übermittelte Datenblöcke in beliebiger Größe und zu beliebigen

Zeitpunkten werden bis zur Übertragung gespeichert, ggf. in übertragbare *Segmente* zerlegt sowie im Empfänger wieder korrekt zusammen gesetzt)

- Vereinbarung dynamischer Ports (zwischen TCP und Anwendungssoftware vereinbarte Nummern zur Schnittstellenbezeichnung).

Als *Sicherungsmechanismen* für die Informationsübertragung werden durch das TCP realisiert:

- Erkennen von Übertragungsfehlern (32 Bit-Prüfsumme)
- Empfangsbestätigung über korrekt erhaltene TCP-Segmente (Quittung, „Acknowledgement“)
- Wiederholung bei Übertragungsfehlern oder Telegrammverlust („Repeat“)
- Zeitüberwachung zwischen Senden und Empfangsbestätigung („Time-Out“).

Diese Mechanismen erhöhen die Verfügbarkeit und Datensicherheit des ETHERNET, und somit steigen die Einsatzchancen für die industrielle Kommunikation ganz wesentlich. Offen bleibt jedoch nach wie vor die Erfüllung der Echtzeitforderungen.

Echtzeitfähigkeit durch Lasteinschränkungen

Ein industrielles Kommunikationssystem muss weiterhin gewährleisten, dass ein Datentelegramm von einem Sender innerhalb eines *bekannten und akzeptierbaren Zeitintervalls* mit sehr hoher Wahrscheinlichkeit bei einem Empfänger eintrifft, sodass dieser *rechtzeitig* reagieren kann. Die hierbei *zulässige* Nachrichtenverzögerung (Meldungsverzögerung, Message Delay) hängt einerseits von den Anwendungsbereichen ab, und innerhalb dieser wird sie durch die jeweilige Kommunikationsebene maßgeblich bestimmt.

Das Industrial-ETHERNET wird für die Übertragung i. allg. oberhalb und ggf. auch innerhalb der Feldbus-Systemebene benötigt. Aus Abb. 2.2 geht hervor, dass in diesen Leit- und Führungsebenen relativ große Datenmengen zu übertragen sind, aber die zulässigen Verzögerungen gleichfalls verhältnismäßig große Werte annehmen dürfen (0,1–10 s). Zur Realisierung solcher weichen Echtzeitforderungen bestehen auch bei dem zufälligen Buszugriffsverfahren CSMA/CD des ETHERNET gute Möglichkeiten, indem die Busbelastung extrem niedrig gehalten wird, z. B. kleiner 10 % (vgl. Abb. 2.17). Eine solche *Lasteinschränkung* wird erreicht, indem in jedem Teilnehmer die übertragbaren Nachrichten durch das Anwendungsprogramm oder die TCP/IP-Software begrenzt werden:

- maximale mittlere Nachrichtenrate (Telegramme pro Sekunde),
- maximale mittlere Dauer einer Nachricht,
- minimaler Zeitabstand zwischen den Nachrichten (Wartezeit des Teilnehmers).

Diese Lastbegrenzung bei allen Teilnehmern verbessert das Echtzeitverhalten von ETHERNET entscheidend und macht es besser industrietauglich. Das zufällige Verhalten des CSMA/CD-Verfahrens bleibt trotz Lastbegrenzung erhalten, aber die Verzögerungszeit wird voraussagbar.

Echtzeitfähigkeit durch Switching-Technologie

Einen anderen Weg zur Erlangung der Echtzeitfähigkeit beim ETHERNET beschreitet die so genannte Switching-Technologie, indem eine veränderte Netzstruktur mit zwei Ebenen eingeführt wird. Die unterlagerte Ebene enthält lokale Segmente, in denen die jeweils hierin befindlichen Teilnehmer ihren Datenaustausch untereinander lokal abwickeln. Dieser lokale (dezentrale) Datenverkehr bleibt also innerhalb des jeweiligen lokalen Segments und belastet daher das restliche Netz nicht.

Die überlagerte Ebene des Netzes besorgt folglich nur die Kommunikation zwischen den lokalen Segmenten, sodass insgesamt eine wesentliche Steigerung der Performance des gesamten Netzwerkes eintritt. Es werden also mehrere Telegramme gleichzeitig ausgetauscht, sodass die Echtzeitfähigkeit entscheidend verbessert wird.

Die weltweite Verbreitung von ETHERNET-TCP/IP wurde auch dadurch bewirkt, dass es zunächst im UNIX-Betriebssystem als Standard-Kommunikationssoftware für LAN-Vernetzung implementiert wurde und folglich mit jeder Workstation mitgeliefert wird. Inzwischen ist die TCP/IP-Software auf allen modernen Rechnerplattformen mit den Betriebssystemen UNIX, WINDOWS, Großrechnerbetriebssystemen sowie zunehmend auch in Echtzeitbetriebssystemen verfügbar. Auf dieser Grundlage ist auch die Integration von ETHERNET-TCP/IP für überlagerte Kommunikationsebenen in Automatisierungssystemen relativ einfach möglich und nimmt an Verbreitung kontinuierlich zu.

Weitere Lösungsmöglichkeiten zur Erzielung eines echtzeitfähigen Ethernet-Systems sind das „Ethernet-MAC“ und „Ethernet mit modifizierten Buszugriffsverfahren und Timing“.

Die Anwendung der Ethernet-Technologie ist den vergangenen Jahren stark gestiegen. Aktuell werden eine große Anzahl an Industrial Ethernet Lösungen angeboten.

2.4.3 ETHERNET-Anwendungen in der industriellen Kommunikation

PROFINET: Offener Standard für Industrial Ethernet von PROFIBUS, s. Abschn. [4.3.4](#).

EtherNet/IP: s. Abschn. [4.3.5](#).

EtherCat: s. Abschn. [4.3.8](#).

POWERLINK: s. Abschn. [4.3.6](#).

SERCOS III: Weiterentwicklung des SERCOS Interface-Standards, Abschn. [4.2.1](#).

CC-Link IE Field: basiert auf Norm IEEE 802.3,

Modbus-TCP: s. Abschn. [4.3.7](#).

Foundation Fieldbus: s. Abschn. [4.2.7](#).

2.5 Feldebene

2.5.1 Anforderungen an ein Bussystem der Feldebene

Resultierend aus der völlig veränderten Übertragungsart mit seriellen Kommunikationssystemen gegenüber den in der Feldebene herkömmlichen Sternstrukturen, lassen sich einige wichtige Anforderungen ableiten.

Topologie

Das Feldbussystem muss Automatisierungsgeräte über Entfernungen von einigen Metern bis zu einigen Kilometern verbinden. Weiterhin wird die Möglichkeit gefordert, das Kommunikationssystem flächendeckend zu verlegen, um die in der Prozessumgebung verteilten Automatisierungsgeräte miteinander zu koppeln. Um relativ autonom arbeitende Teilkomplexe zu gestalten, wird Flexibilität in der Handhabung der Teilnehmer gefordert.

Zeitverhalten

Eines der wesentlichen Kriterien für Automatisierungssysteme ist der Zeitbedarf des Gesamtsystems für den Informationsaustausch zwischen den betreffenden Funktionseinheiten. Auch die Ausführung von Bedienhandlungen und Meldungen bzw. Reaktionen auf jeweilige Alarmzustände müssen in einer bestimmbaren Zeit erfolgen. Damit stellt sich die Forderung nach Echtzeitfähigkeit des Übertragungssystems und die Projektierbarkeit der maximalen Reaktionszeit im System. Die geforderten Reaktionszeiten sind abhängig vom jeweiligen Projekt und liegen im Millisekunden- bis Sekundenbereich. Die wesentliche Rolle zur Absicherung der Echtzeitfähigkeit liegt beim verwendeten Zugriffsverfahren. Daraus resultierend finden determinierte Zugriffsverfahren (z. B. Polling wegen äquidistanter Abtastung) für die meisten Implementierungen in Feldbussystemen Anwendung. Weiterhin sollten kurze Datentelegramme mit hohem Funktionsgehalt (Nutzdaten) verwendet werden.

Zuverlässigkeit

Da alle Teilnehmer gemeinsam das serielle Bussystem zur Kommunikation nutzen, entsteht vorerst eine strukturelle Verschlechterung der Zuverlässigkeit gegenüber herkömmlichen Übertragungsstrukturen. Diese Zuverlässigkeitsnachteile ergeben sich durch die Verringerung der Übertragungskanäle und die zusätzlich notwendigen Steuereinrichtungen (zentral und dezentral). Dies ist ein genereller Nachteil, den alle Systeme mit mehrfach genutzten Übertragungskanälen zeigen. Mit entsprechenden Maßnahmen besteht aber die Möglichkeit, die Ausfallwahrscheinlichkeit des Systems derart zu beeinflussen, dass die Zuverlässigkeit des Gesamtsystems akzeptable Werte annimmt. Durch geeignete Maßnahmen sind evtl. auftretende Fehler zu lokalisieren, und der Datentransport ist weiterhin zu gewährleisten. Auch der Einsatz des jeweiligen Übertragungsmediums sollte im Hinblick auf die Störempfindlichkeit (EMV) wählbar sein. Mit der Verwendung von Codesicherungsverfahren sind die zu übertragenden Daten vor nicht erkennbaren Fehlern,

die durch Signalverfälschung auftreten können, weitgehend zu sichern (Restfehlerwahrscheinlichkeit, vgl. Abschn. 1.4.3.3).

Stromversorgung

Für die stabile Arbeitsweise einer technischen Anlage ist immer die Gewährleistung einer zuverlässigen Hilfsenergieversorgung notwendig. Je nach Anforderung der Automatisierungsanlage ist eine entsprechende Lösung zu konzipieren, die bei sehr hohen Anforderungen redundant bzw. sogar fehlertolerant/unterbrechungsfrei sein muss.

Einsatzbedingungen

Abhängig von der Umgebung, in die Feldbussysteme installiert werden sollen, ergeben sich entsprechende Einsatzbedingungen. Dazu gehören u. a.:

- großer Temperatur- und Feuchtebereich
- Einsatz in Freiluft, Außen- und Innenraum
- Industrieluft, Meeresluft, Staub
- mechanische Schwingungen
- Schutzgrade, Ex-Schutz, Schiffseinsatz.

Diese Einsatzbedingungen sind bei der Auswahl und Projektierung des Feldbussystems für den jeweiligen Anwendungsfall zu berücksichtigen.

Flexibilität

Die Feldbussysteme müssen einen gewissen Grad an Flexibilität gewährleisten. Zusätzliche Teilnehmer sollten in installierte Systeme komplikationslos einzubinden sein. Weiterhin sollten entsprechende Inbetriebnahmehilfen zur Verfügung stehen.

Wirtschaftlichkeit

Ein wichtiges Kriterium bei der Projektierung von Feldbussystemen stellen die entstehenden Kosten dar. Die Anwender sind bestrebt, nicht mehr als etwa 10–20 % der Kosten eines Automatisierungsgerätes zusätzlich für seine Busanschaltung auszugeben. Durch entsprechend abstufbare Lösungen ist dieser Forderung Rechnung zu tragen. Weiterhin sollte die zusätzlich notwendige Entwicklungs- und Serviceumgebung wenig kostenintensiv sein.

2.5.2 Industrielösungen für Busse der Feldebene

CAN: Offenes System (ISO 11898), siehe Abschn. 4.2.6.

FOUNDATION Fieldbus: Offenes System (ISO/OSI-Schicht 1 definiert nach IEC 1158-2, IEC-Feldbusnorm angestrebt), siehe Abschn. 4.2.7.

INTERBUS: Offenes System (DIN E 19 258; Euronorm EN 50 254 in Vorbereitung als Ergänzung zu EN 50 170), siehe Abschn. 4.2.3.

LON: Offenes System (IEC-Normentwurf 62 026), siehe Abschn. 4.2.5.

PROFIBUS: Offenes System (DIN 19 245, Teil 1 bis 4; Euronorm EN 50 170; EN 50 020), siehe Abschn. 4.2.2.

SERCOS: Offenes System (Normung nach IEC 1491), arbeitet mit LWL in Ringstruktur (Serial Realtime Communication System), v. a. für elektrische Antriebe, siehe Abschn. 4.2.1.

2.6 Sensor-Aktor-Ebene

2.6.1 Anforderungen im Sensor-Aktor-Bereich

An Bussysteme für den Einsatz in der Sensor/Aktor-Ebene, wo vorwiegend binäre neben analogen Signalen prozessnah übertragen werden, sind besondere Anforderungen gestellt. Dieser Bereich der Kommunikation in Verbindung mit Prozessleitsystemen, speicherprogrammierbaren Steuerungen (SPS), Robotersteuerungen, numerischen Steuerungen (CNC), Sensoren, Aktoren u. a. Feldgeräten stellt erhöhte bzw. besondere Forderungen an solche Eigenschaften wie:

- Echtzeitverhalten (Systemreaktionszeit 5–10 ms),
- flächendeckende Topologie,
- Übertragungsmedien (z. B. optional LWL wegen erhöhter EMV),
- Zuverlässigkeit und ggf. Fehlertoleranz (Betrieb in elektrisch gestörter Umgebung),
- Einsatz unter Feldbedingungen,
- Wirtschaftlichkeit (sehr geringe Anschlusskosten) usw.

Hier wird die Spezifik der prozessnahen Umgebung ersichtlich. So bewegen sich die geforderten Zykluszeiten teilweise unterhalb von 10 ms. Diese genannten technischen und wirtschaftlichen Forderungen sind von den zu mächtigen Feldbusssystemen der höheren Ebenen nicht zu erfüllen. Auch die geringen Anschaltkosten, die gerade in diesem Bereich durch die Anschaltung von Low-cost-Sensoren und Aktoren äußerst präsent sind (Näherungssensoren ca. 25 €/Stück), unterstreichen die Spezifik dieses Gebietes. Abb. 2.19 zeigt mögliche Busstrukturen für den Sensor-Aktor-Bereich.

Die Sensor-/Aktorbusse für die unterste Kommunikationsebene besitzen wegen der genannten funktionellen und örtlichen Einsatzgegebenheiten sehr spezifische Eigenschaften. Diese sind insbesondere:

- die *Vielzahl* der anzuschließenden Sensoren und Aktoren (> 100 Stück an einem Busstrang),
- der *Low-cost*-Bereich, insbesondere bei binären Sensoren umfassen ca. 80 % aller Sensoren/Aktoren in technologischen Anlagen und Aggregaten)

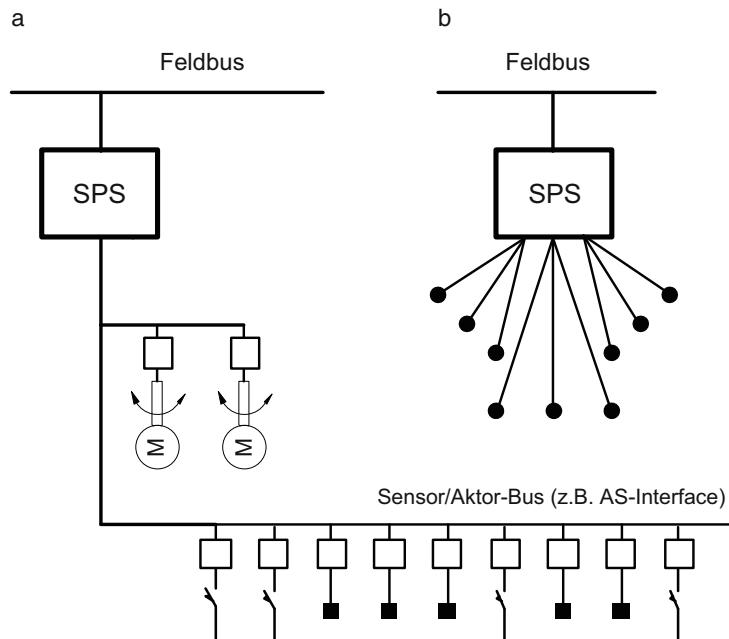


Abb. 2.19 Busstrukturen im Sensor-Aktor-Bereich. **a** SPS mit Sensor-Aktor-Bus, **b** SPS mit konventioneller Sternstruktur

- die *extreme räumliche* („atomare“) Verteilung von Sensoren/Aktoren im Unterschied zu „dezentralen Komponenten“ wie abgesetzte SPSEN,
- der *Vor-Ort-Einsatz* unter Berücksichtigung von EMV, Erdgeschleifen, Schutzart, Unzänglichkeit im laufenden Betrieb usw.,
- die *Installation* und *Montage* unter *Baustellenbedingungen* (einfachste Werkzeuge wie Schraubendreher, Voltmeter u. a.).
- Diagnosefähigkeit, möglichst bis zum Sensor/Aktor um Stillstandszeiten kurz zu halten

Zusätzlich kommen neuartige Bedingungen für die Installations- und Montagetechnologie hinzu, um insbesondere die Gesamtkosten bei gleichzeitiger Erhöhung der Flexibilität weiter zu senken. Hierzu einige spezifische Forderungen im Sensor-/Aktorbereich:

- *Wenig Verkabelung*, d. h. kein herkömmlicher, dicker Kabelbaum,
- *kein Abisolieren*, d. h. geringe Montagezeiten, kein Spezialwerkzeug,
- *kein Löten*, insbesondere von Steckern,
- *lösbare Verbindungen*, hohe Flexibilität, änderungsfreundlich bei Inbetriebnahme,
- *Verwechselungsfreiheit*,
- *2-Leiter-Technik*, d. h. Hilfsenergie und Datenübertragung auf einer gemeinsamen Signalleitung (Stromstärken bis etwa 16 A),

- *kein Durchtrennen* des Buskabels: Verhinderung eines Gesamtausfalls des Busstranges sowie einer Reduzierung des Übergangswiderstandes durch Korrosion; Abzweigdosenprinzip vergleichbar mit einer Elektroinstallation (passive Buskopplung),
- *flexible Installationsstruktur*, d. h. jede Linien- oder Baumstruktur ohne Einschränkungen bis zu einer maximalen Gesamtlänge soll möglich sein.

2.6.2 Industrielösungen für Sensor-Aktor-Bussysteme:

AS-Interface: Offenes System (EN 50295, IEC 62026-2), siehe Abschn. 4.1.1.

KNX: Offenes System (Euronorm EN 50 090), siehe Abschn. 4.1.2.

HART: Punkt-zu-Punkt-Verbindung: ein Feldgerät mit maximal zwei Anzeige- und Bedienkomponenten, kein echtes Bussystem!; alternative Linien-Busstruktur mit Stichleitungen (Multidrop): maximal 15 Feldgeräte sowie zwei Anzeige- und Bedienkomponenten, keine Analogsignalübertragung, Fernspeisung der Teilnehmer möglich; maximale Ausdehnung 3000 m; digitales Master-Slave-Verfahren mit gleichzeitiger analoger 4–20 mA-Signalübertragung; Datenübertragungsrate 1,2 kbit/s (Standard); 19,2 kbit/s (High-speed-Variante); Zykluszeit ca. 500 ms für die Abfrage eines Feldgerätes (Standard); verdrillte Zweidrahtleitung; gleichzeitige Übertragung von Signalen und Hilfsenergie; für den Einsatz im Ex-Bereich ist Eigensicherheit realisierbar; offenes System, steht jedem Anwender zur Verfügung.

M-Bus: Linienstruktur (mit passiver Buskopplung); Topologie auch als Stern, Baum oder Netz mit Segmenten; maximale Leitungslänge ca. 1000 m pro Segment (bei 9,6 kbit/s); maximaler Abstand zwischen Master und Slave 350 m; maximal 250 Teilnehmer pro Segment anschließbar; Repeater zur System- und Netzvergrößerung auf extreme Werte (über 100 Millionen Teilnehmer); Buszugriff mit kontrolliertem Master-Slave-Verfahren; Datenübertragungsraten 0,3 bis 9,6 kbit/s (auch gemischter Betrieb bis 38,4 kbit/s); Zweidrahtleitung (Adernpaar eines Standard-Telefonkabels) für Signal- und Energieübertragung (Fernspeisung); offenes System (Euronormentwurf EN 1434, Teil 3).

2.7 Fazit

Anhand der vorangegangenen Ausführungen wird deutlich, dass es den universellen Bus für alle Kommunikationsanforderungen auf allen Ebenen nicht gibt und auch in Zukunft nicht geben kann. In bestimmten Bereichen können zur Vereinheitlichung der auszuwählenden Bussysteme gewisse Abstriche in Teilaufgaben gemacht werden. Wichtig jedoch bleibt immer die Erfüllung der geforderten Eigenschaften und Strukturen, und dies ist nur durch eine zweckmäßige Kombination von spezialisierten Bussystemen auf mehreren Ebenen zu erreichen.

Abb. 2.20 fasst noch einmal abschließend mögliche Zuordnungen von Bussystemen zu den jeweiligen Anwendungsbereichen und Kommunikationsebenen zusammen.

Anwendungsbereich Ebene	Fertigung (Stückgut)	Prozess (Fließgut)
Unternehmensleitebene	INTERNET INTRANET Ethernet	INTERNET INTRANET Ethernet
Produktionsleitebene	INTERNET INTRANET Ethernet	INTERNET INTRANET Ethernet
Prozessebene	PROFINET PROFIBUS Industrial Ethernet	PROFINET PROFIBUS Industrial Ethernet
Steuerungsebene	INTERBUS PROFIBUS-DP CAN SERCOS	Foundation Fieldbus PROFIBUS-DP Modbus DeviceNet
Feldebene	INTERBUS AS-Interface	HART PROFIBUS PA

Abb. 2.20 Zuordnungen von Bussystemen (Auswahl) zu Kommunikationsebenen

Literatur

1. Etschberger, K. (Hrsg.): CAN Controller-Area-Network. 2. Aufl., München: Hanser 2000
2. Borst, W.: Der Feldbus in der Maschinen- und Anlagentechnik. München: Franzis 1992.
3. Färber, G.: Bussysteme, parallele und serielle Bussysteme, lokale Netze. München: Oldenbourg 1987
4. Gibas, P.: Feldbusnetz mit mehreren Ebenen zur objektnahen Informationsübertragung. Dissertation (Habilitation), TH Leipzig 1988
5. Heß, F.: Fehlertoleranz in objektnahen Kommunikationssystemen der Automatisierungstechnik. Dissertation, TH Leipzig 1990
6. Kriesel, W.; Gibas, P.; Helm, P.: Hierarchisches Feldbusnetz für intelligente Automatisierungs-mittel. 33. Wiss. Kolloquium TH Ilmenau (1988)1, S. 189–192
7. Forst, H.-J. (Hrsg.): Prozess- und Betriebsleittechnik. Funktionale und strukturelle Trends. Berlin: VDE-Verlag 1996.
8. Forst, H.-J. (Hrsg.): SPS-gestützte Leitsysteme. Berlin: VDE-Verlag 1997.
9. Kriesel, W.: Weiterentwicklung von Mikrorechner-Automatisierungssystemen unter dem Ein-fluß lokaler Netze (LAN). messen, steuern, regeln, Berlin 29 (1986) 1, S. 10–14.
10. Kriesel, W.: Weiterentwicklung von Mikrorechner-Automatisierungssystemen unter dem Ein-fluß intelligenter Funktionseinheiten. messen, steuern, regeln, Berlin 29 (1986) 2, S. 50–53.
11. Polke, M. (Hrsg.): Prozesseleittechnik. 2. Aufl. München: Oldenbourg 1994.
12. Seemann, G.: Anwendungsfälle für Kommunikation aus Fertigungstechnik, Verfahrenstechnik und Spezialgebieten. In: Vernetzung durch industrielle Kommunikation. Tagung Langen 5. Mai 1994, VDI Berichte 1123. Düsseldorf: VDI Verlag 1994.
13. Kriesel, W.; Gibas, P.; Heimböld, T.: Vernetzung von Sensor- und Aktuatorssystemen. mikro-elektronik, Berlin 5(1991)3, S. XXXIV–XXXVI

14. Rohbeck, V.: Zuverlässigkeitsuntersuchungen für neuartige Systemstrukturen im objektnahen Bereich von Automatisierungsanlagen. Dissertation, TH Leipzig 1988.
15. Schiff, A.: Auf dem Weg zur Intelligenz. Produktion (1991) Nr. 41
16. Töpfer, H.; Kriesel, W. (Hrsg.): Funktionseinheiten der Automatisierungstechnik – elektrisch, pneumatisch, hydraulisch. 5. Aufl. Berlin: Verlag Technik 1988
17. Schwarz, K.: Manufacturing Message Specification (MMS). Übersicht über die Methoden, Modelle, Objekte und Dienste. Automatisierungstechnische Praxis, München 33 (1991) 7, S. 369–378
18. Kriesel, W.R.; Madelung, O. W. (Hrsg.): AS-Interface – Das Aktuator-Sensor-Interface für die Automation. 2. Aufl. München: Hanser 1999
19. Furrer, F. J.: Ethernet-TCP/IP für die Industrieautomation. Grundlagen und Praxis. 2. Aufl. Heidelberg: Hüthig 2000
20. Kriesel, W.; Heimbold, T.; Telschow, D.: Bustechnologien für die Automation. Vernetzung, Auswahl und Anwendung von Kommunikationssystemen (mit CD-ROM). 2. Aufl. Heidelberg: Hüthig 2000
21. Klasen, F.; Oestreich, V.; Volz, M. (Hrsg.): Industrielle Kommunikation mit Feldbus und Ethernet. Berlin, Offenbach: VDE Verlag 2010



Feldbusnormung

3

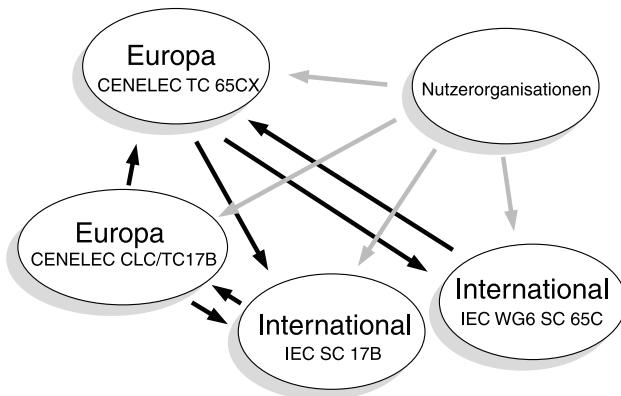
Die Feldbusnormung wird international bei der IEC (International Electrotechnical Commission) und in Europa bei der CENELEC (Comité Européen de Normalisation Electrotechnique) behandelt. In beiden Organisationen arbeiten Mitarbeiter der jeweiligen nationalen Normungsgremien mit, die wiederum von der Industrie und Instituten gestellt werden. In Deutschland ist die DKE (Deutsche Elektrotechnische Kommission im DIN und VDE) für die Feldbusnormung zuständig. Die zunehmende Internationalisierung bewirkt, dass nationale Normen und Normentwürfe im Feldbusbereich hauptsächlich als Vorschläge für die europäische und internationale Normenarbeit von Bedeutung sind. Die Nutzerorganisationen, die sich zur Unterstützung der verschiedenen Bussysteme gebildet haben erarbeiten Normentwürfe und stellen Spezialisten für die Normenarbeitskreise. Dadurch spielen diese Organisationen, obwohl sie kein direktes Stimmrecht in den Normungsgremien haben, eine sehr große Rolle in der Standardisierung. Abb. 3.1 gibt einen Überblick über das Zusammenspiel der Internationalen Gremien und Organisationen im Feldbusbereich.

3.1 Internationale Normungsarbeit

Für die internationale Feldbusnormung ist die Arbeitsgruppe IEC SC65C WG6 zuständig. Das ursprüngliche Ziel der Normungsarbeit war es, einen digitalen Ersatz für die 4–20 mA Schnittstelle in der Prozessindustrie zu finden. Im Laufe der Jahre hat sich der Scope jedoch auf alle Anwendungen sowohl in der Prozessindustrie als auch in der Fabrikautomation erweitert.

Die IEC 61158 umfasst die gesamte Breite der existierenden Feldbusse in einer Norm. Dabei sind die Teile der IEC 61158 nach ISO/OSI Schichten strukturiert. Bei der Vielzahl der existierenden Systeme führt das aber zu einer extrem unübersichtlichen Struktur, weil sich der Leser in der Regel nur für einen Bus interessiert, aber die gleiche Schicht aller genormten Busse mit in der gleichen Norm zu finden sind.

Abb. 3.1 Internationale Normung im Feldbusbereich



Daher wurde die Norm zusätzlich zur Strukturierung nach Schichten um eine Strukturierung nach Feldbusfamilien (CPF = Communication Profile Family) ergänzt (Abb. 3.2). Die Norm IEC 61158-5-3 beschreibt z. B. die Application Layer Service Definition der CPF 3 (Profibus).

Der höchste Grad der Harmonisierung ergibt sich beim Application Layer, während die Unterschiede im Data-Link Layer so groß sind, dass die verschiedenen Typen zueinander vollkommen inkompatibel sind. Die verschiedenen CPFs sind in der IEC 61784-1 bzw. in Falle der neueren Ethernet basierten System in der IEC 61784-2 definiert (Abb. 3.3). Die IEC 61784-3 befasst sich mit Sicherheitsbussystemen.

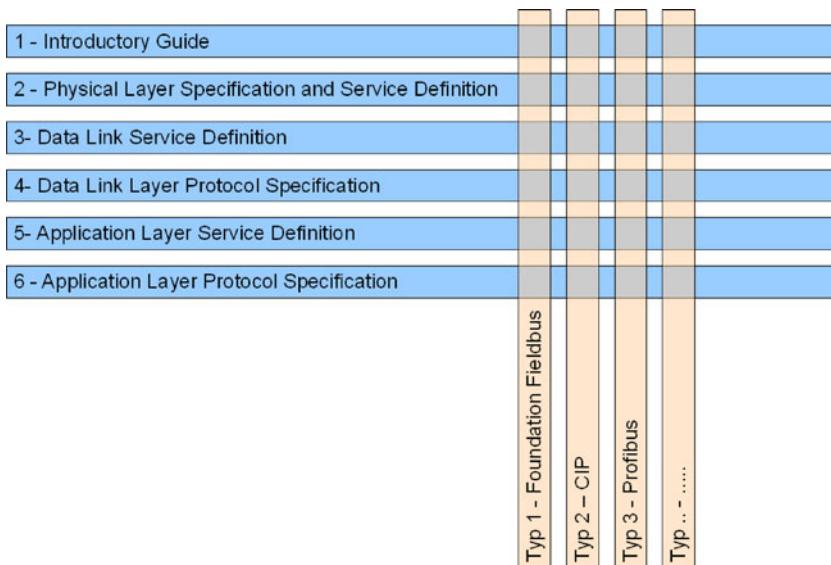


Abb. 3.2 Strukturierung der Normen nach ISO-Schichten bzw. nach Busfamilien

Abb. 3.3 Die Kommunikationsprofil-Familien (CPF) der IEC-Norm

Familie (CPF)	Unterfamilie	Norm	Name
CPF 1			Foundation Fieldbus
	1/1	IEC 61784-1	H1
	1/2	IEC 61784-1	HSE
	1/3	IEC 61784-1	H2
CPF 2			
	2/1	IEC 61784-1	ControlNet
	2/2	IEC 61784-1	EtherNet/IP
	2/3	IEC 61784-1	DeviceNet
CPF 3			PROFIBUS und PROFINET
	3/1	IEC 61784-1	PROFIBUS DP
	3/2	IEC 61784-1	PROFIBUS PA
	3/3	IEC 61784-1	PROFINET CBA
	3/4	IEC 61784-2	PROFINET IO Conformance Class A
	3/5	IEC 61784-2	PROFINET IO Conformance Class B
	3/6	IEC 61784-2	PROFINET IO Conformance Class C
CPF 4			P-Net
	4/1	IEC 61784-1	P-Net RS485
	4/2	IEC 61784-1	P-Net RS232
	4/3	IEC 61784-2	P-Net on IP
CPF 5	5/1, 5/2, 5/3	IEC 61784-1	World FIP
CPF 6	6/1, 6/2, 6/3	IEC 61784-1	INTERBUS
	6/4, 6/5, 6/6	IEC 61784-2	INTERBUS
CPF 8			CC-Link
	8/1	IEC 61784-1	CC-Link/V1
	8/2	IEC 61784-1	CC-Link/V2
	8/3	IEC 61784-1	CC-Link/LT
CPF 9	9/1	IEC 61784-1	HART
CPF 10	10/1	IEC 61784-2	Vnet/IP
CPF 11	11/1, 11/2	IEC 61784-2	TCnet
CPF 12	12/1, 12/2	IEC 61784-2	EtherCAT
CPF 13	13/1	IEC 61784-2	ETHERNET Powerlink
CPF 14	14/1, 14/2, 14/3	IEC 61784-2	
CPF 15			MODBUS RTPS
		IEC 61784-2	MODBUS TCP
		IEC 61784-2	MODBUS RTPS
CPF 16			SERCOS
	16/1	IEC 61784-1	SERCOS I
	16/2	IEC 61784-1	SERCOS II
	16/3	IEC 61784-2	SERCOS III
CPF 17	17/1	IEC 61784-2	PAPIEnet
CPF 18	18/1, 18/2	IEC 61784-2	SafetyNET p

Gremium	Standard	Teil	Titel	Status
SC65C WG6	IEC61158		Fieldbus Standard for Use in Industrial Control Systems	
		-1	Introductory Guide	Norm
		-2	Physical Layer Specification and Service Definition	Norm
		-3	Data Link Service Definition	Norm
		-4	Data Link Layer Protocol Specification	Norm
		-5	Application Layer Service Definition	Norm
		-6	Application Layer Protocol Specification	Norm
SC17B	IEC 62026		Low-voltage Switchgear and Controlgear - Controller-Device Interfaces (CDIs)	
		-1	General rules	Norm
		-2	Actuator sensor interface (AS-i)	Norm
		-3	DeviceNet	Norm
		-5	Smart distributed system (SDS)	zurück-gezogen
		-6	Seriplex (Serial multiplexed control Bus)	zurück-gezogen
		-7	CompoNet	Norm

Abb. 3.4 IEC-Feldbusstandards im Bereich der Sensoren

Zusätzlich zur Normung des IEC-Feldbusses ist im Bereich der Sensornormung ein Projekt zur Standardisierung von so genannten „Device Level Networks“ entstanden (IEC/SC 17B). Damit sind Busse gemeint, die direkt in einfache, kleine Sensoren integriert werden können. Hier wurden AS-I, Device-Net, SDS (Smart Distributed System, inzwischen zurückgezogen) und CompoNet genormt (Abb. 3.4). Ein wichtiger Grund zur Ansiedlung dieses Projektes in der Sensornormung war die Tatsache, dass die jeweiligen Nutzerorganisationen zur Beschleunigung der Normung die damals verfahrene Diskussion im Bereich IEC/SC 65C meiden wollten.

3.2 Europäische Normungsarbeit

Zwischen IEC und der zuständigen europäischen Normungsbehörde CENELEC existiert ein Vertrag aus dem Jahre 1991, der ein so genanntes parallel voting vorsieht. Ein internationaler Normentwurf wird dadurch automatisch auch als europäischer Entwurf zur Abstimmung gestellt. In der Regel werden solche Entwürfe angenommen. Eine europäische Norm bedeutet, dass nationale Normen zum gleichen Thema (z. B. FIP, PROFIBUS) zurückgezogen werden müssen. Die davon betroffenen Bussysteme werden dann bei öffentlichen Ausschreibungen nur noch mit Einschränkungen berücksichtigt.

Diese Situation und die anfänglich langsamem Fortschritte bei der internationalen Normung hatten zur Gründung des CENELEC-Gremiums TC 65CX geführt, das die in nationalen Normen bzw. Normentwürfen behandelten Bussysteme in die europäische Normung überführen sollte. Ziel dieser Normungsarbeit war es, mit europäischen Normen, die im Gegensatz zu nationalen Normen parallel zu einer europäischen IEC-Feldbusnorm bestehen können, die Investitionen in die bestehenden und bewährten europäischen Feldbusse zu sichern. Diese Arbeit hatte zur Verabschiedung der EN 50170 geführt, die unter dem Titel „General Purpose Field Communication System“ P-Net (Dänemark), FIP (Frankreich) und PROFIBUS (Deutschland) normte. In einer weiteren Norm (EN 50254) sollten unter dem Titel „High Efficiency Communication Subsystem for Small Data Packages“ INTERBUS-S, Distributed World FIP und PROFIBUS-DP genormt werden. Außerdem wurde die EN 50295 für AS-Interface geschaffen.

Ein Reihe von CAN Protokollen wurde in der EN 50325 standardisiert, wovon vor allem CANopen als EN 50325-4 noch aktiv ist.

Die Normenfamilien EN 50170 und EN 50254 sind mittlerweile zurückgezogen und in der IEC 61158 aufgegangen. Lediglich die AS-i Norm EN 50295 existiert noch parallel als europäischer Standard.

Damit ergibt sich eine Besonderheit für AS-Interface: Da die EN 50295 eine harmonisierte europäische Norm ist, kann sie zur Konformitätsbewertung nach dem EMV-Gesetz herangezogen werden. Für die anderen Feldbussen müssen für die Konformitätsbewertung andere Normen herangezogen werden.

Es soll an dieser Stelle auch erwähnt werden, dass die Einbeziehung eines Bussystems in eine Norm nicht zwangsläufig den Markterfolg sichert.

Die Arbeit an nahezu allen Feldbussen geht kontinuierlich weiter und wird zum größten Teil in den jeweiligen Nutzerorganisationen geleistet. Daher ist die Spezifikation der Nutzerorganisationen im Normalfall aktueller und fehlerärmer als die offizielle Norm, die nur sehr viel seltener geändert und ergänzt werden kann.

Für die praktische Arbeit, insbesondere bei der Entwicklung von Feldbus-Geräten, sind daher die Spezifikation und die Informationen der Nutzerorganisationen meist wertvoller als die Norm. Nichts desto trotz dokumentiert das Vorliegen einer internationalen Norm einen gewissen Reifegrad und Bedeutung des jeweiligen Bussystems.

Literatur

1. Wieser, M: Kein Land in Sicht, industrie-elektrik+elektronik 37 11/1992
2. NN: Dem Internationalen Feldbus gewidmet, Markt und Technik 13/93
3. Normenreihe IEC 61158, <http://www.iec.ch>
4. Normenreihe IEC 61784, <http://www.iec.ch>
5. DIN-EN50295: Niederspannungsschaltgeräte – Steuerungs- und Geräte-Interface-Systeme – Aktuator Sensor Interface (AS-i) (1999)
6. Fieldbus Foundation: Fieldbus News Nr. 2 (1995)
7. Open Devcie Net Voendor Association, <http://www.odva.org>
8. PROFIBUS International, <http://www.profibus.com>
9. Fieldbus Foundation, <http://www.Fieldbus.org>
10. CENELEC: <http://www.cenelec.be/>
11. IEC: <http://www.iec.ch>
12. DKE: Informationen zur Internationalen Feldbusnormenreihe IEC 61158, <http://www.dke.de/aktuell/infos/gremien/feldbus.sql> (2000)



Beispiele ausgeführter Bussysteme

4

4.1 Sensor/Aktor-Busse

4.1.1 AS-Interface – Aktor/Sensor-Interface

Das AS-Interface® (häufig als AS-i abgekürzt) ist nicht als eigenes Feldbussystem zu verstehen und in keiner Weise eine Konkurrenz zu bereits bestehenden Feldbussystemen wie z. B. PROFIBUS, Ethernet/IP, CAN und anderen. Vielmehr wurde das AS-Interface entwickelt, um auf der Sensor/Aktor-Ebene vor allem im nicht-explosionsgefährdeten Bereich Support für die Feldbussysteme zu liefern. Es soll einfach und kostengünstig die binären Signale einer Anlage an existierende Feldbusse oder direkt an die Steuerung anbinden.

Das AS-Interface nutzt die Vorteile der Feldbustechologie, ein zweiadriges Kabel zur Übertragung der Daten. Es benötigt aber zum Unterschied dazu keinen Schirm und keinen Endwiderstand und ist in der Lage, mit den gleichen zwei Adern des Kabels die notwendige Energie zu übertragen.

Seit Dezember 2004 steht die neuste Spezifikation (Version 3.0) mit erweiterten Funktionalitäten zur Verfügung. Bei allen Änderungen wurde größter Wert auf eine 100 %ige Kompatibilität zum bestehenden System gelegt. Auf die Neuerungen wird in den einzelnen Abschnitten dieses Kapitels eingegangen.

4.1.1.1 Konzept des intelligenten Verkabelungs-Systems

Durch den hohen Kostendruck in der Automatisierungstechnik wurde auf der Feldebene die Parallelverdrahtung durch einen Feldbus bzw. eine Zweileiter-Verkabelungstechnik ersetzt.

Auf der untersten Hierarchieebene, den Sensoren und Aktoren, wurde das mögliche Einsparungspotential bisher wenig genutzt, da die komplizierteren Feldbussysteme auf Grund ihrer Komplexität für die binäre E/A Ebene weniger geeignet sind.

An dieser Stelle setzt das AS-Interface an. Letztendlich ist das AS-Interface eine „intelligente Verkabelung“. Der Kabelbaum wird ersetzt durch ein zweiadriges Kabel, an das

alle Teilnehmer angeschlossen werden. Die „Intelligenz“ besteht aus den daraus resultierenden Möglichkeiten der Überwachung, der Diagnose und der Selbstdressierung, die in den folgenden Abschnitten beschrieben werden.

Das AS-Interface-System besteht aus einem Master, einem speziellen Netzteil, den Slaves und dem verbindenden Kabel.

4.1.1.2 Der Master

Der Master stellt die zentrale Einheit dar, die das System mit der darüber liegenden Ebene, der Steuerung oder einem Feldbus verbindet. Er arbeitet im Prinzip als intelligente Eingangs/Ausgangskarte und wird als solche von der Steuerung verwaltet. Der Master steuert den Datenaustausch mit den Slaves, sendet die Parameterdaten, wertet die Quittungstelegramme der Slaves aus und überwacht die Busfunktion.

Damit Masterimplementierungen sowohl für einfache speicherprogrammierbare Steuerungen als auch für Industrie-PCs oder Gateways zu anderen Bussystemen (etwa PROFIBUS, Ethernet etc.) möglich sind, wurde bei der Spezifikation der Funktionalität auf größtmögliche Einfachheit geachtet.

In der Initialisierungsphase nach dem Einschalten der Betriebsspannung erstellt der Master selbstständig die Konfiguration des angeschlossenen AS-Interface-Strangs und vergleicht diese mit einer abgespeicherten Sollkonfiguration; außerdem werden die Slaves bei Bedarf mit Parameterdaten versorgt. Im sich anschließenden Normalbetrieb werden mit allen angeschlossenen Slaves zyklisch Daten ausgetauscht. Fehlerhafte Telegramme werden identifiziert und nach einem bestimmten Algorithmus wiederholt. Zusätzlich werden die wichtigsten Busfunktionen überprüft: Anwesenheitskontrolle der Slaves, Überwachung der Konfiguration und der Stromversorgung. Der Ausfall oder das Entfernen eines Slaves wird daher sofort erkannt.

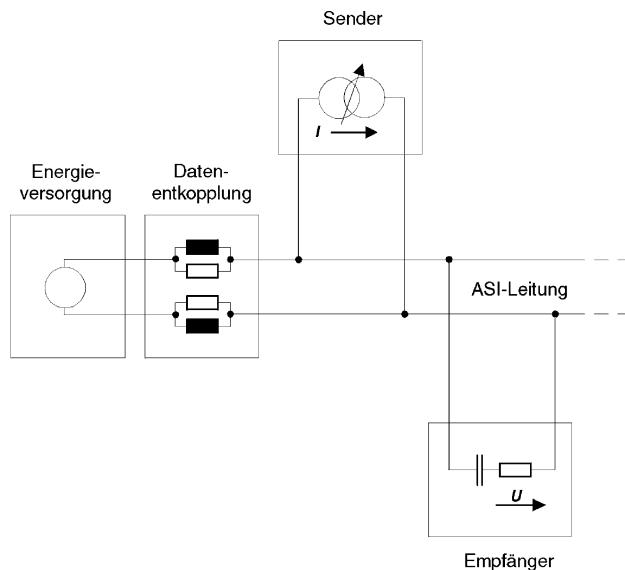
Sogar der Austausch einzelner defekter Slaves, ohne dass dafür spezielle Systemkenntnisse oder Programmiergeräte erforderlich sind, wird mit einer besonderen Masterfunktion ermöglicht. Wird etwa der Slave mit der Adresse 24 beschädigt und reagiert nicht mehr auf die zyklischen Datenaufrufe des Masters, so wird dieser nach einigen vergeblichen Aufrufwiederholungen den Slave als defekt identifizieren und dies melden. Ein gleichartiges Ersatzgerät aus dem Lager kann nun an die Stelle des Slaves Nr. 24 gesetzt werden; dieses hat jedoch zunächst die herstellerseitig eingestellte Adresse 0. Der Master prüft nun selbstständig durch Lesen und Vergleichen eines Identifikationscodes, ob es sich bei dem neuen um ein gleichartiges Gerät handelt und vergibt – bei positivem Ergebnis – automatisch wieder die Adresse 24 an den Slave. Diese wird dort permanent gespeichert, sodass der Originalzustand der Anlage wiederhergestellt ist.

4.1.1.3 Netzteil

Beim AS-Interface erfolgt die Datenübertragung und die Energieversorgung über das gleiche zweiadrige Kabel. Dadurch wird der Verkabelungsaufwand zusätzlich reduziert.

Ein spezielles Netzteil versorgt das AS-Interface. In diesem Netzteil wird auf eine Gleichspannung von 29,5...31,6V zur Energieversorgung eine Wechselspannung mittels

Abb. 4.1 Energieversorgung und Datenaustausch auf derselben Leitung



Datenentkopplung für die Kommunikation aufmoduliert (Abb. 4.1). Netzteile stehen in verschiedenen Bauformen zwischen 2,4 und 8 A zur Verfügung.

Neben dieser klassischen Energieversorgung mittels eines AS-Interface-Netzteiles kann diese auch über einen so genannten Power Extender vorgenommen werden. Der Power Extender ermöglicht eine kostengünstige Energieversorgung in der Schutzart IP67. Dazu wurden die Entkopplungsspulen des Netzteiles (vgl. Abb. 4.1) ausgegliedert und separat in ein IP67-Gehäuse installiert. Die Gleichspannung kann dann von einem normalen Gleichspannungsnetzteil erfolgen. Es lassen sich mehrere Power Extender an ein Netzteil anschließen.

Daneben gibt es auch Master, bei denen die Datenentkopplung bereits eingebaut ist, sodass ein spezielles Netzteil unnötig wird. Es wird lediglich eine entsprechende Gleichspannungsquelle benötigt mit Stabilität und geringer Brummspannung. Auch hier ist der Betrieb mehrerer Master an einer Energieversorgung möglich, sodass oft ein preiswertes (Kosten/Watt) Netzteil für mehrere Master verwendet werden kann.

Um AS-i auch für kleinere, preissensible Applikationen zugänglich zu machen, wurde AS-i Power 24 entwickelt.

Mit dieser Technik ist es möglich, bei kurzen AS-i Kreisen (bis 50 m) den AS-i Kreis über Power-Extender oder Master mit eingebauter Datenentkopplung aus 24 V Standard-Netzteilen zu versorgen. Dies kann die Systemkosten bei kleinen AS-i Kreisen signifikant senken. Die dafür geeigneten AS-i Komponenten sind sowohl für Standard-AS-i als auch für Power24 geeignet. Ob eine Komponente AS-i 24 geeignet ist, gibt der Hersteller an.

4.1.1.4 Elektromechanik

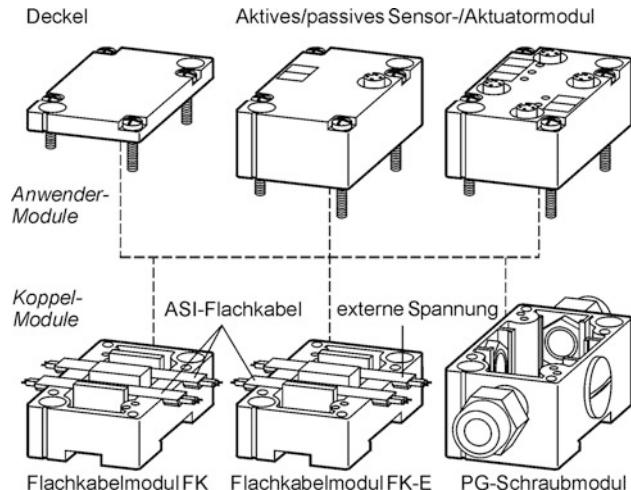
Das speziell entwickelte Flachkabel ermöglicht einen sekundenschnellen und damit kostengünstigen Anschluss eines AS-Interface-Slave an das Netzwerk. Durch die Geometrie des Kabels ist eine Verpolung ausgeschlossen. Bei dem Zusammenschrauben von Ober- und Unterteil der Module durchdringen Schwerter das Kabel und stellen die elektrische Verbindung her (Durchdringungstechnik). In Abb. 4.2 ist ein Beispiel eines Moduls, bestehend aus Ober- und Unterteil, gezeigt. Mittlerweile werden neben dieser Standardbauweise verschiedene andere Gehäusevarianten angeboten. Die gleiche Technik wird bei den intelligenten Sensoren und Aktoren angewandt. Die Durchdringungstechnik erlaubt das Anklemmen von Slaves im nicht-explosionsgefährdetem Bereich im laufenden Betrieb ohne dass dafür das Kabel getrennt, abisoliert und angeschlossen werden muss; sogar in der Schutzart IP67.

Als Alternative werden Unterteile zum Anschluss von Rundkabel angeboten. Das AS-Interface-System wurde so spezifiziert, dass preisgünstige handelsübliche Rundkabel eingesetzt werden können, die der Norm DIN VDE 0281 mit der Bezeichnung HOSVV-F 2X1,5 entsprechen.

Die gleichen Voraussetzungen müssen die Kabel erfüllen, wenn AS-Interface im explosionsgefährdeten Bereich eingesetzt wird. Zusätzlich muss das Kabel einfach mechanisch geschützt werden.

Wie in Abb. 4.2 gezeigt, werden bei vielen Modulen zwei unterschiedliche Flachkabel eingelegt. Das zweite Flachkabel dient zur separaten Versorgung der Module. Diese Variante wurde notwendig, da viele Module die Stromversorgung mit bis zu 2 A belasten. Ein weiterer Grund resultierte aus der Forderung der Industrie, die Ausgänge über das Not-Aus-System abschalten zu können.

Abb. 4.2 Modulsystem für Flach- und Rundkabel



4.1.1.5 Netzwerktopologie

Die Topologie eines AS-Interface-Netzwerkes ist beliebig wählbar, was die Projektierung sehr vereinfacht. Im Gegensatz zu anderen Bussystemen kann sie ganz an die örtlichen Anforderungen angepasst werden. Sie darf linienförmig sein, Stichleitungen enthalten oder sich wie ein Baum verzweigen, ganz so, wie eine normale Elektroinstallation auch aussehen würde. Die Busteilnehmer können gleichmäßig entlang des Buskabels verteilt angeordnet werden oder auch in Gruppen, etwa an den Enden der verschiedenen Äste des Baumes. Die normalerweise nötigen Leitungsabschlusswiderstände sind hier nicht erforderlich.

Der einzige Punkt, der beachtet werden muss, ist die Beschränkung auf eine Gesamtlänge von 100 m. Werden größere Längen benötigt, gibt es verschiedene Möglichkeiten der Erweiterung:

- Repeater, die ein neues AS-i-Segment eröffnen. Jedes Segment hat ein eigenes AS-i-Netzteil. Maximal 2 Repeater dürfen zwischen jedem Slave und dem Master liegen.
- Extender, die ebenfalls ein neues Segment eröffnen. Hier darf aber kein Slave vor dem Extender sitzen, da der Extender eine eigene, nur schwache Datenentkopplung besitzt. Dafür ist in diesem Bereich kein eigenes AS-i-Netzteil nötig. Auch mit Extendern dürfen maximal 2 Repeater oder Extender zwischen jedem Slave und dem Master liegen.
- Durch Abschlussimpedanzen lässt sich die mögliche Gesamtlänge auf bis zu 300 m erhöhen. Es gibt passive Abschlussimpedanzen und aktive, mit Qualitätsmonitor und adaptiver Einstellung (AS-i Tuner), die die optimale Reichweite ermöglichen.

4.1.1.6 Slaves

An ein AS-Interface-Netzwerk können zwei Arten von Slaves angeschlossen werden. Zum einen die intelligenten Sensoren und Aktoren mit integriertem AS-Interface-Chip, zum anderen Eingangs-Ausgangs-Module (genannt EA-Module), an die herkömmlichen Sensoren und Aktoren angeschlossen werden. Letztere werden in der Praxis häufiger eingesetzt als integrierte Sensoren.

Den intelligenten Sensoren und Aktoren stehen alle 4 Nutzdatenbits, die das AS-Interface überträgt, zur Verfügung. Damit erlauben sie eine zusätzliche detaillierte Diagnosemeldung wie z. B. eine Fehlermeldung bei einer gebrochenen Spule eines induktiven Sensors. Der Einsatz dieser Sensoren und Aktoren erlaubt es, von der präventiven Instandhaltung zur „Instandhaltung auf Bedarf“ (Maintenance on demand) überzugehen und somit Kosten zu sparen. Die mögliche Verdoppelung der Anzahl der Slaves pro Master reduziert die Overhead Kosten des Systems (Master und Netzteil) pro Slave, sodass der Einsatz von intelligenten Sensoren und Aktoren auch unter dem Kostenaspekt interessant wird.

Beim Einsatz von Modulen können die bisher benutzten Sensoren und Aktoren weiterverwendet werden. Sie werden mit möglichst kurzen Leitungen an die Module angeschlossen. Damit ist das System vom Modul bis zum Master überwacht. Ein Fehler zwischen

Modul und Sensor/Aktor kann allerdings im Gegensatz zu den intelligenten Sensoren/Aktoren in der Regel nicht erkannt werden.

An einen AS-i Master können bis zu 31 Single-Slaves oder bis zu 62 A/B-Slaves angeschlossen werden. Zwei A/B-Slaves teilen sich dabei eine Standard-Adresse, einer auf der „A-Adresse“ (z. B. 1A), einer auf der B-Adresse (z. B. 1B). In einen solchen A/B-Pärchen werden die Slaves abwechselnd, einer in jedem Zyklus, angesprochen. Die Zykluszeit liegt also für gepaarte A/B-Slaves doppelt so hoch wie für Single-Slaves oder einzelne A/B Slaves. A/B-Slaves wurden mit der Spezifikation 2.11 eingeführt.

4.1.1.7 Bitübertragung

Als Übertragungsverfahren wird die Alternierende Pulsmodulation (APM) eingesetzt (Abb. 4.3). Es handelt sich um eine Kombination aus Manchester-II-Codierung und dem Alternierenden-Flanken-Puls-Verfahren (siehe Abschn. 1.6.4 und 1.6.5). Das APM-Verfahren ist relativ schmalbandig (Hf-Störungen!) und kann einfach vom Sender erzeugt werden. Es ist gleichstromfrei und ermöglicht somit eine einfache Modulation auf die Energieversorgung.

Die Sendebitfolge wird zunächst in eine Bitfolge umcodiert, die bei jeder Änderung des Sendesignals eine Phasenumtastung vornimmt. Daraus wird dann ein Sendestrom erzeugt, der in Verbindung mit einer im System nur einmal vorhandenen Induktivität (Datenentkopplung) den gewünschten Signalspannungspiegel auf der Busleitung erzeugt. Jedes Ansteigen des Sendestromes führt also zu einem negativen, jedes Abfallen zu einem positiven Spannungspuls. Auf diese Weise ist es sehr einfach möglich, Signale zu erzeu-

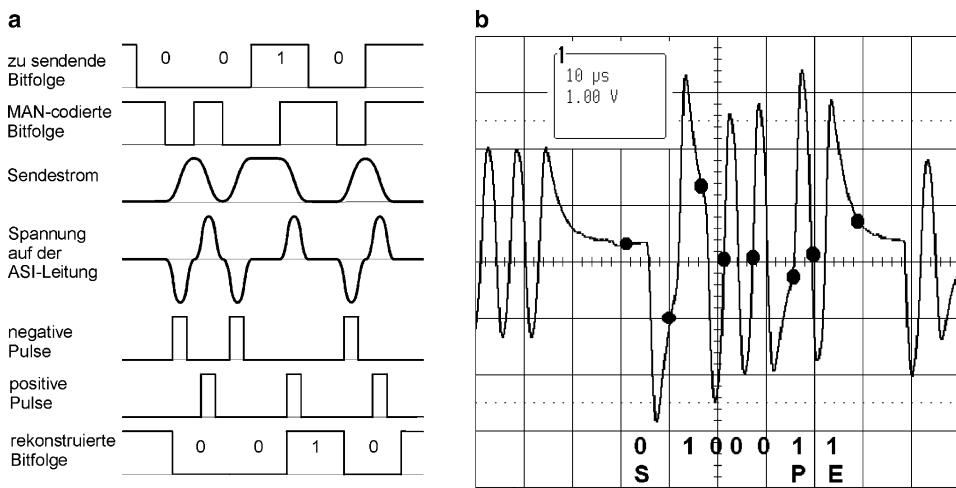


Abb. 4.3 Übertragung einer AS-i-Nachricht. **a** Die Polarität des Pulses in der 2. Bithälfte bestimmt den Bitwert 0/1. **b** Oszilloskopogramm einer 7-Bit-Slave-Antwort (S – Startbit, 1000 – Daten, P – Paritätsbit, E – Endbit)

gen, die eine höhere Spannung als die eigene Versorgungsspannung aufweisen. Auf der Empfangsseite werden diese Spannungssignale auf der Busleitung detektiert und wieder in die gesendete Bitfolge zurückgewandelt.

Da die Spannungspulse näherungsweise wie \sin^2 -Pulse geformt sind, wird damit auch den Forderungen nach niedriger oberer Grenzfrequenz und deshalb geringer Störabstrahlung Rechnung getragen. Mit diesem Modulationsverfahren und den geforderten Leitungen und Topologien sind Bitzeiten von $6 \mu\text{s}$ realisierbar.

4.1.1.8 Buszugriffsverfahren

Da mit AS-Interface sternförmige Punkt-zu-Punkt-Verbindungen ersetzt werden sollen, wurde ein Buszugriffsverfahren gewählt, das diese Art der Verbindung nachbildet und eine definierte Reaktionszeit gewährleisten kann: Der Master-Slave-Zugriff mit zyklischem Polling. Der Master sendet ein Telegramm, das eine bestimmte Slaveadresse enthält und der mit dieser Adresse angesprochene Slave antwortet innerhalb der dafür vorgesehenen Zeit. Im Normalbetrieb wird der Master die Slaves der Reihe nach zyklisch ansprechen, ihnen Daten übermitteln und/oder von ihnen Daten empfangen.

Dieses Verfahren ermöglicht den Bau sehr einfacher und damit kostengünstiger Slaves und bietet gleichzeitig größtmögliche Flexibilität. Zum einen kann der Master im Falle einer kurzzeitigen Störung auf der Busleitung einzelne Telegramme, auf die er keine oder keine gültige Antwort empfangen hat, wiederholen. Dabei ist es nicht notwendig, den gesamten Zyklus noch einmal ablaufen zu lassen. Zum anderen ist es möglich, zu bestimmten Zeiten in den Strom der Datentelegramme azyklische Parameteraufrufe und Organisationsaufrufe einzustreuen, ohne dadurch die Zykluszeit nennenswert zu verlängern. Zum dritten passt sich die erreichbare Zykluszeit automatisch an die Zahl der angeschlossenen Slaves an: Werden etwa nur 6 Slaves an den Bus angeschlossen, kann eine Zykluszeit von 1 ms erreicht werden, im Maximalausbau mit 31 Slaves beträgt sie 5 ms.

Die Bruttoübertragungsrate von AS-Interface beträgt einschließlich aller funktionsnotwendiger Pausen 167 kbit/s. Diese hohe Übertragungsrate ohne Schirm und Endwiderstand konnte nur erreicht werden, weil das Telegramm möglichst kurz gehalten wird. Das Mastertelegramm beinhaltet 14 Bit, das Slavetelegramm 7 Bit (Abb. 4.4). Mit dem Steuerbit und dem Bit I4 wird zwischen den zyklischen Datentransfer und den azyklischen Parameteraufrufen oder anderen Kommandos unterschieden.

Seit der Spezifikation 2.11 mit 62 Slaves unterscheidet sich das Mastertelegramm gegenüber der früheren Spezifikation 2.04 bei Bit I4. Dieses Bit ist normalerweise ein Ausgangsdatenbit, das in der neuen Spezifikation genutzt wird, um die Anzahl der Slaves zu verdoppeln. Dieses Bit wird als „Select“-Bit bezeichnet.

Spezifikation 2.04:	I4	I3	I2	I1	I0
Spezifikation 2.11:	Sel	I3	I2	I1	I0

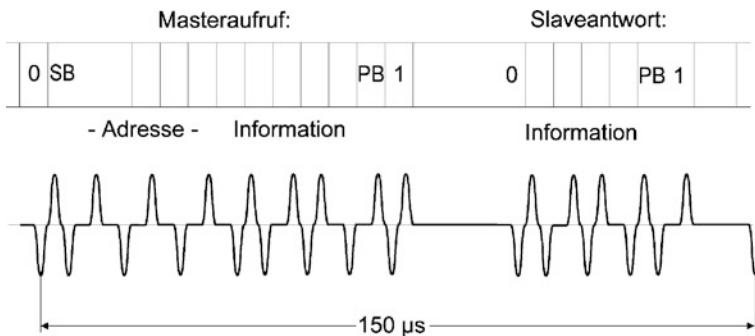


Abb. 4.4 Master- und Slavetelegramm

4.1.1.9 A/B- und Single Slaves

Das Select-Bit besitzt zwei unterschiedliche Zustände, low bzw. 0 und high bzw. 1. Jedem dieser beiden Zustände sind 31 Slaves zugeordnet. Die Slaves 1–31 low werden per definitionem als A-Slaves bezeichnet, die Slaves 1–31 high als B-Slaves. Demzufolge existiert als niedrigste Slaveadresse 1A und als höchste 31B. Diese Art der Darstellung erlaubt eine 100 %ige Kompatibilität zu der bisherigen AS-Interface-Adressierung. So werden die Adressen 1A–31A automatisch von den alten Mastern erkannt und wie die bisherigen Adressen 1–31 behandelt (Abb. 4.5).

Von der Nomenklatur her werden die Slaves, die eine volle Adresse benötigen, also alte Slaves nach Spezifikation 2.04, aber auch alle Sicherheitsslaves und viele andere moderne Slaves als Single Slaves (maximal 31), erweitert adressierbare als A/B-Slaves (maximal 62) bezeichnet. Bei Verwendung von Single Slaves stehen bei Modulen 4 Eingänge und 4 Ausgänge zur Verfügung. Mit 31 Slaves lassen sich dann insgesamt 124 Eingänge und 124 Ausgänge an einen AS-Interface-Master anschließen. A/B-Slaves besitzen ohne überlagertes Protokoll nur maximal 3 Ausgänge, da der vierte Ausgang als „Select“-Bit benötigt wird. Damit stehen bei 62 A/B-Slaves maximal 256 Eingänge und 186 Ausgänge zur Verfügung.

Die Anzahl der Slaves beeinflusst die Zykluszeit. Die bisherige Zykluszeit von 5 ms bei 31 Slaves verdoppelt sich auf 10 ms bei 62 Slaves. Der komplette AS-i-Zyklus wird in zwei Subzyklen aufgeteilt. Bei Nutzung aller 62 Adressen kommuniziert der Master zuerst mit dem Subzyklus der A Slaves (Adressen 1A–31A) und danach mit dem B-Slaves

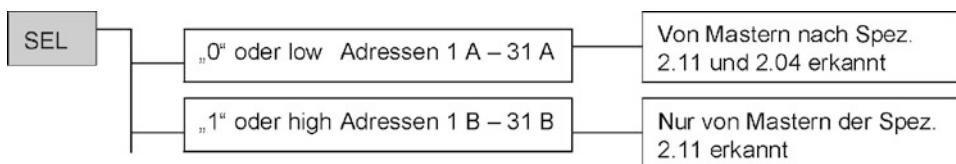


Abb. 4.5 Adressen der A/B-Slaves

(Adressen 1B–31B). Single Slaves werden sowohl im A- als auch im B-Subzyklus angesprochen, weshalb die oben bereits erwähnten 5-ms-Zykluszeiten beibehalten werden. Adressen, die nicht als Pärchen vorhanden sind, werden ebenfalls in beiden Subzyklen abgefragt und besitzen demzufolge ebenfalls eine Zykluszeit von 5 ms. Werden in einem Netzwerk Adressen gar nicht belegt, reduziert sich entsprechend die Zykluszeit, da der Master in der Startphase zuerst alle 62 Adressen abfragt, danach aber nur mit den vorhandenen Slaves kommuniziert. Pro Slave wird dabei eine Übertragungszeit von 150 µs veranschlagt.

Slaveprofil	CTT*	Adr.	Einge-führt mit	Master	Beschreibung
S-7.1	1	Single	Spec. 2.04	–	
S-7.2	1	Single	Spec. 2.04	–	
S-7.3	1	Single	Spec. 2.11	M3	1,2,4-kanalige Analogeingangsmodule 1,2,4-kanalige Analogausgangsmodule
S-7.4	1	Single	Spec. 2.11	M3	1,2,4-kanalige Analogeingangsmodule mit Parametrierung 1,2,4-kanalige Analogausgangsmodule mit Parametrierung Binäre 4E/4A Module mit Parametrierung
S-7.5.5	2	Single	Spec. 3.0	M4	Binäre 2E/2A Module mit 1,2 oder 4 analogen Eingängen und/oder Ausgängen und Parametrierung
S-7.A.5	2	A/B	Spec. 3.0	M4	Binäre 2E/1A Module mit 1 oder 2 analogen Eingängen und/oder Ausgängen und Parametrierung
S-B.A.5	2	A/B	Spec. 3.0	M4	Module mit 1 oder 2 analogen Eingängen und/oder Ausgängen und Parametrierung
S-7.A.7	3	A/B	Spec. 3.0	M4	Binäre 4E/4A Module mit erweiterter Adressierung
S-7.A.A	3	A/B	Spec. 3.0	M4	Binäre 8E/8A Module mit erweiterter Adressierung (Daten liegen im Analogbereich)
S-7.A.8	4	A/B	Spec. 3.0	M4	Analogeingangsmodul mit einem Kanal und einem binären Ausgang
S-7.A.9	4	A/B	Spec. 3.0	M4	2-kanaliges Analogeingangsmodul
S-6.0.x	5	Single	Spec. 3.0	M4	Modul mit einem analogen Eingang und einem analogen Ausgang belegt mehrere Adressen

* CTT = Combined Transaction Type

4.1.1.10 Analogwertübertragung

Obwohl das AS-Interface mit nur 4 Datenbits pro Telegramm arbeitet, lassen sich Analogtelegramme übertragen. Dabei wird der Analogwert in mehrere kleinere Portionen zerlegt, mit Steuerinformationen versehen und in mehreren AS-i-Zyklen übertragen. Es

gibt verschiedene Ausprägungen der Analogwertübertragung, die durch unterschiedliche Slaveprofile gekennzeichnet sind.

Die Tabelle zeigt eine Übersicht der existierenden Slaveprofile.

Mit Profil wird dabei die Kombination von 4 4-Bit-Werten bezeichnet (IO, ID, ID1, ID2), die der Slave vom Hersteller erhält. Damit ist der AS-Interface Master in der Lage, einen Slave zu identifizieren und einer bestimmten Funktion zuzuordnen.

Der Combined Transaction Type kennzeichnet ein bestimmtes Protokoll zur Übertragung der Analogwerte. Der Combined Transaction Type ist nur für die Hersteller von Master und Slaves wichtig, der Anwender bemerkt ihn nicht. Die CTTs sind in der AS-Interface-Spezifikation beschrieben.

Die Adressierung beschreibt, ob ein Slave erweitert (A/B) oder standardmäßig adressiert wird.

Die Spalte Spezifikation gibt an, wann das betreffende Profil in die AS-Interface-Spezifikation eingeführt wurde. Die Spalte Master gibt an, welche Master-Profile das betreffende Slave-Profil unterstützen. Die Beschreibung gibt die wichtigsten Möglichkeiten des betreffenden Slave-Profiles an. Man erkennt, dass es eine Vielzahl von Protokollen gibt, die zunächst vermuten lassen, dass die Übertragung von Analogwerten bei AS-Interface sehr kompliziert ist. Dies ist nur vordergründig der Fall:

Da die CTTs und die Profile sehr präzise dokumentiert sind, können sie vom Master ohne Anwenderkonfiguration oder sonstige Installation abgearbeitet werden. AS-i Slaves nach den oben genannten Profilen sind also „Plug-and-Play“. Die Analogdaten werden vom Master in zwei Speicherbereichen dem „Analog Input Data Image“ und dem „Analog Output Data Image“ abgelegt. Dabei stehen für einen Single Slave je 4 16-Bit-Werte und für einen A/B Slave je 2 16-Bit-Werte zur Verfügung. Beispielhaft ist nachfolgend die Übertragung entsprechend Combined Transaction Type 1 dargestellt:

Jeder Analogwert wird in einzelne Datentripel zerlegt, die in aufeinander folgenden Zyklen versendet werden. Mit dem vierten Datenbit, dem Handshake-Bit, werden die zugehörigen Datentripel zugeordnet.

Abb. 4.6 zeigt das Prinzip dieser Übertragung im Detail. Das msb der 4 Slavebits wird als Handshake-Bit verwendet, indem sowohl der Host als auch der Slave bei jeder Übertragung dieses Bit negieren. Im Beispiel startet der Master seine Übertragung mit dem Invertieren des Handshake-Bits H von 1 nach 0. Daran erkennt der Slave, dass die Ausgangsbits O1...O3 aktualisiert wurden, und liest diese. Als Antwort werden die Eingangsbits I1...I3 übertragen, was durch den Wechsel von H nach 1 angezeigt wird und im Master erkannt wird. Der Master startet danach die nächste Übertragungs-Portion durch erneuten Wechsel der Polarität des H-Bits, bis der komplette Analogwert übertragen wurde. Danach startet die Sequenz von neuem.

Bei CTT1 Übertragungen nach Profil S-7.3 (das vermutlich in der Praxis am häufigsten genutzte Profil), werden grundsätzlich 16-Bit Werte übertragen. Ein 16-Bit-Analogwert benötigt 7 AS-Interface-Zyklen. Bei einer Zykluszeit von 5 ms ergibt sich für ein 12-Bit-Analogtelegramm eine Übertragungszeit von 35 ms.

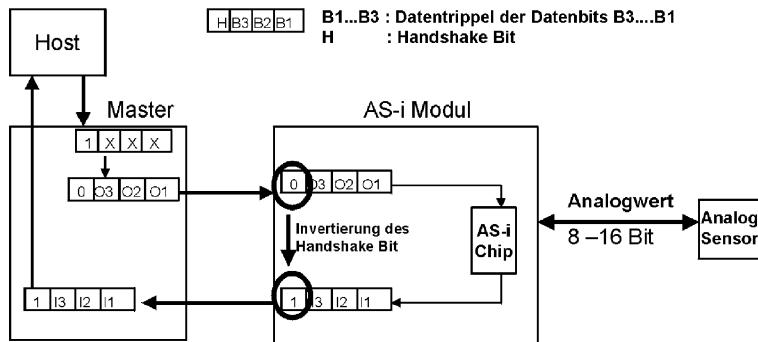


Abb. 4.6 Prinzip der Analogwertübertragung

4.1.1.11 Datensicherheit

Durch den Verzicht auf Endwiderstände (Wellenabschlusswiderstände) und ein geschirmtes Kabel erhält die elektronische Datensicherung eine große Bedeutung. Feldbusssysteme haben üblicherweise eine Hamming-Distanz von $HD = 4$ (Abb. 4.7).

Um die gleiche Datensicherheit konventionell zu gewährleisten, müsste die Anzahl der Bits beim Master- und Slavetelegramm drastisch erhöht werden. Da dies zu Lasten der Nettodatenrate geht, werden die Prüfungen in dem Physical Layer, der Schicht 1 des OSI Referenzmodells, vorgenommen. Folgende Kriterien werden überprüft:

- Signalpegel des Kurvenverlaufes
- Abfolge von positiven und negativen Flanken
- Zeitverlauf des Telegramms
- Parität (Paritätsbit).

Bei einem fehlerhaften Telegramm wiederholt der Master direkt seine Anfrage. Wird das Telegramm ein zweites Mal nicht erkannt, wird eine zusätzliche Wiederholung im nächsten Zyklus vorgesehen. Damit wird gewährleistet, dass auch bei mehreren Fehlern die Zykluszeit nicht wesentlich erhöht wird. Bei einem Fehler pro Zyklus erhöht sich die Zykluszeit (bei 32 Slaves) von 5,1 ms auf 5,25 ms, bei 10 Fehlern pro Zyklus auf nur 6,6 ms.

4.1.1.12 Elektromagnetische Verträglichkeit

Wenn ein Bussystem mit einer relativ hohen Datenübertragungsrate über eine ungeschirmte Leitung betrieben werden soll, so ist natürlich die Frage nach der elektromagnetischen Verträglichkeit von zentraler Bedeutung. Hier können zwei Fälle voneinander unterscheiden werden:

- die Störausstrahlung bzw. die Funkstörfeldstärke, die nach der EMV-Richtlinie der EU die Grenzwerte nach EN 61000-6-3 nicht überschreiten darf und

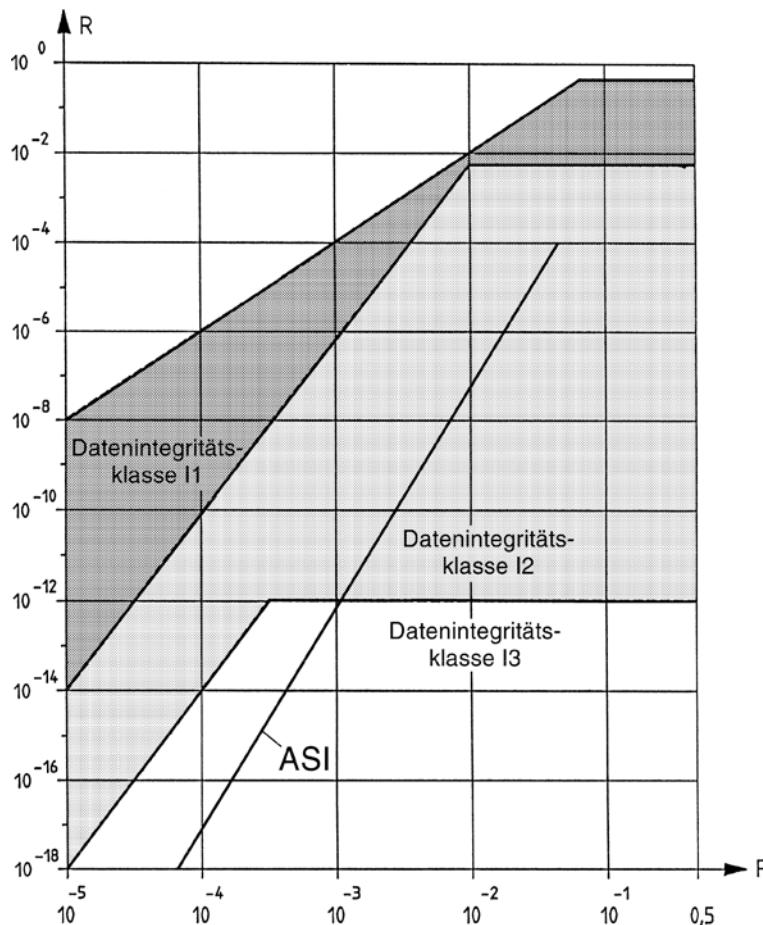


Abb. 4.7 AS-Interface in der Datenintegritätsklasse nach DIN 19244. (p – Bitfehlerwahrscheinlichkeit, R – Restfehlerwahrscheinlichkeit)

- die Störfestigkeit des Systems unter Einwirkung von Störungen, wie sie in der EN 61000-6-2 für den Einsatz in industrieller Umgebung beschrieben sind.

Ausführliche Tests bei den Herstellern von AS-Interface-Produkten sowie Einsatzerfahrungen an zahlreichen Installationen von AS-Interface-Systemen in den verschiedensten Branchen der industriellen Produktionstechnik haben gezeigt, dass die Störausstrahlung trotz Verwendung des nicht abgeschirmten Kabels sicher unter den vorgeschriebenen Grenzwerten bleibt. Die Störfestigkeit gegen elektrostatische Entladungen (ESD), gegen elektromagnetische Felder im Frequenzbereich zwischen 80 MHz und 1 GHz, gegen schnelle transiente Störgrößen (Bursts) sowie gegen leitungsgebundene Störungen im Frequenzbereich zwischen 150 kHz und 80 MHz ist bis zum Schärfegrad 3 gegeben. Im

ungünstigsten Fall werden einzelne AS-Interface-Nachrichten gestört. Dies wird jedoch vom System erkannt und eine Wiederholung veranlasst, was in der Regel sogar innerhalb der Zykluszeitgrenzen möglich ist, also vom Anwender gar nicht bemerkt wird.

Inzwischen sind auch Applikationen von AS-Interface erfolgreich durchgeführt worden, bei denen ein Teil des AS-Interface-Netzes auf einem beweglichen Maschinen- oder Anlagenteil angeordnet und über Schleifringe mit den übrigen Komponenten verbunden worden ist. Hier zeigt sich besonders eindrucksvoll, dass das Übertragungsverfahren von AS-Interface einfach (es werden nur 2 Schleifringe für Daten- und Energieübertragung benötigt), sicher und störunempfindlich ist.

4.1.1.13 AS-Interface im explosionsgefährdeten Bereich

Nachdem sich das AS-Interface im Bereich der Fabrikautomation als Standard etabliert hat, wurde seitens der Industrie auch im Grenzbereich zur Prozessautomation Interesse gezeigt. In der Farbabfüllung beispielsweise ist die ganze Fördertechnik und Verpackung oft ein nicht-explosionsgefährdeter Bereich. Lediglich der Bereich der Farbabfüllung selbst ist als explosionsgefährdet gekennzeichnet. Speziell für diesen Einsatz wurde eine Modulreihe in der Ausführung EEx i entwickelt. Damit kann ein vorhandenes AS-Interface-Netz direkt in den explosionsgefährdeten Bereich hinein erweitert werden. Von diesen Grenzbereich ausgehend wird das AS-Interface auch in der reinen Prozessautomation eingesetzt und stellt eine Alternative für eigensichere Feldbusse wie z. B. dem PROFIBUS-PA dar.

Bei den eigensicheren Feldbussen wird die über den Bus zur Verfügung gestellte Energie durch entsprechende Einspeisebaugruppen soweit begrenzt, dass auch im Fehlerfall keine zündfähigen Funken oder Lichtbögen entstehen können. Dieser Ansatz begrenzt die Anzahl der an den Bus anschaltbaren Teilnehmer beträchtlich und verursacht relativ niedrige Datenraten. Um alle Vorteile, die AS-Interface bietet, auch für Applikationen in explosionsgefährdeten Bereichen (Ex-Bereich) nutzen zu können, geht man bei AS-Interface einen anderen Weg. Um einen Standard AS-Interface-Strang in den Ex-Bereich verlegen zu können, wählt man eine Verkabelung nach EEx e (erhöhte Sicherheit). Hierbei werden zwar besondere Anforderungen an die Art der Verkabelung, die Auswahl der Klemmen und Verschraubungen gelegt, nicht jedoch an die eingespeiste Energie.

Die Verkabelung muss einfach mechanisch geschützt werden. Das Kabel kann dazu in einen Kabelkanal verlegt werden oder alternativ kann ein armiertes Kabel, das der AS-Interface-Spezifikation genügt, verwendet werden. Weiterhin darf der separate Klemmraum des AS-Interface-Kabels nicht während des laufenden Betriebs geöffnet werden. Dies hat in der Regel keine Auswirkung, da die Verkabelung nur einmal bei der Installation vor der Inbetriebnahme durchgeführt wird.

Im laufenden Betrieb lassen sich Sensoren und Aktoren jederzeit austauschen, da der separate Klemmraum in der Zündschutzart EEx i (EN 60947-5-6) ausgelegt wurde. Die eigensichere Elektronik des Moduls ist in der Zündschutzart EEx m (Vergusskapselung) aufgebaut (Abb. 4.8).

Nach diesem Konzept kann ein AS-Interface-Strang ohne Segmentkoppler oder andere sonst notwendige Bauteile frei zwischen dem explosionsgefährdeten und nicht-

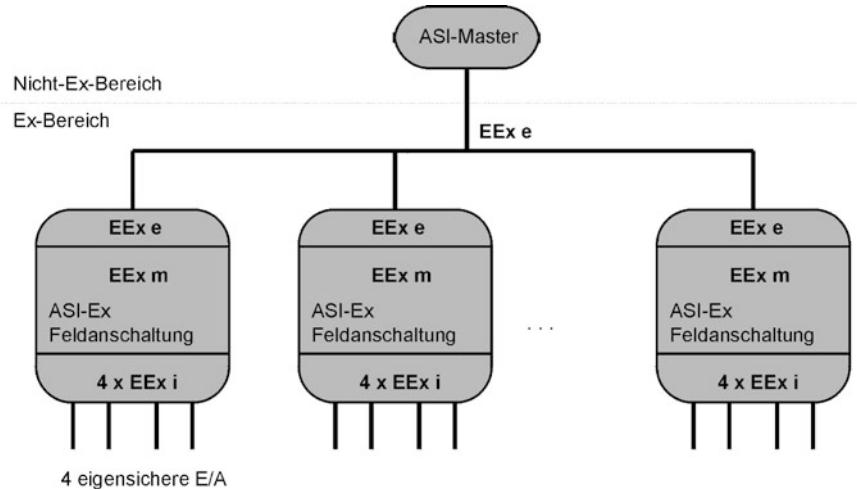


Abb. 4.8 Konzept des Explosionsschutzes AS-Interface-Ex

Abb. 4.9 AS-Interface-Modul für den explosionsgefährdeten Bereich (Werkbild Pepperl+Fuchs)



explosionsgefährdeten Bereich verlegt werden. Die Module werden in den Varianten vier Namur-Eingänge oder vier eigensichere Ausgänge oder vier Namur-Eingänge und zwei eigensichere Ausgänge angeboten (Abb. 4.9). Letztere sind speziell für den Anschluss von Ventilsteuерungen vorgesehen, die in der Regel zwei binäre Eingänge und einen binären Ausgang haben. Somit können pro Modul zwei Ventile angeschlossen werden.

4.1.1.14 Neuerungen nach der Spezifikation 3.0

Die vorherige Spezifikation 2.1 hatte als Hauptneuerungen folgende Erweiterungen eingeführt:

- A/B Addressierung, um 62 Slaves ansprechen zu können.
 - Bessere Identifizierung der Slaves durch Erweiterung der ID-Codes um ID1 und ID2.

- Analogprofile S-7.3 und S-7.4
- Peripheriefehlerbit, mit dem Slaves Diagnosemeldungen absetzen können.

Die aktuelle Spezifikation 3.0 hat in erster Linie eine Vielzahl neuer Slaveprofile eingeführt, die vor allem eine flexiblere Übertragung von Analogwerten erlauben. Außerdem wurde das Profil S-7.A.7 eingeführt, das die Übertragung von 4 Eingangs- und 4 Ausgangsbits mit A/B-Slaves erlaubt (auf Kosten der Zykluszeit).

Dabei wurde auf maximale Kompatibilität Wert gelegt, sodass sich alle Slaves nach allen Spezifikationen problemlos mit den neuesten Mastern betreiben lassen.

4.1.1.15 Safety at work

Das Sicherheitskonzept von AS-Interface ermöglicht die Integration von sicherheitsrelevanten Komponenten wie Not-Aus-Schaltern, Schutztürkontakte, Sicherheitslichtschranken oder -Lichtgitter u. v. m. in ein AS-Interface-Netzwerk. Das Konzept erfüllt die Voraussetzung eines Sicherheitsbusses und darf bis zur Sicherheitskategorie 4 gemäß EN 954-1, SIL3 nach IEC 61508 und EN 62061 bzw. PLe nach EN13849 eingesetzt werden. Sicherheitsrelevante Slaves können parallel zu den Standard AS-Slaves an das gleiche AS-Interface-Kabel angeschlossen werden. Das AS-Interface-Netz, bestehend aus den konventionellen Komponenten wie Netzteil, Master und Slaves, wird um den Sicherheitsmonitor und die Sicherheitsslaves erweitert (Abb. 4.10).

In ein konventionelles AS-Interface-Netz können mehrere Sicherheitsmonitore und maximal 31 Sicherheitsslaves integriert werden. Auch bei Nutzung des neuen Chips sind nur 31 Adressen vorgesehen.

Der Sicherheitsmonitor besitzt je nach Bauart ein bis vier sicherheitsgerechte Ausgänge und kann noch bis zu 32 sichere Ausgänge ansteuern. Der Anwender ordnet die

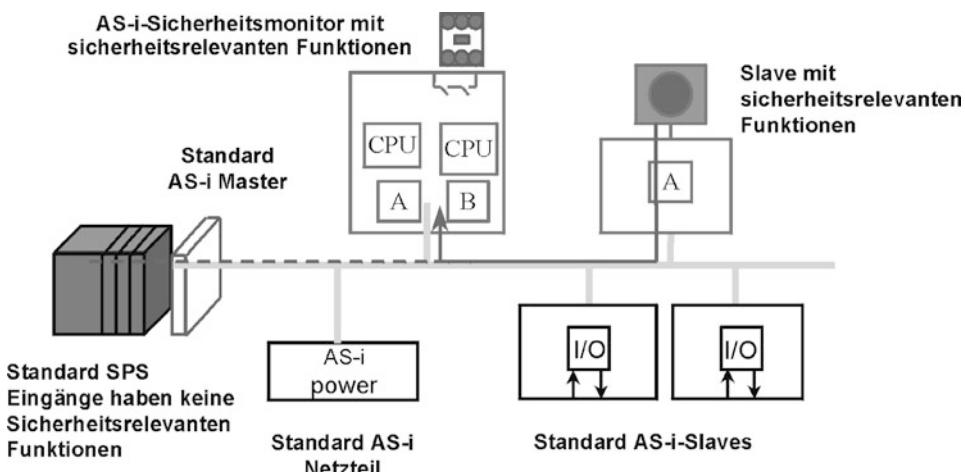


Abb. 4.10 Safety-at-work-Konzept

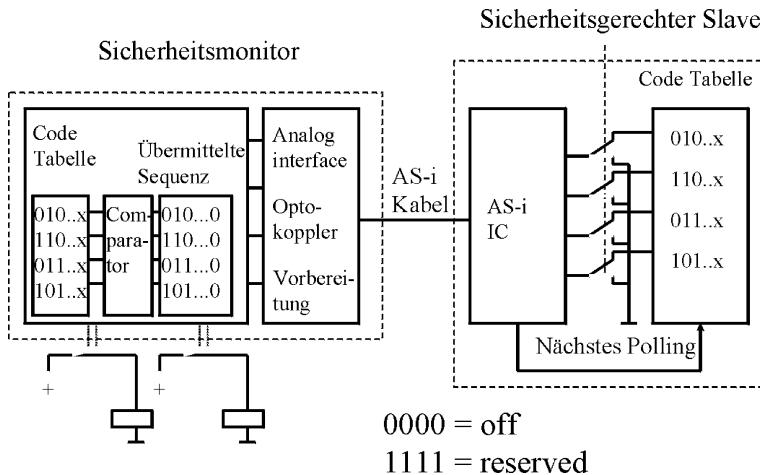


Abb. 4.11 Übertragungsverfahren für sicherheitsgerichtete Daten über AS-i

Sicherheitsslaves den Ausgängen bzw. den Sicherheitsmonitoren zu. Das System ist frei konfigurierbar. Der AS-Interface-Master überwacht die Sicherheitsslaves gleichwertig den Standardslaves. Dadurch ist er in der Lage, als Reaktion auf ein sicherheitsrelevantes Signal einen konventionellen Ausgang zu setzen. Wird z. B. ein Not-Aus-Schalter betätigt, kann und darf nur der Sicherheitsmonitor sicherheitsrelevant reagieren und über den sicherheitsgerechten Ausgang die Anlage abschalten. Da der konventionelle Master das Signal interpretieren kann, ist er in der Lage, entsprechend eines SPS-Programmes z. B. eine Signalleuchte auf rot umzuschalten.

In der Startphase synchronisiert sich der Sicherheitsmonitor mit den Sicherheitsslaves, für die er zuständig ist. Dann startet er die Kommunikation und überwacht parallel die eingehenden Telegramme. Dabei erwartet er innerhalb eines bestimmten Zeitrasters von allen Sicherheitsslaves 4-Bit-Telegramme mit einer bestimmten individuellen Codefolge bestehend aus 8 Werten. (Abb. 4.11). Diese Werte erfüllen bestimmte Bedingungen, um alle Fehler möglichst schnell aufzudecken. Alle geeigneten Code-Folgen wurden mathematisch ermittelt und werden von der Herstellervereinigung verwaltet und an die Hersteller von Slaves verteilt. Innerhalb eines AS-i Kreises darf jede Codefolge nur einmal vorkommen.

Wird das Zeitraster nicht eingehalten oder ein falsches Telegramm empfangen (z. B. bei einem Kabelbruch oder Wackelkontakt an der AS-i-Leitung), setzt der Sicherheitsmonitor den sicherheitsgerichteten Ausgang. Der Ausgang wird ebenfalls gesetzt, wenn vom Slave das Telegramm für den Alarmfall 0-0-0-0 (z. B. Not-Aus aktiviert) gesendet wird. Die Reaktionszeit beträgt 45 ms. Parallel kann der Sicherheitsmonitor den konventionellen Master informieren. Dies ist davon abhängig, wie der Monitor konfiguriert wurde. Er kann als aktiver oder passiver Teilnehmer des AS-Interface-Netzes fungieren. Als aktiver Teilnehmer arbeitet der Sicherheitsmonitor zusätzlich als konventioneller AS-Interface-Slave.

Dem Monitor wird eine Slaveadresse gegeben, über die er wie jeder andere AS-Interface-Slave mit dem konventionellen Master kommunizieren kann.

Auch bei den Sicherheitsslaves gibt es zwei verschiedene Typen. Zum einen Schalter bzw. Sensoren mit integriertem AS-Interface-Chip und zum anderen Module mit 2 bis 4 Kanälen, an die standardmäßige Sicherheitsslaves angeschlossen werden können.

Seit Ende 2007 ist die Definition und Zulassung von sicheren AS-i-Ausgängen abgeschlossen. Erste Produkte sind 2008 auf den Markt gekommen.

Für jeden sicheren Ausgangskanal wird eine AS-i-Adresse reserviert, die ganz konventionell vom Master aufgerufen wird, aber vom Sicherheitsmonitor beantwortet wird. Soll der sichere Ausgangskanal einschalten, sendet der Monitor eine Codefolge, die für jede sichere Adresse unverwechselbar definiert ist. Diese Codefolgen haben im Unterschied zu den Codefolgen der sicheren Eingangs-Slaves nur 7 Werte. Soll abgeschaltet werden, wird analog der Eingangs-Slaves eine Folge aus Nullen gesendet.

Die Codefolge wird von den dezentralen sicheren Ausgangsmodulen mitgehört. Wenn die zur sicheren Adresse gehörige Codefolge erkannt wird, schalten die sicheren Ausgangsmodule ein. Wichtig ist, dass die Ausgangsmodule die Codefolgen selbstständig überprüfen müssen und dazu selbst entsprechend zuverlässig aufgebaut sein müssen. Im Grunde genommen sind die Ausgangsmodule vereinfachte Sicherheitsmonitore. Die Komplexität der sicheren Ausgangsmodule ist sehr viel höher als die der Eingangsmodule.

Da die sicheren Ausgangsmodule selbst nicht kommunizieren (sie hören ja nur zu wie ein Monitor), werden sie meist mit einer zweiten nicht sicherheitsgerichteten Adresse kombiniert, mit deren Hilfe das Ausgangsmodul Diagnoseinformationen austauschen kann.

Neben der beschriebenen Grundfunktion, Codefolge oder 0-Folge, kann der Monitor auch noch zwei so genannte Hilfssignale übertragen, die in die Codefolge hineinmoduliert werden. Mit diesen Hilfssignalen können Fehlerzustände zurückgesetzt werden und Startsignale an die Ausgangsmodule gegeben werden.

Es gibt mittlerweile auf dem Markt Safety-Monitore für bis zu 32 Ausgangskanäle („Freigabekreise“). Die Einsatzmöglichkeiten von AS-i SaW vergrößern sich durch die Ausgangsmodule ganz erheblich.

4.1.2 Das KNX-System für die Haus- und Gebäudesystemtechnik

4.1.2.1 Einführung

Höhere Ansprüche an Sicherheit, Flexibilität und Komfort der Elektroinstallation, verbunden mit der Forderung nach Erhöhung der Energieeffizienz, haben bereits Anfang der 1990er Jahre zur Entwicklung der Gebäudesystemtechnik auf Basis des Europäischen Installationsbus (EIB) geführt. Eine ähnliche Entwicklung mit den gleichen Zielen lief insbesondere im französischen Sprachraum und führte zur Entwicklung des Batibus. Die European Home Systems Association (EWSA) wiederum arbeitete bereits frühzeitig an den Grundlagen der Vernetzung von Hausgeräten (weiße Ware). Der Zusammenschluss

der bisherigen Gesellschaften (EIBA, Batibus und EHSA) zur KNX Association hat den Weg zum weltweit einzigen offenen Standard für Haus und Gebäudesystemtechnik gegeben, dem KNX Standard. KNX ist als Europäischer Standard (CENELEC EN 50090 und CEN EN 13321) und als Internationaler Standard (ISO/IEC 14543-3) anerkannt.

Das KNX-System ist ein offenes, umfassendes Bussystem, das alle Aspekte der Haus- und Gebäudeautomation einschließt. Die Systemverantwortung liegt bei der unabhängigen KNX Association in Brüssel. BAU-Bausteine (Bus Access Unit = Buszugriffsgerät) für den Einsatz mit KNX werden von den unterschiedlichsten Herstellern angeboten. Dennoch, KNX ist in erster Linie eine Spezifikation und nicht eine Ausführung, wie z. B. ein ganz bestimmter Chip oder ein Transceiver. KNX ist somit ein offenes Bussystem: Es kann von jedem Anwender auf jeder gewählten Chip- oder Prozessorplattform realisiert werden – sowohl als Netzwerkgrundlage für eigene, individuelle Produkte als auch für auf OEM-Basis bezogene BAU-Geräte. KNX-spezifische Konformitätsprüfungen sind definiert; die KNX-Zertifikation kann von jedem KNX-Mitglied erworben werden.

Warum nur „in erster Linie“ eine Spezifikation? ... weil KNX das Protokoll in ein allumfassendes System der Haus- und Gebäudesystemtechnik einbettet, komplett mit standardisierten Systemkomponenten (wie BAUs), Netzwerkverwaltung und Interworking-Standards, mit herstellerunabhängigen Werkzeugen und Programmierschnittstellen für PCs, Schulungen für Elektroinstallateure, Zertifizierungsabläufe usw.

4.1.2.2 Netzwerktopologie

Das KNX-System ist ein eindeutiges „Peer-to-Peer“-Netzwerk mit einer Kapazität von bis zu 65 536 Busgeräten (Abb. 4.12). Die logische Topologie erlaubt den Anschluss von bis zu 256 Geräten an eine Buslinie. Bis zu 15 Linien können über eine *Hauptlinie* zu einem *Bereich* zusammengefasst werden. Bis zu 15 solcher Bereiche bilden über eine so genannte *Bereichslinie* eine größere Einheit.

Bei offenen Übertragungsmedien werden nahe zusammenliegende Einheiten durch eine 16-Bit Systemidentifikation logisch voneinander getrennt. Unter Berücksichtigung der für die Koppler reservierten Adressen können in einem KNX-Netz 61 455 Endgeräte ($255 \times 16 \times 15 + 255 = 61\,455$) miteinander verbunden werden. Eventuelle Anschlussbeschränkungen ergeben sich aus den Peripheriegegebenheiten (Übertragungsmedium, Transceiver-Typen, Kapazität der Spannungsversorgung) und den Umgebungsbedingungen (elektromagnetische Störungen, ...). Die Installations-, Anschluss- und Produktrichtlinien sollten in jedem Fall berücksichtigt werden.

Koppler verbinden Linien oder Segmente, zum Beispiel im Rahmen des TP-Übertragungsmediums (TP = Twisted Pair = verdrillte Zweidrahtleitung) oder anderer Übertragungsmedien. Sie können dabei die Funktionen von Verstärkern, Brücken, Paketfiltern (zur Optimierung des Datenverkehrs), Routern usw. oder Firewall ausüben, oder eine Kombination aus diesen Einzelfunktionen übernehmen. KNX definiert eine Reihe von Standardprofilen für Koppler.

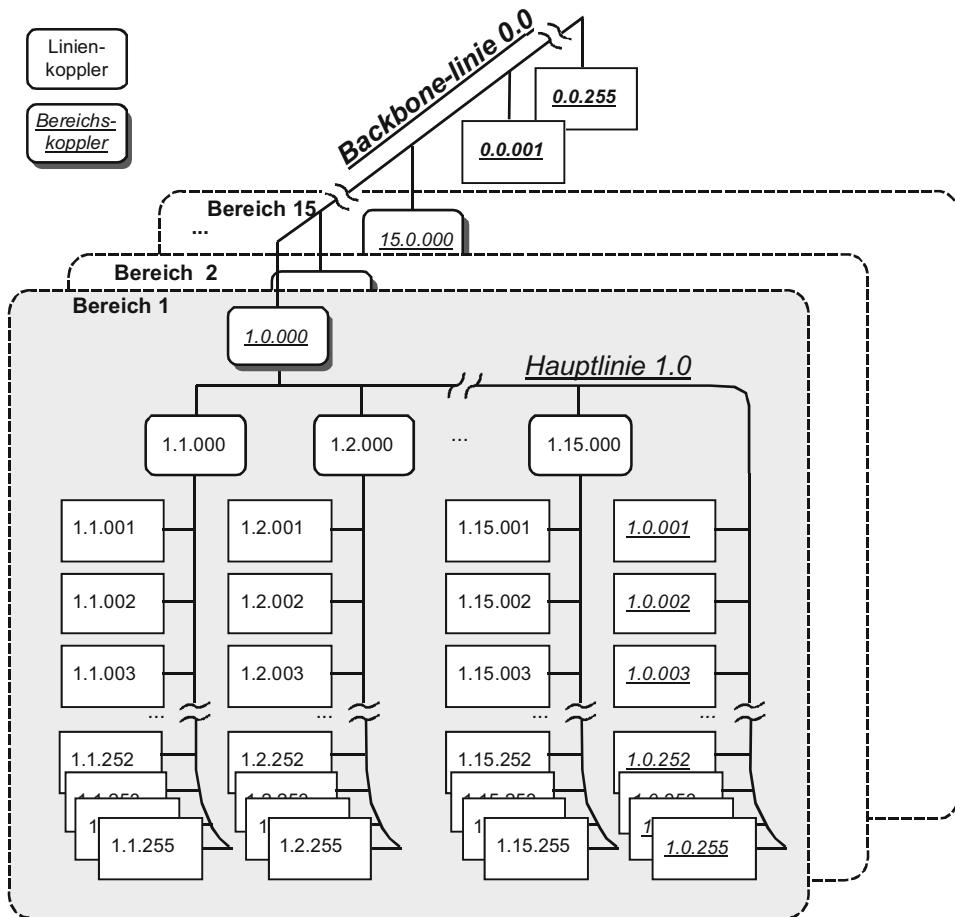


Abb. 4.12 Die logische Topologie von KNX

4.1.2.3 Übertragungsmedien

Die KNX-Zugriffssteuerung ist für jedes einzelne Übertragungsmedium optimal ausgetragen. Dank der unterschiedlichen zur Verfügung stehenden Übertragungsmedien kann das Verhältnis Transceiverleistung/Kosten für jeden Anwendungsfall optimiert werden.

Verdrillte Zweidrahtleitung (Twisted Pair TP)

Im Übertragungsmedium KNX TP wird durch die Kollisionserkennung auf Bitebene mit dominantem logischen Nullimpuls sichergestellt, dass im Falle einer Kollision immer einer der beteiligten Kommunikationspartner erfolgreich senden kann. Wiederholte Übertragungsversuche werden dadurch vermieden, was wiederum zu einer zusätzlichen Leistungssteigerung des Mediums KNX TP führt. Zusammen mit der leistungsstarken KNX „Gruppenadressierung“ sorgt die KNX TP Kollisionsvermeidung für einen extrem hohen

Nutzungsgrad mit Reaktionszeiten von 100 ms bei zwei simultan ablaufenden Übertragungen. Durch ein schnelles Abrufverfahren (polling) können innerhalb von 50 ms die 1 Byte-Statusinformationen von bis zu 14 Busteilnehmern abgefragt werden. Die maximale Gesamtlänge für ein physikalisches TP-Segment beträgt 1000 m.

Stromnetz (Power Line, PL)

Als weiteres Übertragungsmedium im KNX System kann das 230 V Starkstromnetz (Powerline KNX) verwendet werden. Für die Projektierung und Inbetriebnahme von Powerline KNX wird die ETS genutzt. Die Verlegung separater Busleitungen entfällt. Alle KNX-Powerline-Geräte benötigen lediglich den Anschluss des Außen- und Neutralleiters. Anwendungen von Powerline KNX sind in der Nachrüstung, aber auch in der Neuinstallation zu finden. Geräteabmessungen und Gerätebedienungen entsprechen weitestgehend den Twisted-Pair (TP)-KNX-Komponenten. Powerline KNX entspricht den dafür geltenden europäischen Normen, insbesondere der Reihe EN 50065 (Signalübertragung auf elektrischen Niederspannungsnetzen im Frequenzbereich 3 kHz bis 148,5 kHz), der Reihe EN 50090 (Elektrische Systemtechnik für Heim und Gebäude ESHG) und der EN 50491 Reihe (Allgemeine Anforderungen an die Elektrische Systemtechnik für Heim- und Gebäude ESHG und an Systeme der Gebäudeautomation GA). Sobald in bestehenden Anlagen die Installation einer separaten, zusätzlichen Busleitung aus verschiedenen Gründen nicht gewünscht oder möglich ist, eröffnet die Nutzung des bestehenden Starkstromnetzes 230/400 V neue Perspektiven.

Drahtlose Übertragung (Radio Frequency, RF)

KNX RF-Kanäle im 868 MHz ISM-Band sind durch unterschiedliche Trägerfrequenzen physikalisch voneinander getrennt. Unter Freifeldbedingungen beträgt die Übertragungsentfernung etwa 300 m. Durch die Verwendung von Verstärkern können auch umfangreiche Anordnungen innerhalb eines Gebäudes untergebracht werden. Das RF-Übertragungssystem ist so konzipiert, dass die Verstärkerfunktion für die angeschlossenen Busgeräte optimal eingesetzt wird.

