Anwendung von Bussystemen und Protokollen

1

Bereits bei Einführung der ersten mikroprozessorgesteuerten Systeme wie Motronic und ABS im Kraftfahrzeug ab 1980 wurden Daten zwischen diesen Systemen und nach außen ausgetauscht. Während zwischen den Systemen (On-Board) zunächst vor allem Punkt-zu-Punkt-Verbindungen mit Analogsignalen oder einfachen Schaltsignalen verwendet wurden, diente die erste echte Datenkommunikation zum Anschluss des Diagnosetesters in der Kfz-Werkstatt (Off-Board). Sehr schnell war eine herstellerübergreifende Lösung nötig, wobei Bosch als in Europa führender Hersteller elektronischer Steuergeräte eine von vielen Fahrzeugherstellern übernommene Spezifikation vorlegte, die später als ISO 9141 standardisiert wurde. Diese Spezifikation legte zunächst wenig mehr als die Zahl der Verbindungsleitungen, die elektrischen Signalpegel und das Bitformat der Zeichenübertragung fest. Die Bedeutung der übertragenen Daten sowie die in der Werkstatt angewendeten Diagnoseverfahren selbst blieben offen und wurden weiterhin Hersteller-spezifisch implementiert.

Mit der Einführung des von Bosch vorgestellten, später 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 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 (Abb. 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 und die zum Datenaustausch verwendeten Protokolle zu suchen [1–6].

W. Zimmermann und R. Schmidgall, *Bussysteme in der Fahrzeugtechnik*, ATZ/MTZ-Fachbuch,
DOI 10.1007/978-3-658-02419-2_1, © Springer Fachmedien Wiesbaden 2014

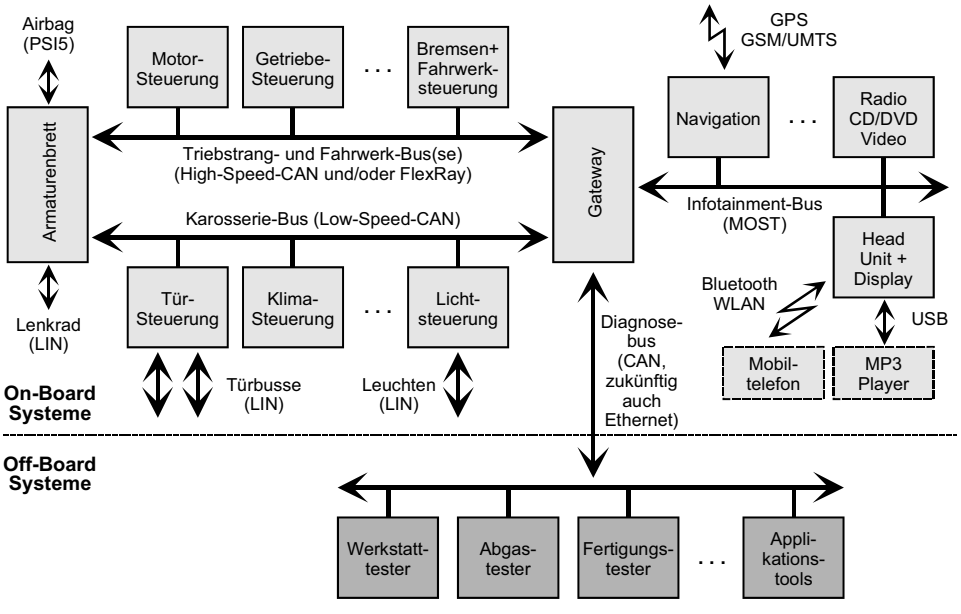

Abb. 1.1 Bussysteme eines modernen Fahrzeugs

1.1 Überblick

Grob lassen sich dabei folgende Anwendungsgebiete unterscheiden (Tab. 1.1):

