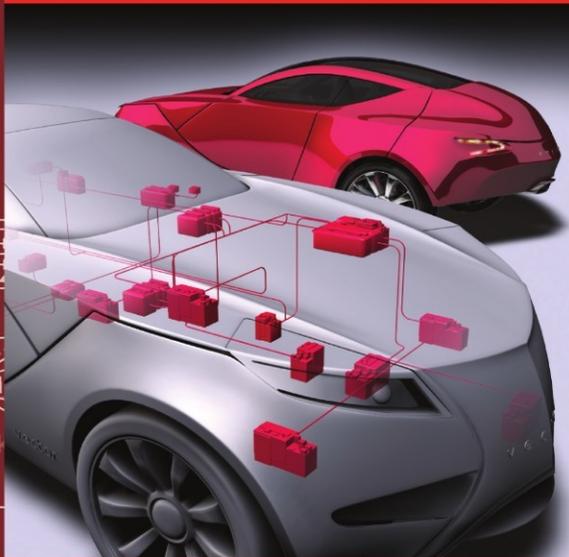


Werner Zimmermann
Ralf Schmidgall

Bussysteme in der Fahrzeugtechnik

Protokolle und Standards

2. Auflage



ATZ/MTZ-Fachbuch

Werner Zimmermann
Ralf Schmidgall

**Bussysteme in der
Fahrzeugtechnik**

Aus dem Programm

Kraftfahrzeugtechnik

Handbuch Verbrennungsmotor

herausgegeben von R. van Basshuysen und F. Schäfer

Lexikon Motorentchnik

herausgegeben von R. van Basshuysen und F. Schäfer

Vieweg Handbuch Kraftfahrzeugtechnik

herausgegeben von H.-H. Braess und U. Seiffert

Bremsenhandbuch

herausgegeben von B. Breuer und K. H. Bill

Nutzfahrzeugtechnik

herausgegeben von E. Hoepke und S. Breuer

Aerodynamik des Automobils

herausgegeben von W.-H. Hucho

Verbrennungsmotoren

von E. Köhler und R. Flierl

Passive Sicherheit von Kraftfahrzeugen

von F. Kramer

Automobilelektronik

herausgegeben von K. Reif

Automotive Software Engineering

von J. Schäuuffele und T. Zurawka

Handbuch Kraftfahrzeugelektronik

herausgegeben von H. Wallentowitz und K. Reif

Die BOSCH-Fachbuchreihe

- **Ottomotor-Management**
- **Dieselmotor-Management**
- **Autoelektrik/Autoelektronik**
- **Sicherheits- und Komfortsysteme**
- **Fachwörterbuch Kraftfahrzeugtechnik**
- **Kraftfahrtechnisches Taschenbuch**

herausgegeben von ROBERT BOSCH GmbH

vieweg

Werner Zimmermann
Ralf Schmidgall

Bussysteme in der Fahrzeugtechnik

Protokolle und Standards

Mit 188 Abbildungen und 99 Tabellen

2., aktualisierte und erweiterte Auflage

ATZ/MTZ-Fachbuch



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

Umschlaggestaltung unter Verwendung eines Fotos der Firma Vector Informatik GmbH, Stuttgart.

1. Auflage April 2006
2., aktualisierte und erweiterte Auflage April 2007

Alle Rechte vorbehalten
© Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Wiesbaden 2007

Lektorat: Reinhard Dapper

Der Vieweg Verlag ist ein Unternehmen von Springer Science+Business Media.
www.vieweg.de



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung: Ulrike Weigel, www.CorporateDesignGroup.de
Technische Redaktion: Fromm MediaDesign GmbH, Selters/Ts.
Druck und buchbinderische Verarbeitung: Wilhelm & Adam, Heusenstamm
Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.
Printed in Germany

ISBN 978-3-8348-0235-4

Vorwort

Dieses Buch soll einen Überblick über die wichtigsten heute in Kraftfahrzeugen eingesetzten Bussysteme und Protokolle geben. Die Betrachtung erfolgt aus Sicht des Anwenders, der als Entwickler von Fahrzeugen und Steuergeräten solche Bussysteme einsetzen will. Ziel ist es, einen ersten Überblick über die nahezu unübersehbare Vielfalt von öffentlichen und proprietären Standards und deren Einsatzgebiete zu geben und demjenigen, der selbst Buskomponenten entwickeln muss, den Einstieg in die verfügbaren Standard-Dokumente zu erleichtern. Neben den eigentlichen, aus zahlreichen Veröffentlichungen mehr oder weniger bekannten Bussystemen wie CAN, LIN, FlexRay oder MOST liegt der Schwerpunkt der Darstellung dabei auf den auf diesen Bussystemen aufsetzenden, höheren Anwendungsschichten.

Die Welt der Kfz-Bussysteme kann als zweigeteilt betrachtet werden. Busse und Protokolle wie ISO 9141 (K-Line) oder ISO 11898 (CAN) und neuere Entwicklungen wie FlexRay, LIN oder MOST wurden ursprünglich von europäischen Kfz-Herstellern und Zulieferern konzipiert, meist auf Initiative der deutschen Hersteller. Viele Lösungen sind von der ISO zwar inzwischen als internationaler Standard anerkannt, werden aber immer noch hauptsächlich in Fahrzeugen europäischer Hersteller eingesetzt. Die amerikanischen Hersteller dagegen haben eigenständige Lösungen entwickelt und diese als SAE Standards, z.B. SAE J1850 oder J1939, etabliert. Asiatische Hersteller verhalten sich pragmatisch und setzen in Europa auf ISO- und in Amerika auf SAE-Standards. Erst als Folge der Globalisierung und des wachsenden Kostendrucks wächst der Zwang zur Vereinheitlichung. So hat die amerikanische Regierung bereits vor einigen Jahren eine Variante von ISO 9141 als zu SAE J1850 gleichberechtigte Schnittstelle für Abgasprüfungen akzeptiert. Beide sollen in zukünftigen Fahrzeugen durch eine auf ISO 11898 (CAN) beruhende Lösung ersetzt werden. Ein anderes Beispiel ist SAE J1939, eine auf CAN basierende Protokollfamilie, die weltweit von praktisch allen Herstellern von Nutzfahrzeugen eingesetzt wird. In diesem Buch stehen die in Europa üblichen Busse und Protokolle im Vordergrund (Kapitel 3 und 4), SAE-Protokolle werden nur am Rande betrachtet.

Bei den *höheren* Protokollen werden sowohl die gängigen Diagnoseprotokolle wie KWP 2000 oder UDS (Kapitel 5) als auch die für die Applikationsphase wichtigen Protokolle wie CCP und XCP erläutert (Kapitel 6). Mindestens genauso wichtig wie der schnelle Datenaustausch zwischen den elektronischen Systemen im Fahrzeug ist aber auch der Informationsfluss zwischen den an der Entwicklung, Fertigung und Wartung eines Fahrzeugs beteiligten Stellen. Gerade in diesem Bereich werden in den verschiedenen Herstellervereinigungen von OSEK, ASAM über HIS bis AUTOSAR viele Standards erarbeitet, um die weiter steigende Komplexität der vernetzten Systeme im Fahrzeug zu beherrschen und mit Hilfe von konsistenten Spezifikationen und Beschreibungen sowie einer automatisierten Datenhaltung Fehler zu vermeiden. Die wichtigsten dieser Standards werden in den Kapiteln 6 und 7 beschrieben, bevor im Kapitel 8 einige darauf basierende Entwicklungswerkzeuge vorgestellt werden.

Bussysteme in Kraftfahrzeugen haben natürlich Berührungspunkte mit den in der Automatisierungstechnik üblichen Feldbussystemen, z.B. Profinet, Interbus, Device-Net oder CANopen, und stammen wie diese von den im Computerbereich üblichen

LANs (Local Area Networks) wie Ethernet oder dem heute nahezu verschwundenen Token Ring ab. Beide werden hier bewusst ausgeklammert, da dazu hinreichend viel Literatur existiert.

Im Text werden naturgemäß sehr häufig Normen und Standardschriften erwähnt und zitiert. Aufgrund der Vielzahl wurde darauf verzichtet, an all diesen Stellen explizite Literaturhinweise anzugeben. Vielmehr finden sich am Ende der Kapitel 3 bis 7 Tabellen mit den jeweils einschlägigen Literaturstellen. Bezugshinweise, bei Industriestandards hauptsächlich Verweise in das World Wide Web, sowie allgemeine Literaturhinweise und eine Liste der wichtigsten Abkürzungen finden sich im Anhang.

Ein steter Diskussionspunkt bei deutschsprachiger Technikliteratur ist die Verwendung englischsprachiger Begriffe. Da es bei ernsthafter Beschäftigung mit dem Thema unvermeidbar ist, die Originaldokumente der Spezifikationen und Normen zu Rate zu ziehen und diese praktisch ausschließlich in Englisch verfügbar sind, haben die Autoren bewusst darauf verzichtet, die normspezifischen Fachbegriffe ins Deutsche zu übersetzen, obwohl es in vielen Fällen durchaus gute deutsche Begriffe gegeben hätte. In der Regel werden bei der ersten Erwähnung der englische Begriff und eine passende deutsche Übersetzung angegeben. Anschließend wird dann der Originalbegriff verwendet. Auf diese Weise sollte der *Wiedererkennungswert* beim Lesen der englischsprachigen Texte im Anschluss an diese Einführung deutlich erhöht werden.

Soweit im Text Hersteller- und Produktnamen verwendet werden, geschieht dies ohne ausdrückliche Erwähnung von eingetragenen Markennamen und Markenrechten.

Danksagungen

Dieses Buch geht auf eine Anregung von *Wolfgang Schmid*, SMART Electronic Development GmbH zurück, ohne dessen beharrliches Drängen die Autoren nie den Mut gehabt hätten, in diesem sich mit hoher Geschwindigkeit weiterentwickelnden Gebiet ein Buch anzugehen, das mehr als eine flüchtige Momentaufnahme sein soll.