Eine KNX Anlage kann sowohl aus einem reinen Funknetz als auch aus einer Kombination von Funk mit einem anderen Medium, wie z. B. KNX mit Twisted-Pair oder KNX Powerline, bestehen. Dafür stehen Medienkoppler zur Verfügung, mit denen Informationen und Befehle von Geräten in dem einen Übertragungsmedium auf Geräte im anderen Medium übertragen werden können.

IP Netzwerk als Übertragungsmedium

Für die Übertragung von KNX Telegrammen über das IP-Netzwerk stehen die Übertragungsprotokolle KNXnet/IP Tunneling und KNXnet/IP Routing zur Verfügung. KNXnet/IP Tunneling wird zur Punkt-zu-Punkt Verbindung zwischen zwei Geräten mit bekannten IP-Adressen verwendet, zum Beispiel zwischen der ETS mit der IP-Adresse 192.168.0.3 und einer IP-Schnittstelle mit der IP-Adresse 192.168.0.5.

KNXnet/IP Routing wird für die Kommunikation zwischen beliebig vielen KNX Geräten verwendet. Dazu wird die von der IANA der KNX Association für die Übertragung von KNX Telegrammen fest zugeordnete Multicast-Adresse 224.0.23.12 in Verbindung mit dem Port 3671 verwendet. Damit können z. B. anstatt Linienkopplern sogenannte IP Router und somit das IP-Netzwerk als Hauptlinie verwendet werden. Für KNXnet/IP Tunneling und KNXnet/IP Routing wird zur Übertragung der Informationen das UDP Protokoll verwendet.

4.1.2.4 KNX OSI Kommunikationsprotokoll

Abb. 4.13a zeigt die KNX Kommunikationshierarchie auf der Grundlage des aus 7 Schichten (Layers) bestehenden OSI-Referenzmodells. Auch der in Abb. 4.13b dargestellte Rahmenaufbau verdeutlicht dieses Prinzip. Die physikalische (physical layer) und die Sicherungsschicht (link layer) hängen offensichtlich von den Eigenschaften des physikalischen Übertragungsmediums ab. Für den Zugriff auf das Medium schreibt KNX den Carrier Sense Multiple Access with Optimized Collision Avoidance (CSMA/CA) vor. Wie bereits erläutert, kann man KNX in dieser Hinsicht als „opportunistisch“ bezeichnen, da das exakte Verfahren für das jeweilige Medium optimiert werden kann. Der Zieladressenmerker (Destination Address Flag, DAF) unterscheidet zwischen gruppen- und geräteorientierten Telegrammen. Über die NPCI-Steuernachricht (Network Protocol Control Information) steuert die OSI Vermittlungsschicht (network layer) die Anzahl der Sprünge (hops); für Geräte, die nicht als Router oder Brücken fungieren, ist diese Schicht nicht von Bedeutung. Die Transportschicht (transport layer) verwaltet logische Kommunikationsbeziehungen, wie z. B.

1. Einer-an-viele, verbindungslos („Gruppe“ Multicast)
2. Einer-an-alle, verbindungslos (Broadcast)
3. Einer-an-einen, verbindungslos
4. Einer-an-einen, verbindungsorientiert

Die Transportschicht ist für die Abbildung der Adressen in Form einer abstrakten internen Darstellung, der Communication_Reference_ID (cr_id), verantwortlich. Alle Dienste werden transparent über die Verbindungsschicht (session layer) und die Anpassungsschicht (presentation layer) weitergegeben; diese beiden Schichten sind reserviert. Die Anwendungsschicht (application layer) liefert die API-Anwendungsschnittstelle (Application Interface) für die Client/Server-Verwaltung der KNX-Netze. Die Gruppenanwendungsschicht (group application layer) befasst sich mit der Zuweisung einer „cr_id“-Gruppenidentifikation an die lokale Instanz eines Gruppen-Kommunikationsobjektes (oder einer verteilten Variablen), und zwar sowohl für den Empfangsvorgang (einer-an-n) als auch für den Sendevorgang (einer-an-einen). Zur Vereinfachung sind Gruppen-Kommunikationsobjekte und verteilte Objekte in der KNX-Anwenderschicht (KNX user layer) zusammengehalten, die der Anwendung die sonst mühselige Kleinarbeit der Anwendungsschicht (application layer) abnimmt. Diese Anwenderschicht übernimmt

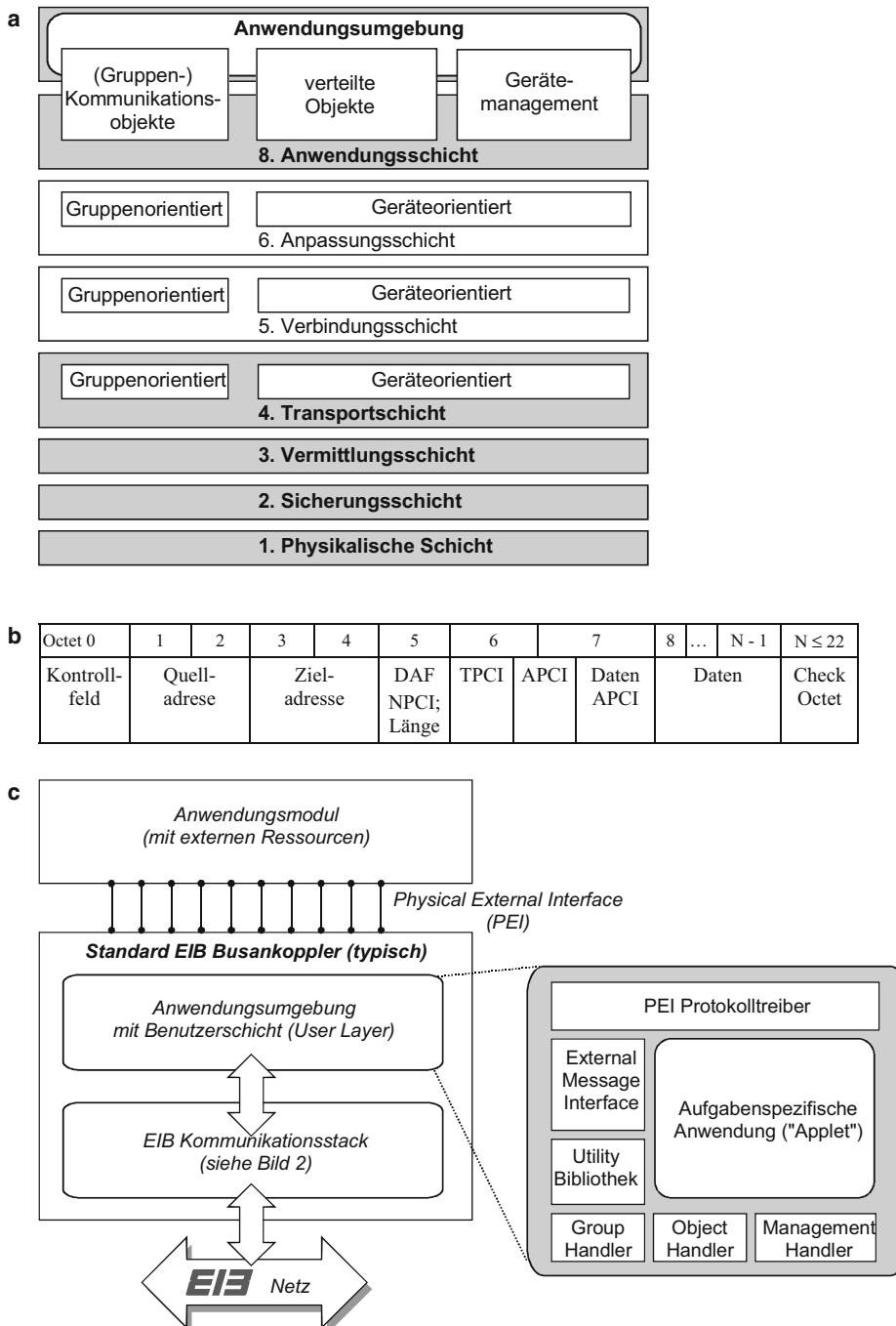


Abb. 4.13 KNX Kommunikation nach dem OSI-Referenzmodell. **a** Übersicht, **b** OSI-Schicht 2: PDU Rahmenstruktur (lange Rahmen lassen $N < 255$ zu), **c** PEI (Physical External Interface), Anwenderschicht und Hostdienste für eine typische KNX BAU

standardmäßig auch die Funktion eines Managementservers. Die KNX PDU-Rahmen (PDU = Protocol Data Unit) können Anwendungs-Datenformate mit einer maximalen Größe von 14 Byte und im „extended frame mode“ von bis zu 254 Byte transportieren.

Im nächsten Kapitel erfahren Sie mehr über die zentrale Bedeutung der dedizierten gruppen-orientierten Funktionen des KNX Betriebssystems.

4.1.2.5 Netzverwaltung und Adressierung

Netzverwaltung

Zur Verwaltung der Netzwerk-Ressourcen (z. B. bei der Konfiguration der Anlage) verwendet KNX eine Kombination aus Broadcast-Kommunikation (Rundsenden) und Punkt-zu-Punkt-Kommunikation. Über die Broadcast-Kommunikation wird jedem Gerät in der Anlage eine eindeutige Physikalische Adresse zugeordnet (wahlweise kann dafür z. B. eine eindeutige Seriennummer des Geräts verwendet werden), die von diesem Zeitpunkt an für die folgenden Punkt-zu-Punkt-Verbindungen maßgeblich ist. Eine Verbindung (wahlweise auch mit Zugriffsberechtigung) kann aufgebaut werden, um zum Beispiel das komplette binäre Abbild („applet“) eines Anwendungsprogramms herunterzuladen.

Ein verbindungsloser Zugriff auf dezentrale KNX-Objekte ist über die <Gerät>. <Objekt>. <Eigenschaft>-Adressierung möglich, die man als den grundlegenden Mechanismus der KNX-Verwaltungsebene für die Statusanzeige und -steuerung bezeichnen kann. Ein dedizierter schneller Abfragemodus (polling), beruhend auf dem Master-Slave-Prinzip, stellt die „Lebendüberwachung“ und Statusüberprüfung von kritischen Untersystemen sicher.

Runtime-Effizienz durch Gruppenadressierung

KNX bietet 100 %ige Unterstützung der Multicast-Adressierung (Gruppen-Adressierung). 100 %ig bedeutet: KNX ist nicht auf das Gruppieren von Geräten beschränkt: Jedes Gerät kann *für sich* mehrere Variablen (bekannt als „(Gruppen-Kommunikationsobjekte“) definieren, die dann unabhängig voneinander zu im gesamten Netz verfügbaren verteilten Variablen gruppiert werden. Ein zusätzlicher Vorteil: Auch die Eigenschaften von dezentralen Objekten können als verteilte Variable zugänglich gemacht werden. Wie bereits für die gruppenorientierte KNX Kommunikationsstapel erläutert wurde, funktioniert der Lese/Schreib-Zugriff auf eine verteilte Variable völlig bidirektional. Auf diese Weise können alle Geräte auch *spontan* Multicast-Telegramme senden. KNX stellt für diese verteilten Variablen einen Adressenraum von 16 Bits zur Verfügung. Das bedeutet, dass auch bei Implementierungen, die auf 15 Bits begrenzt sind, eine Anlage bis zu 32 k verteilte Variable (oder „Gruppenadressen“) haben kann, jede mit einer beliebigen Anzahl von lokalen Instanzen. Das Ergebnis unterstreicht den hohen Netzwirkungsgrad dieser Betriebsart Gruppenadress-Kommunikation, und begründet damit deren Bevorzugung im KNX für den laufenden Betrieb.

Auf diese, vielleicht etwas unerwartete Art und Weise trägt KNX ein ganzes Stück dazu bei, die Notwendigkeit von redundanten Automatisierungs-Hierarchieebenen (und Band-

breite!) anhand von geeigneten Adressierungs- und Gerätemodellen zu reduzieren. Damit wird es möglich bei gleichem Netzwirkungsgrad gegenüber konkurrierenden Systemen mit niedrigerer Bitrate zu arbeiten, womit die bereits sehr hohe Systemzuverlässigkeit weiter erhöht wird. Auch wirkt das sich dabei ergebende einfachere Installationskonzept (Standard-Niederspannungs-Zweidrahtleitung, keine Abschlusswiderstände) noch zusätzlich verstärkend.

Multi-Client-/Multi-Server-Management des OO KNX Netzes

Man kann eine KNX-Anlage als eine Ansammlung von dezentralen Ressourcen bezeichnen, die quer über das Netzwerk verwaltet werden. Zu diesem Zweck übernimmt jedes KNX-Gerät auch die Funktion eines Servers, der die örtlichen Ressourcen steuert einschließlich der Host-Dienste für externe CPU oder Speicherressourcen, auf die durch das serielle PEI (Physical External Interface, (siehe Abb. 4.13c)) zugegriffen wird. Eine Reihe von APCIs (Application Protocol Control Information) machen diese Dienste auch für entfernte Clients zugänglich. Durch die Einführung von KNX verteilten Objekten werden die Netzressourcen praktisch „objektorientiert“ (OO).

Typischerweise greifen Management-Clients zur Ausführung von Steuerungs- oder Konfigurations(-Anlauf)aufgaben auf das Netz zu. Die KNX Association liefert eine komplette Garnitur von herstellerunabhängigen, auf Standard-PCs basierenden Konfigurationswerkzeugen für die Verwaltung von ladbaren Applets. Auch Handgeräte stehen zur Verfügung. Netzwerk-Clients (typischerweise auf einer DIN-Schiene montiert) ermöglichen eine interaktive vereinfachte Selbstkonfiguration (*Easy Configuration*) von (Unter-)Systemen.

4.1.2.6 Datenformate und Interworking

Wie wir bereits gesehen haben, können die heutigen KNX-Telegramme Datenformate bis zu einer Größe von 14 Byte und im „extended frame mode“ bis zu 254 Byte transportieren. Die durch die KNX-Spezifikation festgelegten grundlegenden Datenformate sind derzeit: Boolesches Format (1 Bit), (ohne) Vorzeichen kurz (16 Bits), (ohne) Vorzeichen lang (32 Bits), Gleitkomma kurz (16 Bits), IEEE Gleitkomma (32 Bits), Datum (24 Bits), Zeit (24 Bits), Steuerung (4 Bits) usw.

Für fast alle physikalischen Größen wie Temperatur, Länge, Geschwindigkeit, Feldstärke, Energie, Leistung usw. werden Kennzeichen definiert. Angaben zum Typ werden in der Regel nur bei der Konfigurationserstellung berücksichtigt: sie sind nicht Teil der Übertragung, um einer Leistungsverminderung sowie unnötigen Beschränkungen bei der Kombination von Geräten entgegenzuwirken.

Die Eigenschaften dieser grundlegenden Datentypen werden zu „verteilten Objekten“ gruppiert, die über das Netz zur Verfügung stehen. Die KNX Interworking Standards definieren verschiedene spezialisierte Objekte (DPT) für alle Bereiche der Gebäudeautomation, wie z. B. Beleuchtung (Helligkeitssteuerung, ...), HVAC-Anwendungen (Raumtemperaturregelung, Boilertemperaturregelung, ...), Zeit- und Ereignisverwaltung (Zeitplan-Handler, Ereignis-Handler, ...) sowie Funktionsblöcke mit standardisierten Datenpunkttypen (DPT) und festgelegtem Verhalten.

Host- und Schnittstellenmerkmale des KNX Betriebssystems

Das dezentrale KNX Betriebssystem (BS) bedient nicht nur die entfernt liegenden Netzwerk-Clients, sondern steht mit seinen Diensten natürlich auch den lokalen Client-Applikationen als Server für die Kommunikation und Verwaltung zur Verfügung.

Interne Anwendungen

Den internen Anwendungen stellt die BAU zudem CPU- und Speicherressourcen, Zeitgeber usw. zur Verfügung; man kann sagen, die Anwendung „läuft im BAU“. Fortschrittliche Ausführungen ermöglichen bis zu drei asynchrone Applikationsabläufe (threads).

Dienstbibliothek API

Als Teil der Anwender-Abstraktionsschicht (user abstraction layer) stellt KNX eine standardisierte Dienst- (oderAnwender-) Bibliothek (API) zur Verfügung, die die Applikation mit mehr Infrastruktur versorgt und Funktionen wie Zeitgeber, Entprellung, Arithmetik, Bitlogik, Meldeverarbeitung usw. bereitstellt. Über die API-Schnittstelle kann die Applikation auch auf externe Applikationshardware zugreifen, wie im nächsten Abschnitt beschrieben wird

Hosting von externer Anwendungshardware

Eine weitere einzigartige Dienstleistung des KNX rationalisiert die Hostfähigkeit des Netzes für Anwendungshardware oder externe Betriebsmittel:

- Die standardisierte PEI Anwendungsschnittstelle, Abb. 4.13c.
- Diese Anwendungsschnittstelle definiert sowohl elektromechanische als auch Software-Dienste für die Verbindung eines externen Anwendungsmoduls mit einer BAU. Über einen Typenwiderstand erhält die Anwendung Informationen über die Leistungsfähigkeit des Anwendungsmoduls. Das BAU kann ca. 20 Typen verarbeiten (einschließlich binäre, analoge und serielle E/A) und stellt der Applikation die entsprechenden Dienste zur Verfügung (über die API-Schnittstelle).

Ein zusätzlicher Pluspunkt: Die Kombination API/Anwenderschicht + PEI ermöglicht es, die Kombination „Ladbare Anwendung + Anwendungsmodul“, *so wie sie ist*, auf jedem physikalischen Medium anzuwenden. Besonders für Unterputzgeräte werden die Kosten für zusätzliches Engineering oder nachträgliche Einstellungen in der laufenden Anlagevollständig eliminiert; die Vertriebslogistik vereinfacht sich entsprechend. Obwohl die PEI-Schnittstelle eine Option ist, nutzen sie bestimmte standardisierte BAUs, die „Busankoppler“. Für diese ist die PEI-Schnittstelle notwendig und zertifikationsrelevant.

Meldungsschnittstelle für den Zugriff auf externe Ressourcen

Für die serielle PEI-Schnittstelle definiert KNX eine externe Meldeschnittstelle EMI (External Message Interface). Der EMI-Server ermöglicht lokalen und entfernten Clients den Zugriff auf externe CPUs oder Speicherressourcen.

4.1.2.7 Werkzeugsätze und Software Engineering

KNX bietet ganz bewusst eine bestimmte Methodologie für die Projektierung, d. h. für die Einbindung einer Reihe von einzelnen Geräten in eine funktionsfähige Anlage. Dazu dienen zwei herstellerunabhängige Engineering Tool Software (ETS)-Pakete auf Windows-Oberfläche.

Mit der ETS Entwicklersoftware (Developer's Edition) „übersetzt“ der Gerätehersteller die fernladbaren Applets in eine Reihe von abstrakten Darstellungen, die alle konstruktionstechnischen Details verbergen. Die so entstehende Gerätebeschreibung kann exportiert werden.

Der Projekt-Ingenieur oder Elektroinstallateur importiert dann diese Gerätebeschreibung in die ETS-Projektsoftware. Alle Geräteeigenschaften können an die spezifischen Projektanforderungen angepasst und durch die Zuweisung von Gruppenadressen logisch verknüpft werden. Diese Methode reduziert den Projektierungsaufwand auf ein Minimum.

Die ETS-Software zur KNX-Projektierung baut auf Softwarekomponenten der Windows orientierten PC-Plattformen auf. Wir nennen diese Umgebung *KNX Tool Environment (KTE)*. Dieses API-Paket bildet einen Teil des KNX-Standards. KTE kann für die gewerbliche Nutzung erworben werden.

4.1.2.8 Weitere Systemmerkmale

Entwicklung von Anwendungssoftware für KNX

KNX beschränkt sich nicht auf einen bestimmten Prozessor oder eine bestimmte Prozessorausführung. Je nach Einsatzgebiet kann aus dem großen Angebot von kommerziell erwerblichen Werkzeugen wie Assembler, Kompilierer und Emulatoren ausgewählt werden, angefangen von Shareware-Umgebungen bis hin zu für spezielle Zwecke ausgereifte Umgebungen.

Bestimmte KNX-Systemanbieter bieten die *Integrated Development Environment* an, die eine Entwicklung in ANSI C sowie eine leistungsstarke Fehlerbeseitigung (debugging) ermöglicht und eine spezielle KNX Programmierungs-Infrastruktur zur Verfügung stellt.

Das Entwicklerpaket von ETS seinerseits bietet alle notwendigen Dienste für den reibungslosen Import der Ergebnisse in die KNX Tool Environment.

Systementwicklung mit skalierbarem Zugriff

Der Entwickler eines KNX-Produkts kann durch die Verwendung von standardmäßigen KNX Bus Access Units (BAUs) mit verschiedenen Integrationsgraden (Skalierbarkeit) Zugriff auf das KNX-System erhalten (Abb. 4.14). Als Alternative dazu kann er/sie sich für eine eigene (jedoch kompatible) Lösung auf der Grundlage eines beliebigen Mikroprozessorchips entscheiden.

Der *Busankoppler* (BCU) ist die vollständigste Version eines Buszugriffsgerätes BAU, denn es ist ausgestattet mit Zugriff auf das Medium, mit der KNX OS-Firmware, mit Anwendungs-Hosting-Ressourcen (CPU, RAM, EEPROM, ...) sowie mit PEI- und EMI-Schnittstellen. Der Busankoppler wird in einem kompakten, abgeschirmten und monta-

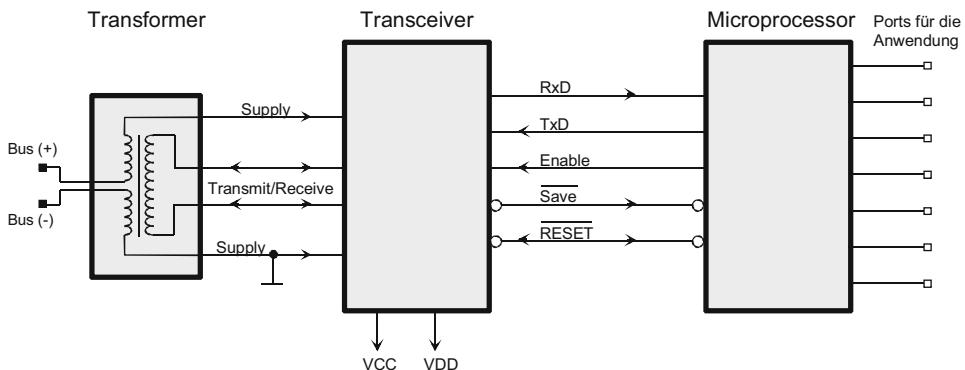


Abb. 4.14 Blockschaltbild einer induktiv gekoppelten Bus Access Unit BAU

gefertigten Gehäuse geliefert. Die Bauformen sind auf die Montageanforderungen in der Praxis (z. B. Montage auf DIN-Schiene oder Unterputzmontage) abgestimmt. Anwendungskompatible Busankoppler sind für sämtliche KNX-Übertragungsmedien erhältlich.

Die KNX *Bus Interface Modules* (BIM) erfüllen dieselben Buszugriffs- und Applikations-Host-Aufgaben (wie im vorherigen Abschnitt erläutert wurde), allerdings nur über die elektrische PEI-Schnittstelle (mit EMI) und ohne EMV-Abschirmung und Gehäuse. Das macht sie ideal für eine engere Einbindung in die anwendungsspezifische Lösung. Ein TP BIM für höhere C-Programmierung mit einem 8–32 kByte EEPROM steht zur Verfügung. BAU-Bausteine sind auch als *Chip-Satz* erhältlich. Um die gewerbliche Nutzung von KNX-Systemen zu fördern, kann der vollständige Quellencode zu nichtdiskriminierenden Bedingungen erworben werden.

Verschlüsselung

KNX IP Secure bietet zusätzliche Sicherheit, indem das KNX IP-Protokoll so erweitert wird, dass die übertragenen Daten vollständig verschlüsselt sind. Dies lässt sich mit kleinem Zusatzaufwand auch in bestehenden Anlagen umsetzen. Wenn Daten nur lokal über KNX gesendet werden, genügt es, die entsprechenden Anwendungsdaten zusätzlich durch eine Erweiterung des Busprotokolls zu schützen. Der spezifizierte Schutzmechanismus KNX Data Secure bewirkt, dass unabhängig vom Medium ausgewählte KNX Telegramme authentifiziert und/oder verschlüsselt werden. Die Schlüssel werden über die ETS den Geräten bzw. Objekten zugeordnet. Da in einem KNX-System gesicherte und ungesicherte Anwendungen möglich sind, müssen nicht alle Geräte gesichert sein. Auch vorhandene Systemkomponenten kann man belassen.

4.1.2.9 Spektrum verfügbarer Produkte

Es wurden bereits im Sommer 1997 von ca. 50 Herstellern 2500 kommerzielle Produktgruppen (jeweils mit mehreren kommerziellen Ausprägungen) angeboten, die ihren Einsatz in den Anwendungsbereichen Heizungsregelung, Energieverwaltung, Sicherheit,

Zeit- und Ereignisverwaltung, Beleuchtungsregelung, Jalousien usw. finden. Aktuell (2017) werden von 400 Herstellern 7000 kommerzielle Produktgruppen angeboten. Obwohl untereinander kompatibel, werden KNX-Ausführungen unter verschiedenen Markennamen wie Instabus, Tebis, i-Bus KNX, Powernet, Home Electronic System, Domotik, ImmoCad usw. vertrieben.

4.2 Feldbusse

4.2.1 Sercos

4.2.1.1 Einleitung

Digital gesteuerte Antriebe bieten für numerisch gesteuerte Maschinen und Anlagen höchste Präzision und Geschwindigkeiten. Man findet sie bei den klassischen CNC-Maschinen (Computerized Numerical Control), wie z. B.:

- Drehmaschinen,
- Freiform-Fräsbearbeitung und HSC (High Speed Cutting),
- Kurbelwellenschleifmaschinen,
- Werkzeugschleifmaschinen,
- Zahnradfeinbearbeitung,
- Textilmaschinen,
- Zeitungsdruckmaschinen,
- Verpackungsmaschinen, usw.

Aber auch in der Automatisierungstechnik finden vernetzte digitale Antriebe Anwendung:

- Montageroboter,
- Handlingsysteme,
- Montagelinien,
- Stapeleinrichtungen,
- Pressenverkettungen, usw.

Zur Steuerung ist eine schnelle, echtzeitfähige, standardisierte digitale Schnittstelle Voraussetzung.

Das Sercos-Bussystem **SErial Real-time COmmunication System**) erfüllt als digitale Antriebsschnittstelle die Summe dieser Anforderungen.

Sercos wurde Mitte der achtziger Jahre von einem Industriekonsortium mit Unterstützung des ZVEI und VDW als digitale Antriebsschnittstelle entwickelt. Die erste Generation von Sercos unterstützte Übertragungsraten von 2 und 4 Mbit/s und kam zunächst vor allem bei anspruchsvollen Werkzeugmaschinenanwendungen zum Einsatz. In den folgenden Jahren fand Sercos weltweit breite Unterstützung und wurde in den unterschiedlichsten Anwendungen erfolgreich eingesetzt. Im Jahre 1995 wurde Sercos als internationale

Norm IEC 61491 anerkannt, 1998 als europäische Norm EN 61491. 1999 folgte die zweite Generation des Standards (Sercos II):

Unter anderem wurde die Übertragungsrate auf 8 und 16 Mbit/s erhöht und der Service-Kanal zur Übertragung asynchroner Daten erweitert. Seit 2001 ist diese Technologie auf der Basis des ASIC-Controllerbausteins SERCON816 verfügbar, wobei eine Abwärts-kompatibilität zur 1. Generation (Sercos I) sichergestellt wurde. Der Einsatz eines Licht-wellenleiterringes stellt darüber hinaus eine sehr hohe Störsicherheit und Robustheit der Übertragung sicher.

Die hardware-basierte kollisionsfreie Datenübertragung auf der Basis eines Zeitschlitz-verfahrens und eines äußerst effizienten Kommunikationsprotokolls ermöglicht eine sehr hohe Performance in Kombination mit einem absolut deterministischen Zeitverhalten. So können beispielsweise bis zu 40 Achsen bei einer Zykluszeit von 1 msec. und einem Jitter von kleiner 1 µs geregelt und hart miteinander synchronisiert werden. Der Einsatz einer optischen Synchronisation ist dabei die Voraussetzung für die sichere Realisierung anspruchsvoller Bewegungsaufgaben, beispielsweise elektronischer Wellen in Zeitungs-druckmaschinen, Verpackungsmaschinen oder mehrachsigen Werkzeugmaschinen.

Obwohl ursprünglich als dedizierte Antriebsschnittstelle konzipiert, hat sich Sercos im Laufe der Jahre zu einer universellen „Motion Control“-Schnittstelle weiterentwickelt. Denn Sercos definiert nicht nur ein echtzeitfähiges Kommunikationssystem, sondern legt über 500 standardisierte Parameter fest, welche das Zusammenspiel von Steuerungen und Antrieben mit einer herstellerübergreifenden Semantik beschreiben. Neben Antrieben kann auch E/A-Peripherie an den Bus gekoppelt werden, so dass bei stand-alone-Maschinen auf einen gesonderten Feldbus verzichtet werden kann.

Die Nutzung der Übertragungsphysik und des Protokolls von Ethernet unter Beibehal-tung der bewährten Sercos Mechanismen hat zur Entwicklung von Sercos III geführt. Mit Sercos III wird eine universelle, echtzeitfähige Vernetzung beliebiger Automatisierungs-komponenten, wie z. B. Servoantriebe, Frequenzumrichter, dezentrale E/A Peripherie, Encodern, Kamerasystem, ermöglicht. Um trotz der Verwendung von Ethernet har-te Echtzeitanforderungen erfüllen zu können, wird bei Sercos III ein kollisionsfreier Echtzeitkanal parallel zu einem optionalen Nicht-Echtzeit-Kanal (UC) geführt. In dem kollisionsfreien Echtzeitkanal werden die von Sercos definierten Telegramme (Ether-type 0x88CD) übertragen. Dieser Kanal zeichnet sich durch eine hohe Protokolleffizienz aus, um auch bei vielen Teilnehmern und jeweils geringen Nutzdaten eine bestmögliche Performance zu erreichen. Parallel zu diesem Echtzeitkanal kann ein Nicht-Echtzeit-Kanal konfiguriert werden, in welchem beliebige Ethernet-Telegramme und IP-basierte Protokolle, wie z. B. TCP/IP und UDP/IP, übertragen werden können.

4.2.1.2 Topologie

Sercos I und II nutzen einen optischen Lichtwellenleiterring, der einen Master mit bis zu 254 Slave-Geräten verbindet (Abb. 4.15). Lichtwellenleiter (LWL) wurden gewählt, um eine hohe Immunität gegen elektromagnetische Störungen zu gewährleisten. Bei der industriellen Anwendung muss mit unvorhersehbaren Störpegeln in Kabelkanälen gerech-

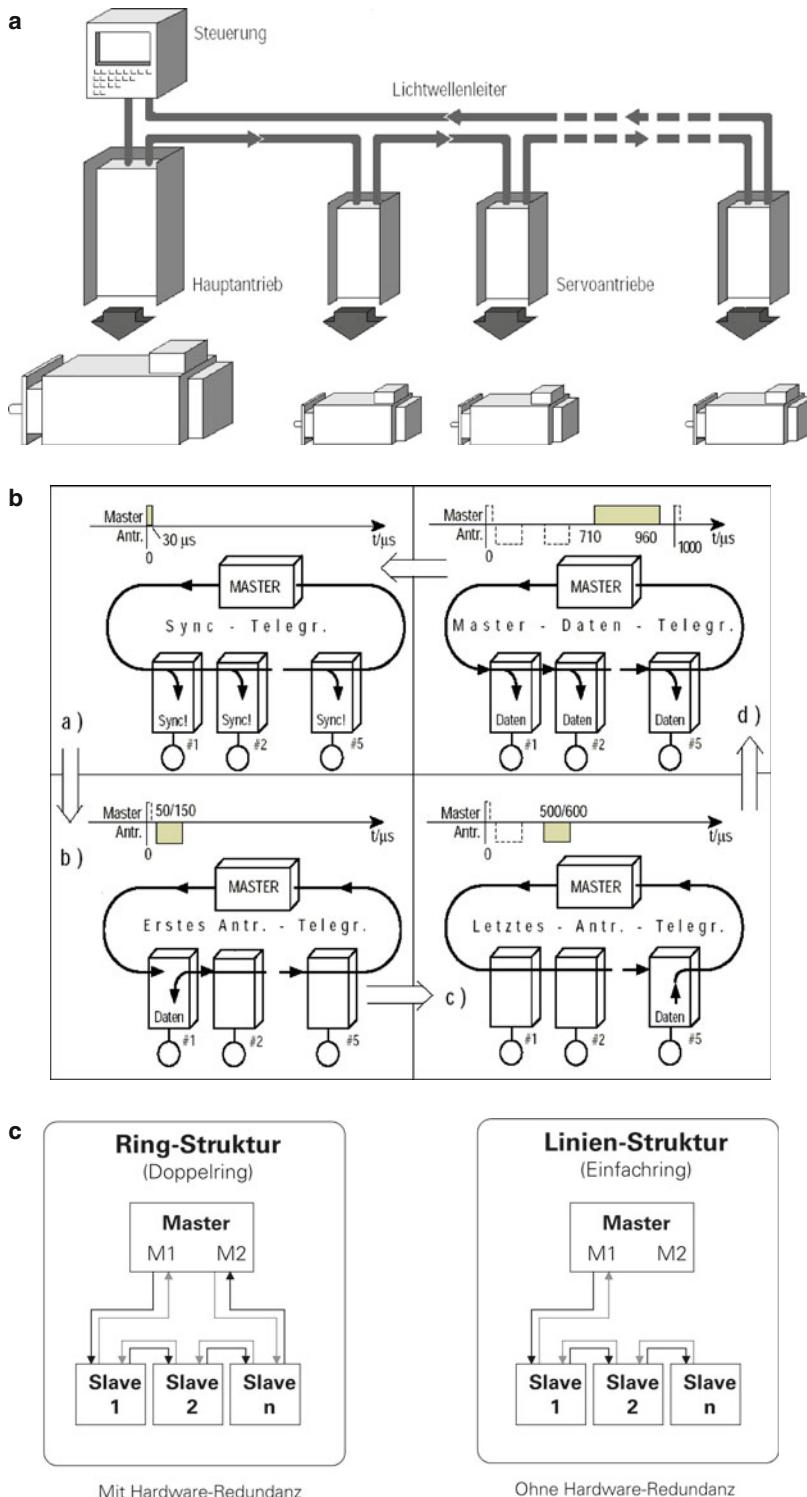


Abb. 4.15 Topologien. **a** Optischer Lichtwellenleiter bei Sercos I/II, **b** Übertragungsverfahren bei Sercos I und II, **a** Master-Sync-Telegramm, **b, c** individuelle Antriebstelegramme, **d** Master-Datentelegramm, **c** Ring- und Linienstruktur bei Sercos III

net werden. Darüber hinaus gilt es, HF-Störungen infolge der Leistungssteuerung durch Pulsweitenmodulation sowie infolge örtlich getrennter Netz- und Erdverbindungen an Steuerungen und Antrieben mit Sicherheit auszuschließen. Die Ringstruktur kommt mit der geringsten Anzahl optischer Komponenten aus und erfordert keine aufwendigen T-Verzweigungen. Die Länge jedes Übertragungsabschnittes kann mit Plastik-LWL bis zu 50 m betragen, mit Glasfaser- LWL bis zu 250 m.

Auch Sercos III unterstützt die bei Sercos I und Sercos II eingesetzte Ringstruktur. Durch die voll-duplex-fähige Ethernet-Physik ergibt sich allerdings nicht ein einfacher Ring, sondern eine Doppelringstruktur (Abb. 4.15c). Die Doppelringstruktur ermöglicht, die Datenübertragung redundant auszuführen. Bei einer Kabelunterbrechung an einer beliebigen Stelle des Rings bleibt die Kommunikationsfähigkeit vollständig erhalten. Das heißt, die Anlage läuft störungsfrei weiter und die integrierte Diagnose meldet eine defekte Kabelverbindung oder eine defekte Station, die ohne Beeinträchtigung der Maschinenverfügbarkeit ersetzt werden kann.

Neben der Ringstruktur wird auch die Linienstruktur zugelassen. Sie hat selbstverständlich nicht den Vorteil der Redundanz, spart allerdings eine Kabelverbindung ein. Das kann bei ausgedehnten Anlagen von Nutzen sein. Sercos III nutzt somit nicht die Sterntopologie des Standard-Ethernet. Hubs oder Switches finden keine Verwendung.

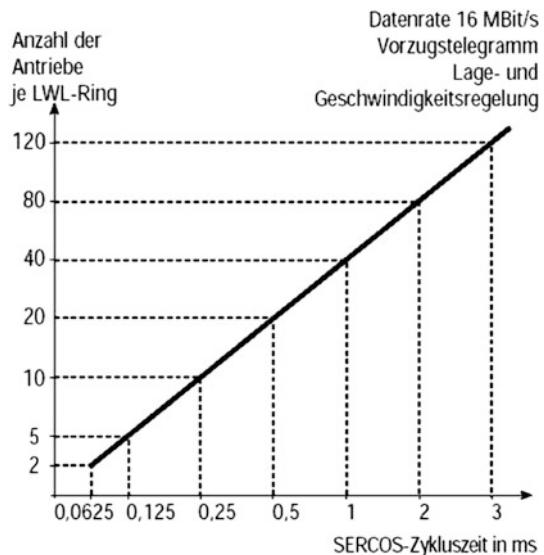
4.2.1.3 Übertragungsverfahren, Synchronisation und Protokollstruktur

Die Kommunikation bei Sercos I und II erfolgt im Betrieb zyklisch als Master-Slave-Kommunikation, mit einer bei der Initialisierung zu wählenden Zykluszeit. Sie kann wahlweise 62 µs, 125 µs, 250 µs oder Vielfache von 250 µs bis zu 65 ms betragen. Die Zykluszeiten sind so spezifiziert, dass die erforderliche Synchronisation mit fixen Arbeitszykluszeiten in Steuerung und Antrieben erzielt wird. Kommunikationsmaster in einem Sercos-Ring ist jeweils die Steuerung. Die Kommunikation erfolgt mit drei Telegrammarten (Bild 4.16): Das Master-Sync-Telegramm (MST) wird von allen Antrieben gleichzeitig empfangen und dient der Synchronisation aller zeitbezogenen Aktionen in der Steuerung und den Antrieben. Das Master-Datentelegramm (MDT) wird ebenso wie das Master-Sync-Telegramm von allen Antrieben gleichzeitig empfangen. Es beinhaltet die zyklischen Daten und die Servicedaten für alle Antriebe am Ring. Die Antriebe senden ihre Telegramme (AT) aufeinanderfolgend in zugeteilten Zeitschlitten.

Die Synchronisation zwischen der zyklisch arbeitenden Steuerung und den ebenfalls zyklisch arbeitenden digitalen Antrieben erfolgt mikrosekundengenau. Da die Istwerte in allen digitalen Antrieben exakt gleichzeitig gemessen werden und alle Sollwerte gleichzeitig wirksam werden, ist eine präzise Koordination der Antriebe gewährleistet. Die Synchronisationsgenauigkeit von unter 1 µs ist unabhängig von der Anzahl der Antriebe, der gewählten Übertragungsgeschwindigkeit und von der konfigurierten Zykluszeit. Bei einer Geschwindigkeit von 60 m/min. entspricht eine Zeitpräzision von 1 Mikrosekunde z. B. einer Maßpräzision von 1 Mikrometer.

Sercos I/II erlaubt Übertragungsraten von 2 und 4 Mbit/s in der 1. Generation, sowie 2, 4, 8 und 16 MBit/s in der 2. Generation. Die exakte Anzahl der max. je LWL-Ring

Abb. 4.16 Der Zusammenhang zwischen Anzahl der Antriebe und der Zykluszeit bei Sercos I, II



bedienbaren Antriebe ist von der erforderlichen Kommunikations-Zykluszeit, dem gewählten Betriebsdatenumfang und der Datenrate abhängig. Abb. 4.16 zeigt die Anzahl der je LWL-Ring anschließbaren Antriebe für eine typische CNC-Anwendung mit den Betriebsarten Geschwindigkeitsregelung und Lageregelung.

Die Anzahl der Antriebe je Steuerung ist durch die Verwendung mehrerer Lichtwellenleiterringe beliebig ausbaubar.

Die Konfigurierbarkeit der Echtzeitdaten erlaubt unabhängig davon die Bedienung beliebiger anderer Betriebsarten. Der Austausch von Servicedaten erfolgt nur auf Anforderung durch den Master. Servicedaten werden mit einer Handshake-Prozedur in 2, 4, 6 oder 8-Byte-Portionen im Service-Datenfeld Info übertragen und beim Empfänger wieder zusammengesetzt. Für die Kommunikation finden NRZI-codierte HDLC-Protokolle Anwendung. Die Echtzeitdaten werden in jedem Kommunikationszyklus komplett im sogenannten konfigurierbaren Datenfeld übertragen (Abb. 4.17). Mit Hilfe des Identnummernsystems kann bei der Initialisierung festgelegt werden, welche Echtzeitdaten übertragen werden. Dies können neben numerischen Daten wie Soll- und Istwerten auch Bitlisten mit I/O-Instruktionen sein. Für die drei grundlegenden Antriebsbetriebsarten – Drehmomentregelung – Drehzahlregelung – Lageregelung und kombinierte Drehzahl/Lageregelung – wurden Vorzugstelegramme mit bestimmten Echtzeitdaten spezifiziert.

Bei Sercos III werden die Antriebs-Telegramme AT außerdem dazu verwendet, Daten direkt zwischen Slave-Geräten auszutauschen (sogenannter direkter Querverkehr, C2C, *Controller to Controller*).

Um die Bandbreite des Fast Ethernet von 100 Mbit/s möglichst effizient zu nutzen, greifen verschiedene Maßnahmen. Zum einen werden die Echtzeittelegramme von mehreren Netzwerkteilnehmern gemeinsam genutzt, wodurch der Overhead signifikant reduziert

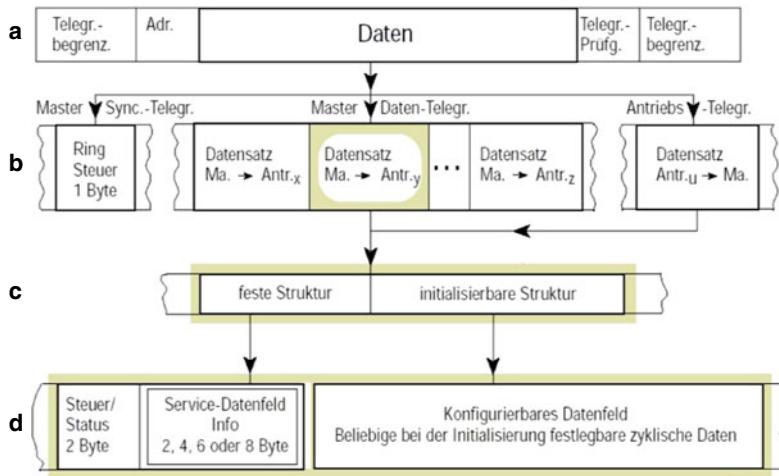


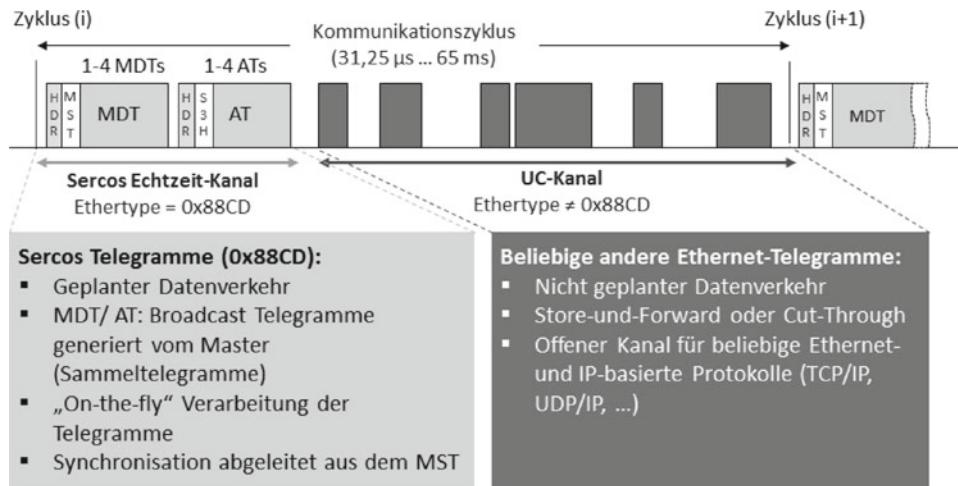
Abb. 4.17 Telegrammaufbau bei Sercos I und II. **a** Gesamtes Telegramm, **b** Service-Datenfeld, **c** Datenstruktur, **d** Datenfeinstruktur

wird. Zum anderen werden – durch eine Verarbeitung dieser Echtzeit-Telegramme während des Durchlaufs („on-the-fly“) und durch den Verzicht auf zusätzliche Netzwerkinfrastrukturkomponenten (wie z. B. Hubs oder Switches) – die Laufzeiten durch das Netzwerk minimiert.

Die Kommunikation bei Sercos III erfolgt – wie bereits bei Sercos I und II – zyklisch. Ein Kommunikationszyklus wird mit Hilfe einer Zeitsteuerung in zwei Kanäle unterteilt (Bild 4.18). Im Echtzeit-Kanal werden die von Sercos III spezifizierten Sammeltelegramme als Broadcast übertragen und im Durchlauf bearbeitet. Im Nicht-Echtzeit-Kanal UC können dagegen Nicht-Echtzeit-Ethernet Frames als Einzeltelegramme an beliebige Geräte des Netzwerkverbundes gesendet werden. Die Kommunikationszyklen und die Aufteilung der Bandbreite von 100 Mbit/s in Echtzeit- und Nicht-Echtzeit-Kanal lassen sich an den jeweiligen Anwendungsfall anpassen.

Normierte Daten

Das funktionelle Zusammenspiel von Steuerungen und Antrieben ist nur gewährleistet, wenn über die eigentliche Kommunikation hinaus die für den Betrieb zwischen der Steuerung und den Antrieben auszutauschenden Daten vereinbart sind. Um das Zusammenwirken von Produkten unterschiedlicher Hersteller zu gewährleisten, wurden bei Sercos über 500 Daten und die Funktionsabläufe von Kommandos spezifiziert. Neben den im Betrieb auszutauschenden Echtzeittypen sind insbesondere folgende Parameter spezifiziert: Einstellung der Kommunikation, Auswahl von Betriebsarten, Adaption unterschiedlichster Maschinenmechanik, Adaption unterschiedlicher Messgeber und Anordnung, Anpassung der Soll- und Istwerte an die Steuerung.

a

Alle Telegramme sind als Standard Ethernet Frames ausgeführt

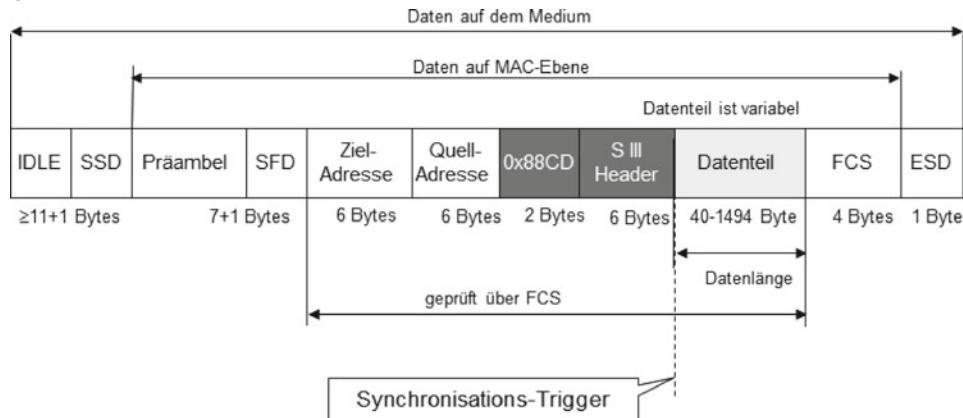
b

Abb. 4.18 Sercos III. **a** Übertragungsprotokoll. Zusätzlich zum Sercos-Telegramm kann ein normales Ethernet-Telegramm (UC) übertragen werden (MST- Synchronisierungssignal), **b** Sercos-Ethernet-Telegramm

Die Adressierung von Daten beim bedarfsgesteuerten Datenaustausch und die Definition der Echtzeitdaten erfolgt bei Sercos mittels sogenannter Identnummern, die als 16 Bit Wert dargestellt werden. Für ID-Nummern steht somit ein Zahlenspektrum von 2^{16} zur Verfügung, wobei ein Bit dazu genutzt wird, zwischen Standardnummern (Bit 15=0) und produkt- bzw. herstellerspezifischen Nummern (Bit 15=1) zu unterscheiden. Die ID-Nummern 1 bis 32767 sind somit für Daten reserviert, die standardisiert sind und die der

Technische Arbeitskreis von Sercos festlegt. Die ID-Nummern 32768 bis 65535 stehen den Produktherstellern zur Verfügung, um Daten oder Parameter zu definieren, die nicht vom Standard abgedeckt sind, jedoch für den Betrieb eines Produktes benötigt werden. Unter der ID-Nummer jedes Datums ist im Antrieb jeweils ein kompletter Datenblock abgelegt. Der Datenblock ermöglicht die Anzeige, Eingabe und Sicherung aller in den Antrieben adressierbaren Daten, Parameter und Diagnosen über die Steuerung mittels eines standardisierten Softwaretreibers.

4.2.1.4 Implementierung

Für eine einfache und kostengünstige Nutzung der Sercos I- und Sercos II-Schnittstellen hat die Sercos-Nutzerorganisation einen Kommunikationscontroller-Baustein, den SERCON816 ASIC, entwickeln lassen. Der SERCON816 erledigt als intelligenter Schnittstellen-Baustein die komplette Kommunikation zwischen LWL-Ring und einem Mikroprozessor. Auf der Basis des SERCON816 werden Entwicklungstools und Anwendungsunterstützung von verschiedenen Dienstleistern angeboten.

Bei Sercos III kommt eine flexiblere Hardware-Plattform zum Einsatz. Basis dafür ist die Entwicklung und Bereitstellung eines Sercos Cores (Sercos III IP), der Hersteller von Komponenten und Systemen in die Lage versetzt, die erforderliche Sercos III Hardware in ein FPGA (field programmable gate array) zu integrieren. Aktuell werden Bausteine der Firmen Xilinx, Intel (ehemals) Altera und Lattice unterstützt.

Optional kann ein Sercos III Master auch ohne Verwendung von Spezialhardware realisiert werden. Anstelle der spezifischen Hardware kommen dann Standard-Ethernet Controller zum Einsatz. Hardwarefunktionen, die Sercos-spezifisch sind, werden dabei in den hardwarenahen und echtzeitfähigen Teil des Mastertreibers verlagert, so dass die masterseitige Anschaltung komplett in Software realisiert werden kann. Diese Master-Realisierung ist beispielsweise für PC-basierte Steuerungsplattformen interessant, die eine Ethernet-Schnittstelle „onboard“ haben, so dass eine zusätzliche Hardware entfallen kann (SoftMaster). Mit der neuen Generation von Ethernet-Controllern gemäß IEEE TSN (Time-Sensitive Networking) wird eine zeitgesteuerte Übertragung von Telegrammen ermöglicht. Damit erreicht ein SoftMaster eine zu einem HardMaster vergleichbare Echtzeit- und Übertragungsperformance.

CIP Safety on Sercos

Das Sicherheitskonzept ist eine zu Sercos III kompatible Protokollerweiterung, um Sercos III auch in Sicherheitsapplikationen bis SIL3 nach IEC 61508, selbst bei kleinsten Zykluszeiten einsetzen zu können. Die sicherheitsrelevanten Daten werden dabei gemeinsam mit den Echtzeitdaten und anderen Standard Ethernet Protokollen über ein einziges physikalisches Medium übertragen. Dadurch entfällt ein zusätzlicher Sicherheitsbus, die Handhabung wird vereinfacht und die Hardware- und Installationskosten werden reduziert.

4.2.2 PROFIBUS

Nachdem in den 80er Jahren des 20. Jahrhunderts eine große Zahl firmenspezifischer Feldbusssysteme entstanden war, hat ein Arbeitskreis „Feldbus“ des ZVEI Anforderungen an Feldbusssysteme in unterschiedlichen Anwendungsbereichen erarbeitet, um eine genormte digitale Feldgeräteschnittstelle festzuschreiben. Anfang 1987 haben sich maßgebende Mitgliedsfirmen des ZVEI darauf verständigt, das Konzept PROFIBUS (PROcess FIeld BUS) zur Grundlage eines Feldbusstandards zu machen. 14 Firmen und 5 Institute haben sich dann in einem öffentlich geförderten Verbundprojekt zusammengetan, um eine Feldbusnorm zu erarbeiten, den Aufbau von Testsystemen zur Verifikation der Implementierung zu definieren und in Pilotinstallationen das System zu erproben und demonstrieren.

Heute ist PROFIBUS das weltweit führende durchgängige, offene, digitale Kommunikationssystem. Es wird vor allem in der Fertigungs- und Prozessautomatisierung, aber auch in der Verkehrstechnik sowie der Energieerzeugung und -verteilung eingesetzt. Die PROFIBUS Nutzerorganisation (PNO) und ihr internationaler Dachverband PROFIBUS & PROFINET International (PI) sorgen für die Pflege, Weiterentwicklung und Qualitäts-sicherung von PROFIBUS als auch von PROFINET, dem industriellen Kommunikations-system auf Basis von Ethernet. Dem Markt stehen weit über 2000 verschiedene Geräte mit PROFIBUS Schnittstelle zur Verfügung. Gegen Ende 2011 wurde die Zahl von über 40 Millionen weltweit installierter PROFIBUS-Knoten überschritten.

PROFIBUS ist für schnelle, zeitkritische und für komplexe Kommunikationsaufgaben geeignet. Die Kommunikation von PROFIBUS ist in den internationalen Normen IEC 61158 und IEC 61784 verankert. Die Anwendungs- und Engineeringaspekte sind in Richt-linien von PI festgelegt. Damit werden die Anwenderforderungen nach Herstellerunab-hängigkeit und Offenheit erfüllt und die Kommunikation zwischen Geräten verschiedener Hersteller ohne Anpassungen an den Geräten garantiert.

4.2.2.1 PROFIBUS als „System-Baukasten“

PROFIBUS ist durch die Bereitstellung einer skalierbaren Kommunikationstechnologie mit verschiedenen Leistungsklassen, zahlreichen Applikations- und Systemprofilen sowie Tools zum Geräte-Management nach dem Baukastenprinzip angelegt. PROFIBUS deckt damit die vielfältigen und anwendungsspezifischen Anforderungen aus Fertigungs- und Prozessautomatisierung gleichermaßen ab.

Aus technologischer Sicht orientiert sich der Systemaufbau von PROFIBUS in seinem unteren Bereich (Kommunikation) an dem ISO/OSI Referenzmodell. Die Schichten 1, 2 und 7 des OSI-Modells sind bei PROFIBUS spezifiziert. Oberhalb der Schicht 7 sind mit den Applikationsprofilen Festlegungen zwischen Herstellern und Anwendern über spezi-fische Geräteanwendungen angeordnet. Übergreifend über mehrere Schichten enthält der PROFIBUS „Systembaukasten“ Funktionen und Tools zur Gerätebeschreibung und Gerä-teintegration (Abb. 4.19).

Aus Anwendersicht stellt sich PROFIBUS in Form von verschiedenen typischen Schwerpunkten dar, die sich aus häufigen Anwendungen als sinnvoll ergeben haben.

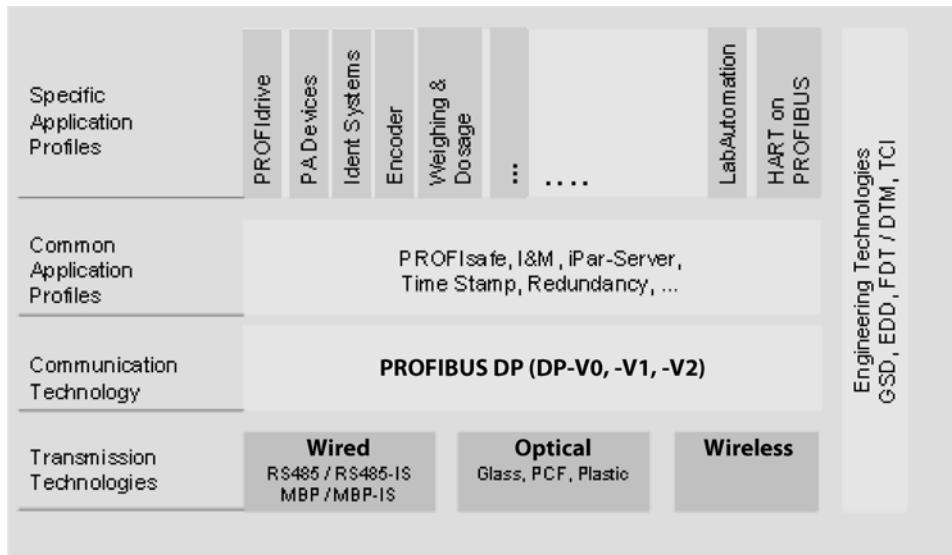


Abb. 4.19 Der PROFIBUS „Systembaukasten“

Jeder Schwerpunkt entsteht durch eine typische (aber nicht zwangsläufig festgelegte) Kombination von Baukastenelementen der Gruppen „Übertragungstechnik“ (Transmission Technologies), „Kommunikationsprotokoll“ (Communication Technologies) und „Applikationsprofile“ (Common and Specific Application Profiles).

PROFIBUS DP bezeichnet im Sprachgebrauch die Variante für die Fertigungsautomatisierung; sie nutzt im Allgemeinen RS485 als Übertragungstechnik und das (wortgleiche) PROFIBUS DP-Kommunikationsprotokoll in einer seiner Leistungsstufen. Oft werden eines oder mehrere der für die Fertigungsautomatisierung typischen Applikationsprofilen benutzt, z. B. Ident Systems oder Robots/NC.

PROFIBUS PA ist die Variante für die Prozessautomatisierung mit typischerweise der MBP-IS-Übertragungstechnik (Manchester Coded, Bus Powered, Intrinsically Safe), der DP-V1 Leistungsstufe des Kommunikationsprotokolls und dem Applikationsprofil PA Devices.

Für schnelle Aufgaben in der Antriebstechnik wird die Leistungsstufe DP-V2 des Kommunikationsprotokolls mit dem Applikationsprofil PROFIdrive eingesetzt.

PROFIsafe ist die Variante für sicherheitsrelevante Anwendungen mit RS485 oder MBP-IS als Übertragungstechnik, einer der möglichen Leistungsstufen von DP.

Auf der Protokollebene bietet PROFIBUS heute mit DP in seinen Versionen DP-V0 bis DP-V2 ein breites Spektrum von Möglichkeiten an, mit dem unterschiedliche Anwendungen optimal kommunizieren können. DP (Decentralized Peripherals) steht für einen einfachen, schnellen, zyklischen und deterministischen Prozessdatenaustausch zwischen

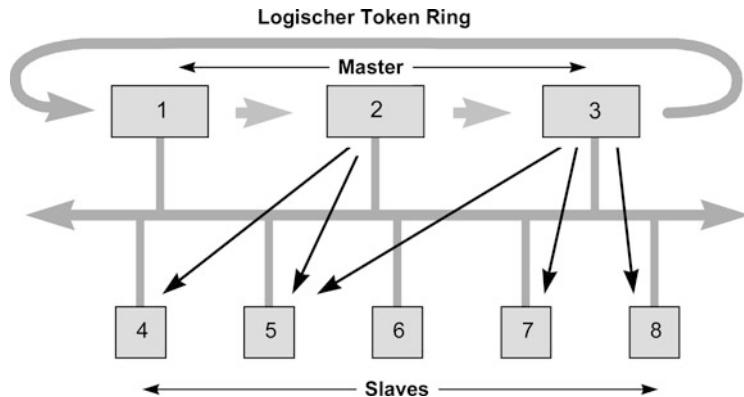


Abb. 4.20 Hybrides Buszugriffverfahren

einem Busmaster und den zugeordneten Slave-Geräten. Diese mit DP-V0 bezeichnete Funktionsstufe wurde um einen azyklischen Datenaustausch zwischen Master und Slave in der Stufe DP-V1 erweitert. In der Stufe DP-V2 ist ein direkter Datenaustausch zwischen Slaves und deren isochronen Betrieb möglich.

Als Buszugriffsverfahren (Schicht 2, Data Link Layer) kennt PROFIBUS das Master-Slave-Verfahren, ergänzt um das Token-Verfahren zur Koordination mehrerer Master am Bus (Abb. 4.20).

Zu den Aufgaben von Layer 2 im OSI-Modell gehören auch Funktionen wie Datensicherung und Abwicklung der Telegramme. Die Schicht 7 definiert die Anwendungsschicht (Application Layer) und bildet die Schnittstelle zum Anwendungsprogramm. Sie bietet unterschiedliche Dienste für den zyklischen und azyklischen Datenaustausch an.

Profile sind von Herstellern und Anwendern getroffene Festlegungen (Spezifikationen) über bestimmte Eigenschaften, Leistungsmerkmale und Verhaltensweisen von Geräten und Systemen. Profilspezifikationen haben das Ziel, Geräte und Systeme, die auf Grund einer „profilgemäßen“ Entwicklung zu einer Profilfamilie gehören, an einem Bus interoperabel und bis zu einem gewissen Grad austauschbar betreiben zu können. Profile berücksichtigen anwendungs- und typspezifische Besonderheiten von Feldgeräten, Steuerungen und Integrationsmitteln (Engineering). Der Profilbegriff erstreckt sich von wenigen Festlegungen für eine bestimmte Gerätekategorie bis hin zu umfassenden Festlegungen für Anwendungen in einer bestimmten Branche. Als übergeordnete Bezeichnung wird der Begriff „Applikationsprofile“ gebraucht.

PROFIBUS unterscheidet zwischen allgemeinen Applikationsprofilen mit Einsatzmöglichkeit bei unterschiedlichen Anwendungen (hierzu gehören beispielsweise die Profile PROFIsafe, Redundanz und Time Stamp) und spezifischen Applikationsprofilen, die jeweils nur für eine ganz bestimmte Art der Anwendung entwickelt wurden, wie z. B. PROFIdrive, SEMI oder PA Devices.

Tab. 4.1 Übertragungstechniken

>	RS485	RS485-IS	MBP	MBP-IS	Fiberoptik
Datenübertragung	Differenzialsignale nach RS485, NRZ	Differenzialsignale nach RS485, NRZ	bitsynchron, Manchester-codierung	bitsynchron, Manchester-codierung	optisch, digital, NRZ
Übertragungsrate	9,6–1200 kbit/s	9,6–1200 kbit/s	31,25 kbit/s	31,25 kbit/s	9,6–1200 kbit/s
Daten-sicherung	HD = 4, Paritybit, Start- und Enddelimiter	HD = 4, Paritybit, Start- und Enddelimiter	Präambel, fehlergesicherte Start- u. Enddelimiter	Präambel, fehlergesicherte Start- u. Enddelimiter	HD = 4, Paritybit, Start- und Enddelimiter
Kabel	verdrillte, geschirmte Zweidrahtleitung Typ A	verdrillte, geschirmte Zweidrahtleitung Typ A	verdrillte, geschirmte Zweidrahtleitung Typ A	verdrillte, geschirmte Zweidrahtleitung Typ A	Multi- und Singlemode Glas- und polymeroptische Fasern
Fernspeisung	über zusätzliche Adern möglich	über zusätzliche Adern möglich	über die Signaladern	über die Signaladern	über Hybrideleitung möglich
Zündschutzart	keine	Eigensicherheit Ex ib	keine	Eigensicherheit Ex ia/ib	keine
Topologie	Linie mit Terminierung	Linie mit Terminierung	Linie und Baum mit Terminierung	Linie und Baum mit Terminierung	typisch Stern und Ring, Linie möglich
Teilnehmerzahl	max. 32 je Segment, max. 126 je Netz	max. 32 je Segment, max. 126 je Netz	max. 32 je Segment, max. 126 je Netz	max. 32 je Segment, max. 126 je Netz	max. 126 je Netz
Anzahl der Repeater	max. 9 mit Signal-auffrischung	max. 9 mit Signal-auffrischung	max. 4 mit Signal-auffrischung	max. 4 mit Signal-auffrischung	mit Signal-auffrischung unbegrenzt

4.2.2.2 Übertragungstechnik

Im ISO/OSI-Schichtenmodell übernimmt die Schicht 1 (Physical Layer) die Festlegung, in welcher Weise die Datenübertragung „physikalisch“, d. h. elektrisch und mechanisch, erfolgt. PROFIBUS stellt verschiedene Ausprägungen der Schicht 1 als Übertragungstechnik zur Verfügung (Tab. 4.1). Alle Ausprägungen beruhen auf internationalen Standards und sind PROFIBUS in der IEC 61158 und IEC 61784 zugeordnet.

4.2.2.2.1 RS 485

Die einfache und kostengünstige Übertragungstechnik RS485 wird bevorzugt für Aufgaben verwendet, die eine hohe Übertragungsrate erfordern. Es wird ein verdrilltes, geschirmtes Kupferkabel mit einem Leiterpaar verwendet. Die Busstruktur erlaubt das rückwirkungsfreie Ein- und Auskoppeln von Stationen oder die schrittweise Inbetriebnahme

des Systems. Spätere Erweiterungen haben innerhalb definierter Grenzen keinen Einfluss auf Stationen, die in Betrieb sind. Mit der Variante RS485-IS (Intrinsically Safe) ist es unter Einhaltung bestimmter Werte auch möglich, in den eigensicheren Bereich zu gehen.

Die Übertragungsrate ist bei RS 485 im Bereich zwischen 9,6 kbit/s und 12 Mbit/s wählbar. Sie wird bei der Inbetriebnahme des Systems einheitlich für alle Geräte am Bus festgelegt. Die maximal zulässige Leitungslänge ist abhängig von der Übertragungsrate. Alle Geräte werden in einer Busstruktur (Linie) angeschlossen. In einem Segment können bis zu 32 Teilnehmer (Master oder Slaves) zusammengeschaltet werden. Anfang und Ende eines jeden Segments wird mit einem aktiven Busabschluss (bus termination) versehen, wobei für einen störungsfreien Betrieb sichergestellt werden muss, dass beide Busabschlüsse ständig mit Spannung versorgt werden. Der Busabschluss ist üblicherweise in den Geräten bzw. den Busanschlusssteckern zuschaltbar realisiert. Bei mehr als 32 Stationen oder zur Erweiterung der Netzausdehnung müssen Leistungsverstärker (Repeater) eingesetzt werden, welche die einzelnen Bussegmente verbinden.

Zur Verbindung der Geräte untereinander sowie mit Netzwerk-Elementen sind am Markt verschiedene Kabeltypen (Typebezeichnung A bis D) für unterschiedliche Einsatzfälle erhältlich.

4.2.2.2 MBP

Die MBP Übertragungstechnik (Manchester Coding with Bus Powering) mit einer festgelegten Baudrate von 31,25 kbit/s kommt in der Prozessautomatisierung zum Einsatz. Sie erfüllt die Anforderungen der Verfahrenstechnik: Eigensicherheit und Busspeisung in Zweileitertechnik.

4.2.2.3 MPB-IS

Die Variante -IS (Intrinsically Safe, eigensicher) ist der Feldbus für Ex-Bereiche. Möglichkeiten und Grenzen von eigensicheren Feldbussen mit MBP-Übertragungstechnik für den Einsatz in explosionsgefährdeten Bereichen sind durch das FISCO-Modell (Fieldbus Intrinsically Safe COncept) festgelegt. Das FISCO-Modell wurde in Deutschland von der Physikalisch Technischen Bundesanstalt (PTB) entwickelt und wird heute international als Basismodell für den Betrieb von Feldbussen in Ex-Bereichen anerkannt. MBP-Übertragung und FISCO-Modell richten sich nach folgenden Grundsätzen:

- Beim Senden eines Teilnehmers wird keine Leistung in den Bus eingespeist.
- In jedem Segment gibt es nur eine einspeisende Quelle, das Speisegerät.
- Jedes Feldgerät nimmt zu seiner Energieversorgung im eingeschwungenen Zustand einen konstanten Grundstrom von mindestens 10 mA auf.
- Die Kommunikationssignale werden vom sendenden Gerät durch Aufmodulieren von ± 9 mA auf den Grundstrom erzeugt.
- Die Feldgeräte wirken als passive Stromsenke.
- Der passive Leitungsabschluss erfolgt an beiden Enden der Bushauptleitung.
- Es sind Netze in Linien-, Baum- und Sterntopologie möglich.

Für den Betrieb eines Feldbusnetzes in Ex-Bereichen ist es erforderlich, dass alle in den Ex-Bereichen verwendeten Komponenten gemäß FISCO-Modell und MBP durch autorisierte Zulassungsstellen wie PTB oder BVS (Deutschland), UL oder FM (USA) zugelassen und zertifiziert wurden. Sind alle verwendeten Komponenten zertifiziert und werden die nachfolgend aufgezeigten Regeln für die Auswahl des Speisegeräts, der Leitungslänge und der Busabschlüsse beachtet, dann ist für die Inbetriebsetzung des Feldbusnetzes keine Systembescheinigung mehr erforderlich.

4.2.2.4 Fiber Optics

Es gibt Feldbus-Einsatzbedingungen, bei denen eine drahtgebundene Übertragungstechnik ihre Grenzen hat, beispielsweise bei stark störbehafteter Umgebung oder bei der Überbrückung besonders großer Entfernung. In diesen Fällen steht die optische Übertragung mittels Lichtwellen-Leitern (LWL) zur Verfügung. Die Realisierung eines LWL-Netzes erfolgt im einfachsten Fall durch Verwendung von elektrisch/optischen Wandlern, die über eine RS485-Schnittstelle mit dem Gerät und andererseits mit dem LWL verbunden sind. Damit besteht auch die Möglichkeit, innerhalb einer Anlage je nach Gegebenheiten zwischen RS485 und LWL-Übertragung zu wechseln. Aufgrund der Übertragungseigenschaften sind Stern und Ring typische Topologiestrukturen; aber auch Liniensstrukturen sind möglich.

4.2.2.3 Kommunikation

Das Kommunikationsprotokoll DP (Decentralized Peripherals) ist für den schnellen Datenaustausch in der Feldebene konzipiert. Hier kommunizieren zentrale Automatisierungsgesäte wie SPS, PC oder Prozessleitsysteme über eine schnelle serielle Verbindung mit dezentralen Feldgeräten wie E/A, Antrieben, Ventilen, Messumformern oder Analysegeräten. Der Datenaustausch mit den dezentralen Geräten erfolgt vorwiegend zyklisch. Die dafür benötigten Kommunikationsfunktionen sind durch die DP-Grundfunktionen festgelegt. Ausgerichtet an den speziellen Anforderungen der unterschiedlichen Einsatzgebiete wurde DP über diese Grundfunktionen hinaus stufenweise um spezielle Funktionen erweitert, sodass DP heute in drei Leistungsstufen DP-V0, DP-V1 und DP-V2 vorliegt. Die Leistungsstufen DP-V0 und DP-V1 enthalten sowohl „Eigenschaften“ (diese sind verbindlich für eine Realisierung) als auch Optionen, während in Stufe DP-V2 nur Optionen spezifiziert sind.

DP-V0 stellt die Grundfunktionalitäten von DP zur Verfügung. Dazu gehören der zyklische Datenaustausch sowie die stations-, modul- und kanalspezifische Diagnose.

DP-V1 enthält Ergänzungen, vor allem den azyklischen Datenverkehr für Parametrierung, Bedienung, Beobachtung und Alarmbehandlung intelligenter Feldgeräte, parallel zum zyklischen Nutzdatenverkehr. Das erlaubt den Online-Zugriff auf Busteilnehmer über Engineering Tools. Weiterhin enthält DP-V1 Alarne. Dazu gehören unter anderem der Statusalarm, Update-Alarm und ein herstellerspezifischer Alarm.

DP-V2 enthält weitere Ergänzungen und ist vorrangig auf die Anforderungen der Antriebstechnik ausgerichtet. Durch zusätzliche Funktionalitäten wie isochroner Slavebetrieb und Slave-Querverkehr (Data Exchange Broadcast, DXB) kann DP-V2 als Antriebsbus zur Steuerung schneller Bewegungsabläufe in Antriebsachsen eingesetzt werden.

4.2.2.3.1 Leistungsstufe DP-V0 (Tab. 4.2)

Die zentrale Steuerung (Master) liest zyklisch die Eingangsinformationen von den Slaves und schreibt die Ausgangsinformationen zyklisch an die Slaves. Hierbei sollte die Buszykluszeit kürzer sein als die Programmzykluszeit des zentralen Automatisierungssystems, die in vielen Anwendungen etwa 10 ms beträgt. Ein hoher Datendurchsatz alleine genügt allerdings nicht für den erfolgreichen Einsatz eines Bussystems. Vielmehr müssen einfache Handhabung, gute Diagnosemöglichkeiten und eine störsichere Übertragungstechnik gewährleistet sein. Bei DP-V0 sind diese Eigenschaften optimal kombiniert.

Die umfangreichen Diagnosefunktionen von DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst. Die Gerätebezogene Diagnose gibt Meldungen zur allgemeinen Betriebsbereitschaft eines Teilnehmers wie z. B. „Übertemperatur“, „Unterspannung“ oder „Schnittstelle unklar“. Die kennungs-(modul)-bezogene Diagnose zeigt an, ob innerhalb eines bestimmten E/A-Teilbereichs (z. B. 8 Bit eines Ausgangsmoduls) eines Teilnehmers eine Diagnose ansteht. In der kanalbezogenen Diagnose werden schließlich die Fehlerursachen bezogen auf ein einzelnes Ein- oder Ausgangsbit (Kanal) angegeben, wie z. B. „Kurzschluss auf Ausgang“.

Mit DP können Mono- oder Multi-Master Systeme realisiert werden. Dadurch wird ein hohes Maß an Flexibilität bei der Systemkonfiguration ermöglicht. Es können maximal 126 Geräte (Master oder Slaves) an einem Bus angeschlossen werden. Die Festlegungen zur Systemkonfiguration beinhalten die Anzahl der Stationen, die Zuordnung der Stationsadresse zu den E/A-Adressen, die Datenkonsistenz der E/A-Daten, das Format der Diagnosemeldungen und die verwendeten Busparameter.

Jedes DP System besteht aus unterschiedlichen Gerätetypen, wobei drei Arten unterschieden werden. Die Master Klasse 1 (DPM1) sind zentrale Bussteuerungen, die in einem festgelegten Nachrichtenzyklus Informationen mit den dezentralen Stationen (Slaves) zyklisch austauschen. Typische DPM1-Geräte sind z. B. speicherprogrammierbare Steuerungen (SPS) oder PC. Ein DPM1 verfügt über einen aktiven Buszugriff, mit welchem er zu festen Zeitpunkten die Messdaten (Eingänge) der Feldgeräte lesen und die Sollwerte (Ausgänge) der Aktoren schreiben kann. Dieser sich ständig wiederholende Zyklus ist die Grundlage der Automatisierungsfunktion.

Master Klasse 2 (DPM2) sind Engineering-, Projektierungs- oder Bediengeräte. Sie werden bei der Inbetriebnahme und zur Wartung und Diagnose eingesetzt, um die angeschlossenen Geräte zu konfigurieren, Messwerte und Parameter auszuwerten sowie den Gerätezustand abzufragen. Ein DPM2 muss nicht permanent am Bussystem angeschlossen sein.

Tab. 4.2 Eigenschaften der Kommunikation mit DP-V0

Buszugriff	<ul style="list-style-type: none"> Token-Passing-Verfahren zwischen Mästern und Master-Slave-Verfahren zwischen Master und Slaves Mono-Master oder Multi-Master Systeme möglich Master und Slave Geräte, max. 126 Teilnehmer an einem Bus
Kommunikation	<ul style="list-style-type: none"> Punkt-zu-Punkt (Nutzdatenverkehr) oder Multicast (Steuerkommandos) Zyklischer Master-Slave Nutzdatenverkehr
Betriebszustände	<ul style="list-style-type: none"> Operate: Zyklische Übertragung von Eingangs- und Ausgangsdaten Clear: Eingänge werden gelesen, Ausgänge bleiben im sicheren Zustand Stopp: Diagnose und Parametrierung, keine Nutzdatenübertragung
Synchronisation	<ul style="list-style-type: none"> Steuerkommandos ermöglichen die Synchronisation der Ein- und Ausgänge Sync-Mode: Ausgänge werden synchronisiert Freeze-Mode: Eingänge werden synchronisiert
Funktionalität	<ul style="list-style-type: none"> Zyklischer Nutzdatentransfer zwischen DP-Master und Slave(s) Prüfen der Konfiguration und dynamisches Aktivieren oder Deaktivieren einzelner Slaves Leistungsfähige Diagnosefunktionen, 3 abgestufte Diagnose-Meldungsebenen Synchronisation der Eingänge und/oder der Ausgänge Optional Adressvergabe für die Slaves über den Bus Maximal 244 Byte Eingangs-/Ausgangsdaten je Slave
Schutzfunktionen	<ul style="list-style-type: none"> Nachrichtenübertragung mit Hamming Distanz HD=4 Ansprechüberwachung beim DP-Slave erkennt Ausfall des zugeordneten Masters Zugriffsschutz für Ausgänge der Slaves Überwachung des Nutzdatenverkehrs mit einstellbarem Überwachungs-Timer beim Master
Gerätetypen	<ul style="list-style-type: none"> DP-Master Klasse 1 (DPM1) z. B. zentrale Automatisierungsgeräte wie SPS, PC DP-Master Klasse 2 (DPM2) z. B. Engineering oder Diagnosetool DP-Slave z. B. Geräte mit binären oder Analogen Eingängen/Ausgängen, Antriebe, Ventile

Ein Slave ist ein Peripheriegerät (z. B. E/A, Antrieb, HMI (Mensch/Maschinen-Interface), Ventil, Messumformer, Analysengerät), welches Prozessinformationen einliest und/oder Ausgangsinformationen zum Eingriff in den Prozess nutzt. Es sind auch Geräte möglich, die nur Eingangs- oder nur Ausgangsinformationen bereitstellen. Slaves sind in Bezug auf die Kommunikation passive Geräte, sie antworten nur auf eine direkte Anfrage. Dieses Verhalten ist einfach und kostengünstig (bei DP-V0 sogar komplett in Hardware) realisierbar.

Bei Mono-Master-Systemen ist in der Betriebsphase des Bussystems nur ein Master am Bus aktiv. Im Multi-Master-Betrieb befinden sich an einem Bus mehrere Master. Sie bilden entweder voneinander unabhängige Subsysteme, bestehend aus je einem DPM1 und den zugehörigen Slaves, oder zusätzliche Projektierungs- und Diagnosegeräte. Die Eingangs- und Ausgangsbilder der Slaves können von allen DP-Mastern gelesen werden. Das Schreiben der Ausgänge ist nur für einen DP-Master (den bei der Projektierung zugeordneten DPM1) möglich.

Um eine weitgehende Geräteaustauschbarkeit vom selben Typ zu erreichen, wurde bei DP auch das Systemverhalten standardisiert. Es wird im Wesentlichen durch den Betriebszustand des DPM1 bestimmt. Dieser kann entweder lokal oder über den Bus vom Projektierungsgerät gesteuert werden. Es werden die Hauptzustände Stopp (kein Datenverkehr zwischen dem DPM1 und den Slaves), Clear (der DPM1 liest die Eingangsinformationen der Slaves und hält die Ausgänge der Slaves im sicheren Zustand) und Operate unterschieden. Im Operate-Zustand werden vom DPM1 in einem zyklischen Datenverkehr die Eingänge von den Slaves gelesen und die Ausgangsinformationen an die Slaves übertragen. Der DPM1 sendet seinen Status in einem konfigurierbaren Intervall mit einem Multicast-Kommando zyklisch an alle ihm zugeordneten Slaves.

Bei der Projektierung des Bussystems legt der Anwender die Zugehörigkeit eines Slaves zum DPM1 fest. Weiterhin wird definiert, welche Slaves in den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden sollen. Der Datenverkehr zwischen dem DPM1 und den ihm zugeordneten Slaves wird dann in einer festgelegten, immer wiederkehrenden Reihenfolge automatisch durch den DPM1 abgewickelt. Er gliedert sich in die Parametrierungs-, Konfigurations- und Datentransferphase. Bevor der Master einen DP-Slave in die Datentransferphase aufnimmt, wird in der Parametrierungs- und Konfigurationsphase überprüft, ob die projektierte Sollkonfiguration mit der tatsächlichen Gerätekonfiguration übereinstimmt. Bei dieser Überprüfung müssen der Gerätetyp, die Format- und Längeninformationen sowie die Anzahl der Ein- und Ausgänge übereinstimmen. Der Benutzer erhält dadurch einen zuverlässigen Schutz gegen Parametrierungsfehler.

Weitere Schutzfunktionen tragen dazu bei, DP gegen einen Ausfall oder gegen Fehlfunktion zu sichern. Dazu gehört auch ein Überwachungsmechanismus beim DP-Master und bei den Slaves in Form einer Zeitüberwachung. Das Überwachungsintervall wird bei der Projektierung festgelegt. Der DPM1 überwacht den Datenverkehr der Slaves mit dem Data_Control_Timer. Für jeden Slave wird ein eigener Zeitgeber benutzt. Die Zeitüberwachung spricht an, wenn innerhalb eines Überwachungsintervalls kein ordnungsgemäßer Nutzdatentransfer erfolgt. Falls die automatische Fehlerreaktion (Auto_Clear = True)

freigegeben wurde, verlässt der DPM1 den Operate-Zustand, schaltet die Ausgänge der zugehörigen Slaves in den sicheren Zustand und geht in den Clear-Zustand über.

Zur Erkennung von Fehlern des Masters oder der Übertragung führt der Slave die Ansprechüberwachung durch. Findet innerhalb des Ansprechüberwachungsintervalls kein Datenverkehr mit dem Master statt, so schaltet der Slave die Ausgänge selbständig in den sicheren Zustand. Zusätzlich ist für die Ausgänge der Slaves beim Betrieb in Multi-Master-Systemen ein Zugriffsschutz erforderlich. Damit ist sichergestellt, dass der direkte Zugriff nur vom berechtigten Master erfolgt. Für alle anderen Master stellen die Slaves ein Abbild der Eingänge und Ausgänge zur Verfügung, das auch ohne Zugriffsberechtigung gelesen werden kann.

4.2.2.3.2 Leistungsstufe DP-V1

Der bei DP-V1 zusätzlich verfügbare azyklische Datenverkehr bildet die Voraussetzung für Parametrierung und Kalibrierung der Feldgeräte über den Bus während des laufenden Betriebes und für die Einführung bestätigter Alarmmeldungen. Die Übertragung der azyklischen Daten erfolgt parallel zum zyklischen Datenverkehr, allerdings mit niedrigerer Priorität. Der DPM1 (Master Class 1) besitzt die Sendeberechtigung (den Token) und korrespondiert per Aufforderung und Antwort mit Slave 1, danach mit Slave 2 usw. in fester Reihenfolge bis zum letzten Slave der aktuellen Liste; danach übergibt er den Token an den DPM2 (Master Class 2). Dieser kann in der noch verfügbaren Restzeit („Lücke“) des programmierten Zyklus eine azyklische Verbindung zu einem beliebigen Slave zum Austausch von Datensätzen aufnehmen. Am Ende der laufenden Zykluszeit gibt er den Token an den DPM1 zurück. Der azyklische Austausch von Datensätzen kann sich über mehrere Zyklen bzw. deren „Lücken“ hinziehen; am Ende nutzt der DPM2 wiederum eine Lücke zum Abbau der Verbindung. Neben dem DPM2 kann in ähnlicher Weise auch der DPM1 azyklisch Datenaustausch mit Slaves durchführen (MS1-Kanal). Als weitere Funktion wurde bei DP-V1 die gerätebezogene Diagnose verfeinert und in die Kategorien Alarne und Statusmeldungen aufgegliedert.

4.2.2.3.3 Leistungsstufe DP-V2

Der Slave-Querverkehr (Data Exchange Broadcast, DxB) und der Isochronous Mode sind die wichtigsten optionalen Ergänzungen bei DP-V2. DxB ermöglicht die direkte und damit Zeit sparende Kommunikation zwischen Slaves via Broadcast ohne den Umweg über einen Master. Dabei betätigen sich die Slaves als „Publisher“, d. h. die Slave-Antwort geht nicht nur zurück an den koordinierenden Master, sondern direkt auch an andere, in den Ablauf eingebundenen Slaves, die sogenannten „Subscriber“. Damit können die Slaves Daten aus anderen Slaves direkt verfolgen und als eigene Vorgaben verwenden. Das eröffnet neue Anwendungsmöglichkeiten, insbesondere für Motion Control Aufgaben, und kann die Reaktionszeiten am Bus um bis zu 90 % reduzieren.

Der Isochronous Mode ermöglicht eine taktsynchrone Regelung in Master und Slaves unabhängig von der Belastung des Busses. Mit Taktabweichungen kleiner einer Mikrosekunde können damit hochgenaue Positionierungsvorgänge realisiert werden. Dabei werden

alle beteiligten Geräte durch ein Broadcast-Telegramm „global control“ auf den Bus-Masterzyklus synchronisiert. Ein spezielles Lebenszeichen (laufende Nummer) gestattet die Überwachung der Synchronisation.

Eine weitere Option bei DP-V2 ist die Uhrzeitführung (Clock Control). Dabei schickt ein Uhrzeit-Master Zeitmarken (time stamps) an alle Slaves und synchronisiert damit alle Busteilnehmer auf eine System-Zeit mit einer Abweichung unter einer Millisekunde. Dadurch können Aktionen (events) zeitgenau verfolgt werden. Das ist vor allem bei der Erfassung zeitlicher Abläufe in Netzwerken mit vielen Mastern hilfreich. Diagnosen über Störungen werden dadurch ebenso erleichtert wie die zeitfolgerichtige Einplanung von Aktionen.

Die Funktion Up- und Download (Load Region) erlaubt das Laden beliebig großer Datenbereiche in ein Feldgerät mit wenigen Kommandos. Damit sind beispielsweise Programm-Updates oder Geräteaustausch ohne manuelle Ladevorgänge möglich.

Die Dienste der Function Invocation erlauben die Beeinflussung (Starten, Stoppen, Rücksetzen, Wiederanlauf) von Programmen oder den Aufruf (Call) von Funktionen (z. B. Messwert ermitteln) in einem DP-Slave.

4.2.2.3.4 Adressierung mit Slot und Index

Bei der Adressierung von Daten geht PROFIBUS davon aus, dass die Slaves physikalisch modular aufgebaut sind oder aber intern in logische Funktionseinheiten, so genannte Module, strukturiert werden können. Dieses Modell spiegelt sich in den DP-Grundfunktionen für den zyklischen Datenverkehr wider, wo jedes Modul eine konstante Anzahl Ein-/Ausgangsbytes besitzt, die an einer festen Position im Nutzdatentelegramm übertragen werden. Das Adressierungsverfahren basiert auf Kennungen, die den Typ eines Moduls als Input, Output oder eine Kombination aus beiden kennzeichnen. Alle Kennungen zusammen ergeben die Konfiguration eines Slaves, die im Hochlauf des Systems auch vom DPM1 überprüft wird.

Auch beim azyklischen Datenverkehr wird dieses Modell zugrunde gelegt. Alle für Schreib- oder Lesezugriffe freigegebenen Datenblöcke werden ebenfalls als den Modulen zugehörig betrachtet und können mit Hilfe von Slot-Number und Index adressiert werden. Die Slot-Number adressiert dabei das Modul, und der Index die einem Modul zugehörigen Datenblöcke. Jeder Datenblock kann bis zu 244 Byte groß sein (Abb. 4.24). Bei modularen Geräten ist die Slot-Number den Modulen zugeordnet. Die Module beginnen bei 1 und werden lückenlos in aufsteigender Reihenfolge festgelegt. Die Slot-Number 0 ist für das Gerät selbst vorgesehen.

Kompaktgeräte werden als eine Einheit von virtuellen Modulen betrachtet. Auch hier gilt die Adressierung mit Slot-Number und Index.

Durch die Längenangabe im Read- bzw. Write-Request können auch nur Teile eines Datenblocks gelesen bzw. geschrieben werden. Wenn der Zugriff auf den Datenblock erfolgreich war, antwortet der Slave mit einer positiven Read- bzw. Write-Response oder kann andernfalls in der negativen Response das Problem klassifizieren.

Tab. 4.3 Allgemeine Applikationsprofile

PROFIsafe	Das Profil definiert die sichere Kommunikation sicherheitsgerichteter Geräte (Not-Aus-Schalter, Lichtgitter u. a.) mit Sicherheitssteuerungen über PROFIBUS
Identification & Maintenance	Das Profil spezifiziert ein Konzept zur Identifikation von PROFIBUS-Geräten und den Internet-Zugriff auf gerätespezifische Informationen
iPar-Server	Das Profil definiert die Speicherung der zusätzlichen iParameter in der Steuerung und das Zurücklesen der iParameter nach einem Geräteaus tausch
Time Stamp	Das Profil definiert die zeitgenaue Zuordnung bestimmter Ereignisse und Aktionen durch Zeitstempelung
Redundancy	Das Profil spezifiziert den Mechanismus für Feldgeräte mit redundanten Kommunikationsverhalten.

4.2.2.4 Allgemeine Applikationsprofile

PROFIBUS besitzt einige Allgemeine Applikationsprofile, die Funktionen und Eigenschaften mit anwendungsübergreifender Bedeutung beschreiben und die in Verbindung mit spezifischen Applikationsprofilen eingesetzt werden können. Das bedeutendste allgemeine Applikationsprofil ist PROFIsafe.

4.2.2.4.1 PROFIsafe

Die dezentrale Feldbustechnik für die Fertigungs- und Prozessautomatisierung musste lange Zeit mit der Einschränkung leben, dass sicherheitstechnische Aufgaben nur mit konventioneller Technik in einer zweiten Ebene oder dezentral über Spezialbusse gelöst werden konnten. PI hat daher mit PROFIsafe für sicherheitsrelevante Anwendungen eine ganzheitliche, offene Lösung geschaffen, die den bekannten Anwenderszenarien für „Functional Safety“ gerecht wird und die sowohl mit PROFIBUS als auch PROFINET einsetzbar ist (Abb. 4.21). Das PROFIsafe-Protokoll arbeitet ohne Rückwirkungen auf PROFIBUS und PROFINET-Netzwerke und macht „Single-Channel“-Lösungen möglich, bei denen die Sicherheits-Nachrichten zusammen mit den Standardnachrichten über das gleiche Buskabel übertragen werden können.

PROFIsafe definiert, wie sicherheitsgerichtete Geräte (Not-Aus-Taster, Lichtgitter, Überfüllsicherungen, ...) mit Sicherheitssteuerungen so kommunizieren, dass sie in sicherheitsgerichteten Automatisierungsaufgaben bis SIL3 gemäß IEC 61508/IEC 62061 beziehungsweise PL „e“ gemäß ISO 13849-1 eingesetzt werden können. Es realisiert die sichere Kommunikation über ein Profil, d. h. über ein besonderes Format der Nutzdaten und ein spezielles Protokoll.

Die Spezifikation wurde von Herstellern, Anwendern, Normungsgremien und Prüfinstituten (TÜV, BIA) gemeinsam erarbeitet. Sie setzt auf einschlägige Standards auf, allen voran der IEC 61508, die besonders auf die Belange von Softwareentwicklungen eingehen.

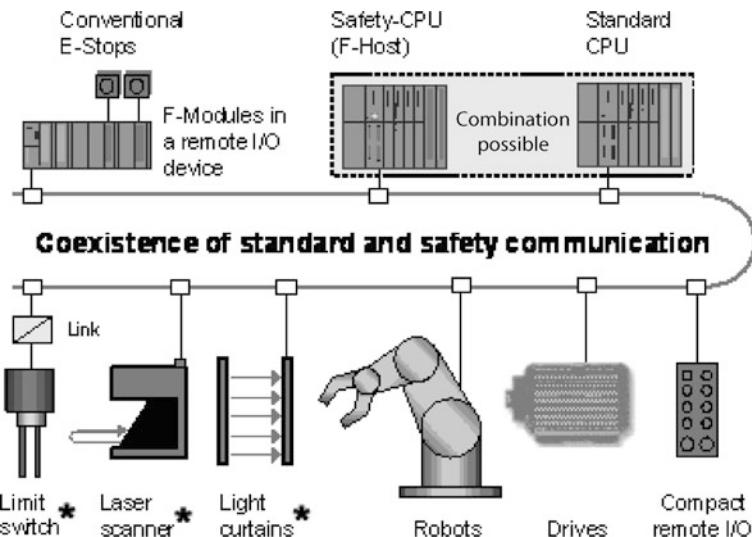


Abb. 4.21 Der „Single Channel“-Ansatz von PROFIsafe. *sicherheitsgerichtet

PROFIsafe berücksichtigt eine Vielzahl von Fehlermöglichkeiten, die bei einer seriellen Buskommunikation auftreten können, wie Verzögerung, Verlust oder Wiederholung von Daten, falsche Reihenfolge, Fehladressierung oder Datenverfälschungen.

Hierfür gibt es eine Reihe von Abhilfemaßnahmen, aus denen für PROFIsafe folgende ausgewählt wurden:

- Fortlaufende Nummerierung der Sicherheitstelegramme.
- Zeiterwartung für ankommende Telegramme und deren Quittierung.
- Kennung zwischen Sender und Empfänger („Passwort“).
- Zusätzliche Datensicherung (CRC, Cyclic Redundancy Check).

PROFIsafe ist eine einkanalige Softwarelösung, die in den Geräten als zusätzliche Schicht (PROFIsafe Layer) oberhalb der Schicht 7 implementiert wird; die Standard-Komponenten wie Leitungen, ASICs oder Protokolle bleiben unverändert. Dadurch sind Redundanzbetrieb und Nachrüstbarkeit gegeben. Geräte mit dem PROFIsafe-Profil können ohne Einschränkung zusammen mit Standardgeräten an ein- und demselben Bus (Kabel) in Koexistenz betrieben werden.

PROFIsafe nutzt azyklische Kommunikation und kann mit RS485-, LWL- oder MBP-Übertragungstechnik betrieben werden. PROFIsafe steht als generischer Software-Treiber für verschiedene Entwicklungs- und Ablaufumgebungen zur Verfügung.

Gemeinsam mit dem PROFIdrive Profil lassen sich mit PROFIsafe die in der IEC 61800-5-2 definierten Sicherheitsfunktionen für Antriebe mit integrierter Sicherheit realisieren. Hierzu zählen die Funktionen Sicher abgeschaltetes Moment, Sicherer Stopp 1,

Sicherer Stopp 2 und Sicherer Betriebshalt sowie die Überwachungsfunktionen Sicher begrenzte Beschleunigung, Sicher begrenzte Geschwindigkeit, Sicher begrenztes Moment/Kraft, Sicher begrenzte Position, Sicher begrenztes Schrittmass, Sichere Bewegungsrichtung und Sicher begrenzte Motortemperatur. Somit können herkömmliche elektromechanische Komponenten durch elektronische sichere Stopp- und Überwachungsfunktionen ersetzt werden. Der Anwender gewinnt die Möglichkeit, systemimmanent die Funktion der Antriebssteuerung zu überwachen und nur bei Ausfällen abzuschalten.

4.2.2.4.2 Redundanz

Ein weiteres Beispiel für die Allgemeinen Applikationsprofile von PROFIBUS ist Slave-Redundanz. In vielen Anwendungen ist die Installation von Feldgeräten mit redundantem Kommunikationsverhalten gewünscht. Bei PROFIBUS wurde hierfür die Spezifikation eines Slave-Redundanz-Mechanismus erarbeitet, bei dem die Slave-Geräte zwei unterschiedliche PROFIBUS-Anschaltungen enthalten, die Primary und die Backup-Anschaltung.

4.2.2.5 Spezifische Applikationsprofile (Tab. 4.4)

PROFIBUS hat erfolgreich das Konzept umgesetzt, einerseits ausgeprägt branchenspezifische Anforderungen der Anwender in den spezifischen Profilen zu berücksichtigen und andererseits die Gesamtheit dieser Anwendungen in ein standardisiertes und offenes Feldbusssystem zu vereinen. Einige der spezifischen Applikationsprofile werden hier kurz vorgestellt.

4.2.2.5.1 PROFIdrive

Das Applikationsprofil PROFIdrive definiert das Geräteverhalten und das Zugriffsverfahren auf Antriebsdaten für elektrische Antriebe an PROFIBUS und PROFINET, vom einfachen Frequenzumrichter bis hin zu hochdynamischen Servoreglern. Die Einbindung von Antrieben in Automatisierungslösungen ist stark von der Antriebsaufgabe abhängig. Daher definiert PROFIdrive sechs Anwendungsklassen, denen sich die meisten Anwendungen zuordnen lassen.

Im einfachsten Fall (Klasse 1) wird der Antrieb über einen Haupt-Sollwert (z. B. Drehzahl) über PROFIBUS gesteuert. Die komplette Drehzahlregelung erfolgt im Antriebsregler. Dieser Anwendungsfall wird vorrangig im Bereich der klassischen Antriebstechnik (z. B. Fördertechnik, Frequenzumrichter) verwendet.

Bei einem Standardantrieb mit Technologiefunktion (Klasse 2) wird der Automatisierungsprozess in mehrere Teilprozesse zerlegt und die Automatisierungsfunktionen sind teilweise vom zentralen Automatisierungsgesetz auf die Antriebsregler ausgelagert. PROFIBUS dient dabei als Technologie-Schnittstelle. Für diese Lösung ist der Slave-Querverkehr zwischen den einzelnen Antriebsreglern Voraussetzung.

Der Positionierantrieb (Klasse 3) schließt eine zusätzliche Positioniersteuerung im Antrieb ein und deckt damit ein sehr weites Anwendungsfeld ab, z. B. das Auf- und Abdrehen von Flaschenverschlüssen. Die Positionieraufträge werden über PROFIBUS an die Antriebsregler übergeben und gestartet.

Tab. 4.4 Spezielle Applikationsprofile bei Profibus

PROFIdrive	Das Profil beschreibt das Geräteverhalten und die Zugriffsverfahren auf Daten für drehzahlveränderbare elektrische Antriebe an PROFIBUS mit Sicherheitssteuerungen über PROFIBUS
PA Devices	Das Profil spezifiziert die Eigenschaften von Geräten der Prozesstechnik in der Prozessautomatisierung an PROFIBUS
Robots/NC	Das Profil beschreibt die Steuerung von Handhabungs- und Montagerobotern über PROFIBUS
Encoder	Das Profil beschreibt die Ankopplung von Dreh-, Winkel- und Linear-Encodern mit Singleturm- und Multiturm-Auflösung
Fluid Power	Das Profil beschreibt die Ansteuerung von hydraulischen Antrieben über PROFIBUS (Zusammenarbeit mit VDMA)
SEMI	Das Profil beschreibt Eigenschaften der Geräte für die Halbleiterherstellung an PROFIBUS (SEMI-Standard)
Low Voltage Switchgear	Das Profil definiert den Datenaustausch für Niederspannungsschaltgeräte (Lasttrenner, Motorstarter u. a.) an PROFIBUS
Dosing/Weighing	Das Profil beschreibt den Einsatz von Wäge- und Dosiersystemen an PROFIBUS
Ident Systems	Das Profil beschreibt die Kommunikation zwischen Geräten zur Identifizierung (Barcode-Leser, Transponder)
Liquid Pumps	Das Profil definiert den Einsatz von Flüssigkeitspumpen an PROFIBUS (Zusammenarbeit mit VDMA)
Remote I/O for PA Devices	Das Profil definiert die Austauschbarkeit von Remote I/O-Geräten in der Prozessautomatisierung
HART on PROFIBUS	Das Profil definiert die Einbindung von HART-Geräten in PROFIBUS-Systeme
LabDevices	Das Profil spezifiziert die Eigenschaften von Laborgeräten bei der Laborautomatisierung an PROFIBUS

Die Zentrale Bewegungssteuerung (Klasse 4 für Drehzahl-Sollwert und Klasse 5 für Positions-Sollwert) ermöglicht den koordinierten Bewegungsablauf mehrerer Antriebe. Die Bewegungsführung wird überwiegend mit einer zentralen numerischen Steuerung realisiert. PROFIBUS dient zur Schließung des Lageregelkreises sowie zur Synchronisation der Takte. Diese Lösung erlaubt durch ihr Lageregelkonzept „Dynamic Servo Control“ auch sehr anspruchsvolle Anwendungen mit Linearmotoren.

Die Dezentrale Automatisierung bei getakteten Prozessen und elektronischer Welle (Klasse 6) kann unter Verwendung des Slave-Querverkehrs und der isochronen Slaves realisiert werden. Beispiele sind Applikationen wie „Elektrisches Getriebe“, „Kurvenscheibe“ oder „Winkelsynchronlauf“.

PROFIdrive definiert ein Gerätemodell aus Funktionsmodulen, die geräteintern zusammenarbeiten und die Intelligenz des Antriebssystems widerspiegeln. Im Gegensatz zu anderen Antriebsprofilen definiert PROFIdrive nur die Zugriffsmechanismen auf die Parameter sowie einen Subset von ca. 30 Profilparametern, wozu z. B. Störfuffer, Antriebssteuerung, Geräteidentifikation u. a. gehören. Alle anderen Parameter (bei komplexen Geräten sind über 1000 möglich) sind herstellerspezifisch, was den Antriebshersteller-

lern große Flexibilität bei der Realisierung der Regelfunktionen gibt. Der Zugriff auf die Elemente eines Parameters erfolgt azyklisch über den DP-V1-Parameterkanal.

Im Markt werden verstärkt Antriebe eingesetzt, die bereits über integrierte Sicherheitstechnik verfügen. Dies bietet den Vorteil, dass keine externen Überwachungsgeräte mehr nötig sind (reduzierter Verdrahtungsaufwand und geringerer Platzbedarf). PROFIdrive und PROFIsafe ergänzen sich hier in idealer Weise. Beim gemeinsamen Einsatz beider Profile entsteht eine harmonische Einheit, mit der Sicherheitsfunktionen zusammen mit Standard-Antriebsfunktionen über den gleichen Bus angesteuert werden können.

4.2.2.5.2 PA Devices

Das Applikationsprofil für PA-Geräte (PA Devices) spielt in der Prozessautomatisierung (PA) eine grundlegende Rolle. Es definiert für verfahrenstechnische Geräte – wie Transmitter, Stellantriebe oder Analysengeräte – herstellerunabhängige Geräteparameter und -funktionen. Damit bildet es das Fundament für die Einheitlichkeit der Anwendungen, vereinfachtes Engineering oder erhöhte Zuverlässigkeit durch standardisierte Diagnoseinformationen. Die Übertragungstechnik von PROFIBUS PA vereinigt als 2-Leitertechnik die Datenübertragung und die Stromversorgung für den Ex-Bereich. Als Kommunikationstechnik wird PROFIBUS DP-V1 eingesetzt, womit sowohl die zyklische als auch die bedarfsgesteuerte azyklische Übertragung realisiert wird.