- **On-Board-Kommunikation** zwischen den Kfz-Steuergeräten im Fahrzeug
  Dieses Aufgabenfeld lässt sich heute in 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 nur wenige Bytes lang, müssen aber periodisch mit hoher Frequenz, kurzer Verzögerungszeit (Latenz) und großer Zuverlässigkeit übertragen werden. Für diese Anwendungen wurde CAN entworfen. Durch moderne Fahrwerkssteuerungen steigen die Anforderungen. Daher wurde CAN zu CAN FD weiterentwickelt und neue Bussysteme wie FlexRay konzipiert.
  - **Low-Speed-Systeme zur Kabelbaum-Vereinfachung**
    Auch für simplere Aufgaben wie das Ansteuern von Lampen und Fenstermotoren werden Bussysteme eingeführt, um den Kabelbaum zu vereinfachen. Da vor allem Schaltsignale übertragen werden, sind die erforderlichen Datenraten niedrig. Hauptgesichtspunkt sind die Kosten je Busknoten. Für diese Aufgaben wird eine verein-

Tab. 1.1 Anwendungsbereiche und Anforderungen an Bussysteme im Kfz

Anwendung	Botschafts-länge	Botschafts-rate	Resultierende 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
Infotainment	Lang	Hoch	Sehr hoch	Mäßig	Mäßig	Hoch
Assistenz	Mischung von Steuerungs- und Infotainment-Anforderungen					
Off-Board-Kommunikation						
Werkstattdiagnose	Kurz	Niedrig	Niedrig	Unwichtig	Gering	Niedrig
Fertigungstest beim Hersteller inkl. End-of-Line-Programming (*Flashen*)	Lang bis sehr lang	Niedrig	Hoch	Unwichtig	Mäßig	Unwichtig
Applikation am Fahrzeug und an Prüfständen in der Entwicklung (*Umprogrammieren*, *Kalibrieren*)	Kurz	Mittel bis hoch bei Messaufgaben	Mittel bis hoch	Kurz	Gering	Unwichtig
Car to X	Mittel	Mittel	Mittel	Hoch	Mittel	Hoch

fachte CAN-Variante ergänzt durch das noch kostengünstigere Bussystem LIN eingesetzt. Daneben entstanden spezielle Busse wie PSI5 oder Safe-by-Wire für die Vernetzung von Rückhaltesystemen.

  - **Infotainment-Systeme**
    Mit der Einführung von Infotainment-Systemen im Fahrzeug für Navigation, Video, Audio und Telefon müssen sehr große Datenmengen beispielsweise zwischen dem DVD-Wechsler, dem verteilten Audio-System und dem Display im Armaturenbrett ausgetauscht werden. Dabei ist der Hauptgesichtspunkt ein hoher Datendurchsatz, während die Anforderungen an die Latenzzeiten und die Übertragungszuverlässigkeit wesentlich geringer sind als bei den High-Speed-Systemen für Echtzeit-Steuerungsaufgaben. Die Anbindung an die Außenwelt und die Einbindung von Geräten der Unterhaltungselektronik erfolgt über Funkschnittstellen wie GSM/UMTS/LTE und GPS oder Consumer-Schnittstellen wie USB bzw. Bluetooth.
  - **Fahrerassistenz-Systeme**
    Diese Systeme verbinden Merkmale aus dem Infotainment-Bereich wie Videoübertragung mit Merkmalen von Echtzeit-Steuerungsaufgaben.

- **Off-Board-Kommunikation** zwischen Fahrzeug und externen Geräten:
  - **Diagnose-Kommunikation in der Werkstatt und beim 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 höhere Datenraten sowie 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, da-

Tab. 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, Diagnose
D	> 1 Mbit/s	FlexRay, TTP, Ethernet	X by Wire, Backbone-Netz
Infotainment	> 10 Mbit/s	MOST, Ethernet	Multimedia (Audio, Video)

mit 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 wurden Standard-Schnittstellen wie ASAM MCD für den Messdatenaustausch, die Parameterverstellung und die Datenhaltung geschaffen.
- **Car to Car und Car to Infrastructure (Car to X)**
  Schon heute kommunizieren Fahrzeuge etwa in Mautsystemen automatisch mit Verkehrszähl- und Leiteinrichtungen (*Car to Infrastructure*). Diese Systeme werden erweitert, um den Verkehrsfluss besser steuern und durch Kommunikation zwischen den Autos (*Car to Car*) Unfälle vermeiden zu helfen.