Joachim Tauscher und *Wolfgang Neu*, Geschäftsführer der SMART Electronic Development GmbH, und ihren Mitarbeitern herzlichen Dank für ihre stete Unterstützung über viele Jahre hinweg und für ihren Beitrag zum Kapitel 8.4.

Hans-Dieter Kübler, Geschäftsführer der Samtec Automotive Software & Electronics GmbH, und seinen Mitarbeitern *Axel Weimann*, *Daniel Oedingen* und *Thomas Januschke* Dank für ihre Anregungen und ihren Beitrag zu den Kapiteln 8.5 und 8.7.

Dank schulden wir auch allen Kolleginnen, Kollegen und Studierenden an der FH Esslingen – Hochschule für Technik, bei der Robert Bosch GmbH und der DaimlerChrysler AG, insbesondere aber den Mitarbeitern am STZ Rechnerersatz Esslingen, die in vielen Projekten mitgeholfen haben, die in diesem Buch beschriebenen Konzepte und Verfahren praktisch umzusetzen, die neuesten Informationen zu beschaffen und das Manuskript kritisch durchzusehen.

Nicht zuletzt gilt unser Dank *Ewald Schmitt* sowie *Reinhard Dapper* vom Vieweg Verlag und ihren Mitarbeitern sowie allen ungenannten Helfern, die in vielfältiger Weise zu diesem Buch beigetragen haben.

Vorwort zur 2. Auflage

Der überraschend große Erfolg der 1. Auflage und die weiterhin stürmische Entwicklung der Automobilelektronik machen es möglich, nach kaum einem Jahr bereits eine zweite, aktualisierte und erweiterte Auflage herauszugeben. Neben einer ganzen Reihe von Detailverbesserungen und der Korrektur einiger Fehler, die trotz aller Sorgfalt anscheinend unvermeidlich sind, wurden folgende Themen in der 2. Auflage neu aufgenommen oder wesentlich erweitert:

- MOST

Durch die neu hinzugekommene Darstellung der *MOST Network Services* und die ausführlichere Beschreibung des gesamten Protokolls hat sich der Umfang dieses Kapitels verdoppelt. Diese Verbesserungen verdanken wir vor allem *Steffen Lang* von der BOSE Automotive GmbH, der die neuen Teile des Kapitels 3.7 beigesteuert hat.

- FlexRay

Neu aufgekomen wurden das Transportprotokoll *AUTOSAR TP für FlexRay* in Kapitel 4.2 sowie das Applikationsprotokoll *XCP on FlexRay* in Kapitel 6.2.

- AUTOSAR 2.x

Nachdem das AUTOSAR Konsortium seine Spezifikationen als Version 2.0 Mitte 2006 erstmals öffentlich zugänglich gemacht und bis Ende 2006 weiter vervollständigt hat, hat die AUTOSAR Architektur inzwischen einen Zustand erreicht, der nun im Kernbereich der Basissoftware und der RTE-Schicht weitgehend stabil sein dürfte. Daher wurde das bisher recht kurze Kapitel 7.6 auf nun über 25 Seiten erweitert und AUTOSAR 2.x im Detail vorgestellt.

- In allen übrigen Themenbereichen wurden die jeweils neuesten verfügbaren Spezifikationen berücksichtigt, darunter LIN 2.1, FIBEX 2.0, ODX 2.1 und MVCI nach ISO 22900 bzw. 22901, Flash-Lader nach HIS 1.1 und die Pass-Through-Programmierung nach ISO 23248 / SAE J2534.

Joachim Tauscher, SMART Electronic Development GmbH, mit den Abschnitten 8.4.4 und 8.4.5 sowie *Thomas Januschke*, Samtec Automotive Software & Electronics GmbH, mit den Abschnitten 8.5.1, 8.6.1 und 8.7.1 haben dazu beigetragen, dass wir nicht nur die neuesten Protokollstandards darstellen, sondern auch das Kapitel über die Entwicklungswerkzeuge aktualisieren und erweitern konnten.

Alle, denen wir schon bei der ersten Auflage großen Dank geschuldet haben, haben uns auch bei der Arbeit an dieser zweiten Auflage unterstützt. Stellvertretend genannt seien wiederum *Ewald Schmitt* und *Reinhard Dapper* vom Vieweg Verlag.

Unser besonderer Dank aber gilt allen Lesern der ersten Auflage, deren Zuspruch, Anregungen und Hinweise uns ermutigt und dazu beigetragen haben, die zweite Auflage weiter zu verbessern.

Die Autoren

Prof. Dr.-Ing. Werner Zimmermann lehrt Regelungstechnik, Systementwurf, Digital- und Rechnersysteme in der Fakultät Informationstechnik an der Hochschule Esslingen. Zuvor leitete er bei der Robert Bosch GmbH eine Entwicklungsabteilung für Motorsteuergeräte für Dieselfahrzeuge.

Dipl.-Ing. (FH) MSc. Ralf Schmidgall arbeitet in der Entwicklung Fahrzeugdiagnose bei der DaimlerChrysler AG. Zuvor war er bei einem Hersteller von Testsystemen in der Softwareentwicklung für Kommunikationsprotokolle tätig.

Unter Mitwirkung von:

Dipl.-Ing. (BA) Joachim Tauscher, Geschäftsführender Gesellschafter der Smart Electronic Development GmbH, hat das Kapitel 8.4 mit verfasst.

Dipl.-Inf. (FH) Thomas Januschke, verantwortlich für die Entwicklung von Diagnosewerkzeugen bei der Samtec Automotive Software & Electronics GmbH, hat die Kapitel 8.5 bis 8.7 mitgestaltet.

Dipl.-Ing. (FH) Steffen Lang, Entwicklungsingenieur bei der Bose Automotive GmbH, hatte wesentlichen Anteil an der Überarbeitung des Kapitels 3.7 für die 2. Auflage.

Inhalt

Vorwort	V
1 Anwendung von Bussystemen und Protokollen	1
1.1 Überblick	2
2 Kfz-Bussysteme, Protokolle und Standards	5
2.1 Standardisierung bei Bussystemen	7
3 Kfz-Bussysteme – Physical und Data Link Layer	9
3.1 Grundlagen für Kfz-Bussysteme	9
3.1.1 Elektrotechnische Grundlagen.....	9
3.1.2 Topologie und Kopplung von Bussystemen	12
3.1.3 Botschaften, Protokollstapel, Dienste (Services).....	14
3.1.4 Zeichen- und Bitstrom-basierte Übertragung, Nutzdatenrate.....	17
3.1.5 Buszugriffsverfahren, Fehlererkennung und Fehlerkorrektur	19
3.1.6 Jitter und Latenz bei der Datenübertragung	22
3.2 K-Line nach ISO 9141 und ISO 14230	23
3.2.1 Entwicklung von K-Line und KWP 2000	23
3.2.2 K-Line Bus-Topologie und Physical Layer.....	24
3.2.3 Data Link Layer	26
3.2.4 Einschränkungen für emissionsrelevante Komponenten (OBD)	30
3.2.5 Schnittstelle zwischen Protokoll-Software und Kommunikations- Controller	30
3.2.6 Ältere K-Line-Varianten	31
3.2.7 Zusammenfassung K-Line – Layer 1 und 2.....	31
3.3 Controller Area Network CAN nach ISO 11898	32
3.3.1 Entwicklung von CAN	32
3.3.2 Bus-Topologie und Physical Layer	33
3.3.3 Data Link Layer CAN 2.0A und 2.0B	35
3.3.4 Fehlerbehandlung	36
3.3.5 Einsatz von CAN – Höhere Protokolle.....	37
3.3.6 Schnittstelle zwischen Protokoll-Software und Kommunikations- Controller	38
3.3.7 Time-Triggered-CAN (TTCAN) – Deterministischer Buszugriff.....	39
3.3.8 Zusammenfassung CAN – Layer 1 und 2	42
3.4 Local Interconnect Network LIN	43
3.4.1 Überblick	43
3.4.2 Data Link Layer.....	44
3.4.3 Zeitsynchrones Senden von Botschaften (Message Scheduling)	46
3.4.4 Neue Botschaftstypen bei LIN V2.0.....	47
3.4.5 ISO Diagnose über LIN.....	48

3.4.6	LIN Configuration Language	49
3.4.7	Dynamische Konfiguration von LIN Slave-Steuergeräten	52
3.4.8	LIN Application Programming Interface (API).....	54
3.4.9	Zusammenfassung LIN – Layer 1 und 2	56
3.5	FlexRay	57
3.5.1	Bus-Topologie und Physical Layer	57
3.5.2	Data Link Layer	59
3.5.3	Netzwerk-Start und Takt-Synchronisation	62
3.5.4	Fehlerbehandlung, Bus Guardian	64
3.5.5	Konfiguration und übergeordnete Protokolle	65
3.5.6	Zusammenfassung FlexRay – Layer 1 und 2	65
3.6	SAE J1850	66
3.7	Media Oriented Systems Transport MOST	69
3.7.1	Bus-Topologie und Physical Layer	69
3.7.2	Data Link Layer	70
3.7.3	Kommunikationscontroller	74
3.7.4	Network Services	76
3.7.5	Netzmanagement.....	78
3.7.6	Höhere Protokollschichten	79
3.7.7	Beispiel für Systemstart und Audioverbindung	79
3.7.8	Zusammenfassung MOST	82
3.8	Normen und Standards	83
4	Transportprotokolle	85
4.1	Transportprotokoll ISO TP für CAN nach ISO 15765-2	85
4.1.1	Botschaftsaufbau	86
4.1.2	Flusssteuerung, Zeitüberwachung und Fehlerbehandlung	87
4.1.3	Dienste für die Anwendungsschicht (Application Layer Services)	89
4.1.4	Protokoll-Erweiterungen	90
4.1.5	Adressierung bei KWP 2000 – Zuordnung von CAN Identifiern.....	90
4.2	Transportprotokoll AUTOSAR TP für FlexRay.....	91
4.3	Transportprotokoll TP 2.0 für CAN.....	94
4.3.1	Adressierung und CAN Message Identifier	94
4.3.2	Broadcast-Botschaften	95
4.3.3	Dynamischer Kanalaufbau und Verbindungsmanagement	96
4.3.4	Datenübertragung	98
4.4	Transportprotokoll TP 1.6 für CAN.....	100
4.4.1	Botschaftsaufbau	100
4.4.2	Dynamischer Kanalaufbau.....	100
4.4.3	Datenübertragung und Datenrichtungswechsel	101
4.5	Transportprotokoll SAE J1939/21 für CAN	102
4.5.1	Übertragungsarten, Adressierung und CAN Message Identifier	102
4.5.2	Segmentierte Datenübertragung (Multi Packet)	105
4.6	Normen und Standards	108