Die kürzlich erschienene Version PA V3.02 vereinfacht die Geräteintegration über den Lebenszyklus einer Produktionsanlage. Dazu gehören unter anderem Vorschriften zur Kennzeichnung der Software-Variante am Gerät, die automatische Anpassung an die Funktionalität von Vorgängerversionen im zyklischen Verkehr sowie herstellerübergreifende Richtlinien für Änderungen der Gerätesoftware und deren Auswirkung auf Kompatibilität. Auch die Abbildung der spezifischen Diagnoseinformationen von Feldgeräten auf standardisierte Kategorien gemäß der NAMUR-Empfehlung NE107 (Selbstüberwachung und Diagnose von Feldgeräten) und der wesentlich schnellere Transfer von Feldgerätedaten, z. B. bei der Übertragung von parametrierten Daten während eines Gerätetauschs, wurden in dem Profil als verpflichtend zu implementierende Funktionalität festgelegt.

4.2.2.5.3 Fluid Power

In enger Anlehnung an die Definitionen des PROFIdrive Profils werden hier Datenaustauschformate und Parameter für Proportionalventile, hydrostatische Pumpen und hydrostatische Antriebe beschrieben. Für die Parameterversorgung der Geräte sind alternativ ein Parameterkanal auf DP-V0 oder die azyklische Kommunikation über DP-V1 vorgesehen.

4.2.2.5.4 Ident Systems

Ident Systems ist ein Profil für Barcode-Lesegeräte und Transponder-Systeme. Diese sind besonders prädestiniert, die DP-V1-Funktionalität zu nutzen. Während der zyklische Datenübertragungskanal für kleine Datenmengen genutzt wird, um die Status- bzw. Steuerinformationen zu tragen, dient der azyklische Kanal der Übertragung der großen Datenmengen, die sich aus der Information im Barcode oder Transponder ergeben.

4.2.2.6 Gerätemanagement

Moderne Feldgeräte stellen vielfältige Informationen bereit und nehmen Funktionen wahr, die früher SPS und Leitsystemen vorbehalten waren. Die Tools für Inbetriebnahme, Wartung, Engineering und Parametrierung dieser Geräte benötigen eine exakte und vollständige Beschreibung deren Daten und Funktionen, wie Art der Applikationsfunktion, Konfigurationsparameter, Maßeinheiten, Wertebereich, Grenzwerte, Defaultwerte u. a.. Gleichermaßen gilt für die Steuerung bzw. das Leitsystem, denen für einen reibungslosen Datenaustausch mit den Feldgeräten, deren spezifische Parameter und Datenformate ebenfalls bekannt gegeben („integriert“) werden müssen.

Für eine derartige Gerätebeschreibung wurden bei PROFIBUS Methoden und Tools („Integrationstechnologien“) entwickelt, mit denen das Gerätemanagement standardisiert werden kann. Die Tools sind in ihrem Leistungsumfang auf bestimmte Aufgaben optimiert, sodass sich hierfür auch der Begriff der skalierbaren Geräteintegration eingebürgert hat.

Im Fertigungsbereich wird historisch bedingt vorwiegend die GSD (Geräte-Stamm-daten-Datei, General Station Description) eingesetzt. Zunehmend findet heute auch FDT (Field Device Tool) Verwendung. In der Prozessautomatisierung werden EDD (Electronic Device Description) und FDT verwendet. FDI (Field Device Integration) soll eine einheitliche Lösung für die Integration von Feldgeräten erreichen und eine Architektur besitzen, die FDT und EDD Technologien zu einem gemeinsamen Standard für Feldgeräteintegration migriert.

4.2.2.7 PROFIBUS Implementierung

Für die Geräteentwicklung bzw. Implementierung des PROFIBUS-Protokolls steht ein breites Spektrum von Basistechnologiekomponenten und Entwicklungswerkzeugen (PROFIBUS ASICs, PROFIBUS Stacks, Monitore, Testtools und Inbetriebnahmewerkzeuge) sowie Dienstleistungen zur Verfügung, die den Geräteherstellern eine effiziente Entwicklung ermöglichen. Für niedrige bis mittlere Stückzahlen eignen sich PROFIBUS-Schnittstellenmodule. Diese bis zu scheckkartengroßen Modulen realisieren das gesamte Busprotokoll und bieten einen festen Schnittstellenumfang für Geräteapplikationen an. Sie können als Zusatzmodul auf die Grundplatine des Gerätes aufgebracht werden. Bei hohen Stückzahlen bietet sich eine individuelle Implementierung auf Basis von handelsüblichen PROFIBUS Basistechnologiekomponenten an.

Für einfache E/A-Geräte bietet sich die Implementierung mit Single-Chip ASICs an. Alle Protokollfunktionen sind bereits auf dem ASIC integriert. Es wird kein Mikroprozessor oder weitere Kommunikations-Software benötigt. Lediglich die Businterface-Treiber, der Quarz und die Leistungselektronik sind als externe Komponenten erforderlich. Eine weitere Möglichkeit stellen Mikroprozessoren mit einem integrierten PROFIBUS-Kern dar. Für die Implementierung komplexer Master-Geräte stehen, wie für Slave-Implementierungen, ASICs unterschiedlicher Hersteller zur Verfügung. Sie können in Kombination mit vielen gängigen Mikroprozessoren betrieben werden.

4.2.2.8 Qualitätssicherung und Zertifizierung

Damit PROFIBUS Geräte unterschiedlicher Typen und Hersteller ihre Aufgaben im Automatisierungsprozess korrekt erfüllen, müssen sie über den Bus Informationen fehlerfrei austauschen. Voraussetzung dafür ist eine normkonforme Implementierung der Kommunikationsprotokolle und Anwendungsprofile durch die Gerätehersteller. Zur Gewährleistung dieser Forderung hat PROFIBUS International (PI) ein Qualitätssicherungs- und Zertifizierungs-Verfahren etabliert. Ziel der Zertifizierung ist es, den Anwendern für den gemeinsamen Betrieb von Geräten unterschiedlicher Hersteller die notwendige Sicherheit für eine fehlerfreie Funktion zu geben. Hierzu werden die Geräte in unabhängigen, gemäß den Qualitätsrichtlinien von PI akkreditierten Prüflaboren mit der notwendigen Prüfschärfe praxisnah getestet. Fehlinterpretationen der Normen durch die Entwickler können so vor dem Einsatz erkannt und vom Hersteller beseitigt werden. Auch das Zusammenspiel des Gerätes mit anderen zertifizierten Geräten ist Gegenstand der Tests. Nach erfolgreich bestandener Prüfung wird auf Antrag durch den Hersteller durch die Zertifizierungsstelle von PI ein Geräte-Zertifikat erteilt.

4.2.3 Interbus

Das Interbus-System hat seinen Einsatzschwerpunkt im Bereich der Sensorik/Aktorik, also unterhalb der Steuerungsebene, ausgelegt. Es wird deshalb auch abgrenzend zu anderen Feldbussystemen als Sensor-/Aktorbus-System bezeichnet. Neben diesem Einsatzschwerpunkt bietet das Protokoll aber auch die Möglichkeit der Vernetzung von komplexen „intelligenten“ Geräten wie z. B. Technologiesteuerungen und ebenso den Anschluss einzelner diskreter Sensoren- und Aktoren. Interbus kann so eine komplexe Automatisierungsaufgabe unabhängig von den so genannten Hierarchie-Ebenen durchgängig lösen.

4.2.3.1 Topologie

Interbus arbeitet mit einem Master-Slave-Zugriffsverfahren, wobei der Bus-Master gleichzeitig die Kopplung an das überlagerte Steuerungssystem realisiert. Topologisch ist Interbus ein Ringsystem, d. h., alle Teilnehmer sind aktiv in einen in sich geschlossenen Übertragungsweg eingekoppelt (Abb. 4.22). An dem vom Master ausgehenden Hautring können zur Strukturierung des Gesamtsystems Subringsysteme angeschlossen werden. Die Ankopplung solcher Subringsysteme erfolgt durch spezielle Komponenten, die als Busklemmen bezeichnet werden. Ein Subsystem kann lokale Ausprägung haben, man spricht dann vom so genannten Peripheriebus, der zur Bildung von lokalen I/O-Clustern innerhalb eines Schaltschrances dient. Es kann aber auch wiederum ein System sein, das dezentrale Teilnehmer über große Distanzen ankoppelt.

Als Besonderheit gegenüber anderen Ringsystemen werden beim Interbus-System sowohl die Datenhинleitung als auch die -rückleitung innerhalb eines Kabels und durch sämtliche Teilnehmer geführt. Hierdurch ergibt sich das physikalische Erscheinungsbild einer Linien- bzw. Baumstruktur.

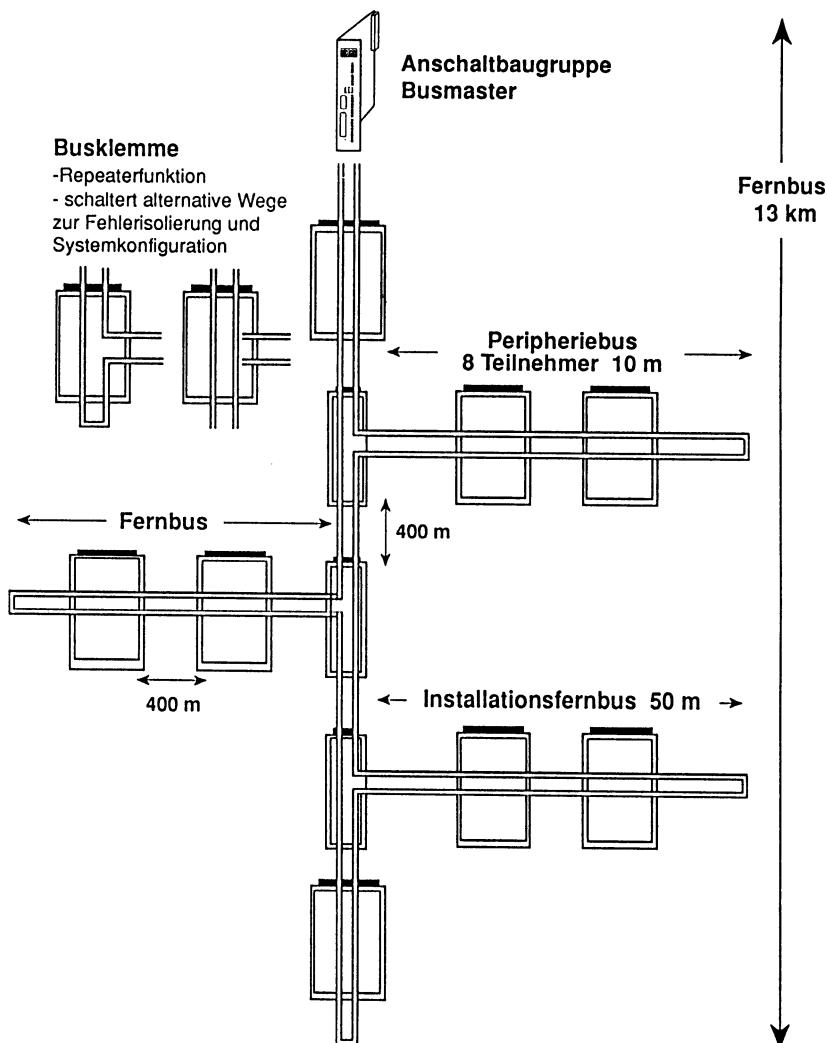


Abb. 4.22 Interbus-Topologie

Die physikalische Ebene des Interbus-Systems wird mit dem RS 485-Standard realisiert. Zur Übertragung der Signale werden verdrillte (twisted-pair-)Leitungen verwendet. Das Interbus-Kabel benötigt aufgrund der Ringstruktur und des zusätzlichen Mitführen des Logic-grounds zwischen zwei Geräten fünf Adern. Bei einer Datenübertragungsrate von 500 kbit/s ist aufgrund der RS 485 Punkt-zu-Punkt-Übertragung eine Distanz von 400 m zwischen zwei Geräten möglich. Durch die integrierte Repeater-Funktion in jedem Teilnehmer lässt sich eine Gesamtausdehnung des Interbus-Systems von bis zu 13 km

erreichen. Die Anzahl der Interbus-Teilnehmer wurde, um das System handhabbar zu halten, auf maximal 256 Geräte begrenzt.

Die Punkt-zu-Punkt-Struktur des Interbus-Systems sowie die Aufteilung in Haupt- und Subringsysteme kommen dem Einsatz von unterschiedlichsten physikalischen Übertragungstechniken und hier speziell der zukunftsweisenden Lichtleiter-Technik ideal entgegen. So kann z. B. heute an jeder beliebigen Stelle des Bussystems eine Umsetzung von der Kupfertechnik auf Lichtwellenleitertechnik, auf Datenlichtschranken oder andere Medien mit am Markt verfügbaren Standardumsetzern erfolgen. Ebenso lassen sich mit lichtwellenleiterfähigen Geräten direkt ganze Netzwerke auf Basis dieser störsicheren und installationsfreundlichen Technik aufbauen. Der Einsatz von aufwändigen selbststeuernden Repeater-Umsetzern, wie er in Multidrop-Strukturen erforderlich ist, ist hier nicht notwendig.

Die Verwendung der Ringstruktur bringt zwei entscheidende Vorteile für das System mit sich. Zum einen bietet der Ring im Gegensatz zur Linienstruktur die Möglichkeit des zeitgleichen Sendens und Empfangens von Daten (Vollduplex) zum anderen lässt sich mit einem Ringsystem eine wesentliche Verbesserung der Eigendiagnostik des Systems erreichen. Bei busförmigen Systemen (Linie) mit so genannter Multi-Drop-Ankopplung der Teilnehmer sind alle Geräte quasi passiv an den Bus angekoppelt. Die Passivität des Teilnehmers beschränkt sich jedoch nur auf den fehlerfreien Betrieb bzw. auf eine Unterbrechung des Teilnehmers. Bewirkt ein Fehler in der Busanschaltung eines Teilnehmers einen Kurzschluss der Busleitung oder wird die Leitung einer anderen Stelle außerhalb des Teilnehmers kurzgeschlossen, so ist in einem solchen System keinerlei Kommunikation mehr möglich. In diesem Fehlerfall kann in busförmigen Systemen keine Selektion der Fehlerstelle über automatische Diagnosefunktionen des Netzwerkes erfolgen. Ein Ringsystem mit aktiver Teilnehmerankopplung erlaubt dagegen eine Segmentierung des Gesamtkomplexes in elektrisch unabhängige Teilsysteme. Bei einem aktiven Fehler eines Teilnehmers sowie bei einem Kurzschluss oder einer Unterbrechung der Busleitung fällt auf diese Weise nur die Kommunikation ab der Fehlerstelle aus. Die Lokalisierung des Fehlerortes durch Netzmanagementfunktionen im Bus-Master ist weiter möglich, sodass eine gezielte Fehlerbehebung durch den Service-Techniker erfolgen kann. Gleichermaßen gilt bei sporadischen Übertragungsstörungen wie sie z. B. durch elektromagnetische Störquellen oder fehlerhafte Verkabelung ausgelöst werden. Im Liniensystem werden hierdurch zufällig beliebige Telegramme zerstört. Eine Lokalisierung des Fehlerorts ist bei dieser typischen Anlagenstörung nicht möglich. Das Ergebnis sind immer wiederkehrende, oft lang anhaltende Betriebsunterbrechungen. Beim Interbus-System ist aufgrund der aktiven Teilnehmerankopplung mit Überwachung jeder einzelnen Übertragungsstrecke auch in diesem Fall eine eindeutige Fehlerlokalisierung möglich.

Die Möglichkeit der Bildung von lokalen Unterringsystemen im Interbus-Netzwerk lässt zusätzlich auch hier ein quasi rückwirkungsfreies An- und Abkoppeln von Teilnehmern zu. Die Koppelemente zwischen den Bussegmenten erlauben, gesteuert vom zentralen Bus-Master, ein An- und Abschalten eines Sub-Systems. Manipulationen am Untersystem sind so ohne Rückwirkungen auf das restliche System möglich.

4.2.3.2 Interbus-Protokoll

Der Interbus-Protokollstack ist entsprechend dem ISO/OSI-Modell in drei Schichten aufgebaut. Zur optimalen Unterstützung der beiden in der Sensorik/Aktorik vorkommenden Datenklassen – den zyklischen Prozessdaten und den azyklischen Parametern – wurde als Besonderheit eine hybride Struktur vorgesehen. Abhängig von der Datenklasse ist der Interbus-Protokollstack unterschiedlich weit ausgebaut.

Basis für beide Datenklassen ist das so genannte zyklische E/A-Protokoll (BLL: Basic Link Layer) des Data-Link-Layers. Bei der Auswahl des hier verwendeten Übertragungsverfahrens wurde der Tatsache Rechnung getragen, dass es sich bei der Masse der in der Sensorik/Aktorik anfallenden Daten um sehr kurze (nur einige Bit) und zyklisch anfallende Prozessinformationen handelt. Aufgrund des hieraus resultierenden geringen Informationsgehalts pro Netzwerkteilnehmer muss ein in Bezug auf den Übertragungs-Overhead (Adressierung, Datensicherung, Kommandofeld) optimiertes Verfahren verwendet werden. Hierzu wurde entgegen den sonst üblichen nachrichtentechnischen Protokollen ein so genanntes Summenrahmenverfahren eingesetzt. Die jeweils sehr kurzen Informationspakete der einzelnen Netzwerkteilnehmer werden zusammengefasst und in einem gemeinsamen Telegrammrahmen übertragen (Abb. 4.23). Hierdurch findet eine Erhöhung des Nutzdatenblocks im Telegramm statt. Das Summenrahmentelegramm enthält so die Information für sämtliche Netzwerkteilnehmer. Es wird gleichzeitig an alle Teilnehmer gesendet, wobei jeder Teilnehmer die für ihn bestimmten Informationen aus dem Gesamtrahmen entnimmt, bzw. die Informationen, die er dem Master liefern will, in den Rahmen einfügt.

Die Effizienz dieses Protokollverfahrens steigt mit der Anzahl der Netzwerkteilnehmer, die typischerweise im Bereich der Sensorik/Aktorik sehr hoch ist. Das Summenrahmenverfahren kann so leicht Protokolleffizienzen von weit über 60 %, bei Verwendung von Teilnehmern mit Ein-/und Ausgängen, sogar über 130 %, bieten, während nachrichtentechnische Protokolle in diesen Anwendungen im Bereich von < 5 % liegen. Da, bedingt

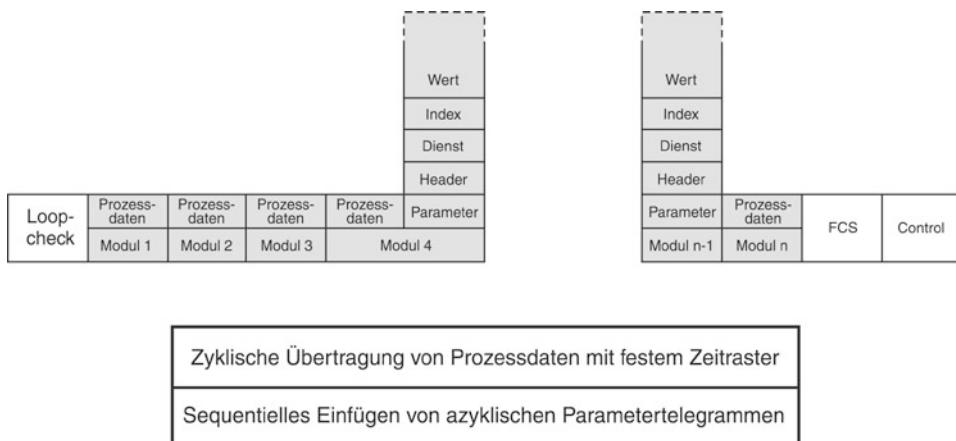


Abb. 4.23 Das Interbus-Übertragungsprotokoll

durch die Protokollstruktur, immer gleichzeitig alle Geräte angesprochen werden, gewährleistet das Summenrahmenübertragungsverfahren Synchronität und Zeitäquidistanz. Um mit diesen vom Prinzip her zyklischen Übertragungsverfahren auch die geforderten azyklischen Parametern zu bedienen, wird, aufbauend auf das zyklische E/A-Protokoll, eine Erweiterung des Data-Link-Layers vorgenommen.

Die Integration von Parameterinformationen in dem Protokollrahmen erfolgt dadurch, dass für Teilnehmer, die mit Parametern versorgt werden sollen, zusätzliche 2–16 Byte breite Zeitschlitz im Telegrammrahmen freigehalten werden (Abb. 4.23).

In diese Zeitschlitz werden dann bei Bedarf Parameterblöcke sequentiell eingefügt. Das bedeutet, dass ein kompletter Parameterblock in einzelne, kurze Informationsteile zerlegt wird, die nacheinander in das zyklische Protokoll eingebracht werden. Abhängig von der Ausprägung des Parameterkanals belasten der Transfer von Parameterblöcken und Programmen dadurch den zyklischen Echtzeit-Transfer nur so stark, wie beispielsweise 16 binäre Prozessinformationen. Mit jedem Übertragungsrahmen und damit jeder Aktualisierung der Prozessdaten wird so ein Teil des kompletten Parameterblocks übertragen. Die Zerlegung des Blocks in diese Teileinformationen und die erneute Zusammensetzung der Teileinformationen beim anderen Kommunikationspartner erfolgt durch die ergänzende Protokollsoftware des Data-Link-Layers und ist für den Anwender des Systems nicht sichtbar. Die Länge des Übertragungsprotokolls und damit die Abtastzeit für Prozessdaten im System bleibt durch die feste Reservierung von Übertragungsfenstern für die Parameter konstant. Dies gilt unabhängig davon, ob die Übertragung eines komplexen Parameterblocks zu einem Teilnehmer erfolgt oder nicht. Auf diese Weise lassen sich mit Interbus ohne Probleme Datenblöcke von mehreren 100 kByte übertragen, ohne dass das zyklisch kurze Abtastraster für die Prozessdaten verändert wird. Das spezielle Interbus-Protokoll lässt andererseits eine beliebige Verringerung der Nutzdaten pro Teilnehmer zu, ohne dass eine Reduzierung der Effektivität und damit des Datendurchsatz auftritt. Über Interbus können so auch problemlos Sensoren und Aktoren direkt vernetzt werden. Diese Möglichkeit wird ergänzt durch eine spezielle physikalische Übertragungstechnik als Interbus Sensor Loop bezeichnet. Kennzeichnend ist hier die Verwendung einer 2-Draht-Leitung ohne Schirm, die gleichzeitig Daten und Energie führt und so einen kostengünstigen Anschluss von Sensoren und Aktoren ermöglicht.

Das für den Einsatzbereich optimierte Protokoll des Interbus-Systems und seine Topologie lassen bei einer Übertragungsrate von nur 500 kbit Übertragungszeiten für Prozessdaten im untersten ms-Bereich zu. Die Übertragungszeit lässt sich sehr einfach entsprechend der unten angegebenen Formel ausrechnen und steht in linearer Abhängigkeit von der Anzahl der Datenpunkte des Systems. Bei einer Länge des Summenrahmenprotokolls von beispielsweise 128 Byte Eingangs- und 128 Byte Ausgangsinformationen ergibt sich so eine Übertragungszeit von < 4 ms.

Formel Übertragungszeit:

$$t_{\text{ü}} = [13 \cdot (6 + n) + 1,5 \cdot m] \cdot t_{\text{Bit}} + t_{\text{sw}} ;$$

$t_{\text{ü}}$ = Übertragungszeit

n = Anzahl der Nutzdatenbyte (pro Teilnehmer nur Eingangs- oder Ausgangsbyte ansetzen)

m = Anzahl der installierten Fernbusteilnehmer

t_{Bit} = Bitdauer $t_{\text{Bit}} = 2 \mu\text{s}$ bei 500 kbit/s

t_{sw} = Softwarelaufzeit $t_{\text{sw}} = 200 \mu\text{s}$.

Die genannten, extrem niedrigen Übertragungszeiten lassen sich, wenn es zukünftige Anwendungen erfordern, durch eine Erhöhung der Übertragungsrate noch wesentlich verringern. So ist eine Steigerung auf 2 MBd technisch möglich.

Die maximale Länge des Summenrahmentelegramms wird heute mit 512 Byte begrenzt. Diese Begrenzung liegt weniger im Protokoll als in den Speicherressourcen der Bus-Master begründet.

4.2.3.3 Protokollrealisierung

Das beschriebene Summenübertragungsverfahren wird beim Interbus durch eine Schieberegisterstruktur realisiert. Jeder Interbus-Teilnehmer fügt sich hierbei durch ein Schieberegister, dessen Länge durch die Anzahl der Prozessdatenpunkte des Teilnehmers festgelegt wird, in den Ring ein (Abb. 4.24). Durch die Aneinanderkopplung aller Teilnehmer ergibt sich so ein Schieberegisterring, dessen Länge und Struktur genau dem Aufbau des Nutzdatenfeldes im Summenrahmentelegramm entspricht. Die Prozessdaten, die an die Peripherie ausgegeben werden sollen, sind entsprechend der physikalischen Reihenfolge der angeschlossenen Ausgabestationen im Ausgabebuffer des Masters hinterlegt. Ein Übertragungszyklus erfolgt nun dadurch, dass nacheinander, gesteuert vom zentralen Bus-Master, alle Ausgabedaten auf dem Bus und damit durch die angeschlossenen Schieberegister getaktet werden. Während diese Datenausgabe durchgeführt wird, erfolgt gleichzeitig der

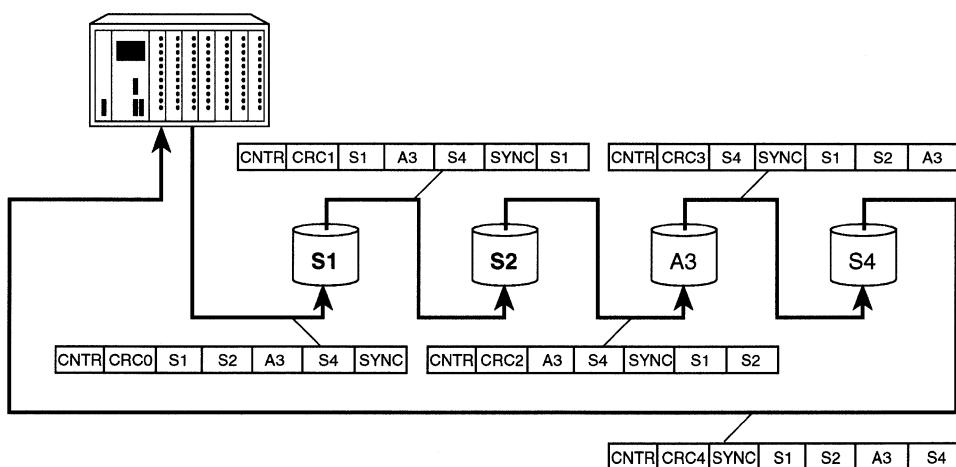


Abb. 4.24 Schieberegister-Struktur

Rückfluss von Prozessinformationen als Eingabedaten in den Eingangsbuffer des Masters. Nachdem so das gesamte Summenrahmentelegramm ausgegeben und gleichzeitig wieder eingelesen wurde, sind alle Ausgabedaten in den Schieberegistern der einzelnen Teilnehmer richtig positioniert. Über ein spezielles Steuerkommando teilt der Master den Teilnehmern das Ende des Übertragungszyklus mit. Nach der Durchführung einer Datensicherungssequenz werden dann die Prozessausgabeinformationen aus den Schieberegistern übernommen, in den Teilnehmern gespeichert und an die Peripherie weitergegeben. Ebenfalls zu diesem Zeitpunkt werden neue Peripherieinformationen in die Schieberegister von Eingabestationen eingelesen und somit der nächste Eingabezyklus vorbereitet. Der beschriebene Vorgang erfolgt nun beständig in einer aufeinander folgenden Weise, sodass Ein- und Ausgabedatenbuffer des Masters zyklisch ausgegeben bzw. aktualisiert werden.

Die im Summenrahmenprotokoll übertragenen Daten beinhalten die folgenden Informationen:

Jedes zyklische Telegramm beginnt mit einem so genannten Loopback-Wort. Dies sind 16 Datenbits, die als erste Informationen vom Master auf den Ring ausgegeben werden. Sie durchlaufen sämtliche Register des Ringsystems und werden als letzte Eingangsinformationen wieder in den Master zurückgelesen. Dem Loopback-Wort folgen die eigentlichen Nutzdateninformationen des Interbus-Systems. Am Ende des Nutzdatenblocks schließt sich die Übertragung einer 16 Bit langen Datensicherungssequenz an.

Diese Datensicherung erfolgt durch einen 16 Bit CRC-Check nach CCITT. Bedingt durch die physikalische Punkt-zu-Punkt-Struktur, findet der Datensicherungsmechanismus immer zwischen zwei benachbarten Teilnehmern statt. Der Austausch und Vergleich der CRC-Polynom-Reste erfolgt gesteuert durch die Rahmenende-Kennung gleichzeitig zwischen allen Geräten, so dass der durch den CRC-Check auftretende Übertragungs-overhead nur einmalig für das gesamte System wirkt.

Diesen CRC-Daten folgt noch einmal eine Übertragung von 16 Bit, in denen die einzelnen Teilnehmer die fehlerfreie Datenübertragung an den Master zurückbestätigen.

Zur Übertragung dieser Informationen wird ein asynchrones Verfahren mit Start- und Stopp-Bit verwendet (Abb. 4.25). Jeweils 8 Informationsbits werden hierbei um einen

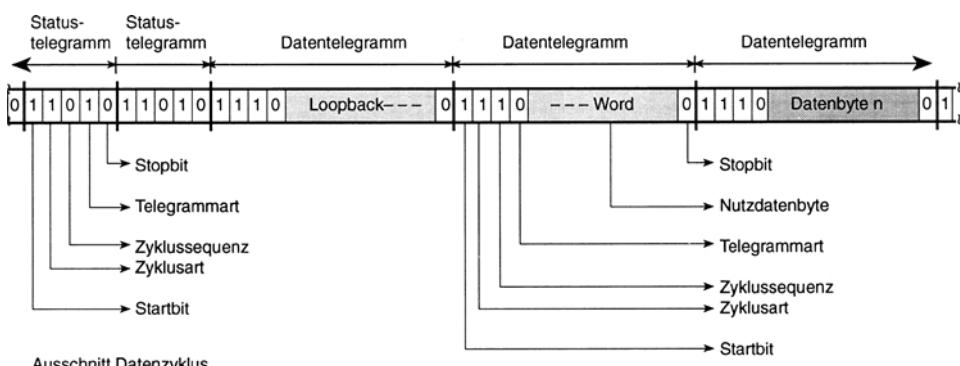


Abb. 4.25 Protokollaufbau

Header, der zusätzliche Informationen wie Rahmenende-Kennung, Funktionscode und Telegrammtyp enthält, ergänzt und als Block übertragen. Diese zusätzlichen Informationen im Blockheader müssen, anders als die reinen Datenbits, die im Schieberegisterring laufen, gleichzeitig an alle Teilnehmer gesendet werden. Eine spezielle Steuerlogik in den Teilnehmern schaltet hierzu von einem quasi Multidrop-Mode, indem alle Teilnehmer gleichzeitig die vom Master ausgesendeten Header-Informationen empfangen, nach Erkennen des Headers in den Schieberegister-Mode um. Dieser Vorgang wiederholt sich jeweils nach der Übertragung von 8 Datenbits zyklisch.

In Übertragungspausen, in denen vom Master keine Daten ausgesendet werden, wird der laufende Datenstrom durch so genannte Statustelegramme aufgefüllt. Sie haben keinerlei Auswirkungen auf den Datenverlauf im Schieberegisterring und dienen lediglich zur Gewährleistung einer permanenten Aktivität im Bussystem. Bleibt diese Datenaktivität für einen Zeitraum $> 20\text{ ms}$ aus, so wird das von allen Teilnehmern als Unterbrechung des Systems interpretiert. Die Teilnehmer werden hierdurch in einen definierten und sicheren Reset-Zustand geschaltet. Das heißt bei einer Unterbrechung des Systems oder einem Ausfall der Masterbaugruppe, werden alle Geräte innerhalb kürzester Zeit in den sicheren Zustand geschaltet.

Das Ringsystem bietet neben dem CRC-Check eine Vielzahl weiterer ergänzender Möglichkeiten zur Datensicherung. Das ausgesendete Loopback-Wort, das in seiner Datenstruktur ständig automatisch vom Master inkrementiert wird, gewährleistet eine sehr sichere Aussage über die Vollständigkeit des Ringsystems und die fehlerfreie Datenübertragung. Ein ständiger Vergleich zwischen ausgegebenen und einlaufenden Taktten, ebenso wie der Vergleich der ausgegebenen und zurücklaufenden Header-Informationen, ergänzt diesen Ringüberwachungsmechanismus. In den einzelnen Teilnehmern wird jeder Datenblock auf Start- oder Stopbitverletzungen überwacht. Fehlsynchronisationen der asynchronen Schnittstellenabtastung können so sehr schnell erkannt und behoben werden. Zusammen mit dem, den gesamten Datenstrom überwachenden CRC-Check ergibt sich so ein sehr einfaches, effizientes und sicheres Datenübertragungsprotokoll.

Die Datenzuweisung aus dem Summenrahmentelegramm an die einzelnen Teilnehmer, auf Basis des Schieberegisterrings, ermöglicht den Wegfall einer manuellen Teilnehmeradressierung. Alle Interbus-Busstationen sind für das zyklische E/A-Protokoll durch ihre physikalische Lage eindeutig adressiert. Zusätzlich zu dieser automatischen physikalischen Adressierung der Bus-Teilnehmer, die im Servicefall einen entscheidenden Vorteil gegenüber anderen Systemen bietet, kann an zentraler Stelle im Bus-Master eine wahlfreie logische Adressierung der Bus-Teilnehmer durch die einfache Erstellung einer Adresszuweisungsliste erfolgen. Hierdurch werden die vom Anwendungsprogramm verwendeten Adressen der Bus-Teilnehmer von der jeweiligen physikalischen Lage im System unabhängig. Das problemlose Entfernen und Hinzufügen von Teilnehmern im Ring ohne Veränderung der bisherigen Adressierung ist dadurch möglich.

Die Ermittlung der Telegrammlänge, d. h. der Schieberegistertakte und des Telegrammaufbaus, erfolgt automatisch durch Managementfunktionen des Netzwerks. Hierzu kann in einem speziellen Konfigurations- und Diagnosemode von jedem Teilnehmer eine Information über Art und Anzahl seiner E/A-Daten abgefragt werden. Diese Teilnehmer-

Identifikation wird für das Interbus-System an zentraler Stelle festgelegt und gepflegt. Der Bus-Master verfügt so nach dem Hochlaufen des Systems über ein genaues Abbild der System-Topologie. Hierdurch sind Management und Diagnosefunktionen, wie das An- und Abschalten von Bussegmenten und die damit verbundene Veränderung des Summenrahmen-Telegrammaufbaus, ohne weitere Eingriffe des Netzwerkbenutzers möglich.

4.2.3.4 Anwendungsschnittstelle

Interbus weist auch für den Zugriff des Anwendungsprogramms auf Netzwerkdaten eine hybride Struktur, getrennt nach Prozessdaten und Parametern, auf.

Für die einfachen, transparenten Prozessdaten realisiert das Interbus-System eine prozessabbildende Darstellung (Abb. 4.26). Ein zyklisch ablaufendes Programm im Bus-Master aktualisiert ständig die Prozessdaten und stellt sie dem Anwender im E/A-Bereich der Steuerung als Prozessabbild zur Verfügung. Durch diesen direkten Speicherzugriff werden zeitaufwändige Dienstzugangsprozeduren vermieden, die ansonsten die Echtzeit-eigenschaften des Protokolls für Prozessdaten massiv verschlechtern würden. Für den Benutzer des Interbus-Systems stellt sich bei dem Zugriff auf Prozessdaten kein Unterschied zwischen der seriellen Verkabelung und der traditionellen Parallelverkabelung dar. Der Anwender muss nicht umdenken und ist nicht genötigt, sich in komplexe Kommunikationsstrukturen einzuarbeiten.

In Bezug auf die Parameterkommunikation lässt sich dieses Verfahren nicht anwenden. Für eine offene Kommunikation sind standardisierte universelle, den Anforderungen sämtlicher Geräte und Anwendungen gerecht werdende Kommunikationsdienste erforderlich.

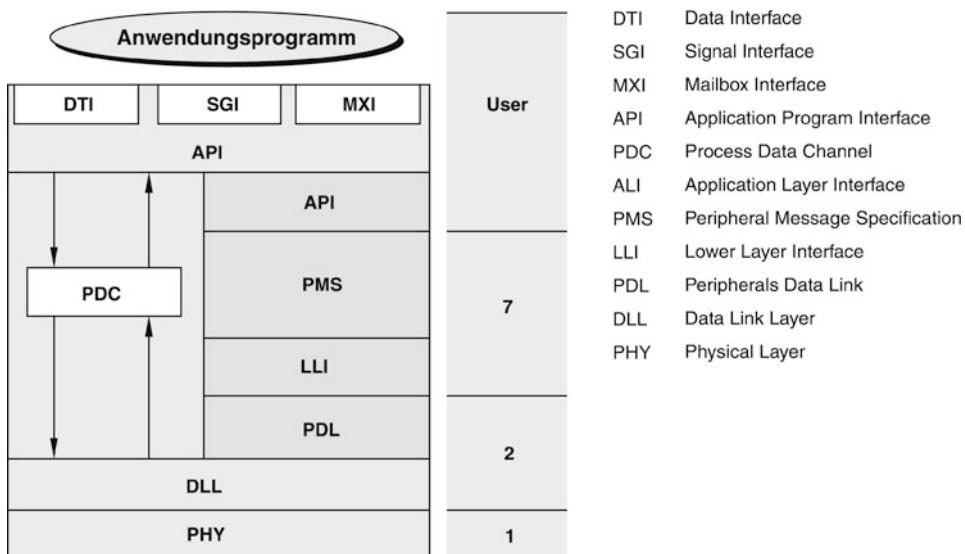


Abb. 4.26 Interbus Protokoll-Stack

Hierzu wurde für das Interbus-System eine kompatible Untermenge der in der DIN 19 245, Teil 2, definierten Kommunikationsdienste (FMS) gebildet, die als PMS (Peripheral Message Specification) bezeichnet wird.

Die derzeit 14 PMS-Dienste erlauben eine einfache Kommunikation mit intelligenten Prozessgeräten. Es stehen u. a. Dienste für den Aufbau und die Überwachung von Kommunikationsverbindungen (Context Management), für das Lesen und Schreiben von Variablen oder Parametern (Read/Write) sowie für das Starten von Programmen (Start/Stopp) zur Verfügung. Eine Erweiterung der Dienstmenge für bestimmte Geräte ist problemlos möglich. Unter Berücksichtigung des ständig ablaufenden zyklischen I/O-Protokolls kann ein im Interbus-Netzwerk arbeitendes Gerät auf PMS-Ebene sowohl Dienste ausführen (Server) wie auch selbstständig Dienste absetzen (Client).

Der Zugriff auf die PMS-Kommunikationsdienste aus dem Anwendungsprogramm erfolgt nicht, wie bei den Prozessdaten, über parallele E/As, sondern wird durch so genannte Funktionsbausteine (Hantierungsbausteine) über eine Nachrichten-Schnittstelle (MXI: Mailbox-Interface) realisiert. Zusätzlich hierzu gibt es noch die Möglichkeit der Vordefinition von Kommunikationsdiensten bei der Projektierung des Systems, die dann zur Laufzeit des Systems als abgespeicherte Aktionsblöcke nur noch über Bitbefehle aktiviert werden. Diese einfache Schnittstelle (SGI: Signal-Interface) entbindet den Anwender von der Erstellung aufwändiger Hantierungsrichtinen zum Austausch der Kommunikationsdaten und reduziert die Nutzung von Kommunikationsdiensten auf das Hantieren von logischen Verknüpfungen. Das Interbus-System wird heute in ca. 350 000 Systemen mit 3,8 Mio. vernetzten Feldgeräten weltweit eingesetzt. Das Protokoll ist als nationale (DIN 19258) und europäische Norm (EN 61158) Anfang 2000 in die internationale Norm IEC 61158 aufgenommen worden.

4.2.4 Modbus-RTU und Modbus-ASCII

Das Protokoll dieser beiden Modbus-Varianten ist kompatibel. Es verkehrt bei beiden ein Master mit bis zu 255 Slaves. Abb. 4.27 zeigt die ganze Modbus-Familie.

Schicht	ISO/OSI	MB-PLUS	MB-RTU	MB-ASCII	MB-TCP
7	Application	MODBUS Application Layer			
6	Presentation	–	–	–	–
5	Session	–	–	–	–
4	Transport	–	–	–	TCP
3	Network	–	–	–	IP/UDP
2	Data Link	Token Passing	Master/Slave		Client/Server
1	Physical	RS 485	RS232/RS485		Ethernet100MBd

Abb. 4.27 Die verschiedenen Varianten des Modbus MB im ISO/OSI-Modell. RTU – Remote Terminal Unit, Client/Server – Master/Slave, UDP – User Data Protocol

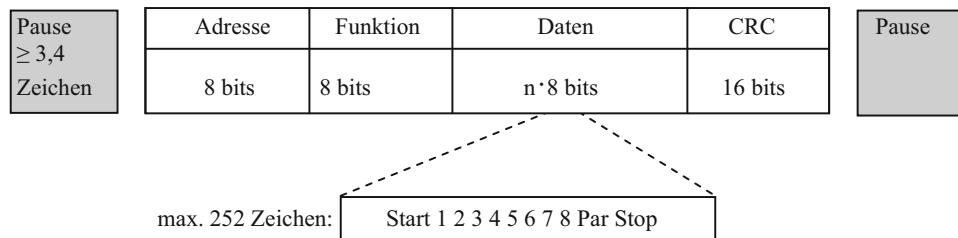


Abb. 4.28 Das Modbus-RTU-Telegramm (frame). Adresse – Slaveadresse (1 – 247 dez.), CRC – Prüfcode (Cyclic Redundancy Check)

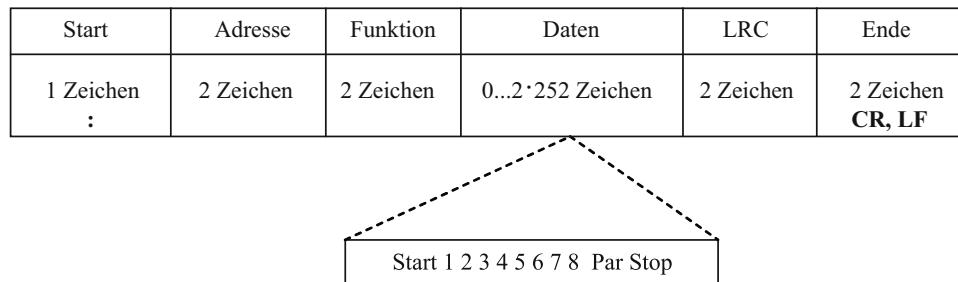


Abb. 4.29 Das Modbus-ASCII-Telegramm. Mit dem Doppelpunkt ASCII 3Ah beginnt der Empfang, mit „CR“ (Carriage Return ASCII 0Dh) und „LF“ (Line Feed ASCII 0Ah) hört er auf. LRC – Longitudinal Redundancy Check (Telegramm-Prüfzeichen)
1 Zeichen = 1 ASCII-Zeichen = 4 Bit

Modbus-RTU (Remote Terminal Unit) Hier werden die Daten direkt in binärer Form an z. B. eine SPS übertragen. Dieses ist die Standard-Übertragungsart des Modbus. Jedes Telegramm ist vom anderen durch eine Sendepause P getrennt (Abb. 4.28), die mindestens 3,5 Zeichen lang sein muss. Ist sie kürzer, verwirft der Empfänger die Nachricht.

Modbus-ASCII Es werden keine Binärfolgen übertragen, sondern ASCII-Zeichen. Damit ist das Telegramm direkt für den Menschen lesbar. Allerdings ist die Datenrate dadurch geringer als bei Modbus-RTU. Jedes Datenbyte benötigt zwei ASCII-Zeichen, also Hex-Zeichen (0...9, A...F) (Abb. 4.29).

Beispiel

Das Byte 0101 1011 ist hexadezimal 5Bhex. ASCII-codiert ist Zeichen „5“ = 35hex = 0011 0101 und Zeichen „B“ = 42hex = 0100 0010. Die Datenrate ist also bei gleicher Baudrate halb so groß wie bei der Binärübertragung durch Modbus-RTU.

4.2.5 LON

LON oder *Local Operating Network* ist ein Synonym für die LonWorks-Technik. Sie beschränkt sich nicht auf ein Übertragungsprotokoll oder Bussystem – wofür LON im engeren Sinne verwendet wird –, sondern stellt eine offene und kaskadierbare Plattform für verteilte, ereignisgesteuerte Automatisierungssysteme in unterschiedlichen Branchen dar.

Die Informationsübertragung auf der Grundlage des LonWorks-Protokolls zwischen den intelligenten LON-Geräten als Teilnehmer eines Netzwerkes ist deshalb stets im Zusammenhang mit der Informationsverarbeitung in diesen zu sehen. Damit steht die LonWorks-Technik in der Tradition des Bitbus, der ebenfalls als Basis für ein echt verteiltes Automatisierungssystem konzipiert wurde. Weitere hervorragende Merkmale der LonWorks-Technik sind die grundsätzliche Interoperabilität von LON-Geräten auf Applikationsebene, die Vielzahl der nutzbaren Medien und Transceiver, ein kollisionsvermeidendes stochastisches Zugriffsverfahren, die Verfügbarkeit eines leistungsfähigen Netzwerk-Betriebssystem (LNS *LonWorks Network Service*) als Basis für Management-Tools sowie zahlreiche Infrastrukturkomponenten für die Netzwerkgestaltung einschließlich der unmittelbaren Anbindung an TCP/IP-basierte Netzwerke.

4.2.5.1 LON-Gerät

Ein LON-Gerät ist vergleichbar mit einer kompakten SPS, die über LON in ein verteiltes Automatisierungssystem integriert wird (Abb. 4.30a). Das dominierende Bauelement ist ein hochintegrierter Schaltkreis, der Neuron-Chip genannt wird und den Mikrorechnerkern darstellt. Der Anschluss externer Hardware (z. B. Sensoren, Aktoren, Hostrechner) erfolgt vorzugsweise über eine programmierbare Anwendungsschnittstelle des Neuron-Chips mit einer zusätzlichen anwendungsspezifischen Elektronik. Die Kommunikationschnittstelle steuert einen Transceiver zur physikalischen Ankopplung an das Medium. Taktgenerator und Stromversorgung sind weitere Komponenten des LON-Gerätes, wobei die Energieversorgung extern oder – im Fall einer Zweidraht-Leitung – direkt über das Übertragungsmedium erfolgen kann.

Neuron-Chip

Der Neuron-Chip stellt eine 8-bit-Mikrorechnerstruktur mit drei Prozessoren dar (Abb. 4.30b). Während die Prozessoren 1 (MAC-CPU) und 2 (Network-CPU) die Kommunikationsaufgaben realisieren, wird durch den Prozessor 3 (Application-CPU) das Anwendungsprogramm für die lokalen Funktionen abgearbeitet. Die drei identischen Prozessoren besitzen eigene Register, nutzen aber gemeinsam eine Verarbeitungseinheit und die Speicher über den Mikrorechnerbus. Grundsätzlich werden zwei Typen von Neuron-Chips unterschieden: Der Typ 3120 mit integrierten Speicherressourcen und der Typ 3150 mit externer Speichererweiterung und ausgelagertem Betriebssystem. Die nachfolgende Tabelle enthält eine Auswahl verschiedener Neuron-Chips mit den wichtigsten Hardwarekomponenten. Auf dem Neuron-Chip 3150 mit 42 KByte extern verfügbarem Speicher

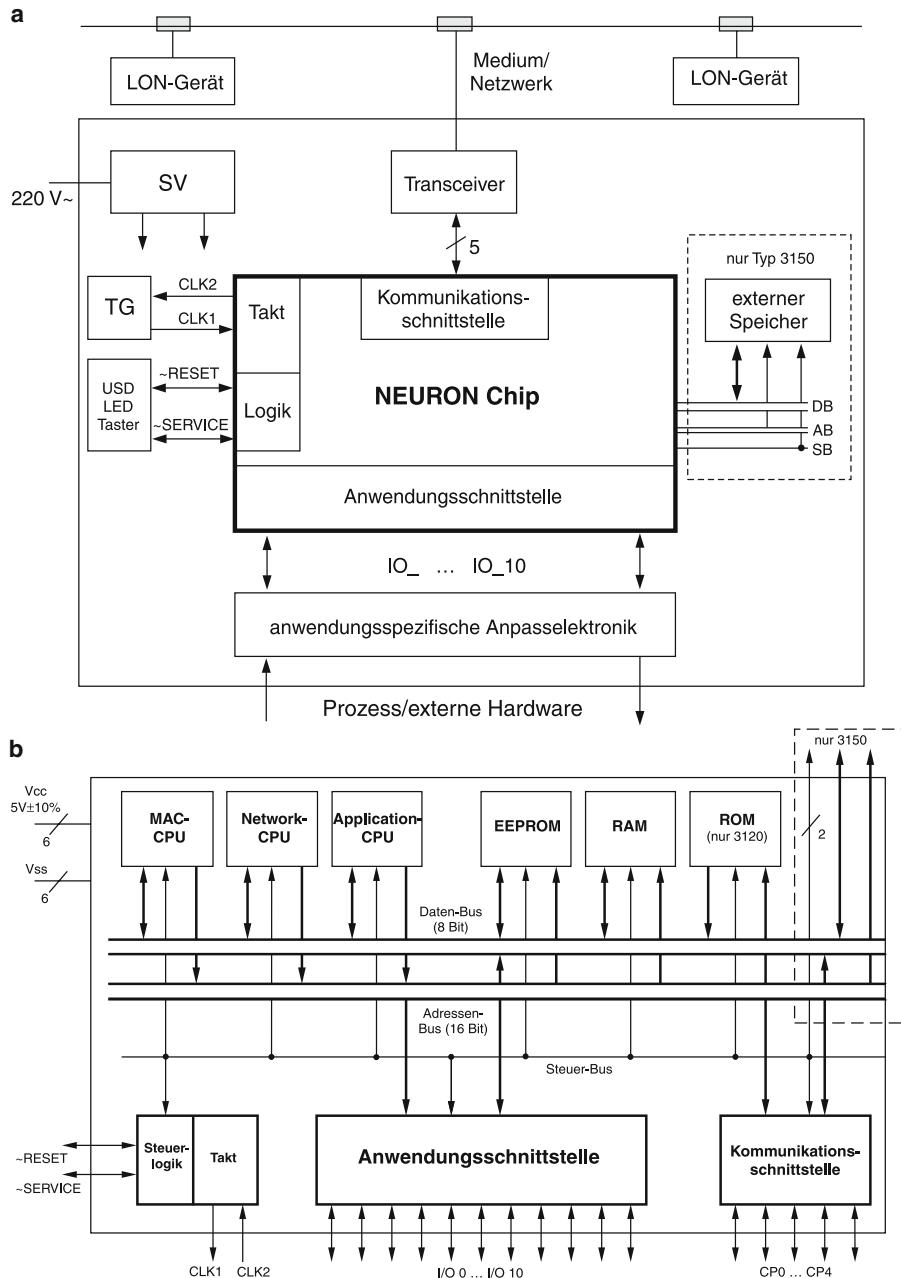


Abb. 4.30 Typischer Aufbau eines LON-Gerätes. **a** Gesamtschaltung (USD Unterspannungsdetektor, LED Leuchtdiode, SV Stromversorgung, TG Taktgenerator, AB Adressenbus, DB Datenbus, SB Steuerbus), **b** Hardwarestruktur der Neuron-Chips

für das Anwendungsprogramm einschließlich RAM sind anspruchsvollere Aufgaben lösbar. Der externe Zugriff auf den chipinternen Bus lässt außerdem eine Erweiterung mit peripherer Technik (Ein-/Ausgabe-Komponenten) oder die Ankopplung eines Hosts zu. Die Kaskadierbarkeit der Leistungsfähigkeit von LON-Geräten und damit die Anpassung an verschiedene Aufgabenumfänge ist eine wichtige Voraussetzung für flexible Lösungsansätze in verteilten Automatisierungssystemen.

Wichtig für die Inbetriebnahme und die Diagnose ist der bidirektionale Service-Pin. Er wird zur Ansteuerung der Service-LED (Anzeige verschiedener Modi des LON-Gerätes) und zum Erfassen des Service-Tasters genutzt, dessen Aktivierung zum Absenden einer Nachricht des LON-Gerätes mit einer weltweit einmalig vergebenen 48-bit-Zahl pro Neuron-Chip an das Netzwerk führt. Dadurch kann unter Nutzung entsprechender Management-Tools das LON-Gerät räumlich exakt identifiziert werden.

Hardware-Komponenten	Neuron-Chip						3150B1AF	
	3120							
	... B1AM	... E1M	... A20U	... FE3M	... FE5M			
Anzahl CPU		3	3	3	3	3	3	
EEPROM	kByte	0,5	1	1	2	3	0,5	
RAM	kByte	1	1	1	2	4	2	
ROM	kByte	10	10	16	16	16	–	
Externes Speicherinterface		–	–	–	–	–	Ja	
Timer/Counter Kanäle		2	2	2	2	2	2	
ADU Kanäle		–	–	3	–	3	–	
Max. Taktfrequenz, MHz		10	10	20	20	20	10	

Funktionsklasse	Unterstützte Funktionen (Auswahl)
Direkte Objekte	Binärer Eingang/Ausgang, byteweise Eingabe/Ausgabe
Zähler-Zeitgeber-Input-Objekte	Messung von Pulszeiten, Infrarotdecoder, Pulsdauermessung, Periodenmessung für Pulsfolge, Ereignismessung für Zeitfenster, Zähler
Zähler-Zeitgeber-Output-Objekte	Frequenzteiler, Pulsgenerator, Einzelpuls, Pulsfolge, tastverhältnis-programmierte Pulsgenerierung
Parallele Objekte	8 Bit-bidirektonaler Port in unterschiedlichen Varianten (bis 3,3 Mbit/s)
Serielle Objekte	Bitshift Input/Output, I ² C-Bus, Magnetkartenleser, Touch I/O u. a.

Anwendungsschnittstelle

Die Anwendungsschnittstelle wird hardwareseitig durch die elf Anschlüsse IO_0 bis IO_10 (IO steht für Input/Output) repräsentiert und mit dem Systemtakt abgefragt und angesteuert. Von besonderer Bedeutung für die Geräteentwicklung sind den Pins

zuordenbare elementare firmwaregestützte Funktionen für die Eingabe und Ausgabe (E/A-Funktionen) von Signalen. Diese E/A-Funktionen – insgesamt stehen 34 instanzierbare E/A-Objekte zur Verfügung – decken ein breites Spektrum von oft benötigten Funktionen in der Industrie- und Gebäudeautomation ab (siehe vorstehende Übersicht). Sie erlauben eine Minimierung der externen Anpasselektronik und eine effiziente Programmierung.

Für die Realisierung genauer zeitabhängiger Funktionen werden zwei eigenständige Zeitgeber-/Zähler-Schaltungen genutzt, die wegen ihrer Hardwarerealisierung keine Prozessorzeit benötigen. Für leistungsfähigere LON-Geräte ist die Ankopplung eines Hosts u. a. über die parallele E/A-Funktion der Anwendungsschnittstelle mit einem Datendurchsatz von 3,3 MByte/s (Systemtakt 10 MHz) möglich.

Betriebssystem und Anwendungsprogramm

Im Neuron-Chip mit drei Prozessoren erfolgt ein komplexes Zusammenwirken von drei „Betriebssystemen“ mit einer partiell zeitlich parallelen Bearbeitung der Aufgaben. Die Kenntnis des „Betriebssystems“ der Application-CPU ist für das Verständnis der Funktionsweise des LON-Gerätes von besonderem Interesse. Im Unterschied zur zyklischen Abarbeitung von Befehlssequenzen des Anwendungsprogramms in der klassischen SPS erfolgt in deren Hauptschleife typischerweise eine ereignisgesteuerte Abarbeitung von Tasks durch einen Scheduler. Hierbei wird zwischen priorisierten Ereignissen und nicht-priorisierten Ereignissen unterschieden. Nur wenn das programmierte Ereignis eintrifft, wird die dazugehörige Task ausgeführt:

```
when (Ereignis)
{
    Task
}.
```

Die ereignisgetriebene und taskorientierte Programmbearbeitung wird durch zahlreiche vordefinierte Ereignisse unterstützt. Neu eintreffende Telegramme stellen ebenfalls Ereignisse dar.

Die Programmiersprache lehnt sich weitgehend an die weltweit eingeführten Hochsprache ANSI C an und wird mit NEURON C bezeichnet. Sie ist gegenüber C „abgerüstet“, enthält aber andererseits zusätzliche Möglichkeiten, die unmittelbar auf die Hard- und Firmwarespezifika der LON-Geräte aufsetzen. Alternativ zur Programmierung in NEURON C bieten einige Hersteller Tools mit den SPS-Fachsprachen nach EN DIN 61131 oder proprietären grafischen Sprachen an. Diese freiprogrammierbaren, teilweise modularen LON-Geräte mit zusätzlichem Prozessor, erreichen die Leistungsfähigkeit von SPSEN. Typischerweise sind die LON-Geräte bereits für bestimmte Aufgaben programmiert und müssen lediglich über den Bus parametriert (oder programmiert) werden. Eine gesonderte Programmierschnittstelle fehlt deshalb. Typisch ist auch die elektronische Bereitstellung von Plug-Ins durch die Hersteller.

4.2.5.2 LonWorks-Protokoll

Dateninterpretation in LonWorks-Anwendungen

Eine der wichtigsten Anwendungsgebiete der LonWorks-Technik ist die Gebäudeautomation. Zunehmend müssen hier ganzheitliche Strategien der Gebäudebewirtschaftung durch gewerkeübergreifende Lösungen realisiert werden. Die Vielzahl der Gewerke und Datenpunkte (und damit LON-Geräten) in größeren Zweck- oder Bürobauten bzw. Liegenschaften erfordern neben der ereignisgetriebenen Kommunikation der LON-Geräte als Multi-Master-System (CSMA-Zugriffsverfahren) und einem großen, logisch adressierbaren und hierarchisch strukturierten Adressenraum vor allem eine eindeutige Dateninterpretation in den kommunizierenden Anwendungsprogrammen. Diese notwendige „Basis“-Interoperabilität, die Fehlinterpretationen von empfangenen Daten in der Applikation ausschließt, wurde im LonWorks-Protokoll (auch mit LonTalk-Protokoll bezeichnet) mit dem Konzept der standardisierten Netzwerk-Variablen (SNVTs *Standard Network Variables Types*) geschaffen und stellt ein herausragendes Merkmal gegenüber anderen offenen Protokollen dar. SNVT sind physikalische und andere technische Größen, deren Einheit, Wertebereich und Auflösung (Schritt) auf der Grundlage allgemein anerkannter internationaler Konventionen festgeschrieben und mit einer Identifikation versehen sind. Es ist somit für die eindeutige Interpretation der Daten im Anwendungsprogramm des Empfängers ausreichend, den Wert der globalen Variablen (SNVT) mit einer eindeutigen Kommunikationsbeziehung über das Netzwerk zu übertragen. Auszugsweise sind einige SNVT in der folgenden Tabelle aufgeführt.

SNVT-Name	Größe	Einheit	Wertebereich	Schritt	Nr.
SNVT_char_ascii	ASCII-Zeichen	Zeichen	0...255		7
SNVT_count	Ereignisse	Anzahl	0–65535	1	8
SNVT_switch	Status/Wert	(Struktur)	(zwei Elemente)		95
SNVT_lev_disc	Zustand diskret	(enum)	(fünf Werte)		22
SNVT_amp	Strom	A	–3276,8–+3276,7	0,1	1
SNVT_lux	Lichtstärke	lux	0–65 535	1	79
SNVT_temp_p	Temperatur	°C	–273,17–+327,66	0,01	105
SNVT_lev_cont	Niveau analog	%	0–100	0,5	21
SNVT_time_stamp	Datums- und Zeitan-gabe	(Struktur)	Jahr, Monat, Tag, h, min, s	1 s	84

Die Fortschreibung dieser Tabelle erfolgt mit Aktualisierungen einer so genannten Master list (zurzeit 11. Ausgabe mit über 160 SNVT), die von der weltweit organisierten LONMARK Interoperability Association veröffentlicht wird. Als SNVT werden neben einfachen auch strukturierte und solche vom Aufzählungstyp verwendet. Alle elementaren SNVT werden – wie auch andere im Anwendungsprogramm deklarierte Variablen – auf wenige elementare Datentypen abgebildet.

Die Deklaration einer globalen Variablen im Anwendungsprogramm als SNVT beinhaltet mindestens die Richtung (Eingang oder Ausgang), die SNVT selbst (Typ) und den benutzerdefinierten Namen der Variablen:

network input|output Typ Variablenname [=Initialwert].

Optional kann z. B. ein Initialwert oder für eine Netzwerk-Ausgangs-Variable auch eine Priorität über eine Verknüpfungs-Information angegeben werden:

network output Typ [bind_info (priority)] Variablenname.

Letztere weist der SNVT eine LON-gerätespezifische Priorität zu, mit der der Netzwerk-Teilnehmer Zugang zum Übertragungsmedium im Vergleich mit anderen Teilnehmern bekommt.

Über die SNVT werden die logischen (informationellen) Beziehungen zwischen den lokalen Funktionen in den LON-Geräten realisiert. Für die Beschreibung dieser Zusammenhänge (z. B. für die Projektierung) ist somit die Angabe des Variablenamens (aus den Applikationen entsprechend der Deklaration), der SNVT und der Übertragungsrichtung ausreichend. Abb. 4.31a zeigt grafisch eine elementare logische Verbindung zweier LON-Geräte bzw. Applikationen. *SNVT_switch* steht für eine beliebige SNVT.

Mit den Vorsilben *nvi* bzw. *nvo* wird (üblicherweise) im Variablenamen die Richtung der Übertragung (ankommende oder ausgehende Variable) angegeben. Für die Kommunikation werden lokale Funktionen durch typisierte Objekte repräsentiert (vgl. Abschn. 4.2.5.3). Weitere zulässige elementare logische Verbindungen sind u. a. die 1-auf-n- und die n-auf-1-Verbindungen. Die logischen Kopplungen werden im jeweiligen LON-Gerät in den Adressen- und Netzwerk-Variablen-Tabellen verwaltet. Insgesamt können infolge der begrenzten Speicherkapazität pro LON-Gerät 62 logische Verbindungen genutzt werden. Die Speicherung der Tabellendaten erfolgt wegen möglicher Änderungen in der Inbetriebnahmephase oder im laufenden Betrieb im spannungsausfallsicheren EEPROM.

Telegrammarten

In einem funktionierenden LonWorks-Netzwerk wird der Informationsaustausch durch die Versendung von Nachrichten mit den Werten von SNVT geprägt. Dieses Anwendungs-telegramm ist der Standard für interoperable Lösungen und wird auch mit *Implicite Message* bezeichnet. Weitere Telegrammtypen – *Explizite Messages* – können für unterschiedliche Zwecke genutzt werden. (Dazu zählt z. B. auch das Service-Pin-Telegramm, mit dem die LON-Geräteidentifikation – die Neuron-Chip-ID – bei Aktivierung des Service-Pins übermittelt wird.) Die Kodierung der einzelnen Anwendungs-Telegrammtypen ist in Abb. 4.31b dargestellt und der nachfolgenden Tabelle zu entnehmen. Die Deklaration eigener Netzwerkvariablen ist – unter Aufgabe der Interoperabilität – auf der Grundlage der elementaren Datentypen möglich.

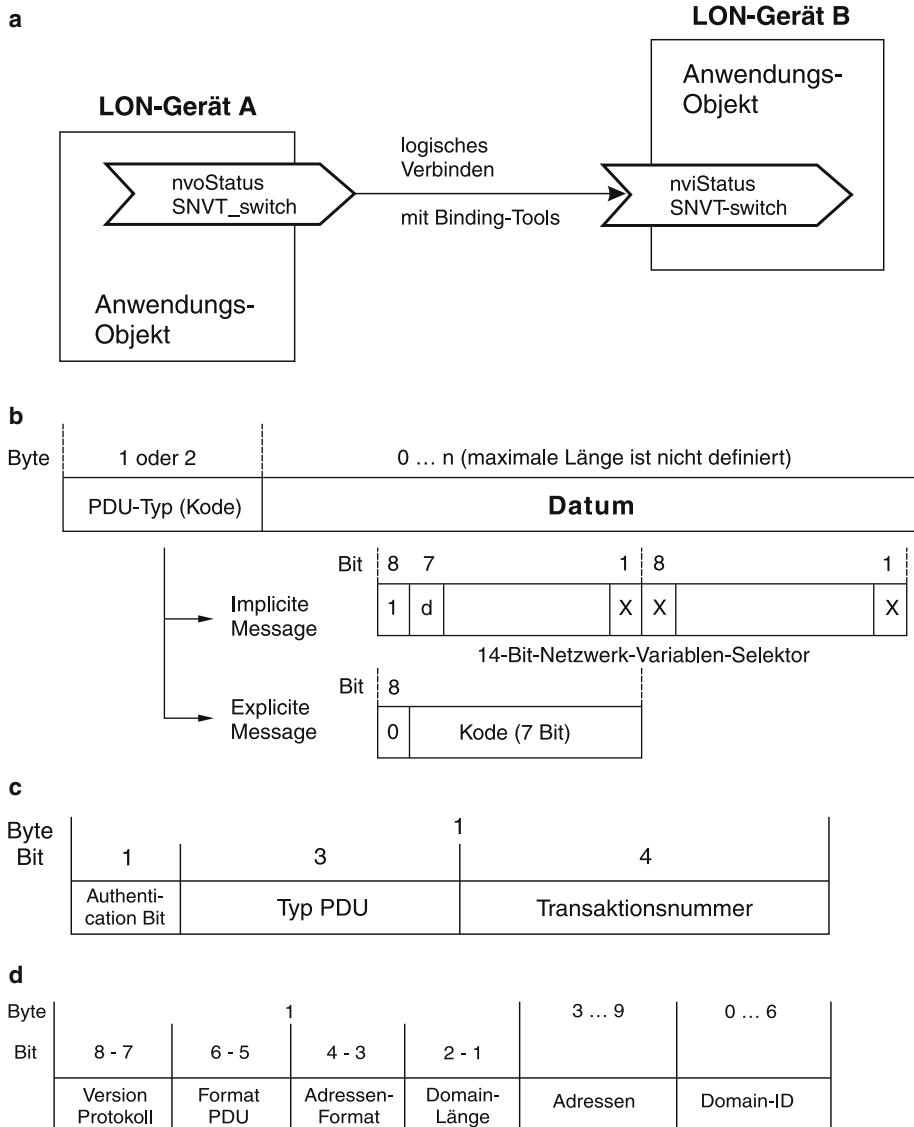


Abb. 4.31 Datenaustausch zwischen LON-Geräten. **a** Logische Verbindungen zwischen LON-Geräten, **b** Telegrammtyp-Kennzeichnung im Header der Application PDU, **c** Header der Transport PDU, **d** Header der Netzwerk PDU

Kodierung	Telegramm für ...	Kodebereich (hexadezimal)
0xxxxxx	allgemeine Anwendungen	00–3E
0100xxxx	Transport fremder Protokolldateneinheiten	40–4F
0101xxxx	Netzwerk-Diagnose	50–5F
011xxxxx	Netzwerk-Management	60–73
0111x1xx	Routerkonfiguration	74–7E
01111111	Service Pin Message	7F
1dxxxxxx xxxxxxxx	Übertragung von Netzwerkvariablen	80–FF plus 1 Byte

Das Standard-Telegramm wird mit einer 1 an der Position 8 (*MSB Most Significant Bit*) des ersten Bytes kodiert. Die Bitposition 7 markiert in diesem Fall eine eingehende ($d = 0$) oder abgehende ($d = 1$) SNVT. Die restlichen vierzehn Bitpositionen beider Bytes werden von dem Netzwerk-Variablen-Selektor zur Identifikation der Kommunikationsbeziehungen belegt. Daten (Wert der SNVT) und Header (vorangestellte Zusatzinformationen) bilden die Protokoll-Daten-Einheit (*PDU Protocoll Data Unit*) der Schicht 7 (*Application Layer*) nach dem OSI-Referenzmodell für die Kommunikation und ist faktisch mit der Schicht 6 (*Presentation Layer*) zusammengefasst. Eine *Implicite Message*, d. h. basierend auf einer Netzwerkvariablen, wird bei Werteänderung sofort vom LON-Gerät an das Netzwerk propagiert, ohne dass es eines gesonderten Auftrages im Anwendungsprogramm bedarf.

Alle anderen Formate von Anwendungstelegramm (*Application-PDUs*) werden in einem Byte mit einer 0 an Bitposition 8 des Headers kodiert. Dies sind anwenderdefinierte Telegramme (allgemeine Telegramme), die auf spezifische Anforderungen zugeschnitten oder beispielsweise in das Netzwerk-Management und die -Diagnose eingebunden sind. Als Nicht-Standard-Telegramme (*Explizite Messages*) werden sie im Anwendungsprogramm unter Nutzung spezifischer Befehle des Application Layer Interfaces programmiert. Die mit den Daten übertragenen Informationen sind individuellen Nutzungen vorbehalten und stellen keine Basis für offene interoperable Anwendungen dar.

Dienste für die Telegrammübertragung

In der folgenden Tabelle sind die unterstützten Dienste für die Übertragung von beliebigen Anwendungstelegrammen an einen adressierten Empfänger, die der *Transport Layer* zur Verfügung stellt, aufgeführt. Der bestätigte Dienst (ACKD) ist voreingestellt. Die Zahl der Wiederholungen und die Zeit zwischen diesen sind parametrierbar.

Für die *Explicit Messages* wird zusätzlich ein Request-Response-Dienst zur Realisierung von Client-Server-Anwendungen zur Verfügung gestellt. Dieser Dienst ist auf der Sitzungs-Schicht (*Session Layer*) angesiedelt. Er wird u. a. auch für das – eher untypische – Pollen von Netzwerk-Variablen (z. B. durch einen LON-Knoten mit zentralen Leittechnik-Aufgaben) genutzt.

Dienst	Kode	Erläuterungen
Acknowledged Service	0	Gesicherte Übertragungsart durch Bestätigung des Empfängers. Bei fehlender Bestätigung: Wiederholung nach Wartezeit. Datenflusskontrolle durch Transaktionsnummer.
Unacknowledged Service	1	Einmaliges Versenden der Nachricht
Unacknowledged/ Repeated Service	2	Wiederholtes Versenden der Nachricht. Datenflusskontrolle durch Transaktionsnummer.

Ebene	Erläuterung
Domain	virtuelles Netzwerk, in dem die Kommunikation stattfindet
Teilnetz (Subnet)	<ul style="list-style-type: none"> – logische Zusammenfassung von Netzwerk-Teilnehmern – es stehen pro Domain 255 Teilnetz-Adressen zur Verfügung
LON-Gerät (Node)	<ul style="list-style-type: none"> – kleinste adressierbare physische Komponente des Netzwerkes – pro Subnet sind 127 Knoten adressierbar
Neuron-ID	weltweit einmalige Identifikationsnummer des Neuron-Chips (48-Bit-Adresse)
Gruppe	<ul style="list-style-type: none"> – Zuordnung von LON-Geräten zu einer logischen Gruppe – es sind 256 Gruppen adressierbar – pro Gruppe 63 LON-Geräte (für bestätigten Dienst) – ein LON-Gerät darf gleichzeitig 15 Gruppen zugeordnet werden
Gruppenmitglied	kleinste adressierbare Komponente einer logischen Gruppe

Die Überwachung des Telegrammflusses über eine laufende Nummer der Wiederholung (zur Vermeidung von Duplizierungen) wird mit Transaktionsnummer bezeichnet, ist für maximal 15 Wiederholungen (vier Bit) ausgelegt und steht als Dienst des *Transaction Control Sublayers* ebenfalls allen Anwendungstelegrammen zur Verfügung. Abb. 4.31c zeigt die Kodierung dieser Zusatzinformationen (Header der Transport-PDU) für den Telegramempfänger. Mit der Bitposition 8 kann die Authentifizierung des Senders angefordert werden. Der Empfänger initiiert diesen Beglaubigungsprozess, indem er eine 64 Bit-Zufallszahl mit einem speziellen Telegrammformat dem Sender übergibt und eine Verschlüsselung dieser Zahl nach einem bekannten Schlüssel erwartet. Über einen Vergleich der durch Empfänger und Sender verschlüsselten Zahlen wird die Zulässigkeit der Datenanforderung bestätigt. Das Authentication-Telegramm, das zum Versenden der 64-Bit-Zufallszahl und des Ergebnisses der Verschlüsselung dient, erfordert ein weiteres Telegrammformat (Authentication-PDU). Dieser Mechanismus stellt eine wichtige spezifische Möglichkeit des LonWorks-Protokolls im Vergleich zu anderen Feldbussystemen dar. Die Überprüfung der Identität des Senders wird ansonsten in kommerziellen und anderen Bereichen – entsprechend modifiziert – zur Übertragung vertrauenswürdiger Informationen genutzt.

Adressierung

Unter planerischem und Projektierungsaspekt kommt der Adressierung eine besondere praktische Relevanz zu. Vor allem die enorme Zahl von LON-Geräten als potentielle Netzwerk-Teilnehmer – Tausende in großen Zweckbauten oder Liegenschaften – und ihr differenziertes logisches (informationelles) Zusammenwirken erfordern Möglichkeiten einer effizienten und gleichermaßen überschaubaren Adressierung. Sie wird durch ein hierarchisches Adressensystem (siehe vorstehende Tabelle) realisiert.

Auf dieser Grundlage werden die nachfolgenden Adressenformate für die Telegrammempfänger zur Verfügung gestellt.

Nr.	Adressenkomponenten für Empfänger	Ziel	Länge [Bytes]
#0	Domain, Teilnetz (= 0)	Alle Knoten der Domain (Broadcast)	3
#0	Domain, Teilnetz	Alle Knoten des Teilnetzes (Multicast)	3
#1	Domain, Gruppe	Alle LON-Geräte einer Gruppe (Multicast)	3
#2a	Domain, Teilnetz, Knoten	Ein logischer Netzwerk-Teilnehmer im Teilnetz	4
#2b	Domain, Teilnetz, Knoten, Gruppe, Gruppenmitglied	Knoten einer Gruppe mit Bestätigung des Telegramms (Multicast)	6
#3	Domain, Neuron-ID	Spezieller (physischer) Knoten	9

Die Kodierung des Adressenformates erfolgt mit 3 Bit und ist in einem Header-Byte mit weiteren Informationen der Network-PDU enthalten (Abb. 4.31d). Je nach Adressenangabe sind drei (Adressierung einer logischen Gruppe) bis zu 15 (Adressierung über Neuron-ID und 6 Byte Domain-Adresse) Byte erforderlich. Die Absenderadresse wird grundsätzlich mit Teil-Netz (*Subnet*) und LON-Gerät (*Knoten* oder *Node*) angegeben. Wird die Domain-Länge mit 00 angegeben (Verwaltung des Netzwerkes in der Domain mit der ID 0), so werden für die Domain-ID keine weiteren Adressenbytes im Header benötigt. Die maximale Zahl von 32 385 adressierbaren Netzwerk-Teilnehmern ergibt sich aus der maximalen Zahl von 255 Teilnetzen pro Domain und maximal 127 LON-Geräten pro Teilnetz. Die logischen Gruppen sind Teilnetz-übergreifend definierbar.

Die unterschiedlichen Möglichkeiten der Adressierung sind unmittelbar mit der planungstechnisch anspruchsvollen Aufgabe der Netzwerkgestaltung zu sehen. Tatsächlich bilden logische Gruppen und Teilnetze wichtige Mittel zur Netzwerkstrukturierung. Hinzu kommen möglicherweise „Zwänge“ durch die notwendige Nutzung unterschiedlicher Übertragungsmedien. Dies erfordert bestimmte Neuronchip-basierte spezialisierte Netzwerk-Komponenten, die weitere Möglichkeiten der Netzwerkgestaltung bieten. Nachfolgend sind wichtige Infrastruktur-Komponenten aufgeführt.

Eine herausragende Rolle kommt hierbei dem Router zu. Selbstlernend oder als programmierbare LON-Geräte funktionieren sie als Telegrammfilter, die den Informationsaustausch auf das Teilnetz beschränken, in dem sich das adressierte LON-Gerät tatsächlich befindet. Dadurch kann eine sinnvolle Segmentierung des Netzwerkes nach bestimmten

Kriterien (räumliche Anordnung, Netzwerkbelastung, Gewerkeorientierung u. a.) umgesetzt werden. Unabdingbar sind Router bei der Nutzung unterschiedlicher Kanäle.

Komponente	Aufgabe
Kanal (Channel)	Physikalische Netzwerkeinheit (Medium) für Signalübertragung, Anschluss der LON-Geräte über Transceiver,
Repeater	Physikalische Verlängerung eines Kanals (Signalregenerierung), leitet alle gültigen Telegramme unselektiert weiter
Bridge	Verbindet zwei Kanäle im Subnet, leitet Telegramme in beiden Richtungen weiter
Router	Verbindet Teilnetze und ggf. unterschiedliche Kanäle (Medien), leitet Telegramme als selbstlernender oder konfigurierter Router selektiv weiter
Gateway	Verbindet unterschiedliche Netzwerke mit verschiedenen Protokollen (hier auch zwei Domains)

Buszugriffsverfahren

Das LonWorks-Protokoll nutzt das stochastische CSMA/CD. Es wird jedoch als ein predictiv p-persistent CSMA – optional prioritätsgesteuert – realisiert, um Kollisionen weitestgehend zu vermeiden (deshalb auch mit CSMA/Collision Avoidance bezeichnet).

Dies erfolgt durch drei Maßnahmen. Erstens erhält jedes LON-Gerät im Initialisierungsprozess eine Zufallszahl zwischen 1 und 16 zugeordnet. Bei niedriger Busbelastung versucht der Busteilnehmer nach einer erkannten Kollision nach entsprechend vielen Zeitquanten – mit β_2 -Zeiteinheiten bezeichnet – den erneuten Zugriff (p-persistent). Dadurch verringert sich die Wahrscheinlichkeit einer erneuten Kollision. Zweitens berechnet (predictiv) jeder LON-Teilnehmer mit der Telegrammzusammenstellung die zukünftige zusätzliche Netzwerkbelastung durch zu erwartende Bestätigungen der Empfänger (falls dieser Dienst vorgesehen ist). Dies kann bei einem Multicast (z. B. Adressierung aller Mitglieder einer Gruppe) durchaus erheblich sein. In Abhängigkeit von der zu erwartenden Netzwerkbelastung (mit *Backlog* bezeichnet) wird der Buszugriff durch alle Teilnehmer zeitlich um einen Faktor, proportional zur zu erwartenden Netzwerkbelastung, gedehnt. Insgesamt ist eine zeitliche Dehnung um $16 \times 63 = 1008 \beta_2$ -Zeiteinheiten möglich. Dadurch werden Kollisionen unwahrscheinlicher.

Drittens können bei besonders wichtigen Informationen (z. B. im Bereich Brandschutz, Personenbeförderung, Torensteuerung) Prioritäten im Bereich von 1 bis 127 für LON-Geräte vergeben werden, wobei die kleinere Zahl der höheren Priorität entspricht. Sie definiert die Zahl der Zeiteinheiten, die ein Netzwerk-Teilnehmer im Falle einer Kollision bis zu einem neuen Zugriff warten muss. Bei der Konkurrenz um die Ressource Bus wird sich dann wegen der kurzen Wartezeit das höher priorisierte LON-Gerät durchsetzen und ein priorisiertes Telegramm absenden können. Die Information zum *Backlog* (6 Bit) wird neben der Kennzeichnung als priorisiertes Telegramm (1 Bit) und der Möglichkeit einer Umschaltung auf einen alternativen Verbindungsweg bei erfolgloser Kommunikation (Alternate Path Bit) im Header der Link-PDU den Telegrammempfängern als Zusatz-

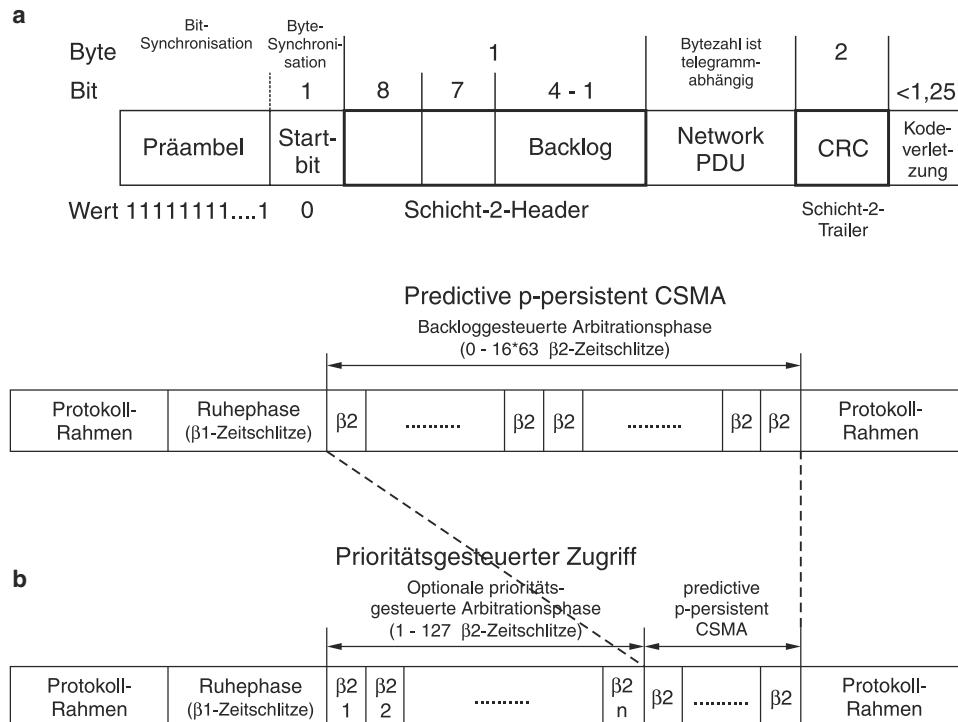


Abb. 4.32 Priorisierte Telegramme. **a** Rahmenformat des Link Layer, **b** Belastungs- und prioritäts-gesteuerter Buszugriff

information mitgeteilt. Das Rahmenformat dieser PDU ist in Abb. 4.32a einschließlich des Sicherungsfelds (2 Byte für CRC-Check nach CCITT-CRC-16-Standard) und der Bit- und Bytesynchronisation gezeigt. Eine gewollte Kodeverletzung (Kodeviolation) – hier des Differential-Manchester-Kodes – gilt als Beendigung des Rahmens und damit der Übertragung. Abb. 4.32b verdeutlicht die Zeitdehnung bei zu erwartender erhöhter Netzwerkbelastung und den priorisierten Teilnehmerzugriff.

Physikalische Kopplung

Bereits mit der konzeptionellen Entwicklung wurden zahlreiche Medien für die Datenübertragung vorgesehen (verdrillte Zweidrahtleitung oder TP, 230-V-AC-Leitung, Infrarot, Lichtwellenleiter, Koaxialkabel, Funkkanal), die über eine Vielzahl von Transceivern an ein LON-Gerät angekoppelt werden können. In der folgenden Tabelle sind für ausgewählte Transceiver wichtige Netzwerk-Kenngrößen zusammengestellt. Die Kommunikations-Schnittstelle des Neuron-Chips berücksichtigt durch verschiedene Modi die unterschiedliche Leistungsfähigkeit der Transceiver: Passive, aktive und intelligente (für die 230-V-AC-Leitung). Mit Ausnahme der letzteren erfolgt die Teilnehmer-

Synchronisation über die Taktrückgewinnung beim Empfänger durch einen Differential-Manchester-kodierten Bitstrom. Die Möglichkeit der frei wählbaren Verbindungen zwischen LON-Geräten als Netzwerk-Teilnehmer ist der Grund für den breiten Einsatz der Transceiver für freie Topologie FTT (*Free Topology Transceiver*) und der LPT (*Link Power Transceiver*) mit gleichzeitiger Übertragung von Energie (42 V DC) und Daten für das Medium TP.

Mit den verfügbaren Transceivern sind alle praktisch relevanten Topologien mit dem Medium TP realisierbar: Linie, Ring, Stern, freie Topologie, Baum.

Medium (Kanal)	Transceiver	ÜR ^a [kBd]	Teilnehmer- zahl ^b	Topologie	Netzausdehnung ^c [m]
TP	TPT/XF78	78	64	Linie	1400
	TPT/XF1250	1250	64	Linie	130
	FTT10-A (Power Line)	78	64	Frei	500
				Linie	2700
	LPT-10 (Link Power)	78	128	Frei	500
				Linie	2200
230-V-AC- Netz	PLT-22	5	Anwendungsspezifisch	Stromnetzspezifisch	Stromnetzspezifisch
Funkkanal	300 MHz	1,2	Anwendungsspezifisch	Anwendungsspezifisch	Anwendungsspezifisch
	450 MHz	4,8			
	900 MHz	39			

^a ÜR Übertragungsrate

^b ohne Repeater

^c leitungstypabhängig

LonWorks-Protokoll und OSI-Referenzmodell

Die gegebene (unvollständige) Übersicht der Funktionalität des LonWorks-Protokolls korrespondiert mit alle Schichten des OSI-Referenzmodells. Abb. 4.33a zeigt die prinzipielle Zuordnung. Die gesamte Protokollfunktionalität ist auf dem Neuron-Chip integriert, so dass Inkompatibilitäten auf diesen Ebenen ausgeschlossen sind. Ein typisches Telegramm mit einer Teilnehmer-zu-Teilnehmer-Übertragung für eine SNVT von 2 Byte Länge umfasst 13 Byte (Abb. 4.33b).

4.2.5.3 Funktionsprofile für LON-Geräte

Das LonWorks-Protokoll mit den SNVT bildet die – im Vergleich zu anderen Bussystemen wohl einzigartige – Grundlage der Interoperabilität zwischen den LON-Geräten. Da die überschaubare Anzahl von SNVT auf weltweit allgemein anerkannten Sachverhalten (physikalische Größen, technische Regeln) basieren, sind sie für alle Branchen mit einem naturwissenschaftlich determiniertem Hauptprozess anwendbar. Allerdings müssen für die „absolute“ Interoperabilität – die Interchangeability der Geräte auch konkurrierender Hersteller – die auszutauschenden Informationen auf einer genormten Funktionalität der entsprechenden Gerätelasse basieren. (Dieses Problem ist bei allen Feldbussystemen zu lösen.) Die genormte oder standardisierte Funktionalität von Gerätelassen bildet die

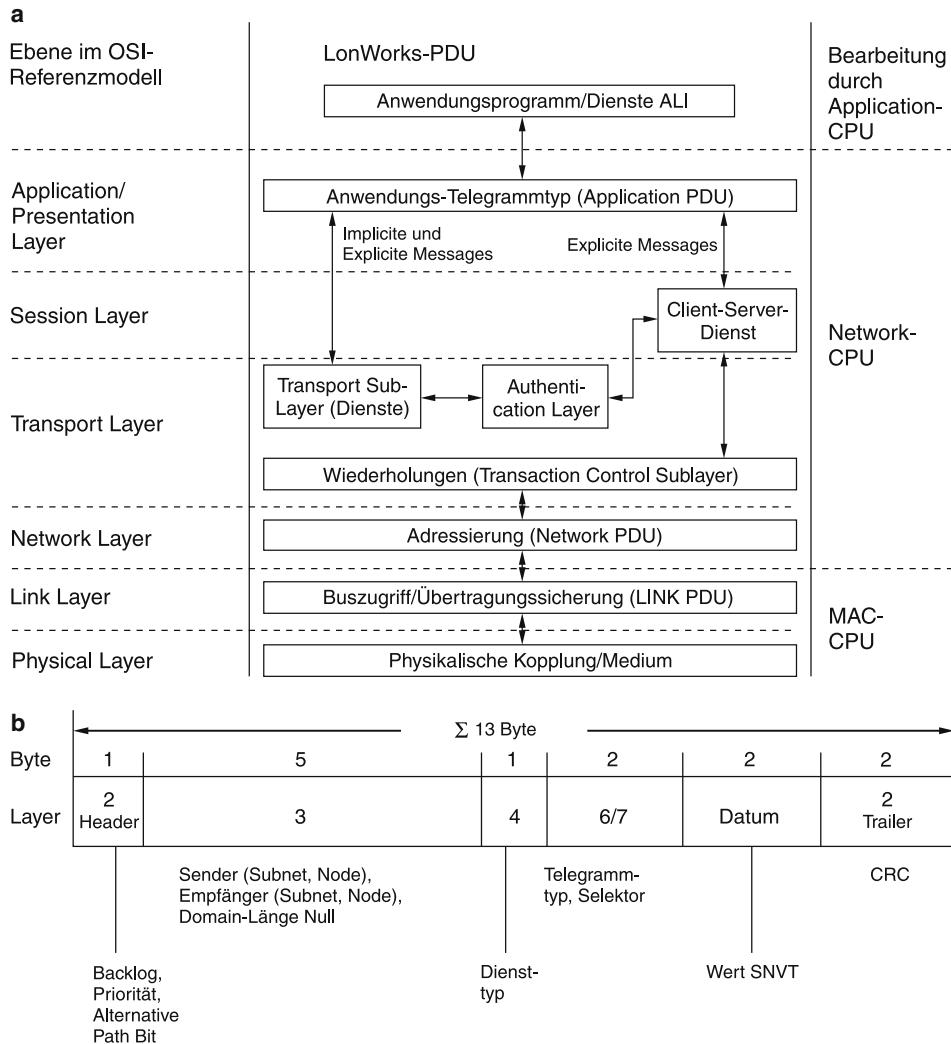


Abb. 4.33 LON im OSI-Referenzmodell und LON-Telegramm. **a** Zuordnung des LONWorks-Protokolls zu den Ebenen des OSI-Referenzmodells, **b** LON-Telegramm für z. B. 2 Byte-Standard-Netzwerk-Variable

Grundlage für die Interoperabilität von Geräten verschiedener Hersteller bis hin zu deren Austauschbarkeit.

Für die Standardisierung von Funktionsprofilen werden in der LonWorks-Technik generische Objekte mit obligatorischen und optionalen SNVT und speziellen Konfigurationsparametern (*SCPT Standard Configuration Property Types*) zugrunde gelegt. SCPT werden wiederum auf bekannte SNVT abgebildet und im EEPROM abgespeichert. Insgesamt

samt stehen fünf generische Objekte zur Verfügung, die – mit Ausnahme eines einzigen Knoten-Objektes pro LON-Gerät – in beliebiger Zahl auf einem LON-Gerät implementiert werden können, wenn die zulässigen Begrenzungen (wie z. B. die Anzahl von 62 für die verwalteten SNVT-basierten Nachrichten in einem LON-Gerät) nicht überschritten werden. Für bestimmte Geräteklassen werden Funktionsprofile von der LONMARK standardisiert und katalogisiert. In den Produktbeschreibungen ist es für die Projektierung hinreichend, nach LonMark zertifizierte Funktionsprofile und/oder die benötigten und generierten SNVT als logische Schnittstelle zum Netzwerk anzugeben.

4.2.5.4 LON-übergreifende Kommunikation

Wegen der Kaskadierbarkeit des Leistungsvermögens der LON-Geräte und des LON-Netzwerkes selbst sind fast beliebige autarke LON-basierte Bussysteme realisierbar. Für zentrale Aufgaben wie z. B. der Gebäudeleittechnik oder des Facility Managements in der Managementebene der Gebäudeautomation (Prozessvisualisierung, Datenarchivierung, Alarmmanagement, Fernwartung u. a.) muss ein aktuelles Prozessabbild verfügbar oder die Veränderung bestimmter Parameterwerte möglich sein. Die Nutzung des Intranets und/oder Internets bzw. offener Schnittstellen ist hierbei Stand der Technik. Für diese Aufgaben wird die „klassische“ Produktpalette der LON-Geräte durch zahlreiche spezielle Hard- und Softwarekomponenten ergänzt: Router mit Übergang zu TCP/IP-basierten Netzwerken, LON-Geräte mit integriertem WEB-Server oder OPC-Server. Auf dieser Grundlage sind offene, schlanke Netzwerk-Strukturen vom Sensor/Aktor bis zur entfernten beauftragten Wartungsfirma realisierbar. Für eine Produktübersicht sei auf die www.-Adressen im Literaturverzeichnis hingewiesen.

4.2.5.5 Netzwerk-Management und Tools

Die umfangreichen Aufgaben im Rahmen des Netzwerkmanagements (Systemintegration, Netzwerkdagnostik und -wartung u. a.) erfordern die Kenntnis von Informationen über Netzwerk-Parameter und die Möglichkeit zu deren Manipulation. Der Zugang zum Netzwerk erfolgt auf der Grundlage von Tools, die typischerweise auf den LNS (*LonWorks Network Service*) aufsetzen. LNS ist als ein Netzwerk-Betriebssystem für Client-Server-Anwendungen anzusehen. Es ermöglicht den datenkonsistenten gleichzeitigen Zugriff zu allen Daten und Diensten im Netzwerk über verschiedene Clients (Gebäudeleittechnik, Systemintegrator, Wartungstechniker). Voraussetzung dafür ist die LNS-Datenbank, in der Netzwerk- und Projektdaten in einem standardisierten Format abgelegt sind. Mit der Verfügbarkeit von LNS ist die Grundlage für einen freien Wettbewerb für die Tool-Entwicklung gegeben. Gleiche Projekte können mit verschiedenen Tools bearbeitet werden. Eine einheitliche Plug-In-Schnittstelle steht zur Entwicklung von gerätespezifischen Plug-Ins durch die Hersteller zur Verfügung. Der Zugriff auf den LNS-Server kann auch über das Intranet bzw. Internet erfolgen. Die Hardwareankopplung erfolgt über spezielles Neuron-Chip-basiertes Interface NSI (*Network Service Interface*).

4.2.6 CAN-basierende Netzwerke

4.2.6.1 Grundlagen

Das serielle Bussystem CAN (Controller Area Network) wurde ursprünglich von Mitarbeitern der Robert Bosch GmbH für die interne Vernetzung von elektronischen Steuergeräten (ECU = electronic control unit) im Kraftfahrzeug entwickelt. Die CAN-Norm (ISO 11898-1) definiert das Datenverbindungsprotokoll (Schicht 2 im siebenschichtigen OSI-Referenzmodell) und das Bit-Timing sowie die Fehlerbehandlung. Es ist eines der zuverlässigsten Netzwerke und hat eine äußerst geringe Restfehlerwahrscheinlichkeit. Das in den CAN-Telegrammen verwendete CRC-Polynom ermöglicht das sichere Erkennen von bis zu fünf beliebig verteilten Bitfehlern.

Die physikalische Übertragung ist in verschiedenen Normen festgelegt. Die am meisten genutzten Transceiver (Sender/Empfänger-Bausteine) entsprechen der Norm ISO 11898-2, die eine Übertragungsrate bis zu 1 Mbit/s erlaubt. Kompatible Transceiver mit Niedrigstrommodus (ISO 11898-5) und partiell Aufwachen (ISO 11898-6) wurden in den letzten Jahren entwickelt. Hervorstechendes Merkmal aller High-speed-Transceiver ist die Robustheit gegenüber elektromagnetischen Störungen. Die Treiberbausteine sind üblicherweise ebenso wie die CAN-Protokoll-Controller im erweiterten Temperaturbereich erhältlich. Es sind auch Transceiver mit integriertem Optokoppler verfügbar.

Darüber hinaus gibt es noch die fehler-toleranten Transceiver entsprechend ISO 11898-3, die in der Automobilindustrie allerdings meistens wegen der integrierten Low-power-Funktion eingesetzt werden. Sie unterstützen Datenraten bis zu 125 kbit/s. Außerdem gibt es noch einige spezielle Transceiver-Spezifikationen, z. B. für die „Eindraht“-Übertragung (SAE J1411) sowie die in Europa vom Gesetzgeber vorgeschriebene fehlertolerante Übertragung zwischen LKW und Anhänger (ISO 11992-1). Diese haben jedoch in der Fabrik- und Prozessautomation keine Bedeutung.

Da CAN nur die beiden unteren Schichten des OSI-Referenzmodells abdeckt, muss der Anwender die benötigten Funktionen der oberen Schichten selbst realisieren. Da die Systemintegratoren und OEMs (Original Equipment Manufacturer) in der Fabrik- und Prozessautomation im Vergleich zur Automobilindustrie nur kleine und mittlere Stückzahlen einsetzen, ist eine konsistente Entwicklung und Programmierung der höheren Protokolle nicht sinnvoll. Die Anwender erwarten, dass die Automatisierungsgeräte ohne Programmierung einsetzbar sind. Eine Konfiguration der Geräte mit generischen Werkzeugen ist akzeptabel. Deshalb wurden für CAN-basierte Netzwerke höhere Protokolle standardisiert. Historisch gab es diverse hersteller-spezifische und semi-offene Lösungen, die aber keine Marktakzeptanz erzielen konnten. Übrig geblieben sind zwei offene und international genormte Protokolle: CANopen (EN 50325-4) sowie Devicenet (IEC 62026-3). Beide haben ähnliche Kommunikationsdienste, unterscheiden sich aber prinzipiell in Bezug auf ihre Flexibilität und Einsatzmöglichkeiten. CANopen basiert auf dem von der Anwender- und Herstellervereinigung CAN in Automation (CiA) entwickeltem CAN Application Layer (CAL), ist inzwischen aber vollkommen unabhängig von diesem. CANopen wurde im Rahmen eines Esprit-Forschungsprojektes entwickelt und umfasst ein Kommunikati-

onsprofil (CiA 301 und CiA 302) sowie diverse Geräte- und Anwendungsprofile. Dieser Kommunikationsstandard eignet sich insbesondere für „eingebettete“ Steuerungen und ist auf Flexibilität optimiert.

Devicenet ist die CAN-basierende Lösung für das CIP-Protokoll (Common Industry Protocol) sowie die zugehörigen Geräteprofile. Es ist auf Durchgängigkeit optimiert und kann ohne großen Aufwand in Controlnet- und Ethernet-IP-Systeme integriert werden. Devicenet stellt in der Regel (ausgenommen CANopen-Anwendungsprofile) eine höhere Plug-and-play-Funktionalität als CANopen zur Verfügung, ist dafür aber weniger flexibel.

4.2.6.2 Physikalische Übertragung

Dies meisten in der Fabrikautomation eingesetzten CAN-Netzwerke verwenden High-speed-Transceiver (ISO 11898-2). Busankopplungen mit einer modifizierten EIA-485-Schaltung gehören der Vergangenheit an. Aufgrund der Busarbitrierungsmethode des CAN-Protokolls muss innerhalb einer Bitzeit von allen Teilnehmern der Bitwert bestimmt werden. Dadurch ergibt sich eine Begrenzung der Übertragungsgeschwindigkeit bei gegebener Netzwerklänge bzw. eine Limitierung der Netzwerklänge bei gegebener Datenrate. Die Grenzen hängen von vielen Faktoren ab: interne Laufzeiten in den CAN-Controllern, Verzögerungen in den Transceivern, Laufzeiten auf dem Kabel und den Steckern, usw.

Devicenet hat sehr präzise Vorschriften bezüglich der physikalischen Auslegungen der CAN-Netzwerke, um einen hohen Grad der Plug-and-play-Funktion zu erreichen. Es ist eine Stammleitungs-/Stichleitungs-Topologie mit bis zu 64 Netzknoten vorgeschrieben. Drei Bitraten (500 kbit/s, 250 kbit/s und 125 kbit/s) sind zugelassen, wobei die maximal erlaubten Netzwerklängen 100 m, 250 m bzw. 500 m betragen dürfen. Auch die Kabel und Stecker sind wohl definiert. Sowohl Signalübertragung als auch Stromversorgung laufen über das Netzwerkkabel. Dies erlaubt den Anschluss netzwerkversorgerter Geräte (z. B. kleiner Sensoren) und von Geräten mit eigener Stromversorgung (z. B. Frequenzumrichter). Sowohl „geschlossene“ als auch „offene“ Steckverbinder sind definiert: Erstere eignen sich beispielsweise für „verpackte“ Näherungsschalter, die im „Feld“ angebracht sind; letztere sind für Geräte bestimmt, die normalerweise in einen Schaltschrank eingebaut werden. Der physikalische Aufbau umfasst Transceiver, Steckverbinder, Verpolungsschutz, Spannungsregler und eine optionale optische Isolierung. Es dürfen nur Transceiver nach ISO 11898-2 eingesetzt werden, die mindestens 64 Teilnehmer treiben können.

CANopen erlaubt acht Datenraten (siehe Tab. 4.5) und gibt nur einige generelle Empfehlungen bezüglich der Verkabelung. Die Belegung der Steckeranschlüsse ist ebenfalls nur eine Empfehlung (CiA 303-1) und keine bindende Vorschrift. CANopen schreibt nicht vor, dass eine minimale Anzahl von Teilnehmern unterstützt werden muss. Die maximale Anzahl ist auf 127 begrenzt.

Die Bits werden mit einer differentiellen Übertragung auf den beiden Busleitungen (CAN-High und CAN-Low) übertragen. Dabei wird zwischen dominanten und rezessiven Bitwerten unterschieden, wobei die dominanten (Differenzspannung ist ungleich 0 V) und rezessiven (Differenzspannung ist gleich 0 V) überschreiben. Die dominante Differenzspannung beträgt nominal 2 V. Dies erlaubt eine maximale Netzwerklänge von rund

Tab. 4.5 In CANopen unterstützte Datenraten und empfohlene maximale Längen

Bitrate [kbit/s]	Maximale Länge [m]	Maximale Einzelstichleitung [m]	Maximale Länge aller Stichleitungen [m]	Ort des Abtastzeitpunktes [%]
1000	25	1,5	7,5	75 bis 90
800	50	2,5	12,5	75 bis 90
500	100	5,5	27,5	85 bis 90
250	250	11	55	85 bis 90
125	500	22	110	85 bis 90
50	1000	55	275	85 bis 90
20	2500	137,5	678,5	85 bis 90
10	5000	275	1375	85 bis 90

1 km. Bei größeren Entfernungen ist eine Auffrischung des Signals durch einen Repeater notwendig.

Der Anwender muss auch für einen Ausgleich von Potentialunterschieden über eine entsprechende Masseleitung sorgen. Devicenet macht hier klare Vorschriften, wie auch für die galvanische Trennung. CANopen und Devicenet schreiben eine Busterminalierung an beiden Enden der Linientopologie und die maximal erlaubten „unabgeschlossenen“ Stichleitungslängen vor.

Die meisten in der Praxis in CAN-basierenden Netzwerken auftretenden Fehler – dies gilt für CANopen und Devicenet, aber auch für hersteller-spezifische Lösungen – hängen mit einer mangelhaften Auslegung der physikalischen Übertragung zusammen. Das beginnt bei nicht vorhandenen Abschlusswiderständen und endet bei Nichtbeachtung der Bit-Timing-Vorschriften.