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

1.2 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

Tab. 1.3 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	**Präsentation**	**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

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, LIN oder FlexRay 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 (Tab. 1.3) dienen, das die Kommunikationshierarchie beschreibt. Die grau markierten Schichten haben für Kfz-Anwendungen (noch) keine Bedeutung.

Vor dem Spiegel realer Standards betrachtet ist das OSI-Modell allerdings eher 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.

Die weithin bekannten Kfz-Bussysteme wie CAN, LIN oder FlexRay (Tab. 1.4) 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 Kap. 2 und 3 beschrieben.
- **Transportprotokolle** (Schicht 3 und 4), werden in Kap. 4 dargestellt.
- **Anwendungsprotokolle** (Schicht 5 bis 7), werden in Kap. 5 und 6 erläutert.
- **Softwarestandards** und **Anwendungen** werden in Kap. 7 bis 10 vorgestellt.

1.3 Standardisierung bei Kfz-Bussystemen und Software

Die Tab. 1.5, 1.6 und 1.7 geben einen Überblick über die im Kfz-Bereich relevanten Standards. Bussysteme mit eingeschränkter Marktbedeutung (z. B. VAN, CCD, D2B, TTP), 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.

Tab. 1.4 Kfz-Busse (Schicht 0 bis 2), Web-Links zu den Herstellern im Anhang

Bustyp	Anwendung	Europ. Standards	US-Standards
Zeichen-basiert (UART)			
K-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**
PWM-basiert			
SAE J1850	Diagnose, Class A/B On-Board		**SAE J1850** (PWM Ford, VPWM GM, Chrysler) 10,4 und 41,6 kbit/s
Bitstrom-basiert			
CAN CAN FD TTCAN	Class B/C On-Board, Diagnose	**ISO 11898** 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 500 kbit/s
FlexRay	Class D On-Board	Herstellerkonsortium, **ISO 17458**, 10 Mbit/s	
TTP	Class D On-Board	Herstellerkonsortium	
MOST	Infotainment	Herstellerkonsortium 25, 50, 150 Mbit/s	
Ethernet	Diagnose Flash Programmieren	**IEEE 802.3**, **ISO 13400** 10/100 Mbit/s	
Diverse			
PSI5, SENT ASRB, DSI	Class A/B On-Board Sensor-Aktor-Bus	Herstellerkonsortien	

Tab. 1.5 Transportprotokolle (Schicht 4)

Transportprotokoll	Anwendung	Europ. Standards	US-Standards
ISO TP	CAN	ISO 15765-2	
FlexRay TP	FlexRay	ISO 10681-2	
SAE J1939	CAN		SAE J1939/21
TP 1.6, TP 2.0	CAN	Hausstandard VW/Audi/Seat/Skoda	
DoIP	Ethernet	ISO 13400-2	

Tab. 1.6 Anwendungsprotokolle (Schicht 7)

Protokoll	Anwendung	Europ. Standards	US-Standards
ISO 9141-CARB	Diagnose US OBD	**ISO 9141-2** Veraltete US-Diagnoseschnittstelle	SAE J1979, J2190
KWP 2000 Keyword Protocol	Diagnose (allgemein und OBD)	**ISO 14230** KWP 2000 Diagnostics on K-Line	
UDS Unified Diagnostic Services	Diagnose (allgemein und OBD)	**ISO 14229** UDS Unified Diagnostic Services **ISO 15765** UDS on CAN	
OBD	Diagnose US OBD European OBD	**ISO 15031** (identisch mit den US-Standards)	SAE J1930, J1962, J1978, J1979, J2012, J2186
CCP Can Calibration Protocol **XCP** Extended Calibration Protocol	Applikation	**ASAM AE MCD** ASAM-Konsortium Automotive Electronics Measurement, Calibration and Diagnostics	