5 Diagnoseprotokolle – Application Layer	109
5.1 Diagnoseprotokoll KWP 2000 (ISO 14230-3 und 15765-3).....	111
5.1.1 Überblick	111
5.1.2 Diagnosesitzungen (Diagnostic Management).....	114
5.1.3 Adressierung der Steuergeräte nach KWP 2000	116
5.1.4 Bussystem-abhängige Dienste (Network Layer Protocol Control)	118
5.1.5 Fehlerspeicher lesen und löschen (Stored Data Transmission).....	119
5.1.6 Daten lesen und schreiben (Data Transmission), Ansteuern von Steuergeräte-Ein- und Ausgängen (Input/Output Control).....	120
5.1.7 Speicherblöcke auslesen und speichern (Upload, Download)	121
5.1.8 Start von Programmen im Steuergerät (Remote Routine Activation)..	122
5.1.9 Erweiterte Dienste (Extended Services).....	123
5.2 Unified Diagnostic Services UDS nach ISO 14229	123
5.2.1 Unterschiede zum KWP 2000 Diagnoseprotokoll	123
5.2.2 Überblick über die UDS Diagnosedienste	124
5.2.3 Response on Event Dienst	131
5.3 On-Board-Diagnose OBD nach ISO 15031 / SAE J1979	133
5.3.1 Überblick OBD Diagnosedienste	133
5.3.2 Auslesen des Fehlerspeichers und von Steuergerätwerten	135
5.3.3 Abfrage der Testergebnisse für abgasrelevante Komponenten.....	138
5.3.4 OBD Fehlercodes	139
5.3.5 Data Link Security.....	141
5.3.6 Pass-Through Programmierung	141
5.4 Normen und Standards	142
6 Anwendungen für Messen, Kalibrieren und Diagnose (ASAM AE MCD)	143
6.1 Einführung	143
6.2 Busprotokolle für Aufgaben in der Applikation (ASAM MCD 1MC)	147
6.2.1 CAN Calibration Protocol CCP	148
6.2.2 Extended Calibration Protocol XCP.....	156
6.2.3 AML-Konfigurationsdateien für XCP und CCP	171
6.2.4 Interface zwischen Busprotokolltreiber und Applikationssystem ASAM MCD 1b	173
6.3 Field Bus Exchange Format FIBEX	177
6.4 Überblick über ASAM MCD 2 und MCD 3	184
6.5 Applikationsdatensätze nach ASAM MCD 2 MC (ASAP 2)	185
6.6 ODX-Diagnosedatensätze nach ASAM MCD 2D	188
6.6.1 Aufbau des ODX-Datenmodells.....	189
6.6.2 Einfache Datenobjekte (Simple DOP und DTC-DOP).....	191
6.6.3 Beschreibung von Diagnosediensten und Diagnosevariablen	197
6.6.4 Komplexe Datenobjekte (Complex DOP)	201
6.6.5 DIAG-LAYER: Beschreibung der Diagnosedienste eines Gerätes	203
6.6.6 ECU-MEM: Beschreibung der Flash-Programmierung	207
6.6.7 ECU-CONFIG: Beschreibung der Steuergeräte-Konfiguration	209

6.6.8	MULTIPLE-ECU-JOB: Diagnosedienste für mehrere Geräte	210
6.6.9	COMPARAM-SUBSET und COMPARAM-SPEC: Beschreibung des Busprotokolls	211
6.6.10	VEHICLE-INFO-SPEC: Fahrzeugzugang und Bustopologie	213
6.6.11	FUNCTION-DICTIONARY: Funktionsorientierte Diagnose.....	216
6.7	ASAM MCD 3 – Server	217
6.7.1	Funktionsgruppe M – Messen	219
6.7.2	Funktionsgruppe C – Kalibrieren	220
6.7.3	Funktionsgruppe D – Diagnose	221
6.8	MVCI-Diagnosetestsysteme nach ISO 22900	223
6.9	Normen und Standards	226
7	Software-Standards: OSEK, AUTOSAR, HIS,	227
7.1	Einführung	227
7.2	OSEK/VDX	230
7.2.1	Ereignisgesteuerter Betriebssystemkern OSEK/VDX OS	231
7.2.2	Kommunikation in OSEK/VDX COM	241
7.2.3	Netzmanagement mit OSEK/VDX NM.....	245
7.2.4	Zeitgesteuerter Betriebssystemkern OSEK Time und fehlertolerante Kommunikation OSEK FTCOM.....	250
7.2.5	Erweiterung von OSEK OS um Schutzmechanismen:Protected OS ..	253
7.3	Hardware-Ein- und Ausgabe (HIS IO Library, IO Driver).....	254
7.3.1	Hardwaretreiber für Mikrocontroller-Standardperipherie	256
7.4	HIS Hardwaretreiber für CAN-Kommunikationscontroller (HIS CAN Driver). ..	259
7.4.1	Zustandsmodell.....	260
7.4.2	Ablauf beim Empfangen einer CAN-Botschaft	261
7.4.3	Ablauf beim Senden einer CAN-Botschaft	263
7.4.4	Schnittstelle zum Betriebssystem	264
7.5	HIS Flash-Lader	265
7.6	AUTOSAR	266
7.6.1	Überblick über die AUTOSAR-Basissoftware.....	268
7.6.2	Betriebssystem AUTOSAR OS.....	276
7.6.3	Kommunikationsstack AUTOSAR COM, NM und Diagnostic	280
7.6.4	Runtime Environment AUTOSAR RTE	289
7.7	Normen und Standards	293
8	Werkzeuge, Anwendungen und Einsatzgebiete	295
8.1	Protokollstapel für Steuergeräte (Intellectual Property)	295
8.2	Entwurf und Test der On-Board-Kommunikation	295
8.3	Werkzeuge zur Applikation von Steuergeräten.....	299
8.4	Flash-Programmierung von Steuergeräten.....	300
8.4.1	Rahmenbedingungen	301
8.4.2	Flash-Speicher.....	304
8.4.3	Flash-Programmierprozess	306

8.4.4 Beispiel eines Flash-Laders: <i>ADLATUS</i> von SMART Electronic	315
8.4.5 Test und Freigabe von Flash-Ladern und Busprotokollen	320
8.5 Diagnosewerkzeuge in Entwicklung und Fertigung	323
8.5.1 Beispiel für Diagnosewerkzeuge: <i>samDia</i> von Samtec Automotive	325
8.6 Autorenwerkzeuge für Diagnosedaten.....	333
8.6.1 Beispiel für Autorenwerkzeuge: <i>ODXplorer</i> von Samtec Automotive	335
8.7 ASAM MCD3–Laufzeitsysteme.....	341
8.7.1 Beispiel für ein Diagnosesystem: <i>samMCD3Server</i> von Samtec Automotive Software & Electronics	342
Literaturverzeichnis	347
Web-Adressen	347
Abkürzungen	349
Sachwortverzeichnis	352

1 Anwendung von Bussystemen und Protokollen

Bereits seit Einführung der ersten mikroprozessorgesteuerten Systeme wie Motronic und ABS im Kraftfahrzeug ab 1980 war es notwendig, zwischen diesen Systemen und von diesen Systemen nach außen Daten auszutauschen.

Während der Datenaustausch zwischen den Systemen (On Board) zunächst vor allem aus Punkt-zu-Punkt-Verbindungen bestand und über Analogsignale oder einfache Schaltsignale erfolgte, wurde die erste echte Datenkommunikation zum Anschluss des Diagnostesteters in der Kfz-Werkstatt (Off Board) eingeführt. Sehr schnell war eine herstellerübergreifende Lösung nötig, wobei Bosch als in Europa führender Hersteller elektronischer Steuergeräte eine von vielen Fahrzeugherstellern übernommene, später als ISO 9141 standardisierte Spezifikation vorlegte. Diese Spezifikation ähnelte der bei PCs üblichen seriellen RS232C/V24-Schnittstelle und legte zunächst wenig mehr als die Zahl der Verbindungsleitungen, die elektrischen Signalpegel und das Bitformat der Zeichenübertragung fest. Selbst dabei waren noch verschiedene Varianten möglich. Die Bedeutung der übertragenen Daten sowie die in der Werkstatt angewendeten Diagnoseverfahren selbst blieben offen und wurden weiterhin Geräte-, Fahrzeug- und im besten Fall Hersteller-spezifisch unterschiedlich implementiert.