CANopen erlaubt auch die Verwendung anderer physikalischer Busanschaltungen. So spezifiziert beispielsweise die Empfehlung CiA 103 eine eigensichere Übertragung mit 3-V-Transceivern nach ISO 11898-2. Diese für explosionsgefährdete Anwendungen geeignete Übertragung wird beispielsweise in den USA in gaschromatografischen Systemen eingesetzt. Eine Übertragung mit Transceivern nach ISO 11898-3 ist in Zusammenhang mit dem Geräteprofil für Unterwasser-Instrumente (CiA 443) vorgeschrieben. Diese hochredundanten Netzwerke überwachen die Bohrungen nach Erdöl im Meer.

4.2.6.3 CAN-Protokoll

Im Gegensatz zu vielen anderen Bussystemen verwendet das CAN-Protokoll keine geräteorientierte Adressierung, sondern eine inhaltsorientierte Adressierung. Allerdings kann der Inhalt einzelner Nachrichten auch teilweise eine Geräteadresse enthalten. Bei einer rein inhaltsorientierten Adressierung, beschreibt der Nachrichten-Identifier eineindeutig die übertragenen Nutzdaten (beispielsweise einen Druck oder einen Druck mit zugehörigem Temperaturwert oder mehrere Binärwerte von unabhängigen Schaltern). Die eine

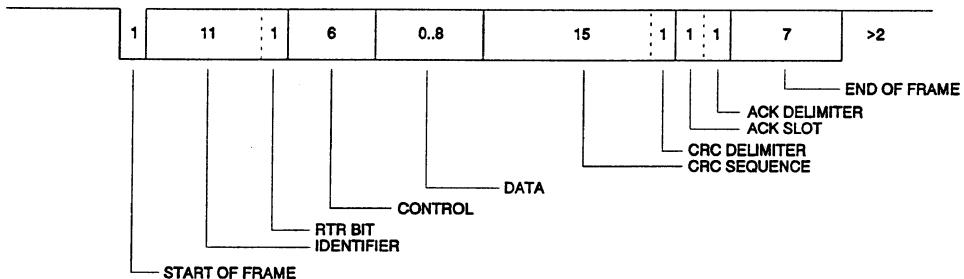


Abb. 4.34 CAN-Datentelegramm im Base-Frame-Format

CAN-Nachricht identifizierende Bitsequenz bestimmt auch gleichzeitig die Priorität mit der sie um den Buszugriff mit anderen Nachrichten konkurriert.

Jede CAN-Nachricht beginnt mit einem dominanten Start-of-Frame-Bit, dem der vom Systementwickler konfigurierte CAN-Identifier folgt. Devicenet erlaubt nur die Verwendung eines 11-bit-Identifiers (Base-Frame-Format). CANopen nutzt ebenfalls nur dieses Format, erlaubt aber auch optional die Verwendung von 29-bit-Identifiern (Extended-Frame-Format). Abb. 4.34 zeigt den Aufbau einer CAN-Nachricht im Base-Frame-Format. Somit lassen sich 2048 unterschiedliche Nachrichten in einem CAN-Netzwerk verwenden. Die Länge des Datenfeldes wird im Control-Feld angegeben (von 0 bis 8 Byte). Auf das Datenfeld folgen die CRC-Sequenz und der immer reressive CRC-Delimiter. Danach überträgt der Sender im ACK-Slot-Bit einen reressiven Wert, den alle Teilnehmer, deren CRC-Prüfung erfolgreich war, mit einem dominanten Wert überschreiben. Somit weiß der Sender, dass zumindest ein Teilnehmer seine Nachricht korrekt verstanden hat, er also nicht alleine im Netzwerk ist. Danach wird das reressive ACK-Delimiter-Bit übertragen, dem weitere sieben reressive Bits folgen (End-of-Frame). Zwischen zwei CAN-Frames muss immer ein Abstand von mindestens zwei Bitzeiten eingehalten werden (Interframe-Space), damit die empfangenen Daten zwischengespeichert werden können.

Die maximale Länge von CAN-Datenframes beträgt zwischen 44 bit und ungefähr 130 bit abhängig von der Länge des Datenfeldes und der zur Resynchronisation vom Sender automatisch eingefügten Stuff-Bits. Die Stuff-Bits werden von den Empfängern automatisch gefiltert, sodass nur auf dem Bus einige zusätzliche Bits übertragen werden. Die Begrenzung auf verhältnismäßig kurze Telegramme hat den Vorteil, dass eine hochpriore Nachricht nicht lange auf die Sendeerlaubnis warten muss (Buslatenzzeit), selbst wenn gerade eine niederpriore Nachricht übertragen wird.

Eine weitere Besonderheit im Vergleich zu anderen Bussen stellt das CAN-Zugriffsverfahren dar. CAN verwendet ein modifiziertes CSMA/CD-Verfahren (Carrier Sense Multiple Access/Collision Detection), wie es von Ethernet bekannt ist. Dabei ist jede Station vollkommen gleichberechtigt und darf immer dann, wenn der Bus nicht durch ein Telegramm belegt ist, auf ihn spontan zugreifen. Wie beim CSMA/CD-Verfahren hört je-

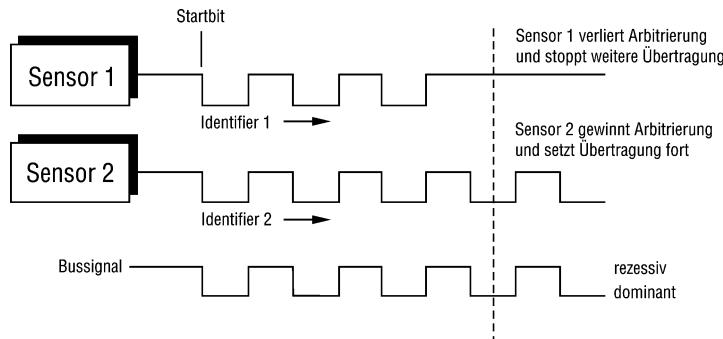


Abb. 4.35 CAN-Busarbitration

der CAN-Teilnehmer immer auf den Busleitungen mit, auch beim Senden. Wollen zwei Teilnehmer gleichzeitig senden, so entscheidet das prioritätengesteuerte Arbitrierungsverfahren, wer den Buszugriff bekommt. Im Kollisionsfall erhält der Teilnehmer mit der höheren Priorität ohne Zeitverlust die Sendeerlaubnis (CSMA/CA, Carrier Sense Multiple Access/Collision Avoidance).

Voraussetzung für dieses Verfahren sind die dominant/rezessiven Bits sowie die eindeutige Identifizierung der Nachrichten. Dies bedeutet, dass sich beim Senden zweier unterschiedlicher logischer Pegel durch mehrere Sender ein Zustand – der dominante – durchsetzt. Die CAN-internen Datenstrukturen verwenden als dominanten Bitzustand die „0“, während bei der physikalischen Realisierung des Übertragungsmediums beide Möglichkeiten erlaubt sind. Abb. 4.35 zeigt nun das Bussignal, das sich einstellt, wenn gleichzeitig zwei CAN-Teilnehmer beginnen, Datentelegramme nach Abb. 4.34 abzusetzen: Beide senden zunächst ein Startbit, das nach der Norm immer dominant sein muss. Danach folgt der 11-bit-Identifier. Im abgebildeten Beispiel sendet Knoten 1 den Identifier „101011...“ und Knoten 2 „101010...“. Beim sechsten Bit erkennt Knoten 1, dass sein rezessives Bit („1“) vom dominanten Bit („0“) eines anderen Knotens überschrieben wird und bricht die weitere Übertragung ab. Knoten 2 gewinnt die Arbitrierung und kann ohne Zeitverlust seine Nachricht senden.

Ein spontan nach diesem Mechanismus ausgesandtes CAN-Datentelegramm wird zunächst von allen anderen Busteilnehmern empfangen. Diese vergleichen den darin enthaltenen Identifier mit einer Liste von Nachrichten, die als zum Empfang gekennzeichnet sind. Nur bei den Nachrichten, deren Identifier mit dem des empfangenen Telegramms übereinstimmt, wird der Dateninhalt zwischengespeichert. Dadurch lassen sich sehr effektiv Multicast- und Broadcast-Verbindungen aufbauen.

Außer der eben geschilderten Kommunikationsbeziehung, bei der ein Sender eine Nachricht an einen oder mehrere Empfänger übermittelt, gibt es bei CAN auch die Möglichkeit der Fernabfrage: Dafür muss der eine Information erfragende Knoten ein Telegramm nach Abb. 4.34 mit einem auf „1“ gesetztem RTR-Bit (Remote Transmission Request) senden. Das Datenfeld hat diesem Falle eine Länge von Null. Der Teilnehmer,

der die korrespondierende Nachricht mit dem gleichen Identifier „besitzt“, sendet daraufhin die angeforderten Daten. Da die Beantwortung der Remote-Frames genannten Telegramme von den Halbleiterherstellern unterschiedlich implementiert ist und sie obendrein oft falsch genutzt werden, sind sie in Devicenet nicht erlaubt. In CANopen sind sie zwar nicht verboten, aber der Autor empfiehlt eindringlich, Remote-Frames nicht zu verwenden.

Das CAN-Protokoll verfügt über fünf fehlererkennende Maßnahmen (Bitfehler, Stuffbit-Fehler, CRC-Fehler, ACK-Fehler und Format-Fehler). Falls ein Fehler von einem Teilnehmer erkannt wird, sendet er ein Fehlertelegramm (entweder sechs dominante oder rezessive Bits und einen 8-bit-Delimiter). Handelt es sich um einen globalen Fehler, tun dies alle Teilnehmer gleichzeitig und der Sender wiederholt automatisch die abgebrochene Nachricht. Im Falle eines lokalen Fehlers, erkennen die anderen Teilnehmer spätestens nach zwölf Bitzeiten, dass ein Teilnehmer die Nachricht nicht korrekt verstanden hat. Sie senden nun ihrerseits auch ein Fehlertelegramm. Nach insgesamt 23 Bitzeiten (einschließlich Interframe-Space) wird die fehlerhafte Nachricht bereits vom Sender wiederholt, wenn keine höherpriore Nachricht die Busarbitrierung gewinnt. Dies ist eine sehr kurze Fehlererholzeit verglichen mit anderen Bussystemen. Damit ein permanent defekter Teilnehmer nicht die gesamte Kommunikation stören kann, enthalten alle CAN-Controller Sende- und Empfangsfehlerzähler, die schneller inkrementieren als herunterzählen (bei fehlerfreier Sendung bzw. fehlerfreiem Empfang werden die Zähler um den Wert 1 reduziert). Empfänger gehen bei einem Fehlerzählerstand von größer als 127 in den Error-Passive-Modus und stören die Kommunikation der anderen nicht mehr (sie senden nur noch eine rezessive Bitsequenz, die vom Sender überschrieben wird). Allerdings können sie so keine Fehler mehr signalisieren, was aus Systemsicht kritisch sein kann. Sender dürfen bei einem Fehlerzählerstand von größer als 255 nicht mehr senden und müssen nach einem Rücksetzen noch eine bestimmte zusätzliche Zeit warten, bevor sie wieder aktiv am Busgeschehen teilnehmen dürfen. Bei einem ordentlich aufgebauten CAN-Netzwerk sollten eigentlich niemals Fehlertelegramme auftauchen. Fehlertelegramme sind immer ein Zeichnen von nicht ausreichender Auslegung der physikalischen Übertragung oder einem Hardwarefehler.

4.2.6.4 CANopen

Die CANopen-Protokolle stellen dem Anwender Funktionen zur Segmentierung von Daten mit einer Länge größerer als 8 Byte (SDO: Servicedatenobjekte), Funktionen zum Zusammenstellen von Echtzeit-Nachrichten (PDO: Prozessdatenobjekte) sowie deren Sende- und Empfangsverhalten zur Verfügung. Diese höheren Kommunikationen entsprechen weitgehend der Anwendungsschicht im OSI-Referenzmodell. Sie sind im Kommunikationsprofile CiA 301 (international als EN 50325-4 genormt) spezifiziert. In dieser Spezifikation sind auch die Adressen sämtlicher Konfigurationsdaten für die Kommunikationsfunktionen festgelegt. Weitere Konfigurationsdaten für programmierbare Geräte und das Aufstarten (boot-up) des Netzwerkes sind in der Spezifikationsreihe CiA 302 definiert. CANopen umfasst darüber hinaus sogenannte Profile, die das applikative Verhalten

der Geräte (z. B. CiA 401 für modulare Ein-/Ausgabegeräte) festlegen. Das Geräteprofil für elektrische Antriebe (CiA 402) ist ebenfalls international genormt (IEC 61800-7-201/301).

Netzwerk-Management (NMT)

Um ein Netzwerk zu verwalten, das heißt, die einzelnen Geräte steuern und überwachen zu können, benötigt man ein Netzwerk-Managementsystem. CANopen spezifiziert deshalb spezielle Nachrichten mit denen der Systementwickler so ein System realisieren kann. In CANopen basiert das Netzwerkmanagement (NMT) weitgehend auf einer „Master/Slave“-Beziehung: Es gibt nur einen aktiven NMT-Master im Netzwerk, der die NMT-Slaves steuert. Jedes CANopen-Gerät muss die NMT-Slave-Zustandsmaschine implementieren (Abb. 4.36), dies gilt auch für das CANopen-Gerät mit der NMT-Master-Funktion. Nach dem Einschalten der Geräte durchlaufen diese eine Initialisierungsphase und gelangen automatisch in den Zustand „Pre-Operational“. In diesem Zustand warten die Geräte auf eine eventuelle Konfiguration bzw. einen generellen Prüfung durch einen Anwendungsmaster.

Bei dem automatischen Übergang vom Zustand „Communication Reset“ zu „Pre-Operational“ müssen die Geräte die Boot-up-Nachricht senden. Das ist eine 1-Byte-Nachricht, die den Wert 0 überträgt. Sie ist quasi eine Anmeldung im Netzwerk. Aus dem verwendeten CAN-Identifier können die Empfänger erkennen welche eindeutige

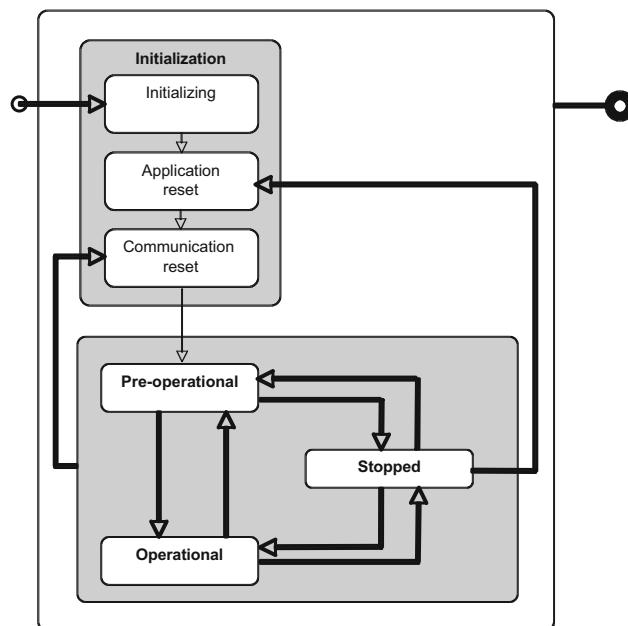


Abb. 4.36 NMT-Slave-Zustandsmaschine

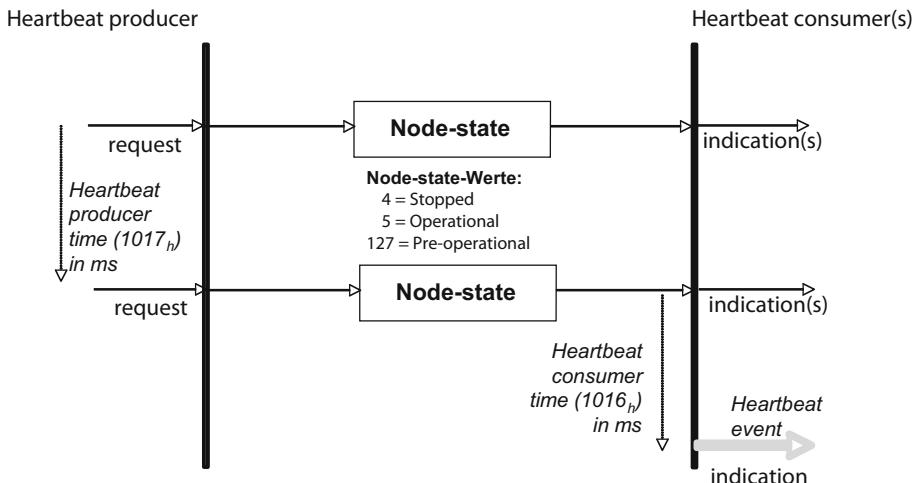
Tab. 4.6 Kommunikationsdienste und NMT-Zustände

Dienst	Pre-Operational	Operational	Stopped
NMT	Ja	Ja	Ja
Heartbeat	Ja	Ja	Ja
SDO	Ja	Ja	Nein
PDO	Nein	Ja	Nein
EMCY	Ja	Ja	Nein
SYNC	Ja	Ja	Nein
TIME	Ja	Ja	Nein

Knotennummer (Node-ID) dieses Gerät hat. Übrigens: Die Boot-up-Nachricht verwendet denselben CAN-Identifier wie für die Heartbeat-Nachricht (siehe weiter unten).

In einem CANopen-Netzwerk können sich auch mehrere Anwendungsmaster befinden, allerdings ist nur einer der aktive NMT-Master, der die Ressource „Kommunikationssystem“ verwaltet. Ist die Startphase einschließlich Konfiguration abgeschlossen, schaltet der NMT-Master die angeschlossenen Geräte in den Zustand „NMT-Operational“, dann werden alle CANopen-Dienste unterstützt (Tab. 4.6). Dazu verwendet er die höchspriore CAN-Nachricht mit dem Identifier „0“. Dieses als Zweibyte-Nachricht spezifizierte Kommando (Abb. 4.37) enthält den eigentlichen Transitionsbefehl und den adressierten Teilnehmer, der ihn ausführen soll. Damit nicht jeder Teilnehmer einzeln gestartet werden muss, gibt es auch ein Broadcast-NMT-Kommando: Dafür wird in der NMT-Nachricht die Node-ID auf „0“ gesetzt.

Deshalb muss der Systemintegrator jedem CANopen-Gerät eine eindeutige Knotennummer (Node-ID) zuteilen. Die CANopen-Norm legt nicht fest, wie diese Zutei-

**Abb. 4.37** Heartbeat-Protokoll

lung zu realisieren ist. Oft werden DIP-Schalter verwendet, aber auch eine geografische Adressierung mit Steckern oder eine Einstellung der Adresse über lokale Displays sind möglich. Falls die Adressierung über das CANopen-Netzwerk erfolgen soll, sind die LSS-Protokolle (CiA 305) oder die in CiA 416 spezifizierte „Node-Claiming“-Prozedur verwendbar.

Der NMT-Befehl wird von den NMT-Slave-Geräten auf Anwendungsebene mit der Heartbeat-Nachricht bestätigt. Der Heartbeat wird periodisch gesendet (Abb. 4.37). Die Periode ist entweder durch ein Profil vorgegeben oder muss in der Konfigurationsphase parametriert werden. Diese Einbyte-Nachricht enthält den jeweiligen NMT-Zustand des Gerätes. So kann der NMT-Master überprüfen, ob sein NMT-Kommando ausgeführt wurde oder nicht. Der Heartbeat wird mit einer sehr niedrigen Priorität gesendet. Der verwendete CAN-Identifier resultiert aus dem Basiswert 800_h plus der zugeteilten Node-ID. Der Teilnehmer mit der Node-ID „7“ sendet also seinen Heartbeat mit der CAN-ID 807_h .

Der NMT-Master kann auch einen oder mehrere Teilnehmer in den Zustand „Stopped“ setzen. Im Zustand „Stopped“ werden nur noch NMT-Kommandos akzeptiert. Damit der Teilnehmer sich nicht mehr an der Kommunikation teilnimmt, muss vorher der Heartbeat abgeschaltet werden, d. h., der Heartbeat-Producer-Timer muss auf null gesetzt werden. Mit Hilfe des Stopp-Kommandos kann der NMT-Master einzelne Geräte zum Schweigen bringen, die nicht benötigt werden oder die die anderen gestört haben.

Der Heartbeat hat noch eine zweite Funktion: er dient dazu festzustellen, ob die Geräte noch aktiv im Netzwerk vorhanden sind. Da viele CANopen-Geräte nur dann etwas senden, wenn sich etwas geändert hat, wissen die potentiellen Empfänger nicht, ob ein Gerät nichts zu vermelden hat oder ob es sich vom Bus selbstständig aufgrund von fehlerhaftem Verhalten zurückgezogen hat (Bus-off-Zustand des CAN-Controllers). Deshalb überwachen alle Teilnehmer die für ihre Anwendung wichtigen Geräte, d. h., sie „hören“ auf deren Heartbeat. Diese müssen selbstverständlich entsprechend konfiguriert werden.

Servicedatenobjekte (SDO)

Die Konfiguration eines CANopen-Gerätes erfolgt mit Hilfe der Servicedatenobjekte (SDO). Dies ist ein bestätigter Kommunikationsdienst, also eine Client/Server-Beziehung, wobei der SDO-Client die Kommunikationsinitiative hat. Er schreibt in das Objektverzeichnis des SDO-Servers oder liest aus diesem Parameter aus. Das Objektverzeichnis ist eine strukturierte Parameterliste mit einem eindeutigen Adressierungsschema (siehe Tab. 4.7). Die Adresse besteht aus einem 16-bit-Index und einem 8-bit-Subindex (also eine 24-bit-Adresse).

Um schreibend oder lesend auf das Objektverzeichnis des SDO-Servers zugreifen zu können, muss der SDO-Client in seiner CAN-Nachricht in einem 1-Byte-Wert (Command Specifier) seine Kommunikationsanforderung (Lesen oder Schreiben) mitteilen sowie in drei weiteren Bytes (MUX) die gewünschte Adresse im Objektverzeichnis (16-bit-Index und 8-bit-Subindex) angeben. In den verbleibenden vier Bytes des Datenfeldes können die zu schreibenden Parameterwerte übertragen werden (Abb. 4.38). Der SDO-Server bestätigt den Schreibzugriff (Download) als erfolgreich mit einer 8-Byte-Nachricht (SDO-

Tab. 4.7 Struktur des CANopen-Objektverzeichnisses

Index-Bereiche [hexadezimal]	Parameter-Funktion
0000	Reserviert
0001 bis 025F	Datentypen
0260 bis 0FFF	Reserviert
1000 bis 1FFF	Kommunikationsparameter
2000 bis 5FFF	Herstellerspezifische Anwendungsparameter
6000 bis 67FF	1. Logisches CANopen-Gerät (Profil 1)
6800 bis 6FFF	2. Logisches CANopen-Gerät (Profil 2)
7000 bis 77FF	3. Logisches CANopen-Gerät (Profil 3)
7800 bis 7FFF	4. Logisches CANopen-Gerät (Profil 4)
8000 bis 87FF	5. Logisches CANopen-Gerät (Profil 5)
8800 bis 8FFF	6. Logisches CANopen-Gerät (Profil 6)
9000 bis 97FF	7. Logisches CANopen-Gerät (Profil 7)
9800 bis 9FFF	8. Logisches CANopen-Gerät (Profil 8)
A000 bis AFFF	Netzwerkvariable (z. B. für IEC 61131-3)
B000 bis BFFF	Systemvariable (z. B. für Netzwerkbrücken)
C000 bis FFFF	Reserviert

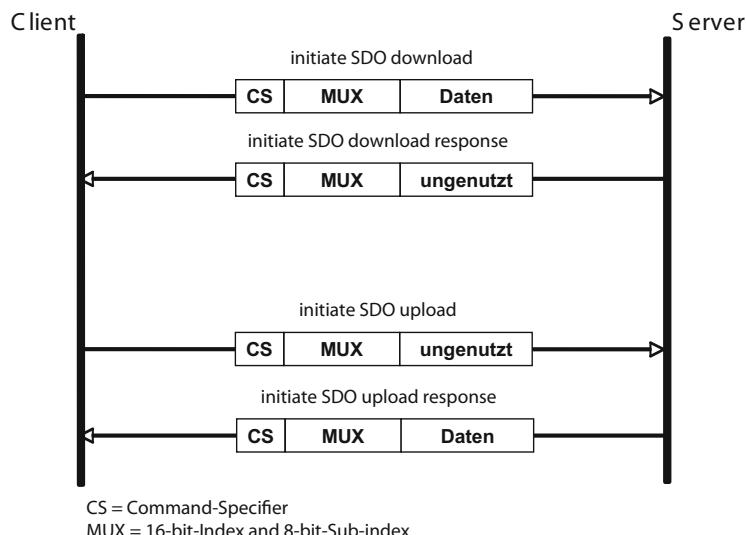


Abb. 4.38 SDO-Protokoll ohne Segmentierung

Response) oder lehnt ihn ab (SDO-Abort). Bei einem Lesezugriff (Upload) enthält die SDO-Antwort des SDO-Servers die angeforderten Daten (bis zu 4 Byte) sowie 1-Byte-Protokollinformation sowie die 3-Byte-Adresse des mitgelieferten Parameters. Für eine SDO-Kommunikation benötigt man zwei CAN-Nachrichten mit unterschiedlichen Identifi-

fieren (vom Client zum Server und vom Server zum Client). Jede SDO-Nachricht hat eine Länge von 8 Byte.

Theoretisch kann jeder Teilnehmer für jedem anderen Teilnehmer SDO-Client und SDO-Server sein. Für eine voll vermaschte SDO-Kommunikation gibt es allerdings bei Verwendung von 11-bit-Identifizieren nicht genügend CAN-IDs für alle 127 Teilnehmer. Da die SDO-Dienste überwiegend für Konfigurations- und Diagnoseaufgaben genutzt werden, ist dies auch nicht erforderlich. In vielen Fällen parametert nur der Anwendungsmaster die ihm zugeordneten CANopen-Geräte. Jedes CANopen-Gerät muss deshalb einen so genannten SDO-Default-Server zur Verfügung stellen. Er besitzt vordefinierte CAN-IDs (600_h plus Node-ID und 580_h plus Node-ID). Der korrespondierende SDO-Client kann in jedem Anwendungsmaster (von denen es in einem CAN-Netzwerk mehrere geben kann) eingerichtet werden. Eine SDO-Verbindung zwischen zwei Geräten ist immer exklusiv. Da einfache CANopen-Geräte aus Ressourcen-Gründen nur den SDO-Default-Server besitzen, kann also außer dem Anwendungsmaster kein externes Diagnose- oder Konfigurationswerkzeug auf dieses Gerät per SDO zugreifen. Um dies dennoch zu ermöglichen, kann der Anwendungsmaster einen SDO-Manager implementieren, der in der Lage ist, den zugehörigen SDO-Client auf Anfrage an ein Werkzeug oder ein anderes Gerät zu verleihen.

Um Parameterdaten größer als 4 Byte schreiben oder lesen zu können, gibt es auch einen segmentierten SDO-Transfer. Dies wird im ersten Segment im 1-Byte-Kommando angekündigt, gefolgt von der 3-Byte-Adresse (Index und Subindex) sowie optional kann noch die Länge des zu schreibenden Parameters mitgegeben werden. Die nächsten Segmente des SDO-Client enthalten neben dem 1-Byte-Kommando bis zu 7-Byte-Parameterdaten. Jedes Segment muss vom SDO-Client bestätigt werden oder er muss im Fehlerfall den SDO-Transfer abbrechen (SDO-Abort). Das letzte SDO-Segment enthält im 1-Byte-Kommando eine Ende-Erkennung, sodass es theoretisch keine Längenbeschränkung gibt. Bei sehr großen Parameterdaten (z. B. Programmdownload) ist die Bestätigung jedes Segments sehr zeitaufwändig. Deshalb gibt es ein weiteres SDO-Protokoll, den SDO-Blocktransfer. Hierbei wird nicht jede SDO-Nachricht des SDO-Client bestätigt, sondern erst eine konfigurierbare Anzahl von Segmenten. Nachteil ist, dass ein eventuell erkannter Fehler erst am Ende des Blocks mitgeteilt werden kann.

Prozessdatenobjekte (PDO)

Man könnte ein CANopen-Netzwerk ausschließlich mit SDO-Diensten betreiben, wenn man keine Multicast- oder Broadcast-Nachrichten benötigt und die Prozessdaten zeitunkritisch sind. In den meisten Steuerungssystemen reicht dies jedoch nicht aus. Die Prozessdaten müssen möglichst ohne Protokoll-Overhead, oft auch an mehrere Empfänger gleichzeitig gesendet werden. Hierzu dienen in CANopen die Prozessdatenobjekte (PDOs). Das sind einzelne CAN-Nachrichten, die im Datenfeld (1 Byte bis 8 Byte) ein oder mehrere Prozessdaten enthalten. Die Interpretation des Datenfeldes erfolgt mit Hilfe von konfigurierten Mapping-Parametern, die der CANopen-Software mitteilen, an wel-

cher Stelle im Objektverzeichnis die empfangen Prozessdaten abzulegen sind, bzw. welche Prozessdaten in einem PDO in welcher Reihenfolge zu übertragen sind.

PDOs werden auf der Kommunikationsebene nicht bestätigt. Der Systementwickler kann die zu sendenden und zu empfangenen PDOs bezüglich ihres Verhaltens und des Dateninhaltes konfigurieren. Hierbei gewährt ihm CANopen ein sehr hohes Optimierungspotential und eine weitgehende Anpassungsfähigkeit an unterschiedlichste Aufgaben. Die PDO-Konfigurationsparameter für die Kommunikation und das Mapping sind mit Hilfe der oben erwähnten SDO-Dienste schreib- und lesbar.

Die PDO-Kommunikationsparameter bestehen aus der so genannten COB-ID, einem 32-bit-Parameter, der neben dem zu verwendenden CAN-Identifier noch einige Zusatzbits enthält. Ein Zusatzbit (Bit 29) legt fest, ob das Base-Frame-Format oder das Extended-Frame-Format genommen werden soll. Das Bit 30 schaltet die Fernabfrage ab (keine Remote-Frames), dies unterstützen allerdings nicht alle CAN-Controller, und Bit 31 aktiviert und deaktiviert das PDO. Man kann also einzelne PDOs ab- und anschalten.

Der Sub-Parameter „Transmission Type“ legt das Sende- bzw. Empfangsverhalten fest. Beim Senden wird prinzipiell zwischen asynchroner und synchroner Übertragung entschieden: Bei asynchroner Übertragung hat das Gerät die Initiative, es sendet beispielsweise, wenn sich eines der gemappten Prozessdaten ändern (Change-of-State), oder wenn der interne Zeitgeber (dieser Event-Timer ist ebenfalls ein konfigurierbarer PDO-Parameter) abgelaufen ist (periodische Übertragung). Bei synchroner Übertragung wird das Abtasten der zu sendenden Prozessdaten von außen durch einen spezielle CANopen-Dienst (SYNC-Nachricht) getriggert. Bei dieser zyklischen Übertragung kann der Systementwickler entscheiden, ob bei jedem Empfang der SYNC-Nachricht, die Daten abgefragt und gesendet werden sollen, oder nur bei jedem zweiten, dritten, vierten, usw. Empfang. So lässt sich die Buslast für sich langsam ändernde Prozessdaten minimieren. Eine weitere Optimierung bezüglich der Buslast erlaubt die azyklische Übertragung. In diesem Fall wird das PDO nur gesendet, wenn das SYNC-Telegramm empfangen wurde und sich eines der gemappten Prozessdaten geändert hat. Eine Anforderung von PDOs per Remote-Frame ist in CANopen erlaubt, aber aus oben genannten Gründen empfiehlt der Autor, keine Remote-Frames zu verwenden.

Um bei einer asynchronen PDO-Übertragung zu verhindern, dass ein hochpriores PDO den Bus ständig belegt, gibt es den Inhibit-Timer (ein konfigurierbarer PDO-Parameter), mit dem das erneute Aussenden für eine gewisse Zeit verhindert wird. Dies gibt niederpriore PDOs die Möglichkeit, auch bei sich ständig ändernden Prozessdaten im hochprioren PDO, den Buszugriff zu erhalten.

Auf der Empfangsseite gibt es nur zwei Möglichkeiten. Die Prozessdaten werden sofort verarbeitet (asynchroner Empfang) oder sie werden mit der nächsten empfangenen SYNC-Nachricht als gültig betrachtet. Oft werden die TPDOs und die korrespondierenden RPDOs als asynchron oder synchron konfiguriert. Eine synchrone Kommunikation von PDOs erlaubt ein gleichzeitiges Erfassen von Messwerten und ein gleichzeitige Ausführen von Befehlen. Dies vor allem in mehrachsigen Maschinen von Bedeutung.

Eine weitere interessante Möglichkeit ist die unterschiedliche Konfiguration von Sende- und Empfangs-PDO. So kann man beispielsweise das Sende-PDO (TPDO) als synchron konfigurieren und die korrespondierenden Empfangs-PDO (RPDO) als asynchron. So erhält man zeitgleich erfasste Prozessdaten, die in den empfangenden Geräten sofort verarbeitet werden. Man kann aber auch PDOs asynchron senden und synchron empfangen. Dann hat man eine zeitgleiche Verarbeitung der ereignis-getriggerten Prozessdaten.

Der Entwickler von CANopen-Geräten muss allerdings sicherstellen, dass zum Zeitpunkt des Empfangs der SYNC-Nachricht die Prozessdaten aktualisiert wurden, insbesondere wenn das Gerät intern die Prozessdaten periodisch auswertet. Da der SYNC-Zyklus normalerweise nicht mit der internen Verarbeitung synchronisiert ist, kann es dazu führen, dass die Prozessdaten erst nach dem Senden der synchronen PDOs aktualisiert werden. Um dieses Problem zu vermeiden, kann man den internen Verarbeitungszyklus mit der SYNC-Periode synchronisieren oder intern die Prozessdaten ereignis-orientiert verarbeiten. Zu Letzterem ist allerdings ein lokales Echtzeit-Betriebssystem erforderlich.

In einigen Anwendungen benötigt man für eine optimale Buslastverteilung mehrere SYNC-Nachrichten. Dies wird in CANopen virtuell mit dem SYNC-Zähler erreicht. Dieser 1-Byte-Zähler wird optional in der SYNC-Nachricht übertragen. In dem SYNC-Counter-Überlauf-Parameter legt der Systementwickler fest, bei welchem Wert der Zähler überläuft und wieder bei „1“ startet. In jedem synchron gesendeten PDO muss der Systementwickler den Startwert konfigurieren (SYNC-Start-Value). Reagiert die synchronen Sende-PDOs nicht auf jede SYNC-Nachricht, so muss der Systementwickler darauf achten, dass Startwert und Anzahl der „bedienten“ SYNC-Nachrichten nicht zu einer ungleichmäßig verteilten Buslast führen. Ein einfaches Beispiel zeigt Abb. 4.39: Der SYNC-Counter läuft beim Wert „2“ über und startet wieder mit dem Wert „1“. Bis auf ein PDO reagieren alle auf jede zweite SYNC-Nachricht. Nur ein PDO „bedient“ jede vierte SYNC-Nachricht. Selbstverständlich kann man mit diesem erweiterten SYNC-Verfahren auch komplexere Szenarien realisieren.

Es gibt noch eine zweite Möglichkeit in CANopen-Netzwerken verschiedenen Teilnehmer zu koordinieren. Dazu ist eine systemweite Zeit erforderlich. Zwar verfügen viele Geräte über eine lokale Echtzeituhr, die aber nicht mit denen der anderen Teilnehmer synchronisiert ist. Die nicht synchronisierten lokalen Uhren weichen mit fortschreitender Zeit immer weiter voneinander ab. Für das Nachjustieren dieser lokalen Uhren bietet CANopen die TIME-Nachricht. Dieses 6-Byte-Telegramm enthält eine absolute Zeit in Millisekunden nach Mitternacht und Tagen nach dem 1.1.1984. Mit dieser absoluten Zeit können nun in den CANopen-Geräten Nachrichten (z. B. PDOs und SDOs) in Abhängigkeit von Datum und Uhrzeit mit einer Genauigkeit von einer Millisekunde versendet werden. Selbstverständlich müssen die lokalen Uhren regelmäßig nachgestellt werden. Es darf nur einen TIME-Produzenten geben, die Anzahl der TIME-Konsumenten ist nicht begrenzt.

Die in CANopen spezifizierten Emergency-Nachrichten (EMCY) sind den PDOs sehr ähnlich. Sie sind vordefiniert und können nicht konfiguriert werden. Sie haben immer eine

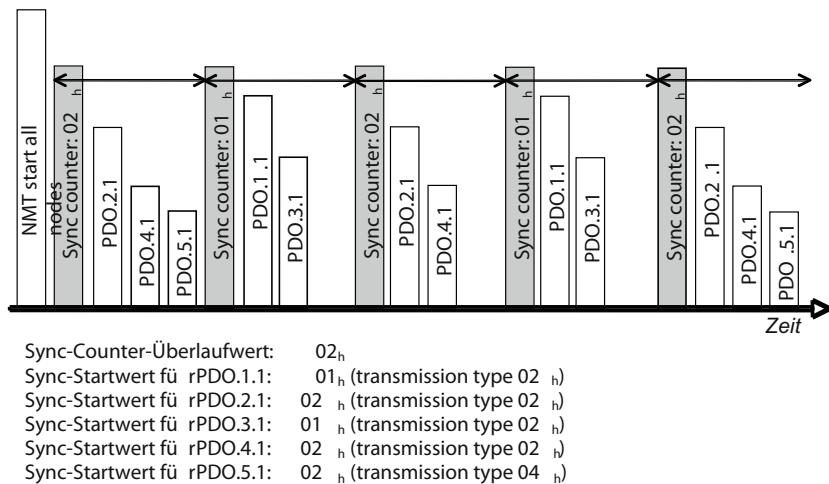
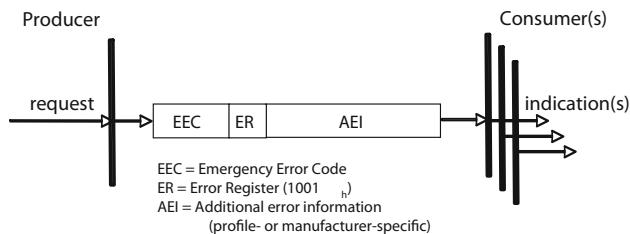


Abb. 4.39 Sync-Nachricht mit Sync-Counter und zyklischen PDOs

Abb. 4.40 Aufbau der Emergency-Nachricht. Sie kann von jedem anderen Teilnehmer empfangen werden



Länge von 8 Byte (siehe Abb. 4.40). In einer Parameterliste kann der Systemintegrator konfigurieren, von welchen Geräten die EMCY-Nachrichten empfangen werden sollen. Neben den Kommunikationsproblemen und einigen generischen Fehlermeldungen, sind in den CANopen-Profilen weitere spezifische Fehlerkodes festgelegt.

Objektverzeichnis

Jedes CANopen-Gerät verfügt über ein Objektverzeichnis, eine Parameterliste, auf die man per SDO über das CAN-Netzwerk zugreifen kann (Tab. 4.7). Die Objektverzeichnis-Struktur ist definiert. Es sind jedoch nur wenige Parameter verpflichtend vorgeschrieben. Die eindeutige Adresse ist ein 24-bit-Wert (16-bit-Index und 8-bit-Subindex). Es wird zwischen Variablen, Arrays und Records unterschieden. Eine Variable hat nur eine Adresse, wobei der Subindex immer 00_h ist. Bei Arrays und Records gibt der Sub-Parameter mit der Unteradresse 00_h den höchsten verwendeten Subindex an. Arrays bestehen aus 1 bis 254 Sub-Parametern, des gleichen Datentyps (z. B. Unsigned8 oder Integer16). Records haben ebenfalls bis zu 254 Sub-Parameter, deren Datentypen unterschiedlich sind.

Von den allgemeinen Kommunikationsparametern (Tab. 4.8) sind der Gerätetyp (1000 00_h), das Fehlerregister (1001 00_h) und das Identity-Array (1018 00_h bis 1018 04_h)

Tab. 4.8 Die wichtigsten allgemeinen CANopen-Kommunikationsparameter

Index [hex]	Art	Funktion	Verpflichtend
1000	Variable	Zeigt das implementierte Profil an	Ja
1001	Variable	Gibt den Gerätestatus an	Ja
1005	Variable	Enthält u. a. die CAN-ID für die SYNC-Nachricht	Optional
1006	Variable	Enthält die SYNC-Zykluszeit	Optional
1010	Array	Speichert die angegebenen Parameter ins EPROM	Optional
1011	Array	Setzt das Gerät auf die angegebenen Parameter zurück	Optional
1014	Variable	Enthält u. a. die CAN-ID für die EMCY-Nachricht	Optional
1015	Variable	Enthält die konfigurierbare EMCY-Verbotszeit	Optional
1016	Array	Enthält die Node-IDs der zu empfangenen Heartbeats und die Zeiten, wann ein Heartbeat-Event ausgelöst wird	Optional
1017	Variable	Legt die Periode des eigenen Heartbeats fest	Empfohlen
1018	Record	Enthält Vendor-ID, Produktcode, Revision- und Seriennr.	Ja/optional
1028	Array	Enthält die CAN-IDs der zu empfangenen EMCYs	Optional
1029	Array	Legt fest, wie sich das Gerät bei schweren Fehlern verhält	(Ja)

vorgeschrieben. Das Identity-Array enthält im Subindex 1 die vom CiA-Verein vergebene, weltweit eindeutige Hersteller-Identifikation (CANopen-Vendor-ID). Die anderen optionalen Sub-Parameter sind der Produktcode, die Revisionsnummer und die Seriennummer, die der Geräte-Hersteller verwaltet. Da es empfohlen ist, das Heartbeat-Protokoll zu unterstützen, ist auch die Heartbeat-Producer-Zeit quasi verpflichtend. Mit ihr wird die Periode des Heartbeat konfiguriert. Ob das Gerät auch Heartbeat konsumieren können muss, hängt von seiner Funktion ab: Alle Geräte mit Ausgabe-Funktionen (z. B. elektrische Antriebe, Displays und Schaltelemente) sollten den Heartbeat der Geräte konsumieren, die ihnen per PDO Kommandos senden.

Die anderen Kommunikationsparameter sind optional. In vielen CANopen-Geräten ist ein permanentes Speichern der Konfigurationsdaten möglich. Dazu wird in den Store-Parameter eine Signatur geschrieben („save“), schreibt man in den Restore-Parameter die Signatur („load“), so wird das Gerät auf die Werkseinstellungen zurückgesetzt. Der Error-Behavior-Parameter ist zwar nur optional zu unterstützen, aber sein Default-Verhalten ist trotzdem verpflichtend. Das heißt, im Falle eines schwerwiegenden Kommunikationsproblems (z. B. Heartbeat-Event, Bus-off des CAN Controllers) nimmt das CANopen-Gerät automatisch den Zustand NMT-Pre-Operational ein. Unterstützt das CANopen-Gerät den EMCY-Dienst sollte nicht nur der Parameter EMCY-COB-ID, in dem der zu verwendende CAN-Identifier hinterlegt ist, sondern auch die EMCY-Inhibit-Zeit implementiert werden. Mit ihr lässt es sich vermeiden, dass die üblicherweise hochpriore EMCY-Nachricht, das Netzwerk für niederpriore Aufgaben blockiert.

Im Objektverzeichnis sind auch die PDO-Kommunikation-Parameter und die PDO-Mapping-Parameter adressierbar (Tab. 4.9). Insgesamt sind pro CANopen-Gerät bis zu

Tab. 4.9 PDO- und SDO-Parameter

Index [hex]	Funktion
1200 bis 127F	SDO-Server-Parameter: CAN-IDs und Node-ID des SDO-Client (optional)
1280 bis 12FF	SDO-Client-Parameter: CAN-IDs und Node-ID des SDO-Server (optional)
1400 bis 15FF	RPDO-Kommunikation: CAN-ID, Transmission-Typ, Event-Timer, usw.
1600 bis 17FF	RPDO-Mapping: Bis zu 64 Zeiger auf die Prozessdaten-Indizes
1800 bis 19FF	TPDO-Kommunikation: CAN-ID, Transmission-Typ, Event-Timer, usw.
1A00 bis 1BFF	TPDO-Mapping: Bis zu 64 Zeiger auf die zu übertragenen Prozessdaten-Indizes

512 Sende-PDOs und bis zu 512 Empfangs-PDOs implementierbar. Dies sollte selbst für sehr komplexe Geräte ausreichen. Werden dennoch mehr PDOs benötigt, so kann das Multiplex-PDO-Protokoll verwenden. Hierbei wird neben einem einzigen Prozessdatum mit einer maximalen Länge von 4 Byte, noch die „Adresse“ (3 byte) des Objektverzeichnisses im Sender (Source-Addressing-Mode) oder im Empfänger (Destination-Addressing-Mode) sowie ein Steuerbyte in der MPDO-Nachricht übertragen.

Das Objektverzeichnis enthält auch Parameter für die SDO-Kommunikation (Tab. 4.9). Da die CAN-Identifier für den Default-SDO-Server nicht konfigurierbar und abschaltbar sind, müssen diese nicht implementiert werden. Alle SDO-Clients und zusätzliche SDO-Server müssen vom Anwender bezüglich der verwendeten CAN-Identifier konfiguriert werden. Maximal sind 127 Clients und 127 Server in jedem Gerät möglich. In der Regel verfügen die einfachen CANopen-Geräte nur über den Default-SDO-Server.

Im Index-Bereich kann der Gerätehersteller seine eigenen Parameter hinterlegen. Der Adressbereich 6000_h bis $9FFF_h$ ist für vom CiA standardisierte Profilparameter reserviert. Der Bereich ist in acht logische Geräte aufgeteilt, sodass ein CANopen-Gerät bis zu acht Profile implementieren kann. Dies müssen nicht unbedingt verschiedene Funktionen sein (z. B. ein Antriebsprofil, ein Encoder-Profil und ein Ein-/Ausgabe-Profil). Man kann auch bis acht Antriebsprofile implementieren, dann steuert ein CANopen-Gerät bis zu acht Elektromotoren.

Geräte- und Anwendungsprofile

CANopen zählt zu den interoperabelsten Netzwerken. Dies liegt an der weitgehenden Standardisierung der Geräte- und Anwendungsprofilen, in denen die Prozessdaten, die anwendungs-spezifischen Konfigurationsparameter sowie applikative Diagnoseinformationen spezifiziert sind. In vielen Profilspezifikationen sind auch die PDO-Parameter (Kommunikation und Mapping) vordefiniert. Geräteprofile beschreiben die Schnittstelle eines einzelnen Gerätes, beispielsweise ein Ein-/Ausgabe-Modul (CiA 401) oder einen elektrischen Antrieb (CiA 402, international genormt in IEC 61800-7-201/301) oder einen Drehgeber (CiA 406). Die PDOs verwenden die vordefinierten CAN-Identifier (siehe Tab. 4.10). Der Systemintegrator muss die korrespondierenden SDO-Clients und PDOs in der oder den Hostcontrollern programmieren bzw. konfigurieren. CANopen erlaubt,

Tab. 4.10 Vordefinierte CAN-IDs für CANopen-Geräteprofile

Service	Funktions-kode	Zusatz	Resultierende CAN-IDs
NMT-Nachricht	0000 _b	0	0
SYNC-Nachricht	0001 _b	0	128
EMCY-Nachricht	0001 _b	Node-ID	129 bis 255
TIME-Nachricht	0010 _b	0	256
TPDO 1	0011 _b	Node-ID	385 bis 511
RPDO 1	0100 _b	Node-ID	513 bis 639
TPDO 2	0101 _b	Node-ID	641 bis 767
RPDO 2	0011 _b	Node-ID	796 bis 895
TPDO 3	0011 _b	Node-ID	897 bis 1023
RPDO 3	0011 _b	Node-ID	1025 bis 1151
TPDO 4	0011 _b	Node-ID	1153 bis 1279
RPDO 4	0011 _b	Node-ID	1281 bis 1407
Default-SDO-Server (tx)	1011 _b	Node-ID	1409 bis 1535
Default-SDO-Server (rx)	1100 _b	Node-ID	1537 bis 1663
Heartbeat/Boot-up	1110 _b	Node-ID	1793 bis 1919

dass sich mehrere Hostcontroller in einem Netzwerk befinden. Sie können sich sogar die Funktionalität eines Gerätes teilen, beispielsweise kann ein Hostcontroller die Hälfte der Eingänge nutzen und ein anderen die restlichen oder ein Neigungssensor kann zwei Steuerungen mit Daten versorgen. Die Hostcontroller sind in der Regel programmierbare Geräte, die kein Profil implementieren. Sie nutzen so genannte Netzwerkvariable, die in PDOs gepackt und versendet werden beziehungsweise in PDOs empfangenen Prozessdaten in den konfigurierten Netzwerkvariablen ablegen (Indexbereich: A000_h bis AFFF_h).

Anwendungsprofile spezifizieren ein komplettes System, d. h., alle PDO-Sender und PDO-Empfänger sind standardisiert, einschließlich einer PDO-Querkommunikation und PDO-Multicast- sowie PDO-Broadcast-Verbindungen. Typische Beispiele sind das Anwendungsprofil für Aufzugsteuerungen (CiA 417) und das für Abfallsammelfahrzeuge (CiA 422). Üblicherweise werden so genannte virtuelle Geräte spezifiziert, das sind funktionale Elemente, die in einem realen CANopen-Gerät implementiert werden. Ein CANopen-Gerät darf mehrere virtuelle Geräte enthalten. Der Systemintegrator muss also nur die funktionalen Elemente konsistent zusammenstellen. Die CAN-Identifier der PDOs hängen nicht von der Knotennummer (Node-ID) ab! Es handelt sich also um Plug&Play-Systeme, eine Konfiguration der PDOs ist nur dann erforderlich, wenn man die voreingestellte Kommunikationsbeziehungen ändern möchte. Die anderen CAN-Identifer sind die gleichen, wie bei den Geräte-Profilen (Tab. 4.10).

Tab. 4.11 CAN-IDs der CANopen-Sonderfunktionen

Funktion	Genutzte CAN-ID
Global-Fail-Safe-Kommando (CANopen-Safety)	1
SRDOs (CANopen-Safety)	257 bis 384
Flying-NMT-Master	113 bis 118
Dynamic-SDO-Request (SDO-Manager)	1760
Node-Claiming-Procedure	1761 bis 1763, 1776 bis 1791
Layer-Setting-Services	2020 und 2021

CANopen-Sonderfunktionen

CANopen stellt eine Reihe von Sonderfunktionen zur Verfügung. Dazu gehören die „Safety-Related Data Objects (SRDO)“, mit denen es möglich ist, Prozessdaten sicherheitsgerichtet zu übertragen. Ein SRDO besteht aus zwei CAN-Nachrichten, die kongruent zu den PDOs sind. Die zweite Nachricht enthält die Daten der ersten Nachricht bitweise invertiert. Der Empfänger macht einen Kreuzvergleich und geht bei einem erkannten Fehler in den sicheren Zustand, d. h. die lokalen Ausgänge werden in den vordefinierten „sicheren“ Zustand geschaltet und die Versendung von SRDOs wird gegebenenfalls „sicher“ eingestellt. Das SRDO-Protokoll und die zugehörige Zeitüberwachung entsprechen einem Sicherheitsintegritätslevel von 3 (SIL-3) nach IEC 61508.

Eine weitere Zusatzfunktion sind die „Flying“-NMT-Master-Protokolle (Tab. 4.11). Sie dienen dazu, mehrere NMT-Master-Geräte in einem CANopen-Netzwerk zu betreiben, wobei immer nur ein NMT-Master aktiv ist. Dies wird mit speziellen Protokollen erreicht, die teilweise kein Datenfeld enthalten, damit sie von mehreren Teilnehmer gesendet werden können, ohne zu unauflösbarer Arbitrierungsproblemen führen. Ein potentieller NMT-Master fragt zuerst, ob sich bereits ein aktiver NMT-Master im Netz befindet. Bei positiver Antwort fragt er nach dessen Priorität und der Priorität aller anderen nicht-aktiven NMT-Mastern. Der Höchstpriorste antwortet immer als erster (dies wird durch entsprechend programmierte eindeutige Verzögerungszeiten sichergestellt). Die anderen NMT-Master bleiben oder werden inaktiv. Fällt der aktive NMT-Master, so bemerken die inaktiven NMT-Master dies am Ausfall des periodisch gesendeten Heartbeats. Jetzt fragen die inaktiven NMT-Master, wer die höchste Priorität hat, und der übernimmt dann das Netzwerk-Management. Falls der höherpriore NMT-Master wieder integriert werden möchte, muss er zuerst fragen, ob es einen aktiven NMT-Master gibt und wird wie oben angegeben verfahren.

Ebenfalls zu den Sonderfunktionen zählt die Netzwerk-Redundanz. Prinzipiell gibt es viele Redundanzkonzepte. In CANopen ist eine Busleitungsredundanz spezifiziert, bei der ein Gerät über zwei getrennte CAN-Schnittstellen verfügt. Falls die Default-CAN-Schnittstelle aus irgendeinem Grund defekt ist, wird auf die zweite Schnittstelle umgeschaltet. Die Kommunikation findet immer auf beiden Kanälen statt, aber nur einer ist aktiv. So wird garantiert, dass beim Umschalten auf die redundanten Busleitungen in der

Regel keine Nachricht verloren geht. Dieses Redundanzkonzept wurde für Marineanwendungen entwickelt, kann aber auch für andere Anwendungen genutzt werden.

Zu den Sonderfunktionen gehören auch die „Layer Setting Services (LSS)“, mit denen man die Node-ID und das Bit-Timing über den Bus einstellen kann. Dazu sendet der LSS-Master eine dedizierte CAN-Nachricht, die im Datenfeld unter anderem einen Multiplexer (command specifier) enthält, mit dem die folgenden sieben Bytes interpretiert werden. Sie wird von allen Geräten mit LSS-Slave-Funktion empfangen und ausgewertet. Bei der Zuweisung Node-ID sendet der LSS-Master zuerst die Vendor-ID, danach den Produktkode, die Revisionsnummer und die Seriennummer. Mit diesen vier 32-bit-Daten lässt sich ein CANopen-Gerät weltweit eindeutig adressieren. Jetzt hört nur noch dieses Gerät zu, die anderen wissen, die folgende Node-ID ist nicht für sie bestimmt ist. Das aktivierte CANopen-Gerät bestätigt den Empfang der „neuen“ Node-ID und wird sie beim nächsten NMT-Reset aktivieren. Die LSS-Dienste stellen noch weitere Funktionen zur Verfügung, mit denen man beispielsweise die Revisionsnummer und die Seriennummer eines Gerätes mit Hilfe von Suchalgorithmen herausbekommen kann. Die Einstellung des Bit-Timing wird immer seltener mit den LSS-Diensten realisiert, da viele CANopen-Geräte heutzutage über eine automatische Bitraten-Erkennung verfügen.

Testen und Prüfen

CANopen-Geräte müssen alle verpflichtenden Funktionen der Spezifikation CiA 301 erfüllen. Sie müssen also über ein Objektverzeichnis verfügen, die NMT-Slave-Zustandsmaschine implementieren, den SDO-Default-Server zur Verfügung stellen und entweder die Heartbeat-Funktion oder das Node/Life-Guarding unterstützen. Am Markt gibt es so genannte CANopen-Master, die im eigentlichen Sinne keine CANopen-Geräte sind, aber CANopen-Geräte mit NMT-Slave-Funktion betreiben können. CANopen-Manager sind dagegen vollwertige CANopen-Geräte mit einer NMT-Slave- und einer NMT-Master-Funktion. Eine weitere Unsitte sind CANopen-Geräte mit hersteller-spezifischen Parametern im Objektverzeichnis zur Einstellung der Node-ID oder noch schlimmer zur Einstellung der Datenrate. Falls solche Geräte auch über eine Store- und Restore-Funktion verfügen, ist es möglich, dass beim Wiederherstellen der Werkseinstellungen die konfigurierte Node-ID bzw. die konfigurierte Bitrate überschrieben werden. Werden die Werkseinstellungen „versehentlich“ in einem laufenden System wieder hergestellt, so führt dies im Fall der Node-ID „nur“ zum Ausfall eines Gerätes oder dazu, dass zwei Geräte die gleiche Node-ID verwenden. Bei der Bitrate wird die gesamte Kommunikation dauerhaft gestört. Solche Geräte passieren in der Regel den CANopen-Konformitätstest nicht.

Der CANopen-Konformitätstest ist nicht vorgeschrieben, aber die Tests sind im Konformitätstestplan standardisiert (CiA 310). Konformitätstests dienen zur Prüfung einer CANopen-Schnittstelle, so ähnlich wie eine Rechtschreib- und Grammatikprüfung in der Textverarbeitung. Es wird nur die formale Korrektheit der statischen Kommunikationschnittstelle getestet. Möchte man die Dynamik sowie die Interoperabilität von Geräten prüfen, so ist ein Systemtest notwendig. Dieser kann in einem „golden“ Netzwerk mit ausgesuchten Geräten oder spontan bei so genannten Plug-Fests durchgeführt werden.

Sicherlich erhöht ein bestandener Konformitätstest die Interoperabilität zu anderen konformen CANopen-Geräten, aber das dynamische Verhalten und das Verhalten unter Stress muss separat ermittelt werden.

4.2.6.5 Devicenet

Hierbei handelt es sich um eine ursprünglich von Rockwell Automation definierte CAN-basierende Anwendungsschicht, die heute von der ODVA (Open Devicenet Vendor Association) weiterentwickelt und gewartet wird. Sie ermöglicht Master/Slave-, Multi-Master- und Peer-to-Peer-Kommunikation. Die Adressierung erfolgt über ein Adressierungsschema mit mehreren Ebenen. Dabei sind folgende Informationen enthalten:

- Geräteadresse: Dieser Wert identifiziert den Netzknoten eindeutig.
- Klassenkennung: Sie identifiziert die Klasse, zu der das angesprochene Objekt zugeordnet wird.
- Instanz-Kennung: Die Instanz-Kennung identifiziert ein Objekt innerhalb einer Klasse.
- Attributkennung: Diese identifiziert ein Attribut, also einen Parameter, eines Objekts.
- Service Code: Der Service Code zeigt einzelne besondere Funktionen eines Objektes an.

Durch die Definition einheitlicher Gerätemodelle wird die Interoperabilität verbessert. Einfache Geräte von unterschiedlichen Herstellern sind sogar untereinander austauschbar.

Die Devicenet Architektur

Das Devicenet Netzwerk legt eine Topologie mit Stamm- und Stichleitungen fest. Stromversorgung und Daten laufen über diese Stamm- und Stichleitungen. Geräte können daher direkt vom Bus betrieben werden und miteinander über dasselbe Kabel auch kommunizieren (Geräte mit eigener Stromversorgung sind vom Netzwerk auf der Transceiver-Ebene entkoppelt). Bis zu 64 logische Netzketten können in einem Devicenet-Netzwerk verbunden sein, wobei sowohl geschlossene als auch offene Steckverbinder erlaubt sind.

Es werden sowohl getaktete, durch „Polling“ abgefragte, periodische, durch Zustandsänderungen als auch durch die Anwendung ausgelöste Datenübertragungen unterstützt. Der Anwender kann je nach Gerät und Anwendung eine Master/Slave, Multi-Master, Peer-to-Peer oder eine kombinierte Beziehung konfigurieren. Diese Wahl der Datenübertragung kann zu einer Reduzierung der Antwortzeit des Systems beitragen.

Die Anwendungsebene

Die Devicenet-Spezifikation legt fest, wie CAN-Identifier (und damit Prioritäten) bestimmt werden und wie das CAN-Datenfeld bei den unterschiedlichen Devicenet-Diensten genutzt wird. Diesen Diensten die CIP-Protokolle zugeordnet. Im Gegensatz zur traditionellen Quelle-Ziel-Adressierung verwendet das CIP-Protokoll bei Devicenet das Producer/Consumer-Modell.

Das Gerät mit einem Sendewunsch platziert (oder produziert) die Daten mit dem geeigneten CAN-Identifier und überträgt ein entsprechendes CAN-Datenframe. Alle Geräte, die Daten benötigen, hören auf Meldungen. Erkennen Geräte einen von ihnen zu akzeptierenden CAN-Identifier, agieren sie und konsumieren somit die Daten.

Beim Producer/Consumer-Modell ist die Meldung nicht mehr für eine besondere Quelle oder ein besonderes Ziel bestimmt. Eine einzelne Meldung von einer Steuerung kann zum Beispiel von mehreren Antrieben verwendet werden (Multicast). Dadurch sind eine effizientere Übertragung und eine besser koordinierte Antwort mit geringerem Bedarf an Übertragungsbandbreite möglich.

Devicenet unterscheidet prinzipiell zwei Arten von Geräte-Nachrichten:

- *E/A-Meldungen*: Diese Meldungen sind für zeitkritische steuerungsorientierte Daten bestimmt. Sie schaffen dedizierte, für einen bestimmten Zweck benötigte Kommunikationspfade zwischen einer produzierenden Anwendung und einer oder mehreren konsumierenden Anwendungen. Sie werden in einzelnen oder mehrfachen Verbindungen ausgetauscht und verwenden normalerweise Identifier von hoher Priorität.
- *Explizite Meldungen*: Diese Meldungen schaffen Mehrzweck-Punkt-zu-Punkt-Kommunikationspfade zwischen zwei Geräten. Sie ermöglichen die typische Frage/Antwort-Netzwerkkommunikation, die für die Konfiguration der Netzketten und für die Problemdiagnose erforderlich ist. Für explizite Meldungen werden in der Regel CAN-Identifier mit niedriger Priorität verwendet.

Bei Meldungen, die länger als 8 Byte sind, z. B. beim Hoch-/Herunterladen einer Textanzeige oder bei der Übertragung von Parametersätzen, wird der Fragmentierungsdienst aktiv. Im Rahmen des Devicenet-Netzwerks werden die Verbindungen in Form eines abstrakten Modells definiert, das die verfügbaren Kommunikationsdienste darstellt und das von außen sichtbare Verhalten des Devicenet-Gerätes beschreibt.

Die vordefinierte Einstellung einer Master/Slave-Verbindung

Obwohl Devicenet über ein leistungsstarkes anwendungsorientiertes Protokoll verfügt, das ein dynamisches Konfigurieren der Verbindungen zwischen den Endgeräten ermöglicht, hat man festgestellt, dass einige Geräte weder den Bedarf noch die Ressourcen haben, diese Leistungsfähigkeit zu nutzen. Aus diesem Grund wurde eine Reihe von CAN-Identifiern – bekannt als Predefined Master/Slave Connection Set (vordefinierte Einstellung einer Master/Slave Verbindung) – festgelegt, um die Bewegung der E/A- und Konfigurationstypdaten, die für eine Master/Slave-Architektur typisch sind, zu vereinfachen (siehe Abb. 4.41).

Viele, wenn nicht sogar die meisten Sensor/Aktor-Geräte, werden für vorher festgelegte Funktionen (Druckmessung, Starten eines Motors etc.) konzipiert, und Art und Menge der Daten, die das Gerät produzieren und/oder konsumieren wird, wird beim Start erkannt.

Diese Geräte liefern üblicherweise Eingangsdaten oder erfordern Ausgangsdaten und Konfigurationstypdaten. Die vordefinierte Master/Slave-Verbindung erfüllt diese Bedürfnisse.

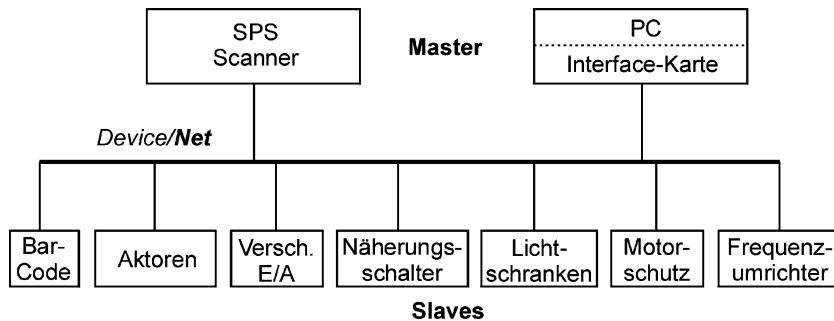


Abb. 4.41 Typische Master/Slave-Architektur

nisse durch Bereitstellung von Verbindungsobjekten, die beim Start des Geräts fast vollständig konfiguriert sind. Das Master-Gerät muss nur noch seinen „Besitzanspruch“ auf diese vordefinierte Einstellung seiner Slave-Geräte erheben, um den Datenfluss einzuleiten.

Devicenet-Adressierung

Um auf interne Komponenten/Logik innerhalb eines Geräts zuzugreifen, definiert das Devicenet-Kommunikationsmodell ein Adressierungsschema, das den Zugang zu Objekten innerhalb eines Geräts ermöglicht. Diese Informationen sind physikalisch innerhalb des Devicenet Protokolls repräsentiert (siehe folgende Abb. 4.42).

Die Verwendung des Adressierungsschemas ermöglicht eine einfache, aber robuste Definition der Geräte, die eine große Bandbreite der Produktfunktionalität und der Kosten abdeckt.

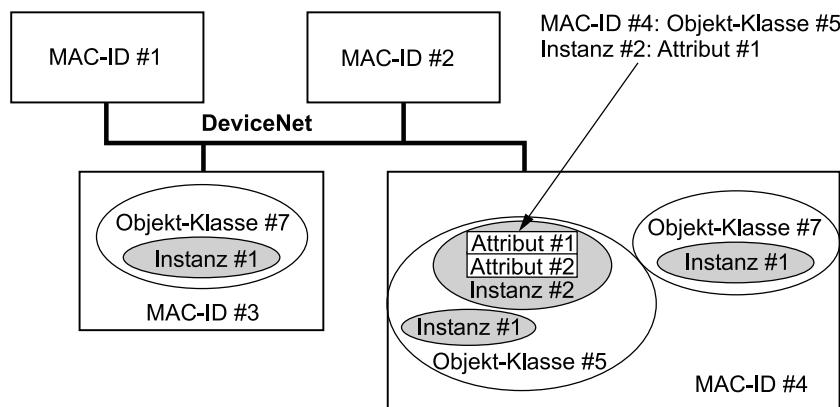


Abb. 4.42 Devicenet-Adressierungsschema

In den Adressierungsinformationen für das betreffende Objekt sind enthalten:

- **Geräteadresse**: Auch als MAC-ID (*Media Access Control Identifier*) bezeichnet. Dabei handelt es sich um einen ganzzahligen Identifikationswert, der für jeden Netzknopen im Devicenet festgesetzt ist. Dieser Wert unterscheidet einen Netzknopen von allen anderen Netzknopen derselben Verknüpfung. Eine Prüfung beim Einschalten garantiert die Einmaligkeit dieses Wertes im Netzwerk (doppelte MAC-ID-Erkennung).
- **Klassenkennung (Class ID)**: Der Begriff „Klasse“ bezeichnet eine Reihe von Objekten, die zum selben Typ von Systemkomponenten gehören. Die Klassenkennung ist ein ganzzahliger Identifikationswert, der jeder vom Netzwerk aus zugänglichen Objektklasse zugewiesen worden ist. Beispiele für Objektklassen sind diskreter Eingang, Identität, übersetzen und analoger Ausgang.
- **Instanz-Kennung (Instance ID)**: Der Begriff „Instanz“ bezeichnet eine aktuelle Darstellung eines Objekts innerhalb einer Objektklasse. Die Instanz-Kennung ist ein ganzzahliger Wert, der einer Objektinstanz zugewiesen worden ist, und der diese Objektinstanz unter allen anderen Instanzen derselben Klasse innerhalb eines bestimmten Geräts identifiziert (der Wert für die Instanzkennung ist einmalig). So hat ein 24-V-Gleichspannungs-Eingangsblock mit 8 Eingängen 8 Instanzen des diskreten Eingangsobjekts.
- **Attributkennung (Attribute ID)**: Attribute sind Parameter, die mit einer Objektklasse und/oder einer Objektinstanz assoziiert sind. Attribute enthalten in der Regel einen bestimmten Typ von Statusinformationen, repräsentieren eine Eigenschaft mit einem konfigurierbaren Wert oder steuern die Bedienung eines Objekts. Die Attributkennung ist ein ganzzahliger Identifikationswert, der einem Objektattribut zugewiesen worden ist. Attributbeispiele sind u. a. Verzögern des Eingangsfilters, Beschleunigungsrate, Alarm, Sollwertüberlastauslösung und Druck.
- **Service Code**: Ein vorhandener Identifikationswert, der eine besondere Objektfunktion anzeigt. Beispiel sind u. a. Attribut_leSEN (Get_Attribute), Attribut_schreiben (Set_Attribute) und Zurücksetzen (Reset).

Geräteprofile

Die Devicenet-Spezifikation geht über eine Spezifikation für ein physikalisches Verbindungsprotokoll hinaus. Sie fördert durch die Festlegung von einheitlichen Gerätmodellen die Interoperabilität. Alle Geräte desselben Modells müssen die allgemeinen Identitäts- und Kommunikationsstatusdaten unterstützen. Gerätespezifische Daten sind in den Gerätprofilen enthalten, die für verschiedene Gerätetypen definiert sind. Einfache Geräte (z. B. Drucktaster, Anlasser, Photozellen, Druckventile usw.) von verschiedenen Herstellern, die mit ihren Gerättypprofilen übereinstimmen, sind untereinander logisch austauschbar. Um Erweiterungen nicht einzuschränken, ist es weiterhin möglich, herstellerspezifische, wertsteigernde Merkmale hinzuzufügen.

Besondere Anwendungsvorteile

Devicenet eignet sich gut sowohl für die Verbindung einfacher Geräte mit dem Steuerungssystem als auch für intelligente Geräte, z. B. den Motorschalter, der eine Laststromrückkopplung für eine bessere Prozesssteuerung bietet. Devicenet bietet außerdem bessere Fehlersuchkapazitäten, weil ein Kommunikationsfehler bis zum einzelnen Gerät zurückverfolgt werden kann – im Gegensatz zum Chassis oder zur Blockebene in einem E/A-Netzwerk. Anders als festverdrahtete Ein-/Ausgänge, die der Steuerung lediglich mitteilen können, ob der Sensor an oder aus ist, ist ein Devicenet-Netzwerk mit intelligenten Geräten in der Lage, eine höheren Informationsgrad zu liefern. So kann zum Beispiel ein Bediener zwischen einem ausgeschalteten und einem defekten Sensor unterscheiden.

Heutige Devicenet-Scanner unterstützen häufig eine Eigenschaft, die von Anwendern sehr geschätzt wird: Den Austausch defekter Devicenet-Geräte ohne Neukonfigurieren (ADR – Automatic Device Replacement). Die SPS-Scanner merken sich die aktuellen Konfigurationsparameter aller angeschlossenen Devicenet-Geräte. Wird nach einem Ausfall ein neues (gleichartiges) Ersatzgerät installiert, so reagiert der Scanner und lädt die gespeicherten Parameter des ausgetauschten Gerätes automatisch in dieses Ersatzgerät hinunter – ein Konfigurationstool wird in diesem Fall nicht benötigt.

Unterscheidungsmerkmale von Devicenet

Es gibt einige Merkmale, die Devicenet von anderen CAN-Lösungen unterscheiden. Die Geräte in einem Devicenet-Netzwerk sind für die Verwaltung ihrer eigenen CAN-Identifier verantwortlich. Alle Netzknoten haben eine ganze Reihe von Meldungsprioritäten, die ihnen unabhängig von ihren MAC-IDs zur Verfügung stehen. Aufgrund des Algorithmus zur Erkennung von doppelten MAC-IDs ist die Einmaligkeit der CAN-Identifier garantiert, ohne dass ein zentrales Werkzeug oder eine Registrierung für jedes einzelne Netzwerk erforderlich ist. Mit diesem Mechanismus werden doppelt adressierte Geräte sofort erkannt – anders als bei andern CAN-Netzwerken (besser, einer doppelten Adressierung gleich vorzubeugen als sie später erst mühsam zu lokalisieren).

Ein weiterer erheblicher Vorteil bei Netzknoten mit eigenen Identifier besteht darin, dass der Anwender Netzknoten hinzufügen oder entfernen und/oder zusätzliche Peer-to-Peer-Meldungen bei den vorhandenen Netzknoten jeder Zeit hinzufügen kann, ohne die bestehende Konfiguration zu kennen. Es ist nicht nötig, eine zentrale Registrierung zu lokalisieren oder zu rekonstruieren. Da Netzknoten wissen, welche IDs bereits verwendet werden, muss ein Werkzeug einfach eine zusätzliche E/A-Verbindung zwischen zwei Geräten anfordern, wobei der Prioritätsgrad, der Datenpfad (Klasse, Instanz, Attribut) und der Produktionsauslöser (zyklisch, abgefragt, Zustandsänderung) festzulegen sind.

Die Geräte tauschen dann Identifier aus, und die Verbindung wird konfiguriert und aktiv. Das Werkzeug kann außerdem einen spezifischen Identifier anfordern, und falls dieser nicht schon verwendet wird, wird das Gerät zustimmen.

Unterstützung von Master/Slave und Peer-to-Peer Der Devicenet-Standard definiert mehrere Verfahren der Datenübertragung für explizite und E/A-Meldungen. Der E/A-

Austausch kann getaktet, zyklisch, durch Zustandsänderung oder durch die Anwendung ausgelöst sein. Die Definitionen der Datenübertragung ermöglichen den Devicenet-Netzknoten, eine Vielfalt von Anwendungen zu implementieren, darunter Master/Slave, Multi-Master und Peer-to-Peer – entweder getrennt voneinander oder alle auf derselben Leitung.

Effizienz und Flexibilität Das Devicenet-Design erlaubt eine effiziente und flexible Datenverwaltung. Das ist wichtig, wenn an das Netzwerk viele verschiedene Gerätetypen von unterschiedlicher Komplexität angeschlossen sind. Die Verbindungsorientierung von Devicenet ermöglicht den Kommunikationsgeräten, die mit dem E/A-Austausch assoziierten Eigenschaften vor dem Datenaustausch vorzudefinieren. Das heißt, Devicenet hat keinen Protokollaufwand innerhalb des CAN Datenfeldes für eine E/A-Meldung von 8 Byte oder weniger.

Im Devicenet ist außerdem ein Fragmentierungsdienst für E/A-Meldungen definiert, der bei der Übertragung von größeren E/A-Datenpäckchen (über 8 Byte) aktiv wird. Jedes Fragment einer E/A-Meldung enthält nur ein einzelnes Byte zur Verwaltung des Fragmentprotokolls. Die Anzahl der Fragmente ist nicht begrenzt.

Die Fragmentierung ist auch für explizite Meldungen definiert. Diese Flexibilität garantiert dem Devicenet-Anwender, dass neue Geräte seinem bestehendem Devicenet-Netzwerk hinzugefügt werden können, selbst wenn die Geräte technisch weiterentwickelt sind oder heute relativ einfache Geräte mit erweiterten Kapazitäten ausgestattet werden. Mit seinem objektbezogenen Design und seinem Adressierungsschema kann Devicenet praktisch unbegrenzt erweitert und neuen Bedürfnissen angepasst werden, ohne dass das Basisprotokoll und das Verbindungsmodell zu ändern sind.

Am anderen Ende des Spektrums kann ein einfacher Slave mit zwei Meldungsanschlüssen (E/A- und expliziter Anschluss) die vollständige Implementierung einschließlich der Geräteanwendung mit weniger als 4 KiB ROM und 175 Byte RAM abwickeln (Motorola 68HC05X4, eine CPU mit eingebauter CAN-Schnittstelle).

Wiederherstellung der Kommunikation bei Fehlern Devicenet verfügt über ein robustes Kommunikationsmodell in Bezug auf Fehler, die während einer Kommunikation auftreten können. Es verfügt über Einrichtungen, die die Kommunikation wiederherstellen, wenn ein Fehler auftritt.

Stromversorgung Der Devicenet-Standard ist in der Lage, Stromversorgungsanschlüsse überall im Netzwerk hinzufügen zu können. Daraus ergeben sich zwei deutliche Vorteile. Erstens: Der Anwender kann redundante Netzteile haben, falls die Anwendung das fordert. Zweitens: Mit einer mit 8 A ausgestatteten Hauptleitung kann eine beachtliche Energiemenge für nichtisolierte Geräte im Netzwerk bereitgestellt werden. Die Option der isolierten Geräte ist wichtig, weil so Geräte mit hohem Energieverbrauch wie Antriebe, Starter und Magnete auf demselben Netzwerk koexistieren können. Andere auf CAN

basierende Netzwerke lassen nur ein Netzteil (wenn überhaupt) für das gesamte Netzwerk zu.

Profile Im Devicenet-Standard ist ein so genanntes elektronisches Datenblatt (Electronic Data Sheet, EDS) definiert. Dabei handelt es sich um ein einfaches Dateiformat, mit dem die produktsspezifischen Informationen eines Produkts allen Verwendern bereitgestellt werden können. Das ermöglicht anwenderfreundliche Konfigurationswerkzeuge, die leicht aktualisiert werden können, ohne die Konfigurationssoftware-Werkzeuge ständig überarbeiten zu müssen.

Allgemein/herstellerspezifisch Die Devicenet-Spezifikation schafft ein ausgewogenes Verhältnis zwischen allgemeinen Objektklassen, Diensten und Attributen, die alle in der Devicenet-Spezifikation definiert sind, und hersteller-spezifischen Objektklassen, Diensten und Attributen, die von einzelnen Herstellern hinzugefügt werden können. Dadurch können Hersteller ihren Kunden zusätzliche Funktionen bieten, die in der Spezifikation nicht enthalten sind. Werden diese herstellerspezifischen Ergänzungen allgemein üblich, so tritt ein Übertragungsmechanismus in Kraft und die herstellerspezifischen Funktionen werden verallgemeinert.

4.2.7 FOUNDATION Fieldbus H1

4.2.7.1 Übersicht

FOUNDATION Fieldbus H1 ist ein für die Prozessindustrie entwickelter Feldbus, der oft einfach nur ‚FF‘ genannt wird. Der FF verbindet die Leittechnik mit den Feldgeräten und überträgt auf einer Zweidrahtleitung Daten und Speiseenergie von einem Leitrechner an die Feldgeräte. Dabei erfüllt der FF die in der Prozessautomation geforderten Randbedingungen:

- Langzeitstabilität für kontinuierlich laufende Prozesse
- Dimensionierung für raue Umgebungsbedingungen mit weiten Temperaturbereichen, aggressive Medien und hohe EMV Belastung
- Explosionsschutz in allen explosionsgefährdeten Bereichen sowie
- Lange Kabelwege von bis zu 1900 m und eine hohe Zahl von bis zu 31 Teilnehmern je elektrisches Segment.

In der Prozessindustrie spielt die Explosionsschutzart Eigensicherheit (Ex i) eine besondere Rolle. Sie erlaubt den Eingriff in die Anlage im laufenden Betrieb ohne Heißarbeits-erlaubnisschein.

Der FF ist als offener Feldbus definiert und ist in der Prozessindustrie weltweit¹ allgemein akzeptiert und eingeführt. Viele Hersteller bieten Feldgeräte und Leittechnikkomponenten mit FF-Funktionalität an. Der FOUNDATION Fieldbus H1 wird durch die Fieldbus Foundation verwaltet. Die Fieldbus Foundation verwaltet und pflegt die Definition von Feldbusphysik und Protokoll. Sie prüft und zertifiziert Baugruppen und Feldgeräte in Bezug auf Konformität und Interoperabilität. Damit ist sichergestellt, dass Feldgeräte unterschiedlicher Hersteller harmonisch und fehlerfrei in einem System zusammengeschaltet werden können.

Der Feldbus überträgt Messwerte, auch mehrere Messwerte von multivariablen Sensoren – und Steuerkommandos digital, das heißt ohne Drift der Messstrecke. Zusätzlich senden Feldgeräte Diagnoseinformationen über ihren eigenen ‚Gesundheitszustand‘ und empfangen Konfigurationsdaten über den Bus. Mit dem FOUNDATION Fieldbus H1 lassen sich Installation, Inbetriebnahme und die Überwachung im laufenden Betrieb effizient und pro-aktiv realisieren.

Der FOUNDATION Fieldbus H1 implementiert die Ebenen 1, 2 und 7 nach dem ISO/OSI-Modell (Vergleiche Abschn. 1.2.1). Dieser Abschnitt beschreibt die Feldbusphysik und deren Hauptbegriffe (Ebene 1) und Komponenten, Buszugang (Ebene 2) und Implementierung von Geräteinformationen (Ebene 7) sowie Motivation und Vorteile aus Sicht der Anwendergruppen vom Planer über Installateur und Inbetriebnehmer bis zu Anlagenfahrer und Wartungsfachmann.

4.2.7.2 Die Feldbusphysik

Komponenten

Der FOUNDATION Fieldbus H1 überträgt Daten und Stromversorgung über eine geschirmte, verdrillte Zweidrahtleitung. Ein Netzwerk, der elektrische Stromkreis, der eine Leittechnikausgangskarte mit mehreren Feldgeräten verbindet, heißt Segment. Standard und Richtlinien lassen alle klassischen Topologien wie Baum-, Bus- oder Sternstruktur zu (Siehe auch Kap. 1). In der Praxis hat sich eine Topologie mit Stamm- (engl.: Trunk) und Stichleitungen (engl.: Spur) durchgesetzt. Sie wird auch Trunk-and-Spur-Topologie genannt. Sie ist besonders übersichtlich, einfach zu installieren und erfüllt viele Bedingungen einer Prozessautomation auch im explosionsgefährdeten Bereich (Abb. 4.43).

Von der in der Leitwarte installierten Busspeisung führt der typischerweise lange Trunk zu einem Feldverteiler (engl.: Junction Box). Der Spur führt vom Verteiler bis zum Feldgerät. Aus Gründen der Übersichtlichkeit und Verfügbarkeit wird je Spur nur ein Feldgerät angeschlossen. Die Verdrahtungskomponente im Feldverteiler kann optional mit einem Kurzschlusschutz ausgestattet sein. Dieser verhindert, dass sich Kurzschlüsse im Feldge-

¹ Speziell in Deutschland erfreut sich der PROFIBUS PA, das „Wettbewerbsprodukt“ von PROFIBUS International größerer Beliebtheit. Dies liegt darin begründet, dass die Wahl des Bussystems eine abhängige Entscheidung ist, die mit der Wahl des Leitsystems getroffen wird. In Deutschland haben Leitsystemanbieter, die PROFIBUS PA bevorzugen, einen höheren Marktanteil.

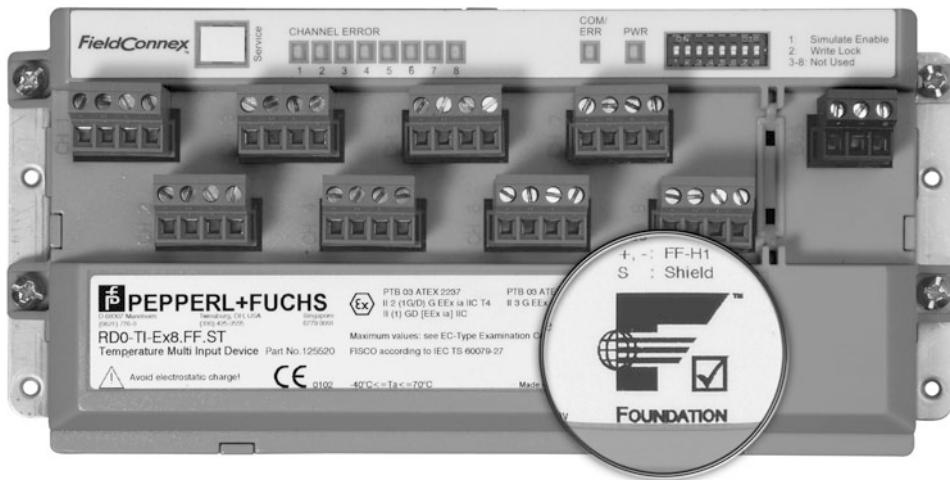


Abb. 4.43 Für den FOUNDATION Fieldbus H1 zertifiziertes Feldgerät

rät, die insbesondere beim Setzen oder Entfernen von Feldgeräten entstehen können einen negativen Einfluss auf das Segment haben und damit den Anlagenbetrieb gefährden.

Jeweils am Ende des Trunks wird ein Terminator angeschlossen. Dieser ist häufig in die Feldbusstromversorgung und den Feldverteiler integriert. Der Terminator hat eine Impedanz von 100Ω bestehend aus einem Widerstand und einer Kapazität und erfüllt zwei Aufgaben:

1. Die Unterdrückung von Reflektionen am Ende der Leitung.
2. Übersetzung des als Stromänderung übertragenen Datensignals in eine für alle Teilnehmer übersetzbare Spannung.

Kommunikation, Installation und Diagnose

Die Kommunikationssignale werden mit Manchester-II-Kodierung übertragen. Jeweils die Flanke enthält logisch „1“ und „0“ (Abschn. 1.6.5). Die Manchester-II-Kodierung zeichnet sich durch eine sehr hohe Stabilität bei der Übertragung aus. Die Datenübertragungsrate beträgt 31,25 kbit/s. Störungen wie etwa Rauschen oder Stromstöße, beispielsweise verursacht durch Energieleitungen in der Nähe, wirken sich nur gering auf die Kommunikation aus:

Der Empfänger detektiert zu einem taktsynchronen Zeitpunkt die Flanke. Zwischen den Flanken eingestreute Störungen werden ignoriert. Um ein Bit tatsächlich zu zerstören, müsste das Störsignal eine sehr hohe Flankensteilheit und -amplitude aufweisen. Voraussetzung für diese Störfestigkeit ist eine gute Installation mit hoher Isolation beider Leiter gegen Schirmung und Erde. Des Weiteren ist ein für die Anlage passendes Erdungs- und Schirmungskonzept zu planen und umzusetzen.

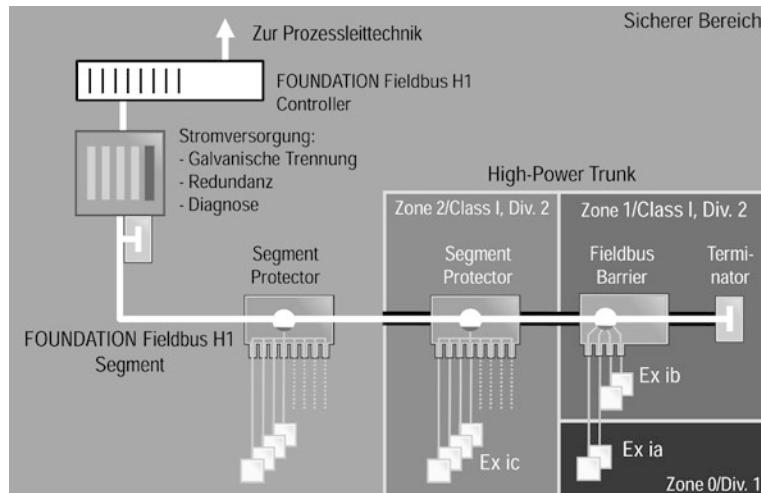


Abb. 4.44 Trunk- und Spur-Topologie für den FOUNDATION fieldbus H1 ist auch für den explosionsgefährdeten Bereich (Ex ia, Ex ic) geeignet

In einer Prozessanlage kommt es auf eine sehr gute Verfügbarkeit der Kommunikation an. Eine Betrachtung der Feldbusphysik selbst ermöglicht die Erfassung und Messung von dessen Qualität. Die Feldbusphysik wird durch folgende elektrische Größen erfasst:

- Versorgungsspannung und -Strom: Von der Feldbusspeisung erzeugte Spannung und Laststrom zur Versorgung der Feldgeräte.
- Unbalance (Erdfehler): Eine Differenz der Isolation zwischen den Leitern bezogen auf das Erdpotential. Dabei bedeutet dabei einen harten Kurzschluss des Minus-Leiters gegen Erde, +100 % entsprechend für den Plus-Leiter. 0 % zeigt eine gleichmäßige Isolation und damit die geringste Störanfälligkeit an.
- Signalpegel: Spannungspegel des Datensignales. Das als Stromänderung von + / - 9 mA übertragene Signal wird von den beiden Terminatoren von jeweils 100Ω in einen Spannungspegel übersetzt, der von allen Teilnehmern detektiert werden kann (Abb. 4.44).
- Rauschen: Eine dem Signal überlagerte Störspannung (Abb. 4.44).
- Jitter: Abweichung des Zeitpunktes der Signalflanke vom optimalen Zeitpunkt (Abb. 4.45).

Der Jitter ist besonders geeignet, um die Qualität einer Feldbusinstallation zu detektieren: An einem Segment sind alle Komponenten über Kabel parallel geschaltet. Eine Veränderung der Feldbusphysik wird deswegen immer eine Änderung der elektrischen Eigenschaften nach sich ziehen. Beispielsweise sorgt Über- oder Unterterminierung für eine Änderung der Gesamtkapazität des Segmentes. Diese Verschiebung verursacht eine Veränderung der Signalform vom idealen Rechtecksignal zu einer stärker gedämpften oder stär-

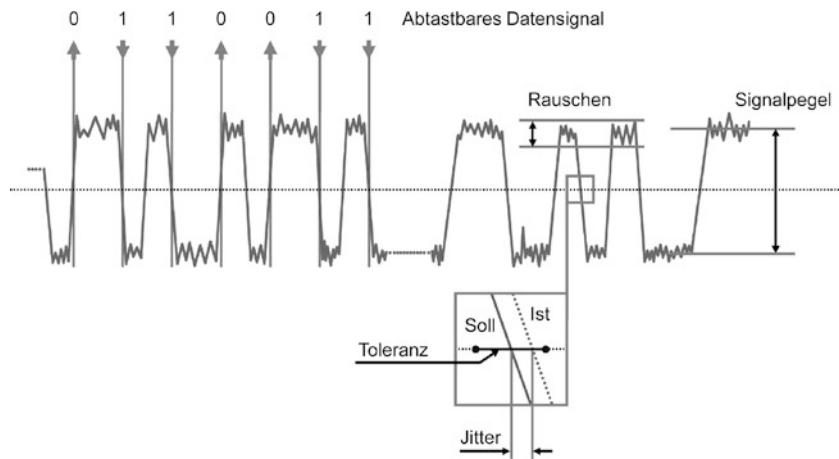


Abb. 4.45 Elektrische Messgrößen – Größen des Feldbussignals

ker schwingenden Signalform – und damit eine Veränderung des Flankennulldurchgangs. Moderne Diagnosewerkzeuge können diese Änderung detektieren und machen so auf eine Verschlechterung der Feldbusinstallation aufmerksam.

Der größte Störfaktor für den Feldbus nach qualitativ guter Installation ist jedoch nicht die Umgebung: Energiekabel, Motoren, Umweltbedingungen. Studien zeigen, dass eine ungewollte Veränderung am Feldbus durch den manuellen Eingriff, zum Beispiel beim Austausch eines Gerätes, die Störempfindlichkeit der Feldbusinfrastruktur erhöht. Speziell für den Feldbus entwickelte Diagnosefunktionen schaffen Abhilfe.

Explosionsschutz

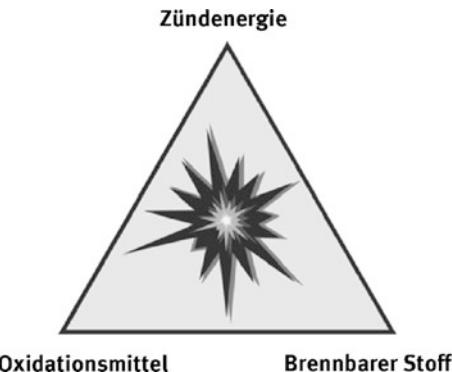
Für explosionsgefährdete Bereiche ist der FF ausgelegt. Beim Explosionsschutz spricht man von drei Elementen, die zu einer Explosion notwendig sind (Abb. 4.46).

- Explosives Gas oder Staub
- Sauerstoff
- Wärmequelle

Die Methoden des Explosionsschutzes sorgen für die Abwesenheit von mindestens einer dieser drei Stoffe. In der Prozessautomatisierung ist dies in der Regel die Wärmequelle in der Form einer warmen Oberfläche oder eines Funkens. Zwei Schutzarten haben sich in der Prozessautomation durchgesetzt (Tab. 4.12):

Die Schutzart „Eigensicherheit“ (Ex i) realisiert eine Leistungsbegrenzung auf ca. 2 W. Elektrische Funken können beim Öffnen eines Stromkreises nur so wenig Energie in Wärme umsetzen, dass eine Explosion sicher verhindert wird. Vorteil: Es darf an einem eigensicheren Segment oder Gerät stets und ohne einen Heißarbeitserlaubnisschein gear-

Abb. 4.46 Explosionsdreieck.
(Nur beim Vorhandensein aller dreier Komponenten kommt es zu einer Explosion)



Tab. 4.12 Methoden des Explosionsschutz mit Relevanz für den Feldbus nach IEC/EN 60079

Methode	Zone	Grundlegende Merkmale
Druckfeste Kapselung Ex „d“	1, 2	Relativ problemlos anwendbar, jedoch mit bestimmten mechanischen Anforderungen. Aufwand bei Wartungen und Prüfungen.
Erhöhte Sicherheit Ex „e“	1, 2	Verwendbar für Betriebsmittel, die bei Normalbetrieb nicht funken (Anschlussgeräte, Klemmen, Lamenfassungen, Motoren). Besondere Anforderungen an die Konstruktion.
Eigensicherheit Ex „ia“	0, 1, 2	Ideal für Prozessinstrumentierung. Einfache Installation, Wartung und Prüfung während des Betriebes. Begrenzung auf niedrige Leistung, sicher auch bei zwei Fehlern
Eigensicherheit Ex „ib“	1, 2	Ähnlich wie Ex „ia“, sicher bei einem Fehler
Eigensicherheit Ex „ic“	2	Ähnlich wie Ex „ia“, sicher Normalbetrieb

beitet werden. Nachteilig ist die geringe zur Verfügung stehende Leistung, die nur sehr wenige Feldgeräte je Segment erlaubt.

Die Schutzarten „Druckfeste Kapselung“ (Ex d) und „erhöhte Sicherheit“ (Ex e) verhindern ein ungewolltes Öffnen eines Stromkreises. Vorteil: Es kann eine hohe Leistung übertragen werden. Nachteile: Die Installationskosten sind im Vergleich zur Eigensicherheit höher. Für das Arbeiten am Segment oder Feldgerät ist eine Heißarbeitserlaubnis erforderlich.

Die Zoneneinteilung erlaubt dem Anwender eine Zuordnung der Anlage zu einer bestimmten Eintrittswahrscheinlichkeit für die Präsenz von zündfähiger Atmosphäre. Zusätzlich erhalten zugelassene Produkte Angaben über die minimal erforderliche Zündenergie in Form einer Gasgruppe (IIA...IIC) und einer Temperaturklasse (T 1...6), die die maximal mögliche Oberflächentemperatur definiert.

Der High-Power Trunk kombiniert die positiven Eigenschaften einer hohen Leistung und der Eigensicherheit am Feldgerät durch eine gekonnte Mischung aller oben beschriebenen Methoden. Dabei wird der Trunk in geschützt verlegt und eine hohe Leistung von typischerweise 28–30 V und 500 mA gespeist. Damit sind lange Kabelwege und hohe

Gerätezahlen gleichzeitig möglich. Arbeiten am Trunk erfordern eine Heißarbeitserlaubnis – sind in aller Regel aber auch nicht erforderlich.

Die „Umsetzung“ des Segments auf die Schutzart Eigensicherheit erfolgt im Feldverteiler. Hier kommen je nach Gefährdungsgrad unterschiedlich ausgerüstete Komponenten, Segment Protektor oder Feldbarriere genannt, zum Einsatz. Der Ausgang des Feldverteilers ist eigensicher für den Anschluss eigensicherer Feldgeräte. Am Spur und Feldgerät darf ohne Heißarbeitserlaubnis gearbeitet werden. Mit dem High-Power Trunk Konzept erlangte der FF eine hohe Akzeptanz in der Prozessindustrie.

Die Revolutionierung der Eigensicherheit – DART Technologie

In der weiteren Entwicklung wurden in neuester Zeit auch die Leistungsgrenzen der Eigensicherheit überwunden. Hintergrund ist eine dynamische Erkennung eines Funkens. DART steht für Dynamic Arc Recognition and Termination. DART ermöglicht eine höhere Ausgangsleistung und ist nach IEC 60079-11 eigensicher Ex ib IIC zertifiziert. In FF-Applikationen kann der Trunk selbst durch Stromversorgung und Feldverteiler mit Hilfe von DART geschützt werden. Für Leittechnik und Feldgeräte bleibt der Anschluss unverändert. Heutige eigensichere Feldbusfeldgeräte können an einem DART Feldbus betrieben werden. Somit lässt sich nun auch der Feldbusstrunk eigensicher ausführen bei trotzdem langen Kabellängen und Gerätezahlen.

4.2.7.3 Die Kommunikation

Buszugriff und Datenübertragung

Für den Zugriff auf den Bus wird die Busarbitration verwendet (Abschn. 1.3.6). Diese ermöglicht die Kommunikation zwischen allen Teilnehmern eines Segmentes, auch zwischen Feldgeräten selbst.

Über den Bus werden zwei prinzipiell verschiedene Arten von Nachrichten übertragen. Die zyklischen Daten (Scheduled Messages) enthalten die zur Steuerung einer Anlage notwendigen Daten. Sie werden in einem vorbestimmten, festen Zeitraster übertragen. Die azyklischen Daten (Unscheduled Messages) dienen z. B. zur Parametrierung und Überwachung des Systems. Die Übertragung von azyklischen Daten wird vom jeweiligen Teilnehmer beim Bussteuerwerk angemeldet, das dann dafür eine Zeit zuweist, die nicht von einer zyklischen Übertragung belegt ist.

Anwendungsschicht

Die Anwendungsebene des FF basiert auf der „Fieldbus Message Specification“ (FMS). Die FMS codiert und decodiert die Aufrufe der Anwendung. Der Zugriff auf Objekte kann sowohl über Indizes als auch über ihre Namen (Tag Nummern) erfolgen.

Der User Layer des FF unterstützt die Implementierung von verteilten Steuerungssystemen. Der User Layer gliedert sich im Wesentlichen in drei Elemente. Die Funktion der einzelnen Busteilnehmer wird über Function Blocks beschrieben, während das System Management für das Verhalten des gesamten Kommunikationssystems verantwortlich ist.

Die Device Description (Gerätebeschreibung) schließlich ist eine detaillierte Beschreibung der Gerätefunktion, die in einer herstellerunabhängigen Sprache verfasst ist.

Function Blocks

Funktionsblöcke beschreiben die Funktion und das Verhalten von Geräten in standardisierter Form. Jedes Feldgerät enthält je nach Funktionsumfang einen oder mehrere Funktionsblöcke, die über den Bus mit Funktionsblöcken anderer Geräte verbunden werden um im System Steuerungs- und Regelungsfunktionen auszuführen. Die Nutzung von Funktionsblöcken erlaubt es, hersteller- und gerätespezifische Funktionen allgemein gültig darzustellen.

Eine Füllstandsregelung (Abb. 4.47) besteht dann zum Beispiel aus einem Transmitter, mit einem AI-(Analog In)Funktionsblock, der den Füllstand auf den Bus überträgt. Das zur Regelschleife gehörende Ventil empfängt die Füllstandswerte vom Bus, verarbeitet die Daten in einem Reglerblock (PID) und gibt sie über einen AO(Analog Out)-Funktionsblock an den internen Stellungsregler weiter. Wie dieses Beispiel zeigt, können nur mit Feldgeräten im Foundation Fieldbus komplett Regelschleifen aufgebaut werden, ohne dass dazu eine Steuerung oder ein Leitsystem den Regelalgorithmus abarbeitet.

Ein weiteres Beispiel zeigt Abb. 4.48 aus der Sicht eines Konfigurationswerkzeuges in einem Prozesseleitsystem. Die drei verwendeten Funktionsblöcke sind physikalisch auf zwei verschiedene Feldgeräte verteilt. Es handelt sich um eine einfache Zweipunktregelung, bei der die Sensoren an eine Eingangsbaugruppe für Namur-Sensoren und die Ventile an eine Ventilanschaltung angeschlossen sind. Wie man sieht, ist es für die Applikation nicht von Bedeutung, welcher Funktionsblock sich in welchem Gerät befindet. Die Applikation wird einfach aufgebaut, indem die entsprechenden Aus- und Eingänge der Blöcke miteinander verbunden werden und die Funktionsblöcke parametriert werden.

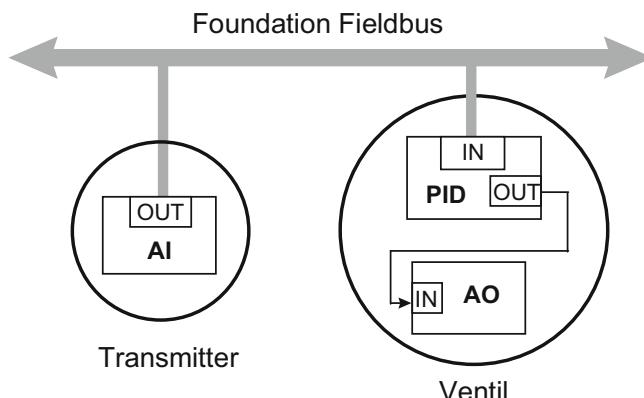


Abb. 4.47 Regelschleife mit Function Blocks

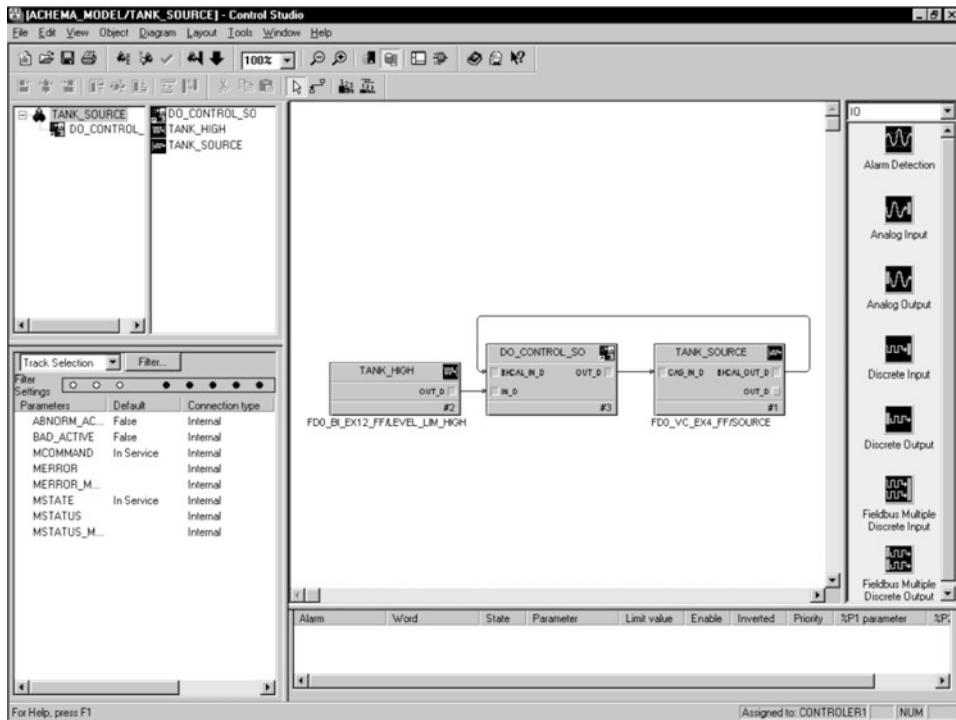


Abb. 4.48 Foundation-Fieldbus-Konfiguration

System Management

Das System Management ist für die exakte Synchronisation aller Abläufe in einem Foundation Fieldbus basierten System verantwortlich (Abb. 4.49). In einem Automatisierungssystem ist es wichtig, dass alle Funktionen in einem bestimmten, deterministischen Zeitraster ablaufen. Diese Aussage bezieht sich natürlich nicht nur auf die Datenübertragung sondern auch z. B. auf die Abarbeitung von Regelungsfunktionen. Das Festlegen der zeitlichen Abläufe wird beim FF Scheduling genannt.

Das System Management teilt die Zeitscheiben des Bussteuerwerks unter den einzelnen Geräten und deren Funktionsblöcken auf. Die Zeit, die nicht für die Übertragung der Ein- und Ausgangsdaten benötigt wird, kann von jedem Teilnehmer für die Übermittlung von azyklischen Nachrichten (z. B. Parameter) genutzt werden, ohne dass die zyklische Nutzdatenübertragung dadurch beeinflusst wird.

Zu den weiteren Funktionen des System Managements gehört die Synchronisation der Uhrzeit in allen Teilnehmern und die automatische Vergabe von Teilnehmeradressen.

Feldgerät steuert Feldgerät

Parallel zu den typischen Sensor- und Aktor-Funktionsblöcken verfügt die Mehrzahl der FOUNDATION Feldgeräte ebenfalls über Steuerungsfunktionen. Ein Stellungsregler

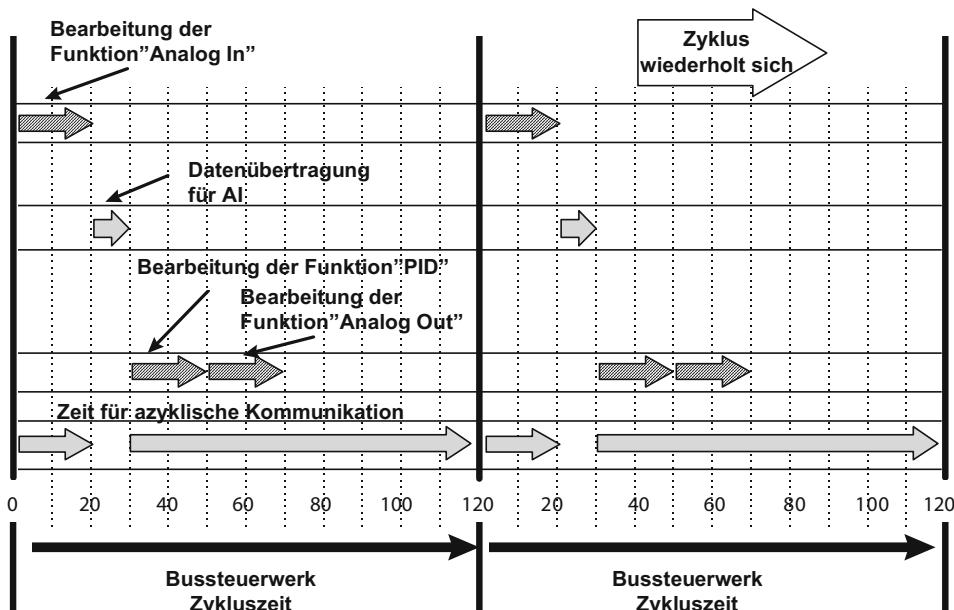


Abb. 4.49 Foundation Fieldbus Buszugriffs- und Zeitsteuerung

kann in Zusammenarbeit mit einem Flowmeter eigenständig die Durchflussrate steuern, der entsprechende PID Block arbeitet nicht in dem weit entfernten Leitsystem sondern in einem der beteiligten Feldgeräte. Dadurch gibt es keine Möglichkeiten für Fehler in übergeordneten Systemen, selbst bei Ausfall der Bedien- und Beobachtungseinheiten wäre der korrekte Fluss des Mediums gewährleistet. Control-in-the-field verbessert die Mess- und Regelleistung durch schnellere Abtastraten sowie geringere Latenzzeiten und ermöglicht damit eine exaktere Kontrolle und erhöhte Verfügbarkeit.

Einbindung in die Leittechnik

Die Einbindung von FF-Teilnehmern in ein Leittechniksystem soll einfach und automatisch möglich sein (Abb. 4.50). Etwa muss der Austausch eines Gerätes der Marke ‚A‘ durch ein Gerät des Herstellers ‚B‘ mit minimalem Konfigurationsaufwand stattfinden können ohne dabei auf Funktionen des Gerätes oder die Integration in das Leitsystem zu verzichten. Hierzu haben sich zwei konkurrierende Systeme etabliert, die den Zugriff auf die Daten des Teilnehmers und die Einbindung in das System ermöglichen.

Bei der Enhanced Device Description Language (EDDL) (deutsch: Erweiterte Gerätebeschreibungssprache) handelt es sich um ein von Herstellern und Bussystemen unabhängig definierte Skriptsprache. Alle Aspekte eines Feldgerätes von der Kommunikation (z. B.: Schnittstelle, Buskommunikation) über Variablen (Durchfluss, Temperatur, Druck), Funktionsblöcke (PID-Regler) bis hin zur graphischen Funktionen (Historienverläufe) definiert werden können. EDDL-fähige Leitsysteme und Plant Asset Managementsysteme

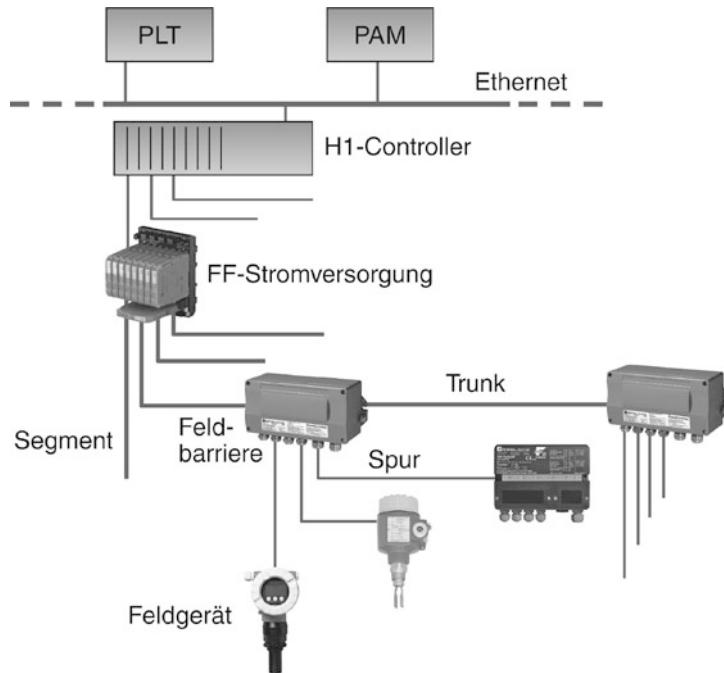


Abb. 4.50 Komponenten eines FOUNDATION Fieldbus H1 Segmentes mit Integration in übergeordnete Systeme

interpretieren die mit dem Gerät gelieferte EDDL, die Darstellung ist vollständig vom Leitsystem realisiert.

Field Device Tool, kurz FDT ist eine Schnittstellendefinition. Das Leitsystem stellt eine Rahmenapplikation, die mit einem Betriebssystem vergleichbar ist zur Verfügung. Für jedes Gerät existiert ein Softwaremodul, der Device Type Manager (DTM). Ähnliche einem Gerätetreiber für einen Computer bildet der DTM die Schnittstelle zum Feldgerät und regelt den Datenfluss von und zum Feldgerät. Darüber hinaus kann ein DTM zusätzliche Funktionen zur Verfügung stellen, etwa Datenanalysen basierend auf den Informationen aus dem Feldgerät, die anderweitig auch nicht abbildungswürdig wären. Die Rahmenapplikation kann die Daten aus allen DTM-Frames zusammenführen für übergeordnete Prozessleistungsfunktionen.

4.2.8 ControlNet

Für die Daten-Kommunikation zwischen Steuerungen und E/A-Peripherie haben sich unterschiedliche Feldbusse in der Industrie seit langem bewährt. Warum aber wurde mit ControlNet noch einmal ein neues Bussystem entwickelt und in den Markt eingeführt?

Weil bei zeitkritischen Anforderungen, zunehmendem Informationsbedarf und höherer Anlagenkomplexität die klassischen Bussysteme mit Master-/Slave-Eigenschaften an ihre Grenzen kommen – für solche Anforderungen ist ein modernes Bussystem wie ControlNet wesentlich besser geeignet. So zeichnet sich ControlNet durch besondere Leistungsmerkmale aus: Producer/Consumer-Technologie, Konstante Übertragungsgeschwindigkeit von 5 Mbit/s über alle Kabellängen, Determinismus für zeitkritische Steuerungsanwendungen, Medienredundanz für erhöhte Verfügbarkeit, Einsatz bis in den Ex-Bereich. ControlNet eignet sich für alle typischen Anwendungen in der Automatisierung: Zum Daten-Erfassen, zum Gerät-Konfigurieren sowie zum direkten Steuern der Peripherie. Die vielfältigen Kommunikationsoptionen (zeitgesteuert, multicast, peer-to-peer) erweitern den Anwendungsbereich und tragen zur Senkung der Busbelastung bei.

Ein ControlNet-Netzwerk ist einfach und kostengünstig aufzubauen und bietet flexible Installationsalternativen. ControlNet erlaubt nahezu jede Topologie (Bus, Baum, Stern und Kombinationen daraus). Geräte-Abzweige können beliebig angeordnet werden, ohne Rücksicht auf Mindestabstände. ControlNet nutzt ein RG-6-Koaxialkabel mit BNC-Steckern (weit verbreitet in der Kabel-TV-Industrie). Damit steht für ControlNet-Installationen eine preiswerte Technologie zur Verfügung, die von verschiedenen Herstellern für bestimmte Anwendungsbereiche angeboten werden (hochflexibel, bewehrt, hitze- und korrosionsbeständig usw.). Zusätzlich ist dieses Kabel dank der hervorragenden Störunempfindlichkeitswerte (EMV) sehr gut für die Fertigungsumgebung geeignet. Falls erforderlich, kann man mit LWL-Verbindungen die Netzwerkausdehnung und Störsicherheit noch weiter steigern.

ControlNet trägt dazu bei, die Netzwerkarchitektur im Werk flach zu halten. Die Kombination von zyklischem E/A-Update mit unregelmäßigem Nachrichtenverkehr (Peer-to-Peer-Nachrichten, Konfigurations- und Programm-Up-/Download-Anwendungen) gelingt hier mit einem einzigen Netzwerk. Dabei ist sichergestellt, dass weder Durchsatz noch Determinismus oder zyklische Wiederholbarkeit beeinflusst werden: Zeitkritische Daten haben garantierte Übertragungszeitpunkte, der restliche Datenverkehr lässt sich hinsichtlich seiner maximalen Übertragungszeiten eindeutig vorherbestimmen. Einfache Anwendungen (z. B. 32 Knoten mit je 8 Bit E/A-Daten) laufen innerhalb von 2 ms Netzykluszeit ab.

Das **Producer/Consumer-Verfahren** ermöglicht mehreren Controllern die E/A-Steuerung auf der gleichen Leitung. Auch die Multicast-Übertragung von Eingängen und Peer-to-Peer-Daten trägt effektiv zur Reduktion des Datenverkehrs auf dem Netzwerk bei, denn so kann z. B. ein Signal gleichzeitig von mehreren Steuerungen gelesen werden. Auch die individuelle Einstellbarkeit der Update-Intervalle und anderer Kommunikationsoptionen für bestimmte Datenverbindungen trägt zur weiteren Effizienzsteigerung bei und erlaubt ein genaues Tuning auf die Applikationsanforderungen. So kann z. B. eine kontinuierliche, zyklische Multicast-Verbindung den regelmäßigen E/A-Datenaustausch sicherstellen, während eine andere azyklische Punkt-zu-Punkt-Verbindung nur ereignisbezogene Daten übermittelt. Anwendungen zur Zeitsynchronisation lokaler Uhren profitieren von der hohen Genauigkeit des Übertragungssystems.

Schließlich kann man auf ControlNet von jedem Knoten aus über das so genannte „Network-Access-Port“ zugreifen, ein Vorteil bei Programmierung und Fehlersuche. Viele Netzinstandhaltungseingriffe können on-line erfolgen, ohne dass die laufende Kommunikation gestört wird; dazu gehören das Hinzufügen oder Entfernen von Geräten oder das Umkonfigurieren bestimmter Geräteparameter.

4.2.8.1 Zielanwendungen

ControlNet wurde mit Blick auf die hohen Echtzeit- und Performance-Anforderungen im Automatisierungs- und Steuerungsbereich entworfen. Das Netzwerk passt optimal für Anwendungen, bei denen Determinismus und wiederholbares Antwortverhalten, hoher Datendurchsatz, hohes Datenvolumen (analog und digital), Datenerfassung über ausgedehnte Entfernung und hoch-synchronisierte Steuerungs- oder Verriegelungsaufgaben benötigt werden. Da ControlNet aber ebenso effektiv die E/A-Updates und Nachrichtenübertragung für Peer-to-Peer-Anwendungen, Remote-Programmierung und Diagnose beherrscht, kann ControlNet ebenfalls als einziges Netzwerk für einfache Anwendungen installiert werden.

Mit ControlNet lassen sich auch viele komplexe Steuerungssysteme integrieren, die auf mehreren Controllern und/oder DCS-Geräten aufgebaut sind. Dazu gehören beispielsweise koordinierte Antriebssteuerungen, Schweißsteuerungen, Motion-Controller, Vision-Systeme, komplexe Batch- und Prozesssteuerungssysteme mit hohem Datenbedarf sowie Anlagen mit mehreren Steuerungen und Bedien- und Anzeigegeräten.

4.2.8.2 Das ControlNet-Protokoll

Die ControlNet-Spezifikation entstand aus der technologischen Zusammenarbeit mehrerer Anbieter hinsichtlich Design, Review und Test von Produkten. Die Dokumentation basiert auf dem klassischen OSI-Referenzmodell (ISO/IEC 7498-1), bestehend aus Physical Layer, Data Link Layer, Network&Transport Layer, Application Layer (mit Kommunikationsobjekten und Diensten) sowie den zugeordneten Managementfunktionen.

Eigenschaften des Physical Layers (physikalische Ebene)

ControlNet nutzt drei Varianten von Übertragungsmedien:

- RG6-Koaxialkabel wird in Verbindung mit BNC-Steckern als passive Bustechnologie genutzt (Abzweige mit Stichleitungen dienen zum Geräteanschluss an die Fernleitung)
- LWL kann bei Punkt-zu-Punkt-Verbindungen optional eingesetzt werden
- NAP (Network Access Port) ist eine lokale RS422-Verbindung zum temporären Direktanschluss ans ControlNet für Konfigurations-, Diagnose- und Programmierzwecke.

Sowohl Koaxial- als auch LWL-basierende Systeme können mit bestimmten Komponenten bis in den EX-Bereich verlegt werden. Unterschiedliche Netzwerktopologien wie Bus, Baum, Stern sowie Kombinationen daraus werden unterstützt. Bis zu 99 Knoten erlaubt ControlNet, maximal 1000 m Busausdehnung sind möglich bei 2 Teilnehmern (250 m mit

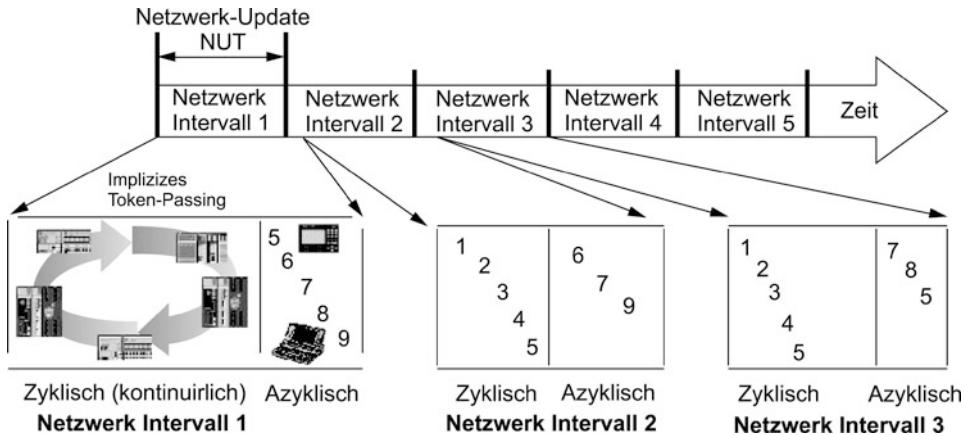


Abb. 4.51 ControlNet-Update-Intervalle auf der Zeitachse (Beispiel)

48 Teilnehmern), maximal 25 km mit Repeatern. Dabei unterstützt ControlNet auch eine kostengünstige Medienredundanz: Alle angeschlossenen Geräte überprüfen kontinuierlich die Signalqualität beider Kanäle und entscheiden automatisch, welcher Kanal benutzt wird.

Die Daten werden mit 5 Mbit/s manchester-kodiert übertragen, wodurch exakte Zeitsynchronisation und hohe Übertragungssicherheit garantiert wird; zusätzliche Fehlererkennung garantiert ein 16-Bit-CRC für jedes Datenpaket.

Eigenschaften des Data Link Layers

Für den Medienzugriff benutzt ControlNet ein Verfahren namens CTDMA (Concurrent Time Domain Multiple Access). Dieses wurde speziell entworfen im Hinblick auf die Performance von E/A-Daten oder Analogwerten sowie von zeitkritischen Verriegelungssignalen, die nicht unter der zusätzlichen Übertragung anderer Nachrichten für Programmier- oder Konfigurationsanwendungen leiden darf.

Dabei wird ein Zeitscheibenverfahren zugrunde gelegt (Abb. 4.51): Zyklische Übertragungsintervalle können zwischen 2 und 100 ms in der so genannten NUT (Network Update Time) festgelegt werden. In jeder NUT wird zwischen zeitkritischen (zyklischen) und zeitunkritischen (azyklischen) Daten unterschieden.

Der Medienzugang wird den individuellen Knoten durch ein implizites Tokenverfahren innerhalb jedes Intervalls garantiert. Es gibt bei ControlNet keinen zentralen Bus-Scheduler, die Busverwaltung erfolgt dezentral. Alle Teilnehmer werden fortlaufend synchronisiert und wissen daher zu jedem Zeitpunkt, wann sie an der Reihe sind und wer gerade sendet. Der Tokenumlauf (round-robin) setzt sich fort jeweils bis zur höchsten konfigurierten Adresse (SMAX, UMAX) und beginnt im nächsten NUT-Intervall wieder von vorn. Fällt ein Teilnehmer aus, so wartet der Knoten mit der nächsthöheren Adresse (MAC ID) einen „Time-Slot“ ab und beginnt dann seinerseits mit dem Senden. Wird der

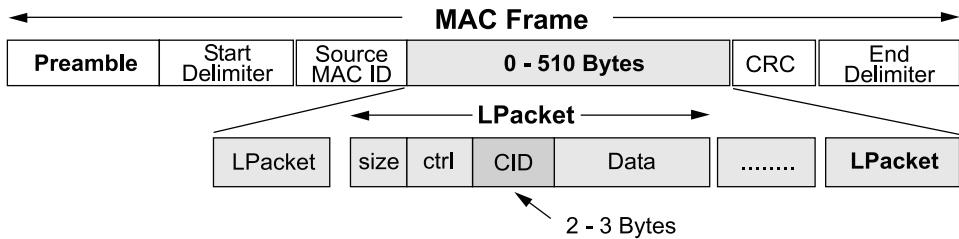


Abb. 4.52 Der Aufbau eines ControlNet-Datenrahmens

ausgefallene Teilnehmer später wieder aktiv, reiht er sich automatisch wieder in den Sendenumlauf ein, ohne dass der Netzwerkbetrieb angehalten wird. Hat eine Station nichts zu senden, schickt sie ein „Null Frame“.

In der ControlNet-Konfiguration wird die Übertragungsbandbreite für zeitkritische Daten im Voraus reserviert. Dieser zyklische Dienst (Scheduled Service) ist streng deterministisch und reproduzierbar. Die dafür reservierte Zeit richtet sich nach den zu erwartenden Applikationsanforderungen (unabhängig davon können die Sende-Zeitpunkte von Datenpaketen individuell eingestellt werden).

Die übrige Zeit bis zum Ende der NUT wird für den Transport zeitunkritischer und azyklischer Daten benutzt (Unscheduled Service). Anders als im zyklischen Dienst ist dieses Zeitfenster nicht bestimmten Knoten fest zugewiesen, sondern wird je nach Bedarf von beliebigen Teilnehmern genutzt. Das bedeutet, dass die Datenübertragung im azyklischen Dienst keinerlei Auswirkung auf die zyklischen Daten hat und hinsichtlich der maximalen Übertragungszeit vorhersagbar bleibt.

Zum Ende einer NUT werden im „Guardband-Slot“ Synchronisationsdaten übermittelt.

Aus Effizienzgründen werden die Applikationsdaten in einem MAC-Frame (MAC = Media Access Control) mit variabler Paketgröße zusammengefasst (Abb. 4.52).

Für den Datenaustausch dieser Datenpakete nutzt ControlNet das Producer/Consumer-Kommunikationsmodell. Anstelle individueller Quell- und Zieladressen enthalten die Datenpakete eine CID-Kennzeichnung (CID = Connection Identifier). Ein Producer (Erzeuger) gibt per Broadcast ein Datenpaket auf die Leitung, und alle am Empfang interessierten Consumer (Verbraucher) können dieses Paket gleichzeitig aufnehmen, indem sie über den CID die entsprechenden Pakete herausfiltern und die enthaltenen Daten nutzen (Abb. 4.53). Dieses Modell unterstützt praktisch alle bekannten Kommunikationsbeziehungen, von Master/Slave über Multimaster bis zu Peer-to-Peer.

Eigenschaften des Network & Transport-Layers

Bevor Daten übertragen werden, muss bei ControlNet eine virtuelle Verbindung zwischen zwei Applikationen aufgebaut werden. Eine solche virtuelle Verbindung ist definiert über die beiden Endpunkte für den Datentransfer. Spezielle Mechanismen stellen den Verbindungsauflauf über besondere (verbindungslose) Transfers sicher.

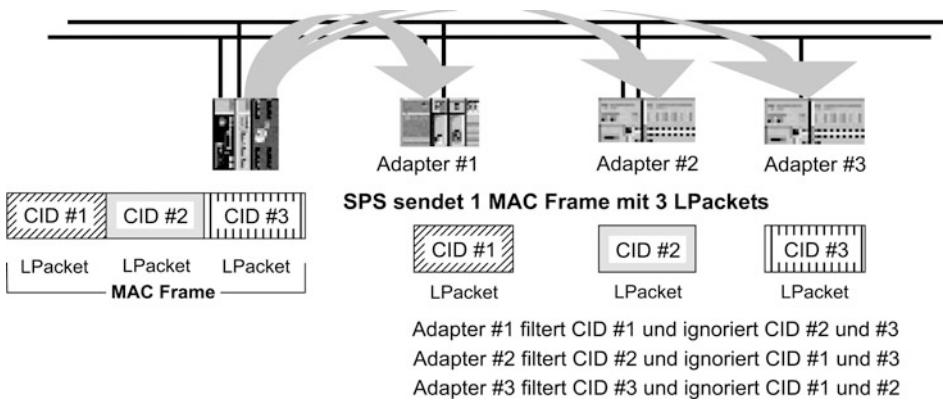


Abb. 4.53 ControlNet fasst mehrere kleine Datenpäckchen in einem größeren Rahmen zusammen

Es gibt unterschiedliche Transportdienste für diese Verbindungen. Sie können applikationsabhängig und vielseitig kombiniert werden. Verbindungen können entweder „Multicast“ oder „Peer-to-Peer“ sein. Für die Initialisierung der Datenübertragung kann zwischen „zyklisch zeitgesteuert“, „ereignisgesteuert“ oder „applikationsgesteuert“ gewählt werden. Unterschiedliche Transportklassen gewährleisten Duplikat-Erkennung, Bestätigung, Verifizierung und Fragmentierung (für lange Nachrichten).

Eigenschaften des Application-Layers

Die oberen Ebenen des ControlNet-Protokollstacks basieren auf dem neuesten Stand objekt-orientierter Designregeln. Sowohl die Kommunikations- als auch die Applikationselemente werden als „Objekte“ verstanden. Spezielle ControlNet-Messages können Dienste anfordern, die auf korrespondierende Objektinstanzen (oder deren Attribute) angewendet werden. Dieses Schema verbessert den expliziten Zugriff auf alle Konfigurations-, Status- und Runtime-Variablen in einem Netzknoten. Gleichzeitig erlauben implizite E/A-Verbindungen einen direkten und besonders effizienten Austausch von E/A-Daten, ohne zusätzliche Zwischenverarbeitung.

Management-Eigenschaften

ControlNet erlaubt eine dynamische Netzwerk-Rekonfiguration (Rescheduling). Jeder Netzknoten hält eine Kopie der Verbindungsparameter sowie der lokalen Timing-Vorgaben. Spezielle Netzknoten, „Keeper“ (Wächter) genannt, enthalten die Verbindungs- und Timing-Parameter für das gesamte Netzwerk: Der primäre Keeper sichert die Konsistenz der gesamten Konfiguration während des Startup und der On-line-Rekonfiguration des Netzwerks, während sekundäre Keeper automatisch als Backup für den primären Keeper arbeiten. Das bedeutet, dass individuelle zyklische Verbindungen (Scheduled Connections) geändert oder aufgelöst werden können, ohne andere existierende Übertragungen

zu beeinflussen; sogar Netzknoten können hinzugefügt oder entfernt werden ohne Einfluss auf andere Knoten.

Das ControlNet-Protokoll definiert z. B. auch spezielle Broadcast-Messages für die Zeiteinstellung, die in Verbindung mit den Clock-Synchronisationsmechanismen im Data Link Layer die Realtime-Clocks in den Knoten mit hoher Genauigkeit (besser als 10 Mikrosekunden im gesamten Netzwerk) synchronisieren.

Für die Netzwerkdagnostik ist ein einheitliches Verhalten der Kommunikationsparameter innerhalb von ControlNet spezifiziert.

4.3 Ethernetbasierte Feldbusse

4.3.1 Industrial Ethernet – Was ist das eigentlich?

„Industrial Ethernet“ wurde Ende der 90er Jahre zu einem der meistdiskutierten Themen in der Automatisierungstechnik. Kaum eine einschlägige Messe kam ohne eine Podiumsdiskussion zu diesem Thema aus, und in allen Fachzeitschriften waren wiederholt Beiträge dazu zu lesen. Den klassischen Feldbussen wurde der rasche Tod prophezeit und es gründeten sich verschiedene Organisationen mit dem Ziel, den Einsatz von Ethernet in der Automatisierungstechnik zu fördern oder gar „Industrial Ethernet“ zu definieren. Die Anwender verfolgten dieses Spiel mit großem Interesse und der Hoffnung, dass sie das aus der Büroumgebung vertraute Ethernet zukünftig auch in ihren Fabriken und Anlagen einsetzen können und die babylonische Sprachverwirrung der Feldbusprotokolle ein Ende hat.

Dieser anfänglichen Euphorie folgte dann aber bald die Erkenntnis, dass „Industrial Ethernet“ keineswegs ein wohldefinierter Standard war, sondern eher ein Gattungsbegriff, unter dem all das subsumiert wurde, was mit dem Einsatz von Ethernet in der Automatisierungstechnik zu tun hatte.

Praktisch alle großen Steuerungs- und Leitsystemhersteller setzten ja schon seit Jahren Ethernet-basierte Kommunikationssysteme im oberen Bereich der Automatisierungspyramide, also zwischen SPS und übergeordneten Systemen, ein. Allerdings verwendeten all diese Systeme proprietäre Applikationsprotokolle, waren also nicht „offen“.

Weil es aber im Bürobereich bewährte, auf Ethernet basierende Kommunikationssysteme gibt, die – über das Internet – auch einen weltweiten standardisierten Datenaustausch ermöglichen, liegt die Idee nahe, dies auch auf die Automatisierungstechnik auszudehnen. Zudem sind Ethernet-Komponenten aufgrund der großen Stückzahlen äußerst preisgünstig. Eine Ethernet-Karte für einen PC kostet beispielsweise nur einen Bruchteil einer Feldbus-Anschaltung für einen PC.

Kann aber Ethernet die Kommunikationsanforderungen der Automatisierungstechnik erfüllen? Um diese Frage zu beantworten, ist es notwendig, einen genaueren Blick auf Ethernet zu werfen und dessen Fähigkeiten mit den Eigenschaften moderner Feldbussysteme zu vergleichen.

4.3.2 Grundlagen des Ethernet

Die Historie von Ethernet

Ethernet ist ein Standard für die Schichten 1 und 2 des ISO/OSI-Referenzmodells (siehe Abschn. 1.2.1) und bildet damit nur die „Unterkante“ eines Kommunikationssystems. Aber es definiert Kabel, Stecker und Topologie und damit den für den Anwender sichtbaren Teil eines Bussystems.

Der Begriff „Industrial Ethernet“ kann damit gar nicht für ein vollständiges, alle Schichten abdeckendes Kommunikationssystem stehen. Aber er fungiert als Sammelbegriff für diejenigen Kommunikationssysteme in der industriellen Automatisierungstechnik, die auf den untersten beiden Schichten Ethernet einsetzen.

Ethernet ist weitaus älter als die heute eingesetzten Feldbusprotokolle. 1973 entwickelte Robert Metcalfe bei Xerox die ersten Ideen zu Ethernet. 1979 veröffentlichten die drei Firmen Digital Equipment, Intel und Xerox den nach ihren Anfangsbuchstaben benannten DIX-Standard. 1982 übernahm dann IEEE die Ethernet-Spezifikation in die Normengruppe IEEE 802.

Zeitlich parallel zur Entwicklung des Ethernets entstand in vom US Verteidigungsministerium getriebenen Forschungsprojekten das Konzept eines Rechnernetzes, welches auch bei einer teilweisen militärischen Zerstörung noch funktionsfähig bleibt. Dieses Konzept sollte später als Internet bekannt werden, und die Familie von Kommunikationsprotokollen, die für dieses Netz entwickelt wurde, erhielt nach ihren beiden wichtigsten Vertretern die Bezeichnung „TCP/IP Protocol Suite“.

Anfang der 80er Jahre waren damit wesentliche Technologien erstmals verfügbar, die in ihrer Kombination später zu einer beispiellosen Entwicklung in der Informationstechnik führen sollten:

- Ethernet als leistungsfähiges lokales Netz
- Das Internet als heterogenes Weitverkehrsnetz
- Die TCP/IP-Protokollfamilie als Basis leistungsfähiger Dienste im Internet
- Workstations auf UNIX-Basis und später PCs als Netzknoten

Diese Technologien haben viele Bereiche des modernen Lebens revolutioniert, und es liegt damit nahe, sie auch in der Automatisierungstechnik einzusetzen.

Die wesentlichen technischen Merkmale von Ethernet

Wie bereits erwähnt, decken die Ethernet-Standards im ISO/OSI-Referenzmodell die Schichten 1 und 2 ab. Sie definieren also die Busphysik (Übertragungsmedien, Stecker, elektrische Eigenschaften), das Buszugriffsverfahren und ein Protokoll zur Übertragung einzelner Datenpakete innerhalb eines „Ethernets“.

Ethernet ist in seiner ursprünglichen Form ein logischer Bus, auch wenn es physikalisch nicht immer als Bus realisiert ist. Es gibt ein gemeinsames Übertragungsmedium („Ether“) in Form einer Busleitung, das sich alle daran angeschlossenen Stationen teilen.

Alle Stationen sind gleichberechtigt. Jede Station prüft vor dem Senden, ob der Bus frei ist. Ist dies der Fall, dann beginnt sie sofort mit der Übertragung. Ansonsten wartet sie das Ende der laufenden Übertragung ab und beginnt dann nach einer kurzen Ruhephase von 9,6 µs zu senden.

Beginnen zwei Stationen gleichzeitig zu senden, kommt es zu einer Kollision. Dieser Fall kann trotz der vorherigen Prüfung des Mediums aufgrund der endlichen Laufzeiten der elektrischen Signale auftreten. Beide Stationen erkennen eine solche Kollision, brechen die Übertragung ab und senden ein „Jamming“-Signal. Nach einer zufälligen Zeit wiederholen die Stationen den Sendeversuch.

Dieses Buszugriffsverfahren wird mit CSMA/CD (Carrier Sense Multiple Access with Collision Detection) bezeichnet. Aufgrund der möglichen Kollisionen, deren Wahrscheinlichkeit mit zunehmender Auslastung des Ethernets immer größer wird, kann keine sichere Annahme für die maximale Übertragungszeit eines Datenpakets getroffen werden. Ethernet gilt deshalb (zumindest in seiner originären Form) als nicht deterministisches Protokoll, welches unter Echtzeitbedingungen nicht eingesetzt werden kann.

Ethernet-Pakete sind minimal 64 und maximal 1518 Byte groß. Der von den darüberliegenden Schichten nutzbare Anteil liegt zwischen 48 und 1500 Byte. Die Pakete enthalten eine Quell- und eine Zieladresse mit jeweils 48 bit. Diese auch als MAC-Adressen bezeichneten Ethernet-Adressen sind weltweit eindeutig und werden bereits durch die Hersteller der Hardware festgelegt, können also nicht mehr geändert werden. Bei der Zieladresse sind auch Gruppenadressen für Multicast und Broadcast möglich. Anhand einer ebenfalls im Datenpaket enthaltenen vier Byte langen Prüfsumme kann der Empfänger erkennen, ob das Paket korrekt übertragen wurde. Der Sender erhält jedoch keine Empfangsbestätigung vom Empfänger, wie das bei Feldbussen häufig der Fall ist. Dies ist bei Ethernet eine Aufgabe der darüberliegenden Transportschicht.

Übertragungsbandbreite und Busphysik wurden bei Ethernet immer wieder der technischen Entwicklung angepasst. Nachdem lange Zeit nur eine Übertragungsgeschwindigkeit von 10 Mbit/s möglich war, wurden Mitte der 90er Jahre mit *Fast Ethernet* 100 Mbit/s erreicht. Heute stehen 1 GBit/s zur Verfügung und Mitte 2002 wurde der Standard für 10 GBit/s verabschiedet.

Zunächst war Ethernet für koaxiale Kupferkabel spezifiziert worden. Das berühmte *Yellow Cable* (10Base5) der Anfangszeit war sowohl bezüglich der Kabelführung als auch der Anschlusstechnik schwer zu handhaben. Im Bürobereich verbreitete sich deshalb vor allem das *Thin Ethernet* oder *Cheapernet* (10Base2), ein dünneres Koaxialkabel, an das die Teilnehmer über T-Stücke mit BNC-Steckern angeschlossen sind. Diese Linienstruktur hat aber generell den Nachteil, dass bei jeder Änderung der Topologie (z. B. beim Einfügen neuer Teilnehmer) der Bus aufgetrennt werden muss und deshalb für eine bestimmte Zeit das gesamte Netz nicht funktioniert. Deshalb ging man in den weiteren Entwicklungsschritten von der Linienstruktur zu einer Sternstruktur über, welche aus Punkt-zu-Punkt-Verbindungen über Twisted-Pair-Leitungen (Kabel mit mehreren verdrillten Aderpaaren) und Sternkopplern bestehen.

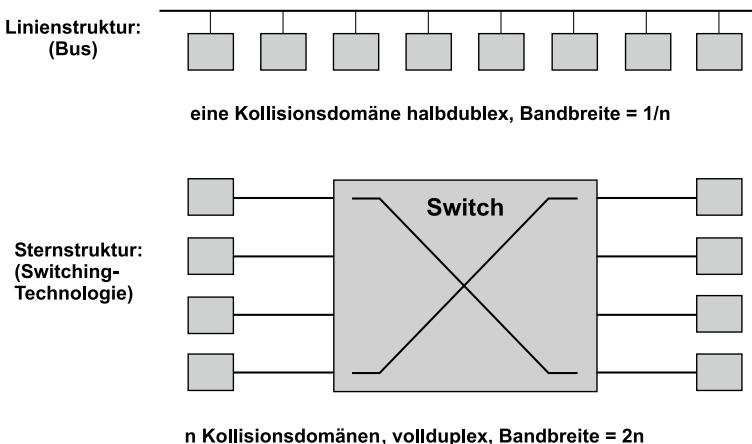


Abb. 4.54 Linienstruktur und Sternstruktur bei Ethernet

Bei Twisted-Pair-Leitungen werden die Sende- und Empfangssignale getrennt und auf verschiedenen Leitungspaaren übertragen (Vollduplexbetrieb). Sollen nur zwei Stationen miteinander kommunizieren, dann kann dies mit einem Kabel mit gekreuzten Leitungspaaren (Crossover-Kabel) erfolgen. Müssen dagegen mehr als zwei Teilnehmer verbunden werden, dann kommen Sternkoppler (Hubs) zum Einsatz. *Hubs* sind Repeater mit mehreren (typischerweise vier oder acht) Anschlüssen (Ports), welche die an einem Port empfangenen Daten an allen anderen Ports wieder ausgeben.

Um die Bandbreite eines Ethernets besser nutzen zu können, geht man jedoch zunehmend von der generellen Weiterleitung aller Datenpakete aller Teilnehmer zu einer selektiven Weiterleitung über. Dazu verwendet man intelligente Sternkoppler mit Vermittlungsfunktion. Solche *Switches* analysieren die Quelladressen der eingehenden Pakete und lernen im Laufe der Zeit, an welchen Ports welche Stationen angeschlossen sind. Trifft ein Paket für eine bekannte Station ein, dann wird es nur an dem Port gesendet, über den diese Station erreicht werden kann. Switches können Ethernet-Pakete zwischenspeichern (*Store-and-Forward*), wenn die Übertragungsstrecke am Sendeport gerade belegt ist. Nachdem mittlerweile acht verschiedene Prioritätsebenen für Ethernet-Pakete definiert wurden, ist auch eine priorisierte Weitervermittlung von Paketen möglich. Hochpriore Nachrichten können also solche von niedrigerer Priorität in den Switches überholen.

Solche mit Switches aufgebauten Netze unterscheiden sich im Übertragungsverhalten deutlich vom ursprünglichen Ethernet. Dies sei an folgendem (etwas hypothetischen) Szenario in Abb. 4.54 verdeutlicht.

Acht Teilnehmer sollen an einem Ethernet betrieben werden, wobei sie jeweils paarweise und bidirektional miteinander kommunizieren wollen. Beim klassischen Ethernet hängen alle Teilnehmer an einer Busleitung und teilen sich die Bandbreite dieses Mediums. Da zu einem Zeitpunkt nur ein Teilnehmer senden kann, ist nur Halbduplex-

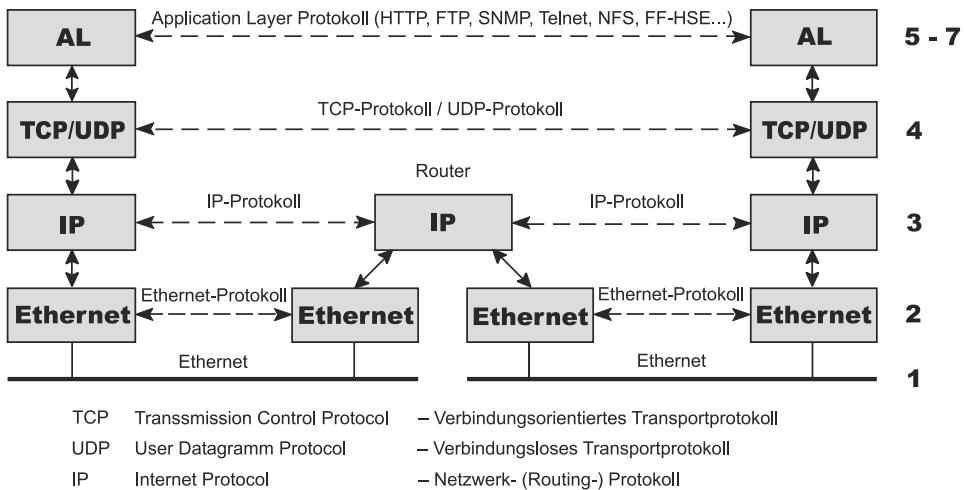


Abb. 4.55 Ethernet im ISO/OSI-Referenzmodell

betrieb möglich und es kann zu Kollisionen kommen. Man sagt deshalb, alle Teilnehmer bilden eine *Kollisionsdomäne*.

Wenn diese acht Teilnehmer dagegen über einen Switch mit acht Ports so verbunden sind, dass an jedem Port genau ein Teilnehmer über Twisted-Pair-Leitungen vollduplex verbunden ist, dann steht gegenüber der Buslösung bei gleicher Übertragungsgeschwindigkeit eine 16-fach höhere Bandbreite zur Verfügung (8 Vollduplexverbindungen gegenüber einer Halbduplexverbindung). Dieses Szenario ist zwar aufgrund der angenommenen paarweisen Kommunikation sehr konstruiert, zeigt aber die Vorteile der Switching-Technologie, nämlich hohe Verfügbarkeit, hohe Bandbreiten und eine drastische Reduzierung der Kollisionen.

Aufgrund dieser Vorteile wandelt sich Ethernet zunehmend von der kollisionsbehafteten Linienstruktur zu einer weitgehend kollisionsfreien Stern- oder Baumstruktur mit einem deterministischeren Verhalten und damit einer besseren Echtzeiteignung.

Ethernet im ISO/OSI-Referenzmodell

Ethernet deckt die Schichten 1 und 2 des ISO/OSI-Referenzmodells ab, sorgt also nur für die Übertragung von Einzelpaketen innerhalb eines physikalischen Netzes. Für eine sinnvolle Anwendung muss es deshalb durch höhere Protokollsichten ergänzt werden, wobei meist die TCP/IP Protocol Suite eingesetzt wird (Abb. 4.55).

Die TCP/IP-Protokollfamilie verwendet oberhalb von Ethernet auf Schicht 3 das Internet-Protokoll (IP), welches als Netzprotokoll für das „Routing“, also für die Suche des optimalen Verbindungswegs zwischen den Kommunikationspartnern zuständig ist. In einem kleinen lokalen Netzwerk können Sender und Empfänger einer Nachricht am selben Ethernet hängen, beim Zugriff aufs Internet wird der Übertragungsweg dagegen

möglicherweise über Kontinente hinweg und damit über eine Vielzahl von Übertragungsstrecken und Vermittlungsknoten führen. In allen Fällen sorgt das Internet-Protokoll anhand der in jedem Datenpaket enthaltenen IP-Adressen dafür, dass es seinen Weg vom Sender zum Empfänger findet.

Unterhalb des Internet-Protokolls muss aber keineswegs immer Ethernet verwendet werden. Wie jede Kommunikationsschicht verdeckt auch das Internet-Protokoll die Funktionalität der darunterliegenden Schicht. Statt des Ethernets können auch Übertragungsmedien wie Modem-, ISDN- oder Funkverbindungen genutzt werden.

Aus Sicht von Ethernet und Internet-Protokoll besteht zwischen einzelnen Datenpaketen kein Zusammenhang. Der Anwender erwartet jedoch eine sichere Kommunikationsverbindung in Form eines logischen Kanals, in dem eine Sequenz von Daten korrekt von Endpunkt zu Endpunkt übertragen wird. Die einzelnen Pakete, in die dieser Datenstrom zerteilt wird, dürfen weder verloren gehen noch vertauscht oder dupliziert werden. Diese Funktionalität erbringt die Transport-Schicht (Schicht 4). In Kombination mit dem Internet-Protokoll wird dabei in der Regel das Transport Control Protocol (TCP) verwendet. Es sorgt für eine virtuelle Punkt-zu-Punkt-Verbindung zwischen den beiden Kommunikationspartnern.

In manchen Fällen reicht es jedoch, nur einzelne Datenpakete an einen oder mehrere Empfänger zu übertragen. Dann verwendet man das unquittierte User Datagram Protocol (UDP).

Die Schichten 1 bis 4 bilden die Basis für eine ganze Reihe von Applikationsprotokollen, aber auch für viele der in der Automatisierungstechnik eingesetzten Ethernet-Lösungen.

Bekannte Applikationsprotokolle oberhalb TCP/IP sind:

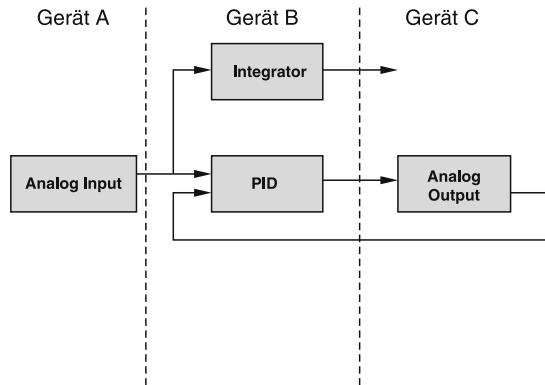
- Das Hypertext Transfer Protocol (HTTP) für den Zugriff auf Webseiten im Internet
- Das TELNET-Protokoll für Remote Login
- Das File Transfer Protocol (FTP) für Upload und Download von Dateien
- Das Simple Network Management Protocol (SNMP) für Network Management
- Das Simple Mail Transfer Protocol (SMTP) für die Übertragung elektronischer Post

Alle diese Applikationsprotokolle sind aber primär auf die Kommunikationsbedürfnisse in der Büroumgebung ausgerichtet. Es fehlen die typischen Automatisierungsfunktionen, wie sie von den klassischen Feldbussen angeboten werden.

4.3.3 Ethernet im Vergleich zu modernen Feldbussystemen

Betrachtet man die reinen Kommunikationsfunktionen der unteren Protokollschichten, dann bringen Ethernet und die TCP/IP-Protokolfamilie mittlerweile recht gute Voraussetzungen mit, um in der industriellen Kommunikation eingesetzt zu werden. Aber moderne Feldbusse bieten weit mehr als bloße Kommunikationsfunktionalität. Dies soll an zwei

Abb. 4.56 Beispiel eines verteilten Prozesses



Beispielen aus der Prozessautomatisierung und aus der Fertigungsautomatisierung aufgezeigt werden.

Typische Anforderungen aus der Prozessautomatisierung

Abb. 4.56 zeigt einen Regelkreis als kleinen Ausschnitt eines verteilten Prozesses. Ein Durchflussmesser erfasst den Durchfluss eines Mediums als analoge Größe, welche von einem PID-Regler verarbeitet wird, um über ein Stellventil den Durchfluss konstant zu halten. Gleichzeitig soll ein Integrator aus dem Durchfluss die Durchflussmenge berechnen. Diese vier Funktionsblöcke seien auf drei physikalische Geräte (Durchflussmesser, Steuerung, Stellventil) verteilt, welche über einen Feldbus miteinander kommunizieren.

Um diesen Prozess einrichten, parametrieren und betreiben zu können, sind eine Reihe von Festlegungen und Funktionalitäten erforderlich, wie sie von modernen, für die Prozessautomatisierung geeigneten Feldbussystemen wie FOUNDATION Fieldbus oder PROFIBUS-PA angeboten werden:

- Zunächst müssen die Funktionsblöcke mit all ihren Parametern und internen Zuständen standardisiert werden, um eine Austauschbarkeit zwischen den Geräten verschiedener Hersteller zu gewährleisten. Solche Standardfunktionsblöcke decken beispielsweise die Funktionen Analogeingang, Analogausgang, PID-Regler oder diskrete Ein-/Ausgänge ab. Um die Funktionalität eines Aktors oder Sensors auf diese generischen Funktionsblöcke abbilden zu können, werden zusätzlich *Transducer Blocks* spezifiziert, die die gerätespezifischen Parameter enthalten. Während also ein Temperaturtransmitter und ein Durchflussmesser beide über einen identischen Funktionsblock „Analogeingang“ modelliert werden, unterscheiden sie sich in ihren Transducer Blocks, die die zum Messverfahren gehörigen Parameter enthalten.
- Um Geräte verschiedener Hersteller mit ein- und demselben Inbetriebnahmewerkzeug parametrieren zu können, muss das Geräteverhalten in einer standardisierten Form beschrieben werden. Dies erfolgt über Gerätebeschreibungssprachen (Device Description

Language, DDL) oder über einheitliche Bedien- und Parametrieschnittstellen (Field Device Tool, FDT-Konzept).

- Für die Inbetriebnahme sind Management-Funktionen zur Geräteidentifikation und für die Zuweisung von Adressen und symbolischen Namen erforderlich, die vom Kommunikationsprotokoll bereitzustellen sind.
- Die Ein- und Ausgänge der Funktionsblöcke müssen entsprechend der Gesamtfunktion miteinander verschaltet werden. Sofern solche Verbindungen über Gerätegrenzen hinweg verlaufen, sind die dafür benötigten Kommunikationsverbindungen zu projektierten. Dies erfordert entsprechende Management-Funktionen im Kommunikationsprotokoll. Da häufig ein Ausgangssignal mit mehreren Eingängen verbunden werden soll, muss das verwendete Kommunikationssystem in der Lage sein, auch 1-zu-n Verbindungen (Multicast) zu unterstützen.
- Im operativen Betrieb müssen die Funktionsblöcke zyklisch in einer festgelegten Reihenfolge abgearbeitet und die Ausgangswerte über das Kommunikationssystem übertragen werden. Dazu ist zunächst ein Verfahren zur Synchronisierung der Uhrzeit bei allen Teilnehmern erforderlich. Auf Basis dieser systemweit gültigen Uhrzeit können dann die Funktionsblöcke nach globalen „Fahrplänen“ (Schedules) in der richtigen Reihenfolge abgearbeitet werden. Analog dazu werden auch die Ausgangssignale der Funktionsblöcke vom Kommunikationssystem zu festgelegten Zeitpunkten publiziert.
- Als letzte typische Anforderung aus der Prozessautomatisierung sei hier die Eigensicherheit genannt. Unter *Eigensicherheit* oder *Intrinsic Safety* versteht man eine Form des Explosionsschutzes, bei der die Energie innerhalb der im explosionsgefährdeten Bereich betriebenen Stromkreise so begrenzt wird, dass kein zündfähiger Funke entstehen kann. Diese Forderung kann nur mit speziellen Ausprägungen der physikalischen Schicht erreicht werden, wie z. B. dem Physical Layer nach IEC 61158-2, der von PROFIBUS-PA und FOUNDATION Fieldbus H1 verwendet wird. Dort erfolgt die Übertragung der Information und der Versorgungsenergie für die Feldgeräte über das-selbe Adernpaar (Busspeisung).

Typische Anforderungen aus der Fertigungsautomatisierung

Auch in der Fertigungsautomatisierung haben sich die Anforderungen an Feldbussysteme stetig verschärft und erweitert, was am Beispiel PROFIBUS sehr gut nachzuvollziehen ist.

PROFIBUS wurde zunächst in der Variante FMS (Fieldbus Message Specification) als Zellenbus spezifiziert, d. h. zur Vernetzung von Steuerungen und anderen komplexen Automatisierungsgeräten. FMS beinhaltet deshalb ein recht mächtiges Objektmodell mit entsprechenden Diensten. Diese Komplexität führte in Verbindung mit den zunächst noch niedrigen Übertragungsraten von maximal 500 kbit/s zu Übertragungszeiten, die den Anschluss schneller zyklischer Peripherie an PROFIBUS nicht zuließen.

Deshalb wurde mit PROFIBUS DP eine spezielle Protokollvariante zum Anschluss dezentraler Peripherie mit Übertragungsraten bis zu 12 Mbit/s geschaffen. Dieses Protokoll ist darauf optimiert, das Prozessabbild in einer Steuerung zyklisch von und zu den E/A-Komponenten zu übertragen. Der Erfolg von PROFIBUS DP ließ den Wunsch entstehen,

auch komplexe E/A-Geräte mit einer großen Anzahl von azyklisch les- und schreibbaren Parametern anzuschließen. Die dafür notwendigen Protokollerweiterungen wurden mit DP-V1 bezeichnet und ließen auch den Einsatz in der Prozessautomatisierung zu (Profil PROFIBUS PA).

Schließlich entstand der Wunsch, Feldbusse auch in der Antriebstechnik einzusetzen und die mechanische Kopplung von Achsen durch eine elektronische Kopplung per Feldbus zu ersetzen. Dies erspart nicht nur mechanische Komponenten, sondern ermöglicht auch beim Einricht- und Anfahrbetrieb eine individuelle Ansteuerung der einzelnen Achsen. Die Kopplung von schnellen synchronisierten Antrieben über einen Feldbus erfordert sehr kurze und vor allem äquidistante Buszyklen. PROFIBUS DP-V2 bietet einen solchen isochronen Modus mit einer Abweichung der Zykluslänge von weniger als einer Mikrosekunde. In Verbindung mit einer bei PROFIBUS DP-V2 als Querverkehr bezeichneten Publisher-Subscriber-Kommunikation zwischen Slave-Geräten kann damit die Positionsinformation einer Master-Achse mit einer zeitlichen Varianz von unter einer Mikrosekunde an mehrere Slave-Achsen übergeben werden.

Schließlich seien noch die anwendungsspezifischen Festlegungen bei PROFIBUS erwähnt. Solche *Profile* legen branchenspezifische oder gerätespezifische Nutzungsarten von PROFIBUS fest, um eine Interoperabilität von Geräten verschiedener Hersteller zu gewährleisten. Aber auch spezielle ergänzende Protokolle wie beispielsweise das PROFIsafe-Protokoll zum Einsatz in sicherheitsgerichteten Anwendungen fallen unter diese Kategorie. Erst die durch diese Profile erreichte Standardisierung des Applikationsverhaltens von Feldbusteilnehmern führt zu wirklich offenen Systemen.

Was bietet Ethernet für die Automatisierungstechnik?

Wie an den beiden genannten Beispielen zu sehen ist, decken moderne Feldbussysteme einen weiten Bereich an Funktionalität ab. Feldbusstandards definieren nicht nur das Kommunikationsverhalten, sondern spezifizieren das gesamte Geräteverhalten, bieten Funktionen des System Management und Network Management zu Konfiguration, Inbetriebnahme und Diagnose an.

Was die reine Kommunikationsfunktionalität betrifft, können Ethernet und TCP/IP einiges für die Automatisierungstechnik bieten:

- Ethernet stellt mit derzeit 1 GBit/s und demnächst 10 GBit/s ausreichend Bandbreite für die Automatisierungstechnik bereit. Die Telegramme sind zwar bei kleinen Nutzdatenlängen im Vergleich zu Feldbussen sehr ineffizient, aber die Übertragungsrate liegt dafür um Größenordnungen darüber.
- Das früher als Hauptargument gegen Ethernet angeführte nicht deterministische Verhalten gilt für sternförmig aufgebaute Netze mit Switches nicht mehr, da dort keine Kollisionen auftreten. Echtzeitfähigkeit bedeutet ja, dass das Kommunikationssystem in der Lage ist, alle Kommunikationsaufgaben in „Echtzeit“ zu erledigen, also schneller zu sein als die Dynamik des technischen Prozesses dies erfordert. In vielen An-

wendungsbereichen sind Zyklus- bzw. Reaktionszeiten im Millisekundenbereich völlig ausreichend.

- IP, TCP und UDP sind stabile und bewährte Netzwerk- und Transportprotokolle und bieten mehr als die entsprechenden Schichten bei Feldbussen. Die Routing-Funktion des Internet-Protokolls ermöglicht Kommunikationsverbindungen über viele heterogene Teilnetze hinweg, wie wir dies vom Internet her kennen. Feldbusse bestehen dagegen meist aus nur einem logischen Bus ohne eine Möglichkeit des Netzübergangs und bilden damit Kommunikationsinseln, die höchstens über Steuerungen als Gateways mit der Außenwelt kommunizieren können. Auch TCP ist als Transportprotokoll bei der Übertragung großer Datenmengen und bei der End-zu-End-Kontrolle den Feldbusprotokollen überlegen, welche auf die Übertragung kleiner Nutzdatenmengen in einem Segment optimiert sind und die Fehlerbehandlung häufig dem Anwender überlassen.
- Die TCP/IP-Protokollfamilie enthält eine Reihe von Management- und Hilfsprotokollen, die auch für die Automatisierungstechnik von Vorteil sind. Hier sei zum Beispiel das Address Resolution Protocol (ARP) genannt, welches für eine IP-Zieladresse die Ethernet-Adresse findet, über die der Empfänger erreicht werden kann. Damit kann die bei Feldbussen in aller Regel statische Adresskonfiguration überwunden werden, was für größere Netze unabdingbar ist.
- Oberhalb von TCP und UDP gibt es verschiedenste Anwendungsprotokolle wie FTP, HTTP und Telnet, die zwar nicht für die Automatisierungstechnik gedacht waren, dort aber gut eingesetzt werden können. Beispiele hierfür sind:
 - das Simple Network Management Protocol (SNMP) als Zugriffsprotokoll auf Managementdaten
 - das Simple Network Time Protocol (SNTP) zur Uhrzeitsynchronisierung
 - das File Transfer Protocol (FTP) zum Up- und Download von größeren Datenmengen, wie Steuerungsprogrammen, Protokollierungsdaten, Software-Updates und Anwendungsdaten.
 - das Hypertext Transfer Protocol (HTTP) zum Zugriff auf Embedded Web Server in den Automatisierungsgeräten, über die Parametrierung der Geräte oder das Abrufen von Zustandsinformationen erfolgt (Web based Management).
 - Das Simple Mail Transfer Protocol (SMTP), mit dem beispielsweise Gerätealarme als E-Mails gesendet werden können, um Servicepersonal im Bereitschaftsdienst zu erreichen.
 - Protokolle für Remote Procedure Calls (RPC) wie Microsoft's DCOM.
- Ein weiterer Vorteil der TCP/IP-Protokollfamilie liegt darin, dass diese Protokolle die Basis des Internets bilden und damit die Nutzung des „Netzes der Netze“ für Automatisierungsaufgaben möglich wird. Natürlich gibt es hier Sicherheitsaspekte zu beachten, und die Übertragungszeiten im Internet prädestinieren es nicht gerade für Automatisierungsaufgaben. Aber zur zentralen Wartung und Diagnose von geographisch weit verteilten Anlagen wird das Internet bereits heute vorteilhaft genutzt.
- Als letzter Pluspunkt für Ethernet und TCP/IP sei hier paradoxe Weise angeführt, dass nicht immer Ethernet als Schicht-2-Protokoll verwendet werden muss. Wie jede

Kommunikationsschicht verdeckt auch das Internet-Protokoll die darunter liegenden Schichten und macht sie damit austauschbar, ohne dass der Anwender etwas davon merkt. IP funktioniert beispielsweise auch über analoge oder digitale Telefonleitungen, über Funk und optische Übertragungsstrecken.

Was fehlt Ethernet im Vergleich zu modernen Feldbusssystemen?

Nach den vielen positiven Aspekten mag es verwundern, warum Ethernet nicht schon lange die klassischen Feldbusse ersetzt hat. Aber Ethernet und TCP/IP weisen trotz der gewaltigen Funktionalität einige schwerwiegende Defizite auf, die der Verwendung in der Automatisierungstechnik entgegenstehen:

- An wichtigster Stelle sei das Fehlen eines Objektmodells für die Automatisierungstechnik genannt. Prozessvariablen, Merker, Zähler und Alarne und aus diesen elementaren Objekten zusammengesetzte Objekte wie beispielsweise Prozessabbilder, Funktionsblöcke und Trends sind bei keinem der gebräuchlichen Applikationsprotokolle oberhalb von TCP definiert. Die für die Feldbusse spezifizierten Anwendungsprofile bauen aber auf solchen Objektdefinitionen auf.
- Auch die Kommunikationseigenschaften müssen speziell auf die Automatisierungstechnik abgestimmt werden. Neben Client-Server-Verbindungen werden dort auch Publisher-Subscriber-Verbindungen benötigt, bei denen eine Datenquelle (z. B. der Ausgang eines Funktionsblocks oder eine Eingangsklemme) zyklisch ein Datum publiziert, welches von einer oder mehreren Datensenken abonniert wird. Auch für die Alarmübertragung und -quittierung sind spezielle Kommunikationsverfahren notwendig. Es fehlt also ein auf die Anforderungen der Automatisierungstechnik abgestimmtes Applikationsprotokoll, welches auf TCP und UDP aufsetzt.
- Ein streng isochrones Protokoll, dessen Jitter weit unter einer Mikrosekunde liegt (wie PROFIBUS DP-V2) kann ohne Protokollmodifikationen mit Ethernet nicht realisiert werden, ist jedoch Voraussetzung für den Einsatz in der Antriebstechnik.
- Für die Prozessautomatisierung wird eine eigensichere Busphysik gefordert, für den Einsatz in der Installationstechnik die Verwendung der installierten Stromleitungen als Übertragungsmedium (Powerline-Betrieb). Beides ist für Ethernet noch nicht verfügbar.
- Darüber hinaus fehlt bei Ethernet und TCP/IP vieles, was klassische Feldbusse zusätzlich zur reinen Kommunikation leisten, wie z. B. standardisierte Applikationen und Funktionsblöcke, branchenspezifische Profilfestlegungen, Gerätebeschreibungsmethoden und vieles mehr.

Zusammenfassung

Zusammenfassend kann also festgestellt werden, dass Ethernet „nur“ ein Protokoll für die Schichten 1 und 2 ist und dass die TCP/IP-Familie nur ein Satz von mehr oder weniger gut für die Automatisierungstechnik geeigneten Kommunikationsprotokollen ist. Ein „richtiger Feldbus“ leistet aber deutlich mehr.

„Industrial Ethernet“ macht also noch keinen Feldbus, hat aber das Potential, die Basis für neue und leistungsfähige Architekturen zu bilden. Ethernet ist mittlerweile sehr gut geeignet, die reinen Kommunikationsanforderungen der Automatisierungstechnik abzudecken. Die Nutzung der TCP/IP-Protokollfamilie bietet zudem die Möglichkeit, auch in der Automatisierungstechnik weltweite Informationsverbunde aufzubauen. Es fehlt jedoch ein einheitliches und standardisiertes Objektmodell und Applikationsprotokoll für die Automatisierungstechnik. In explosionsgefährdeten Bereichen, bei schnellen Antrieben und in der Installationstechnik werden sich die klassischen Feldbusse noch einige Zeit behaupten können.

Alle großen Hersteller und Feldbusorganisationen beschäftigen sich mit Ethernet und TCP/IP als neuen Protokollvarianten. Da aber sowohl die wesentlichen Investitionen als auch der Kundennutzen bei Feldbussen im Bereich der Applikationsprotokolle und der Profile liegen, wollen sie ihre über Jahre gewachsene Gesamtarchitektur erhalten und lediglich die unteren Schichten um Ethernet ergänzen. Dies führt zur Inkompatibilität der seit Mitte der 90er Jahre entstandenen „Industrial Ethernet“ Lösungen.

Und um Investitionen in die Feldbusse und die installierte Basis zu schützen, wird außerdem nach Integrationsmöglichkeiten klassischer Feldbusse in ethernet-basierte Netze gesucht.

In den folgenden Abschnitten werden die wichtigsten Lösungen für „Industrial Ethernet“ vorgestellt.

4.3.4 PROFINET

In der Automatisierungstechnik von heute bestimmen neben den Feldbussen zunehmend Ethernet und die Informationstechnologie (IT) mit den etablierten Standards, wie z. B. TCP/IP und XML, das Geschehen. Durch die Integration der Informationstechnik in die Automatisierung eröffnen sich deutlich verbesserte Kommunikationsmöglichkeiten zwischen Automatisierungssystemen, weit reichende Konfigurations- und Diagnosemöglichkeiten und netzweite Servicefunktionen.

Bereits im Jahr 2000 haben die PROFIBUS Nutzerorganisation (PNO) und ihr internationaler Dachverband PROFIBUS & PROFINET International (PI) mit PROFINET konkrete Strategien für die durchgängige Nutzung von Ethernet in der industriellen Kommunikation vorgestellt. Die enge Zusammenarbeit von Herstellern und Anwendern – insbesondere der AIDA (AutomatisierungsInitiative Deutscher Automobilhersteller) – stellt die Praxistauglichkeit und Anwenderfreundlichkeit von PROFINET sicher.

Heute ist PROFINET die in der IEC 61158 und IEC 61784 standardisierte offene Lösung für Industrial Ethernet, die alle Anforderungen der Automatisierungstechnik abdeckt. Mit PROFINET können Lösungen für die Fertigungs- sowie Prozess-Automatisierung, für Safety-Anwendungen und das gesamte Spektrum der Antriebstechnik bis hin zu isochronen Motion Control-Anwendungen realisiert werden. Im Jahr 2011 wurde die Zahl von über 4 Millionen weltweit installierter PROFINET-Knoten überschritten.

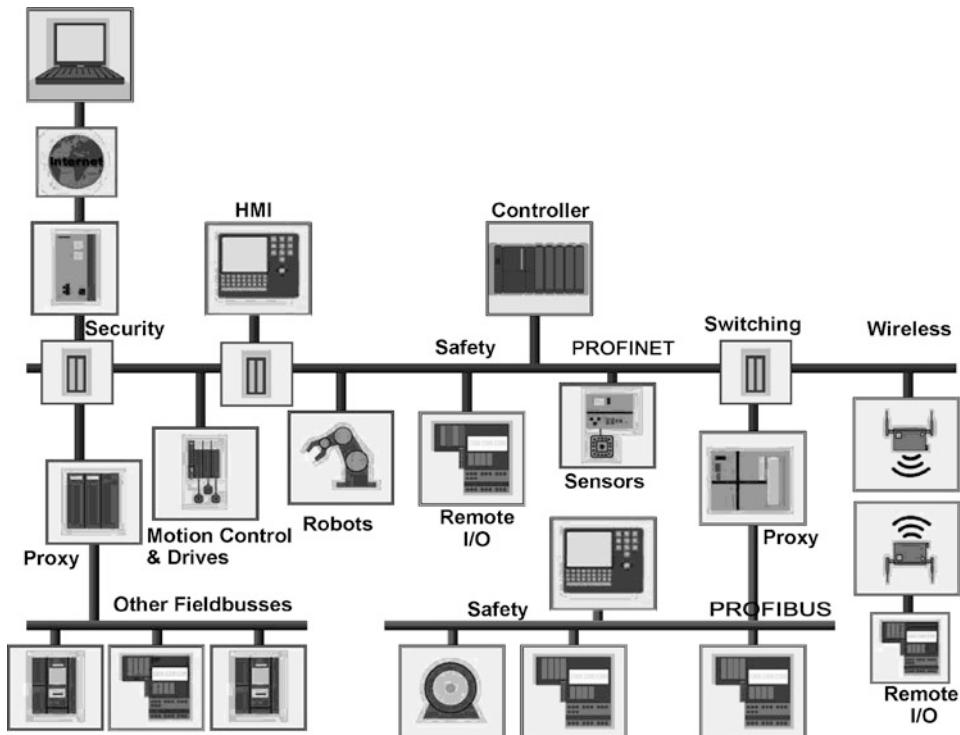


Abb. 4.57 Exemplarischer Aufbau eines PROFINET Netzwerkes mit unterlagerten Feldbussen

PROFINET ist Ethernet-kompatibel gemäß IEEE-Standard. Nur bei den Anforderungen, die Standard-Ethernet nach IEEE 802 nicht oder nicht zufriedenstellend löst, wurde der Ethernet-Standard entsprechend erweitert. Neben der Real-Time-Fähigkeit und der Nutzung der IT-Technologie spielt bei PROFINET der Aspekt Investitionsschutz eine wichtige Rolle. PROFINET ermöglicht die Einbindung existierender Feldbussysteme wie PROFIBUS, AS-Interface, INTERBUS und DeviceNet ohne Änderungen der existierenden Feldgeräte (Abb. 4.57).

Das PROFINET-Konzept ist modular aufgebaut, sodass der Anwender die benötigte Funktionalität selbst wählen kann. Die Funktionalität unterscheidet sich im Wesentlichen durch die Art des Datenaustauschs, um den teils sehr hohen Anforderungen an Geschwindigkeit gerecht zu werden.

PROFINET IO beschreibt eine I/O-Datensicht auf die dezentrale Peripherie. Sie umfasst die Real-Time- (RT) und die isochrone Kommunikation IRT (Isochronous Real-Time) mit der dezentralen Peripherie. Die Bezeichnungen RT und IRT beschreiben lediglich die Echtzeit-Eigenschaften bei der Kommunikation innerhalb von PROFINET IO.

PROFINET CBA eignet sich für die komponentenbasierte Maschinen-Maschinen-Kommunikation über TCP/IP und die Real-Time-Kommunikation für Echtzeitanfor-

derungen im modularen Anlagenbau. Es ermöglicht eine einfache Modularisierung von Anlagen und Produktionslinien durch verteilte Intelligenz mittels grafischer Konfiguration der Kommunikation intelligenter Module.

PROFINET CBA und PROFINET IO können sowohl separat betrieben als auch kombiniert werden, sodass eine PROFINET IO-Teilanlage in der Anlagensicht als ein PROFINET CBA-Modul erscheint. Die Realisierung einer Teileinheit kann beispielsweise in Form einer deterministisch ausgelegten Kommunikation über PROFINET IO mit IRT-Funktionalität realisiert sein. Das Zusammenfügen der Teilanlagen, die einzeln vorgefertigt und getestet werden können, zu einer gesamten PROFINET-Einheit erfolgt dann mit PROFINET CBA.

Dieser Artikel beschreibt im Wesentlichen die Funktionalität von PROFINET IO.

4.3.4.1 PROFINET im Überblick

Die vier Schüsselfunktionen von PROFINET sind:

- Performance: Automatisierung in Real-Time
- Safety: Sicherheitsgerichtete Kommunikation mit PROFIsafe
- Diagnose: Hohe Anlagenverfügbarkeit durch schnelle Inbetriebsetzung und effiziente Fehlersuche
- Investitionsschutz: Nahtlose Integration von Feldbussystemen

Dazu legt PROFINET folgende Mindestanforderungen an die Datenkommunikation fest:

- 100 Mbit/s-Datenkommunikation mit Cu oder FiberOptic-Übertragung (100BaseTX und 100BaseFX)
- Vollduplex-Übertragung
- Switched Ethernet
- Auto Negotiation (Aushandeln der Übertragungsparameter)
- Auto Crossover (Sende- und Empfangsleitung werden im Switch gekreuzt)
- Wireless-Kommunikation auf Basis von WLAN und Bluetooth.

Als überlagertes Protokoll verwendet PROFINET für den bedarfsorientierten Datenaustausch UDP/IP. UDP steht für User Datagram Protocol und beinhaltet die ungesicherte, verbindungslose Broadcast-Kommunikation in Verbindung mit IP. Parallel zur UDP/IP-Kommunikation basiert PROFINET für den zyklischen Datenaustausch auf einem skalierbaren Real-Time-Konzept.

PROFINET Highlights

Ein kaskadierbares Redundanzkonzept garantiert die stoßfreie Umschaltung der Kommunikationswege im Fehlerfall, was die Anlagenverfügbarkeit deutlich erhöht.

Um Feldgeräte einfach tauschen zu können, besitzt PROFINET die integrierte Funktionalität der Nachbarschaftserkennung. Mit diesen Informationen ist es auch möglich, die Anlagen-Topologie sehr übersichtlich grafisch darzustellen.

Prozessdaten und Alarne werden bei PROFINET IO in Real-Time (RT) übertragen, basierend auf den Definitionen von IEEE und IEC. Real-Time-Daten werden gegenüber TCP(UDP)/IP-Daten mit höherer Priorität behandelt.

Für Motion Control-Anwendungen oder hochgenauen Regelungsaufgaben ist bei PROFINET das Isochronous-Real-Time (IRT)-Konzept definiert. Die Datenaustausch-Zyklen liegen hier normalerweise im Bereich von ein paar Hundert Mikrosekunden- bis hin zu einer Millisekunde. Der Unterschied zur Real-Time-Kommunikation liegt im Wesentlichen in der Isochronität, so dass der Beginn eines Buszyklus mit höchster Präzision eingehalten wird. Der Beginn eines Buszyklus darf maximal um $1 \mu\text{s}$ abweichen (Jitter).

PROFINET IO-Geräteklassen

PROFINET folgt beim Datenaustausch dem Provider-Consumer-Modell (Abb. 4.58). Der Provider (gewöhnlich das prozessnahe Feldgerät) stellt die Prozessdaten einem Consumer (normalerweise eine SPS mit Verarbeitungsprogramm) zur Verfügung. Prinzipiell können in einem PROFINET IO-Feldgerät beliebige Funktionsanordnungen (Provider/Consumer) enthalten sein. Zur besseren Strukturierung von PROFINET IO-Feldgeräten sind folgende Geräteklassen definiert:

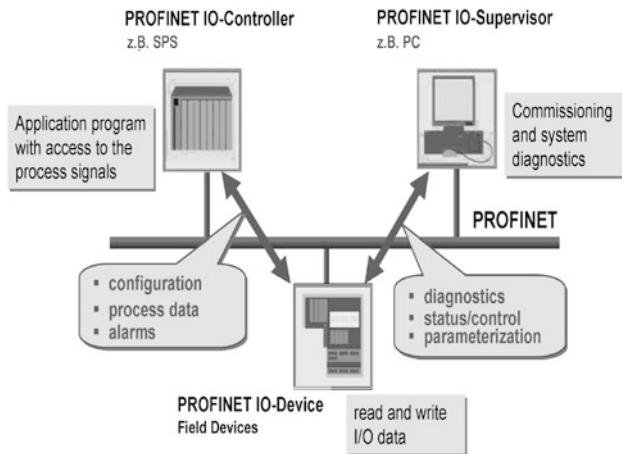
- IO-Controller: Dies ist typischerweise die Speicherprogrammierbare Steuerung (SPS), in der das Automatisierungsprogramm abläuft (entspricht der Funktionalität eines Klasse-1-Masters bei PROFIBUS). In einer Teilanlage gibt es mindestens einen IO-Controller und ein oder mehrere IO-Devices.
- IO-Supervisor: Dies kann ein Programmiergerät (PG), Personal Computer (PC) oder Human Machine Interface-Gerät (HMI) zu Inbetriebsetzungs- oder Diagnosezwecken sein (entspricht einem Klasse-2-Master bei PROFIBUS). Ein IO-Supervisor wird für Inbetriebnahmezwecke und Fehlersuche meist nur temporär eingebunden.
- IO-Device: Ein IO-Device ist ein dezentral angeordnetes I/O-Feldgerät, das über PROFINET IO angekoppelt wird (entspricht der Funktion eines Slaves bei PROFIBUS). Ein IO-Device kann mit mehreren IO-Controllern Daten austauschen.

4.3.4.2 Grundlagen von PROFINET IO

Der Anschluss der PROFINET IO-Feldgeräte erfolgt über Switches als Netzwerk-Komponenten – entweder sternförmig über separate Mehrport-Switches oder linienförmig mit im Feldgerät integrierten Switches.

Ein PROFINET IO-Feldgerät wird innerhalb eines Netzwerks durch seine Geräte-MAC-Adresse (Medium Access Control) adressiert. Bei manchen Telegrammen (z. B. bei Synchronisation oder Nachbarschaftserkennung) wird nicht mit der Geräte-MAC-Adresse, sondern mit der MAC-Adresse für den jeweiligen Port gearbeitet. Deshalb benötigt jeder Switchport in einem Feldgerät eine separate Port-MAC-Adresse. Bei einem 2-

Abb. 4.58 Datenaustausch
Provider–Consumer



Port-Feldgerät sind demnach 3 MAC-Adressen im Auslieferzustand enthalten. Diese Port-MAC-Adressen sind allerdings für den Anwender nicht sichtbar. PROFINET sieht durch den Anschluss der Feldgeräte über Switches immer nur Punkt-zu-Punkt-Verbindungen (wie bei Ethernet) (Abb. 4.59).

PROFINET-taugliche Switches müssen „Auto Negotiation“ (das automatische Aushandeln der Übertragungsparameter) und „Auto Crossover“ (das Kreuzen der Sende- und

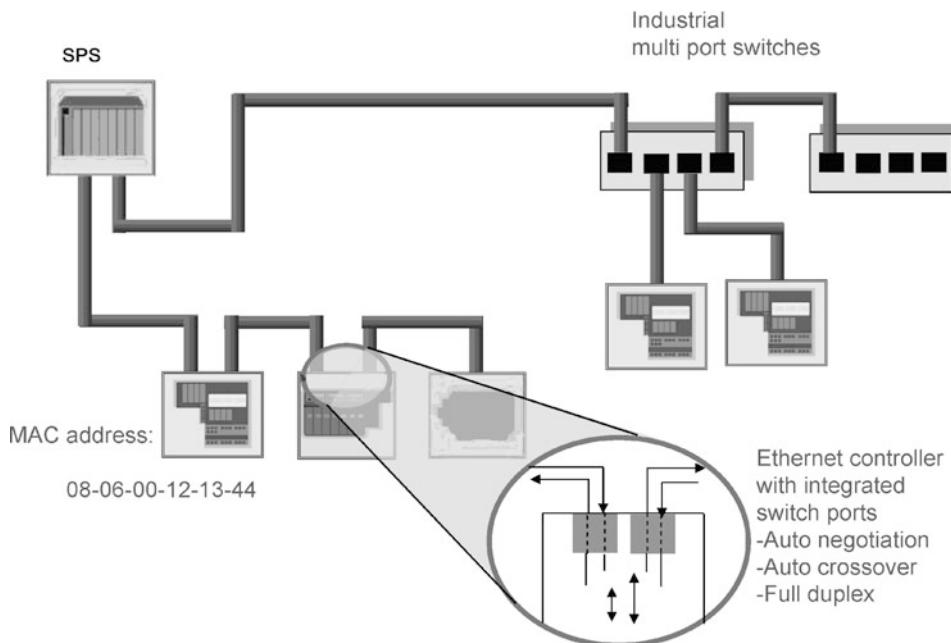


Abb. 4.59 Der Anschluss von PROFINET IO-Feldgeräten erfolgt über Switches

Empfangsleitung im Switch) unterstützen. Dadurch kann die Kommunikation selbstständig aufgebaut werden und die Konfektionierung der Übertragungskabel ist einheitlich.

Nach der Anlagenprojektierung lädt das Engineering-Tool alle zum Datenaustausch erforderlichen Informationen in den IO-Controller inklusive der IP-Adressen für die angeschlossenen IO-Devices. Anhand des Namens (und der damit verbundenen MAC-Adresse) kann ein IO-Controller die projektierten Feldgeräte erkennen und ihnen die festgelegten IP-Adressen mit dem in PROFINET IO integrierten DCP-Protokoll (Discovery and Configuration Protocol) zuweisen. Nach der Adressauflösung folgt der Systemhochlauf mit der Übertragung der Parameter an die IO-Devices. Anschließend kann die Anlage in den Produktivdatenverkehr übergehen.

Gerätemodell

Zum leichteren Verständnis der Prozessdaten-Adressierung in einem PROFINET IO-Feldgerät unterscheidet man kompakte Feldgeräte, bei denen der Ausbaugrad im Auslieferzustand bereits festgelegt ist, und modulare Feldgeräte, deren Ausbaugrad für unterschiedliche Anwendungen beim Projektieren der Anlage individuell an den Einsatzfall angepasst werden kann.

Alle Feldgeräte sind in ihren technischen und funktionellen Möglichkeiten in einer vom Feldgeräteentwickler zu erstellenden GSD-Datei (General Station Description, Geräte-Stammdaten) beschrieben. In ihr ist u. a. das Gerätemodell repräsentiert, das durch den DAP (Device Access Point) und den definierten Modulen für eine bestimmte Gerätefamilie bestimmt ist. Ein DAP ist sozusagen die Busanschaltung (der Zugangspunkt für die Kommunikation) mit der Ethernet-Schnittstelle und dem Verarbeitungsprogramm. Es ist in der GSD-Datei mit seinen Eigenschaften und Möglichkeiten definiert. Ihm kann eine Vielzahl von Peripheriebaugruppen zugeordnet werden, um den eigentlichen Prozessdatenverkehr zu bewerkstelligen.

Das bewährte Gerätemodell von PROFIBUS wurde für PROFINET IO weitgehend übernommen und an die Anforderungen der Endanwender angepasst. Daraus resultiert bei PROFINET IO eine weitere Schachtelungstiefe: Ein Slot kann in mehrere Subslots eingeteilt werden.

Der Slot kennzeichnet den physikalischen Steckplatz einer Peripherie-Baugruppe in einem modularen I/O-Feldgerät. Anhand der unterschiedlichen Slots werden die projektierten Module adressiert, die einen oder mehrere Subslots (die eigentlichen I/O-Daten) für den Datenaustausch enthalten. Innerhalb eines Slots bilden die Subslots die eigentliche Schnittstelle zum Prozess (Ein-/Ausgänge). Die Granularität eines Subslots (bit-, byte-, wortweise Aufteilung der I/O-Daten) bestimmt der Hersteller. Der Dateninhalt eines Subslots wird immer durch eine Statusinformation begleitet, aus der die Gültigkeit der Daten abgeleitet werden kann.

Ein Index spezifiziert die Daten innerhalb eines Slots/Subslots, die azyklisch über Read/Write-Services gelesen oder geschrieben werden können. Anhand des Index können beispielsweise Parameter an eine Baugruppe geschrieben oder herstellerspezifische Baugruppendaten ausgelesen werden.

Damit es bei der Definition von Anwenderprofilen (wie z. B. bei PROFIdrive, Wiegen und Dosieren, usw.) nicht zu konkurrierenden Zugriffen kommen kann, wurde mit dem API (Application Process Identifier/Instance) eine zu den Slots und Subslots zusätzliche Adressierungsebene definiert. Durch diesen Freiheitsgrad ist es möglich, unterschiedliche Applikationen auch separat zu behandeln, um die Überschneidung von Datenbereichen (Slots und Subslots) zu verhindern.

Real-Time-Kommunikation

Es gibt in der industriellen Automatisierung Anforderungen hinsichtlich Zeitverhalten und Isochronität, die über den UDP/IP-Kanal von Standard-Ethernet nicht oder nur unzureichend erfüllt werden können. PROFINET definiert für diese Anforderungen ein skalierbares Real-Time-Konzept.

Die Übertragung von RT-Daten (ohne TCP/IP-Informationen) basiert auf dem zyklischen Datenaustausch mit einem Provider/Consumer-Modell. Dafür reichen die Kommunikationsmechanismen der Schicht 2 (nach ISO/OSI-Modell) und Standard-Netzkomponenten wie z. B. Switches und Standard-Ethernet-Controller aus. Zur optimierten Verarbeitung von RT-Frames innerhalb eines IO-Devices wurde zusätzlich zum VLAN-Tag nach IEEE802.1Q (Priorisierung von Datenframes) ein spezieller Ethertype eingeführt, der eine schnelle Kanalisierung dieser PROFINET-Frames in der übergeordneten Software des Feldgerätes ermöglicht. Ethertypes werden von der IEEE als ein eindeutiges Unterscheidungskriterium zu anderen Ethernet-Protokollen vergeben. Der Ethertype 0x8892 ist in der IEEE spezifiziert und wird bei PROFINET IO für den schnellen Datenaustausch benutzt.

Um die Kommunikationsmöglichkeiten und damit auch den Determinismus bei PROFINET IO besser skalieren zu können, wurden Real-Time-Klassen für den Datenaustausch definiert. Aus Anwendersicht handelt es sich hierbei um eine unsynchronisierte und eine synchronisierte Kommunikation. Die Details werden in den Feldgeräten selbständig abgewickelt.

Real-Time beinhaltet bei PROFINET automatisch eine Erhöhung der Priorität gegenüber UDP/IP-Frames, um die Durchleitung der Daten in den Switches zu priorisieren. Folgende Klassen der RT-Kommunikation sind definiert, wobei der Unterschied nicht in der Performance, sondern im Determinismus liegt.

Die unsynchronisierte RT-Kommunikation innerhalb eines Subnetzes (*RT_CLASS_1*) ist bei PROFINET IO die übliche Art der Datenübertragung. Es sind keine speziellen Adressierungsinformationen notwendig; der Zielteilnehmer wird nur anhand der „Dest. Addr“ identifiziert. Wenn sich der RT-Datenverkehr auf ein Subnetz (gleiche Netz-ID) beschränken lässt, ist diese Variante die einfachste. Dieser Kommunikationsweg ist parallel zur UDP/IP-Kommunikation standardisiert und in jedem IO-Feldgerät implementiert.

Auf die Verwaltungsinformationen von UDP/IP wurde hier bewusst verzichtet. Empfangene RT-Frames werden bereits beim Empfang anhand des Ethertypes (0x8892) identifiziert und an den RT-Pfad zur unmittelbaren Verarbeitung weitergeleitet. In dieser RT-Klasse können industrietaugliche Standard-Switches eingesetzt werden.

Die *RT_CLASS_2*-Frames können synchronisiert oder unsynchronisiert übertragen werden. Bei der synchronisierten Variante der Kommunikation wird der Beginn eines Buszyklus für alle Teilnehmer definiert. Damit ist das Zeitraster genau festgelegt, wann Feldgeräte senden dürfen. Dies ist für alle an der Kommunikation beteiligten Feldgeräte in der *RT_CLASS_2* immer der Anfang des Buszyklus. PROFINET taugliche Switches müssen bei *RT_CLASS_2* diese Synchronisation unterstützen. Für diese auf Performance ausgelegte Datenübertragung sind spezielle Hardware-Vorkehrungen zu treffen (Ethernet-Controller/Switch mit Unterstützung der Isochronität).

Bei der synchronisierten *RT_CLASS_3*-Kommunikation erfolgt das Senden der Prozessdaten nach einer genauen, beim Anlagen-Engineering festgelegten Reihenfolge mit höchster Präzision (maximal erlaubte Abweichung vom Beginn eines Buszyklus ist $1 \mu\text{s}$). Diese mit Hilfe der Topologie optimierte Datenübertragung wird auch als IRT-Kommunikation bezeichnet (Isochronous Real-Time). Bei der *RT_CLASS_3*-Kommunikation kommt es zu keinen Wartezeiten. Um den Vorteil der auf höchste Performance getrimmten Datenübertragung nutzen zu können, sind spezielle Hardware-Vorkehrungen zu treffen (Ethernet-Controller mit Unterstützung der Isochronität).

Die Kommunikation *RT_CLASS_UDP* ermöglicht den Datenaustausch zwischen Teilnehmern in unterschiedlichen Subnetzen und benötigt Adressierungsinformationen über das Zielnetzwerk (IP-Adresse). In dieser RT-Klasse können Standard-Switches eingesetzt werden. Für RT-Frames ist es ausreichend, Datenzyklen von 5 ms bei 100 Mb/s im Vollduplex-Betrieb mit VLAN-Tag zu erreichen. Diese RT-Kommunikation kann mit allen verfügbaren Standard-Netzwerk-Komponenten realisiert werden.

Datenverkehr

Die zyklischen I/O-Daten werden in einem parametrierbaren Raster zwischen Provider und Consumer als Real-Time-Daten unquittiert übertragen. Die Verbindungsüberwachung erfolgt anhand einer Zeitüberwachung. Bei der Datenübertragung im Frame werden die Daten eines Subslots von einem nachfolgenden Providerstatus begleitet. Diese Status-Informationen werden vom jeweiligen Consumer der I/O-Daten ausgewertet. Damit kann er die Gültigkeit der Daten allein aus dem zyklischen Datenaustausch beurteilen. Zusätzlich werden die Consumer-Status für die Gegenrichtung übertragen.

Pro Telegramm sind im Anschluss an die „Data Unit“ Begleitinformationen (Trailer) enthalten, die Aussagen über die globale Gültigkeit der Daten treffen, Informationen zur Redundanz tragen und den Diagnosezustand bewerten (Data Status, Transfer Status). Auch die Zyklusinformation (Cycle Counter) des Providers ist angegeben, sodass man dessen Aktualisierungsrate auf einfache Weise ermitteln kann. Das Ausbleiben der zyklischen Daten wird vom jeweiligen Consumer der Kommunikationsbeziehung überwacht. Bleibt der Empfang der projektierten Daten innerhalb der Überwachungszeit aus, schickt der Consumer eine Fehlermeldung an die Applikation.

Mit dem azyklischen Datenaustausch können IO-Devices parametriert, konfiguriert oder Statusinformationen ausgelesen werden. Dies wird mit den Read-/Write-Frames über

die Standard-IT-Dienste mittels UDP/IP bewerkstelligt. Neben den für den Gerätehersteller frei nutzbaren Datensätzen sind folgende Systemdatensätze speziell definiert:

- Diagnoseinformation, die der Anwender von jedem Gerät zu jedem Zeitpunkt auslesen kann.
- Logbuch-Einträge (Alarme und Fehlermeldungen), mit denen detaillierte zeitliche Aussagen zu den Ereignissen innerhalb eines IO-Gerätes erzielt werden können.
- Identifikationsinformationen wie in der Richtlinie „I&M Functions“ spezifiziert.
- Auskunftsfunctionen über reale und logische Modulstrukturierung.
- Rücklesen der I/O-Daten.

Für den direkten Datenaustausch mit mehreren Partnern wurde der Datenquererverkehr (Multicast Communication Relation, MCR) definiert. MCRs innerhalb eines Segments werden als RT-Frames des Typs RT_CLASS_1 oder RT_CLASS_2 ausgetauscht. Segment übergreifende MCR-Daten folgen dem Datenaustausch nach RT_CLASS_UDP. Die Daten, die über MCR auszutauschen sind, unterliegen dem IO-Gerätemodell und sind Subslots zugeordnet.

Die Übertragung von Ereignissen (Events) ist bei PROFINET IO innerhalb des Alarmkonzepts modelliert. Es deckt sowohl systemdefinierte Ereignisse (wie Ziehen und Stecken von Baugruppen) ab, als auch anwenderdefinierte Ereignisse, die in der eingesetzten Steuerungstechnik erkannt wurden (z. B. Lastspannung defekt) bzw. schon im zu steuern Prozess auftreten (z. B. Temperatur zu hoch). Beim Auftreten eines Ereignisses muss für die Datenübertragung ausreichend Kommunikationsspeicher bereit stehen, damit ein Datenverlust ausgeschlossen werden kann und die Alarrrmeldung vom IO-Device schnell transportiert werden kann. Dies ist Sache der Anwendung in der Datenquelle. Alarme gehören zu den azyklischen RT-Daten.

Netzwerk-Diagnose bei PROFINET

Zu Wartungszwecken und Überwachung der Netzwerk-Komponenten hat sich als internationaler Standard SNMP (Simple Network Management Protocol) etabliert. SNMP kann lesend als auch schreibend (für Administration) auf Netzwerk-Komponenten zugreifen, um Statistikdaten, die das Netzwerk betreffen, sowie portspezifische Daten und Informationen zur Nachbarschaftserkennung auszulesen. Eine weitere Möglichkeit, ein Netzwerk zu diagnostizieren, ist die Integration von Überwachungsfunktionen direkt in die Netzwerk-Komponenten wie Switches. IEEE-konforme Standard-Switches sind nur dazu ausgelegt, die Diagnoseinformationen der angeschlossenen Feldgeräte an einen IO-Controller weiterzuleiten. Zusätzliche Überwachungsfunktionen sind normalerweise nicht integriert. Switches können auch als IO-Devices ausgelegt werden und können solche Überwachungsfunktionen besitzen.

Anlagen-Engineering und GSD

Um ein Anlagen-Engineering durchführen zu können, sind die GSD-Dateien der zu projektierenden Feldgeräte erforderlich. Diese hat der Feldgerätehersteller zu liefern. Der Projektor führt die in der GSD-Datei definierten Module/Submodule zusammen, um sie auf die reale Anlage abzubilden und den Slots/Subslots zuzuordnen.

Jedem Feldgerät wird ein logischer Name zugeordnet, der einen Bezug zur Funktion in der Anlage oder zum Einbauort haben sollte und schließlich bei der Adressauflösung zur Zuteilung einer IP-Adresse führt. Die Namenszuweisung kann immer mit dem standardmäßig in jedem PROFINET IO-Feldgerät integrierten DCP-Protokoll (Discovery and Configuration Protocol) erfolgen. Da DHCP (Dynamic Host Configuration Protocol) international große Verbreitung gefunden hat, sieht PROFINET die Adresseinstellung optional über DHCP oder über herstellerspezifische Mechanismen vor. Welche Möglichkeiten ein IO-Feldgerät unterstützt, ist in der GSD-Datei für das jeweilige Feldgerät definiert.

Systemhochlauf

Unter Systemhochlauf versteht man den Anlauf oder Wiederanlauf einer Automatisierungsanlage nach „Power on“ oder nach einem „Reset“. Er wird von einem IO-Controller anhand der Projektierungsdaten angestoßen und läuft aus Sicht des Anwenders selbstständig ab. Jeder Datenaustausch ist in eine Application Relation (AR) eingebettet. Hierzu wird zwischen der überlagerten Steuerung (IO-Controller oder IO-Supervisor) eine genau spezifizierte Applikation (Verbindung) aufgebaut. Beim Einrichten der Applikationsbeziehung werden neben allgemeinen Kommunikationsparametern alle Daten für die Gerätemodellierung in das IO-Device geladen. Gleichzeitig werden die Kommunikationskanäle für den zyklischen/azyklischen Datenaustausch (IO Data CR, Record Data CR), die Alarne (Alarm CR) und die Querverkehrsbeziehungen (MCR) eingerichtet.

Innerhalb einer Application Relation AR müssen Kommunikationsbeziehungen (CR) für den Datenaustausch aufgebaut werden. Damit ist ein eindeutiger Kommunikationskanal zwischen einem Consumer und einem Provider spezifiziert.

Nachbarschaftserkennung

Automatisierungsanlagen können in Form einer Linienstruktur, sternförmig oder baumförmig aufgebaut sein. Aus diesem Grund ist es wichtig zu wissen, welche Feldgeräte an welchem Switch-Port angeschlossen sind und wer der jeweilige Port-Nachbar ist. Die Nachbarschaftserkennung mit dem „Link Layer Discovery Protocol“ (LLDP) nach IEEE 802.1 AB und den PROFINET-spezifischen Erweiterungen ist Teil des Gesamtkonzepts „Gerätetausch ohne Engineering Tool“. Die Daten der Nachbargeräte werden mit den LLDP-Diensten port-granular ermittelt und der übergeordneten Steuerung zur Verfügung gestellt (Abb. 4.60). Aus dieser Kombination kann eine Anlagentopologie und eine komfortable Anlagendiagnose nachgebildet werden, sowie ein Gerätetausch ohne zusätzliche Hilfsmittel vollzogen werden.

PROFINET Feldgeräte tauschen über jeden Switchport die vorhandenen Adressierungs-Informationen mit den angeschlossenen Nachbargeräten aus, die zur eindeutigen Identifi-

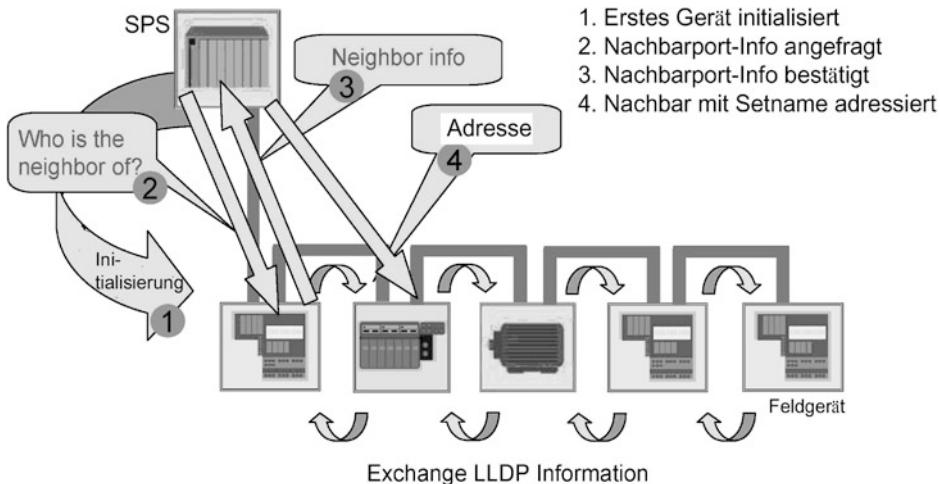


Abb. 4.60 PROFINET unterstützt die Ermittlung der Anlagentopologie und den komfortablen Gerätetausch

fikation und zur Ermittlung der physikalischen Lage dienen. Das LLDP-Protokoll ist in Software realisiert und beansprucht daher keine spezielle Hardware-Unterstützung.

Durch die Erkennung der Anlagentopologie ist es möglich, beim Ausfall eines Feldgerätes zu überprüfen, ob das Ersatzgerät an der richtigen Position angeschlossen wurde. Auch die Forderung der Anlagenbetreiber, einen Gerätetausch ohne zusätzlichem Engineering Tools zu ermöglichen, wird durch den Einsatz von PROFINET-Feldgeräten erfüllt.

4.3.4.3 IRT-Kommunikation bei PROFINET IO

Manche Prozesse erfordern es, I/O-Daten isochron und mit höchstmöglicher Performance zu übertragen. Isochron bedeutet dabei, dass der Beginn eines Buszyklus taktgenau erfolgt, d. h. mit einer maximal zulässigen Abweichung (Jitter) beginnt und ständig synchronisiert wird. PROFINET IO hat dafür die synchronisierte IRT-Kommunikation (Isochronous Real-Time Communication) eingeführt, mit der Buszyklen deutlich kleiner einer Millisekunde mit einer maximalen Abweichung vom Bustakt von $1 \mu\text{s}$ erzielt werden.

Für den isochronen Datenaustausch bietet PROFINET ein skalierbares Konzept an, das einerseits eine sehr flexible Art der Kommunikation vorsieht. Technisch gesehen handelt es sich hierbei um eine synchronisierte RT_CLASS_2-Kommunikation.

Andererseits bietet PROFINET eine auf höchste Performance ausgelegte Kommunikation, die eine genaue Planung der Kommunikationswege im Vorfeld voraussetzt. Die verfügbare Bandbreite wird hier optimal ausgenutzt, da es zu keinem Zeitpunkt zu Wartezeiten bei der Datenübertragung kommen kann. Technisch betrachtet handelt es sich um eine synchronisierte RT_CLASS_3-Kommunikation.

Der Determinismus ist für beide Varianten gleich, sie unterscheiden sich nur im Datendurchsatz. Die Kommunikation wird in ein reserviertes und in ein offenes Intervall

aufgeteilt. In dem reservierten Intervall werden nur die zeitkritischen I/O-Daten übertragen, während in der offenen Phase alle anderen Daten gesendet werden. Hierzu ist kein zusätzliches unterlagertes Protokoll erforderlich.

Diese auf höchste Performance getrimmte Datenübertragung erfordert eine Hardwareunterstützung bei den eingesetzten Switches. Die Synchronisation der Teilnehmer übernimmt ein definierter Clock-Master, der normalerweise im IO-Controller integriert ist.

IRT-Steuerung

Alle an einer IRT-Kommunikation beteiligten Feldgeräte werden vom selben Clock-Master synchronisiert. Die IRT-Kommunikation erfolgt ausschließlich innerhalb *eines* Subnetzes, da die Adressierungsmöglichkeiten über TCP/IP fehlen. Die vorhandenen Adressierungsmechanismen wurden (wie auch bei der unsynchronisierten Kommunikation) so reduziert, dass eine Adressierung der Feldgeräte innerhalb eines Subnetzes anhand der MAC-Adresse ausreicht. Die Aufteilung des Buszyklusses erfolgt in reservierte IRT- und offene Phasen. Im reservierten Intervall (IRT-Phase) werden nur IRT-Aufträge bearbeitet, im offenen Intervall wird die Auftragsbearbeitung nach den Regeln von IEEE 802 abgewickelt. Alle Feldgeräte innerhalb einer IRT-Domäne müssen die Isochronität unterstützen, auch wenn die Applikation nicht synchron arbeitet.

Die IRT-Überwachungsfunktionen müssen aufgrund der hohen Anforderungen von der eingesetzten Hardware unterstützt werden.

IRT-Intervalle

Die IRT-Übertragung bei PROFINET IO definiert die vier Intervalle grün, rot, orange und gelb, die jeweils nur für ihnen zugeordnete Daten genutzt werden dürfen (Abb. 4.61). Im roten Intervall dürfen nur RT_CLASS_3-Frames durch die Switches weitergeleitet werden. Die Weiterleitregeln nach IEEE802.1D werden hier durch die Regeln gemäß IEC61158 ersetzt. Der Startzeitpunkt des roten Intervalls wird ständig synchronisiert. Alle RT_CLASS_3-Frames werden bereits beim Engineering in ihrer zeitlichen Abfolge festgelegt. Wenn während eines roten Intervalls UDP/IP-Frames eintreffen oder generiert werden, werden sie in einem IRT-fähigen Switch zwischengespeichert und erst nach Beendigung des reservierten roten Intervalls gesendet. Die benutzten Frame IDs zur Identifizierung der unterschiedlichen Frames werden während der Anlagen-Konfiguration im Engineering-Tool festgelegt. Die Zeitpunkte für den Empfang der zyklischen Daten liegen genau fest, sodass die synchrone Applikation direkt ohne Verzögerungen gestartet werden kann.

Im orangen Intervall werden nur RT_CLASS_2-Frames durch die Switches weitergeleitet. Die Weiterleiteregeln nach IEEE802.1D finden hier ihre Anwendung. Das orangefarbene Intervall startet (wenn vorhanden) unmittelbar am Anfang eines Send Clocks oder nach dem roten Intervall. RT_CLASS_2-Frames bedürfen keiner vorherigen Planung. Dadurch wird auch die verfügbare Bandbreite nicht optimal ausgenutzt. Die Zeitpunkte für den Empfang der zyklischen Daten liegen nicht exakt fest. Daher ist eine Sicherheitsreserve einzuplanen.

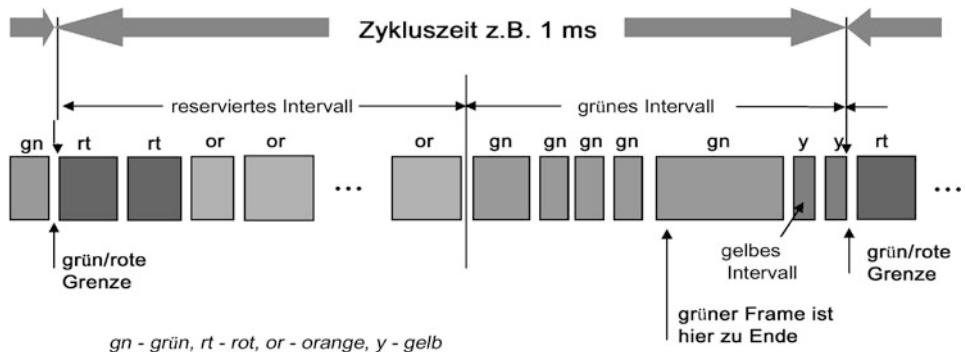


Abb. 4.61 Die IRT-Kommunikation teilt den Buszyklus in eine reservierte und eine offene Phase auf

Im grünen Intervall kommen für die Weiterleitung der Datenframes in den Switches die Regeln gemäß IEEE 802.1D zum Tragen. Die Priorisierung kann anhand der IEEE802.1Q (VLAN-Tag) erfolgen. Aufgrund der maximalen Framelänge bei Ethernet/PROFINET ergibt sich eine Mindestdauer des grünen Intervalls von 125 µs ($4 \times 31,25 \mu\text{s}$); es muss aber innerhalb einer Phase nicht zwangsläufig vorhanden sein.