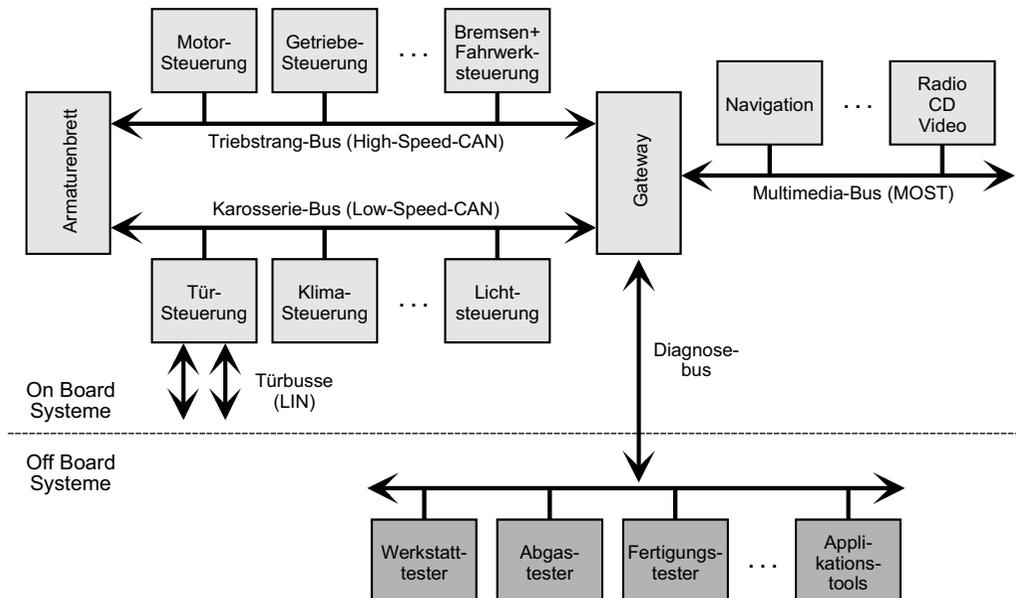


Bild 1.1: Bussysteme eines modernen Mittelklassefahrzeugs mit Vollausrüstung

Mit der Einführung des von Bosch vorgestellten, später unter anderem als ISO 11898 und SAE J1939 standardisierten CAN Busses ab 1990 hielt auch bei der On-Board-Kommunikation zwischen den Steuergeräten innerhalb des Fahrzeugs ein Datennetz

(*Bussystem*) Einzug. Auch hier war, wenn auch wesentlich präziser als bei ISO 9141, zunächst im Wesentlichen die Bitebene spezifiziert, während die Bedeutung der ausgetauschten Daten (*Protokoll*) nicht festgelegt war und immer noch je nach Gerät, Fahrzeug oder Hersteller unterschiedlich implementiert wurde.

Mit dem Siegeszug der Mikroelektronik in modernen Fahrzeugen wurde das System derart komplex und das Datenaufkommen so hoch, dass Neufahrzeuge heute über mehrere, miteinander vernetzte Bussysteme (*Netzwerke*) verfügen (Bild 1.1).

Die Beherrschung dieser Komplexität, der Kostendruck, der Wunsch, Fahrzeuge weltweit anzubieten, sowie Vorschriften der Gesetzgeber zwangen die Fahrzeughersteller und ihre Zulieferer schließlich, nach standardisierten Lösungen für die Bussysteme selbst und die zum Datenaustausch verwendeten Protokolle zu suchen.

1.1 Überblick

Grob lassen sich dabei vier Anwendungsgebiete unterscheiden (Tabelle 1.1):

- **On-Board-Kommunikation** zwischen den Kfz-Steuergeräten im Fahrzeug
Dieses Aufgabenfeld lässt sich heute in drei Teilgebiete unterteilen, für die in der Regel unterschiedliche Buskonzepte verwendet werden:
 - **High Speed Systeme für Echtzeit-Steuerungsaufgaben**
Die Steuerung und Regelung von Motor, Getriebe, Bremsen und Fahrwerk ist nur möglich, wenn die Steuergeräte Sensorinformationen austauschen und Stelleingriffe über mehrere Systeme hinweg koordinieren. Die notwendigen Informationen sind dabei nur wenige Bytes lang, müssen aber periodisch mit hoher Frequenz, sehr kurzer Verzögerungszeit (Latenz) und extrem hoher Zuverlässigkeit übertragen werden. Für diesen Anwendungsbereich z.B. wurde CAN entwickelt. Durch die Einführung von Brake-by-Wire und Steer-by-Wire werden die Anforderungen zukünftig noch steigen. Daher wurde CAN zum Time-Triggered CAN (TTCAN) weiterentwickelt und neue Bussysteme wie FlexRay konzipiert.
 - **Low Speed Systeme zur Kabelbaum-Vereinfachung**
Auch für einfache Aufgaben wie das Ansteuern von Lampen, Fenstermotoren usw. werden zunehmend Bussysteme eingeführt, um den Kabelbaum zu vereinfachen. Da im Wesentlichen nur Schaltsignale übertragen werden, sind die erforderlichen Datenraten niedrig. Hauptgesichtspunkt sind die Kosten je Busknoten. Für diese Aufgaben wird oft eine leicht vereinfachte CAN-Variante oder das speziell hierfür entwickelte neue Bussystem LIN als noch billigere Lösung eingesetzt.
 - **Multimedia-Bussysteme**
Mit der Einführung von Multimedia-Systemen im Fahrzeug für Navigation, Video, Audio und Telefon müssen sehr große Datenmengen beispielsweise zwischen dem CD-Wechsler, dem verteilten Audio-System und dem Display im Armaturenbrett ausgetauscht werden. Da diese Informationen aber nicht für echte Steuer- und Regelaufgaben eingesetzt werden, ist der Hauptge-

sichtspunkt dabei ein hoher Datendurchsatz, während die Anforderungen an die Latenzzeiten und die Übertragungszuverlässigkeit wesentlich geringer sind als bei den High-Speed-Systemen für Echtzeit-Steuerungsaufgaben.

Tabelle 1.1: Anwendungsbereiche und Anforderungen an Bussysteme im Kfz

Anwendung	Bot- schafts- länge	Bot- schafts- rate	Resultie- rende Datenrate	Latenz- zeiten	Fehler- sicherheit	Kosten
On-Board- Kommunikation						
• High-Speed-Steuerung	Kurz	Hoch	Hoch	Sehr kurz	Extrem hoch	Mittel
• Low-Speed-Steuerung	Kurz	Niedrig	Niedrig	Mäßig	Sehr hoch	Sehr niedrig
• Multimedia	Lang	Hoch	Sehr hoch	Mäßig	Mäßig	Hoch
Off-Board- Kommunikation						
• Werkstattdiag- nose	Kurz	Niedrig	Niedrig	Unwichtig	Gering	Niedrig
• Fertigungstest beim Fahrzeug- und beim Gerä- tehersteller inklusi- ve End-of-Line- Programmierung (Flashen)	Lang bis sehr lang	Niedrig	Hoch	Unwichtig	Mäßig	Unwichtig
• Applikation am Fahrzeug und an Prüfständen in der Entwick- lungsphase (Um- programmieren, Kalibrieren)	Kurz	Mittel (hoch bei Messauf- gaben)	Mittel bis hoch	Kurz	Gering	Unwichtig

- **Off-Board-Kommunikation** zwischen Fahrzeug und externen Geräten:
 - **Diagnose-Kommunikation in der Werkstatt und im Abgastest**

Für die Fehlersuche und für Abgasprüfungen in Werkstätten ist eine Kommunikationsschnittstelle zu einem externen Tester notwendig. Die Anforderungen an Datenrate und Fehlertoleranz sind relativ gering. Die Schnittstelle ist teilweise vom Gesetzgeber standardisiert (US OBD, European OBD), darüber hinaus hat jeder Kfz-Hersteller seinen Hausstandard. Veränderungen an dieser Schnittstelle verursachen einen hohen Folgeaufwand in den Werkstätten, da die Werkstatttester weltweit angepasst werden müssen. Steuergeräte unterstützen daher in der Regel verschiedene herstellerspezifische Varianten des Diagnoseprotokolls, wobei sich Steuergerät und Tester automatisch auf eine zu beiden kompatible Protokollvariante verständigen müssen. Zunehmend wird

erwartet, dass die Steuergeräte eines Fahrzeugs in der Werkstatt umprogrammiert werden können. Dabei sind Zugriffsschutzmechanismen wichtig, um sicherzustellen, dass nur die zum Fahrzeug passende, vom Fahrzeughersteller freigegebene Softwarevariante verwendet und die Umprogrammierung nur durch autorisierte Werkstätten durchgeführt wird.

- **Fertigungstest beim Kfz- und beim Steuergeräte-Hersteller**

Kfz-Hersteller versuchen, mit wenigen Steuergeräte-Hardwarevarianten viele Fahrzeugmodelle mit Ausstattungsvarianten und Länderausführungen durch Softwarevarianten im Steuergerät abzudecken. Um die Logistik zu vereinfachen, wird häufig die gesamte oder ein großer Teil der Software erst beim Kfz-Hersteller ins Steuergerät geladen (*Flashen*). Für den Programmiervorgang sowie für die Fertigungsprüfung des Fahrzeugs wird ebenfalls die Diagnose-Schnittstelle eingesetzt. Gelegentlich wird diese dabei in einem Modus betrieben, der einen höheren Datendurchsatz zulässt, um kürzere Zykluszeiten zu erreichen. Da die Werkstatt im Prinzip dieselbe Diagnoseschnittstelle benützt, sind Zugriffsschutzmechanismen notwendig, damit sicherheitskritische Daten des Steuergeräts nicht von Unbefugten manipuliert werden.

- **Applikation am Prüfstand und im Fahrzeug in der Entwicklungsphase**

Während der Anpassung eines Steuergerätes an ein bestimmtes Fahrzeug ist die Durchführung umfangreicher Messungen an Komponentenprüfständen und im Fahrzeug notwendig. Dabei müssen teilweise mehrere tausend Parameter im Steuergerät appliziert werden. Da die Steuergeräte-Hard- und Software bereits in einem seriennahen Zustand sein sollen, erfolgt die Datenerfassung und die Adaption der Parameter in der Regel über die Standardschnittstellen für die On- und Off-Board-Kommunikation, wobei die Schnittstellenprotokolle für diesen Zweck erweitert werden. Da der Applikationsaufwand bei praktisch jeder Fahrzeugvariante neu anfällt, wird inzwischen herstellerübergreifend versucht, die Applikation mit Werkzeugen zu unterstützen und zu automatisieren. Dazu werden Standard-Schnittstellen wie ASAM MCD für den Messdatenaustausch, die Parameterverstellung und die Datenhaltung geschaffen.