Im gelben Intervall werden für die Weiterleitung der Datenframes in den Switches die Regeln gemäß IEEE 802.1D evtl. außer Kraft gesetzt, um den Beginn der nächsten reservierten Phase sicher einhalten zu können. Die Priorisierung kann anhand der IEEE802.1Q (VLAN-Tag) erfolgen.

4.3.4.4 PROFINET IO-Controller und -Devices

Ein PROFINET IO-Controller ist die Station in einer Automatisierungsanlage, in der das Steuerungsprogramm abläuft. Er fordert die Prozessdaten (Eingänge von den projektierten IO-Devices im Hochlauf) an, bearbeitet sein Steuerungsprogramm und überträgt die auszugebenden Prozessdaten (Ausgänge) an die jeweiligen IO-Devices. Um diesen Datenaustausch durchführen zu können, benötigt er die Daten der Anlagenprojektierung, in der alle Kommunikationsdaten enthalten sind. Bei der Anlagenprojektierung werden der Ausbaugrad eines IO-Devices, die Parametrierungen für ein IO-Device, die Übertragungshäufigkeit, der Ausbaugrad der Automatisierungsanlage und Informationen zu Alarmen und Diagnosen festgelegt. Der IO-Controller baut die Application Relations (AR) und die Communication Relations (CR) zu den projektierten IO-Devices selbstständig auf.

In einem PROFINET-System können mehrere IO-Controller eingesetzt werden. Sollen diese IO-Controller auf dieselben Daten in den IO-Devices zugreifen können, so ist das bereits beim Projektieren anzugeben (Shared Devices, Shared Inputs).

Ein IO-Controller kann zu mehreren IO-Devices jeweils eine AR aufbauen. Innerhalb einer AR können mehrere IOCRs und APIs (Application Process Identifier) für den Datenaustausch genutzt werden. Dies kann beispielsweise sinnvoll sein, wenn mehrere

Anwenderprofile (PROFIdrive, Encoder, etc.) an der Kommunikation beteiligt sind, die unterschiedliche Subslots benötigen. Innerhalb einer IOCR erfolgt die Unterscheidung durch die angegebenen APIs. Dadurch ist eine Vermischung der Daten zwischen den APIs nicht möglich. Der Zugriff auf die Anwenderdaten ist durch das Anwenderprogramm zu koordinieren. PROFINET IO lässt zu, dass mehrere Anwenderprofile innerhalb derselben AR definiert werden können.

Die Funktionalität der PROFINET IO-Devices wird mit der XML-basierten GSD beschrieben. Den internationalen Standards folgend, ist die Beschreibungssprache GSDML (General Station Description Markup Language) entstanden, die als XML-Datei Sprachen unabhängig ist. Jeder Hersteller eines PROFINET IO-Devices muss eine dazugehörige GSD-Datei gemäß GSDML-Spezifikation liefern, deren Prüfung auch Bestandteil eines Zertifizierungstests ist. Zur Beschreibung von PROFINET IO-Devices wird jedem Hersteller ein XML-Schema zur Verfügung gestellt. Dadurch ist ein leichtes Erstellen und Prüfen einer GSD-Datei möglich.

4.3.4.5 Conformance Classes (CC)

Da nicht immer der komplette Funktionsumfang von PROFINET IO benötigt wird, ist es hinsichtlich unterstützter Funktionalität skalierbar in Conformance Classes (CC) eingeteilt. Die daraus resultierenden Applikationsklassen ermöglichen dem Anlagenbetreiber, eine einfache Auswahl von Feldgeräten und Buskomponenten mit eindeutig definierten Mindesteigenschaften zu treffen. Die Conformance Classes sind außerdem Grundlage für die Zertifizierung und für die Verkabelungsrichtlinien.

Die CC-A bietet Grundfunktionen für PROFINET IO mit RT-Kommunikation bei Nutzung der Infrastruktur eines bestehenden Ethernet-Netzwerkes. Alle IT-Services können uneingeschränkt eingesetzt werden. Typische Anwendungen findet man beispielsweise in der Gebäudeautomation oder der Prozessautomatisierung.

Die CC-B erweitert das Konzept um Netzwermdiagnose über IT-Mechanismen sowie Topologieinformationen. Zusätzlich zur CC-A ermöglicht der Funktionsumfang der CC-B einen einfachen und komfortablen Gerätetausch ohne Engineeringtool. Typische Anwendungen findet man beispielsweise in Automatisierungsanlagen mit überlagerter Maschinensteuerung bei eher geringen Ansprüchen an einen deterministischen Datenzyklus. Die für die Prozessautomatisierung wichtige Funktion „Systemredundanz“ ist in einer Erweiterung der CC-B zur CC-B(PA) enthalten.

Zusätzlich zur CC-B beschreibt die CC-C die Basisfunktionen für Geräte mit hardware-unterstützter Bandbreitenreservierung und Synchronisation (IRT-Kommunikation) und ist damit Basis für taktsynchrone Applikationen. Die integrierte Medienredundanz ermöglicht die stoßfreie Umschaltung des I/O-Datenverkehrs im Fehlerfall. Typische Anwendungen findet man z. B. im Bereich Motion Control.

4.3.4.6 Applikationsprofile für PROFINET IO

Applikationsprofile sind von Herstellern und Anwendern getroffene, gemeinsame Festlegungen (Spezifikationen) über bestimmte Eigenschaften, Leistungsmerkmale und Ver-

haltensweisen von Geräten und Systemen. PROFIBUS bietet eine große Zahl solcher Profile, die nach Bedarf Schritt für Schritt in PROFINET übernommen werden. Dies ist für PROFIsafe, PROFIdrive, Encoder, Low Voltage Switch Gear und Identifikationssysteme bereits realisiert. Die Profile PROFIsafe und PROFIdrive sind im PROFIBUS-Kapitel dieses Buches näher beschrieben. Speziell für PROFINET wurden in enger Zusammenarbeit mit Anwendern die Profile Train Applications und PROFIenergy entwickelt.

Train Applications

Das Profil Train Applications legt die Anwendungsebene für Geräte der Bahnautomatisierung fest. Durch den Einsatz von PROFINET stehen damit Ethernet-basierte Echtzeit- und IT-Kommunikation für Anwendungen in Schienenfahrzeugen zur Verfügung.

PROFIenergy

In enger Zusammenarbeit mit der AIDA hat PI das Profil PROFIenergy entwickelt, mit dem das Energiemanagement in Anlagen effizienter gestaltet werden kann. Das PROFIenergy-Profil stellt die Basis für ein abgestimmtes und standardisiertes Vorgehen beim Energiemanagement von Anlagen mit PROFINET-Funktionalität dar. Durch zielgerichtetes Abschalten von nicht benötigten Verbrauchern lassen sich Energiebedarf und Kosten deutlich senken. Studien bestätigen ein Einsparpotenzial von 50 % und mehr in Stillstandphasen durch PROFIenergy.

Damit das neue Profil so praxisnah wie möglich arbeiten kann, wurden folgende Use Cases (UC) definiert:

UC 1 (Ab- und Zuschalten in kurzen Pausen) Bei diesen Pausen mit einer typischen Dauer unter einer Stunde, zum Beispiel Mittagspausen, hält das System die Anlage definiert an und schaltet Energieverbraucher aus, die in kurzen Zeiträumen Energie sparen, aber auch schnell genug wieder hochfahren können. Dementsprechend werden nur einzelne Geräte oder Teilanlagen angesprochen. Wichtige sicherheitstechnische Funktionen bleiben weiterhin erhalten. Bei der Aufnahme des Produktionsbetriebes aktiviert das System die Verbraucher in einer definierten Einschaltreihenfolge und prüft, ob alle Verbraucher korrekt angelaufen sind.

UC 2 (Ab- und Zuschalten in langen Pausen) Aufgrund der längeren Abschaltzeiten in Vergleich zu UC 1 – typisch mehrere Stunden oder einige Tage – können zusätzliche Geräte abgeschaltet werden. Heute liegt in vielen Betrieben in der produktionsfreien Zeit am Wochenende der Energieverbrauch der Anlagen im Vergleich zur Produktion noch bei rund 60 Prozent, da Abschaltung und Wiederanlauf der Anlagen zu aufwändig im Engineering und in eventuell zusätzlich zu installierender Hardware ist.

UC 3 (Ab- und Zuschalten in ungeplanten Pausen) Im Gegensatz zu UC 1 und UC 2 sind der Zeitpunkt und die Dauer der Pause unbekannt. Klassisches Beispiel sind Unterbrechungen auf Grund von Anlagenstörungen. Deshalb wird zunächst der Energiebedarf

soweit abgesenkt, als handle es sich um eine kurze Pause. Stellt sich heraus, dass die Servicearbeiten doch länger dauern, so besteht die Möglichkeit, die Anlage in einen noch energiesparenden Zustand zu versetzen.

UC 4 (Erfassen von Messdaten) Neben Messgeräten im eigentlichen Sinne ist in der Anlage eine Vielzahl von Geräten eingebaut, die Energiewerte heute schon implizit erfassen. Klassische Vertreter davon sind z. B. Frequenzumrichter. Unter Nutzung dieser Daten erlaubt PROFIenergy auch eine lastabhängige Maschinensteuerung sowie das Vermeiden von Lastspitzen.

Die Handhabung von PROFIenergy in der Praxis ist denkbar einfach. Die Anwendung sendet dem Gerät ein Kommando, das die Länge der Pausenzeit enthält. Das unterlagerte Gerät entscheidet daraufhin selbstständig, welche Teile abschaltbar sind, um nach Ablauf dieser Zeit wieder betriebsbereit zu sein. Dabei ist es gleichgültig, ob es sich bei dem Gerät um eine Einzelkomponente, wie beispielsweise einen Antrieb, oder um ein komplexes Gerät, wie z. B. eine Werkzeugmaschine, handelt.

4.3.4.7 Integration von Feldbus-Systemen

PROFINET spezifiziert ein Modell zur Einbindung existierender PROFIBUS- und anderer Feldbus-Systeme. Damit lassen sich beliebige Mischsysteme aus Feldbus- und Ethernet basierten Teilsystemen aufbauen. So wird ein kontinuierlicher Technologieübergang von feldbusbasierten Systemen zu PROFINET möglich.

Feldbus-Lösungen lassen sich über Proxies bzw. Gateways einfach und nahtlos in ein PROFINET-System einbinden. Der Proxy fungiert hierbei als Repräsentant der Feldbus-Geräte am Ethernet. Er integriert die an ein unterlagertes Feldbussystem angeschlossenen Teilnehmer in das übergeordnete PROFINET-System. Bei PROFIBUS DP beispielsweise arbeitet der Proxy auf der einen Seite als PROFIBUS-Master, der den Datenaustausch mit den PROFIBUS-Teilnehmern durchführt. Auf der anderen Seite ist er ein Ethernet-Teilnehmer mit Ethernet-basierter PROFINET-Kommunikation. Proxies können beispielsweise in SPS, in PC-based Control oder als reine Gateways realisiert werden.

Mit Hilfe der vorgestellten Integrationsmethoden ist es möglich, neben PROFIBUS beliebige andere Feldbussysteme wie INTERBUS, Foundation Fieldbus, DeviceNet, usw. in PROFINET zu integrieren. Dabei ist die Ethernet-Kommunikation durch die zur Verfügung stehende Software bereits definiert. Es ist nur noch die Versorgung der vom unterlagerten Feldbus bereit gestellten Prozessdaten an PROFINET sicher zu stellen. Dieses Konzept ermöglicht es, beliebige Feldbusse mit geringem Aufwand an PROFINET anzubinden, indem ein Proxy den Repräsentanten des unterlagerten Bussystems darstellt.

4.3.4.8 Netzwerkinstallation

Die internationale Norm ISO/IEC 11801 sowie ihr europäisches Äquivalent EN 50173 definieren eine anwendungsneutrale, informationstechnische Standardvernetzung für einen Gebäudekomplex (Abb. 4.62). Dieser Verkabelungsstandard bildet das Rückgrat für die Anforderungen an eine Ethernetverkabelung auch in der industriellen Automatisierung.

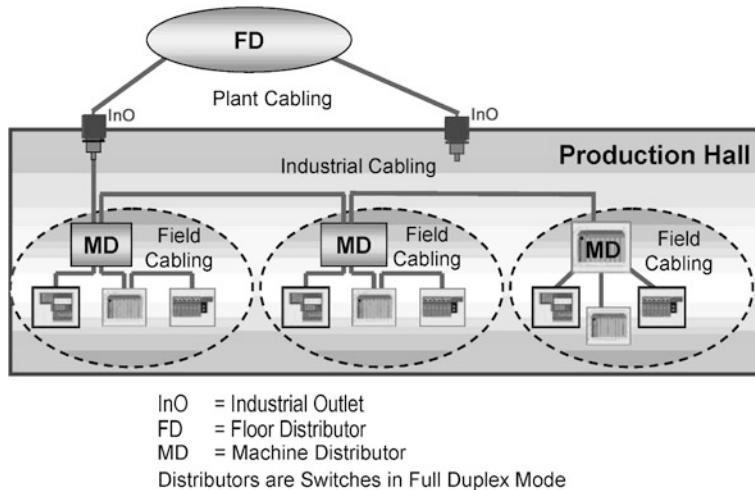


Abb. 4.62 Typische Struktur eines Ethernet-Netzwerks im industriellen Umfeld

PROFINET Verkabelung

Die PROFINET Verkabelung basiert auf der Norm IEC 61918. PROFINET-spezifische Festlegungen enthält wiederum die IEC 61784-5-3. Nutzt der Anwender aber bereits Netzwerke, die der ISO/IEC 11801 entsprechen, so können diese unter entsprechenden Randbedingungen auch für PROFINET eingesetzt werden.

Entsprechend der Offenheit von PROFINET ist für die Conformance Class A auch eine geschirmte, generische Verkabelung nach ISO/IEC 24702 nutzbar. Für alle Automatisierungsapplikationen der Conformance Classes B und C muss die PROFINET Verkabelung eingesetzt werden, die sich durch hohe Performance und einfache Planung und Montage auszeichnet.

Es wird immer mit beidseitig gleich konfektionierten Systemkabeln gearbeitet. Beim Anschluss wird auf das Crossing verzichtet, da die PROFINET-Netzkomponenten das Auto-Crossing unterstützen. Neben den Systemkabeln gibt es passive Koppler, die entweder zur Durchführung durch die Schaltschränke oder als Kupplung eingesetzt werden. Damit lassen sich alle Übertragungsstrecken einfach realisieren. Um die Komplexität bei der Konfektion vor Ort zu reduzieren, wurde bei den Kabeln und Steckverbindern durchgängig eine zweipaarige Technik vorgegeben. Die Kupfer-Kabel sind einheitlich in AWG 22 ausgeführt. Das liefert ausreichend Systemreserve auch für viele Kabel/Steckverbinder-Übergänge.

PROFINET unterstützt sowohl die Linientopologie, die zu einem Ring geschlossen werden kann und die vorrangig Endgeräten mit integrierten Switches im Feld verbindet, als auch die Sterntopologie, die einen zentralen Switch voraussetzt, der sich vorrangig im Schaltschrank befindet. Beide Topologien können zu komplexen Baumtopologien kombiniert werden.

Die PROFINET Umgebungsklassen für die Automatisierungsapplikation wurden in eine Inside Enclosure-Klasse innerhalb geschützter Umgebungen wie z. B. in einem Schaltschrank und eine Outside Enclosure-Klasse außerhalb von Schaltschränken bei Anwendungen direkt im Feld unterteilt.

Die PROFINET Kabel entsprechen den in der Industrie verwendeten Kabeltypen

- A: Standard fest verlegt, keine Bewegung nach der Installation
- B: Standard flexibel, gelegentlich Bewegung oder Vibration
- C: Sonderanwendungen wie hochflexibel und/oder permanente Bewegung (Schleppkette oder Torsion).

Faseroptische Datenübertragung bietet sich durch die galvanische Trennung vor allem dann an, wenn ein Potentialausgleich zwischen einzelnen Anlagenbereichen schwer herzustellen ist. Auch für extreme EMV-Anforderungen bietet die optische Faser Vorteile gegenüber Kupfer. Für faseroptische Übertragungstechnik wird die 1 mm polymeroptische Fasern (POF) unterstützt, die in der Handhabung optimal den industriellen Anforderungen entspricht.

Steckverbinder für Daten

Die Auswahl der Steckverbinder wird von der Applikation bestimmt. Steht ein universelles Netzwerk im Vordergrund, das zum Office kompatibel sein soll, kommt für die Datenübertragung der RJ 45-Stecker zum Einsatz. Für die Feldumgebung außerhalb geschützter Räume oder Schränke wurde ein PushPull-Steckverbinder entwickelt, der für die Datenübertragung ebenfalls mit dem RJ 45 bestückt wird. Auch der M12-Stecker ist für PROFINET spezifiziert.

Für die optische Datenübertragung werden wegen ihrer einfachen Montage zumeist polymeroptische Fasern (POF) verwendet. Für PROFINET wurde der SC-RJ spezifiziert, der auf dem SC-Steckverbinder basiert. Er kommt sowohl „Inside“ als auch in Verbindung mit dem PushPull-Gehäuse in der „Outside“-Umgebung zum Einsatz. Auch für die M12-Familie steht ein optischer Steckverbinder zur Verfügung, der für die 1mm POF eingesetzt werden kann.

Steckverbinder für die Stromversorgung

Abhängig von der Topologie wurden unterschiedliche Steckverbinder definiert, die in zwei Power Classes eingeteilt sind. In der Linientopologie kommt insbesondere in der Automobilindustrie der 4-polige PushPull-Steckverbinder mit zusätzlicher Funktionserde zum Einsatz. Der maximale Leiterquerschnitt beträgt $2,5 \text{ mm}^2$. Der Steckverbinder ist ausgelegt für eine Stromtragfähigkeit von 16 A. Diese hohe Stromtragfähigkeit ermöglicht den Aufbau von längeren Liniensstrukturen, d. h., es können sehr viele Geräte über lange Übertragungsstrecken über T-Stücke versorgt werden. Alternativ zum PushPull Steckverbinder kann auch der 7/8“ Steckverbinder oder der 3 A Hybridsteckverbinder RJ 45 eingesetzt werden.

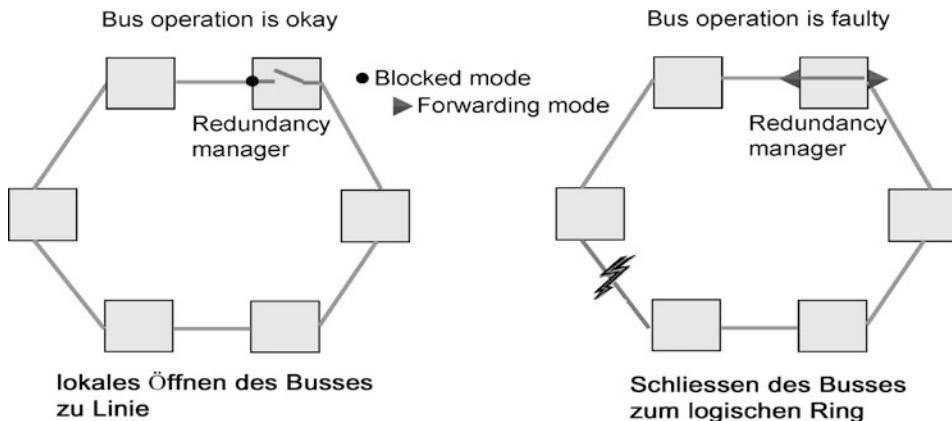


Abb. 4.63 Ringstruktur mit Redundanz-Manager

In der Sterntopologie kommt ein M12-Steckverbinder zum Einsatz, der auf einen Stromkreis und auf einen Strom von 4 A limitiert ist. Zum Aufbau eines Power-Bus wurde ein Steckverbinder mit hoher Stromtragfähigkeit definiert. Bei der 400 Volt-Versorgung nutzt PROFINET den international standardisierten Power-Bus gemäß ISO 23570-3.

Redundanz

Redundante Kommunikationswege sind in einer Automatisierungsanlage in bestimmten Fällen notwendig, um die Anlagenverfügbarkeit deutlich zu erhöhen. PROFINET besitzt Redundanz-Mechanismen mit dem Media Redundancy Protocol (MRP) nach IEC 61158 bzw. der neuen IEC 62439 (High Availability) mit typischer Rekonfigurierzeit der Kommunikationswege bei TCP/IP- und RT-Frames im Fehlerfall von < 200 ms. Ein Redundanz-Manager (RM) überprüft die durch die Projektierung vorgegebene Ringstruktur auf Funktionstüchtigkeit durch das Aussenden von zyklischen Testframes (Abb. 4.63). Solange die Testframes von ihm selbst wieder empfangen werden, ist die Ringstruktur intakt. Durch dieses Verhalten verhindert ein RM das Zirkulieren von Frames und führt eine Ringstruktur in eine Linienstruktur über. Der RM muss Änderungen im Ring allen beteiligten Clients (Switches als sogenannte „Durchreicher“) durch spezielle Änderungen in den Topologie-Frames mitteilen.

Das MRRT-Protokoll (Media Redundancy for Real-Time) gemäß IEC 61158 beschreibt die Behandlung von RT-Frames der RT_CLASS_1 und RT_CLASS_2 für den Redundanzbetrieb. Für den Betrieb von MRRT wird immer der Betrieb von MRP vorausgesetzt. Die IEC 61784 beschreibt wie das MRRT-Protokoll angewendet wird. Mit dem MRRT-Protokoll ist bei der RT-Kommunikation eine quasi stoßfreie Umschaltung der Kommunikationswege im Fehlerfall möglich. Auf Empfängerseite treffen bei fehlerfreier, redundanter Übertragung immer 2 RT-Frames ein. Zur Applikation wird nur der zuerst eintreffende Frame weitergeleitet.

Die IEC 61158 beschreibt das Redundanz-Konzept für RT_CLASS_3-Frames als „Media Redundancy for planned duplication“. In der IEC 61784 ist der Einsatz der Redundanzklasse 3 für RT_CLASS_3-Kommunikation mit stoßfreier Umschaltung der Kommunikationswege im Fehlerfall beschrieben. Der IO-Controller lädt im System-Hochlauf die Daten der Kommunikationswege für beide Kommunikationskanäle (Richtungen) in einem Kommunikationsring in die einzelnen Teilnehmer. Somit ist es unerheblich, welcher Teilnehmer ausfällt, da der geladene „Fahrplan“ für beide Wege in den Feldgeräten vorhanden ist und in jedem Fall überwacht und eingehalten wird. Alleine durch das Laden des „Fahrplans“ ist das Kreisen von Frames in dieser Variante ausgeschlossen, da die Zielports eindeutig definiert sind.

Industrial Wireless

Die Flexibilität und Mobilität drahtloser Netzinfrastrukturen ermöglicht neue Lösungen in Bereichen, wo elektrische Leitungen aufgrund mechanischer Begrenzungen, Sicherheitsanforderungen oder anderer Umgebungsbedingungen nicht oder nur eingeschränkt geeignet sind. PROFINET ermöglicht die Kommunikation über drahtlose Kommunikationsnetze mit WLAN und Blue Tooth hinweg und ist damit in der Lage, mit unterschiedlichen Funktechnologien für verschiedenste Einsatzgebiete mit jeweils spezifischen Parametern hinsichtlich Transferrate, Reichweite, Teilnehmeranzahl und ähnlichem umzugehen. Deshalb werden entsprechend den jeweiligen Technologien Profile spezifiziert, die beschreiben, wie die Integration in PROFINET erfolgt, welche Topologien und Performancewerte mit der Technologie realisierbar sind und welche Randbedingungen, z. B. bei Security-Anforderungen, gelten.

4.3.4.9 PROFINET IO-Zertifizierung

PROFINET ist in der IEC 61158 genormt. Die Norm ist Basis dafür, dass Geräte in industriellen Anlagen miteinander vernetzt werden können und ihre Daten fehlerfrei miteinander austauschen können. Die Sicherstellung von Interoperabilität in Automatisierungsanlagen erfordert entsprechende Qualitätssicherungsmaßnahmen. Aus diesem Grund hat PROFIBUS/PROFINET International (PI) ein Zertifizierungsverfahren etabliert, bei dem auf der Basis von Prüfberichten, die von akkreditierten Prüflabors stammen, Zertifikate für PROFINET-Geräte ausgestellt werden.

Während bei PROFIBUS die Zertifizierung eines Feldgerätes von PI empfohlen, aber nicht vorgeschrieben ist, hat man die Zertifizierung aller Feldgeräte, die den Namen PROFINET tragen, zur Pflicht erhoben. Zertifizierte Geräte garantieren eine weltweite Konformität eines PROFINET-Produkts in einer Anlage mit Teilnehmern unterschiedlicher Hersteller.

4.3.5 Ethernet/IP

Ethernet Industrial Protocol (EtherNet/IP) ist ein offener Standard der Fa. Rockwell für industrielle Netzwerke, der zyklische E/A-Nachrichtenübertragung sowie azyklische (explizite) Nachrichtenübertragung unterstützt und mit kommerziellen, standardmäßigen Ethernet-Kommunikationschips und physikalischen Medien arbeitet. Der Einsatz von Ethernet-Produkten ist nicht nur ein allgemeiner Trend in der Technologie – dank Ethernet können Benutzer auf alle Daten der Geräteebenen zugreifen – sogar vom Internet aus. EtherNet/IP wurde geschaffen, weil sich in der Industrie ein Trend abzeichnet, Ethernet-Netzwerke auch in Steuerungsanwendungen einzusetzen.

EtherNet/IP gilt als offenes Netzwerk, da es folgende Standards benutzt:

- Standard IEEE 802.3 (Physical Media und Data Link)
- Ethernet-TCP/IP-Protokolle (Transmission Control Protocol/Internet Protocol), der Ethernet-Industriestandard
- Control and Information Protocol (CIP) – das Protokoll, das Echtzeit-E/A-Nachrichtenübertragung und Information/Peer-to-Peer-Nachrichtenübertragung bietet. ControlNet (IEC61158 Teil 6) und DeviceNet-Netzwerke (EN50325 Teil 2) nutzen ebenfalls CIP.

TCP/IP ist das Transport- und Netzwerkprotokoll des Internets und wird üblicherweise mit Ethernet-Installationen und Business-Anwendungen in Verbindung gebracht. TCP (Transmission Control Protocol) beschreibt ein Verfahren, das zusammen mit dem Internet Protocol (IP) verwendet wird, um über das Netzwerk (Internet/Intranet) Nachrichten zwischen Computern zu versenden. Während IP für die eigentliche Übermittlung der Daten zuständig ist, überwacht TCP die einzelnen Datenpakete einer Nachricht, um so sichere Punkt-zu-Punkt-Verbindungen zu gewährleisten. Die Ethernet-Technologie und die Protokollumgebung wie TCP/IP sind inzwischen allgemein akzeptiert, es gibt viele standardisierte Software-Tools und Netzwerkprodukte. Der Vorteil ist, dass man mit einer bekannten, bewährten und verfügbaren Technologie arbeiten kann.

Auch das UDP/IP-Protokoll (User Datagram Protocol) wird zusammen mit Ethernet-Netzwerken eingesetzt. UDP/IP ermöglicht den schnellen und effizienten Datentransport, wie er für den Datenaustausch in Echtzeit erforderlich ist. Um den Erfolg von EtherNet/IP zu sichern, wurde CIP auf TCP/UDP/IP aufgesetzt, wodurch eine gemeinsame Anwendungsschicht zur Verfügung steht. Aus diesem Grund erhalten Sie, wenn Sie ein EtherNet/IP-Produkt auswählen, gleichzeitig auch ein Produkt mit CIP-Funktionen. Zudem setzt EtherNet/IP – ebenso wie DeviceNet- und ControlNet-Netzwerke – das Producer/Consumer-Netzwerkmodell ein. Mit der Einführung von fast Ethernet sowie der Ethernet-Switch-Technologie und der Vollduplex-Datenübertragung wurden Datenkolliktionen eliminiert und die Performance des Netzwerks deutlich gesteigert.

Geräteanwendungen

Zu den Geräten, die typischerweise über ein EtherNet/IP-Netzwerk kommunizieren, gehören:

- Mainframe-Rechner
- SPS
- Roboter
- HMI
- E/A-Adapter

Ziel-Anwendungen sind unter anderem:

- Interaktion zwischen Plant Management und MES-Systemen (Manufacturing Execution Systems), Fördertechniksystemen, SCADA-Applikationen u.v.m.
- Konfiguration, Datenerfassung und Steuerung in einem einzigen High-Speed Netzwerk
- Zeitkritische Anwendungen ohne festgelegten Zeitplan (wie sie von ControlNet zur Verfügung gestellt werden)

Netzarchitektur

Die Bussysteme von Rockwell fügen sich über CIP (Control and Information Protocol) harmonisch in das offizielle ISO-OSI-Protokoll ein (Abb. 4.64).

Netztopologie

EtherNet/IP-Netzwerke nutzen in der Regel eine aktive Sterntopologie, in der Geräte über eine Punkt-zu-Punkt-Verbindung an einen Switch angeschlossen sind. Der Vorteil einer Sterntopologie liegt darin, dass sie Produkte mit einer Übertragungsrate von 10 Mbit/s wie auch von 100 Mbit/s unterstützt. Ebenso kann man Produkte beider Übertragungsraten (10 Mbit/s und 100 Mbit/s) miteinander kombinieren, da die meisten Ethernet-Switches die Übertragungsgeschwindigkeit automatisch aushandeln. Die Vorteile der Sterntopologie sind z. B. einfach einzurichtende Verbindungen, leichte Fehlersuche und -behebung sowie eine einfache Wartung.

EtherNet/IP kann große Mengen von Nachrichtendaten verarbeiten. Doch nicht nur die Verarbeitung großer Datenmengen, sondern auch die Übertragungsraten von EtherNet/IP (10/100 Mbit/s) machen diese Art der Datenübertragung noch interessanter. Die breite Akzeptanz, die die Ethernet-Technologie über die Jahre gefunden hat, schlägt sich in den rapide sinkenden Kosten nieder, die für physikalische Ethernet-Medien anfallen. Aufgrund dieser Eigenschaften ist EtherNet/IP für viele Steuerungsanwendungen eine gute Wahl.

Die EtherNet/IP-Kabelkomponenten bieten Flexibilität hinsichtlich Kosten und Lieferanten. Aufgrund der großen Zahl von Drittanbietern steht eine breite Palette an Medienkomponenten zu unterschiedlichen Preisen zur Auswahl. So können die Anwender beim

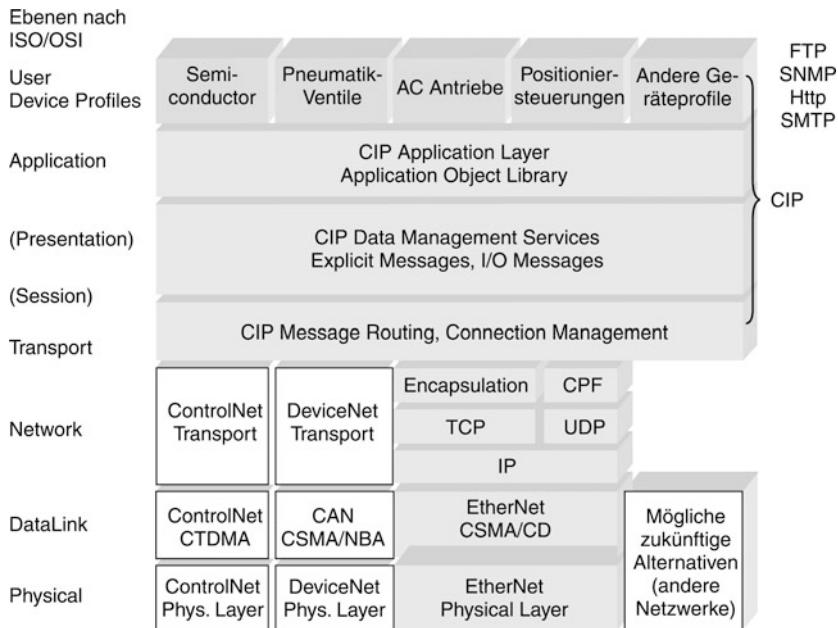


Abb. 4.64 Der Protokoll-Stack für EtherNet/IP

Aufbau ihres Netzwerks unter einer Vielzahl von Komponenten wählen: Kabel, Transceiver, Hubs, Repeater, Router und Switches.

Standardmäßige Twisted-Pair- und Glasfaserkabel sind mit EtherNet/IP voll funktionsfähig. Je nach Umgebung sollten die Anwender Produkte in Betracht ziehen, die sich für den Einsatz in Industrieumgebungen bewährt haben. Je nach Netzwerkkonfiguration eignet sich ein Ethernet-Hub oder ein Switch. Mit einem **Hub** steht eine kostengünstige Lösung für den Anschluss an Informationsnetzwerke (Shared Ethernet) zur Verfügung. Ein **Switch** reduziert die Zahl der möglichen Kollisionen und empfiehlt sich daher für Echtzeit-Steuerungsanwendungen (Switched Ethernet). **Router** kommen zum Einsatz, um im Netzverkehr Steuerungsdaten von anderen Datenarten wie z. B. Bürodaten zu trennen, um im Fertigungsbereich Informationsdaten von Steuerungsdaten zu trennen oder aus Gründen der Sicherheit (d. h. für den Aufbau von Firewalls). **Repeater** dienen dazu, die maximale Ausdehnung des Netzwerks noch zu erweitern. Zudem sind sie in der Lage, unterschiedliche physikalische Medien in einem Netzwerk miteinander zu verbinden.

EtherNet/IP-Übertragungsarten

Das auf TCP und UDP aufgesetzte EtherNet/IP-Kommunikationsprotokoll heißt „Control and Information Protocol“ (CIP) und wurde 1999 eingeführt, um Interoperabilität zu gewährleisten. Der Steuerungsteil von CIP wird für zyklische Echtzeit-E/A-Nachrichtenübertragung (Implicit Messaging) verwendet. Der Informationsteil von CIP

dient der expliziten Nachrichtenübertragung (Explicit Messaging). Diese Definitionen veranschaulichen die verschiedenen Übertragungsarten, die in der nachstehenden Übersicht aufgeführt sind.

- Information. Nicht zeitkritische Datenübertragungen – typischerweise große Datenpakete. Bei der Übertragung von Informationsdaten handelt es sich um kurzlebige explizite Verbindungen zwischen einem Sender und einem einzelnen Zielgerät (Schreiben und Lesen). Informationsdatenpakete verwenden das TCP/IP-Protokoll und nutzen die TCP-Funktionen zur Datenverarbeitung.
- E/A-Daten. Zeitkritische Datenübertragungen – typischerweise kleinere Datenpakete. Bei der Echtzeit-Übertragung von E/A-Daten handelt es sich um langfristige implizite Verbindungen zwischen einem Sender und einer beliebigen Zahl von Zielgeräten. E/A-Datenpakete verwenden UDP/IP-Protokolle und nutzen den extrem schnellen Datendurchsatz, den UDP bietet.
- Real-time Interlocking (Echtzeit-Verriegelung). Zyklische Datensynchronisation zwischen einem Producer-Knoten und einer beliebigen Zahl von Consumern (implizit). Interlock-Datenpakete verwenden die schnelleren UDP/IP-Protokolle und nutzen den hohen Datendurchsatz, den UDP bietet.

HTTP-Funktion

Die meisten EtherNet/IP-Module verfügen über eine integrierte Web-Server-Funktion. Modul, Netzwerk und Systemdateninformationen sind über jeden standardmäßigen Web-Browser (z. B. Internet Explorer oder Netscape) zugänglich.

EtherNet/IP-Produkte bieten verschiedene Funktionen wie:

- Lesen/Schreiben von Daten
- Diagnose
- Senden von E-Mails
- Bearbeiten von Konfigurationsdaten

Ethernet-E/A

Hierunter versteht man zum einen Produkte mit Ethernet-Ports für Scan-Funktionen (Master) und zum anderen Produkte, bei denen die Ethernet-Ports die Funktion eines Adapters übernehmen, um E/A-Geräte über ein EtherNet/IP-Netzwerk zu steuern. Eines der ersten EtherNet/IP-Produkte für die E/A-Kommunikation ist „ControlLogix“ – einer SPS eines bekannten Herstellers, die mit 10 Mbit und 100 Mbit über EtherNet/IP kommuniziert. Sie kann als lokale Brücke zu einem anderen Feldbus eingesetzt, an einer anderen ControlLogix für eine Peer-to-Peer-Verbindung oder an einen dezentralen E/A-Adapter angeschlossen werden. Das Modul unterstützt die Übertragung von E/A-Steuerungsdaten, Processor-Interlocking und das nahtlose Routing von Nachrichten zwischen EtherNet/IP, ControlNet und DeviceNet-Netzwerken. Mit der hohen Datenübertragungsrate und der Unterstützung für weit verbreitete E/A-Modulreihen stellt das EtherNet/IP-Netzwerk eine einfache und flexible Methode zur Übertragung von E/A-Daten dar.

4.3.6 Echtzeit-Ethernet: Powerlink

Grundsätzlich ist Ethernet nicht deterministisch, d. h. nicht echtzeitfähig. Wenn man bedenkt, dass echtzeitfähig eigentlich nur heißt, dass der Bus schneller seine muss als der von ihm versorgte Prozess, dann ergeben sich daraus drei Bereiche:

- Ethernet ist an sich schneller als der Prozess, es ergeben sich keine Zeitprobleme,
- durch geswitchte Netzwerk-Bereiche reduziert man die Kollisionsdomänen, sodass die Kollisionswahrscheinlichkeit klein genug wird (weiche Echtzeit),
- durch Eingriff in die Bus-Steuerung vermeidet man Kollisionen vollständig (harte Echtzeit).

Harte Echtzeit ist z. B. vonnöten, wenn es um die elektronische Synchronisation von Antriebssträngen (Druckmaschinen, Verpackungsmaschinen) geht.

Um ein Ethernet-Netzwerk deterministisch und damit echtzeitfähig zu machen, müssen entsprechende Maßnahmen vollzogen werden, die Kollisionen vollständig verhindern. Bei Ethernet ist das mit Hilfe von „organisatorischen“ Eingriffen möglich, ohne Ethernet an sich zu verändern:

- **Segmentierung eines Netzwerks mit Switches:** Jede Station hängt über einen Switch am Netzwerk. Dadurch entstehen lauter bidirektionale Quasi-Punkt-zu-Punkt Verbindungen, so genannte „Collision Domains“ zwischen genau zwei Stationen. Kollisionen sind damit ausgeschlossen. Switches besitzen teilweise die Features der Ethernet-Erweiterung IEEE 802.1Q und 802.1p. Damit können Datenpakete priorisiert und im Switch entsprechend schnell bzw. über Zwischenspeicherung entsprechend verzögert weitergeleitet werden. Mit diesem Zusatz ist eine Performance-Steigerung möglich. Nachteil des gesamten Verfahrens ist der Aufwand für die Switches. Entweder jedes Gerät wird an einem Switch angeschlossen oder besitzt einen eigenen internen Switch. Alle Switches müssen in Bezug auf das gesamte betroffene Netzwerk optimal parametriert werden, damit die bestmögliche Gesamtperformance erzielt werden kann. Es entstehen einerseits organisatorischer Aufwand und andererseits zusätzliche Hardwarekosten. Daneben ist zu berücksichtigen, dass die eingebaute „Intelligenz“ eines Switches entsprechende Verzögerungen verursacht. Da alle eingehenden Datenpakete analysiert und dann nur gezielt weiter versendet werden, ergeben sich hier wesentlich größere Latenzzeiten als bei reinen Hubs, die außerdem Schwankungen unterworfen sind, d. h. Jitter verursachen.
- **Definierte Organisation der Kommunikation:** Um die vorhandene Bandbreite von Ethernet optimal auszunützen zu können, reicht die Einrichtung von durchgängig „geswitchten“ Netzwerken nicht aus. Es muss vielmehr die gesamte Kommunikation organisiert werden. Wenn jeder Netzwerkeinnehmer nur nach fest gelegten Regeln, d. h. zu fix definierten Zeiten, senden darf, können auch ohne „Collision Domains“ Kollisionen gänzlich vermieden werden. Maßnahmen wie Priorisierung etc. sind nicht

notwendig. Die Netzwerkauslastung kann, zumindest theoretisch, gegen 100 % gehen. Mögliche Organisationsformen sind z. B. dem Ethernet überlagerte Zeitscheiben-(„Time Slot“)-Verfahren.

Kommunikationszeiten

Lässt sich die Kollisions-Problematik von Ethernet mit Hilfe der oben beschriebenen Maßnahmen in den Griff bekommen, so bleiben immer noch die relativ langen Abarbeitungszeiten der Standard-Protokolle TCP/IP und UDP/IP. Verschiedene Untersuchungen unterschiedlicher Institute und Firmen haben deutlich gemacht, dass bei ganzheitlicher Betrachtung der Kommunikationszeiten die reine Übertragung am geringsten in das Gewicht fällt. Die Übertragungszeiten für ein Ethernet-Minimaltelegramm mit 46 Byte Nutzdaten und einem Overhead von 38 Byte beträgt in einem 100 Mbps Netzwerk 6,7 µs. Ein maximales Ethernet-Paket mit 1538 Byte Gesamtlänge benötigt 123 µs.

Vergleicht man hierzu die Stack-Abarbeitungszeiten für die Übertragung mit einem UDP/IP Stack, die mit einem Pentium 166 bei ca. 400 bis 500 µs liegen, so bewegt man sich hier in anderen Dimensionen. Ein Pentium 166 erscheint im Vergleich mit aktuellen Gigaherz-PCs als nicht mehr zeitgemäße Hardware. Wenn man aber bedenkt, dass entsprechende Rechenleistung in jeder SPS, I/O oder Antrieb zur Verfügung stehen muss und die damit verbundenen Kosten direkt einfließen, ist ein Pentium 166 dann doch relativ groß.

Zusätzlich zur Stack-Abarbeitung fällt das Potential von Antwort-Latenzzeiten auf Datenanforderungen gegenüber den reinen Übertragungszeiten massiv in das Gewicht. Diese Latenzzeiten werden direkt von den verwendeten Stacks bestimmt. Eine Erhöhung der Übertragungsgeschwindigkeit bedeutet daher nur ein noch schlechteres Ausnutzen der Bandbreite, ohne dass die gesamte Übertragung wesentlich schneller wird.

Ausgehend von diesen Überlegungen ist es einerseits notwendig, die zur Verfügung stehende Bandbreite mit geeigneten Verfahren optimal zu nutzen und andererseits, die Latenzzeit bzw. Stack-Abarbeitungszeit so gering wie möglich zu halten. Genau diesen Ansatz verfolgt ETHERNET Powerlink von B&R mit einem optimierten Echtzeit-Stack für die Übertragung deterministischer Daten, ohne auf die Möglichkeiten mit dem Standard-Stack TCP/IP verzichten zu müssen.

Funktionsweise ETHERNET Powerlink

Anforderungsprofil

Basierend auf den Untersuchungen und den daraus gewonnenen Erkenntnissen bezüglich des Echtzeitverhaltens von Ethernet ergaben sich folgende Anforderungen:

- Organisation des Netzwerkes, um Kollisionen zu verhindern.
- Entwicklung eines optimierten und schnellen Echtzeit-Stacks

Ziel war es ein möglichst jitterfreies, deterministisches Netzwerk auf Basis von Standard Fast Ethernet zu schaffen, das den harten Echtzeitbedingungen für die Vernetzung von SPS, I/O und Antrieben entspricht die für „High-Performance“-Anwendungen benötigt werden. Die Kenngrößen sollten dabei ein maximaler Netzwerkjitter von unter $1 \mu\text{s}$ und Netzwerkzykluszeiten unter $500 \mu\text{s}$ für 10 Netzwerkstationen sein. Daneben muss nicht-deterministischer Datenaustausch, z. B. mit dem TCP/IP-Protokoll möglich sein, ohne den Echtzeitdatenverkehr zu beeinflussen.

Dass diese Anforderungen in einem völlig offenen Ethernet-Netzwerk mit allen möglichen Teilnehmern nicht umzusetzen sind, zeigten die vorgenommenen Recherchen. Beträgt man aber das eigentliche Einsatzgebiet, die Automatisierung von Maschinen und Anlagen, so kann man hier von abgeschlossenen Einheiten sprechen, die sich in der Netzwerkstruktur widerspiegeln dürfen. In sich abgeschlossene lokale Netzwerke, optimiert und begrenzt für ihre Anwendung sind kein Widerspruch bezüglich der Anforderung an Offenheit. Voraussetzung dafür ist der nahtlose Übergang vom Echtzeitnetzwerk auf ein „öffentlichtes“ Ethernet mit Hilfe von entsprechenden Echtzeit-Switches, die den normalen TCP/IP-Verkehr regeln. Ebenso wichtig war die Anforderung, ein und dasselbe Gerät in beiden Welten einsetzen zu können.

Die Forderung nach einfacher Installation und Wartung im Feld entschied die Organisationsform des Datenverkehrs. Prinzipiell gibt es zwei unterschiedliche Methoden ein Zeitscheibenverfahren auf Ethernet umzusetzen:

- **Dezentrale Organisation:** Alle Netzwerkteilnehmer besitzen eine über das gesamte Echtzeit-Netzwerk exakt synchronisierte Uhr. Dazu gibt es verschiedene Verfahren, die eine entsprechende Synchronisierung erlauben, sodass mit entsprechender Hardwareunterstützung eine Synchronisierung im Bereich von unter $1 \mu\text{s}$ zu erzielen ist (z. B. IEEE 1588). Damit lassen sich Aktionen der Teilnehmer über das Netzwerk exakt abstimmen. Es können z. B. Kommandos an Antriebe geschickt werden, die dann zu einem definierten Zeitpunkt quasi gleichzeitig auf allen Antrieben ausgeführt werden können. Dieses Verfahren alleine bringt zwar einen minimalen Jitter, eine geringe Zykluszeit wird damit aber noch nicht erreicht. Für viele Applikationen, gerade im Antriebsbereich, ist das aber ein wesentlicher Faktor. Deshalb müssen alle Teilnehmer zusätzlich einen „Kommunikationsplan“ besitzen, der genau definiert, wann welcher Netzwerkteilnehmer mit welcher Framelänge kommunizieren darf. Dieser Plan muss entsprechend erstellt und bei jeder Änderung neu aktualisiert und allen Teilnehmern neu übertragen werden. Ein entsprechender Verwaltungsaufwand ist damit im Feld unumgänglich, sowohl bei der Inbetriebnahme, bei Änderungen oder im Servicefall.
- **Zentrale Organisation:** Die einfachste und für das Einsatzprofil beste Möglichkeit der Datenverkehrsorganisation besteht in der Kommunikationssteuerung mittels einer zentralen Intelligenz. Im Prinzip gibt es dann in einem Echtzeit-Ethernet einen Taktgeber, der die gesamte Zeitscheibenverwaltung übernimmt. Verwaltung bzw. Parametrierung reduziert sich auf einen Teilnehmer. Änderungen, Kommunikationsfehler, Stationsausfälle oder zusätzliche neue Stationen können mit minimalem Aufwand beherrscht wer-

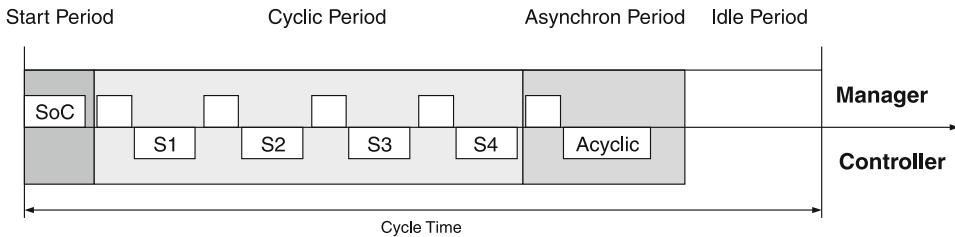


Abb. 4.65 Die verschiedenen Perioden eines Powerlink-Zyklus

den. Synchronisierungsmechanismen entfallen bzw. sind automatisch Bestandteil der Kommunikationssteuerung, wenn diese geschickt aufgebaut wird. Zusätzlicher Hardwareaufwand ist nicht erforderlich um Netzwerkjitter kleiner 1 µs zu realisieren.

Als Schlussfolgerung aus allen diesen Gründen wird bei ETHERNET Powerlink die zentrale Organisation mit einem optimierten Echtzeitstack eingesetzt.

Funktionsprinzip

ETHERNET Powerlink ist ein streng deterministisches, isochrones Echtzeitprotokoll, das Standard Fast Ethernet als Medium verwendet, konform zu IEEE 802.3u. Die prinzipielle Funktionsweise basiert auf einem isochronen Time-Slot-Verfahren, dem „Slot Communication Network Management“ (SCNM). TCP/IP wird dabei durch den echtzeitfähigen Powerlink Stack ersetzt.

Im Netzwerk gibt es dabei genau einen Manager, der die Kommunikationssteuerung übernimmt. Alle anderen Stationen haben reine Controller Funktionalität, d. h. senden nur, wenn sie vom Manager die entsprechende Berechtigung erhalten. Die vom Controller gesendeten Daten sind Broadcasts und können damit von allen anderen Stationen empfangen werden. Der Datenverkehr im Netzwerk läuft in deterministischen, isochronen Zyklen ab. Die Zykluszeit ist im Manager konfigurierbar. Innerhalb eines Powerlink-Zyklusses werden vier Zeitbereiche unterschieden (Abb. 4.65):

- **Start Period:** Der Manager sendet einen Start of Cyclic Frame (SoC). Auf diesen Frame erfolgt die Synchronisierung aller Stationen im Powerlink-Netzwerk.
- **Cyclic Period:** Abarbeiten des zyklischen Datenverkehrs aller aktiven Stationen
- **Asynchron Period:** Kommunikations-Slot für azyklische, nicht deterministischen Datenaustausch, z. B. über TCP/IP
- **Idle Period:** „Restzeit“ vom Ende der Kommunikation bis zum Beginn des neuen Powerlink-Zyklus

Wesentlich für die Qualität des Netzwerks, bzw. für die Minimierung des Netzwerkjitters, ist der periodisch exakte Zeitpunkt des Sendens des Start of Cyclic Frames. Danach folgt der zyklische Datenverkehr der aktiven Stationen. Der Manager teilt dazu jeder

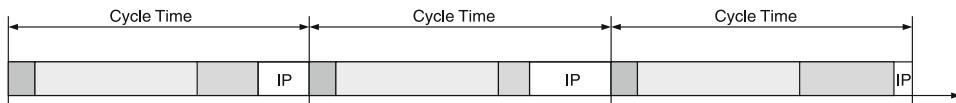
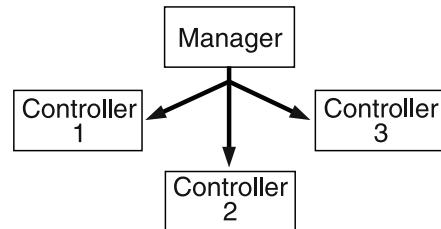


Abb. 4.66 Die Restzeit (IP, idle period) eines Zyklus ist variabel

Abb. 4.67 Broadcast beim Start der zyklischen Übertragung



Station sequenziell einen Kommunikations-Slot zu. Abschließend ist in jedem Zyklus fix ein Zeit-Slot reserviert für azyklische, d. h. asynchrone Kommunikation. Senderecht bekommen Stationen auf vorherige Anforderung. Bis zum Beginn des nächsten Zyklus verbleibt entsprechend Restzeit, die bei optimaler Bandbreitenausnutzung gegen Null gehen kann. Diese Idle-Time kann durch unterschiedlich lange azyklische Kommunikation auch schwanken (Abb. 4.66). Danach beginnt der neue Powerlinkzyklus.

Hier noch einmal im Detail das Prinzip der ETHERNET Powerlink Kommunikation in einer anderen Darstellung (Abb. 4.67, 4.68, 4.69 und 4.70):

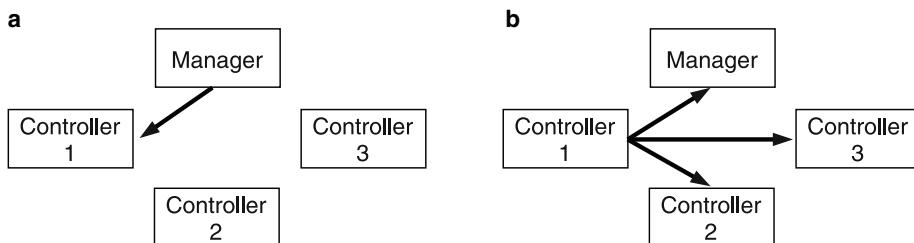


Abb. 4.68 Station 1 erhält Kommunikationsrecht (a) und sendet (b)

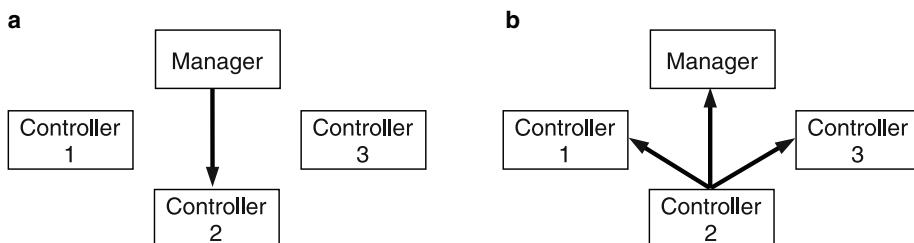


Abb. 4.69 Station 2 erhält Kommunikationsrecht (a) und sendet (b)

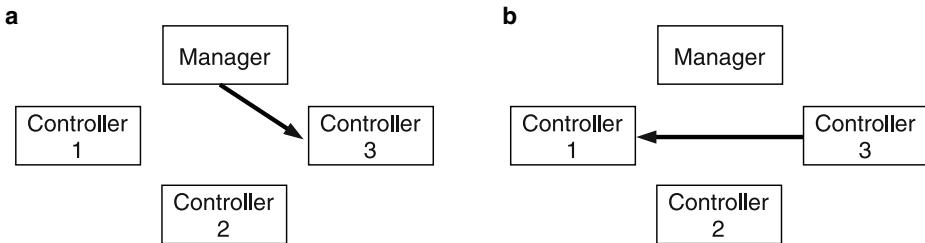


Abb. 4.70 Station 3 erhält die Erlaubnis (a) zum Senden azyklischer Daten (b)

Der Manager startet die Kommunikation mit dem Start of Cyclic Frame (SoC) an alle Stationen (Controller) mit einem Broadcast. Dieser Frame wird exakt zeitgesteuert und bildet die Grundlage der gemeinsamen Zeitbasis aller Stationen.

Nach dem Senden des SoC beginnt der zyklische Datenaustausch. Der Manager gibt der ersten konfigurierten und aktiven Station das Kommunikationsrecht. Die Station antwortet mit einem Broadcast und kann damit ihre Daten allen Stationen im Netzwerk, ohne Umweg über den Manager, direkt senden.

Durch die Nutzung des Broadcasts im Ethernet ist Powerlink nicht auf einen reinen Master-Slave Datenverkehr festgelegt. Vielmehr gibt es damit die Möglichkeit des beliebigen Querverkehrs und der Manager fungiert im Extremfall nur noch als zentraler Taktgeber.

Diese oben beschriebene Prozedur wiederholt sich für alle Stationen im zyklischen Datenverkehr.

Nach Beendigung der zyklischen Kommunikation ist in jedem Zyklus eine Zeitscheibe für asynchronen Datenaustausch vorgesehen. Eine Station, die hier aktiv werden möchte, muss im Antwort-Frame an den Manager im zyklischen Teil das Senderecht für asynchrone Daten anmelden. Entsprechend der Sendeanforderungs-Queue bekommt immer nur eine Station vom Manager die Aufforderung zum Senden ihrer azyklischen Daten. Der asynchrone Frame kann dann an jede beliebige Station gehen.

Durch die endliche Zykluszeit und wegen der nicht unendlich kleinen Übertragungs- bzw. Latenzzeiten ist die Anzahl der Stationen unmittelbar von der eingestellten Zykluszeit abhängig. Es ist aber nicht immer für alle Stationen unbedingt notwendig, in jedem Zyklus Daten zu senden. Oft genügt es, dass immer Daten empfangen werden können. Klar wird das am Beispiel Antriebe: Hier gibt es oft Masterachsen, die in jedem Zyklus Daten brauchen und auch entsprechende Führungsgrößen an alle ihre Slave-Achsen weitergeben müssen. Diese Master-Achsen müssen in jedem Powerlink-Zyklus volles Kommunikationsrecht bekommen. Die Slave-Achsen hingegen müssen in jedem Zyklus nur Daten empfangen können. Es reicht bei ihnen vollkommen aus, wenn sie in einem langsameren, übergeordneten Zyklus ihre Statusmeldungen absetzen können.

Durch dieses Verhalten ist es möglich, eine zweite Klasse an Powerlink-Netzwerk-Teilnehmern einzuführen:

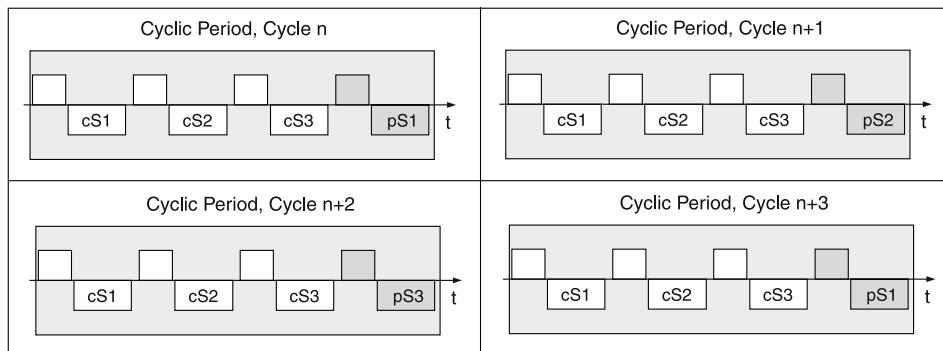


Abb. 4.71 Jeweils eine Prescaled-Station pS_x hat Kommunikationsrecht

- Klasse 1 – cyclic: Stationen haben in jedem Powerlink-Zyklus volles Kommunikationsrecht
- Klasse 2 – prescaled: Stationen haben nur jeden n -ten Powerlink-Zyklus volles Kommunikationsrecht, können aber in jedem Powerlink-Zyklus Daten empfangen (Abb. 4.71).

Die Anzahl der Klasse-2-Stationen pro Zyklus ist parametrierbar und wird durch die Anzahl der Klasse-1-Stationen und der eingestellten Zykluszeit bestimmt. Entsprechend ergibt sich daraus der „Prescale-Zyklus“.

In Abb. 4.71 sind drei Klasse-1-Stationen (cS_x) dargestellt und drei Klasse-2-Stationen (pS_x), von denen jeweils eine pro Zyklus parametriert ist.

Mit diesem Verfahren können sehr viel mehr Stationen am Datenverkehr teilnehmen, als normalerweise in einen Zyklus möglich wären.

Performancedaten von ETHERNET Powerlink

Die folgenden Beispiele wurden ausschließlich mit Klasse-1-Stationen, 80 Byte Daten in Sende- und Empfangsrichtung und jeweils 50 m Kabel zwischen den Stationen gerechnet.

- $200 \mu s$ Netzwerkzyklus: 2 Stationen
- $500 \mu s$ Netzwerkzyklus: 12 Stationen
- $1,0 \text{ ms}$ Netzwerkzyklus: 30 Stationen
- $2,0 \text{ ms}$ Netzwerkzyklus: 66 Stationen
- $3,0 \text{ ms}$ Netzwerkzyklus: 102 Stationen

Bei einem praktischen Aufbau mit 50 I/O-Stationen mit jeweils < 46 Byte Nutzdaten und 50 Antrieben mit jeweils ca. 80 Byte Nutzdaten, immer in beide Richtungen, ergab sich eine reale Zykluszeit von 2,4 ms.

Der Netzwerkjitter ist unabhängig von der Stationsanzahl immer < $1 \mu s$.

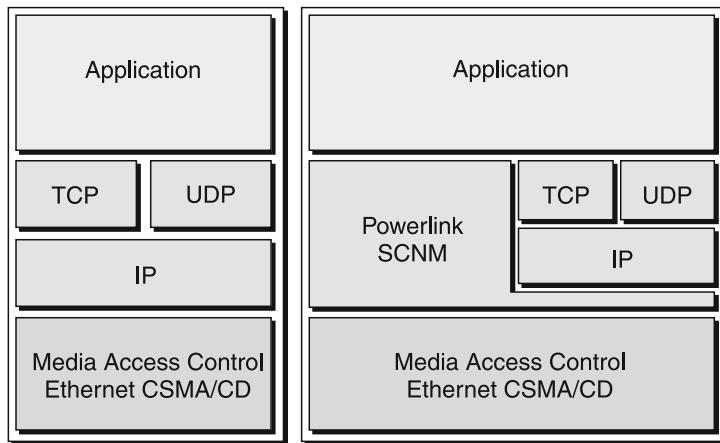


Abb. 4.72 Für normales Ethernet ist Powerlink-SCNM (slot communication network management) transparent

TCP/IP über ETHERNET Powerlink

Die Tatsache, dass ETHERNET Powerlink auf Standard Fast Ethernet aufsetzt, prädestiniert es geradezu für TCP/IP-Kommunikation (Abb. 4.72). Das ist auch eine der Hauptanforderungen der Automatisierungsbranche, wenn es um den Einsatz von Ethernet im Feld geht. Das Problem ist dabei, den Spagat zwischen harten Echtzeitanforderungen einerseits und prinzipiell nicht deterministischer TCP/IP-Kommunikation andererseits zu schaffen. ETHERNET Powerlink bietet diese Möglichkeit. Mit dem beschriebenen Zeitscheibenverfahren werden Zykluszeiten von bis zu $200\ \mu s$ erreicht, bei einem Netzwerkjitter von unter $1\ \mu s$. Trotzdem steht immer eine fest reservierte Bandbreite für spontane Kommunikation zur Verfügung, wie z. B. für TCP/IP.

Diese Möglichkeit kann man verschieden nutzen:

- Zyklische und azyklische Powerlink-Kommunikation
- Senden und Empfangen eines (beliebigen) Ethernet Frames. Die Powerlink-Anschaltung verhält sich wie eine Ethernet-Karte.

Für einen darüber liegenden TCP/IP-Stack ist dabei transparent, ob die darunter liegende Hardware eine Ethernet-Netzwerkkarte oder eine Powerlink-Anschaltung ist.

Einem IP-Stack präsentiert sich die Powerlink-Anschaltung als Ethernet-Karte. Bei deterministischer Kommunikation setzt die Applikation direkt auf Powerlink auf. Ebenso kann das gleiche Gerät in einer Nicht-Powerlink Umgebung eingesetzt werden – in einem Standard Ethernet-Netzwerk. Der Powerlink Stack verhält sich dann quasi transparent. Damit ergibt sich ein Anschluss für normales und echtzeitfähiges Ethernet.

Für die Anbindung eines nicht Powerlink fähigen Ethernet-Geräts ist die Anschaltung über einen speziellen Powerlink-Koppler notwendig. Das ist im Prinzip ein spezieller

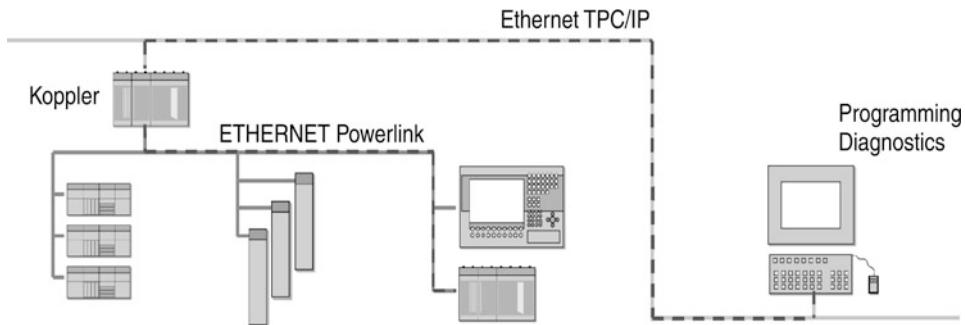


Abb. 4.73 Verbindung Ethernet/Powerlink über speziellen Koppler

Switch, der ankommende TCP/IP Frames aus der Nicht-Echtzeit-Welt zwischenspeichert, bis er vom Powerlink Manager das Sende-Recht im azyklischen Slot bekommt. Damit ist die Verbindung zwischen den geschützten Echtzeit-Netzbereichen und der kollisionsbehafteten normalen Ethernet-Welt vollständig durchgängig möglich.

Topologie

Dadurch, dass ETHERNET Powerlink direkt auf Fast Ethernet aufsetzt, gelten auch grundsätzlich die gleichen Topologie-Daten. Das bedeutet bis zu 100 m Segmentlänge und die Verwendung von Kat. 5 Patchkabel. So ist z. B. auch der Einsatz von Lichtleitern zur Datenübertragung möglich.

Der Topologieaufbau wird mit Hilfe von Hubs (Koppler) realisiert (Abb. 4.73). Durch die vollständige Kollisionsfreiheit in einem ETHERNET Powerlink Netzwerk ist hier auch keine Kollisionserkennung notwendig. Damit fällt die Topologiebeschränkung in der Hub-Kaskadierung weg. In einem Powerlink Netzwerk können bis zu 10 Hubs in Reihe geschalten werden, sehr zum Vorteil eines feldgerechten Topologieaufbaus, besonders wenn ein Hub integraler Bestandteil eines Gerätes ist.

Einsatz von ETHERNET Powerlink

Dass Ethernet für den Einsatz unter harten Echtzeitbedingungen in der Automatisierung tauglich gemacht werden kann, beweisen inzwischen etliche Powerlink-Applikationen. Angefangen von Spritzgussmaschinen mit 3 Achsen in 400 µs Zykluszeit über Verpackungsmaschinen mit 19 Achsen in 800 µs bis zu Großanlagen, bei denen 50 Achsen plus 50 I/O-Stationen in 2,4 ms und auf einem Powerlink-Netzwerk laufen, zeigen eindrucksvoll die Möglichkeiten, die ETHERNET Powerlink heute schon bietet.

Durch die Offenlegung des Protokolls wird diese Technologie auf breiter Ebene zugänglich und einsetzbar. Damit soll ein Standard für echtzeitfähiges Ethernet geschaffen werden, der einerseits in definierten Netzwerkbereichen einen deterministischen Datenaustausch ermöglicht und gleichzeitig die Anforderung nach durchgängiger Kommunikation mit TCP/IP erfüllt.

4.3.7 Modbus-TCP

Der Modbus-TCP ist der dienstälteste und übersichtlichste (und vermutlich verbreitetste) Vertreter der Industrial-Ethernet-Busse. Er arbeitet nicht mit dem CSMA/CD-Zugriffsverfahren, sondern nach dem Master/Slave-Prinzip, hier Client/Server genannt. Er ist Teil der Modbus-Familie, wie Abb. 4.74 zeigt.

Der Modbus-TCP kann als schneller Bus zur Vernetzung von Automatisierungsgeräten dienen und/oder als übergeordneter Backbone-Bus für prozessnahe Busse, wie z. B. den Modbus-RTU (Abb. 4.75). Die Telegramme der einzelnen Ebenen sind strukturell gleich, wie Abb. 4.76 zeigt. Man beachte, dass die PDU des Modbus-TCP keinen eigenen Fehlererkennungscode transportiert. Die Fehlererkennung erfolgt in der unterlagerten TCP-Schicht des Ethernet-Protokolls.

Als Lotse im Ethernet-Bereich dient der MBAP-Header, der in Abb. 4.76 eingetragen und in Abb. 4.77 detailliert zu sehen ist. Seine 7 Byte sind alles, was die Telegrammefizienz des Modbus-TCP belastet. Alle PDUs des Modbus-TCP werden von den Servern an die feste Port-Adresse 502 des Clients gesendet.

Ein Grund für die Verwendung des Ethernet in der Automatisierungstechnik ist dessen hohe Datenübertragungsrate. Beim Modbus-TCP ist die Bitübertragungsrate 100 MBd. Nehmen wir einen zu übertragenden Prozesswert mit 16 bit an und eine Telegrammfizienz von 60 %, so erhält man eine theoretische Datenübertragungsleistung von $3,75 \cdot 10^6$ Werten/s. Es wird in www.anybus.de von einem Praxistest berichtet, bei dem als Client eine Modicon Momentum SPS über Modbus-TCP 4000 E/A-Feldgeräte pro Sekunde zu bedienen in der Lage war. Jedes Feldgerät hatte 32 digitale E/A und 16 analoge E/A. Es fallen damit $4000 \cdot 2 \cdot (2 + 16) = 144\,000$ Werte pro Sekunde an. Damit beträgt die tatsächliche Datentransportleistung knapp 4 % der theoretisch möglichen. Der Engpass, wenn es denn tatsächlich einer ist, liegt an den Kommunikationsschnittstellen (SPS, Feld-

Schicht	ISO/OSI	MB-PLUS	MB-RTU	MB-ASCII	MB-TCP
7	Application	MODBUS Application Layer			
6	Presentation	–	–	–	–
5	Session	–	–	–	–
4	Transport	–	–	–	TCP
3	Network	–	–	–	IP
2	Data Link	Token Passing	Master/Slave		Client/Server
1	Physical	RS 485	RS232/RS485		Ethernet 100MBd

Abb. 4.74 Die verschiedenen Varianten des Modbus im ISO/OSI-Modell. RTU – Remote Terminal Unit, Client/Server – Master/Slave

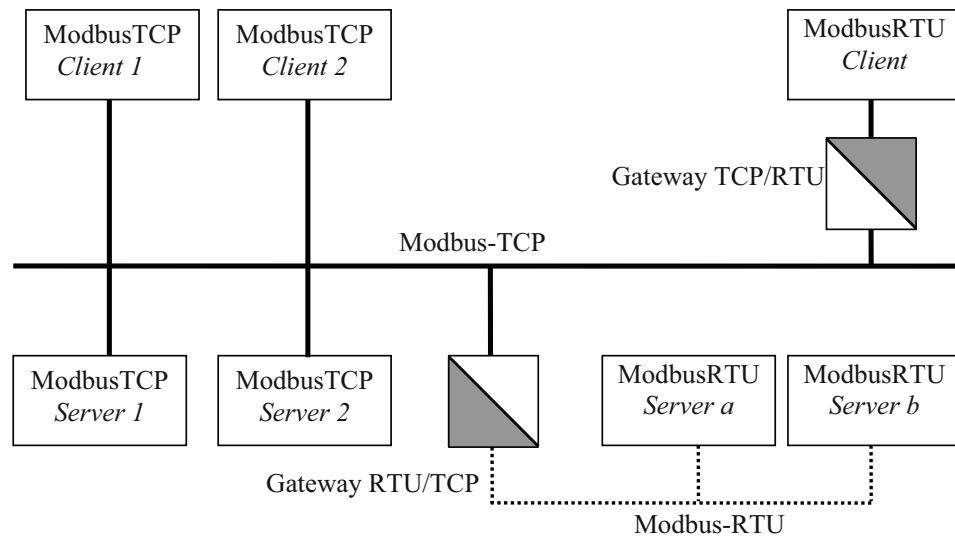


Abb. 4.75 Beispiel einer Vernetzung von Modbus-TCP mit Modbus-RTU. Die physikalische Ebene OSI-1 ist dieselbe

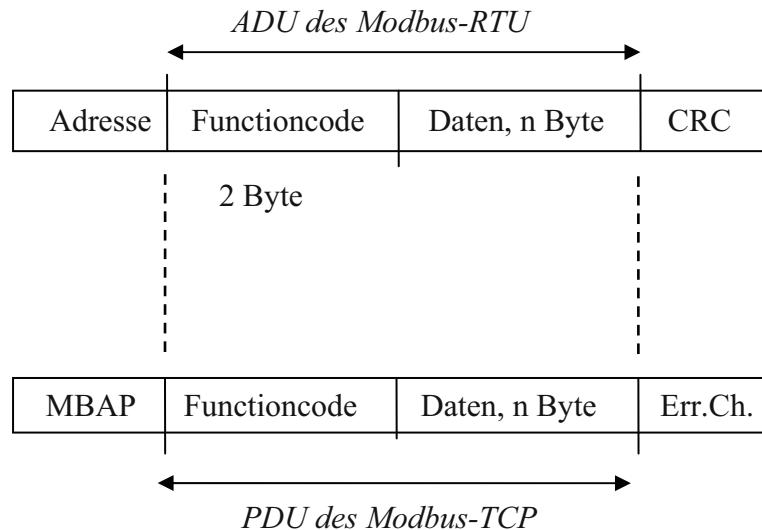


Abb. 4.76 Die ADU des Modbus-RTU ist als PDU in das Telegramm des Modbus-TCP eingebettet.
ADU – Application Data Unit, PDU – Process Data Unit, MBAP – Modbus Application Protocol
Err. Ch. – Error Check, Fehlerprüfung in OSI-Schicht 2

MPAP-Header			
Transaction Identifier	Protokoll Identifier	Länge	Unit Identifier
2 Byte	2 Byte	2 Byte	1 Byte
Request/Response	0 = Modbus-Protokoll	Adresse eines folgenden Byte	Anzahl der RTU-Slaves

Abb. 4.77 Der Header des Ethernet-Telegramms des Modbus-TCP

geräte, Gateways) und nicht am Bus selbst. Dabei haben wir die Dauer der eigentlichen Prozessdatenverarbeitung noch nicht berücksichtigt. Das ist bei praktisch allen Bussystemen so.

4.3.8 Echtzeit Ethernet EtherCAT

Mit der EtherCAT-Technologie werden die prinzipiellen Begrenzungen vieler anderer Ethernet-Lösungen überwunden: Das Ethernet-Paket wird nicht mehr in jeder Anschaltung zunächst empfangen, dann interpretiert und die Prozessdaten werden weiterkopiert. Die EtherCAT-Slave-Geräte entnehmen nur die für sie bestimmten Daten, während das Telegramm das Gerät durchläuft. Ebenso werden Eingangsdaten im Durchlauf in das Telegramm eingefügt (siehe Abb. 4.78). Die Telegramme werden dabei nur wenige Nanosekunden verzögert.

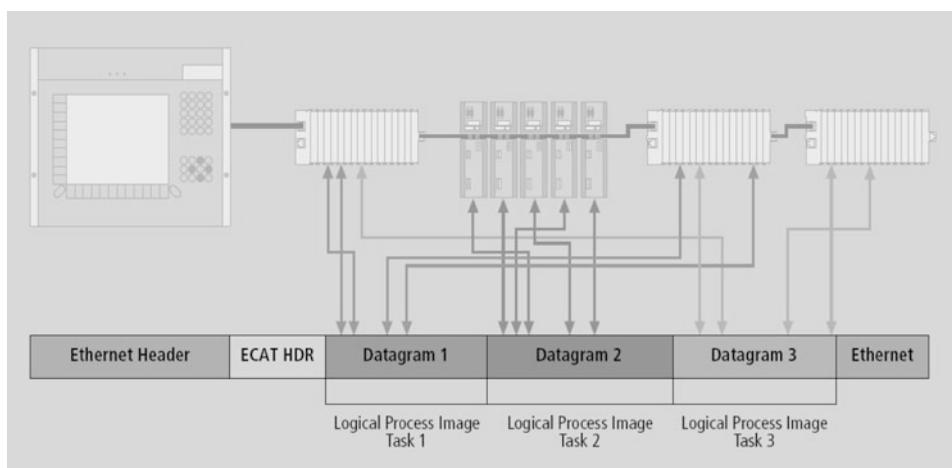


Abb. 4.78 Prozessdaten werden in das Telegramm eingefügt

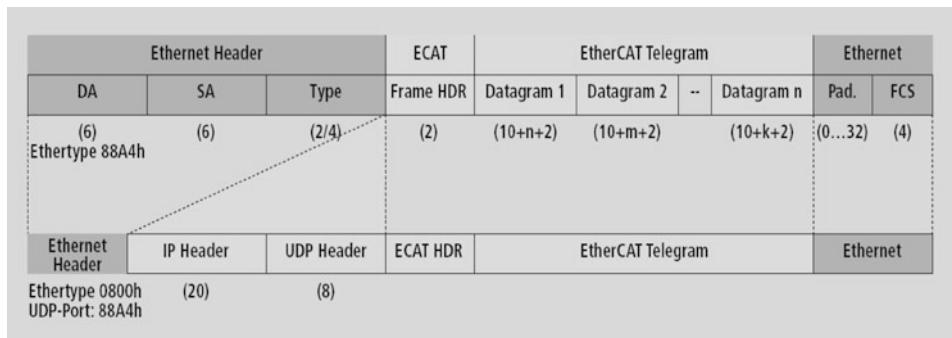


Abb. 4.79 EtherCAT: Standard Frames nach IEEE 802.3

Da ein Ethernet Frame sowohl in Sende- als auch in Empfangsrichtung die Daten vieler Teilnehmer transportiert, steigt die Nutzdatenrate auf über 90 % an. Dabei werden die Voll-Duplex-Eigenschaften von 100BASE-TX vollständig ausgenutzt, sodass effektive Datenraten von über 100 Mbit/s (> 90 % von 2×100 Mbit/s) erreichbar sind.

Das Ethernet-Protokoll gemäß IEEE 802.3 bleibt auch bei modularen Baugruppen bis ins Endgerät – z. B. die elektronische Reihenklemme – erhalten, der Sub-Bus entfällt. Lediglich die Übertragungsphysik wird im Koppler von Twisted Pair bzw. Lichtleiterphysik auf E-Bus (ein alternativer Ethernet Physical Layer: LVDS, Low Voltage Differential Signaling) gewandelt, um den Anforderungen modularer Geräte gerecht zu werden. Am Ende einer modularen Station wird wieder auf 100BASE-TX umgesetzt.

EtherCAT-Eigenschaften: Das Protokoll

Das für Prozessdaten optimierte EtherCAT-Protokoll wird dank eines speziellen Ether-types direkt im Datenbereich des Ethernet Frame transportiert. Es kann aus mehreren Sub-Telegrammen bestehen, die jeweils einen Speicherbereich des bis zu 4 GByte großen logischen Prozessabbildes bedienen.

Die datentechnische Reihenfolge ist dabei unabhängig von der physikalischen Reihenfolge der Ethernet-Klemmen im Netz, es kann wahlfrei adressiert werden. Broadcast, Multicast und Querkommunikation zwischen Slaves sind möglich. Die Übertragung direkt im Ethernet Frame wird stets dann eingesetzt, wenn höchste Performance gefragt ist und die EtherCAT-Komponenten im gleichen Subnetz wie die Steuerung betrieben werden. Der Einsatzbereich von EtherCAT ist jedoch nicht auf ein Subnetz beschränkt: EtherCAT UDP verpackt das EtherCAT-Protokoll in UDP/IP-Datagramme (siehe Abb. 4.79).

Hiermit kann jede Steuerung mit Ethernet-Protokoll-Stack EtherCAT-Systeme ansprechen. Selbst die Kommunikation über Router hinweg in andere Subnetze ist möglich. Selbstverständlich hängt die Leistungsfähigkeit des Systems in dieser Variante von den Echtzeiteigenschaften der Steuerung und ihrer Ethernet-Protokollimplementierung ab. Die Antwortzeiten des EtherCAT-Netzwerks an sich werden jedoch nur minimal eingeschränkt: Lediglich in der ersten Station muss das UDP-Datagramm entpackt werden.

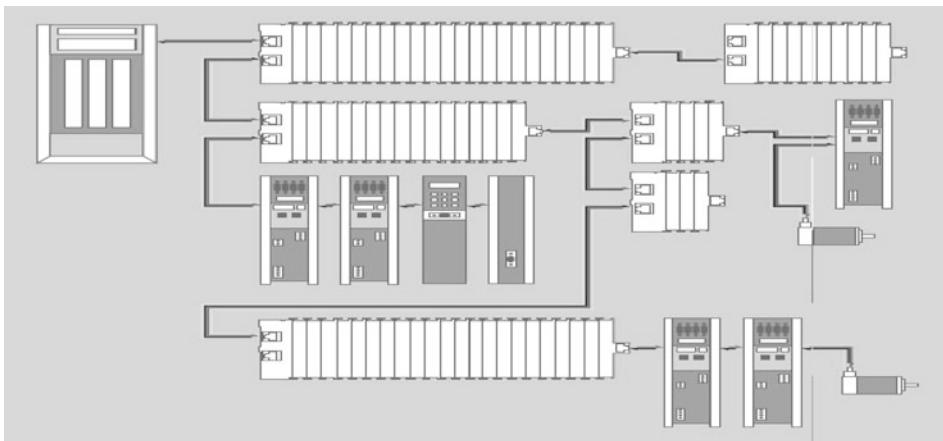


Abb. 4.80 Flexible Topologien: Linie, Baum oder Stern

Netzwerkvariablen ermöglichen zudem die effiziente Kommunikation zwischen Mästern, z. B. bei der Verkettung von Anlageteilen oder hierarchisch aufgebauten Systemen mit unterlagerten Steuerungen.

EtherCAT verwendet ausschließlich Standard-Frames nach IEEE 802.3, sie werden nicht verkürzt. Damit können EtherCAT-Frames von beliebigen Ethernet Controllern (Master) verschickt werden, und Standard Diagnose Tools (z. B. Monitor) können eingesetzt werden. Die Größe eines Frames kann je nach Datenaufkommen bei jedem Zyklus variabel sein.

Die EtherCAT-Topologie

Mit EtherCAT werden nahezu alle beliebigen Topologien, wie Linie, Baum oder Stern, unterstützt (siehe Abb. 4.80). Die von den Feldbussen her bekannte Bus- oder Linienstruktur wird damit auch für Ethernet verfügbar.

Besonders praktisch für die Anlagenverdrahtung ist die Kombination aus Linie und Abzweigen, bzw. Stichleitungen: Die benötigten Schnittstellen sind auf vielen Geräten vorhanden, zusätzliche Switches werden nicht benötigt. Die maximale Flexibilität bei der Verdrahtung wird durch die Auswahl verschiedener Leitungen vervollständigt. Flexible und sehr preiswerte Standard-Ethernet-Patch-Kabel übertragen die Signale auf Fast Ethernet-Art (100BASE-TX). Lichtwellenleiter ergänzen das System für spezielle Anwendungsfälle. Die gesamte Bandbreite der Ethernet-Vernetzung – wie verschiedenste Lichtleiter und Kupferkabel – kann in der Kombination mit Switches oder Medienumsetzern zum Einsatz kommen.

Die Fast-Ethernet-Physik erlaubt eine Leitungslänge von maximal 100 m zwischen zwei Teilnehmern, der E-Bus (LVDS) ist nur bei modularen Geräten als physikalische Schicht vorgesehen. Für jede Leitungsstrecke kann die Signalvariante individuell ausge-

wählt werden. Da bis zu 65 535 Teilnehmer angeschlossen werden können, ist die gesamte Netzausdehnung nahezu unbeschränkt.

Dank der optionalen Leitungsredundanz, die hardwareseitig lediglich einen zweiten Ethernet-Port im Master erfordert, werden auch Hochverfügbarkeitsanforderungen erfüllt. Gerätetausch bei laufendem Netzwerk ist dann ebenfalls möglich.

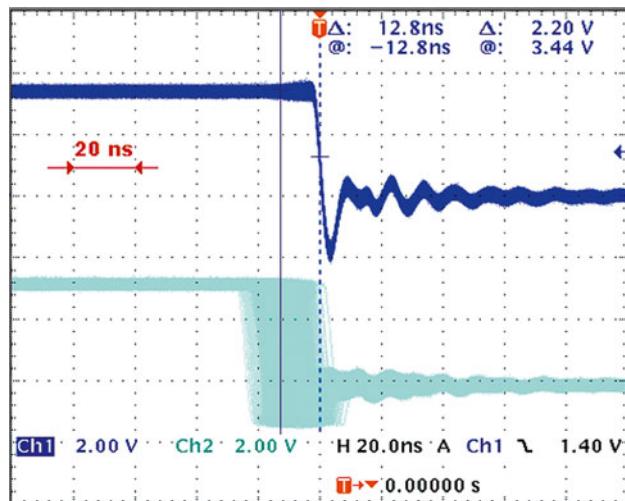
Verteilte Uhren

Der exakten Synchronisierung kommt immer dann eine besondere Bedeutung zu, wenn räumlich verteilte Prozesse gleichzeitige Aktionen erfordern. Das kann z. B. in Applikationen der Fall sein, wo mehrere Servo-Achsen gleichzeitig koordinierte Bewegungen ausführen.

Der leistungsfähigste Ansatz zur Synchronisierung ist der exakte Abgleich verteilter Uhren – wie auch im Standard IEEE 1588 beschrieben. Im Gegensatz zur vollsynchrone Kommunikation, deren Synchronisationsqualität bei Kommunikationsstörungen sofort leidet, verfügen verteilte abgeglichechte Uhren über ein hohes Maß an Toleranz gegenüber möglichen störungsbedingten Verzögerungen im Kommunikationssystem.

Bei EtherCAT basiert der Prozessdatenaustausch vollständig auf einer reinen Hardware-Maschine. Da die Kommunikation eine logische (und dank Voll-Duplex Fast Ethernet auch physikalische) Ringstruktur nutzt, kann die Mutter-Uhr den Laufzeitversatz zu den einzelnen Tochter-Uhren einfach und exakt ermitteln, und umgekehrt. Auf Basis dieses Wertes werden die verteilten Uhren nachgeführt, und es steht eine hochgenaue, netzwerkweite Zeitbasis zur Verfügung, deren Jitter deutlich unter einer Mikrosekunde beträgt (siehe Abb. 4.81). Die externe Synchronisierung – z. B. standortweit – erfolgt dann über IEEE 1588.

Abb. 4.81 Synchronität und Gleichzeitigkeit. Oszillogramm an zwei Geräten. Zwischen den 2 Messpunkten liegen 300 Knoten und 120 m Leitungslänge



Messergebnisse zeigt Abb. 4.81: Das Oszilloskop triggert auf das Sync-Interrupt des ersten Teilnehmers (oben) und zeigt unten denjenigen des dreihundertsten. Die Breite der Signalflanke (Nachleuchtzeit auf unendlich) entspricht dem Synchronisationsfehler und liegt bei ± 20 ns. Die Abweichung des Mittelwertes vom Triggerpunkt des ersten Teilnehmers entspricht dem Gleichzeitigkeitsfehler und liegt hier bei ca. 12 ns.

Hochauflösende verteilte Uhren dienen aber nicht nur der Synchronisierung, sondern können auch exakte Informationen zum lokalen Zeitpunkt der Datenerfassung liefern. Steuerungen berechnen beispielsweise häufig Geschwindigkeiten aus nacheinander gemessenen Positionen.

Speziell bei sehr kurzen Abtastzeiten führt schon ein kleiner zeitlicher Jitter in der Weg erfassung zu großen Geschwindigkeitssprüngen. Konsequenterweise werden mit EtherCAT auch Timestamp-Datentypen eingeführt. Mit dem Messwert wird die hochauflösende Systemzeit verknüpft – die große Bandbreite von Ethernet macht das möglich. Damit hängt dann die Genauigkeit einer Geschwindigkeitsberechnung nicht mehr vom Jitter des Kommunikationssystems ab. Sie wird um Größenordnungen besser als diejenige von Messverfahren, die auf jitterfreier Kommunikation basieren.

Performance

Dank Hardware-Integration im Slave und DMA-Zugriff auf die Netzwerkkarte im Master erfolgt die gesamte Protokollbearbeitung in Hardware und ist damit unabhängig von der Laufzeit von Protokollstacks, von CPU-Performance oder Software-Implementierung. Die Update-Zeit für 1000 E/As beträgt nur 30 μ s – einschließlich I/O-Durchlaufzeit (siehe Tabelle in Abb. 4.82). Mit einem einzigen Ethernet Frame können bis zu 1486 Byte Prozessdaten ausgetauscht werden – das entspricht fast 12 000 digitalen Ein- und Ausgängen. Für die Übertragung dieser Datenmenge werden dabei nur 300 μ s benötigt.

Die Kommunikation mit 100 Servoachsen kann alle 100 μ s erfolgen. In diesem Zeitabstand werden alle Achsen mit Sollwerten und Steuerdaten versehen und melden Ist-Position und Status. Durch das Verfahren der verteilten Uhren können die Achsen

Prozessdaten	Update-Zeit
256 verteilte digitale E/A	11 μ s - 0,01 ms
1000 verteilte digitale E/A	30 μ s
200 analoge E/A (16 Bit)	50 μ s <-> 20 kHz
100 Servoachsen, je 8 Byte Ein- und Ausgangsdaten	100 μ s
I Feldbus Master Gateway (1486 Bytes Eingangs- und 1480 Bytes Ausgangsdaten)	150 μ s

Abb. 4.82 Performance-Übersicht EtherCAT

dabei mit einer Abweichung von deutlich weniger als einer Mikrosekunde synchronisiert werden.

Die extrem hohe Performance der EtherCAT-Technologie ermöglicht Steuerungs- und Regelungskonzepte, die mit klassischen Feldbussystemen nicht realisierbar waren. Mit EtherCAT steht eine Kommunikationstechnologie zur Verfügung, die der überlegenen Rechenleistung moderner Industrie-PCs entspricht. Das Bussystem ist nicht mehr der Flaschenhals im Steuerungskonzept. Verteilte E/As werden schneller erfasst, als dies mit den meisten lokalen E/A-Schnittstellen möglich ist. Das EtherCAT-Technologieprinzip ist skalierbar und nicht an die Baudrate von 100 MBd gebunden – eine Erweiterung auf GBit-Ethernet ist möglich.

Diagnose

Die Erfahrungen mit Feldbussystemen zeigen, dass die Verfügbarkeit und Inbetriebnahmezeiten entscheidend von der Diagnosefähigkeit abhängen. Nur eine schnell und präzise erkannte sowie eindeutig lokalisierbare Störung kann kurzfristig behoben werden. Deshalb wurde bei der Entwicklung des EtherCAT Systems besonderer Wert auf vorbildliche Diagnoseeigenschaften gelegt.

Bei der Inbetriebnahme gilt es zu prüfen, ob die Ist-Konfiguration der Geräte mit der Soll-Konfiguration übereinstimmt. Auch die Topologie sollte der Konfiguration entsprechen. Durch die eingebaute Topologie-Erkennung bis hinunter zu den einzelnen Sub-Modulen kann nicht nur diese Überprüfung beim Systemstart stattfinden – auch ein automatisches Einlesen des Netzwerkes ist möglich (Konfigurations-Upload).

Bitfehler in der Übertragung werden durch die Auswertung der CRC-Prüfsumme zuverlässig erkannt. Das 32 Bit CRC-Polynom weist eine minimale Hamming-Distanz von 4 auf. Neben der Bruchstellenerkennung und -lokalisierung erlauben Protokoll, Übertragungsphysik und Topologie des EtherCAT-Systems eine individuelle Qualitätsüberwachung jeder einzelnen Übertragungsstrecke. Die automatische Auswertung der entsprechenden Fehlerzähler ermöglicht die exakte Lokalisierung kritischer Netzwerkabschnitte. Schleichende oder wechselnde Fehlerquellen wie EMV-Einflüsse, fehlerhafte Steckverbindungen oder Kabelschäden werden erkannt und lokalisiert, auch wenn sie die Selbstheilungsfähigkeit des Netzwerkes noch nicht überfordern.

EtherCAT statt IPC

Mit der fortschreitenden Verkleinerung der PC-Komponenten wird die Baugröße von Industrie-PCs zunehmend von der Anzahl der benötigten Steckplätze bestimmt. Die Bandbreite von Fast Ethernet zusammen mit der Datenbreite der EtherCAT-Kommunikations-Hardware ermöglicht hier neue Wege zu gehen. Klassisch im IPC vorgesehene Schnittstellen werden in intelligente Schnittstellenklemmen am EtherCAT-System ausgelagert (siehe Abb. 4.83).

Über einen einzigen Ethernet-Port im PC können dann neben den dezentralen E/As, Achsen und Bediengeräten auch komplexe Systeme wie Feldbus-Master, schnelle serielle Schnittstellen, Gateways und andere Kommunikations-Interfaces angesprochen werden.

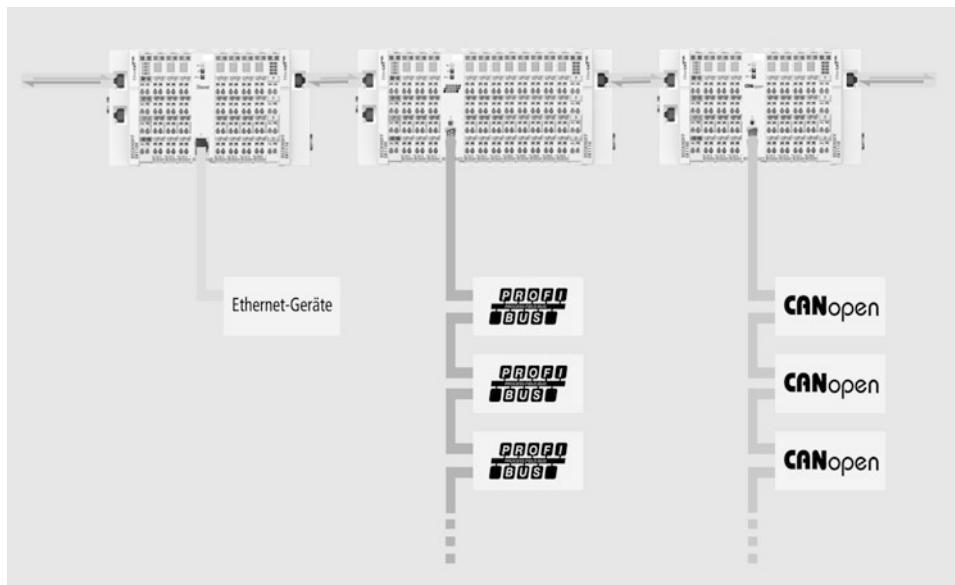


Abb. 4.83 Dezentrale Feldbus-Schnittstellen

Selbst weitere Ethernet-Geräte mit beliebigen Protokollvarianten lassen sich über dezentrale Switchports anschließen. Der zentrale IPC (Master) wird kleiner und damit kostengünstiger, eine einzige Ethernet-Schnittstelle genügt zur kompletten Kommunikation mit der Peripherie.

Geräteprofile

Die Geräteprofile beschreiben die Anwendungs-Parameter und das funktionale Verhalten der Geräte, einschließlich der geräteklassenspezifischen Zustandsmaschinen. In der Feldbustechnik gibt es für viele Geräteklassen bereits bewährte Geräteprofile – z. B. für E/A-Geräte, Antriebe oder Ventile. Die Anwender sind mit diesen Profilen, den entsprechenden Parametern und den dazugehörigen Tools vertraut.

Daher verzichtet man bei EtherCAT darauf, eigene Geräteprofile für bereits abgedeckte Geräteklassen zu entwickeln. Vielmehr werden einfache Schnittstellen für bestehende Geräteprofile angeboten denn diese sorgen für die Kompatibilität und den effizienten Datenaustausch zwischen Steuerung und Antrieb. Damit wird die Migration vom bisherigen Feldbus zu EtherCAT sowohl für Anwender als auch für Gerätehersteller deutlich erleichtert.

CANopen over EtherCAT (CoE)

CANopen-Geräte- und Applikationsprofile stehen für eine große Vielfalt von Geräteklassen und Anwendungen zur Verfügung: Angefangen von den E/A-Baugruppen über

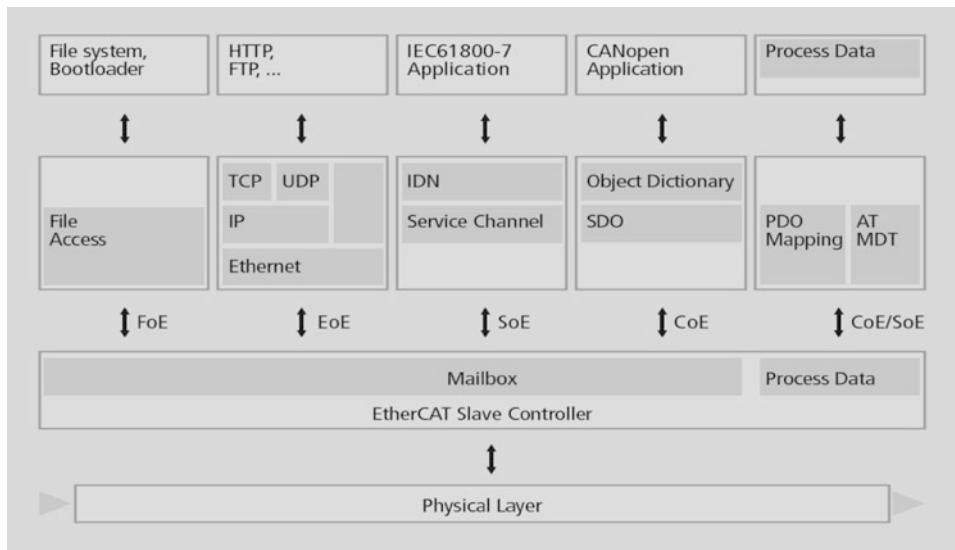


Abb. 4.84 Multiprotokollfähigkeit der EtherCAT-Mailbox. Mehrere Geräteprofile und Protokolle sind nebeneinander möglich

Antriebe (z. B. Antriebsprofil DS 402), Encoder, Proportionalventile und Hydraulikregler, bis hin zu Anwendungsprofilen etwa für die Kunststoffverarbeitung oder Textilmaschinen. EtherCAT kann die gleichen Kommunikationsmechanismen bereitstellen, wie sie vom CANopen Standard EN 50325-4 her bekannt sind: Objektverzeichnis, PDO (Process Data Objects) und SDO (Service Data Objects), selbst das Netzwerkmanagement ist vergleichbar. So kann EtherCAT auf Geräten, die bisher mit CANopen ausgestattet waren, mit minimalem Aufwand implementiert werden, große Teile der CANopen Firmware sind wieder verwendbar. Dabei lassen sich die Objekte optional erweitern, um einerseits die 8-Byte-Beschränkung aufzuheben und andererseits die vollständige Auslesbarkeit des Objektverzeichnisses zu ermöglichen. So wird auch der größeren Bandbreite von EtherCAT Rechnung getragen.

Servodrive-Profil over EtherCAT (SoE)

SERCOS interfaceTM ist als leistungsstarke Echtzeit-Kommunikationsschnittstelle insbesondere für anspruchsvolle Motion Control Anwendungen weltweit anerkannt und geschätzt. Das SERCOS-Profil für Servoantriebe und die Kommunikationstechnologie sind in der IEC 61800-7 genormt. In dieser Norm ist auch das Mapping des Sercos-Servodrive-Profils auf EtherCAT enthalten. Der SERCOS-Servicekanal, und damit der Zugriff auf alle antriebsinternen Parameter und Funktionen, wird auf die EtherCAT-Mailbox abgebildet (siehe Abb. 4.84).

Auch hier stehen sowohl die Kompatibilität zum bestehenden Profil (Zugriff auf Wert, Attribute, Namen, Einheiten etc. der IDNs) als auch die Erweiterungsmöglichkeit bezüg-

lich der Datenlängenbeschränkung im Vordergrund. Die Prozessdaten, bei SERCOS in Form von AT- und MDT-Daten, werden wiederum mit den Mitteln des EtherCAT-Slave-Controllers übertragen. Das entsprechende Mapping erfolgt analog zu SERCOS.

Die EtherCAT-Slave-State-Machine lässt sich ebenfalls gut auf die Phasen des SERCOS-Protokolls abbilden. Damit lassen sich die Vorteile dieses vor allem im CNC-Bereich weit verbreiteten Geräteprofils mit denen von EtherCAT kombinieren. Die netzwerkweite exakte Synchronisierung ist dabei durch die verteilten Uhren gewährleistet. Wahlweise können Sollposition, Drehzahl oder Drehmoment übertragen werden.

Je nach Implementierung können sogar die gleichen Konfigurations-Tools für die Antriebe weiter verwendet werden.

Ethernet over EtherCAT (EoE)

Die EtherCAT-Technologie ist nicht nur vollständig Ethernet-kompatibel, sondern „by design“ durch besondere Offenheit gekennzeichnet: Das Protokoll verträgt sich mit weiteren Ethernet-basierten Diensten und Protokollen auf dem gleichen physikalischen Netz, in der Regel mit nur minimalen Einbußen bei der Performance. Beliebige Ethernet-Geräte können innerhalb des EtherCAT-Segmentes via Switchport angeschlossen werden. Die Ethernet Frames werden durch das EtherCAT-Protokoll getunnelt, wie es bei den Internet-Protokollen üblich ist (z. B. TCP/IP, VPN, PPPoE (DSL) etc.). Das EtherCAT-Netzwerk ist dabei für die Ethernet-Geräte voll transparent und die EtherCAT-Echtzeiteigenschaften werden nicht beeinträchtigt.

EtherCAT-Geräte können zusätzlich über andere Ethernet-Protokolle (wie z. B. HTTP) verfügen und damit nach außen wie ein Standard Ethernet-Teilnehmer auftreten. Der Master fungiert dabei wie ein Layer-2-Switch, der die Frames gemäß der Adressinformation zu den entsprechenden Teilnehmern weiterleitet.

Damit können sämtliche Internet-Technologien auch im EtherCAT-Umfeld zum Einsatz kommen: Integrierte Webserver, Email, FTP-Transfer etc.

File Access over EtherCAT (FoE)

Dieses an TFTP angelehnte, sehr einfache Protokoll ermöglicht den Zugriff auf beliebige Datenstrukturen im Gerät. Damit ist z. B. ein einheitlicher Firmware-Upload auf Geräte möglich, unabhängig davon, ob diese TCP/IP unterstützen.

Safety over EtherCAT

Das EtherCAT-Safety-Protokoll ermöglicht es, Standard-Steuerungstechnik und Sicherheitstechnik mit einem Bussystem abzudecken. Damit entfällt der separate Sicherheitsbus und die Safety-Daten stehen auch in Standard-Steuerungen zur Verfügung. Die TÜV-zertifizierte Technologie wurde nach IEC 61508 entwickelt und deckt Anforderungen bis SIL4 ab. Eine Geräteimplementierung wird in der Regel nach SIL3 durchgeführt. Sie eignet sich gleichermaßen für sichere E/A, wie auch für die sichere Antriebstechnik.

Dabei verursacht Safety over EtherCAT keine Einschränkung bezüglich Übertragungsgeschwindigkeit und Zykluszeit, da EtherCAT als einkanaliges Kommunikationsmedium

genutzt wird. Das Transportmedium wird dabei als „Black Channel“ betrachtet und nicht in die Sicherheitsbetrachtung mit einbezogen.

Masterimplementierung

EtherCAT verwendet Standard Ethernet-Controller dort, wo es tatsächlich Kosten spart, nämlich im Master. Da in der Regel nur ein Ethernet Frame je Zyklus abgeschickt werden muss, kann auf Kommunikations-Coprozessoren verzichtet werden. Damit ist EtherCAT eine Ethernet-Lösung für harte Echtzeitanforderungen, die ohne spezielle Master-Einsteckkarten auskommt, der on-board Ethernet-Controller oder eine günstige Standard-NIC-Karte genügen. Master als reine Software-Lösung wurden auf einer Vielzahl von Betriebssystemen implementiert.

Speziell für die kleine und mittlere Steuerungstechnik und für klar umrissene Anwendungen ist die Implementierung eines EtherCAT Masters sehr einfach: Betrachtet wird eine SPS mit einem einzigen Prozessabbild: Wenn dieses 1486 Byte nicht übersteigt, so genügt das zyklische Versenden eines einzigen Ethernet Frames, und zwar mit der Zykluszeit der SPS. Da sich der Header zur Laufzeit nicht ändert, muss also lediglich ein konstanter „Vorspann“ zum Prozessabbild hinzugefügt und das Ergebnis dem Ethernet Controller übergeben werden.

Dabei ist das Prozessabbild bereits fertig sortiert, da das Mapping bei EtherCAT nicht im Master, sondern in den Slaves erfolgt. Die Peripheriegeräte fügen ihre Daten an die entsprechende Stelle im durchlaufenden Frame ein und lesen die für sie bestimmten Daten im Durchlauf.

Für Masterimplementierungen stellt die EtherCAT-Gemeinde auf der EtherCAT Webseite eine ganze Bandbreite an Möglichkeiten bereit, von Open Source EtherCAT-Master-Library über Master-Sample-Code für Windows bis hin zu kompletten Paketen für verschiedene Echtzeit-Betriebssysteme und Prozessoren.

Monitor-Tools

Da EtherCAT Standard Ethernet Frames nach IEEE 802.3 verwendet, eignet sich jedes handelsübliche Ethernet Monitor Tool zur Beobachtung der EtherCAT Kommunikation. Zusätzlich gibt es kostenlose Parser-Software für Wireshark (ehemals Ethereal, ein Open Source Monitoring Tool) und den Microsoft Netzwerk-Monitor, mit der mitgeschnittener EtherCAT-Datenverkehr komfortabel aufbereitet und zur Anzeige gebracht wird.

Slave

Im Slave-Gerät kommt ein kostengünstiger EtherCAT-Slave-Controller (als ASIC oder FPGA) zum Einsatz. Für einfache Geräte ist kein zusätzlicher Microcontroller erforderlich. Bei komplexeren Geräten ist die Kommunikations-Performance bei EtherCAT nahezu unabhängig von der Leistungsfähigkeit des verwendeten Controllers, d. h. die Anschaltung wird entsprechend günstig. In den meisten Fällen ist ein 8-Bit-Microcontroller ausreichend.

EtherCAT-Slave-Controller sind von mehreren Herstellern verfügbar. Sie verfügen in der Regel über internes DPRAM und bieten eine Auswahl an Prozessdaten-Schnittstellen (PDI) zum Zugriff auf diesen Anwendungsspeicher:

Das serielle SPI (Serial Peripheral Interface) ist speziell für Geräte mit kleiner Prozessdatenmenge gedacht, wie z. B. analoge E/A-Module, Geber, Encoder oder auch einfache Antriebe.

Das 32-Bit-Parallel-E/A-Interface eignet sich für den Anschluss von bis zu 32 digitalen Ein- und Ausgängen, aber auch für einfache Sensoren oder Aktoren, die mit 32 Datenbits auskommen.

Das parallele 8/16-Bit-Microcontroller-Interface entspricht herkömmlichen Schnittstellen bei Feldbus-Controllern mit DPRAM-Schnittstelle. Es eignet sich besonders für komplexe Teilnehmer mit größerem Datenaufkommen.

Zur Unterstützung einer Slave-Implementierung werden Evaluation-Kits angeboten, die auch Slave-Anwendungssoftware im Source sowie Test-Master enthalten. Somit kann in wenigen Schritten ein voll funktionsfähiges Master-Slave-EtherCAT-Netzwerk in Betrieb genommen werden.

Infrastruktur-Kosten

Da EtherCAT auf Hubs und Switches verzichten kann, entfallen die entsprechenden Kosten für diese Geräte samt Spannungsversorgung, Einbau etc. Es kommen Standard Ethernet-Kabel und auch Standard Steckverbinder (z. B. RJ45, M12) zum Einsatz, wenn die Umgebungsbedingungen dies erlauben.

EtherCAT Technology Group

Jeder soll EtherCAT nutzen und implementieren können. Für diesen Ansatz steht die EtherCAT Technology Group (ETG). In der ETG finden sich Endanwender aus unterschiedlichen Branchen, Maschinenhersteller und Anbieter von leistungsfähiger Steuerungstechnik zusammen, um die EtherCAT-Technologie zu unterstützen und zu fördern.

Das unter Mithilfe von ETG-Mitgliedern entwickelte Conformance Test Tool stellt die Interoperabilität und Protokollkonformität der EtherCAT-Geräte sicher und ist damit eine Voraussetzung für den weiteren Erfolg der EtherCAT-Technologie.

Die ETG wurde im November 2003 gegründet und ist mit über 700 Mitgliedsfirmen (Stand März 2008) mittlerweile die weltgrößte Industrial-Ethernet-Organisation. Aktuell sind Firmen aus 40 Ländern in 8 Kontinenten in der ETG vertreten.

Internationale Normung

Die Offenlegung wird nicht nur innerhalb der EtherCAT Technology Group betrieben, auch die Internationale Normung von EtherCAT ist erfolgt. Die EtherCAT Technology Group ist offizieller Normungspartner der IEC-Arbeitsgruppen für digitale Kommunikation.

EtherCAT ist als „Communication Profile Family 12“ in der IEC 61158 (Protokolle und Dienste) und IEC 61784-2 (Kommunikationsprofile) integriert. In der IEC 61158 sind

EtherCAT-Protokoll und die Dienste standardisiert, während die IEC 61784-2 die Profile für die spezifischen Geräteklassen definiert.

In der IEC 61800-7 (Antriebsprofile und -kommunikation) ist EtherCAT als Kommunikationstechnologie für das SERCOS- und das CANopen-Antriebsprofil genormt. Auch in ISO 15745-4 (Gerätebeschreibung mit XML) ist EtherCAT enthalten. Seit September 2007 ist EtherCAT zudem SEMI-Standard: Die E54.20 beschreibt den Einsatz der Technologie in Halbleiter- und Flachdisplay-Fertigungsanlagen.

Zusammenfassung

- Kurze Zykluszeiten: 1000 E/As in 30 µs.
- Flexible Topologie mit nahezu unbeschränkter Ausdehnung: Mit EtherCAT kann die Ethernet-Sterntopologie durch eine einfache Linienstruktur ersetzt werden, teure Infrastrukturkomponenten entfallen. EtherCAT kann aber auch klassisch mit Switches verkabelt werden, um andere Ethernet-Teilnehmer zu integrieren.
- EtherCAT kommt mit kostengünstigen Standard Ethernet-Karten (NICs) aus.
- Ethernet bis in die E/A-Ebene wird durch EtherCAT technisch möglich und wirtschaftlich sinnvoll.
- Schneller Antriebs- und E/A-Bus am Industrie-PC oder auch in Kombination mit kleiner Steuerungstechnik.
- Vielzahl verfügbarer Geräte unterschiedlicher Hersteller.

4.4 Peripheriebusse am PC

4.4.1 Vergleich USB – Firewire

Beide Bussysteme sind sich sehr ähnlich und dienen – zumindest vom Konzept her – dem gleichen Zweck: Die Verbindung mehrerer Peripheriegeräte mit dem PC über das Medium Bus (im Gegensatz zu der herkömmlichen Methode vieler einzelner, spezifischer Buchsen am PC). Der wesentliche Unterschied ist die marktbeherrschende Stellung des USB (Universal Serial Bus) dank der „Erfinder“ Intel und Microsoft. USB ist heute PC-Standard und findet auch in kleineren Automatisierungssystemen Anwendung, während Firewire (= IEEE 1394-Bus, = iLink) nur optional angeboten wird. Bei Apple-Rechnern ist Firewire Standard. Eine vergleichende Übersicht zeigt die Tabelle in Abb. 4.85. In der Zwischenzeit hat sich der USB auch im Bereich Messen, Steuern, Regeln etabliert.

Die Übertragungsart „isochron“ bedeutet, dass ein Synchronisationssignal ganze Datenblöcke (von z. B. 125 µs) taktet, nicht aber, wie bei einer synchronen Übertragung, jedes Telegramm. Die isochrone Übertragung ist bei Audio- und Videosignalen notwendig, um eine „ruckelfreie“ und schnelle Übertragung zu gewährleisten. Die sonst bei allen Bussen übliche Fehlererkennung bzw. -korrektur ist aus demselben Grund bei isochroner Übertragung nicht vorgesehen.

	IEEE 1394 (FIREWIRE, ILINK)	USB UNIVERSAL SERIAL BUS
Topologie	Linie, (daisy chain)	Stern; kaskadierbar PC ist Master und erster Hub
Übertragungsrate	1394A: 400 MBaud 1394B: 800 MBaud	USB 1.1: 12 MBaud USB 2.1: 480 MBaud
Übertragungsart	asynchron, isochron	asynchron, isochron (USB 2.1)
Teilnehmer	≤ 63	≤ 128
Telegrammlänge	isochron: $\leq 6,25$ KByte (Telegramm = Zyklus)	isochron: $\leq 7,5$ KByte asynchron: 8 – 256 Byte (Telegramm = Microframe)
Leitung	$\leq 4,5$ m	≤ 5 m
Anwendung	Multimedia	PC-Peripherie
ab Windows...	2000	98, nicht NT
Start-Firmen	Apple, Sony	Intel, Microsoft

Abb. 4.85 Tabellarischer Vergleich USB und Firewire

Der Anwender bemerkt eigentlich nur an der Topologie den Unterschied zwischen USB und Firewire. USB hat eine Sternstruktur, wobei die einzelnen Sternpunkte (Hubs) kaskadierbar sind (bis zu 5 Ebenen), der erste Hub sitzt im PC (Abb. 4.86). Der Hub übernimmt

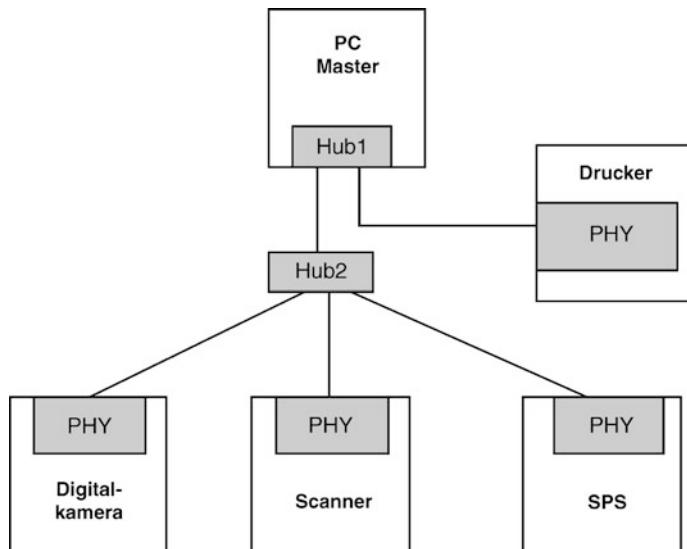
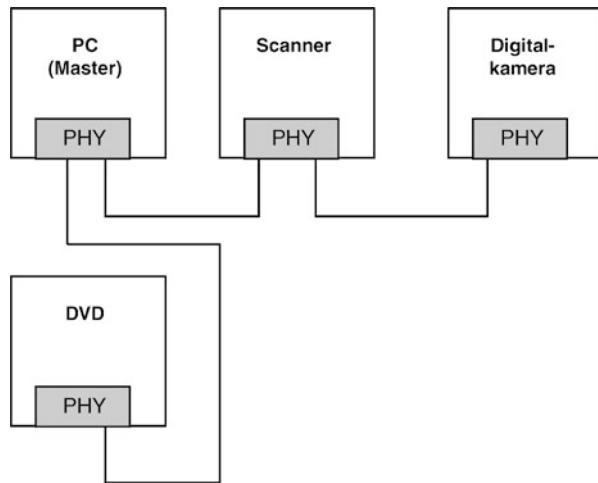


Abb. 4.86 Topologie USB. Die externen Hubs sind kaskadierbar. (PHY – physikalische Schicht, z. B. RISC Cypress CY7C63000)

Abb. 4.87 Topologie Firewire (PHY – physikalische Schicht)



neben der Stromversorgung (5 V, 100 mA/Gerät) auch die Verteilung des Datenflusses der angeschlossenen Geräte. Der PC ist der Master, also hat man trotz Sternstruktur ein Master/Slave-System.

Im Gegensatz zu USB hat Firewire eine Linienstruktur (daisy chain), die von einem Master, dem PC, gesteuert wird. Die physikalische Schicht des 7-Schichtenmodells wird von einem Chip PHY im Gerät verwaltet, d. h., der Busknoten sitzt im Gerät selbst (Abb. 4.87).

4.4.2 USB

Die aktuelle Version USB 2.0 bietet 3 „Kanäle“:

low speed: 1,5 MBd für einfache Peripheriegeräte, wie Maus, Tastatur oder auch Sensoren und Aktoren der Automatisierungstechnik. Hier sind auch längere Kabel als die spezifizierten 5 m möglich.

full speed: 12 MBd für schnelle Peripheriegeräte wie Scanner, Drucker, Kameras usw.

high speed: 480 MBd für Audio- und Videoanwendungen.

Die Datenübertragung auf OSI-Schicht 1 erfolgt in NRZI-Codierung (Non Return to Zero Insert = Nullen bewirken einen Pegelwechsel, Einsen lassen den Pegel unverändert. Bei 6 oder mehr Einsen erfolgt bit stuffing). Der Empfänger generiert aus dem NRZI-Signal Daten und den Takt.). Zur Synchronisation besitzt jedes Datenpaket (8–256 Byte) als Vorspann ein Synchronisationsbyte 0000 0001. All dies sind Elemente der klassischen Bus- und Übertragungstechnik.

Das USB-Kabel hat 4 aktive Leitungen und 2 Stecker-/Buchsentypen (Abb. 4.88). Man unterscheidet:

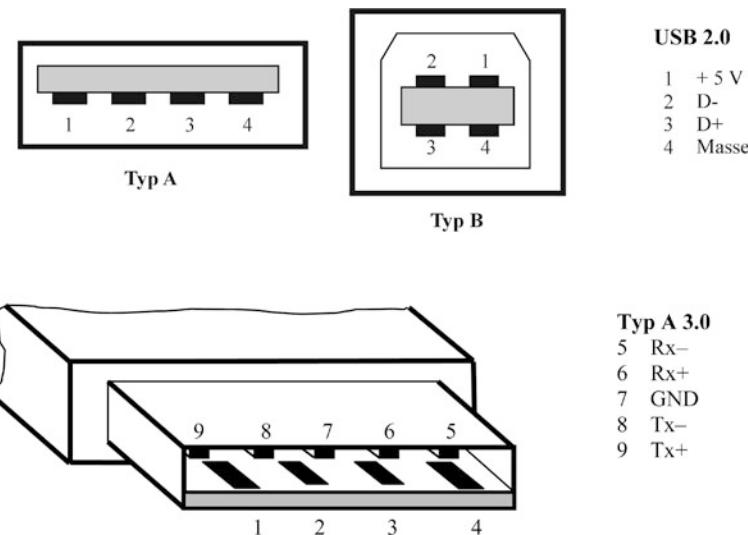


Abb. 4.88 Buchsenformen einer USB-Einheit. Typ A: PC oder Hub, Typ B: Peripheriegerät, Typ A 3.0: Superspeed

- Die Datenleitungen D+ und D-. Sie tragen das Datensignal symmetrisch zur Masse, wie dies in RS 485 spezifiziert ist. Dies ergibt eine hohe Störfestigkeit.
- Die Versorgungsleitungen 5 V und 0. Damit werden die kleineren Peripheriegeräte direkt mit Strom versorgt (5 V, 100 mA/Gerät, maximal 500 mA), z. B. Maus, Tastatur, ja selbst Scanner.

USB-Peripheriegeräte haben eine Typ-B-Buchse, der Host (= PC) bzw. Hub hat Typ-A-Buchsen. Die Kontaktstreifen für die Stromversorgung sind etwas länger als die Datenkontakte. Damit liegt das Gerät beim Einsticken an Spannung, bevor der Datenaustausch beginnt.

Eine Eigenheit von USB ist, dass dem PC über das Potential der Datenleitungen hardwaremäßig signalisiert wird, ob er es mit einem Lowspeed- oder einem Fullspeed-Gerät zu tun hat (Abb. 4.89). Bei Lowspeed liegt die Datenleitung D- auf 3,3 V-Potential, bei Fullspeed die Leitung D+.

Die Schichten 1 und 2 des OSI-Modells (Datencodierung und Leitungsüberwachung) werden hardwaremäßig von einer SIE (Serial Interface Engine) betreut. Diese kann als Einzelbaustein zum Anschluss an einen separaten μ Controller vorliegen oder auf demselben Chip mit einem RISC (Reduced Instruction Set Controller) verbunden sein, der nicht nur Schicht 3 (Datenorganisation), sondern auch OSI-Schicht 6 (Schnittstelle zur Gerätehardware) realisiert. Als Beispiel für einen lowspeed USB- μ Controller dieser Art zeigen wir das vereinfachte Blockbild des CY7C63000X von Cypress (Abb. 4.90).

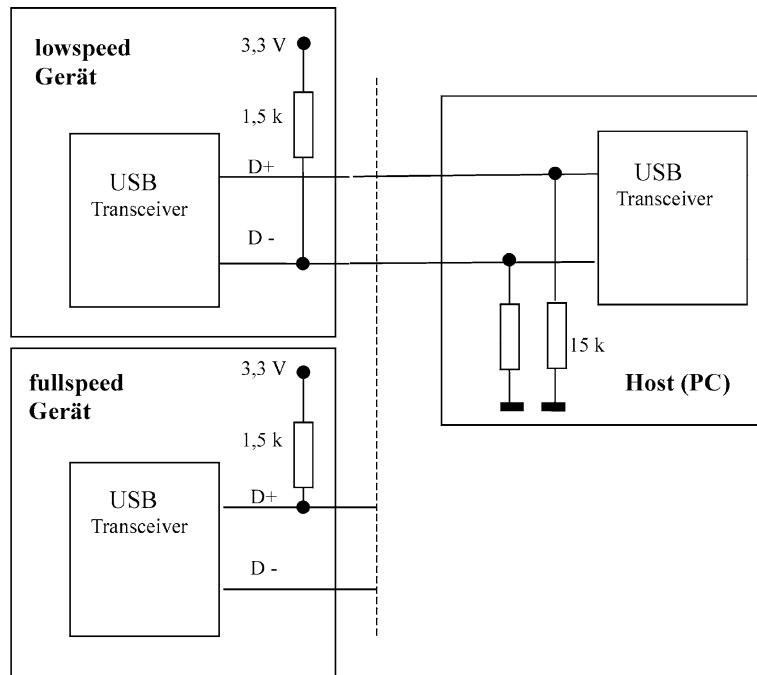


Abb. 4.89 Die Erkennung von Full- und Lowspeed-Geräten erfolgt über das Potential der Datenleitungen

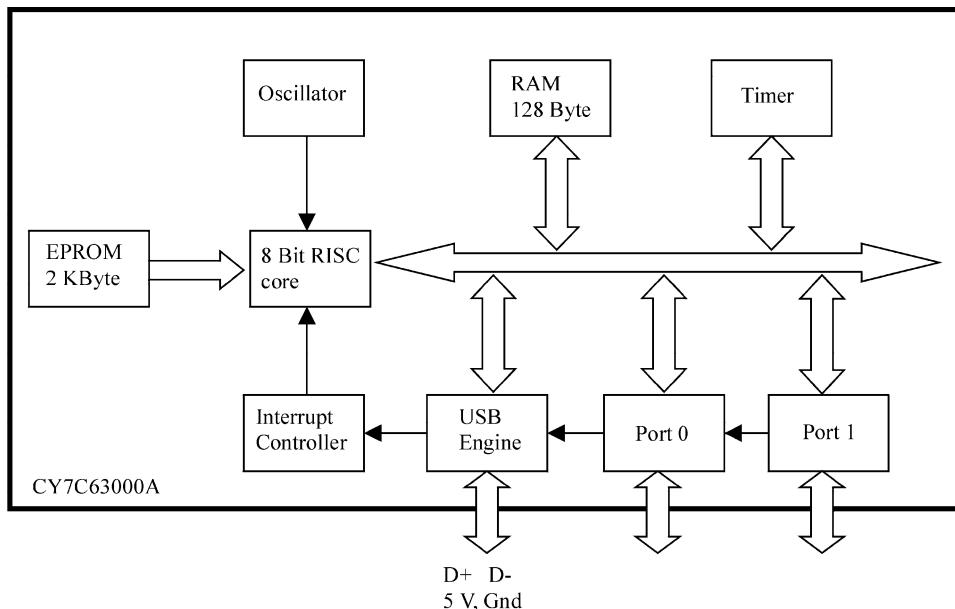


Abb. 4.90 Vereinfachtes Blockbild des USB-Controllers Cypress CY7C763000 für lowspeed-Anwendungen

Er ist für einfache USB-Geräte gedacht, wie Maus, Tastatur, Joystick usw. Sein RISC ist mit 40 Befehlen programmierbar. Da er über 16 frei programmierbare bidirektionale Ein- und Ausgänge (2 Ports) verfügt, kann er auch für kleinere Automatisierungsaufgaben eingesetzt werden. Die Ports können LEDs ansteuern, Schalter abfragen oder auch analog/digital gewandelte Messwerte übernehmen. Sie sind die Schnittstelle zur „Realität“. Fertige Platinen mit derartigen USB-Chips sind im Handel erhältlich.

Die Kommunikation zwischen USB-Gerät und Master (PC) beginnt mit der *Enumeration*. Dabei wird vom Master jedem angeschlossenen USB-Gerät eine eigene Adresse zugeordnet. Diese gilt nur für die laufende Kommunikation, bei der nächsten Inbetriebnahme des PC kann dasselbe Gerät eine andere Adresse zugewiesen bekommen. Nach der Enumeration erfolgt der Datenaustausch.

Dabei gibt es vier mögliche Transferarten:

- *Control Transfer*: Zur Steuerung der USB-hardware werden *Control Requests* verschickt.
- *Bulk Transfer*: Dabei werden große Datenmengen übertragen, die nicht zeitkritisch sind, aber eine Fehlerüberwachung benötigen. Die Datenrate wird der Auslastung des Busses angepasst, d. h., die Priorität dieser Daten ist niedrig. Beispiele: Drucker, Scanner, Digitalkamera.
- *Isochroner Transfer*: Dabei werden große Datenmengen mit definierter und fester Datenrate übertragen. Es erfolgt keine Fehlererkennung, da sie bei Audio- und Videoübertragung nur zu sichtbaren Verzögerungen führen würde.
- *Interrupt Transfer*: Diese Bezeichnung ist irreführend, da eigentlich ein Polling stattfindet: Lieferanten kleiner Datenmengen (Maus, Tastatur u. Ä.) werden vom Master alle 10 ms nach neuen Daten abgefragt. Dabei werden bis zu 8 Byte Antwort übertragen.

Die USB-Daten durchlaufen im Master vom Bus bis zur Anwendersoftware mehrere Schichten:

- Der USB-Treiber ist im Lieferumfang des USB-Geräts eingeschlossen bzw. im PC bereits vorinstalliert (Maus!). Er muss zu der Firmware im EPROM des USB- μ Controllers im Gerät passen.
- Der USB-Treiber hat eine Schnittstelle zu Windows. Diese wird durch IRPs (I/O-Request Packets) realisiert, die im Windows-Kernel aktiv sind. Windows übernimmt die Enumeration selbstständig, wenn es über den Anschluss des USB-Geräts informiert wird und stellt die Daten für die übergeordnete Anwendung bereit.
- Das übergeordnete Anwendungsprogramm kann in beliebiger Sprache geschrieben sein (Visual Basic, LabView, C++ usw.). Es kommuniziert mit Windows über das API (Application Programming Interface). Das API enthält DLL (Dynamic Link Libraries), die von dem Anwendungsprogramm direkt angesprochen werden können.

Im Jahr 2008 wurde die neue Spezifikation USB 3.0 *Super Speed* vorgestellt. Damit sollen Datenübertragungsraten bis 5 Gbit/s erreichbar sein. Dafür sind – neben der

dafür ausgelegten Hard- und Software – auch neue Kabel (zusätzliche 5 Adern), neue Stecker und Buchsen erforderlich. Die USB 3.0-Kabel, -Stecker und -Buchsen sind auf USB 2.0 abwärtskompatibel, d. h., USB 3.0 ersetzt nicht USB 2.0, sondern läuft parallel für Anwendungen, die diese hohe Übertragungsgeschwindigkeit benötigen. Das wird in der Automatisierungstechnik selten vorkommen.

Wie die Abwärtskompatibilität gedacht ist, zeigt Abb. 4.88 beispielhaft. Der gängige Stecker Typ A, der in die Buchse des Hosts kommt, ist mechanisch bei USB 2.0 und USB 3.0 gleich. Auch die vier Flachkontakte von USB 2.0 bleiben unverändert. Es kommen für USB 3.0 lediglich fünf neue Kontakte hinzu. Sie sind im Typ A-Stecker oberhalb und nach hinten versetzt angeordnet (in Abb. 4.88 nicht dargestellt). Diese Kontakte sind mit den zusätzlichen fünf Adern des USB 3.0-Kabels verbunden: Rx– und Rx+ sind verdrillt und besorgen den Datentransfer vom externen Gerät zum Host, Tx– und Tx+ sind ebenfalls verdrillt und dienen dem Datentransfer vom Host zum Gerät. Beide Adernpaare sind jeweils über Masse (GND) geschirmt.

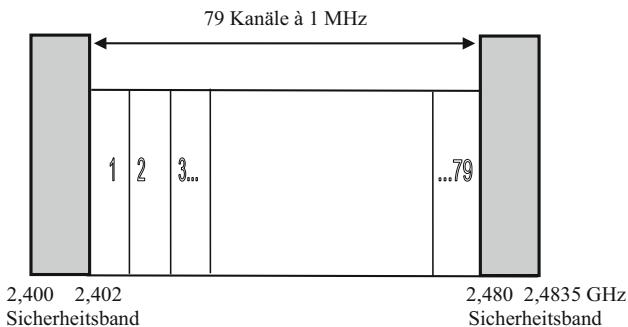
4.4.3 Bluetooth

Der PC-Peripheriebus USB ist Standard im Bereich Computertechnik, der Bus Firewire/iLink ist ein Nischenprodukt geblieben. Diesen beiden drahtgebundenen Bussen steht gegenüber die drahtlose Funkverbindung Bluetooth, die heute in allen Laptops, Tablets und Smartphones als Peripherieschnittstelle zu finden ist. Die Tabelle in Abb. 4.91 zeigt einige charakteristische Daten.

Abb. 4.91 Einige Kenndaten der Funkverbindung Bluetooth

Topologie	typisch 1 Master ↔ 1 Slave
Übertragungsrate	asymmetrisch: Empfänger: 723,2 KBd Sender: 57,6 KBd symmetrisch: 432,6 KBd
Medium Funk	ISM Band: 2,402 GHz – 2,480 GHz
Übertragungsart	Controller sendet MAC-Adresse
Teilnehmer	1↔1; möglich ≤8 (Piconet, 3Bit-Adr.)
Telegramme	Zugriffscode 72 Bit Header 54 Bit Nutzdaten 0 – 2745 Bit
Reichweite/Sende-Leistung	1 m/1mW (Klasse 3) 10 m/2,5 mW (Klasse 2) 100 m/100 mW (Klasse 1)
Startfirmen	Ericson, Nokia, Intel

Abb. 4.92 Die Bluetooth-Kanäle mit den 2 MHz- bzw. 3,5 MHz-Sicherheitsbändern im 2,4–2,5 GHz ISM-Band



Bluetooth ist ein in den 1990er Jahren von der Bluetooth SIG (*Special Interest Group*) entwickelter Industriestandard gemäss IEEE 802.15 (WPAN, *Wireless Personal Area Networks*, *Reichweite 0,2–50 m*). Es ist wohl auf die damals hauptsächlich aktiven Firmen Ericsson und Nokia zurückzuführen, dass diese Funkverbindung nach dem alten Wikingerfürsten Harald Blauzahn (engl. Bluetooth) benannt wurde. Das Bluetooth-Symbol zeigt die nordischen Runen H und B.

Bluetooth ist im Grunde eine Master-Slave-Verbindung, die nur einen Slave bedient. Millionenfach dient als Master ein Smartphone oder Laptop oder Tablet und der Slave ist ein Lautsprecher, ein Kopfhörer oder Headset, ein Drucker, usw. Auch Spiel-Konsolen verwenden diese Funkverbindung.

Da Bluetooth eine sehr zuverlässige und störfeste und erprobte Funkverbindung im ISM-Band darstellt, ist dieser Standard zunehmend auch für Industrieprodukte von Interesse, vor allem im Bereich Automatisierungstechnik. Die Profibus-Nutzerorganisation PNO hat Bluetooth für kabellose Verbindungen im Feld adaptiert. Auch CAN in Automation/CiA plant, Bluetooth für kabellose Verbindungen einzusetzen. Es ist zu erwarten, dass auch im Sensor/Aktorbereich das Prinzip Bluetooth ergänzend zur Anwendung kommen wird, wenn Kabelverbindungen nicht möglich sind. In der Medizintechnik ist diese Funkverbindung bereits im Einsatz (Insulinpumpen, intelligente Prothesen, Hörgeräte-Adapter).

Die ISM-Frequenzbänder (*Industrial, Scientific, Medical Band*) sind weltweit frei zugänglich, also lizenzzfrei, und deshalb vielfach genutzt. Um hohe Störfestigkeit zu erreichen, (Mikrowellenherde beispielsweise arbeiten bei 2,455 GHz) unterteilt Bluetooth das von ihm genutzte Band zwischen 2,4 und 2,5 GHz in 79 Kanäle à 1 MHz und wechselt diese Kanäle bis zu 1600-mal pro Sekunde (Frequenzsprungverfahren, *adaptive frequency hopping, AFH*) siehe Abb. 4.92.

Es existieren zurzeit 5 Bluetooth-Versionen:

Bluetooth 1.0: Erste Version mit einer Datenrate von 723,2 KBd. Sicherung nur mit AFH.

Bluetooth 2.0: Höhere Datenrate von 2,1 MBd. Abwärtskompatibel.

Bluetooth 3.0: High speed Version mit 3 MBd. Nutzung von WLAN.

Bluetooth 4.0: Niedrigenergie-Version BLE (Bluetooth Low Energy). Abwärtskompatibel.

Bluetooth 5: Erweiterter Entfernungsbereich 200 m und höhere Datenrate 2 MBd.

Die weite Verbreitung von Bluetooth im Consumer-Bereich hat dazu geführt, dass eine Vielzahl von Chips bis hin zur Version 5 verfügbar ist.

Literatur

Literatur zu Abschn. 4.1.1

1. W. Kriesel, O. W. Madelung (Hrsg.): AS-Interface, Das Aktuator-Sensor-Interface für die Automation, 2. überarbeitete und erweiterte Auflage Hanser, München, Wien 1999
2. C. Stoppok, H. Sturm: Vergleichende Studie von verfügbaren und in Entwicklung befindlichen Feldbussen für Sensor- und Aktuatorsysteme. VDI/VDE-TZ Informationstechnik GmbH, Berlin 1990
3. Dose, W.: Explosionsschutz durch Eigensicherheit, Braunschweig 1994
4. AS-Interface – Die Lösung in der Automation, zu beziehen über die Geschäftsstelle AS-International Association, Zum Taubengarten 52, 63571 Gelnhausen

Literatur zu Abschn. 4.1.2

5. The EIB Handbook Issue 2.21. EIBA 1996
6. The EIB Handbook Issue 3 – Vol. 2, Developer's Guide. EIBA 1997
7. Goossens, M.: A Survey of the EIB System. In: EIBA Proceedings 1997, p 49. EIBA 1997.
8. Goossens, M.: Communication and Addressing on EIB. In: EIBA Proceedings 1997, p 59. EIBA 1997
9. Goossens, M.: Object-based Distributed Application Design. In: Feldbustechnik in Forschung, Entwicklung und Anwendung (Dietrich, Ed.) p 152, Springer, 1997
10. Goossens, M.: Easy Installation and Home Management. In: Feldbustechnik in Forschung, Entwicklung und Anwendung (Dietrich, Ed.), p 387, Springer, 1997
11. Goossens, M.: Component-based Project Engineering. In: Feldbustechnik in Forschung, Entwicklung und Anwendung (Dietrich, Ed.), p 370, Springer, 1997
12. Heite, Ch., Rosch R.: A Powerline Carrier Communication System with Matched Filter Receiver Technology. In: GMM Fachbericht 17 Microelektronik 97, VDE Verlag Berlin, 1997
13. Heite, Ch., Zapp R.: Powernet – das neue EIB Medium. In: Elektrotechnik und Informations-technik – ÖVE Verbandszeitschrift 5/97

Literatur zu Abschn. 4.2.1

14. Allgemeine Einführung in die Bitbus-Terminologie. Phoenix, Kontakt, Blomberg.
15. Borst, W.: Der Feldbus in der Maschinen- und Anlagentechnik, München 1992.
16. Furrer, Frank J.: Bitbus, Grundlagen und Praxis, Heidelberg, 1994
17. Goller, V. und W. L. Giesler: Der stille Feldbus-Riese, ice 41. Jahrgang 1966, Nr. 8
18. www.bitbus.org (Bitbus European Users Group e.V.)
19. www.wikipedia.org: Bitbus

Literatur zu Abschn. 4.2.2

20. Manfred Popp, Der neue Schnelleinstieg für PROFIBUS DP; PROFIBUS Nutzerorganisation, 2002, Best.-Nr. 4.071

21. Josef Weigmann und Gerhard Kilian, Dezentralisieren mit PROFIBUS-DP/DPV1; Publicis Corporate Publishing, 2002
22. Christian Diedrich und Thomas Bangemann (Hrsg.), Profibus PA; Oldenbourg Industrieverlag, 2006
23. Karl-Heinz Niemann und Timo Kröger, Profibus Diagnose und Messungen; Oldenbourg Industrieverlag, 2010

Literatur zu Abschn. 4.2.4

24. www.de.wikipedia.org: Modbus
25. www.modbus-IDB.org (hier findet man die gesamte Spezifikation)
26. www.anybus.de

Literatur zu Abschn. 4.2.5

27. LON Nutzer Organisation: Internet-Publikationen, www.lno.de
28. EIA-709.1 Electronic Industrial Alliance, Control Network Specification. March 1998
29. Toshiba: Neuron Chip TMPN3150/3120, Datenbuch, 2001
30. LonMark Interoperability Association: Internet-Publikationen, www.lonmark.org
31. Echelon Corporation: Internet-Publikationen, www.echelon.com
32. Dietrich, D.; Loy, D.; Schweinzer, H.-J.: LON-Technologie, Hüthig Verlag Heidelberg 1999

Literatur zu Abschn. 4.2.6

33. CAN Newsletter, 1992 bis 2011. Vierteljährlich erscheinende Zeitschrift des CiA, Nürnberg.
34. CiA-Webseite; www.can.cia.org
35. Etschberger, K. u. a.: Controller Area Network. Carl Hanser, Wien, Berlin, 2000.
36. Lawrenz, W., u. a.: CAN. VDE-Verlag, Berlin, Offenbach, 2011.
37. Pfeiffer, O., u. a.: Embedded Networking with CAN and CANopen. Copperhill, Greenfiled (USA), 2003.
38. ODVA-Webseite; www.odva.org
39. Wikipedia, Stichworte: CAN, CANopen, und Devicenet; www.wikipedia.de
40. Zeltwanger, H., u. a.: CANopen. VDE-Verlag, Berlin, Offenbach, 2009.
41. Weitere Informationen zu DeviceNet: www.odva.org, www.ab.com/networks

Literatur zu Abschn. 4.2.7

42. Website der Fieldbus Foundation: <http://www.fieldbus.org>.
43. I. Verhappen, A. Pereira: Foundation Fieldbus, 3rd Edition. ISBN 978-1-934394-76-2
44. DART – The new dimension in intrinsic safety (PCIC 2008) Nachdruck erhältlich bei Pepperl+Fuchs 211671
45. G. Rogoll, R. Kitchener: Advanced Online Physical Layer Diagnostics. Technische Information, Pepperl+Fuchs 198641

Literatur zu Abschn. 4.2.8

46. www.controlnet.org
47. www.ab.com/networks
48. www.odva.org
49. <http://ab.rockwellautomation.com/networks-and-communication>

Literatur zu Abschn. 4.3.4

50. Manfred Popp; Industrielle Kommunikation mit PROFINET; PROFIBUS Nutzerorganisation (2007)
51. Raimond Pigan, Mark Metter: Automatisieren mit PROFINET: Industrielle Kommunikation auf Basis von Industrial Ethernet; Publicis Corporate Publishing (2005)
52. Manfred Popp; Das PROFINET IO-Buch; Hüthig Verlag (2005)
53. www.profinet.com

Literatur zu Abschn. 4.3.5

54. <http://www.ethernetip.de>
55. <http://www.odvaeurope.de>
56. <http://www.ab.com/networks>

Literatur zu Abschn. 4.3.6

57. www.ethernet-powerlink.org
58. www.br-automation.org
59. www.wikipedia.org: Powerlink

Literatur zu Abschn. 4.3.8

60. EtherCAT Technology Group, <http://www.ethercat.org>
61. IEEE 802.3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications
62. IEEE 802.3ae-2002: CSMA/CD Access Method and Physical Layer Specifications: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation
63. ANSI/TIA/EIA-644-A, Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits
64. IEEE 1588-2002: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
65. EN 50325-4: Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces. Part 4: CANopen
66. IEC 61800-7-301/304 (Ed.1.0), Adjustable speed electrical power drive systems
67. IEC 61508-0:2005: Functional Safety of E/E/PE safety-related systems
68. IEC 61158-3/4/5/6-12 (Ed.1.0), Industrial communication networks – Fieldbus specifications – Part 3-12: Data-link layer service definition – Part 4-12: Data-link layer protocol specification – Part 5-12: Application layer service definition – Part 6-12: Application layer protocol specification – Type 12 elements (EtherCAT)
69. IEC 61784-2 (Ed.1.0), Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3
70. www.wikipedia.org: EtherCAT. Hier finden Sie eine Simulation

Literatur zu Abschn. 4.4

71. Mörschel, M. und A. Wolf: Entwicklung einer digitalen und analogen I/O-Karte für USB. Fachhochschule Frankfurt 2001
72. Kainka, B.: Messen, Steuern und Regeln mit USB. Poing bei München 2000
73. Axelson, J.: USB Handbuch für Entwickler. Landsberg 2001
74. Kelm (Hrsg.): USB 1.1 – Universal Serial Bus. Poing 2000



5.1 ISDN

Datenfernübertragung verlief über lange Jahre ausschließlich in analoger Technik. Erst in den letzten 25 Jahren wurde nach und nach Digitaltechnik zur Daten- und Sprachübertragung eingesetzt. Die digitale Vernetzung der Übertragungsstrecken ist heute abgeschlossen. (Der Bereich von der Ortsvermittlungsstelle zum Teilnehmer ist Eigentum der Deutschen Telekom (DTK), die Weitverkehrsnetze betreiben neben der DTK auch andere Anbieter.)

Die Vorteile der digitalen Übertragung ISDN (*Integrated Services Data Network*) sind die bessere Tonqualität sowie die Integration verschiedener Dienste (s. u.). Manche dieser jetzt möglichen digitalen Übertragungsdienste sind auch für die Automatisierungstechnik interessant.

Die Dienste des integrierten digitalen Netzes (ISDN):

Telefax/Fax

Normen G2 und G3 (2,4 bis 33,6 kbit/s). ISDN nach Norm G4 (64 kbit/s).

Dieser Dienst steht in Konkurrenz mit email.

Telex (Teleprinter Exchange)

Benötigt spezielle Schreibmaschine. Übertragungsrate 50 Zeichen/s. Am 31.12.2007 stellte die Deutsche Telekom diesen Dienst ein.

Teletex

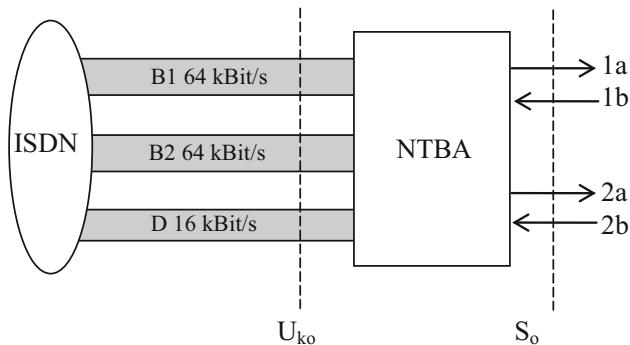
Weiterentwicklung von Telex. Hat sich nicht durchgesetzt.

Btx (Bildschirmtext)

Am 31.12.2001 offiziell eingestellt.

Abb. 5.1 Basisanschluss

BRI (Basic Rate ISDN).
 U_{ko} : Basisanschluss mit zwei Kupferadern, S_0 : Teilnehmeranschluss, B1, B2: Trägerkanal (Bearer), D: Steuerkanal (Daten), NTBA: Netzwerk Terminationspunkt Basis Anschluss

**Temex**

Data over voice-Technik. Dieser Dienst war für automatisierungstechnische Anwendungen gedacht, die keine schnelle Datenübertragung erfordern (Zählerabfrage, Brandschutzmelder usw.). Wurde 1994 wieder eingestellt.

Datex P (Data exchange, paketorientiert)

Datenübertragung mit X.25-Schnittstelle, auch für Automatisierungszwecke. Daten werden als adressierte Pakete über verschiedene Leitungswege übertragen. Übertragungsraten von 50 Bd (Akustikkoppler) bis 64 kbit/s. Bestehende Anschlüsse werden noch unterstützt, aber Neuanschlüsse werden nicht mehr installiert.

Datex L (Data exchange, leitungsorientiert)

Datenübertragung für u. a. Automatisierungszwecke mit X.21-Schnittstelle.

Das Netz sorgt für Aufbau der Verbindung. Übertragungsraten von 0,2 bis 9,6 kbit/s. 1996 von der Deutschen Telekom eingestellt.

ISDN (Integrated Service Digital Network)

Hier sind, wie der Name sagt, alle Weitverkehrsdienele in ein digitales System integriert. Es werden beim Basisanschluss BRI (*Basic Rate ISDN*) zwei Datenkanäle B1 und B2 (Bearer) à 64 kbit/s und ein Steuerkanal D mit 16 kbit/s zeitlich verschachtelt über dieselbe Leitung übertragen (Abb. 5.1).

Die B-Kanäle sind bittransparent und synchron, sodass beliebige Codes verwendet werden können. Die B-Kanäle des BRI können auch zu $2 \times 64 = 128$ kbit/s gebündelt werden. Die Schnittstellen beim Teilnehmer U_{ko} und S_0 sind international genormt (I.430). Alle ISDN-Recommendations der UIT (Union Internationale des Télécommunications, Genf): I.100, I.200, I.300, I.400, I.500, I.600.

Die Pläne, ISDN statt des analogen Telefons beim Teilnehmer zu etablieren, haben sich allerdings nicht voll erfüllt. Das auch deshalb, weil durch DSL (vgl. Abschn. 5.2) eine sehr potente Alternative für die digitale Datenübertragung sich entwickelt hat. So reduziert sich heute ISDN im Wesentlichen zum Dienst für Teilnehmer, die zwei analoge Telefone mit

verschiedenen Rufnummern und/oder zusätzlich digitale Übertragungen (z. B. zwischen zwei PCs) benötigen.

	2008	2009	2010	
Analoges Telefon	21,99	20,33	19,19	Mio. Anschlüsse
ISDN	12,46	11,95	11,65	Mio. Anschlüsse
DSL/Festnetz	20,9	22,4	23,0	Mio. Anschlüsse
VoIP/DSL	2,47	3,98	4,78	Mio. Anschlüsse
DSL über Kabel u. Satellit	–	–	3,2	Mio. Verbindungen

Die vorstehende Übersicht zeigt die Marktentwicklung.

Bis 2019 sollen laut Telekom alle ISDN-Dienste eingestellt und durch VoIP (Voice over IP (Internet Protokoll)) ersetzt werden.

Neben dem oben erwähnten und im Folgenden näher beschriebenen Basisanschluss BRI gibt es noch den Primärmultiplexanschluss PRI (*Primary Rate Interface*). Er bietet 30 mal 64 kbit/s für Daten, einen Steuerkanal mit 64 kbit/s, eine Synchronisationskanal mit 64 kbit/s). Durch Bündelung erreicht man bei PRI 1920 kbit/s).

ISDN arbeitet im Sinne des OSI-Sieben-Schichten-Modells. Es stellt Transportdienste zwischen Endsystemen zur Verfügung. Der D-Kanal und die B-Kanäle laufen parallel. Beide werden über einen gemeinsamen TSAP (*Transport Service Access Point*) angesprochen. OSI-Schicht 4 wird als X.224-Schnittstelle ausgeführt (ISO 8073). Die Trennung von Daten- und Steuerkanälen ist für den Benutzer des Transport-Service transparent. D- und B-Kanäle haben jeweils ein eigenes Protokoll für die Schichten 3 und 2. Der D-Kanal benutzt die Q.930/931-Norm für Schicht 3 und Q.920/921 für Schicht 2.

Für die B-Kanäle wird die Schnittstelle X.25 (ISO 8208) für Schicht 3 angewendet, während X.75, ein symmetrisches HDLC-Protokoll, als Schicht 2 dient (Sicherungsschicht, *High Level Data Link Control*). Die unterste Schicht 1, also die Hardware, ist in I.430 beschrieben. Die Einordnung des ISDN-Frequenzbandes in das gesamte Übertragungsspektrum (incl. POTS, DSL) findet man in Abschn. 5.2.

Die Umsetzung des U_{k0} -Anschlusses des Netzes der DTK an den S_0 -Bus des Teilnehmers erfolgt in der Anschlussdose NTBA, wie Abb. 5.2 zeigt.

Der NTBA transformiert den Bitstrom U_{k0} in die physikalischen Anschlüsse 1a, 1b, 2a, 2b (Abb. 5.3). Man beachte, dass die Anlage, im Gegensatz zum klassischen Analogtelefon, nur mit Netzspeisung funktioniert. Bei Ausfall des 230 V-Netzes hat man über den Notbetrieb die Möglichkeit, die 40 V des Telefonnetzes wenigstens für einen Teilnehmer anzuzapfen (max. 0,4 W).

Will man an den NTBA ein analoges Telefon anschließen, so benötigt man einen D/A-Wandler (mit Netzanschluss). So erhält man zwei Telefone mit verschiedenen Rufnummern. Die dazu notwendigen TAE haben die gewohnten F- und N-codierten Steckkontakte. An den S_0 -Bus direkt angeschlossen werden können – über Western Digital-Stecker – ISDN-Telefone, ISDN-Faxgeräte der Gruppe 4, Kartenlesegeräte, ISDN-Steckkarten für PCs. Mit letzteren beiden ist eine Brücke zur Automatisierungstechnik geschlagen.

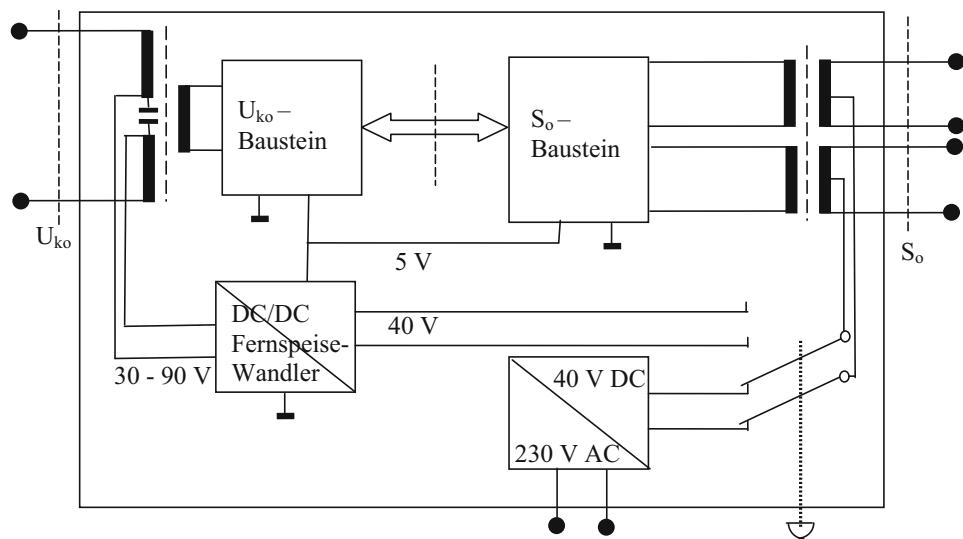


Abb. 5.2 Blockbild der Netzwerk-Terminationspunkt-Basis-Anschlusseinheit (NTBA)

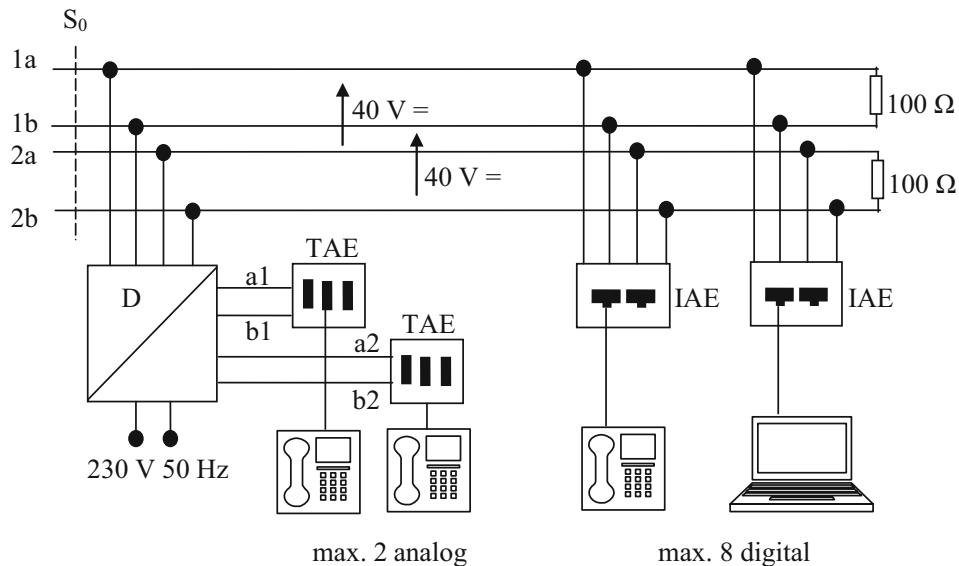


Abb. 5.3 ISDN-Teilnehmer hinter der NTBA-Dose. TAE: Telekommunikationsanschlusseinheit (POTS), IAE: ISDN-Anschlusseinheit (Western-Stecker), D/A: Digital/Analog-Wandler

5.2 DSL – Öffentliches Breitbandnetz

Der Name sagt nicht viel: DSL – Digital Subscriber Line (digitaler Teilnehmeranschluss). Es handelt sich dabei um ein weltweit aufgespanntes Netz über Telefonleitungen mit hohen Übertragungsraten. DSL ist die Hardware, Internet ist die Software. Es hat zunächst keinerlei Bezug zur Automatisierungstechnik. Aber die Verfügbarkeit an jedem Telefonanschluss, die wegen der hohen Teilnehmerzahl niedrigen Hardware-Preise und die niedrigen Netzkosten (Flatrates!) machen DSL auch für Industrieanwendungen interessant.

In Deutschland gibt es zurzeit (März 2011) rund 26,5 Millionen Anschlüsse und eine unüberschaubare Vielfalt von Tarifen. Üblich ist ADSL (asymmetrisches DSL), bei dem die Bitrate downstream (zum Teilnehmer hin) wesentlich höher ist, als upstream (vom Teilnehmer weg). Die Tabelle in Abb. 5.4 zeigt eine Übersicht.

Die Bezeichnung „over ISDN“ bedeutet, dass das ADSL-Frequenzband *oberhalb* des ISDN-Bandes liegt. Es gibt auch „over POTS“. Dadurch wird das ADSL-Band breiter, es wird aber in Deutschland der Einheitlichkeit halber nicht verwendet.

Es gibt auch symmetrische DSL-Verfahren, wie z. B.

- HDSL (High Data DSL) mit Bitraten zwischen 1,54 und 2,04 Mbit/s,
- SHDSL (Symmetric DSL) mit maximal 3 Mbit/s.

a

System	Bandbreite/Hz	max. Übertragungsrate in Bit/s
POTS	300 – 3,4 k	56 k
ISDN	0 – 120 k	2 · 64 k Nutzkanal + 16 k Steuerkanal
ADSL (over ISDN)	138 k – 1,1 M	6 M downstream, 0,5 M upstream
ADSL2+ (over ISDN)	138 k – 2,2 M	20 M downstream, 1 M upstream
VDSL (Glasfaser)	138 k – 12 M	25 M downstream, 5 M upstream

b

Leitungsdämpfung/dB	<55	<46	<43	<39,5	36,5	<32	<18
ÜR in MBit/s	0,384	0,768	1	1,5	2	3	6

Abb. 5.4 Übersicht über Bandbreiten **a** und Übertragungsraten **b** bei DSL. (POTS: Plain Old Telefon Service)

Die in der Tabelle angegebenen Datenraten werden erreicht, wenn auch die „letzte Meile“ zwischen Postvermittlungsstelle und Teilnehmer dies zulässt. Dies ist außerhalb der Städte oft nicht der Fall.

Beispiel

Übliche Telefonkupferleitungen von 0,5 mm Drahtdurchmesser haben typisch eine Dämpfung von $\approx 10 \text{ dB/km}$, bezogen auf 300 kHz. Die Telekom gibt bei einer vorliegenden Dämpfung von 36,5 dB eine mögliche Datenrate von 2 Mb/s an (Abb. 5.4b). Also darf die „letzte Meile“ $36,5 \text{ dB} : 10 \text{ dB/km} = 3,6 \text{ km}$ lang sein. Hätte die Leitung eine Dämpfung von z. B. 55 dB/km, so läge die mögliche Datenrate nur noch bei 384 kbit/s („Dorf-DSL“).

In Abb. 5.5 ist die grundsätzliche Konfiguration einer DSL-Verbindung gezeigt. Die Abkürzungen sind die einschlägig üblichen. Der Access Node erhält das digitale Datenignal über eine virtuelle, paketadressierte Verbindung V-C (C – Central Office, Vermittlungsstelle) aus dem Breitbandnetz WAN (Wide Area Network). Die Daten werden im Modem ATU-C (ADSL Transmission Unit) den HF-Trägern aufmoduliert (downstream) bzw. von den Trägern demoduliert (upstream).

Die Modulation ist ein DMT-Verfahren (Discrete Multitone), verbunden mit QAM (Quadratur-Amplitudenmodulation). Im Splitter werden die mit den Datenbits modulierten Träger mit dem niederfrequenten analogen POTS-Signal (Plain Old Telephone Service) additiv gemischt.

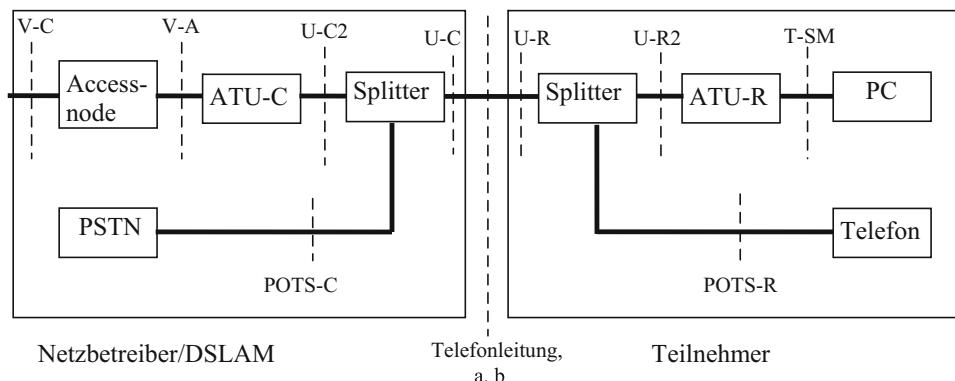


Abb. 5.5 Blockdiagramm einer ADSL-Verbindung. *DSLAM (Access Multiplexer)*: V-C: Verbindung zum Breitbandnetz, V-A: Verbindung zum Modem, U-C2: ADSL-Modemanschluss/Central Office, U-C: Central Office-Schnittstelle zum Teilnehmer, POTS-C: Schnittstelle Plain Old Telephone Service/Central Office, ATU-C: ASDL Transmitter Unit/Central Office, PSTN: Public Switched Network (öffentliches Netz), Splitter: Frequenzweiche 138 kHz; *Teilnehmer*: U-R: Remote Steckverbindung zum Netzbetreiber (z. B. Telekom), U-R2: Remote Modemanschluss, T-SM: USB-Anschluss zum PC, POTS-R: Remote Verbindung zum Plain Old Telephone Service/ISDN, ATU-R: ASDL Unit Remote

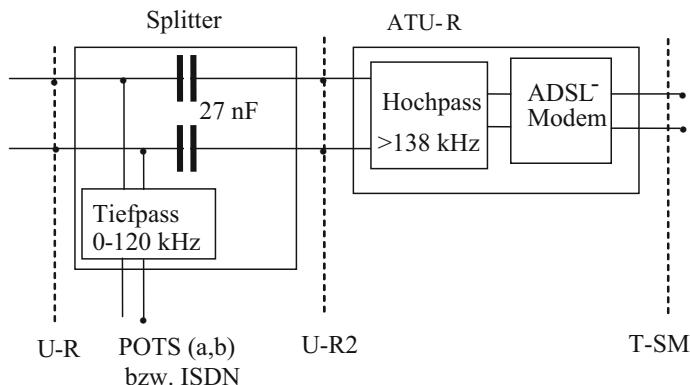


Abb. 5.6 Blockbild eines Splitters mit Modem ATU-R beim Teilnehmer

Der Splitter ist eine passive Frequenzweiche, wie Abb. 5.6 zeigt.

Genau genommen filtert der Splitter nur die Telefonfrequenzen (POTS bzw. ISDN) heraus und überlässt dem Modem die Herausfilterung des ADSL-Signals (HP). Das gesamte Spektrum des POTS/ISDN/ADSL-Frequenzgemisches zeigt Abb. 5.7. Es ist immer gleich, auch wenn der Teilnehmer kein ISDN verwendet.

Die Modulation der Träger ab Band 33 mit den Datenbits bzw. deren Demodulation erfolgt in zwei Schritten, wie folgt:

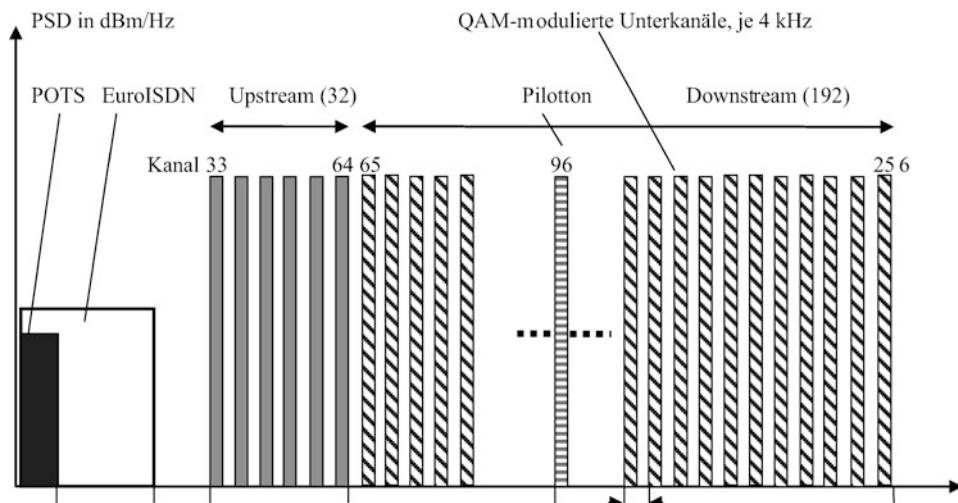


Abb. 5.7 Amplitudenspektrum der ADSL-over-ISDN-Übertragung mittels DMT. Die Bitinformation wird den einzelnen Trägern per QAM (Quadratur AM) aufmoduliert. Der Pilotton dient zur Synchronisation. DMT – Diskrete Multitone Transmission, Multiträgerverfahren, PSD – Power Spectrum Density, POTS – Plain Old Telefon Service, Analogtelefon

DMT (Discrete Multitone Transmission, Übertragung über mehrere einzelne Träger) ist ein Verfahren, bei dem der Bereich von 0 bis 1,040 MHz in 256 Bänder (Kanäle) von je 4,3125 kHz Bandbreite eingeteilt wird:

$$1040 \text{ kHz} / 256 = 4,3125 \text{ kHz} .$$

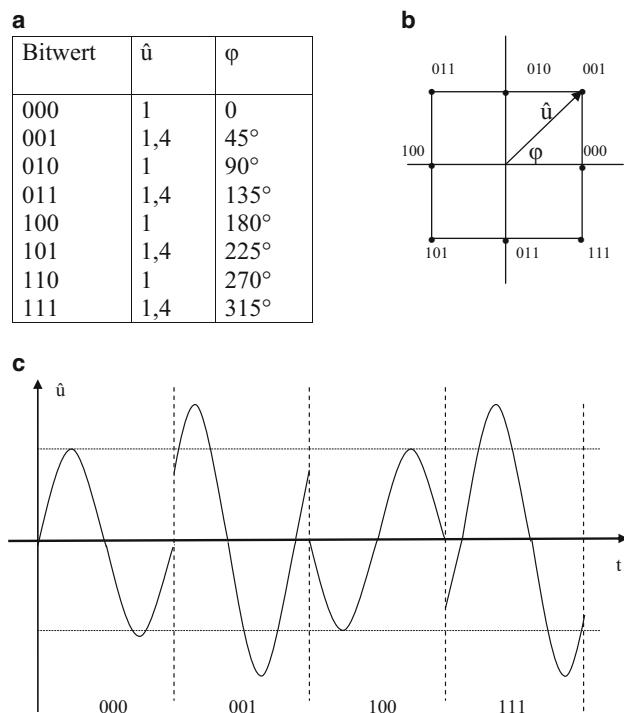
Zwischen den einzelnen 4-kHz-Trägern ist also noch etwas Abstand. Jeder einzelne Träger kann gemäß dem Frequenzgang der realen Telefonleitung mit einer angepassten Bitrate moduliert werden, um Störungsauswirkungen zu minimieren. Für ADSL downstream stehen $256 - 65 - 1$ Pilotton = 190 Träger zur Verfügung. Würde jeder Träger mit 2^{15} QAM (also mit 15 Bit je Träger) moduliert, so könnten maximal

$$15 \text{ bit} \cdot 4 \text{ kHz} \cdot 190 = 11,4 \text{ Mbit/s}$$

übertragen werden. Durch die bitratenadaptierende Störungskompensation sind es in der Praxis deutlich weniger, typisch sind 2 Mbit/s.

QAM (Quadratur Amplitudenmodulation) ist ein aus der Rundfunktechnik bekanntes Verfahren, bei dem Amplituden- und Frequenzmodulation kombiniert werden. Zweck ist, möglichst viele Bits über das 1.04 MHz-Band zu übertragen. QAM ist im Prinzip einfach, die Realisierung ist allerdings nur über einen schnellen Signalprozessor möglich, der u. a.

Abb. 5.8 Beispiel einer Quadraturamplitudenmodulation:
8QAM. **a** Codierungstabelle,
b Zeigerdarstellung, **c** Oszillogramm



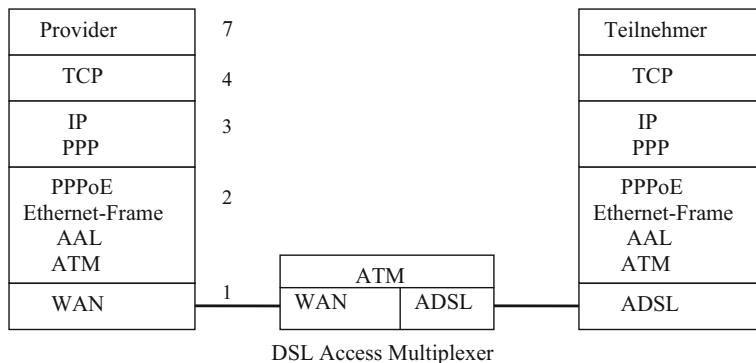


Abb. 5.9 ADSL-Übertragung im ISO/OSI-Schichtenmodell. TCP – Transmission Control Protocol; IP – Internet Protocol; PPP – Point to Point Protocol; PPPoE – PPP over Ethernet; AAL – ATM Adaptation Layer; ATM – Asynchronous Transfer Mode; WAN – Wide Area Network

die schnelle Fourier-Transformation durchzuführen in der Lage sein muss. Es wird je nach Störpegel 2^2 QAM bis 2^{15} QAM angewandt. Abb. 5.8 zeigt beispielhaft die 2^3 QAM. Man kann hier 3 Bit pro Sinusperiode übertragen.

Das *ISO/OSI-Schichtenmodell* eignet sich gut dazu, den Weg der digitalen Information vom Zentralrechner bis hin zum Teilnehmer-PC zu verfolgen (Abb. 5.9).

- In der Transportschicht 4 (TCP, Transmission Control Protocol) werden fehlerfreie logische Kanäle für den Datenaustausch zwischen zwei Teilnehmern festgelegt (Adressierung, Fehlerbehandlung usw.).
- In Netzwerkschicht 3 (IP, Internet Protocol) werden die Daten in Paketen verpackt und das Routing wird festgelegt. Bei DSL ist das PPP (Point to Point Protocol) aktiv, das u. a. die Identifizierung und Bestätigung (Authentifikation) der Datenaustauschpartner regelt.
- In der Netzzugangsschicht 2 werden die aus Schicht 3 übernommenen Daten in versandfertige Telegramme in zwei Stufen verpackt. PPPoE (PPP over Ethernet) packt die Daten in den klassischen Ethernet-Rahmen. In der Unterschicht AAL (ATM Adaptation Layer) wird der bisher synchrone (d. h. lückenlose) Datenstrom von maximal 1500 Bytes/Rahmen in ATM-Zellen fragmentiert, die so adressiert werden, dass sie beim späteren Empfang (rechter Block unseres Bildes) wieder assembliert werden können.
- Die physikalische Schicht 1 besorgt die Verbindung vom Provider über WAN zum DSLAM (DSL-Access Multiplexer) in der Vermittlungsstelle. Das DSLAM hat die Funktion einer Bridge. Von dort gehen die Daten über ADSL zum Teilnehmer, wo sie die OSI-Schichten von unten nach oben durchlaufen. Der Ablauf in der Gegenrichtung ist prinzipiell derselbe.

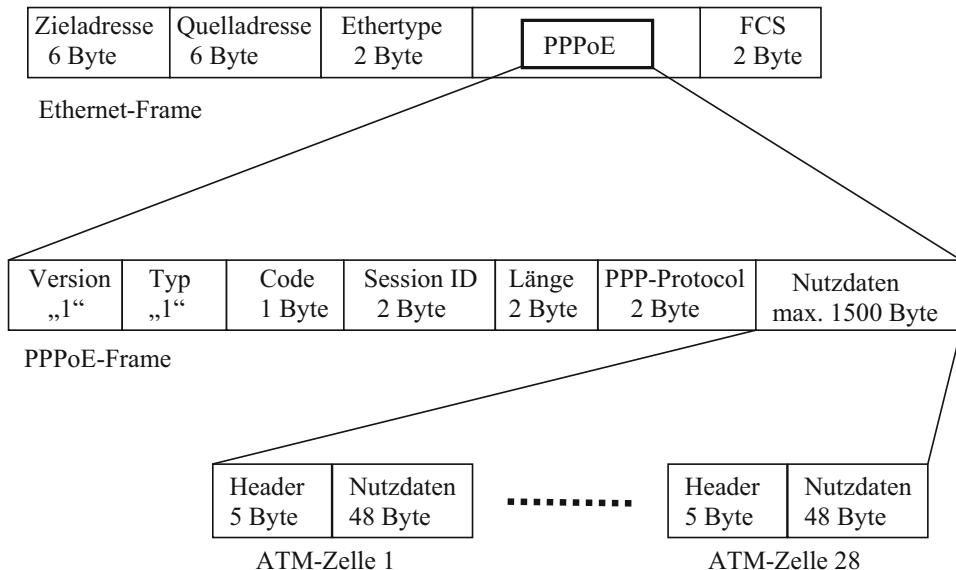


Abb. 5.10 Die Verschachtelung der Telegramme bei ADSL

Im Feld *Ethertype* des Ethernet Frame (Abb. 5.10) wird signalisiert, ob man sich im Zustand des Verbindungsaufbaus befindet (Discovery stage: 8853h) oder im Dauerzustand eines laufenden Datenaustausches (Session stage: 8864h).

Im *Codefield* des PPPoE-Frame werden die 5 Schritte eines Verbindungsau- und -abbaus (PPPoE Action Discovery) signalisiert:

- Initiation (PADI): Anfrage des Teilnehmers X an alle (broadcast),
- Offer (PADO): Antworten diverser DSLAMs an X,
- Request (PADR): X wählt seinen gewünschten DSLAM,
- Session confirmation (PADS): Der ausgewählte DSLAM vergibt die für diese Session gültige Identifikationsnummer Session-ID und beginnt die Übertragung,
- Terminate (PADT): X oder DSLAM signalisiert das Ende der Session.

Im Feld *Session ID* steht bei Broadcast („an alle“) der Wert „0“, bei PADS wird ein fester Wert vom DSLAM festgelegt, der während der ganzen Übertragung stehen bleibt.

Mittels Abb. 5.10 ist die Verschachtelung der Telegramme leicht nachzuverfolgen.

Literatur

Literatur zu Abschn. 5.1

1. ISDN-Recommendations I.100ff: www.uit.int/rec/T-REC-I/en
2. Jahresbericht 2010 der Bundesnetzagentur: www.Bundesnetzagentur.de/Telekommunikation/Marktbeobachtung
3. www.wikipedia.org/wiki/ISDN
4. Skript Prof. Dr. Winkler: Vermittlungstechnik 3: ISDN. www.hs-mittweida.de
5. Skript Prof. Plate: ISDN Grundlagen. www.netzmafia.de/skripten/telefon/isdn-a.html (FH München)
6. Jansen, H.: Telekommunikation mit ISDN und ADSL. Haan-Gruiten, 2006
7. Frey, H.: ISDN, DSL-, WLAN-, und Internet-Telefonie. München, 2008

Literatur zu Abschn. 5.2

8. www.wikipedia.org/wiki/DigitalSubscriberLine
9. ADSL-Recommendation G992: www.itu.int/rec/T-REC-G/en
10. www.wikipedia.org/wiki/DiscreteMultitone
11. www.wikipedia.org/wiki/Quadraturamplitudenmodulation
12. Jansen, H.: Telekommunikation mit ISDN und ASDL. Haan-Gruiten, 2006
13. www.wikipedia.org/wiki/AsynchronousTransferMode
14. www.wikipedia.org/wiki/PPPoE
15. A Method for Transmitting PPP over Ethernet, 1999. www.tools.ietf.org/html/rfc2516

Installationsbeispiele aus der Bus-Praxis

6

6.1 Verbindung von Feldgeräten über PROFIBUS und OPC mit Anwendersoftware

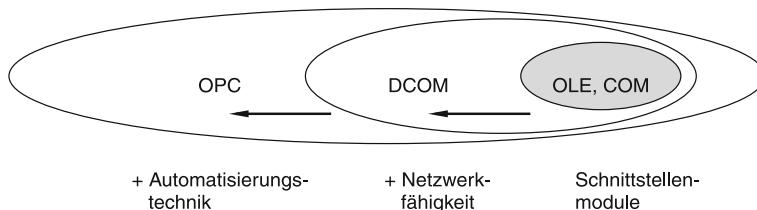
6.1.1 Kurze Einführung in OPC

Die Aufgabe ist es, zwei Windows-Programme (z. B. LOOKOUT und PROFIBUS Master-sw) derart zu verbinden, dass sie Daten austauschen können. Dazu benötigen sie eine sw-Schnittstelle.

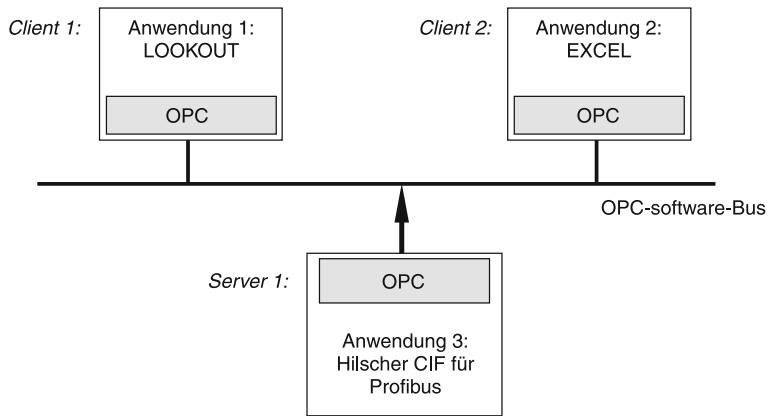
Lösung 1: DDE (Dynamic Data Exchange): Standard-Lösung, aber für manche Steuerungsfälle zu langsam. Jede Schnittstelle enthält ihr individuelles DDE-Modul, verknüpft mit einem individuellen DDL-Modul.

Lösung 2: OPC (OLE for Process Control, OLE – Object Linking and Embedding): Universelle Schnittstelle für industrielle Steuerungen, „Software-Bus“.

Nachfolgende Skizze zeigt die Windows-spezifischen Akronyme im Zusammenhang (COM – Component Object Model, DCOM – Distributed COM):



Bezogen auf die Sichtweise der Bussysteme ist der Sachverhalt auch wie folgt darstellbar:



Der OPC-Client entspricht dem Master, der OPC-Server entspricht dem Slave der Hardware-Buswelt. Die OPC-Schnittstellen sind zur „Busseite“ hin alle gleich.

6.1.2 Die Aufgabe: PROFIBUS an Visualisierungssoftware

Hardwaremäßig ist das Problem gelöst: Der Bus-Master steckt als Karte im PC oder in der SPS. Die Software-Anbindung an das Anwenderprogramm (z. B. eine Visualisierung) geschieht entweder über ein individuelles DDE-sw-Interface oder über die universelle OPC-sw-Schnittstelle. Der PROFIBUS ist der Server, die Anwendung der Client.

Nachfolgend wird als Anwendungsbeispiel der PROFIBUS-OPC-Server von Hilscher mit der Prozessvisualisierung LOOKOUT von National Instruments, die über einen OPC-Client verfügt, verbunden. Als PROFIBUS-Slave dient eine digitale I/O-Peripherieeinheit ET 200. Wir steuern 8 LEDs als Ausgänge an (= 8 Bit-Analogwert) und betätigen mit einem Schalter einen Eingang (Abb. 6.1).

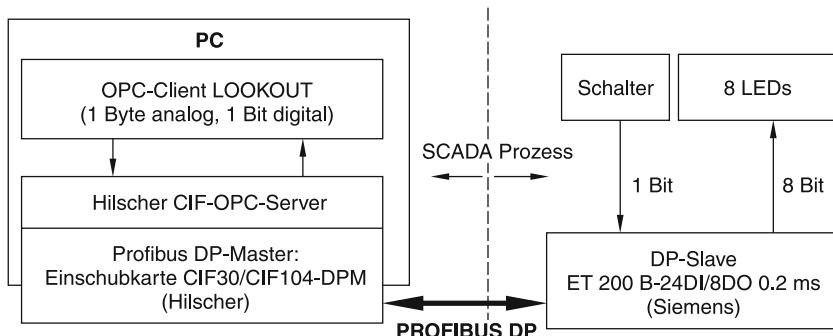


Abb. 6.1 Konfiguration unseres Beispiels

6.1.3 Konfiguration des PROFIBUS

Der PROFIBUS DP-Master benötigt Informationen über den Bus und den beteiligten Slave. Mit diesen Informationen wird die PROFIBUS-Karte geladen (konfiguriert) und ist dann bereit für einen selbstständigen Datenaustausch mit den Slave.

Als Konfigurationswerkzeug verwenden wir die Software „SyCon SystemConfigurator“, die mit der PROFIBUS-Karte CIF30/CIF104-DPM (Hilscher GmbH) geliefert wird.

Jeder PROFIBUS-DP-Gerätehersteller definiert die Eigenschaften seines Gerätes in einer so genannten Gerätetammdatei (GSD-Datei). Diese GSD bildet die Grundlage der Konfiguration. Die GSD für den Slave ET 200 muss vor dem Start des SyCon System Configurators im Verzeichnis `../SysCON/..GSD` abgelegt werden.

Nach dem Start des SyCon SystemConfigurators wählen wir *Datei → Neu*. SyCon startet im Konfigurationsmodus und öffnet ein Fenster mit einer Busleitung (Abb. 6.2). Dort fügen wir den Master CIF30/CIF104-DPM ein, indem wir den Menüpunkt *Einfügen – Master...* oder das entsprechende Ikon anklicken. Der Cursor zeigt nun einen Pfeil mit einem M. Klicken wir die Busleitung an, so öffnet sich ein Menü *Master einfügen*. Nach der Auswahl des Masters und der Eingabe der Stationsadresse 1 wird der Master hinzugefügt.

Den Slave fügen wir hinzu, indem wir den Menüpunkt *Einfügen – Slave ...* oder das entsprechende Ikon anklicken. Der Cursor zeigt nun einen Pfeil mit einem S. Klicken wir auf die Busleitung, wird ein Menü mit den verfügbaren Slaves geöffnet. Nach Auswahl des Slaves und Eingabe der Stationsadresse 3 wird der Slave hinzugefügt.

Zum Abschluss laden wir den Menüpunkt *Online-Download...* die Konfiguration, wie wir sie in Abb. 6.2 festgelegt haben, auf die Station.

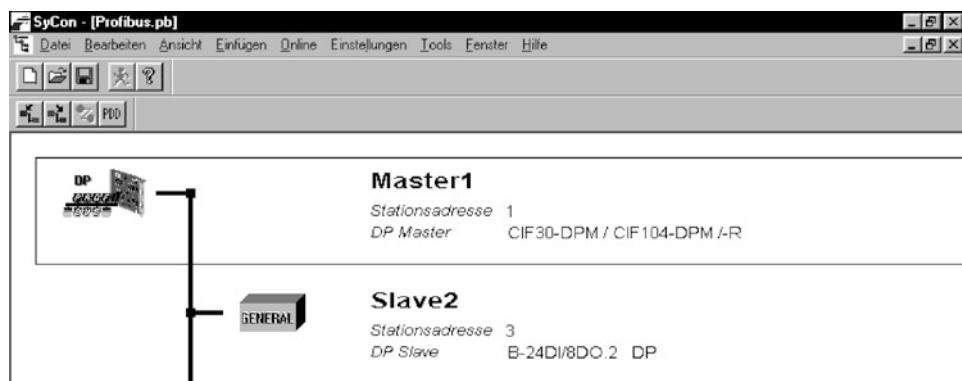


Abb. 6.2 Konfiguration des PROFIBUS DP

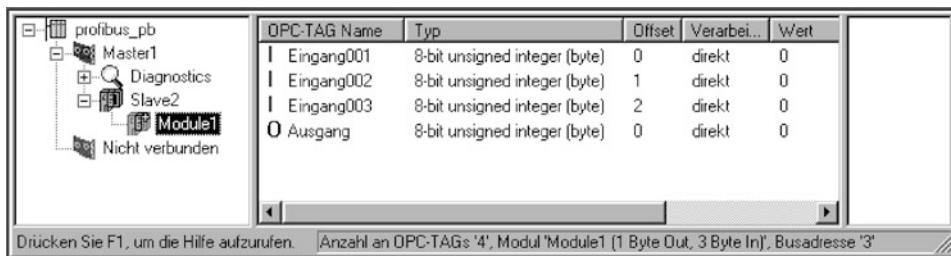


Abb. 6.3 SyCon-Netzwerkdarstellung

6.1.4 Konfiguration des OPC-Servers

Mit Hilfe des OPC-Server-Konfigurators kann der Anwender seine Hardwarekonfiguration in den OPC_Server abbilden.

Dazu legen wir das PROFIBUS-Projekt in den Hintergrund und erhalten die SyCon-Netzwerkdarstellung (Abb. 6.3).

In der Netzwerkdarstellung (Abb. 6.3, links) wird vom Slave2 das Modul1 aufgesucht.

Durch Doppelklick auf *Modul1* können wir die Eingangs- und Ausgangs-TAGs editieren.

Im Ausgangs-TAG Fenster (Abb. 6.4a) wählen wir im Feld *Byte* den OPC-TAG Namen „AD“. Entsprechend im Eingangs-TAG Fenster (Abb. 6.4b) im Feld *Byte* „Eingang 001“ und im Feld *Bit* den OPC-TAG Namen „Bit1“ (d. h.: Im 1. Byte das Bit 1).

Das gesamte Projekt wird unter den Namen „Profibus.pb“ abgespeichert.

Die damit festgelegte Netzwerkdarstellung wird in Abb. 6.5 dargestellt. Damit ist Busseitig der „OPC-Stecker“ betriebsbereit. Nunmehr muss die „OPC-Steckdose“ auf der SCADA-(LOOKOUT-)Seite konstruiert werden.

6.1.5 SCADA-Projekt und OPC-Konfiguration

Zur Erstellung des SCADA-Projektes (Supervision, Control And DataAcquisition) wird die Visualisierungssoftware LOOKOUT 4.01 von National Instruments verwendet.

Zunächst wird ein einfaches Projekt in Lookout erzeugt (Abb. 6.6): Mit dem Schieberegler kann der (beliebig gewählte) Analogwert zwischen 0...24 eingestellt werden mit einer Auflösung von 8 Bit. Dieser 1 Byte-Wert soll über den OPC-Clienten von Lookout an den OPC-Server von SyCon übergeben werden: Die 8 LEDs des Slave ET 200 zeigen den eingestellten Analogwert an. Die Signallampe repräsentiert den Zustand des Schalters am Slave ET 200.

Als nächstes wird in Lookout unter dem Menüpunkt *Object – Create* der passende OPC-Client angelegt. Hierzu wird unter der Kategorie *Drivers* das Modul *OPCClient* ausgewählt (Abb. 6.7).

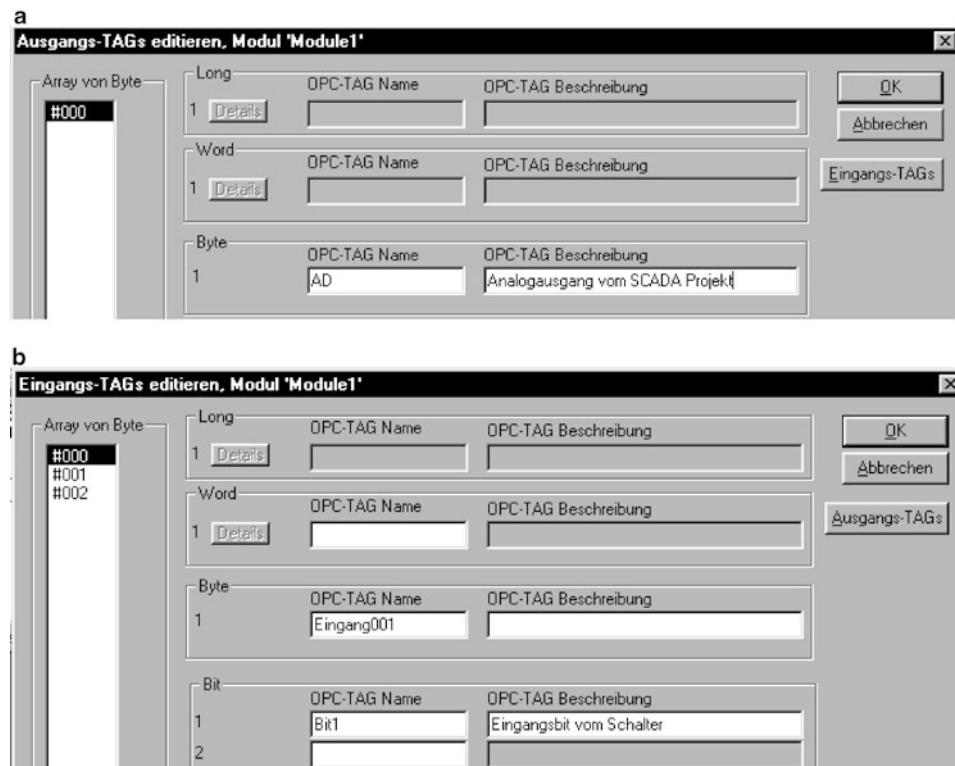


Abb. 6.4 a Ausgangs-TAG-Fenster, b Eingangs-TAG-Fenster

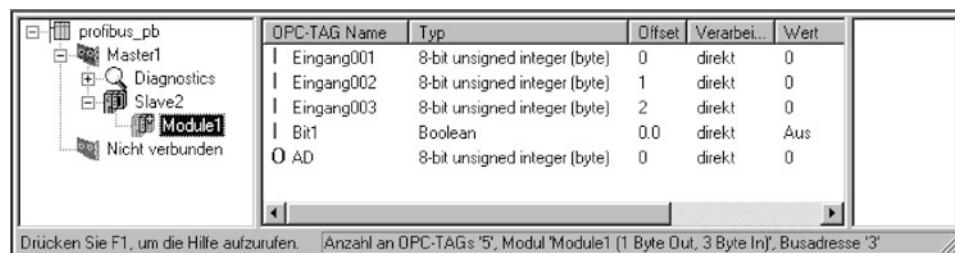


Abb. 6.5 Das fertig konfigurierte Netzwerk

Als Name kann man „OPCPROFIBUS“ eintragen. Im Fenster *OPC Server Settings* ist unter *Server Name* der OPC-Server „HilscherGmbH.CifOpcServer“ anzuwählen. Nach der Bestätigung mit *OK* ist der OPC-Client angelegt.

Um mit dem OPC-Server des PROFIBUS-Masters kommunizieren zu können, müssen als nächstes die verschiedenen OPC-Items als Variable deklariert werden. Das entsprechende Fenster (Abb. 6.8) findet man unter dem Menüpunkt *Objekt – Edit Database* nach Anwahl des Objektes *OPCProfibus*.

Abb. 6.6 Lookout-Projekt:
Visualisierung 1 Byte aus/1 Bit
ein



Im Fenster *OPCPROFIBUS database* findet man in der Liste *Native members* die verschiedenen Signale, die als Variable genutzt werden können. Der Eintrag *L0 – L99999* steht für logische Signale, also einzelne Bits, während *N0 – N99999* die numerischen Variablen verwaltet. In unserer Anwendung werden wir als logische Variable das „Bit1“ (Eingang) und die numerische Variable „AD“ (Ausgang) deklarieren.

Im Feld *Alias (optional)* kann dem Signal ein Name zugewiesen werden.

Der Pfad zum eigentlichen OPC-Server-Signal wird unter der Rubrik *Description* angegeben. Zur Vereinfachung kann hierfür die dortige Taste betätigt werden, um in einem weiteren Fenster die vom Server zur Verfügung gestellten Signale auszuwählen.

Hier wählen wir:

Signalrichtung	Alias	Member	Description
Ausgang	AD	N0	~~PROFIBUS_pb.Master1.Slave1.Module.AD
Eingang	Bit1	L0	~~PROFIBUS_pb.Master1.Slave1.Module.Bit1

Abb. 6.7 OPC-Client in Lookout anlegen

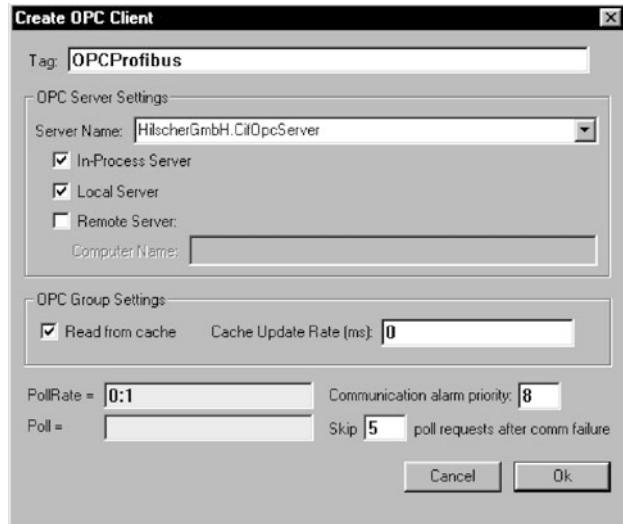




Abb. 6.8 Variable des OPC-Servers deklarieren

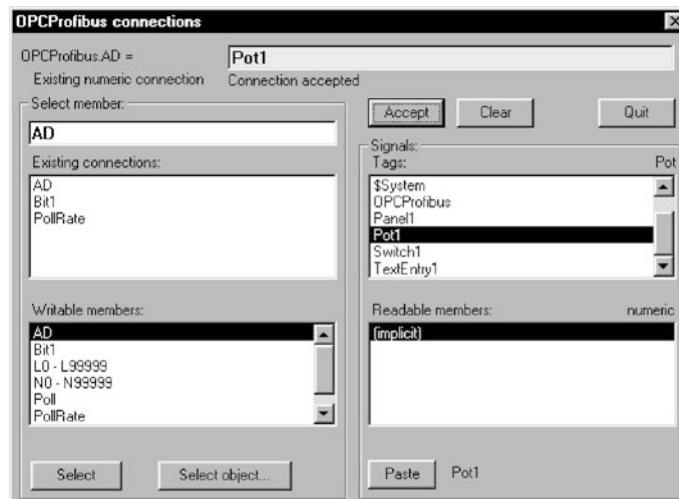


Abb. 6.9 Verbinden von OPC-Signalen mit der Visualisierung

Als letzter Schritt werden die Ein-/Ausgänge am PROFIBUS-Slave zur Visualisierung hin geschaltet. Dies geschieht bei Lookout über den Menüpunkt *Object – Edit Connections*. Nachdem man das Objekt *OPCProfbus* ausgewählt hat, erscheint das Fenster *OPCProfbus connections*, in dem die Verbindungen hergestellt werden können (Abb. 6.9).

Um eine Verbindung zu schaffen, muss zunächst ein Signal aus der Tabelle *Writeable members*, zum Beispiel „AD“, ausgewählt werden. Die Taste *Select* zeigt die Auswahl. Anschließend wird als Tag *Pot1* gewählt. Mit der Schaltfläche *Accept* wird die Eingabe übernommen und in die Liste *Existing connections* übernommen.

Dieser Vorgang wird mit dem Signal „Bit1“ und „Switch1“ wiederholt. Damit ist die Konfiguration abgeschlossen.

Beim Neustart des PC ist zunächst zu prüfen, ob der Server noch für unser Projekt konfiguriert ist (SyCon). Startet man jetzt das SCADA-Projekt unter Lookout, so müssen die 8 LEDs am Slave ET 200 die aktuelle Position des Schiebereglers anzeigen. Umgekehrt wird die Schalterstellung an der ET 200 durch Lookout angezeigt (vgl. Abb. 6.1). Der „Softwarebus“ OPC arbeitet mit dem Hardwarebus PROFIBUS zusammen!

6.2 Prozesssteuerung über das Internet-Netzwerk

6.2.1 Die Aufgabe

Es ist heute üblich, Prozesse von der Ferne zu überwachen und evtl. sogar zu beeinflussen. Der Prozess ist dabei mit der Beobachtungsstation z. B. via Internet verbunden (Abb. 6.10).

Der folgende Abschnitt beschreibt eine einfache Prozesssteuerung über das Internet. Die Erstellung der Applikation (= Visualisierung des Prozesses) erfolgt mit „LabView“ von National Instruments. LabView verfügt über ein Zusatz-Tool, welches die Anbindung an das Internet ermöglicht (Internet-Tool mit Internet-Server, gehört nicht zur Grundausstattung von LabView).

Aufgabenstellung (Abb. 6.11): In einem y/t -Kurvengraph werden die Momentanwerte eines numerischen Anzeigeelementes dargestellt. Das Abbild dieses „Prozesses“ wird mit

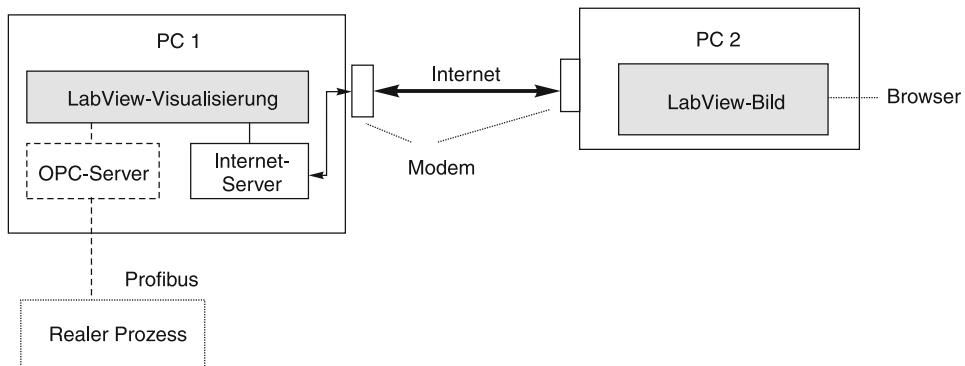
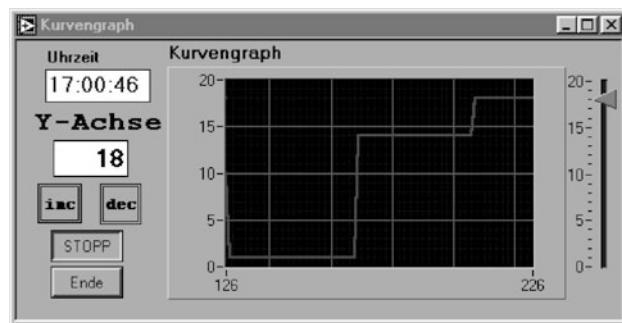


Abb. 6.10 Die beschriebene Internet-Übertragung. (Die gestrichelte Anbindung ist nicht beschrieben)

Abb. 6.11 Der Prozess „Kurvengraph“ in LabView

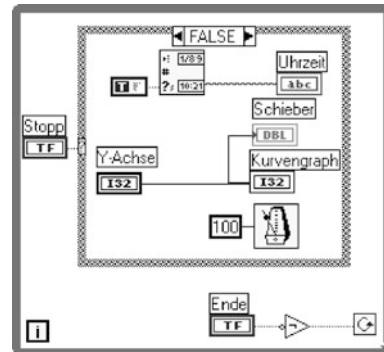


einem Internet-Browser (Netscape oder Explorer) auf der entfernten Station dargestellt und ständig aktualisiert. Die Einstellung der Momentanwerte erfolgt im Browserbild über die Felder „inc“ bzw. „dec“. Durch eine Stopp-Taste wird der Prozess angehalten. Die Ende-Taste beendet die Übertragung.

6.2.2 Erstellung der LabView-Applikation

Die Visualisierungs- (bzw. SCADA-)Software LabView wird mit einer graphischen „Sprache“ programmiert. Das graphische Programm unserer kleinen Anwendung „Kurvengraph“ wird in der folgenden Abb. 6.12 dargestellt.

Abb. 6.12 LabView-Programm von „Kurvengraph“



Zur Erläuterung des LabView-Programms beschreiben wir nachfolgend die einzelnen Funktionselemente:

	String-Ausgabe		numerische Ausgabe (double format)		numerische Ausgabe (integer, 32 Bit)			
	Auszabe der Uhrzeit (um Laufzeiten im Internet zu messen)							
	Zeitgeber zum Ausführen der While Schleife (100 ms)							
	Negator							
	While-Schleife		Schleifenzähler					
	Case-Struktur (TRUE,FALSE)							
	Boolesche Variable True/False							

Innerhalb der While-Schleife wird dem Kurvengraph alle 100 ms der Wert vom Anzeigeelement mit dem Label „Y-Achse“ übergeben. Dies gilt für den Fall, dass die Taste Stopp nicht gedrückt ist, d. h., der Wert FALSE wird der Case-Struktur übergeben.

Bei gedrückter Stopp-Taste erfolgt eine Übergabe von TRUE, d. h., der Prozess wird nicht ausgeführt (Leeranweisung im Programm).

Für tiefergehende Information über die graphische Programmierung sei auf die Unterlagen von LabView verwiesen.

6.2.3 Internetanbindung

Die Prozesssteuerung über das Internet erfordert die Erstellung einer oder mehrerer HTML-Seiten (HyperText Markup Language).

In unserem Beispiel erstellen wir zunächst die Startseite (Abb. 6.13). Sie dient zum Ausführen der Applikation mit entsprechender Bedienmöglichkeit. (Lesehinweis: Die HTML-Standard-Anweisungen sind in normaler Schrift angegeben, die speziellen LabView-Internet-Toolkit-Befehle für die Applikation sind fett gedruckt, Kommentare kursiv.)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>

<HEAD>                                         Parameter für die HTML-Seite
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-
        8859-1">
    <LINK rel=STYLESHEET href=".../itk.css" Type="text/css">

    <META NAME="Author" Content="Internet Toolkit">
    <TITLE>Startseite</TITLE>
</HEAD>

<BODY background="fbe_hint.gif">                  Gestaltung der Seite
    <TABLE WIDTH=100%>
        <TR><TD WIDTH=10%></TD>
        <TD WIDTH=90%><FONT SIZE=5 FACE="Arial"><B>Fachhochschule Frankfurt
            am Main</B><BR>University of Applied Sciences<BR>Fachbereich
            Elektrotechnik<BR>Technische Informatik</FONT></TD>
    </TR></TABLE>
    
    <TABLE WIDTH=100%>

<P> Prozesssteuerung starten inclusive Bedienung
    <A HREF="/cgi-bin/victrl.vi?..internet.vi&command=run&open&redirect=
        /HTML_S Seiten/test.htm">
        Start</A>. </P> Öffnen der Applikation „internet.vi“ und Ausführen

<P> Prozesssteuerung starten (Anzeige der Koordinaten in der Anwendung)
    <A HREF="/cgi-bin/victrl.vi?..internet.vi&command=run&open&redirect=
        /HTML_S Seiten/testkoor.htm">
        Start</A> </P>
        <HR>
        <A HREF="">
            Zurück zur Startseite des
            Fachbereichs Elektrotechnik</A>
        <HR>
</BODY>

</HTML>

```

Abb. 6.13 HTML-Startseite: „index.htm“

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>

<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-1">
  <LINK rel=STYLESCHEET href="../../itk.css" Type="text/css">

  <META NAME="Author" Content="Internet Toolkit">
  <TITLE>G Prozesssteuerung</TITLE>
</HEAD>

<BODY>
<H1>Prozesssteuerung</H1>
<P><A HREF="/"><IMG SRC="/.snap?internet.vi BORDER="0" ISMAP></A><P>
Abbildung der Applikation „internet.vi“ im Browser

<P>Durch Bewegen der Maus im Fenster der Abbildung werden die Koordinaten</P>
<P>der jeweiligen Bedienelemente erfasst.</P>

<P> Prozesssteuerung beenden und zurück zur Startseite
<A HREF="/cgi-in/victrl.vi?..internet.vi&command=close&redirect
  =/index.htm">Ende</A><P>
Applikation „internet.vi“ schließen
</BODY>

</HTML>

```

Abb. 6.14 HTML-„Werkzeug“-Seite 2: Prozessteuerung ohne Bedienung „testkoor.htm“

Die Bedienfelder „inc, dec, Stopp, Ende“ des Browserbildes sind Bestandteil der LabView-Applikation, werden aber über die HTML-Seite 3 aktiviert. Ihre Koordinaten werden mit der „Werkzeug“-Seite 2 (Abb. 6.14) und dem Programm-Modul „Anzeige der Koordinaten in der Anwendung“ im Browserbild angezeigt (gestrichelter Block in Abb. 6.13, kann bei der Endanwendung entfallen), wenn man mit der Maus die Eckpunkte links oben und rechts unten anfährt. Diese Koordinaten werden in der HTML-Seite 3 (Abb. 6.15) unter dem Schlüsselwort „coords“ verwendet. Damit erscheint dann die berühmte „Internet-Hand“ im Browser-Bild. Die Verknüpfung der Labels „Y-Achse, Stopp, Ende“ mit der LabView-Applikation besorgt das Internet-Toolkit automatisch.

Die „Arbeits“-Seite 3, test.htm, mit der man die Applikation abbildet und die die Koordinatenfelder als Bedienfelder anzeigt, ist in Abb. 6.15 zu sehen.