Aus dem dargestellten Anforderungsprofil insbesondere bezüglich der Übertragungsgeschwindigkeit ergibt sich eine Einteilung der Bussysteme in verschiedene Klassen (Tabelle 1.2), wobei die Grenzen zwischen den Klassen natürlich fließend sind.

Tabelle 1.2: Klassifikation der Bussysteme nach der Bitrate

Class	Bitrate	Typische Vertreter	Anwendung
Diagnose	< 10 Kbit/s	ISO 9141-K-Line	Werkstatt- und Abgastester
A	< 25 Kbit/s	LIN, SAE J1587/1707	Karosserieelektronik
B	25 ... 125 Kbit/s	CAN (Low Speed)	
C	125 ... 1000 Kbit/s	CAN (High Speed)	Antriebsstrang, Fahrwerk, zunehmend auch Diagnose
C+	> 1 Mbit/s	FlexRay, TTP	Steer and Brake by Wire
Infotainment	> 10 Mbit/s	MOST	Multimedia (Audio, Video)

2 Kfz-Bussysteme, Protokolle und Standards

Wenn von Kfz-Bussystemen gesprochen wird, fallen in der Regel Schlagworte wie CAN, LIN, FlexRay usw. Neulinge identifizieren mit diesen Begriffen hauptsächlich den physikalisch direkt sichtbaren Teil der Kommunikationsschnittstelle mit Steckern, Kabeln und den für die Kommunikation zuständigen Elektronikschaltungen. In Veröffentlichungen werden in diesem Zusammenhang vor allem übertragungstechnische Details auf Bitebene wie Signalpegel, Zugriffsverfahren und die Reihenfolge und Bedeutung der einzelnen Bits auf den Busleitungen ausführlich beschrieben. Die Softwareentwickler, die die Übertragungssoftware implementieren müssen, interessieren sich vor allem für die Programmierschnittstelle der eingesetzten *Buscontroller*, Fragen der Formatierung und Zwischenspeicherung der Daten sowie der Behandlung von Übertragungsfehlern. Für den Anwender dagegen sind vor allem die Bedeutung der übertragenen, eigentlichen Nutzdaten und deren Formate von Interesse, die durch die typischen Spezifikationen von Bussystemen wie CAN oder LIN überhaupt nicht definiert sind.

Um die verschiedenen Aufgabenstellungen bei der Datenkommunikation auseinander zu halten, kann das von der ISO standardisierte Open-System-Interconnect-(OSI)-Schichtenmodell (Tabelle 2.1) dienen, das die Kommunikationshierarchie beschreibt. Die grau markierten Schichten haben für Kfz-Anwendungen (noch) keine Bedeutung.

Tabelle 2.1: OSI-Schichtenmodell für Bussysteme und Protokolle (Datennetze)
Hinweis: Schicht 0 ist keine offizielle Schicht des OSI-Modells.

Layer (Schicht)			Aufgabe
7	Application	Anwendung	Allgemein verwendbare Dienste für den Anwender, z.B. Lesen des Fehlerspeichers usw.
6	Presentation	Darstellung	
5	Session	Sitzungssteuerung	
4	Transport	Datentransport	Aufteilung und Zusammensetzen der Daten mehrerer Botschaften (Segmentierung)
3	Network	Vermittlung	Routing, Adressvergabe, Teilnehmererkennung und -überwachung
2	Data Link	Sicherung	Botschaftsaufbau, Buszugriff, Fehlersicherung, Flusskontrolle
1	Physical	Bitübertragung	Elektrische Signalpegel, Bitcodierung
0	Mechanical	Mechanik	Steckverbinder und Kabel

Vor dem Spiegel realer Standards betrachtet ist das OSI-Modell allerdings oft nur akademischer Natur. Es hilft zwar beim Verständnis, aber die realen Standards definieren häufig nur Teile einer Schicht, fassen Aufgaben mehrerer Schichten in einer Ebene zusammen oder trennen eine Schicht auf mehrere Standards auf. Dazu kommt, dass für dieselbe Aufgabenstellung oft mehrere, voneinander abweichende Standards existieren oder umgekehrt dieselbe technische Lösung in verschiedenen Standards beschrieben wird. Noch unübersichtlicher wird die Situation dadurch, dass die unterschiedlichen Normen, selbst wenn sie vom selben Normungsgremium stammen, Begriffe unterschiedlich, gar nicht oder sogar unterschiedliche Dinge gleich benennen.

Tabelle 2.2: Kfz-Busse (Schicht 0 bis 2), Web-Links zu den Herstellern im Anhang

Bustyp	Anwendung	Europ. Standards	US-Standards
<i>Zeichen-basiert (UART)</i>			
K/L-Line	Diagnose	ISO 9141	
SAE J1708	Diagnose, Class A On Board		SAE J1708 (Truck and Bus) 9.6 kbit/s
LIN	Class A On Board	Herstellerkonsortium 20 kbit/s	SAE J2602
<i>PWM-basiert</i>			
SAE J1850	Diagnose, Class A/B On Board		SAE J1850 (PWM Ford, VPWM GM, Chrysler) 10,4 und 41,6 kbit/s
<i>Bitstrom-basiert</i>			
CAN	Class B/C On Board	ISO 11898 1-3 Bosch CAN 2.0 A, B 47.6 ... 500 kbit/s ISO 11992 CAN für Zugfahrzeuge und Anhänger ISO 11783 ISOBUS CAN für Landmaschinen (Basis J1939)	SAE J2284 (Passenger Cars) 500 kbit/s SAE J1939 (Truck and Bus) 250 kbit/s
TTCAN	Class C(+) On Board	ISO 11898-4 Bosch, 1 Mbit/s	
FlexRay	Class C+ On Board	Herstellerkonsortium 10 Mbit/s	
TTP	Class C+ On Board	Herstellerkonsortium	
MOST	Multimedia	Herstellerkonsortium 25 Mbit/s	

Die weithin bekannten Kfz-Bussysteme wie CAN, LIN oder FlexRay legen in der Regel lediglich die Schichten 0 bis 2 fest, wobei meist sogar nur ein Ausschnitt exakt spezifiziert wird. So lässt etwa die von Bosch veröffentlichte Grundspezifikation CAN 2.0 A/B die mechanischen Aspekte (Stecker, Kabel) völlig außen vor und beschreibt die Schaltung zur Busankopplung und die Signalpegel nur beispielhaft ohne exakte Vorgaben. Die darauf aufsetzende ISO 11898 übernimmt die Bosch-Spezifikation für Schicht 2 weitgehend und liefert für Schicht 1 in nachträglich hinzugekommenen Anhängen Spezifikationsvorschläge nach. Der für die Kfz-Diagnoseschnittstelle grundlegende Standard ISO 9141 dagegen spezifiziert zunächst nur die Schichten 0 und 1 und lässt selbst dabei noch eine Reihe von nur bedingt kompatiblen Varianten zu. Erst spätere Ergänzungen beschreiben Teile der Schicht 2.

Für die höheren Schichten existieren erst seit jüngster Zeit Standards, z.B. wird die Schicht 7 für die abgasrelevanten Teile der Kfz-Diagnose durch den Gesetzgeber in ISO 15031 definiert. Dabei bleibt zunächst offen, ob auf den unteren Schichten die klassische ISO 9141-Schnittstelle oder ein CAN-Bus zum Einsatz kommt. In zusätzlichen Ergänzungen des Standards mit ISO 14230 bzw. ISO 15765 dagegen werden dann Spezifikationen nachgeliefert, die auf K-Line- oder CAN-Spezifika abheben, obwohl es sich eigentlich um eine Schicht 7 Spezifikation handelt.

Da die unteren Schichten immer von den höheren, die höheren Schichten oft, aber praktisch nie vollständig von den unteren Schichten unabhängig sind, wird in den folgenden Kapiteln die Darstellung aufgespalten in die großen Bereiche:

- **Kfz-Busse** (Schicht 0 bis 2), werden in Kapitel 3 beschrieben.
- **Transportprotokolle** (Schicht 3 und 4), werden in Kapitel 4 dargestellt.
- **Anwendungsprotokolle** (Schicht 5 bis 7), werden in Kapitel 5 und 6 erläutert.

2.1 Standardisierung bei Bussystemen

Die Tabellen 2.2 bis 2.5 geben einen Überblick über die im Kfz-Bereich relevanten Standards. Bussysteme mit eingeschränkter Marktbedeutung (z.B. VAN, CCD, D2B), Busse, deren Hersteller erklärt haben, das Bussystem nicht weiterzuentwickeln oder ablösen zu wollen (z.B. ABUS, Byteflight), und Bussysteme, die nur für sehr spezielle Anwendungen, z.B. die Vernetzung von Airbag-Steuergeräten, eingesetzt werden, wurden nicht aufgeführt.

Tabelle 2.3: Transportprotokolle (Schicht 4)

<i>Transportprotokoll</i>	<i>Anwendung</i>	<i>Europ. Standards</i>	<i>US-Standards</i>
ISO TP	CAN	ISO 15765-2	
AUTOSAR TP	FlexRay	AUTOSAR Standard	
SAE J1939	CAN		SAE J1939/21
TP 1.6 TP 2.0	CAN	Hausstandard VW/Audi/Seat/Skoda (Basis OSEK COM 1.0)	

Tabelle 2.4: Anwendungsprotokolle (Schicht 7)

Protokoll	Anwendung	Europ. Standards	US-Standards
ISO 9141-CARB	Diagnose US OBD	<ul style="list-style-type: none"> • ISO 9141-2 Veraltete US-Diagnoseschnittstelle 	SAE J1979, J2190
KWP 2000 Keyword Protocol	Diagnose (allgemein und OBD)	<ul style="list-style-type: none"> • ISO 14230 KWP 2000 Diagnostics on K-Line • ISO 15765 KWP 2000 Diagnostics on CAN 	
UDS Unified Diagnostic Services	Diagnose (allgemein und OBD)	<ul style="list-style-type: none"> • ISO 14229 UDS Unified Diagnostic Services 	
OBD	Diagnose US OBD European OBD	<ul style="list-style-type: none"> • ISO 15031 (identisch mit den US-Standards) 	SAE J1930, J1962, J1978, J1979, J2012, J2186
CCP Can Calibration Protocol XCP Extended Calibration Protocol	Applikation	<ul style="list-style-type: none"> • ASAM AE MCD 1 ASAM-Konsortium Automotive Electronics Measurement, Calibration and Diagnostics 	

Tabelle 2.5: Standards für die Implementierung von Anwendungen mit Bussystemen

Standard	Anwendung	Europ. Standards
OSEK/VDX	Betriebssystem Kommunikation Netzmanagement	ISO 17356-3 OSEK OS ISO 17356-4 OSEK COM ISO 17356-5 OSEK NM
ASAM AE MCD	Applikation	Standardisierte Mess-, Kalibrier- und Diagnosewerkzeuge mit den wichtigen Teilstandards <ul style="list-style-type: none"> • ODX Datenformat für Diagnosedaten • FIBEX Datenformat für die Beschreibung der Buskommunikation
HIS	Flashen Hardwaretreiber	Hersteller Initiative Software HIS Softwaremodule für Steuergeräte
AUTOSAR	Softwarearchitektur	Softwarestruktur zukünftiger Steuergeräte

3 Kfz-Bussysteme – Physical und Data Link Layer

In diesem Kapitel soll ein Überblick über die wichtigsten Kfz-Bussysteme K-Line, CAN, LIN, FlexRay, SAE J1850 und MOST gegeben werden.

3.1 Grundlagen für Kfz-Bussysteme

Zunächst sollen die für die Anwendung im Kfz wichtigen Grundeigenschaften von Bussen dargestellt werden. Dabei wird davon ausgegangen, dass der Leser mit den allgemeinen Grundlagen und Grundbegriffen von Datennetzen vertraut ist.

3.1.1 Elektrotechnische Grundlagen

Die Datenübertragung im Kfz erfolgt in der Regel bitweise seriell. Im einfachsten Fall sind zwei Geräte direkt miteinander verbunden (Bild 3.1.1).

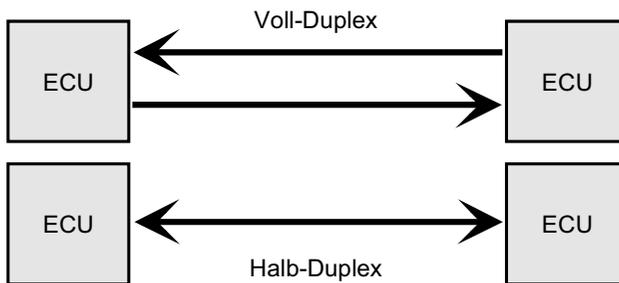


Bild 3.1.1: Punkt-zu-Punkt-Verbindung

Abhängig davon, ob eine gemeinsame bidirektionale Leitungsverbindung oder ein Paar von jeweils unidirektionalen Leitungsverbindungen vorgesehen wird, ist eine Halb- oder Voll-Duplex-Verbindung möglich:

- **Voll-Duplex:** Gleichzeitiges Senden und Empfangen möglich.
- **Halb-Duplex:** Jedes Steuergerät kann nur abwechselnd senden oder empfangen. Standardverfahren bei Kfz-Bussen.

Die Verbindungen können als Ein-Draht- oder als Zwei-Draht-Leitung ausgeführt werden (Bild 3.1.2). Bei der kostengünstigen Ein-Draht-Leitung erfolgt die Signalarückführung über die Fahrzeugkarosserie (Masse). Derartige Verbindungen sind anfällig für die Ein- und Abstrahlung elektromagnetischer Störungen. Daher muss mit relativ großen Signalpegeln, im Kfz häufig mit Batteriespannungspegel, und niedrigen Bitraten gearbeitet werden. Üblich ist dies z.B. bei LIN oder ISO 9141 mit einer bidirektionalen Ein-Draht-Leitung (K-Line) als Halb-Duplex-Verbindung, seltener mit zwei unidirektionalen Ein-Draht-Leitungen als Voll-Duplex-Verbindung.

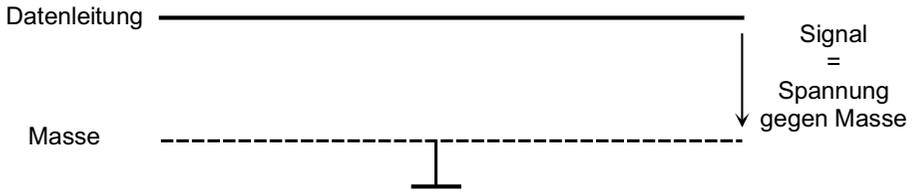
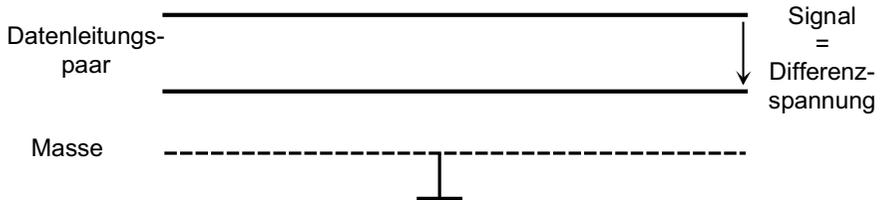
Ein-Draht-Leitung:**Zwei-Draht-Leitung:**

Bild 3.1.2: Ein-Draht- und Zwei-Draht-Leitung

Zwei-Draht-Leitungen, deren Adern oft noch zusätzlich verdreht werden (*Twisted Pair*) sind weniger anfällig für EMV-Probleme. Daher kann mit kleineren Signalpegeln und höheren Bitraten gearbeitet werden. Im Kfz werden für Class B und C Netze, z.B. CAN, üblicherweise Halb-Duplex-Zwei-Draht-Verbindungen eingesetzt.

Abgeschirmte Zwei-Draht-Leitungen oder Paare von Zwei-Draht-Leitungen, also insgesamt vieradrige Verbindungen, wie sie heute bei Voll-Duplex-Ethernet-LANs üblich sind, werden im Kfz in der Regel aus Kostengründen nicht verwendet. Dasselbe gilt auch für optische Netze mit Lichtwellenleitern (LWL), die mit Ausnahme des Multimedia-Bussystems MOST im Kfz bisher nicht eingesetzt werden. Neben den teuren, in jedem Steuergerät notwendigen elektrooptischen Wandlern sind auch die Steckverbindungen für LWL fehleranfällig und die LWL selbst knickempfindlich, so dass die Verlegung zusammen mit den üblichen Kabelbäumen problematisch sein kann.

Bei den Ein-Draht-Verbindungen sind die Signale in der Regel unipolar, bei den Zwei-Draht-Verbindungen bipolar (Bild 3.1.3). Die digitale Übertragung erfolgt meist mit positiver Logik, d.h. die höhere Spannung (*High*) entspricht der *logischen 1*, die kleinere Spannung (*Low*) der *logischen 0*.

Da die EMV-Störstrahlung um so größer ist, je häufiger Signalfanken auftreten, arbeitet man in der Regel mit Non-Return-to-Zero-(NRZ)-Codierung, d.h. das Signal bleibt für die gesamte Dauer eines Bits konstant auf Low oder High. Lediglich das in Europa bedeutungslose SAE J1850 arbeitet mit pulsweitenmodulierten Signalen.

Der Wert
$$f_{\text{bit}} = \frac{1}{T_{\text{bit}}}$$

wird als *Bitrate* bezeichnet. Heute übliche Bitraten liegen im Bereich um 10 kbit/s (Class A-Netz), d.h. *Bitdauer* $T_{\text{bit}} = 100 \mu\text{s}$, bis zu 500 kbit/s (Class C-Netz), d.h.

$T_{\text{bit}} = 2 \mu\text{s}$, bei Multimedia und zukünftigen Class C+-Netzen bis über 10 Mbit/s, d.h.
 $T_{\text{bit}} = 100 \text{ ns}$.

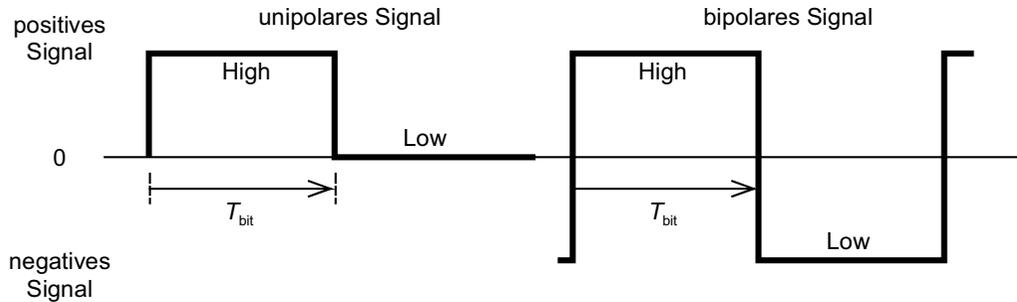


Bild 3.1.3: Unipolare und bipolare Signale

Ebenfalls aus EMV-Gründen versucht man, die Steilheit (*Slew Rate*) der Signalfanken durch entsprechende Treiberschaltungen so langsam zu machen wie für eine gegebene Bitrate gerade noch zulässig.