Als Nächstes legen wir das „DocumentRoot“-Verzeichnis http (Hypertext Transport Protocol) an, in dem sowohl unsere 3 HTML-Seiten als auch die verwendeten Standard-Dateien liegen. Darauf greift das Internet-Toolkit zu.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>

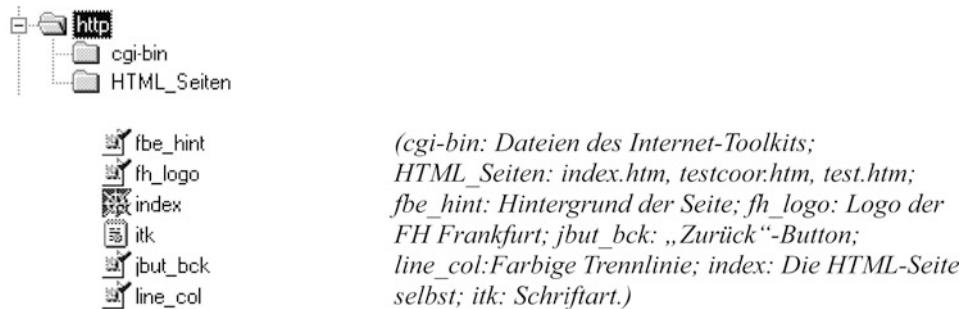
<HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-
        8859-1">
    <LINK rel=STYLESHEET href="../../itk.css" Type="text/css">
    <META NAME="Author" Content="Internet Toolkit">
    <TITLE>G Prozesssteuerung mit Bedienung</TITLE>
</HEAD>

<BODY>
<H1>Prozesssteuerung</H1>
<P>
    <IMG SRC="/.monitor?internet.vi&refresh=1" ALIGN="BOTTOM"
        BORDER="1" USEMAP="#panel" ISMAP></P>
        Abbildung der Applikation „internet.vi“ im Browser und  
Erneuerung des Bildes jede Sekunde (1...60s möglich).
<P>Durch Anklicken des Feldes <B>inc</B> wird der Wert der Y-Achse um 1 erhöht.</P>
<P>Durch Anklicken des Feldes <B>dec</B> wird der Wert der Y-Achse um 1  
erniedrigt.</P>
<P>Die Taste <B>Stopp</B> unterbricht den Prozess.</P>
<P>Die Taste <B>Ende</B> beendet die Anwendung und es erfolgt eine Rückkehr zur  
Startseite.</P>
    <MAP Name="panel">
        <AREA Shape="Rect" coords = "19,110,49,136"
            HREF="/cgi-bin/victrl.vi?internet.vi&SetInt,Y-Achse,inc=1&redirect">
            Incrementieren der Integervariable „Y_Achse“ (Label in der  
Applikation „internet.vi“) mit der Schrittweite =1.
        <AREA Shape="Rect" coords = "63,113,93,136"
            HREF="/cgi-bin/victrl.vi?internet.vi&SetInt,Y-Achse,dec=1&redirect">
            Decrementieren der Integervariable „Y_Achse“ (Label in der  
Applikation „internet.vi“) mit der Schrittweite =1.
        <AREA Shape="Rect" coords = "29,149,76,166"
            HREF="/cgi-bin/victrl.vi?internet.vi&SetBool,Stopp,toggle&redirect">
            Setzen der booleschen Variable „Stopp“ (Label in der Applikation „internet.vi“)
        <AREA Shape="Rect" coords = "30,178,78,194"
            HREF="/cgbin/victrl.vi?internet.vi&SetBool,Ende&command=close&redirect
            =/index.htm">
            Setzen der booleschen Variable „Ende“ (Label in der Applikation „internet.vi“)
    </MAP>
</BODY>

</HTML>

```

Abb. 6.15 HTML-Seite 3, „test.htm“: Prozess-Steuerung mit Bedienung



Bei der Einwahl eines Benutzers in unseren Server wird die Seite „index.htm“ aufgerufen.

Die weiteren Seiten befinden sich im Unterverzeichnis *HTML_Seiten*.

Im Unterverzeichnis *cgi-bin* werden alle Dateien aus dem gleichnamigen Verzeichnis des Internet-Toolkits abgelegt.

6.2.4 Die Konfiguration des HTTP-Servers

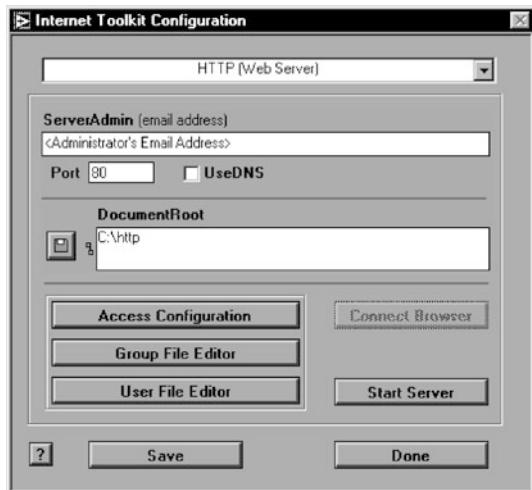
Die Konfiguration des Servers erfolgt im entsprechenden Fenster des Programms LabView:

Menü → Projekt → Internet Toolkit → Internet Toolkit Configuration ...

In diesem Fenster (Abb. 6.16) wird im Feld „DocumentRoot“ das Verzeichnis C:\http eingetragen, in dem u. a. unsere Start-HTML-Seite „index.htm“ abgelegt ist.

Danach erfolgt der Start des HTML-Servers durch Mausklick.

Abb. 6.16 Das Fenster „Internet Toolkit“ von LabView



Wählen wir nun z. B. die Adresse: <http://tilab16.fbe.fh-frankfurt.de> an, so erscheint Abb. 6.10 in unserem Browser-Fenster. Von da aus steuern wir den Prozess, also „Kurvengraph“, der auf dem Rechner „tilab16“ der Fachhochschule Frankfurt läuft. Man beachte, dass der Browser des Beobachtungsrechners das Abbild des Prozesses zeigt und die Aktivierung der Elemente erlaubt, aber keine Modifikation der LabView-Applikation.

6.3 Konfiguration AS-i/Interbus-Gateway an Interbus

6.3.1 Aufbau der Bus-Systeme

Bei der Konfiguration Interbus und AS-i-Gateway handelt es sich um eine Kombination zweier Bussysteme, des Interbus und des AS-i-Bus. Auf die Grundlagen des Interbusses sowie des AS-i-Busses wird an dieser Stelle nicht eingegangen.

In Abb. 6.17 ist der schematische Aufbau eines solchen Systems gezeigt. Aus dem Bild geht hervor, dass AS-i dem Interbus untergeordnet ist. Des Weiteren sei noch erwähnt, dass der Interbus-Slave mit (hier) Adresse 2 und der AS-i-Master in *einem* Gerät (Interbus /AS-i Gateway) enthalten sind.

In Abb. 6.18 sind die einzelnen Module spezifiziert gezeigt.

Die Anlage wird von einer SPS Modicon TSX Compact (Schneider Electric) gesteuert.

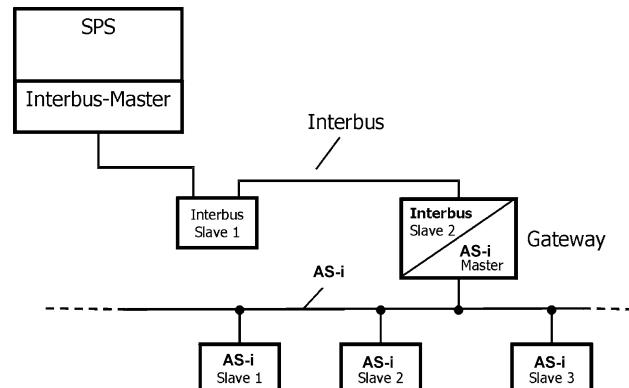
Als Interbus-Master dient der Einschub Modicon BKF 201, (Schneider Electric). Für den Betrieb dieses Masters ist die Programmiersoftware Concept ab Version 2.1 erforderlich.

Das Modul BKF 201 wird über den RS 232-Port der SPS angesprochen.

Außerdem haben wir zwei E/A Module Modicon DAP 212 (Schneider Electric) in Gebrauch.

Als Interbus Slave 1 dient das Modul 170 BDM 346 00 (Schneider Electric).

Abb. 6.17 Hardware-Aufbau
(Übersicht)



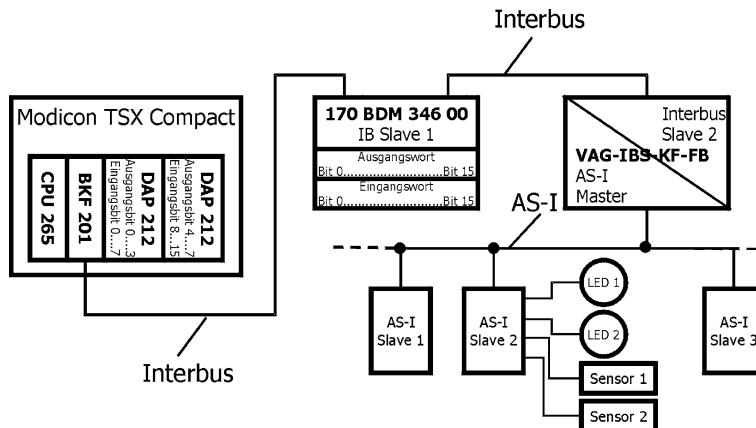


Abb. 6.18 Hardware-Aufbau mit Modulen

Als AS-i/Interbus-Gateway dient der Baustein VAG-IBS-KF-FB (Pepperl+Fuchs). Dieses Gateway ist für den Interbus der Slave mit Adresse 2.

Die AS-i-Slaves 1 und 3 sind induktive Näherungsschalter (Pepperl+Fuchs).

Der AS-i-Slave 2 ist ein 2E/2A-Modul (Pepperl+Fuchs), das zwei induktive Schalter als Eingänge und 2 LEDs als Ausgänge hat.

Das Ausgangswort des Interbus an den Interbus-Slave 1 wird bitweise mit den LEDs angezeigt, die direkt am Slave 1 angebracht sind. Das Eingangswort kann bitweise durch Schalter, die ebenfalls am Slave 1 angebracht sind, gesetzt werden. Die AS-i-Slaves 1 und 3 sind busfähige Sensoren, die direkt mit dem AS-i verbunden sind. Der AS-i-Slave 2 übernimmt die Signale der Sensoren 1 und 2, die LEDs 1 und 2 sind Aktoren dieses Slaves.

Der Interbus arbeitet mit RS 485 im Ring. Der Sub-D-Stecker ist belegt wie in Abb. 6.19 gezeigt.

Die Tabelle zeigt die Kontaktbelegung des Interbus-Steckers:

Kontakt	Signal	Funktion
1	DO	Daten senden
2	DI	Daten empfangen
3	GND	Ground
4	Nicht belegt	
5	5 V aus	Für Lichtwellenleiter-Schnittstelle
6	/DO	Negierte Daten senden
7	/DI	Negierte Daten empfangen
8	Nicht belegt	
9	Nicht belegt	

Abb. 6.19 Interbus-Steckerkontaktbelegung

6.3.2 Konfiguration des AS-i

Das Interbus-Master-Modul BKF 201 erlaubt es nicht, den AS-i von der Interbus-Oberfläche aus zu konfigurieren, obwohl das Gateway VAG-IBS, das gleichzeitig AS-i-Master ist, dies erlauben würde. Wir helfen uns, indem wir den AS-i mittels Drucktasten und LEDs am Gateway selbst „per Hand“ konfigurieren: Slave 1, 2 und 3. Nach der Konfiguration schalten wir den AS-i-Master vom Projektiermodus in den geschützten Betriebsmodus um.

Verwendet man den Interbus-Master von Phoenix Contact, der als PC-Einschubkarte angeboten wird, so kann man von der Interbus-Oberfläche aus (CMD tools) auch AS-i konfigurieren.

6.3.3 Kommunikation des AS-i/Interbus-Gateway mit dem Interbus

Im Folgenden wird erklärt, wie die Informationen des AS-i/Interbus-Gateways, nachfolgend Gateway genannt, vom Interbus-Master abgefragt werden können und welche Funktionen des Gateways über den Interbus steuerbar sind bzw. abgefragt werden können.

Interbustelegramm des Gateways

Die Kommunikation erfolgt bidirektional mit Telegrammen von zehn Worten (1 Wort = 2 Byte). Das Telegramm besteht aus acht Worten Ein- bzw. Ausgangsdaten, einem Wort PCP (PCP = Peripherals Communication Protocol) und einem Wort als Steuer- bzw. Statusbyte (Abb. 6.20). Mit dem jeweiligen PCP-Wort in den Aus- bzw. Eingangsdaten wäre es möglich, die in Abb. 6.18 gezeigten Konfigurationen remote über den Interbus vorzunehmen. Voraussetzung hierbei ist, dass der Interbus-Master diese Funktion unterstützt. Dies ist beim BKF 201 jedoch nicht der Fall.

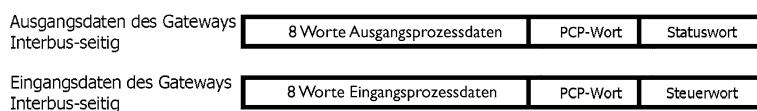


Abb. 6.20 Interbus-Telegramm des Gateways

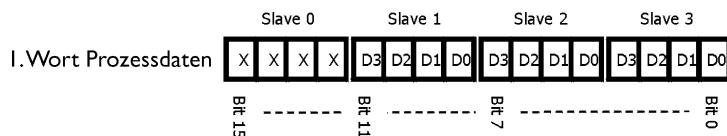


Abb. 6.21 Wort in bitweiser Darstellung

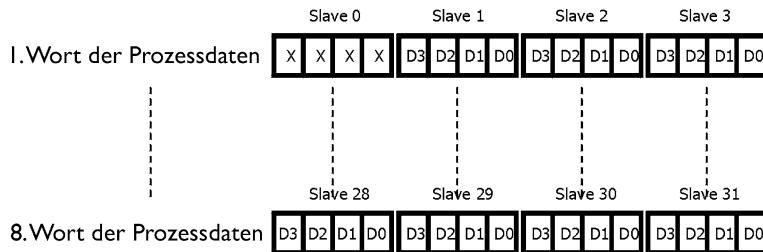


Abb. 6.22 Aufteilung der AS-i Nutzdaten auf die Ein- bzw. Ausgangsprozessdaten des Interbus

Nutzdaten des Gateways

In den Ausgangs- und Eingangsprozessdaten des Interbus sind die Nutzdaten aus dem AS-i enthalten. Je nachdem, ob die Bits der Slaves als Ein- oder Ausgänge konfiguriert wurden, können sie gelesen bzw. beschrieben werden. In Abb. 6.21 wird dargestellt, wie ein Wort bitweise aufgelöst aussieht.

AS-i stellt pro Slave 4 Bit Nutzdaten zur Verfügung. Aus Abb. 6.21 geht hervor, dass ein Wort die Nutzdaten für vier AS-i-Slaves enthält. Weil ein AS-i-Master 32 Slaveadressen verwalten kann, braucht man 8 Worte für die Ein- bzw. Ausgangsprozessdaten (siehe Abb. 6.22). Für Slave 0 ergibt sich eine Besonderheit, hier können weder Bits geschrieben noch gelesen werden, weil es sich hierbei um die Slave-Konfigurationsadresse handelt. In den Werten selbst sind die niederwertigen Bits immer den höherwertigen Slaveadressen zugeordnet und umgekehrt, z. B. gehören im 1. Wort die Bits 0–3 zu Slave 3 und die Bits 12–15 zu Slave 0.

Funktion und Bedeutung von Steuer- und Statuswort

Ein Wort im Telegramm wird je nach Richtung unterschiedlich bezeichnet: Im Ausgangstelegramm heißt es Statuswort, im Eingangstelegramm hingegen Steuerwort. Mit dem Steuerwort können einige AS-i Funktionen remote über den Interbus ausgeführt werden. Mit dem PCP-Wort sind diese Funktionen wesentlich umfangreicher. Jedoch werden, wie bereits erwähnt, diese Funktionen des PCP-Wortes nicht vom Modicon-Interbus-Master unterstützt. Mit dem Statuswort können allgemeine Informationen über den Status des

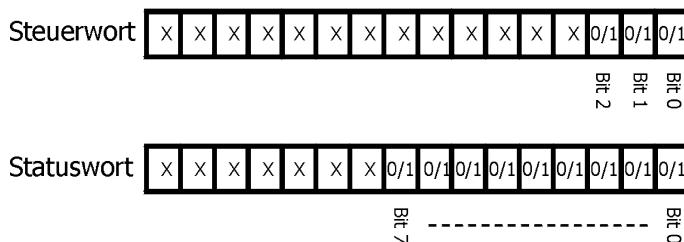


Abb. 6.23 Die verwendeten Bits bei Status- und Steuerwort

Bit	Statuswort	Steuerwort
0	Konfig_OK Die Soll-Konfiguration stimmt mit der Ist-Konfiguration überein	Datenaustausch_inaktiv Der Datenaustausch im AS-I Bus wird unterbrochen, aber beim Löschen diese Bits unmittelbar fortgesetzt
1	LES.0 Es befindet sich ein Slave auf der Konfigurationsadresse 0	Offline Alle AS-I Slaves werden zurückgesetzt. Im AS-I Bus findet keine Kommunikation statt. Wird dieses Bit wieder gesetzt, wird der AS-I Bus neu initialisiert.
2	Auto_prog automatisches Adressieren eines Slaves ist möglich	Auto_prog_Dis Schaltet die Möglichkeit zum automatischen Adressieren ab
3	Auto_prog_available Der vorliegende Fehler kann durch automatisches Programmieren behoben werden. Tritt auf, wenn ein Slave in der IST-Konfiguration fehlt. IST-Konfiguration stimmt nicht mit SOLL-Konfiguration überein.	
4	Projektierung_Aktiv Das Gateway befindet sich im Projektierungsmodus	
5	Normalbetrieb_Aktiv Das Gateway befindet sich im geschützten Betriebsmodus	
6	APF Spannungsausfall am AS-I. Die Slaves werden neu initialisiert. (siehe Offline_ready)	
7	Offline_ready Nach dem Einschalten, wenn der Interbus Master das Gateway über das Steuerwort (siehe Offline) zurückgesetzt hat und nach einem Spannungsausfall benötigt das Gateway einige Zeit zum Überprüfen und Initialisieren des AS-I Busses. Offline_ready wird erst nach Abschluss dieser Phase aktiv.	

Abb. 6.24 Tabelle der einzelnen Bits von Status- und Steuerwort

AS-i abgefragt werden. Bei beiden werden nicht alle Bits genutzt, beim Steuerwort werden Bit 0–2 und beim Statuswort Bit 0–7 genutzt (vgl. Abb. 6.23 und 6.24).

Die folgende Tabelle (Abb. 6.24) zeigt alle Bits von Steuer- und Statuswort.

6.3.4 Die sw-Verknüpfung Interbus/AS-i

Konfiguration der Modicon SPS

Die Konfigurierung der Modicon-SPS wird hier nicht im Einzelnen vorgeführt. In Abb. 6.25 zeigen wir die Auflistung der 3 Baugruppen unter der Konfigurations- und Programmiersoftware Concept.

Lokal TSX Compact E/A-St.							
E/A-Station			Baugruppe				
Baugruppen:	3	Eing.bits:	1040	Ausg.bits:	1040 <th>Eing.bits:</th> <td>0</td>	Eing.bits:	0
Statustab.:		ASCII-Port-Nr.:	Keine	Löschen	Params	Ausschneiden	Kopieren
		Vorherige	Nächste				Einfügen
Mag.-Platz	Baugruppe	Gelesen	Ein.Anf.	Ein.End.	Aus.Anf.	Aus.End.	Beschreibung
1-1	CPU						TSX COMPACT
1-2	CPU						TSX COMPACT
1-3	BKF201(64 W)		300001	300064	400001	400064	Interbus 64 W In/Out
1-4	DAP2x2/253		100001	100008	000001	000008	DC 24 V 8-IN 4-OUT/2A
1-5	DAP2x2/253		100009	100016	000009	000016	DC 24 V 8-IN 4-OUT/2A

Abb. 6.25 Die Auflistung der Baugruppen unter Concept

Variablen Deklaration

Nach der Konfiguration erfolgt die Deklarierung der im SPS-Programm benötigten Variablen (Abb. 6.26). Hierbei sind „Ausgang1“ bis „Ausgang6“ LEDs direkt an der SPS (Module DAP 212). Auch „E2_1“ bis „E2_3“ sind direkt an der SPS, jedoch handelt es sich hierbei um Eingänge (Schalter). Die Worte „AW4“ und „EW4“ sind die Ausgangs- bzw. Eingangsprozessdaten des Gateways. Da im AS-i nur drei Slaves enthalten sind und diese auf die Adressen 1, 2, 3 gesetzt wurden, wird nur jeweils ein Wort benötigt. Die Ausgangs- bzw. Eingangs-PCP-Worte des Gateways wurden mit „PCPOUT“ und „PCPIN“ zwar angelegt, jedoch im Steuerprogramm nicht genutzt. „Interslaveaus“ bzw. „Interslaveein“ sind die Ausgangs- bzw. Eingangsprozessdaten des Interbus Slave 1. Durch entsprechende Programmiertechnik können die Bits der Worte von Prozessdaten einzeln dargestellt werden.

Abb. 6.26 Variablen-Editor

	Exp	Variablenname	Datentyp	Adresse
1	■	Ausgang1	BOOL	000001
2	■	Ausgang2	BOOL	000002
3	■	Ausgang3	BOOL	000003
4	■	Ausgang4	BOOL	000004
5	■	Ausgang5	BOOL	000009
6	■	Ausgang6	BOOL	000010
7	■	AW4	WORD	400005
8	■	E2_1	BOOL	100001
9	■	E2_2	BOOL	100002
10	■	E2_3	BOOL	100003
11	■	EW4	WORD	300005
12	■	Interslaveaus	WORD	400002
13	■	Interslaveein	WORD	300002
14	■	PCPIN	WORD	300003
15	■	PCPOUT	WORD	400003
16	■	Status	WORD	300001
17	■	Steuer	WORD	400001
18	■	Steuer_ASi	WORD	400004

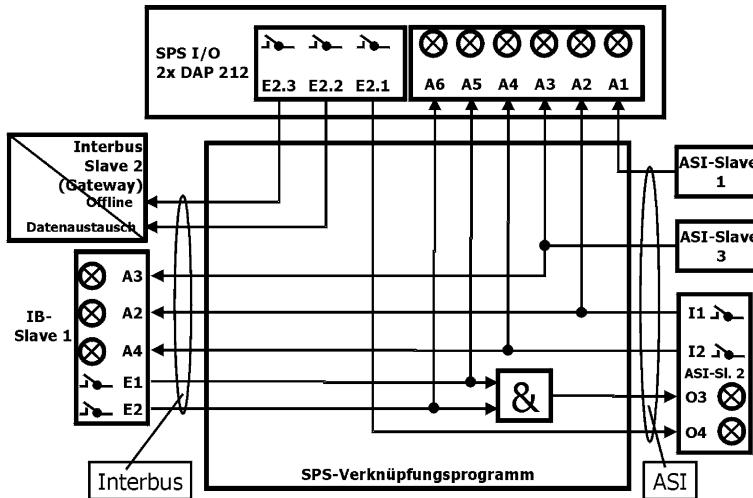


Abb. 6.27 Kommunikationsablauf schematisch

Die Variablen „Status“ und „Steuer“ sind das Status- bzw. Steuerwort des Masters BKF201; das Statuswort wird im Steuerprogramm nicht verwendet. „Steuer_AS-i“ ist das Steuerwort des Gateways.

Kommunikationsablauf

Mit Hilfe der in Abb. 6.18 gezeigten Gerätekonfiguration wurde folgender Kommunikationsablauf beispielhaft realisiert (Abb. 6.27): Zwei Interbus-Eingänge (E1 und E2) werden verundet zum AS-i-Slave 2 übertragen. Die Eingänge E2.2 und E2.3 steuern über den Interbus das Gateway.

Drei AS-i-Slaves und ein Interbus-Slave werden über die SPS Modicon direkt verknüpft. Das exakte Verknüpfungsprogramm ist in Abb. 6.28 dargestellt. Die Programmierung erfolgte nach IEC 1131-3 in ST (Structured Text).

6.4 Die Verbindung einer SPS mit dem PROFIBUS DP

Das folgende Beispiel zeigt die „klassische“ Anwendung einer dezentralen Peripherie: Eine Steuerung mit Masterfunktionalität erreicht über den PROFIBUS periphere Baugruppen mit Slave-Funktionalität.

Eingesetzt wird eine SPS-Station der Reihe SIMATIC S7-300 mit einer Baugruppe CP342-5 als PROFIBUS-Master (Abb. 6.29).

Die Kommunikation zwischen CPU und dem PROFIBUS-Master CP342-5 erfolgt über den Rückwand-Bus der Station. Hierzu werden im Anwendungsprogramm der CPU entsprechende Bibliotheksfunktionen verwendet.

Kommentar (* *) rechtsbündig

```

Steuer:=12;                                (* Setzen der Steuerbits „start-cycle“, „quit-error“*)
Interslaveaus:=16#0000;                      (* IBS Ausgänge auf 0 setzen*)
IF (EW4 AND 16#0001) =16#0001 THEN          (* ASI-S 1 aktiv = SPS A 1 aktiv*)
    A 3:=TRUE;
ELSE      A 3:= FALSE;                     (* ASI-S 1 deaktiv = SPS A 1 deaktiv*)
END_IF;
IF (EW4 AND 16#0010) =16#0010 THEN          (* ASI-S 2 I-1 aktiv = SPS A 2 aktiv *)
    A 2:=TRUE;
ELSE      Ausgang2:= FALSE;                (* ASI-S 2 I-1 deaktiv = SPS A 2 deaktiv *)
END_IF;
IF (EW4 AND 16#0020) =16#0020 THEN          (* ASI-S 2 I-2 aktiv = SPS A 4 aktiv *)
    A 4:=TRUE;
ELSE      A 4:= FALSE;                   (* ASI-S 2 I-2 deaktiv = SPS A 4 deaktiv *)
END_IF;
IF (EW4 AND 16#0100) =16#0100 THEN          (* ASI-S 3 aktiv = SPS A 3 aktiv*)
    A 1:=TRUE;
ELSE      A 1:= FALSE;                  (* ASI-S 3 deaktiv = SPS A 3 deaktiv*)
END_IF;
IF (E2_1)=TRUE THEN                         (* SPS E 2.1 aktiv = ASI-S 2 O-4 aktiv*)
    AW4:=(AW4 OR 16#0080);
ELSE      AW4:=(AW4 AND 16#FF7F);           (* SPS E 2.1 deaktiv = ASI-S 2 O-4 deaktiv*)
END_IF;
IF (E2_2)=TRUE THEN                         (* SPS E 2.2 aktiv = ASI-Gateway Datenaust. deaktiv*)
    Steuer_ASi:=(Steuer_ASi OR 16#0001);
ELSE      Steuer_ASi:=(Steuer_ASi AND 16#FFFE); (* SPS E 2.2 aktiv=ASI-G D'aust. aktiv*)
END_IF;
IF (E2_3)=TRUE THEN                         (* SPS E 2.3 aktiv = ASI-Gateway Offline aktiv*)
    Steuer_ASi:=(Steuer_ASi OR 16#0002);
ELSE      Steuer_ASi:=(Steuer_ASi AND 16#FFFF); (* SPS E 2.3 aktiv=ASI-G Offline deaktiv*)
END_IF;
IF (Interslaveein AND 16#0001) = 16#0001 THEN (* IBS E1 aktiv = SPS A 5 aktiv*)
    A 5:= TRUE;
ELSE      A 5:= FALSE;                   (*IBS E1 deaktiv = SPS Ausgang 5 deaktiv*)
END_IF;
IF (Interslaveein AND 16#0002) = 16#0002 THEN (*IBS E2.0 aktiv = SPS A 6 aktiv*)
    A 6:= TRUE;
ELSE      A 6:= FALSE;                   (*IBS E2 deaktiv = SPS A 6 deaktiv*)
END_IF;
IF (A 5 AND A 6)= TRUE THEN                (*IBS E1 und E2 aktiv = ASI-Sl. 2 O-3 aktiv*)
    AW4:=AW4 OR 16#0040;
ELSE      AW4:=AW4 AND 16#FFBF;             (*IBS E1 oder E2 deaktiv = ASI-S 2 O-3 deakt.)
END_IF;
IF (A 2)= TRUE THEN                        (* ASI-S 2 aktiv = IBS A 2 aktiv*)
    Interslaveaus:=Interslaveaus OR 16#0002;
ELSE
    Interslaveaus:=Interslaveaus AND 16#FFFF; (* ASI-S 2 deaktiv = IBS A 2 deaktiv*)
END_IF;
IF (A 3)= TRUE THEN                        (* ASI-S 3 aktiv = IBS A 3 aktiv*)
    Interslaveaus:=Interslaveaus OR 16#0004;
ELSE
    Interslaveaus:=Interslaveaus AND 16#FFFF; (* ASI-S 3 deaktiv = IBS A 3 deaktiv*)
END_IF;
IF (A 4)= TRUE THEN                        (* A 4 aktiv = IBS A 4 aktiv*)
    Interslaveaus:=Interslaveaus OR 16#0008;
ELSE
    Interslaveaus:=Interslaveaus AND 16#FFF6; (* A 4 deaktiv = IBS A 4 deaktiv*)
END_IF;

```

Abb. 6.28 Realisierung des Verknüpfungsprogramms in ST

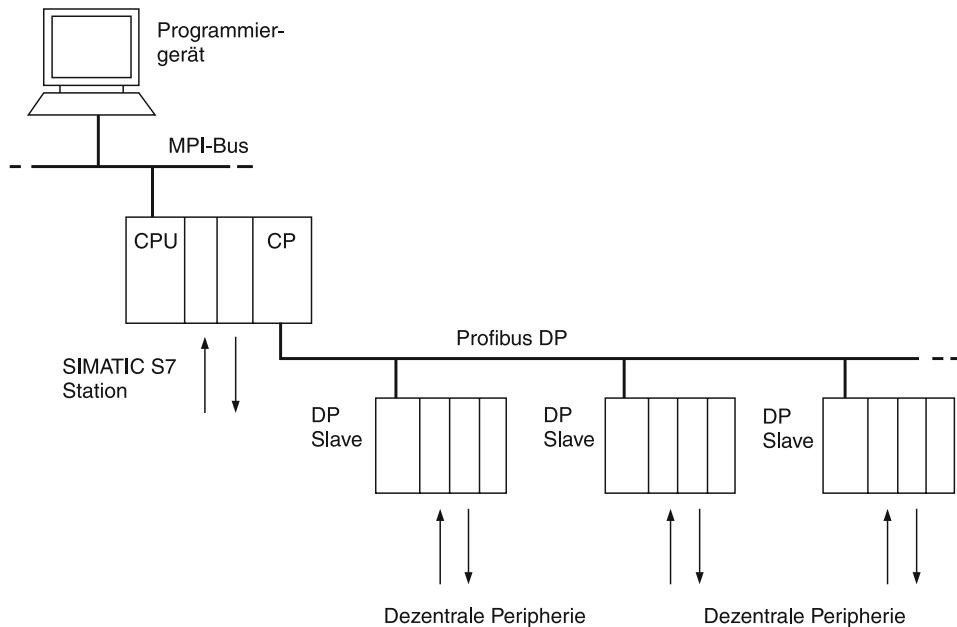


Abb. 6.29 Übersicht der Konfiguration

Zur Konfiguration und Programmierung der Steuerung wird das Programm STEP7 benötigt.

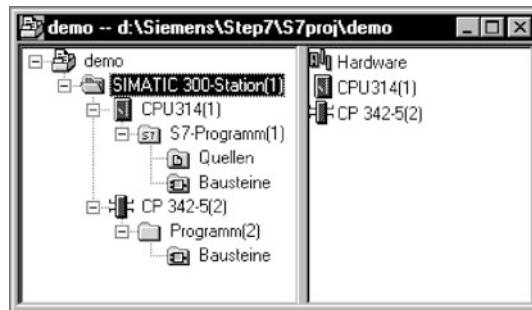
6.4.1 S7-Projekt

In einem S7-Projekt werden alle Konfigurations- und Programmdaten der S7-Station anschaulich geordnet.

Für jede „intelligente“ Baugruppe entsteht ein eigener Ordner für die jeweiligen Daten (Abb. 6.30). Die CPU benötigt eine Systemkonfiguration und das Anwenderprogramm, die Kommunikationsbaugruppe benötigt kein Anwenderprogramm, sondern nur eine Systemkonfiguration. Die Daten werden über die MPI-Schnittstelle in die Station geladen.

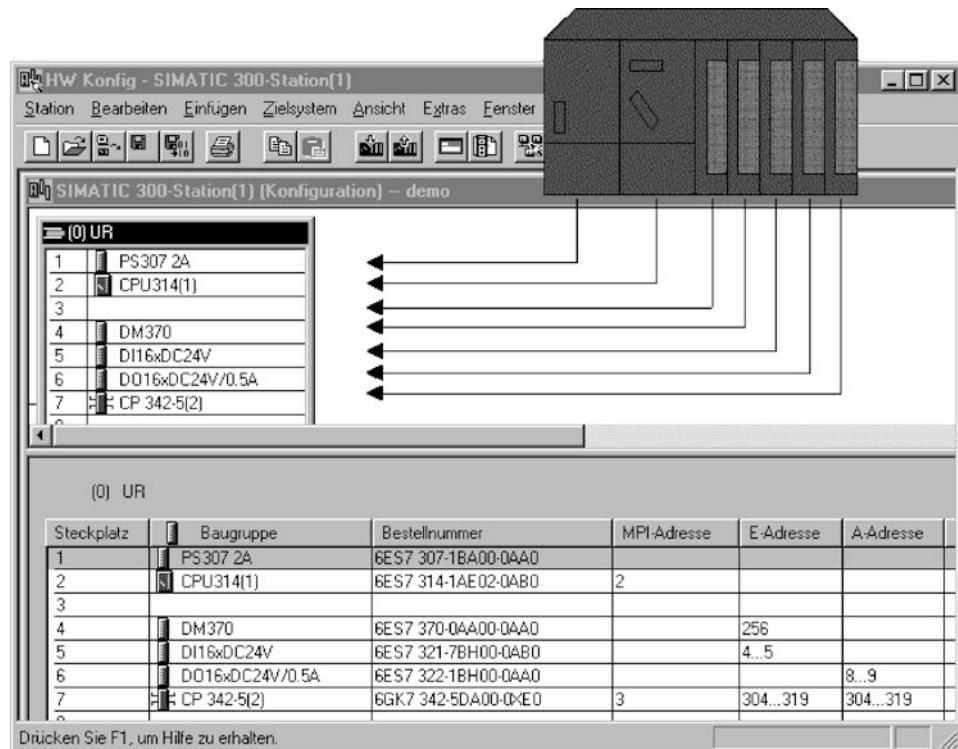
6.4.2 Konfiguration der S7-Station

- Die Konfigurationsdaten für die S7-Station bestehen aus
- der Hardwarezusammensetzung der S7-Station und
- der Buskonfiguration samt
- der dezentralen Peripherie.

Abb. 6.30 STEP7-Projekt

Man öffnet das Objekt „Hardware“ des S7-Projekts und kann dann aus einem Hardwarekatalog die einzelnen Objekte kopieren (Abb. 6.31).

Zunächst werden die zentralen Baugruppen im Baugruppenträger angeordnet, entsprechend der tatsächlichen Bestückung. Im Beispiel sind dies eine Stromversorgung PS307, zentrale Steuereinheit CPU314, zentrale I/O-Baugruppen und der PROFIBUS-Kommunikationsprozessor CP342-5.

**Abb. 6.31** Konfiguration der zentralen Baugruppen

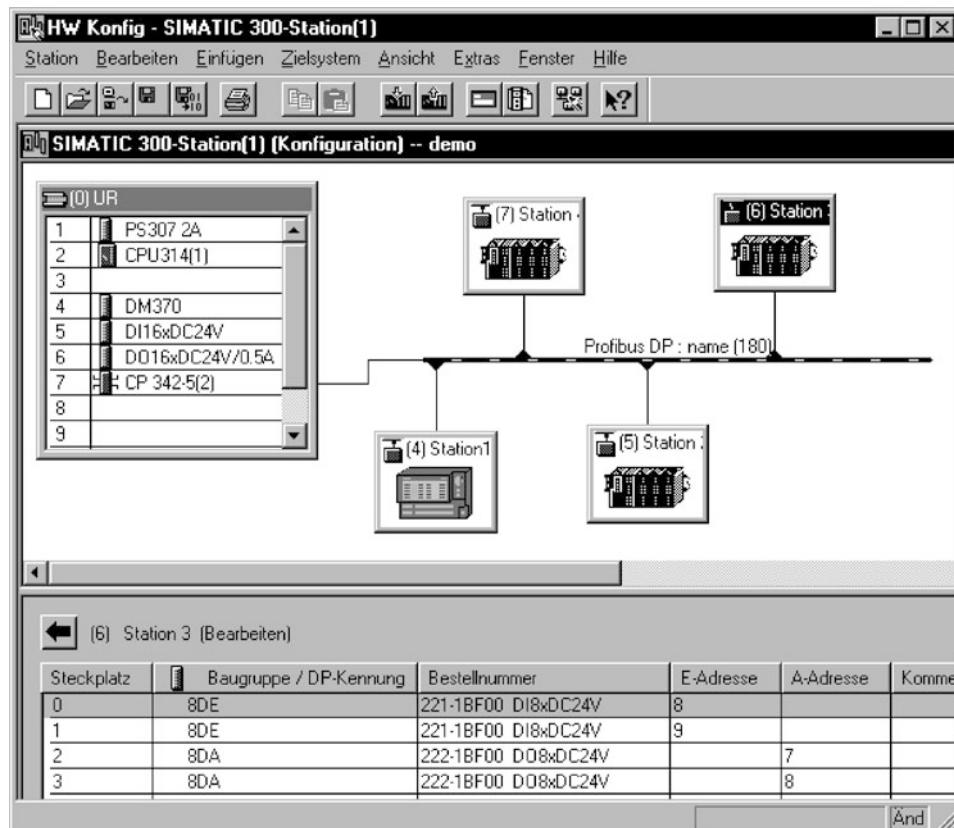


Abb. 6.32 Konfigurieren der dezentralen Peripherie

Die Adressen ergeben sich dabei automatisch. Beispielsweise ergibt sich als Anfangsadresse für den CP342-5 die Adresse 304 (= 130 hex). Diese Adresse wird später benötigt bei der Projektierung der Kommunikation mit der CPU.

Nun werden die Eigenschaften des Kommunikationsprozessors CP342-5 definiert, unter anderem die

- Busparameter,
- Teilnehmeradressen und
- Angaben über die Slaves.

Zunächst wird der Bus konfiguriert: Hierzu markiert man den CP, dann kann ein „DP-Mastersystem“ eingefügt werden. Die Busparameter erscheinen als Eigenschaften des DP-Mastersystems. Bei der Einstellung der Baudrate muss diejenige des langsamsten Busteilnehmers gewählt werden. Ansonsten sollten Anwender ohne intime Kenntnisse der Buszeiten die voreingestellten Werte übernehmen.

Aus dem Hardwarekatalog werden nun die entsprechenden DP-Slaves zum DP-Mastersystem kopiert (einfach mit der Maus auf den Bus ziehen), vgl. Abb. 6.32. Falls es sich um Geräte anderer Hersteller handelt, muss vorher die zugehörige GSD-Datei in die STEP7-Umgebung kopiert werden. Das geschieht im Menü: Extras ⇒ neue GSD installieren. Diese Geräte sind dann im Hardwarekatalog im Verzeichnis „weitere Feldgeräte“ zu finden.

Wichtigste Eigenschaften der Slaves sind PROFIBUSAdresse und die Adressen der zugehörigen Eingangs- und Ausgangsdaten im PROFIBUSmaster.

Aus Abb. 6.32 ist beispielsweise erkennbar, dass der markierte DP-Slave mit dem Namen „Station 3 (Bearbeiten)“ die PROFIBUSAdresse 6 besitzt und jeweils 16 Bit-Eingänge/Ausgänge verarbeiten kann. Außerdem werden die Daten der Eingänge vom PROFIBUSmaster abgelegt an den E-Adressen 8 und 9. Die Adressierung erfolgt immer byteweise.

Im Menü: Ansicht ⇒ Adressübersicht (Abb. 6.33) können die Adressen für die Eingangs- und Ausgangsdaten der gesamten dezentralen Peripherie übersichtlich betrachtet und ausgedruckt werden.

Beispielsweise belegt der DP-Slave mit der PROFIBUSAdresse 4 die E-Adressen 0 und 1, also 2 Byte. Die Spalte „S“ bezieht sich auf den Aufbau des jeweiligen DP-Slaves und kennzeichnet die Slot-Nr. der betreffenden Baugruppe.

The screenshot shows the 'Adresseübersicht' (Address Overview) dialog box. At the top left, it says 'Adressen von:' followed by a list box containing 'CP 342-5(2)' and 'CPU314[1]'. To the right, there are fields for 'Adressebereich von:' (0), 'bis:' (239), 'Freie Adressevergabe:' (Ja), 'Baugruppenträger/' (0/7), and 'Steckplatz:' (CPU-Nr. 1). Below these are filter checkboxes for 'Eingänge' (checked), 'Ausgänge' (checked), and 'Adreßlücken' (unchecked), and a 'Drucken...' button. The main area is a table with columns: Typ, Adr. von, Adr. bis, Baugruppe, DP, R, S, and IF. The data is as follows:

Typ	Adr. von	Adr. bis	Baugruppe	DP	R	S	IF
E	0	1	16DE	180(4)	-	1	-
E	2	2	8DE	180(5)	-	0	-
E	3	3	8DE	180(5)	-	1	-
E	4	7	113	180(5)	-	3	-
E	8	8	8DE	180(6)	-	0	-
E	9	9	8DE	180(6)	-	1	-
E	10	10	8DE	180(7)	-	0	-
E	11	11	8DE	180(7)	-	1	-
A	0	1	16DA	180(4)	-	0	-
A	2	2	8DA	180(5)	-	2	-

At the bottom left is a 'Schließen' (Close) button, and at the bottom right is a 'Hilfe' (Help) button.

Abb. 6.33 Adressübersicht

6.4.3 Kommunikation zwischen CPU und CP

Wenn der CP mit seinen Konfigurationsdaten geladen wurde, ist er bereit zum zyklischen Datenaustausch mit den Slaves: Er holt die Eingangsdaten der projektierten Slaves und legt sie in seinen E-Datenbereich und schreibt aus seinem A-Datenbereich die Ausgangsdaten der Slaves. Für die Kommunikation zwischen CPU und CP stehen 4 spezielle Funktionen aus der Standardbibliothek (im Verzeichnis Stdlib30 ⇒ NetDP) zur Verfügung:

DP_RECV	Empfangen der dezentralen Eingangsdaten
DP_SEND	Senden der dezentralen Ausgangsdaten
DP_DIAG	Diagnosefunktionen
DP_CTRL	spezielle Steueraufträge an den CP

Diese Funktionen werden in das zyklische CPU-Programm eingebunden. Abb. 6.34 zeigt den typischen Ablauf des CPU-Programms. Mit der ersten Anweisung im OB1 werden

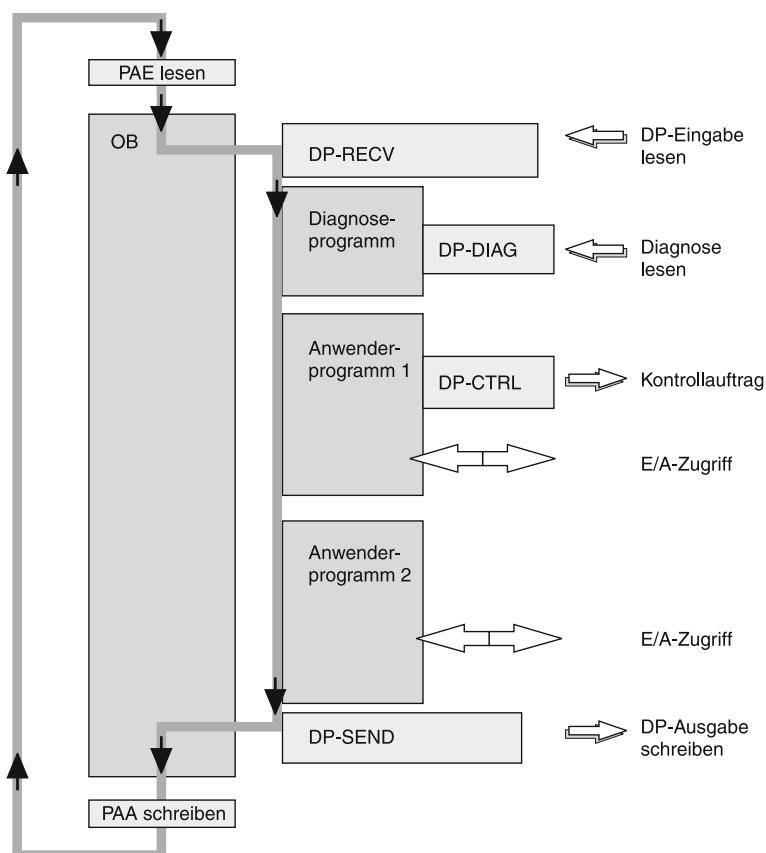


Abb. 6.34 Programmablauf

mit DP_RECV die dezentralen Eingänge gelesen, die letzte Anweisung im OB1 betrifft mit DP_SEND die dezentralen Ausgänge. Je nach Bedarf kann mit DP_DIAG die Kommunikation CPU \leftrightarrow CP und die Situation am PROFIBUS überwacht werden. Mit der Funktion DP_CTRL können spezielle Anweisungen für die Funktion des CP gegeben werden (z. B. STOP, RUN, OFFLINE, CLEAR).

6.4.4 Programmbeispiel

Die dezentralen Eingangsdaten werden im angegebenen CPU-Speicherbereich (hier 12 Byte ab MB100) abgelegt (Abb. 6.35). Die Reihenfolge der Daten entspricht denjenigen des CP-Speichers (siehe auch Adressübersicht Abb. 6.33).

```
OB1 : Zyklisch ablaufendes Programm
Netzwerk 1: Dezentrale Eingangsdaten vom CP holen
    CALL "DP_RECV"
    CPLADDR :=W#16#130           // CP Address (slot 7)
    RECV   :=P#H 100.0 BYTE 12     // Buffer for Receive Data MB100..MB111
    NDR    :=M0.0                  // =1: New Data Received
    ERROR  :=M0.1                  // Error flag
    STATUS  :=MW2                  // Status Communication CPU <=> CP
    DPSTATUS:=MB1                 // Status CP (DP Master)

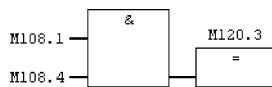
Netzwerk 2 : Steuerungsprogramm
    CALL FC 100                  // Ausgangsdaten =f(Eingangsdaten) berechnen

Netzwerk 3 : Dezentrale Ausgangsdaten schreiben zum CP
```

```
CALL "DP_SEND"
CPLADDR:=W#16#130           // CP Address (slot 7)
SEND   :=P#H 120.0 BYTE 11     // Buffer for Send Data MB120..MB130
DONE   :=M0.2                  // =1: Done
ERROR  :=M0.6                  // Error flag
STATUS  :=MW4                  // Status Kommunikation CPU <=> CP
```

FC100 : Steuerung, Beispiele für Zugriffe auf die dezentrale Peripherie

Netzwerk 1: Beispiel für Binäre Verknüpfung



Netzwerk 2 : Beispiel für Byte-Zuweisung

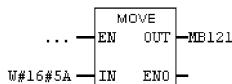


Abb. 6.35 Beispiel für Anwenderprogramm in STEP7

Die dezentralen Ausgangsdaten werden in den angegebenen CPU-Speicherbereich (hier 11 Byte ab MB120) eingetragen und mit der DP_SEND zum CP übertragen.

In der Funktion FC100 (Abb. 6.35) sind Beispiele für den Zugriff auf die dezentrale Peripherie angegeben. Beispielsweise betreffen die binären Operanden M108.1 und M108.4. dezentrale Eingänge des DP-Slave mit der PROFIBUSadresse 6. Der binäre Operand M120.3 repräsentiert einen dezentralen Ausgang des DP-Slaves mit der PROFIBUSadresse 4. Dies lässt sich nachvollziehen durch Vergleich mit der Adressübersicht in Abb. 6.33.

Damit komfortabel programmiert werden kann, können selbstverständlich in geeigneter Weise symbolische Bezeichnungen für die dezentralen I/O verwendet werden.

6.5 Konfiguration AS-i/Ethernet/IP-Gateway an Ethernet/IP

6.5.1 Aufbau der Bussysteme

Bei diesem Beispiel handelt es sich um die hierarchische Kombination zweier Bussysteme (Abb. 6.36). Der AS-i Bus ist dabei dem Ethernet/IP unterlagert. Auf die Grundlagen der beiden Systeme wird an dieser Stelle nicht eingegangen.

Am AS-i Bus sind vornehmlich die einfachen Geräte, wie z. B. Näherungsschalter, Ventile oder einfache Antriebe angeschlossen.

Ethernet/IP dient dazu, die AS-i Kreise zu vernetzen und komplexe Feldgeräte wie z. B. Servo-Antriebe anzuschließen.

Die hier verwendete Steuerung CompactLogix L32E hat die Ethernet/IP-Schnittstelle bereits integriert.

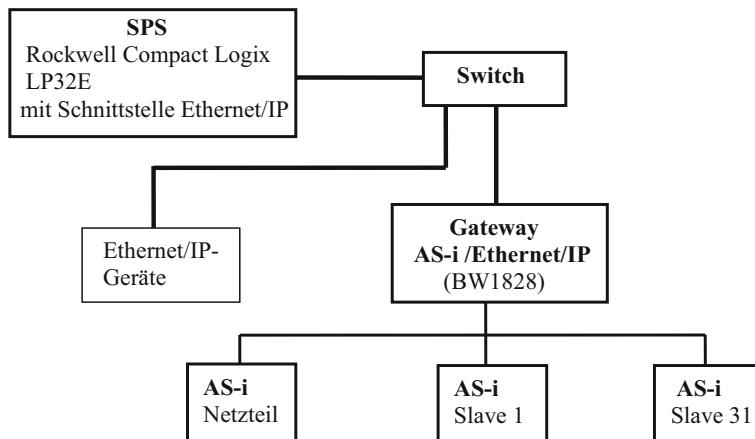


Abb. 6.36 AS-i ist dem Ethernet/IP unterlagert

Die Ethernet-Kommunikation erfolgt mit der 100Base-TX Physik. Dazu ist der Switch nötig, an den keine weitergehenden Anforderungen gestellt werden. Im industriellen Umfeld ist aber darauf zu achten, dass die verwendeten Ethernet-Komponenten hinsichtlich der Umweltbedingungen wie Hitze, Vibrationen und EMV geeignet sind.

Der AS-i Slave mit der Adresse 1 sei ein 4E/4A-Modul, der Slave mit der Adresse 31 sei ein Analogeingangs-Modul, bei dem nur zwei Kanäle benutzt werden.

6.5.2 Konfiguration des AS-i-Netzwerks und des AS-i/Ethernet/IP-Gateways

Im ersten Schritt wird das AS-i Netzwerk konfiguriert und in Betrieb genommen. Dazu erhalten die AS-i Slaves zuerst Ihre Adressen. Dies kann entweder mit einem Hand-Adressiergerät oder den AS-i Master geschehen. Im Falle der Adressierung über den AS-i Master kann je nach Umfang des Projekts und persönlichen Vorlieben die Adressvergabe über ein PC-Programm (AS-i Control Tools) oder über das Display und die Tasten am Master durchgeführt werden.

AS-i Slaves werden vom Hersteller immer mit der Adresse 0 ausgeliefert. Daher ist es wichtig, bei der Adressvergabe über den Master immer einen Slave nach dem anderen anzuschließen und die jeweilige Adresse 0, also den neuen Slave auf seine Zieladresse zu programmieren.

Wenn man die Adresse mit dem Handadressiergerät vergibt, kann man die bei den meisten AS-i Slaves vorhandene Adresserbuchse verwenden, die den Slave automatisch beim Stecken des Programmierkabels vom Bus trennt und somit sicherstellt, dass nur ein Slave gleichzeitig am Adressiergerät angeschlossen ist.

Sind alle Slaves adressiert und am AS-i Kreis angeschlossen kann im AS-i Projektierungsmodus die Funktion überprüft werden. Zum einen können die vorhandenen Adressen am Display überprüft werden. Zum anderen aber können auch die von den Sensoren gelieferten binären und analogen Signale schon jetzt geprüft werden und eventuelle Fehler in der Verkabelung oder im Sensor frühzeitig behoben werden. Diese ist eine besonders nützliche Eigenschaft des AS-i Systems, da somit Teile einer Anlage schon in Betrieb genommen und getestet werden können, ehe das Gesamtsystem fertig montiert ist.

Wenn alles in Ordnung ist, kann die AS-i Konfiguration im AS-i Master projektiert (= nichtflüchtig gespeichert) werden.

Nun fehlt noch die Festlegung der Ethernet/IP-Parameter.

Entweder wird dazu eine statische Konfiguration, bestehend aus IP-Adresse, Netzwerk-Maske und Gateway festgelegt oder per DHCP (Dynamic Host Control Protocol) automatisch vergeben. Bei Vergabe über DHCP muss der DHCP-Server konfiguriert werden, damit das Gerät eine definierte IP-Adresse erhält, über die dann in der Steuerung die Adressierung erfolgt.

Damit ist die Projektierung des AS-i Kreises abgeschlossen.

6.5.3 Kommunikation über Ethernet/IP

Ethernet/IP gehört mit DeviceNet und ControlNet zur Familie der CIP-Protokolle. Bei diesen Protokollen sind die Daten in Objekten gruppiert, die in einem Objekt/Instanz/Attribut-Schema aufgebaut sind.

Mehrere Datensätze, die zusammen zyklisch übertragen werden sollen, werden zu Assembly-Objekten zusammengefasst, die dann als ein einzelnes Objekt übertragen werden können. Das AS-i Ethernet/IP-Gateway bietet eine ganze Reihe von Assemblies (genauer: Instanzen des Assembly-Objekts) an, die sich in der Zusammenstellung der zu übertragenden Daten unterscheiden.

In unserem Beispiel soll die Instanz 104 als Eingangs-Assembly und die Instanz 140 als Ausgangs-Assembly verwendet werden. Diese Instanzen übertragen die Binärdaten aller Single- und A-Adressen sowie die Analogdaten der Slaves 29–31 (Abb. 6.37). Außerdem

Daten	Größe (Byte)
Binärdaten Slave 1(SingleA)..31(SingleA)	16
Analogdaten Slave 29(Single/A/B)..31(Single/A/B)	24
Kommmandoschnittstelle	12
Summe	52

Abb. 6.37 Eingangs-Assembly

Bit	7	6	5	4	3	2	1	0
Byte 0	Flags					Daten des Slave der Adresse 1, bzw. 1A		
	F3	F2	F1	F0	D3	D2	D1	D0

Abb. 6.38 As-i-Telegramm

Abb. 6.39 Flags im As-i-Telegramm

Flags		
	Eingangsdaten	Ausgangsdaten
F0	ConfigError	Off-line
F1	APF	LOS-master-bit
F2	PeripheryFault	→ ConfigurationMode
F3	ConfigurationActive	→ ProtectedMode

ConfigError: 0=ConfigOK, 1=ConfigError
APF: 0=AS-i-Power OK, 1=AS-i-Power Fail
PeripheryFault: 0=PeripheryOK, 1=PeripheryFault
ConfigurationActive: 0=ConfigurationActive, 1=ConfigurationInactive
Off-Line: 0=On-Line, 1=Off-Line
LOS-master-bit 0=Off-Line bei ConfigError deaktiviert
1=Off-Line bei ConfigError aktiviert.

Bit	7	6	5	4	3	2	1	0
Byte 16								Slave 31 bzw. 31A, Analog-Kanal 1, High-Byte
Byte 17								Slave 31 bzw. 31A, Analog-Kanal 1, Low-Byte
Byte 18								Slave 31 bzw. 31A, Analog-Kanal 2, High-Byte
Byte 19								Slave 31 bzw. 31A, Analog-Kanal 2, Low-Byte
Byte 20								Slave 31, Analog-Kanal 3, bzw. 31B, Analog Kanal 1, High-Byte (hier nicht benutzt)
Byte 21								Slave 31, Analog-Kanal 3, bzw. 31B, Analog Kanal 1, Low-Byte (hier nicht benutzt)
Byte 22								Slave 31, Analog-Kanal 4, bzw. 31B, Analog Kanal 2, High-Byte (hier nicht benutzt)
Byte 23								Slave 31, Analog-Kanal 4, bzw. 31B, Analog Kanal 2, Low-Byte (hier nicht benutzt)
Byte24.. Byte31								Analogdaten Slave 30 bzw. 30A/30B
Byte 32.. Byte 39								Analogdaten Slave 29 bzw. 29A/30B

Abb. 6.40 Jeder Analog-Slave belegt 8 Bytes in der Assembly

wird auch die Kommandoschnittstelle für zu Diagnose und Konfiguration von AS-i über die SPS unterstützt.

Die Daten D0...D3 des ersten AS-i Slaves (Adresse 1), also des 4-mal E/4A-Moduls finden sich bei den Binärdaten im ersten Byte, also Byte 0, jeweils für Ein- und Ausgänge (Abb. 6.38).

Die Flags F0...F3 sind in Abb. 6.39 erläutert.

Die Daten des zweiten AS-i Slaves, also des Analogmoduls mit der Adresse 31, finden sich im Analogdatenblock am Anfang, ab Adresse 0. Innerhalb der Assembly ist dies also Byte 16, bzw. Word-Adresse 8 wegen der davor übertragenen Analogdaten (Abb. 6.40).

6.5.4 Die Software-Verknüpfung zwischen AS-i und Ethernet/IP

Dieses Kapitel zeigt beispielhaft die Inbetriebnahme des AS-i 3.0 EtherNet/IP-Gateways BW1828. Dabei wird als Programmierumgebung für die SPS die Software RSLogix 5000, Version 13,00 von Rockwell Automation verwendet. Dies ist die Rockwell Standard-Software zur Programmierung der CompactLogix SPS-Familie und zur Projektierung der zugehörigen Netzwerke.

1. Starten Sie die Software RSLogix 5000 (Abb. 6.41).
2. Wählen Sie *New* aus dem Menü *File*.
3. Wählen Sie jetzt Ihren Controller aus (in unserem Fall die SPS L32E) und bestätigen Sie mit *OK*.

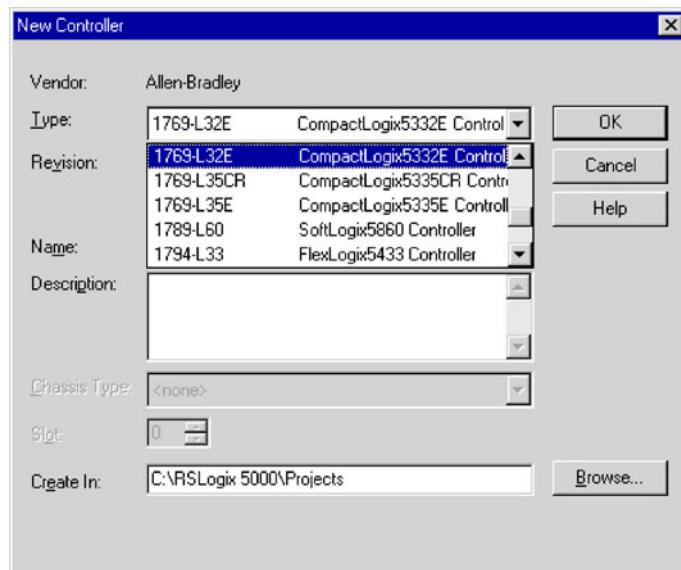
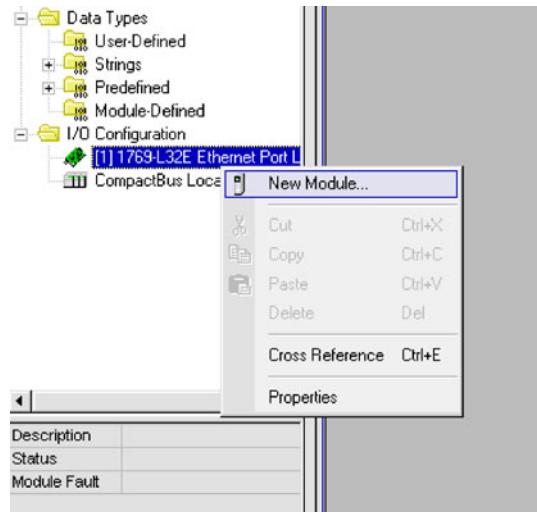


Abb. 6.41 RSLogix-Fenster: Wahl der SPS

Abb. 6.42 RSLogix-Fenster:
Wahl des Ethernet-Ports



4. Klicken Sie im Baum-Ansichtsfenster mit der rechten Maustaste auf den Ethernet-Port des 1769-L32E-Controllers (Abb. 6.42).
5. Klicken Sie im PopUp-Fenster mit der linken Maustaste auf *New Module*.
6. Wählen Sie den Eintrag *Generic Ethernet Module* und betätigen Sie mit *OK*. Dies bedeutet, dass das Modul in seinen Eigenschaften frei beschreibbar ist (Abb. 6.43).
7. Tragen Sie jetzt alle erforderlichen Eigenschaften des Moduls ein (Abb. 6.44):

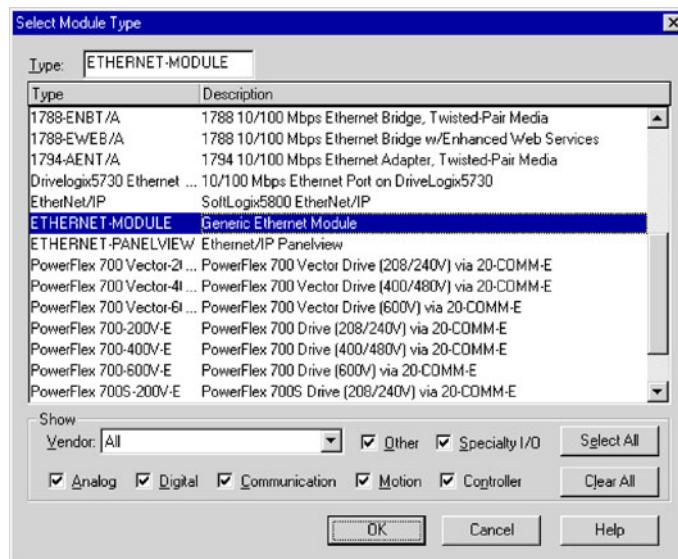


Abb. 6.43 RSLogix-Fenster: Frei wählbares Modul

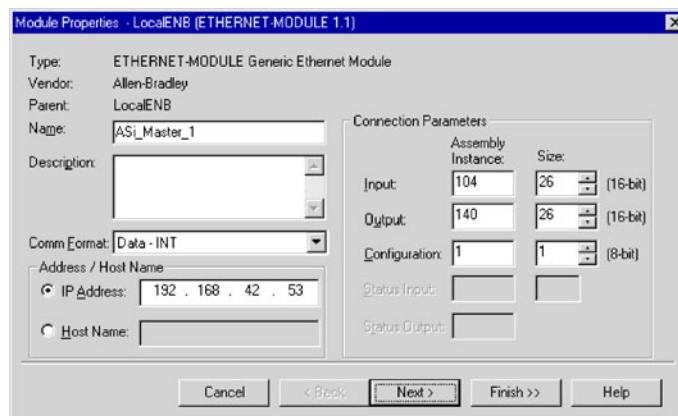


Abb. 6.44 RSLogix-Fenster: Parametrierung des Moduls

- Controller-Name, ein frei vergebbarer Name, z. B., „AS-i-Master_1“
- Comm.-Format (wählen Sie „Data - DINT“),
- IP-Adresse,
- Verbindungsparameter:
- Assembly Instance-Input/Output: Tragen Sie hier die Instanzen 105 für Input und 140 für Output ein, Länge jeweils 13 32-bit-Wörter (oder 52 Bytes).
- Assembly Instance-Configuration: Die Einträge hier sind nicht relevant, müssen aber ungleich 0 sein. Tragen Sie „1“ ein.

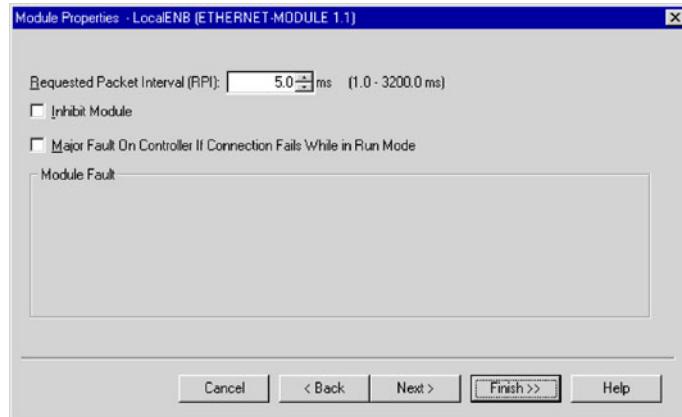


Abb. 6.45 RSLogix-Fenster: Buszykluszeit

8. Betätigen Sie den Button *Next*.
9. Tragen Sie im Feld *Request Packet Interval* (RPI) die Zeit 5 ms ein (Abb. 6.45).
10. Betätigen Sie den Button *Finish*.

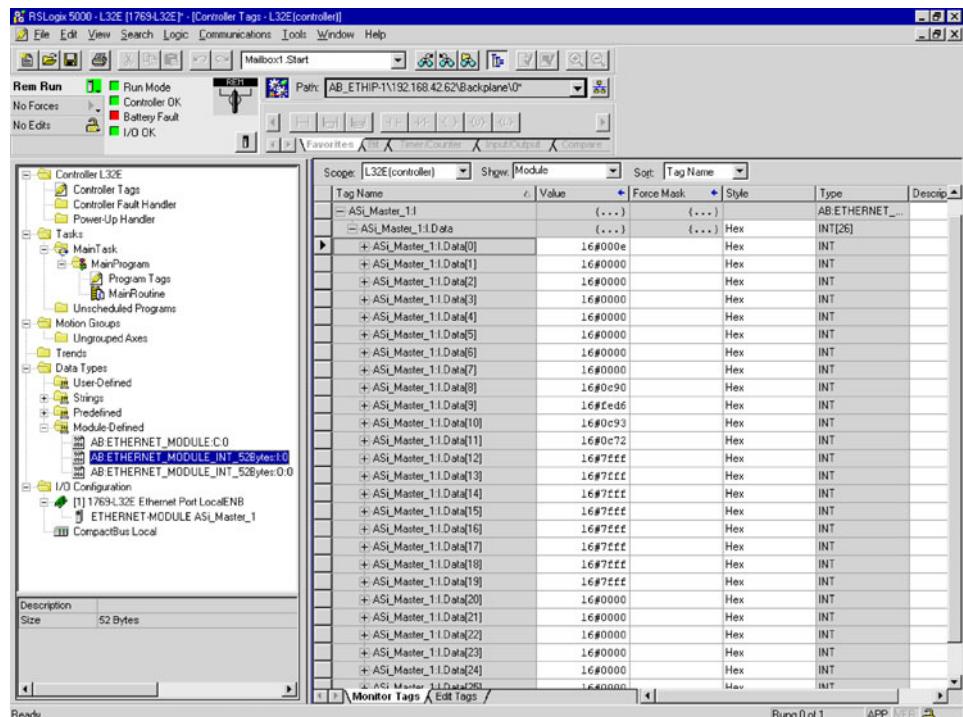


Abb. 6.46 RSLogix:-Fenster: Daten (0) ... (25)

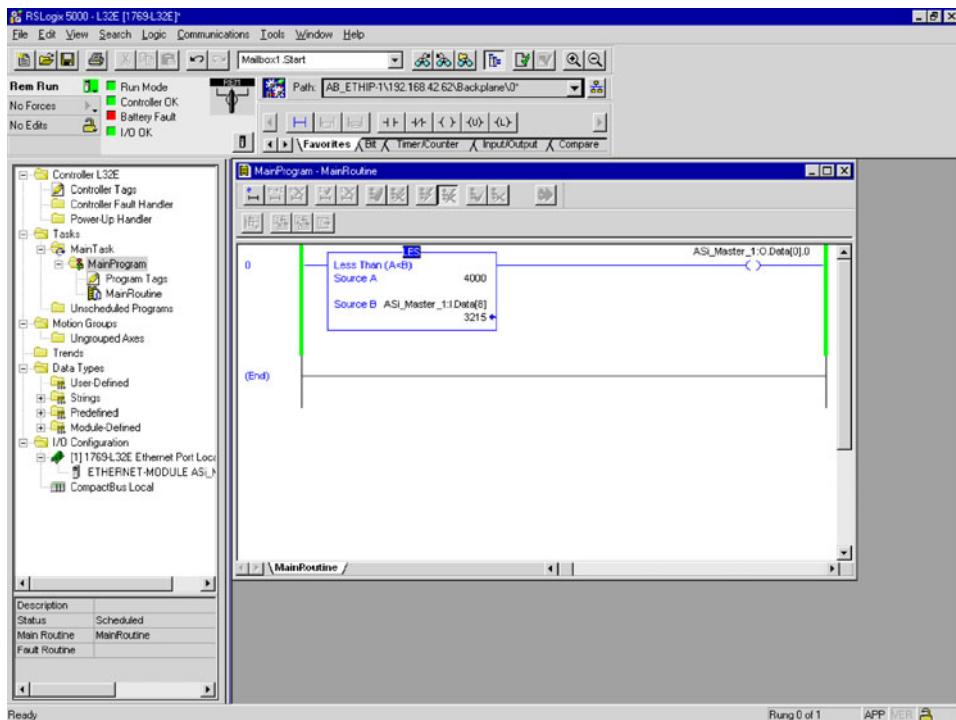
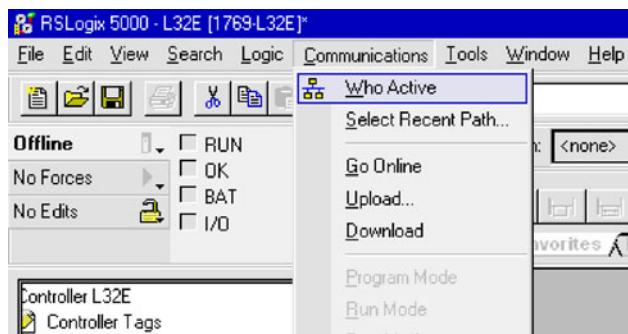


Abb. 6.47 RSLogix-Fenster: Demoprogramm

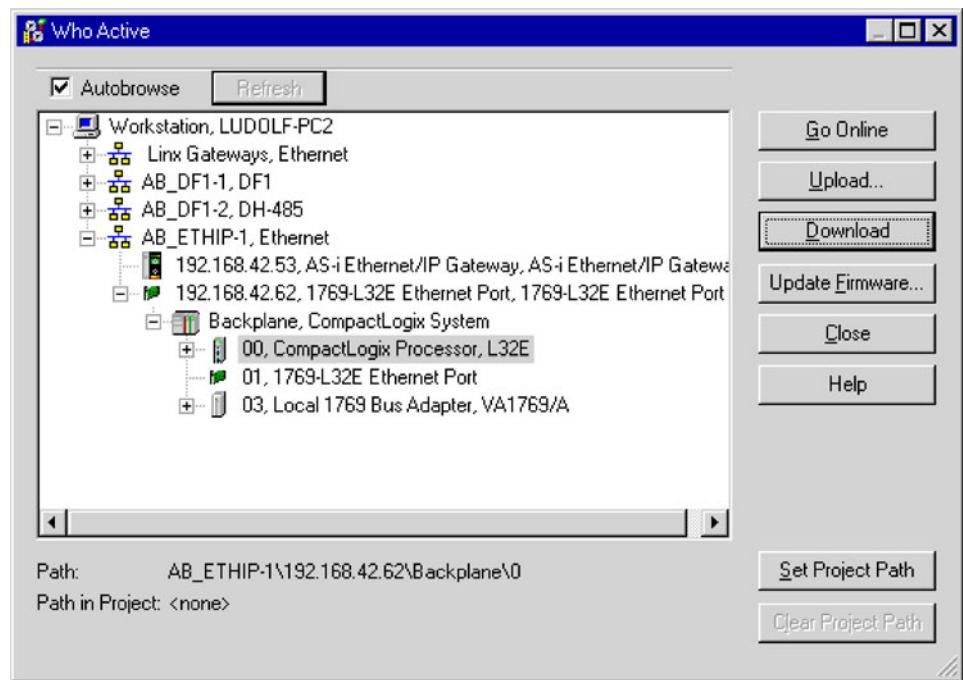
11. Damit ist das Ethernet-Gateway mit der SPS logisch verbunden und Sie können die Daten jetzt für das eigene Programm verwenden. Die Daten finden sich unter Data-Types, Module-Defined AB_Ethernet Module unter ASI_Master 1 (Abb. 6.46). Die Analogdaten von Analog-Kanal 1/Slave 31 finden sich im Eingangswort Data (8). Beispielhaft soll jetzt ein Programm geschrieben werden, das den Ausgang 1 des Slave 1 setzt, wenn der Analogwert von Kanal 1 eine festgelegte Grenze (4000) überschreitet.
- Der Ausgang 1 ist Bit 0 in Ausgangswort Data (0).
- Das fertige Programm ist in grafischer Darstellung in der obigen Abb. 6.47 gezeigt. Es muss nun noch in die Steuerung geschrieben werden.
12. Beim erstmaligen Downloaden der Software muss der Übertragungspfad angegeben werden. Wählen Sie dazu aus dem Menü *Communications* den Eintrag *Who active* (Abb. 6.48).
13. Mit Doppelklick auf das Piktogramm *CompactLogixProcessor* beginnen Sie mit dem Download (Abb. 6.49).

Abb. 6.48 Fenster RSLogix:

Vorbereitung Download



Das Beispiel zeigt, dass sich offene Feldbusse-Systeme auch über Architekturgrenzen hinweg sehr gut vernetzen lassen. Auch ist eine Integration von Geräten unterschiedlicher Hersteller meistens kein Problem.

**Abb. 6.49** RSLogix:-Fenster: Download Demoprogramm

Datenblätter

7

Wir weisen darauf hin, dass diese Übersicht keinen Anspruch auf Vollständigkeit erhebt und dass sich manche der angegebenen Parameter durch Weiterentwicklung ändern können.

7.1 AS-i (Aktor/Sensor-Interface)

Hersteller	Siemens, Pepperl+Fuchs, ifm, Balluff, Festo, Sick, Leuze, Baumer, Turck, Schneider Automation, Bihl+Wiedemann u. a.
Offener Bus?	ja
Nutzerorganisation	AS-International Association Zum Taubengarten 52 63571 Gelnhausen Deutschland Tel: +49 (6051) 4732-12 (über 250 Mitgliedsfirmen) www.as-interface.net
Normen	EN 50295 IEC 62026-2
Topologie	offene Baumstruktur
Teilnehmer, max.	62 Slaves, (Mehrfachanschaltung bis zu 4 Sensoren/Aktoren pro Slave möglich),
ohne/mit Repeater	maximal 2 Repeater zwischen Master und Slave
Buszugriffsverfahren	Master/Slave in zyklischem Polling
Übertragungsrate(n) und Leitungslänge(n)	167 kBd, 100 m max. ohne Repeater und 300 m in jede Richtung mit Reatern
Telegrammformat(e)	fest, 4 Bit Daten, 4 Bit Parameter. Feste Protokolle zur Mehrbit-Übertragung im Master implementiert.
Datensicherung	Paritätsbit, Mehrfachabtastung und weitere Sicherungsmaßnahmen in physikalischer Schicht
Buspegel	4 VSS Datenpegel und 24 V Hilfsenergie (8A max.)

7.2 KNX (ehemals EIB, European Installation Bus)

Hersteller	Optionale Standardimplementierungen von mehr als 10 Herstellern. ETS (Engineering Tool Software, KNX Association)
Offener Bus?	ja
Nutzerorganisation	KNX Association, De Kleetlaan 5, Bus 11, B1831 Diegem (Brussels) www.knx.org
Normen	ISO/IEC 14543-3 ; CENELEC EN 50090 ; CEN EN 13321-1 ; GB/T 20965
Topologie	logisch: Anlage/Bereich/Line; physisch: abhängig vom Medium
Teilnehmer, max.	mehr als 60.000
Buszugriffsverfahren	Abhängig vom Übertragungsmedium, CSMA (CA) für TP
Übertragungsraten und Leitungslängen	PL (Stromnetz): 1,2 kBd bis zu 600 m TP (twisted pair): 9,6 kBd bis 1000 m RF (Funk): 9,6 kBd bis 300 m IEC 802-2 (Ethernet): 10 MBd
Telegrammformate	14 Bytes Nutzdaten (254 für lange Rahmen); standardisierte „Interworking“-Formate
Datensicherung	Abhängig vom Medium: TP: Paritybit für jedes Zeichen, Prüfsumme über das Telegramm
Buspegel	abhängig vom Medium

7.3 Sercos I, II und III

Hersteller	Über 300 Produkte von über 100 Herstellern (Sercos I-III), weltweit über 5 Millionen Echtzeitknoten im Einsatz
Offener Bus?	ja (I seit 1987, II seit 1999, III seit 2005)
Nutzerorganisation	Sercos International e.V., Küblerstr.1, 73079 Süssen Tel. +49 (0)7162-9468-65, Fax +49 (0)7162-9468-66 info@sercos.de ; www.sercos.de
Normen	IEC 61784-1/61158 (CP 16/1: Sercos I) IEC 61784-1/61158 (CP 16/2: Sercos II) IEC 61784-2/61158 (CP 16/3: Sercos III)
Topologie	I, II: Ring; III: Ring und Linie
Teilnehmer	1 Master, ≤ 254 Slaves/Ring; III: ≤ 511 Slaves
Zykluszeit	konfigurierbar; I, II: $t \geq 62,5 \mu\text{s}$; III: $\geq 31,25 \mu\text{s}$. Jitter $\ll 1 \mu\text{s}$
Übertragungsraten	I: 2 und 4 MBit/s II: 2/4/8 und 16 MBit/s III: 100 MBit/s Voll-Duplex
Telegrammformate	I, II: 2, 4, 6 oder 8 Bytes Nutzdaten. Telegrammlänge proportional zur Teilnehmerzahl. III: 40 – 1494 Bytes
Basisprotokoll	I, II: HDLC (<i>high level datalink control</i>). III: Ethernet
Leitung	Plastik-LWL bis 50 m; Glasfaser-LWL bis 250 m. III: auch verdrillte Kupferleiter
Sonstiges	I, II: ASIC-Kommunikationscontroller SERCON816. III: Sercos III IP in FPGA. Oder Standard-Ethernet-Controller

7.4 PROFIBUS

7.4.1 PROFIBUS-DP

Hersteller	Siemens, Festo, Bosch, Pepperl+Fuchs, Schneider Automation, u. a.																
Offener Bus?	ja																
Nutzerorganisation	PROFIBUS-Nutzerorganisation Haid- und Neu-Straße 7 76131 Karlsruhe http://www.profibus.com/ (weltweit ca. 700 Mitgliedsfirmen)																
Normen	EN 50170, DIN 19 245, Teil 3																
Topologie	Linie																
Teilnehmer, max. ohne/mit Repeater	max. 126 Teilnehmer;																
Buszugriffsverfahren	Master/Slave																
Übertragungsrate(n) und Leitungslänge(n)	<table border="1"> <thead> <tr> <th>Kupfer</th> <th>DIN 19 245 Teil 3</th> </tr> </thead> <tbody> <tr> <td>9,6 kBd</td> <td>1200 m</td> </tr> <tr> <td>19,2 kBd</td> <td>1200 m</td> </tr> <tr> <td>93,75 kBd</td> <td>1200 m</td> </tr> <tr> <td>187,5 kBd</td> <td>1000 m</td> </tr> <tr> <td>500 kBd</td> <td>400 m</td> </tr> <tr> <td>1,5 MBd</td> <td>200 m</td> </tr> <tr> <td>12 MBd</td> <td>100 m</td> </tr> </tbody> </table> <p>LWL: > 100 km</p>	Kupfer	DIN 19 245 Teil 3	9,6 kBd	1200 m	19,2 kBd	1200 m	93,75 kBd	1200 m	187,5 kBd	1000 m	500 kBd	400 m	1,5 MBd	200 m	12 MBd	100 m
Kupfer	DIN 19 245 Teil 3																
9,6 kBd	1200 m																
19,2 kBd	1200 m																
93,75 kBd	1200 m																
187,5 kBd	1000 m																
500 kBd	400 m																
1,5 MBd	200 m																
12 MBd	100 m																
Telegrammformat(e)	typ. 32 Byte Daten, max. 246 Byte																
Datensicherung	Längs- und Querparität, HD = 4																
Buspegel	RS 485, NRZ																
Leitung	Zweidrahtleitung verdrillt, geschirmt																
Sonstiges	gemischter Betrieb mit PROFIBUS FMS und PA möglich																

7.4.2 PROFIBUS-PA

Hersteller	Siemens, Pepperl+Fuchs, Endress u. Hauser, Samson, Krohne, u. a.
Offener Bus?	ja
Nutzerorganisation	wie 7.4.1
Normen	DIN 19 245, Teil 4, IEC1158-2, EN 50170 (pr Amendment 2)
Topologie	Linie
Teilnehmer, max. ohne/mit Repeater	32 Slaves, 127 mit Repeater, bei Ex-i 10 Tln. je 10 mA
Buszugriffsverfahren	Master/Slave
Übertragungsrate(n) und Leitungslänge(n)	31,25 kBaud 1900 m
Telegrammformat(e)	siehe PROFIBUS-DP
Datensicherung	CRC
Buspegel	0,9 V _{SS} / ± 9 mA an 100 Ohm, Manchester II
Leitung	Zweidrahtleitung für Daten und Teilnehmerversorgung verdrillt, geschirmt
Sonstiges	für explosionsgefährdete Bereiche, Schutzart eigensicher

7.5 Interbus

Hersteller	Phoenix Contact 32817 Blomberg
Offener Bus?	ja
Nutzerorganisation	DRIVE COM SAFE COM Interbus Club, Fach 1108, 32817 Blomberg www.interbusclub.com
Normen	DIN E 19 258; EN 50254, IEC 61158
Topologie	Ring-Struktur
Teilnehmer, max.	max. 256 E/A-Module; Fernbus: 256, Busklemmen und E/A-Module (haben Repeaterfunktion) Nahbus (Peripheriebus): 8 E/A-Module
Buszugriffsverfahren	verteiltes Schieberegister
Übertragungsrate(n) und Leitungslänge(n)	Fernbus: 500 kBd, max. 12,8 km Peripherie-Bus: 500 kBd, max. 10 m
Telegrammformat(e)	max. 512 Byte Summenrahmentelegramm: Steuerinfo, Nutzdaten aller Stationen, Steuerinfo
Datensicherung	16-bit-CRC, HD = 4
Buspegel	Fernbus: RS 485 Peripherie-Bus: TTL-Pegel
Leitung	Peripherie: 15-adrig paarweise verdrillt Fernbus: 5-adrig paarweise verdrillt
Sonstiges	Lichtwellenleiter als optionales Medium für Fernbus verfügbar

7.6 Modbus Plus

Hersteller	Schneider Electric SA, Hilscher GmbH, u. a.
Offener Bus?	ja, offengelegt
Nutzerorganisation	www.modbus.org www.hilscher.com
Normen	–
Topologie	Linie
Teilnehmer, max. ohne/mit Repeater	RS 485: 32 Teilnehmer
Buszugriffsverfahren	Token-Passing
Übertragungsrate(n) und Leitungslänge(n)	1 MBd, 500 m je Bussegment
Telegrammformat(e)	1 Byte Start, 1 Byte Broadcastadresse, MAC-Datenfeld, 2 Byte CRC, 1 Byte Stop
Datensicherung	16-Bit-CRC
Buspegel	RS 485
Leitung	Zweidrahtleitung verdrillt
Sonstiges	Modbus ist internationaler De-facto-Standard

7.7 Industrial Ethernet

Hersteller	Siemens AG
Offener Bus?	ja
Nutzerorganisation	www.automation.siemens.com/net
Normen	Basis ISO 8802/3 (Ethernet), ISO-Protokollstack, TCP/IP, UDP
Topologie	Linie (Kupfer) Linie, Ring, Stern (LWL)
Teilnehmer, max. ohne/mit Repeater	100 je Segment 1024 für das gesamte Netz
Buszugriffsverfahren	CSMA/CD
Übertragungsrate(n) und Leitungslänge(n)	10 MBd 500 m ohne Repeater > 4 km mit Sternkoppler > 100 km mit Switching
Telegrammformat(e)	72 Bytes ... 1526 Bytes total Ethernet
Datensicherung	4-Byte-CRC
Buspegel	high: 0 V, low: -2,05 V, idle: -2,05 V
Leitung	Triaxkabel (Koaxialkabel mit doppelter Schirmung), LWL, Industrial TP
Sonstiges	Industrial Ethernet ist Teil des Gesamtkonzepts PROFINet

7.8 LON (Local Operating Network)

Hersteller	Echelon (Technologieträger), Toshiba und Cypress (Neuron Chips), weltweit sehr viele Produktanbieter (LON-Geräte, Netzwerk-Komponenten, Tools)
Offener Bus?	ja
Nutzerorganisation	LONMARK Interoperability Association (weltweit) LON Nutzerorganisation e.V. (Deutschland) c/o TEMA, Junkerstr. 77, 52064 Aachen u. a. www.lonmapk.de
Normen	EIA-709.1 Control Network Specification. March 1998
Topologie	alle Topologien (Ring, Linie, Stern, Baum, freie Topologie)
Teilnehmer, max.	medien-, transceiver- und topologieabhängig (z. B. 64 Teilnehmer pro Segment in freier Topologie, TP, ohne Repeater), max. 32385 Teilnehmer
Buszugriffsverfahren	predictive p-persistent CSMA, optional prioritätsgesteuert
Übertragungsraten und Leitungslängen	medien-, transceiver- und topologieabhängig von 1,2 KBAud bis 1250 KBAud (z. B. 78 KBAud, freie Topologie, 500 m; 1250 KBAud, Linientopologie, 130 m)
Telegrammformate	Standard-Daten-Telogramme mit normierten globalen Variablen (bis 31 Byte Daten), sonst Datenfeld ohne Begrenzung, verschiedene Dienste und Adressierungsmöglichkeiten
Datensicherung	Sicherungsfeld nach CCITT-CRC-16-Standard: $x^{16} + x^{12} + x^5 + 1$
Buspegel	transceiverabhängig, typisch Differential-Manchester-Kodierung und transformatorentkoppelt für TP
Leitung	Übertragungsmedien: verdrillte Zweidraht-Leitung, 230 V AC-Leitung, Koaxialkabel, Lichtwellenleiter, Infrarot, Funkkanal, Ultraschall
Sonstiges	LonWorks-Protokoll ist eine Komponente der LonWorks-Technik (verteiltes Automatisierungssystem mit intelligenten LON-Geräten); Betriebssystem für LonWorks-Tools (LNS <i>LonWorks Network Service</i>)

7.9 CAN (Controller Area Network)

Hersteller	Chips: Intel, Philips, Semiconductors, Motorola, NEC, Siemens usw. Systeme: I + ME, Softing, ESD, Bosch, Daimler Benz, Schneider Automation, u. a.
Offener Bus?	ja
Nutzerorganisation	CAN in Automation (CiA e.V., 81058 Erlangen) www.can-cia.org
Normen	ISO/DIS 11519-1 (Schicht 2) ISO/DIS 11898, 9141 (Schicht 1)
Topologie	Linie mit kurzen Stichleitungen
Teilnehmer, max. ohne/mit Repeater	ohne Repeater: 32; mit Repeater: unbegrenzt
Buszugriffsverfahren	prioritätsgesteuert CSMA/CA
Übertragungsrate(n) und Leitungslänge(n)	1 MBd, 40 m 50 kBd, 1000 m
Telegrammformat(e)	Nutzdaten 8 Byte
Datensicherung	15 Bit-CRC, HD = 6
Buspegel	RS 485, modifiziert (falls ISO/DIN 11898)
Leitung	Zweidrahtleitung verdrillt, geschirmt
Sonstiges	ursprünglich als Car Area Network für Pkw konzipiert

7.10 Foundation Field Bus

Hersteller	National Instruments, Pepperl + Fuchs, Schneider Automation, u. v. a. (vor allem in USA)
Offener Bus?	ja
Nutzerorganisation	Fieldbus Foundation 9390 Research Blvd. Austin, Texas 78759-9780 http://www.fieldbus.org/information/
Normen	Schicht 2: ISA S50.02 - 1992 IEC 1158-2
Topologie	Linie
Teilnehmer, max. ohne/mit Repeater	32
Buszugriffsverfahren	Token Passing mit Zugriffsverwalter (Arbitrator) LAS (Link Active Scheduler)
Übertragungsrate(n) und Leitungslänge(n)	31,25 kBd: 1900 m; 1,0 und 2,5 MBd: 750 m
Telegrammformat(e)	4...25 Byte Daten 11...276 Byte total
Datensicherung	16 Bit FCS
Buspegel	31,25 kBd: ±10mA an 50 Ohm -> 1 Vss 1 und 2,5 MBd: ±60 mA an 750 Ohm -> 9 Vss
Leitung	verdrillte Zweidrahtleitung, STP
Sonstiges	31,25 kBd und 1 MBd auch als EX-i Bus mit Stromversorgung über Busleitung

7.11 Eigensichere Feldbusse

7.11.1 PROFIBUS PA (siehe Abschn. 7.4.2)

7.11.2 PROFIBUS (DP) Ex-i

Hersteller	R. Stahl Schaltgeräte GmbH D-74653 Künzelsau u. a.														
Offener Bus?	ja														
Nutzerorganisation	PROFIBUS-Nutzerorganisation D-76131 Karlsruhe														
Normen	DIN 19 245, Teil 1 und 3														
Topologie	Linienstruktur														
Teilnehmer, max. ohne Repeater	pro Ex i-Segment: 13, im Gesamtnetzwerk: max. 126														
Buszugriffsverfahren	hybrides Verfahren: Master/Slave + Token passing														
Übertragungsrate(n) und Leitungslänge	<table border="1"> <thead> <tr> <th>Kupfer</th> <th>DIN 19 245 Teil 3</th> </tr> </thead> <tbody> <tr> <td>9,6 kBd</td> <td>1200 m</td> </tr> <tr> <td>19,2 kBd</td> <td>1200 m</td> </tr> <tr> <td>93,75 kBd</td> <td>1200 m</td> </tr> <tr> <td>187,5 kBd</td> <td>1000 m</td> </tr> <tr> <td>500 kBd</td> <td>400 m</td> </tr> <tr> <td>1500 kBd</td> <td>200 m</td> </tr> </tbody> </table>	Kupfer	DIN 19 245 Teil 3	9,6 kBd	1200 m	19,2 kBd	1200 m	93,75 kBd	1200 m	187,5 kBd	1000 m	500 kBd	400 m	1500 kBd	200 m
Kupfer	DIN 19 245 Teil 3														
9,6 kBd	1200 m														
19,2 kBd	1200 m														
93,75 kBd	1200 m														
187,5 kBd	1000 m														
500 kBd	400 m														
1500 kBd	200 m														
Telegrammformat	siehe PROFIBUS – feste Telegrammlänge entfällt														
Datensicherung	Längs- und Querparität, HD = 4														
Buspegel	RS 485 (Ex-i)														
Leitung	Zweidrahtleitung verdrillt, geschirmt LWL														
Sonstiges	Eigensichere Variante des PROFIBUS DP zum Einsatz in explosionsgefährdeten Bereichen. Ex-i-Segmente über Feldbus-Trennübertrager 9373/21 von nicht Ex-i-Segmenten galvanisch getrennt.														

7.12 DeviceNet

Hersteller	Allen Bradley/Rockwell Automation, Balluff, Beckhoff, Bosch-Rexroth, Cutler-Hammer, Danfoss, ESD, Festo, Fraba, Hilscher, HMS, Lumberg, Omron, National Instruments, SEW, TR, Turck, TWK, Wago, Woodhead-SST/Applicom, u.v.a.m. (über 300)
Offener Bus?	ja
Nutzerorganisation	ODVA (Open DeviceNet Vendor Association) www.odva.org
Normen	CAN / ISO 11898, EN50325, IEC 62026
Topologie	Bus mit Stichleitungen
Teilnehmer	max. 64
Buszugriffsverfahren	CSMA/NBA (prioritätsgesteuert) mit Producer/Consumer-Services
Übertragungsraten und Leitungslängen	500 kbit/s (100m), 250 kbit/s (250m), 125 kbit/s (500m), (erweiterbar mit Repeater)
Telegrammformate	0–8 Byte Nutzdaten in einem CIP-Rahmen, längere Datenpakete (bis 64 kB) werden fragmentiert. CIP ist das objekt-orientierte Application-Layer-Protokoll (Control & Information Protocol).
Datensicherung	16 Bit CRC, Hammingdistanz = 6
Buspegel	Signalleitungen: mod. RS485, siehe CAN; Energie: 24V DC max 8 A.
Leitung	4 Leitungen paarig verdrillt (2xSignal und 2xEnergie) als Rundkabel (geschirmt) oder Flachkabel (ungeschirmt)
Sonstiges	Duplicate MAC-ID Check; implizite I/O-Messages (Cyclic/Change-of-State / Strobe/Polling) sowie explizite Messages: Peer-to-Peer

7.13 ControlNet

Hersteller	ABB, Allen Bradley / Rockwell Automation, Cutler-Hammer, Endress & Hauser, Hilscher, HMS, Honeywell, Pilz, Pyramid Solutions, Yaskawa, u.v.a.m.
Offener Bus?	ja
Nutzerorganisation	ControlNet International www.odva.org
Normen	IEC 61158, EN 50170
Topologie	Bus, Baum, Stern
Teilnehmer	48/99 (ohne/mit Repeater)
Buszugriffsverfahren	CTDMA (kollisionsfrei) mit Producer/Consumer-Services. Synchrone Protokoll ohne Busmaster.
Übertragungsraten und Leitungslängen	5 Mbit/s über alle Leitungslängen (Segment bis 1000 m); maximal 30 km mit (Opto-)Repeater
Telegrammformate	0–510 Byte Nutzdaten (LPacket) in einem CIP-Rahmen. CIP ist das Application-Layer-Protokoll (Control & Information Protocol). Mehrere LPackets pro CIP-Rahmen möglich.
Datensicherung	16 Bit CRC (Polynom $x^{16} + x^{12} + x^5 + 1$, wie bei SDLC), besondere Start- und Ende-Delimiter, HD=4
Buspegel	min. 510 mV pp
Leitung	Coax RG6 (vierfach abgeschirmt) mit 75 Ohm Abschlusswiderständen, LWL (auch in Kombination möglich)
Sonstiges	Multicast, Redundanz, Einsatz bis in EX-Zone möglich, Network Access Ports (NAP) für PG an jedem Knoten

7.14 EtherNet/IP

Hersteller	Allen Bradley / Rockwell Automation, HMS, Hilscher, Mettler Toledo, Pyramid Solutions u. a.
Offener Bus?	ja
Nutzerorganisation	ODVA www.odva.org
Normen	Basis ISO 8802.3 (Ethernet CSMA/CD) Transport: UDP / TCP / IP (RFC bei IETF) Application Layer: CIP (IEC61158, EN50170)
Topologie	Shared Ethernet: Bus oder Stern Switched Ethernet: Stern (empfohlen für zeitkritische Anwendungen)
Teilnehmer	theoretisch unbegrenzt
Buszugriffsverfahren	CSMA/CD mit Producer/Consumer-Services
Übertragungsraten und Leitungslängen	10/100 Mbit/s (100 m zwischen Switch und Endgerät); Switches kaskadierbar
Telegrammformate	0–510 Byte Nutzdaten in einem CIP-Rahmen, eingepackt in ein Standard TCP oder UDP Frame. CIP ist das Application-Layer-Protokoll (Control & Information Protocol).
Datensicherung	4 Byte CRC
Buspegel	H: 0 V, L: -2,05 V, idle: -2,05 V
Leitung	UTP / STP (CAT5)
Sonstiges	Direkter I/O-Datenverkehr wird unterstützt; Peer-to-peer Messaging, I/O-Multicast, Embedded Webserver (produktabhängig)

Stichwortverzeichnis

- 4B5B-Codierung, 75
8QAM, 346
10 MBd Ethernet, 73
10GigaBit Ethernet, 78
20 mA-Stromschleife, 66
100 MBd-Ethernet (Fast Ethernet), 75
1000Base Ethernet, 76
- A**
Abschlusswiderstand, 81
Adressierung mit Slot und Index, 196
ADSL (asymmetrisches DSL, 343
AFP, 59
Anwendungsschicht, 14
Application Layer, 14
ARP, 279
ARQ, 31
AS, 98
AS-i Slaves, 380
AS-i/Interbus-Gateway, 365
ASK, 60
AWG, 82
AWL, 98
azyklische Daten, 260
azyklischen Datenverkehr, 191
- B**
Basisbandübertragung, 87
Baumstruktur, 6
Bipolar-Kodierung, 57
Bit Stuffing, 45
Bitfehlerrate, 31
Bitübertragungsschicht, 11
Blocksicherung, 35
Bluetooth, 333
Breitbandübertragung, 88
- Bridges, 90
Busarbitration, 29
Bus-Struktur, 3
Buszugriffsverfahren, 19
Buszykluszeit, 21, 108
- C**
CAN, 229
CANopen, 234
Carrier Sense Multiple Access, 25
CENELEC, 149
Cheapernet, 272
CIP, 301
Collision Avoidance, 28
Collision Detection, 26
COM, 351
CPF, 146
CRC, 37
CSMA, 25
CSMA/CA, 28, 233
CSMA/CD, 26, 232, 272
Cyclic Redundancy Check, 37
- D**
Darstellungsschicht, 14
Data Link Layer, 11
Datagram, 92
Datenintegrität, 32
Datenintegritätsklassen, 40
Datensicherung, 30
DCOM, 351
DDE, 101, 351
Deterministische Bussysteme, 106
Device Description Language, 277
Devicenet, 248
dezentrale Peripherie, 371

- Diagnosefunktionen, 192
DIX-Standard, 271
DKE, 145
DLL, 101
DMT-Verfahren, 344
DSL – Digital Subscriber Line, 343
- E**
Effizienz, 33
Eigensicherheit, 70, 72, 277
EN 954-1, 197
Enumeration, 332
EtherCAT, 316
Ethernet, 53
EtherNet/IP, 301
Ethernet-Standard, 271
ETHERNET-TCP/IP, 137
Ethernet-Telegramm, 53
Ethernet-Übertragungsarten, 72
- F**
Fabrikautomatisierung, 122
Fast Ethernet, 272
FBS, 98
FDT-Konzept, 277
Fehlerarten, 31
Fehlererkennung, 34
FIP, 149
Firewire (= IEEE 1394-Bus, = iLink), 327
FISCO-Modell, 67
Flag-Byte, 45
Fluid Power, 201
Formatklassen, 43
Frame Check Sequence FCS, 49
Frequency-Shift-Keying, 51, 87
FSK, 60
FTP, 275
Function Block, 260
- G**
Gateway, 94
Gebäudeautomation, 218
Gigabit Ethernet, 76
Gleichstrombehaftete Kodierungen, 56
Gleichstromfreie Kodierungen, 58
GSD, 353
GSD-Datei, 376
- H**
Hamming-Distanz, 32
- Handshake, 62
HART, 142
HART-Protokoll, 49
 HDB_n , 57
 HDB_n -Format, 57
HDLC, 45
HDLC-Protokoll, 44
Host-Rechner, 95
HTML-Seite, 362
HTTP, 275
HTTP-Funktion, 304
HTTP-Server, 364
Hub, 78, 273, 303, 328
- I**
 I_{FDE} , 71
 I_{signal} , 71
Ident Systeme, 201
IEC, 145
IEC 61508, 197
Industrial Ethernet, 270
INTERBUS-S, 149
Internetanbindung, 360
Internet-Netzwerk, 358
Internet-Server, 358
Intrinsic Safety, 277
IP, 134
ISO/OSI-Referenzmodell, 9
isochron, 327
- J**
Jitter, 319
- K**
Kategorie 4, 197
KNX, 167, 390
Koaxialleitung, 84
Kollisionsdomäne, 274
Kollisionserkennung, 27, 28
Konfiguration des OPC-Servers, 354
Konfiguration des PROFIBUS, 353
Konfigurationswerkzeug (FF), 261
KOP, 98
- L**
LabView, 358
LAN, 131
Leitungsarten, 79
Leitungslänge, 83
Leitungsschleifenwiderstand, 64

- letzte Meile, 344
Lichtwellen-Leiter, 191
Lichtwellenleiter, 86
Linienstruktur, 3, 4
LON, 214
LonWorks-Technik, 214
LOOKOUT, 354
- M**
Manchester-II, 59
Manchester-II-Kodierung, 59
Manufacturing Message Spezification, 105
MAP, 129
Master/Slave-Verfahren, 20
M-Bus, 142
Meldungsverzögerung, 110
MLT-3, 75
MMS, 105, 130
Modbus-ASCII, 212
Modbus-RTU, 212
Modbus-TCP, 314
Monomode-LWL, 87
Multimode-LWL, 87
Multiplexer, 2
- N**
Network Layer, 12
Netz-Topologie, 190
Netzwerkhierarchien, 117
Netzwerkschicht, 12
NEURON C, 217
Neuron-Chip, 214
Nichtdeterministische Bussysteme, 109
Node, 223
NRZ, 56
NRZI, 58
NRZ-Verfahren, 57
numerische Apertur, 86
- O**
OLE, 101
OPC, 101, 351
OPC-Client, 352
OPC-Server, 352
OSI-Modell, 10
- P**
PAM 5-Codierung, 77
Paralleldrahtleitung, 80
Paritätsbit, 35
- Performance Level (PL), 197
PHY-Baustein, 73
Physical Layer, 11
polling, 20
POTS-Signal (Plain Old Telephone Service), 344
Powerlink, 305
PPPoE – PPP over Ethernet, 347
Presentation Layer, 14
PROFIBUS, 48, 149, 186
 DP, 187
 DPM, 192
 Nutzerorganisation, 186
PROFIBUS PA, 278
PROFIBUS-Master CP342-5, 371
PROFIdrive, 199
Profile, 188
PROFIsafe, 201
Programmierung von SPS, 98
Prozessautomatisierung, 123
Prüfsumme, 49
PSK, 60
- Q**
QAM (Quadratur-Amplitudenmodulation), 344, 346
- R**
Reaktionszeit, 21
Repeater, 89, 190, 303
Restfehler, 41
Restfehlerrate, 32
RG 58, 84
Ringstruktur, 7
RJ-45, 74
RJ-XX (Registered Jack), 74
Router, 92
RS 232, 60
RS 232-Schnittstelle, 61
RS 422, 63
RS 485, 63, 189
RZ, 56
RZ-Verfahren, 57
- S**
SCADA, 354
Scheduled Messages, 260
Scheduling, 262
Schleifenwiderstand, 82, 83
Schutzart „i“, 113

- SCNM, 308
SDLC, 47
Segmentkoppler, 91
Sercos, 178, 391
Service Access Points, 13
Session Layer, 13
shielded twisted pair, 80
Sicherheitsbussysteme, 110
Sicherheitstechnik, 201
Sicherungsschicht, 11
Sitzungsschicht, 13
Slave-Querverkehr, 195
Smart-Transmitter, 50
SNAP-Frame, 54
SNMP, 275
SNVT, 218
Splitter, 345
SPS-Ankopplung, 96
ST, 99
STEP7, 373
Sternkoppler, 273
Sternstruktur, 8
Store-and-Forward, 273
Switch, 78, 273, 303, 305
System Management, 260, 262
- T**
Tagged MAC-Frame, 55
Taktsynchronisation, 192
TC 65CX, 149
TCP, 135
TCP/IP, 18
TCP/IP-Protocol, 17, 55
Telegrammformate, 41, 44
Thin Ethernet, 272
Token-Bus, 23
Token-Passing, 24
Token-Prinzip, 22
Token-Ring, 23, 53
- Token-Telegramm, 53
Trägerfrequenzübertragung, 87
Transducer Block, 276
Transmission Control Protocol (TCP), 17
Transmitter, 50
transparente Codes, 31
Transport Control Protocol (TCP), 275
Transport Layer, 13
Transportschicht, 13
Trellis-Code-Modulation, 77
Twisted-Pair-Leitungen, 272
- U**
UART, 47
Übertragungsrate, 5
UDP/IP, 301
Uhrzeitsynchronisation, 196
Unscheduled Messages, 260
unshielded twisted pair, 80
USB, 327
USB 2.0, 329
USB-Chip, 332
User Datagram Protocol (UDP), 275
- V**
V.24, 60
- W**
WAN, 131
Wellenwiderstand, 80, 81
Windows, 101
- Y**
Yellow Cable, 272
- Z**
Zweipunktverbindungen, 1
Zyklische Daten, 260
zyklischer Datenaustausch, 377
Zykluszeit, 107