Die Ausbreitung der Signale auf den Leitungen erfolgt mit Lichtgeschwindigkeit. Aufgrund der dielektrischen Eigenschaften des Kabelmaterials reduziert sich diese gegenüber der Lichtgeschwindigkeit im Vakuum etwa um den Faktor 2 ... 3 auf ca.

$$c = 10 \dots 15 \frac{\text{cm}}{\text{ns}}.$$

Bei einer 20 m langen Verbindungsleitung, wie sie in einer LKW-Zugmaschine mit Auflieger oder einem Omnibus durchaus denkbar ist, beträgt die Signallaufzeit auf der Leitung in einfacher Richtung

$$T_d = \frac{l}{c} = 125 \dots 200 \text{ ns}.$$

Zusätzliche Verzögerungen entstehen in den Leitungstreibern und -empfängern in den Steuergeräten. Für $T_d > 0,1 \cdot T_{\text{bit}}$ treten in der Praxis bereits deutliche Welleneffekte auf (Reflexionen an den Leitungsenden und an Stoßstellen wie Abzweigungen und Steckverbindern), welche die Übertragungssicherheit beeinträchtigen. Für Bussysteme mit höheren Bitraten müssen die Kabelführung, die Steckverbinder und die Kabelausführung sorgfältig gewählt werden. Zusätzlich müssen die Kabel elektrisch *abgeschlossen* werden, d.h. an den beiden Kabelenden sollten an das Kabel angepasste Abschlusswiderstände vorgesehen werden, um die genannten Welleneffekte zu vermeiden. Meist werden diese Abschlusswiderstände in diejenigen Steuergeräte eingebaut, die an den beiden Kabelenden angeschlossen werden. Seltener werden sie in das Kabel oder die Steckverbinder integriert. Steuergeräte, die bei Bussystemen nicht an den Leitungsenden sitzen sondern über Stichleitungen dazwischen angeschlossen sind (Bild 3.1.5), dürfen keine Abschlusswiderstände enthalten.

Im Gegensatz zu einer Punkt-zu-Punkt-Verbindung sind bei einem echten Bus mehrere Steuergeräteaushänge an dieselbe Leitung angeschlossen. Die Signalverknüpfung

erfolgt dabei nach dem Wired-OR-Grundprinzip, wie es im Bild 3.1.4 für eine Ein-Draht-Busleitung dargestellt ist.

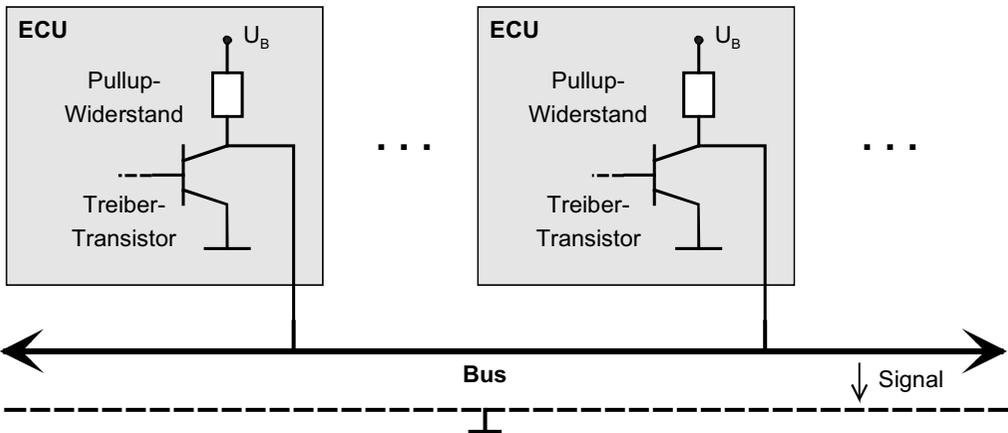


Bild 3.1.4: Wired-OR-Prinzip

Wenn beide Steuergeräte ein *High*-Signal oder gar kein Signal senden wollen, sperren sie den jeweiligen Treibertransistor. Durch die so genannten Pull-Up-Widerstände wird die Busleitung dann auf den Pegel der Versorgungsspannung U_B gezogen. Sobald ein Steuergerät ein *Low*-Signal senden will, schaltet es den jeweiligen Treibertransistor durch. Da der leitende Transistor viel niederohmiger ist als die Pull-Up-Widerstände, wird die Busleitung damit praktisch kurzgeschlossen und das Signal *Low* gesendet. Wenn dagegen zwei Steuergeräte gleichzeitig zu senden versuchen und ein Gerät *Low*, das andere Gerät *High* sendet, „gewinnt“ das Steuergerät, das das *Low*-Signal auf den Bus legt. Man bezeichnet das *Low*-Signal in diesem Fall als dominant, das bei einer solchen Kollision „unterlegene“ *High*-Signal als rezessiv. Um solche Kollisionen zu erkennen, muss jeder Sender die tatsächlich auf dem Bus liegenden Signalpegel überwachen und mit seinem Sendewunsch vergleichen.

3.1.2 Topologie und Kopplung von Bussystemen

Im Gegensatz zu einer Punkt-zu-Punkt-Verbindung können bei einem Datennetz mehrere Steuergeräte (ECU Electronic Control Unit) untereinander Daten austauschen. Für solche Datennetze existieren eine Reihe von unterschiedlichen Aufbauformen (Topologien), von denen die in der Praxis im Kfz am häufigsten zu findende Topologie der so genannte *Bus* ist. Aus diesem Grund wird der Begriff *Bus* häufig, aber nicht ganz präzise, als Synonym für alle Datennetze im Kfz verwendet.

Ein *Bus*, noch präziser ein *Linien-Bus*, entsteht, wenn mehrere Steuergeräte über kurze Stichleitungen an dieselben Verbindungsleitungen angeschlossen werden (Bild 3.1.5). Durch ein Buszugriffsverfahren muss geregelt werden, welches Gerät senden darf, sonst kommt es zu einer Kollision.

Die gesendeten Daten können von allen anderen Steuergeräten empfangen werden. Wenn die gesendeten Daten nicht für alle angeschlossenen Geräte (Broadcast) bestimmt sind, muss bei Beginn der Datenübertragung oder mit jedem Datenpaket eine Adressierungsinformation übertragen werden, die einen einzelnen Empfänger (Unicast) oder mehrere Empfänger (Multicast) auswählt.

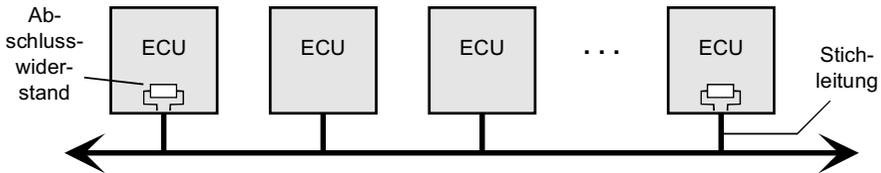


Bild 3.1.5: Linien-Bus

Andere Bus-Topologien wie Stern-, Baum-, Ring- oder Maschen-Netze (Bild 3.1.6) sind im Kfz derzeit noch selten. Ringe sind beim Multimedia-Bus MOST üblich, Flex-Ray wird neben Linien- auch Stern-Strukturen unterstützen.

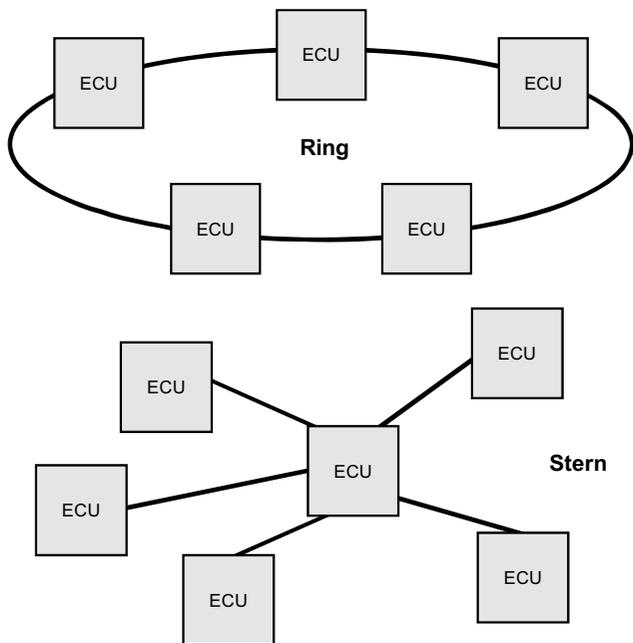


Bild 3.1.6: Im Kfz seltener verwendete Netz-Topologien

Baum-ähnliche Strukturen findet man heute als Zusammenschaltung mehrerer Linien-Bussysteme. So kann man aus Sicht des Diagnosetesters das Fahrzeugnetz in Bild 1.1 als Baum sehen, von dessen zentralem Knoten, dem Gateway, die Bussysteme für den Triebstrang, die Karosserie und die Infotainment-Kommunikation als Zweige

abgehen, wobei der Karosserie-Bus weiter in die Türbusse verzweigt. An den Stellen, an den Steuergerät und Bussystem bzw. mehrere Bussysteme zusammenschaltet werden, muss die Koppelstelle bestimmte Umsetzungsaufgaben erledigen:

Tabelle 3.1.1: Kopplung von Bussystemen

<i>Bus-Koppler</i>	<i>Aufgabe</i>
Transceiver (Bild 3.1.9)	Umsetzung der Bussignale auf die Steuergeräte-internen Spannungspegel für die Sendesignale (Transmitter) und die Empfangssignale (Receiver), andere Bezeichnung <i>Bustreiber</i> oder <i>Leitungstreiber</i> , oft auch physikalisches Businterface (PHY). Findet sich in jedem Steuergerät.
Repeater	Signalverstärker zwischen Leitungsabschnitten ohne <i>Eigenintelligenz</i> . Erlaubt aus elektrotechnischer Sicht längere Leitungen, führt aber zu zusätzlichen Signallaufzeiten. Logisch bilden die beiden über einen Repeater gekoppelten Leitungstücke einen einzigen Bus. Möglich z.B. zur Kopplung von Zugmaschine und Anhänger, wird aber in der Regel als <i>Gateway</i> realisiert.
Gateway (Bild 1.1)	Kopplung mehrerer Netze mit unterschiedlichen Protokollen, z.B. Umsetzung von CAN-Botschaften auf einen LIN-Bus, oder Bussen mit unterschiedlichen Adressbereichen und Bitraten, z.B. zwischen einem Triebstrang-CAN-Bus mit 500 kbit/s und 11 bit-Adressen (Identifier) und einem Karosserie-CAN-Bus mit 125 kbit/s und 29 bit-Adressen.
Switch, Hub, Router	Bei Kfz-Netzen nicht üblich.

3.1.3 Botschaften, Protokollstapel, Dienste (Services)

Bei der Übertragung werden die Nutzdaten mit zusätzlichen Informationen versehen (Bild 3.1.7). Vor den eigentlichen Nutzdaten (*Payload*) wird üblicherweise ein Vorspann (*Header*) übertragen, der Adressinformationen enthält (Sender und Empfänger der Daten), Angaben über die Anzahl der zu übertragenden Daten und gegebenenfalls zu deren Art. An die zu übertragenden Nutzdaten wird häufig ein Nachspann (*Trailer*) angehängt, der Informationen zur Fehlerprüfung und -korrektur enthält.

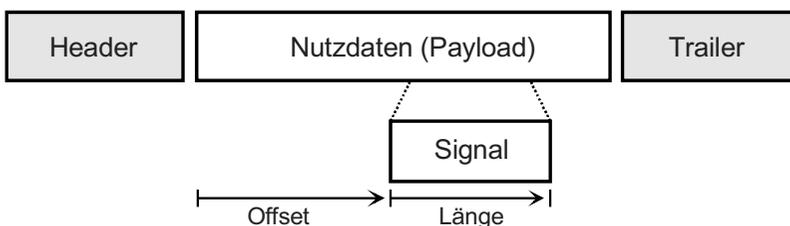


Bild 3.1.7: Aufbau einer Botschaft

Bei einigen Protokollen werden auch zwischen die Nutzdaten nach bestimmten Vorschriften noch weitere Daten zur Übertragungssteuerung oder Trennung von Datenblöcken eingefügt, z.B. *Stuffing-Bits* oder ähnliches.

Im Bereich der Nutzdaten werden häufig mehrere, inhaltlich nicht notwendigerweise direkt zusammengehörige Datenwerte übertragen, z.B. die Motordrehzahl und die Kühlwassertemperatur. Wenn es sich bei diesen Daten um Messwerte handelt, spricht man häufig von *Signalen*. Ein derartiges *Signal* wird durch seine Länge sowie seine Lage bezogen auf den Anfang des Nutzdatenbereichs (Offset in Bit oder Byte) und gegebenenfalls eine Umrechnungsvorschrift definiert, die den Zusammenhang zwischen dem eigentlichen physikalischen Wert, z.B. Drehzahl in 1/min, und dessen hexadezimaler Codierung beschreibt.

Gemäß dem OSI-Schichtenmodell durchlaufen die Daten von der Anwendung bis zur eigentlichen Übertragung über die Busleitungen mehrere Schichten (vergl. Tabelle 2.1), die jeweils eigene Header und Trailer verwenden. Die Botschaft der nächsthöheren Ebene wird beim Senden auf der nächstniedrigeren Ebene als Payload verstanden und um die Header und Trailer der aktuellen Schicht ergänzt. Beim Empfangen dagegen werden Header und Trailer der aktuellen Schicht entfernt, bevor die Payload dagegen werden Header und Trailer der aktuellen Schicht entfernt, bevor die Payload dieser Schicht an die nächsthöhere Ebene weitergegeben wird. Innerhalb einer Schicht werden im Idealfall nur die zugehörigen Header und Trailer verarbeitet, der Inhalt der jeweiligen Payload dagegen ist für die Schicht selbst bedeutungslos. Auf diese Weise entsteht ein *Protokollstapel* (Bild 3.1.8). Der Header einer Schicht wird häufig als *Protocol Control Information PCI* bezeichnet, weil er das für diese Schicht spezifische Protokoll beschreibt, die Nutzdaten als *Service Data Unit SDU* und der gesamte Datenblock als *Protocol Data Unit PDU*.

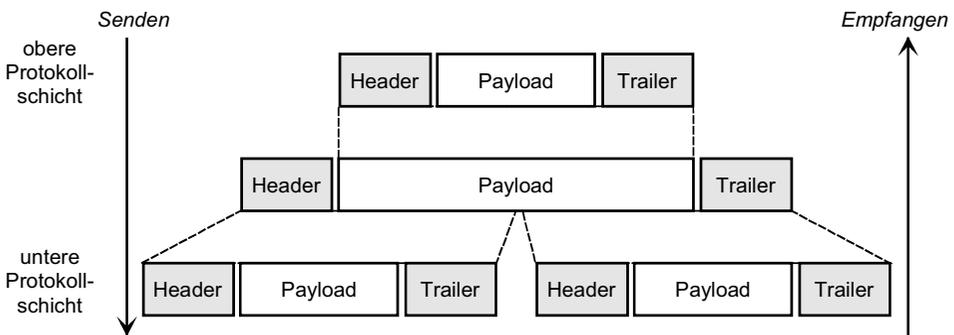


Bild 3.1.8: Protokollstapel (*Protocol Stack*)

Häufig ist die Länge einer Botschaft auf der untersten Schicht begrenzt. So erlaubt z.B. CAN nur maximal 8 Byte, d.h. 64 bit Nutzdaten in einer Botschaft. In diesem Fall wird eine längere Botschaft der oberen Schicht beim Senden in mehrere Botschaften aufgeteilt (*Segmentierung*) und beim Empfangen wieder aus mehreren Botschaften der unteren Schicht zusammengesetzt (*Desegmentierung*). Der umgekehrte Fall, dass eine untere Ebene längere Botschaften erlaubt als eine höhere Ebene und daher mehrere Botschaften beim Senden zu einer Botschaft zusammengefasst werden können, wie dies bei Büro-LANs geschieht, ist bei Kfz-Bussen derzeit noch nicht üblich.

Bei sauber spezifizierten Protokollen besitzt jede Schicht des Protokollstapels eine exakt definierte Reihe von Funktionen, so genannte *Dienste (Services)*. Die übergeordnete Protokollschicht verwendet diese Dienste, um Botschaften an die darunter liegende Schicht zu senden, von dieser zu empfangen oder Konfigurations- und Statusinformationen mit dieser Schicht auszutauschen.

Der Begriff *Botschaft* (engl.: *Message*) ist dabei nicht eindeutig. Je nach Ebene in einem Protokoll und in unterschiedlichen Standards mit unterschiedlicher Bedeutung gebraucht, finden sich auch die Begriffe *Datenblock* und *Rahmen (Frame)*.

Für die Adressierung von Botschaften gibt es zwei gängige Verfahren:

- **Gerätebasierte Adressierung** (Stations- oder Teilnehmeradressierung)
- **Inhaltsbasierte Adressierung** (Botschafts-Kennung).

Bei der *gerätebasierten Adressierung* (Stations- oder Teilnehmeradressierung) wird jedes Steuergerät durch eine eindeutige Adresse (Nummer) identifiziert, wobei jede Botschaft neben der *Zieladresse*, d.h. der Adresse des Empfängers, in der Regel auch die *Quelladresse*, d.h. die Adresse des Senders enthält. Für die Aussendung von Botschaften an mehrere (Multicast) oder alle Empfänger (Broadcast) existieren spezielle Adressen. Da die Anzahl der Steuergeräte in einem Fahrzeug relativ klein ist, genügen für solche Adressen einige Bit, typ. 5...8. Werden mehrere Busse über ein Gateway verknüpft, wird in den höheren Protokollschichten meist eine für das Gesamtsystem eindeutige Geräteadresse verwendet, die vom Gateway in einer unteren Protokollschicht in eine nur für das adressierte Bussystem eindeutige Geräteadresse umgesetzt wird.

Anstelle der gerätebasierten Adressierung kann auch eine *inhaltsbasierte Adressierung* verwendet werden, wie z.B. bei CAN. Dabei wird eine Botschaft mit einer bestimmten, adressähnlichen Kennung (Identifizier) versehen, die den Inhalt der Botschaft kennzeichnet, also anzeigt, dass eine Botschaft beispielsweise Informationen über die Motordrehzahl enthält. Da in heutigen Fahrzeugen sehr viele solcher Informationen über die Bussysteme gesendet werden, benötigt man hierfür längere *Adressen*, bei CAN z.B. sind alternativ 11 bit oder 29 bit Kennungen üblich. Die inhaltsbasierte Adressierung führt zu einer geringeren Busbelastung und zu einer besseren Datenkonsistenz im gesamten Netzwerk, wenn Daten häufig an mehrere Teilnehmer versendet werden müssen.

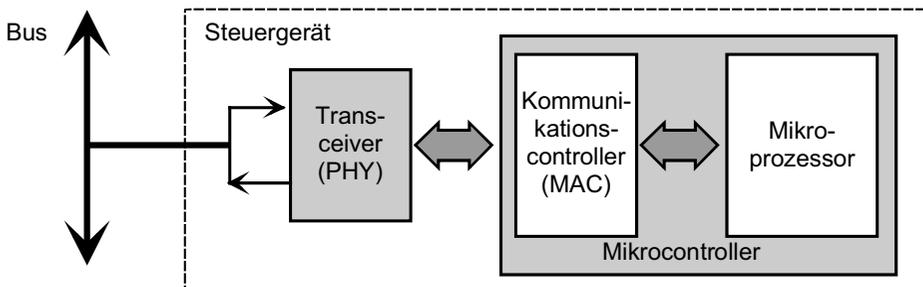


Bild 3.1.9: Typisches Bus-Interface eines Steuergerätes
(PHY Physical Interface, MAC Media Access Control)