1.4 Neuere Entwicklungen

Nachdem der Beginn des Jahrhunderts zunächst mit LIN, FlexRay und MOST eine Vielfalt neuer Bussysteme im Fahrzeug gebracht hat, setzte sich allmählich die Erkenntnis durch, dass die heterogene Vielfalt letztlich aufwendig und nur schwer zu beherrschen ist. Die Wirtschaftskrise im Jahr 2009 hat diese Überlegungen sicher noch beschleunigt. Für diese Vielfalt gab es wohl drei Hauptursachen, die überwunden werden müssen, bevor eine echte Konsolidierung möglich ist:

- Im Automobilbereich gibt es ein ausgeprägtes Kostenbewusstsein, das sich allerdings stark auf die Stückkosten konzentriert. Der Engineeringaufwand dagegen wird im Hinblick auf die hohen Stückzahlen oft wenig beachtet, es sei denn er verursacht zu lange

Tab. 1.7 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 ODX/OTX für Diagnosedaten und -tests FIBEX Datenformat für die Beschreibung der Buskommunikation
HIS	Flashen Hardwaretreiber	Hersteller Initiative Software HIS Softwaremodule für Steuergeräte
AUTOSAR	Softwarearchitektur	Softwarestruktur zukünftiger Steuergeräte inklusive AUTOSAR OS, COM und NM

Entwicklungsdauern (*Time to Market*). LIN ist ein klassisches Beispiel für eine Lösung, die technisch nicht mehr kann als der schon lange vorher etablierte CAN, aber geringfügig niedrigere Stückkosten hat. Welcher Aufwand in die Grundsatzentwicklung, Einführung und Pflege von LIN, den Upgrade existierender CAN-Werkzeuge für LIN, die Schulung von Mitarbeitern und die Entwicklung unzähliger CAN/LIN-Gateways geflossen ist, ist schwer zu beziffern und wird von den Protagonisten neuer Lösungen oft sogar bewusst klein gerechnet. Die Kosten von Feldausfällen, die direkt oder indirekt aus Spezifikations- oder Testlücken beim Zusammenspiel von LIN und CAN entstanden, sind wohl selbst den Verantwortlichen nicht wirklich bekannt. Ehrlicherweise muss man diese verborgenen Kosten einer neuen, oft komplexeren Lösung der möglichen Stückkostenreduktion im Vergleich zu einer vorhandenen, einfacheren Lösung gegenüberstellen.

- Viele Lösungen spiegeln die Organisationsstrukturen der Automobilhersteller und ihrer Zulieferer wieder. Historisch wurden Motor, Getriebe, Fahrwerk und Karosserie von verschiedenen Abteilungen, oft sogar Entwicklungs- oder Geschäftsbereichen verantwortet. Auch wenn die Beteiligten an den Schnittstellen ihrer Komponenten zur Zusammenarbeit gezwungen waren, wird doch oft versucht, eine für den eigenen Zuständigkeitsbereich optimale Lösung zu finden. Nicht überzeugend auch, wenn neue Konzepte in Forschung und Vorentwicklung ohne Rücksicht auf Erfahrungen aus der Serien- und Fertigungspraxis entwickelt werden. Auf diese Weise entstehen typische domänenspezifische Lösungen. MOST orientiert sich fast ausschließlich an den Anforderungen des Infotainment. FlexRay merkt man an, dass die Entwickler den Fahrwerksbereich mit seinen zeitkritischen verteilten Regelsystemen und hohen Sicherheitsanforderungen im Blick hatten, während die Flexibilität, mit der Karosserieelektroniker noch kurz vor Serienanlauf einige weitere Signale in eine vorhandene oder neue Busbotschaft packen wollen, beim Entwurf sicher nicht im Vordergrund stand. Ärger bereiten solche Insellösungen aber erst dann, wenn Steuergeräte im Fahrzeug dort eingebaut werden müssen,

wo noch ein freier Bauraum zu finden ist. Und wenn ausgerechnet dort das für die Aufgabenstellung eigentlich zuständige Bussystem nicht zur Verfügung steht. Dann wird die Ansteuerung des Motorlüfters schon einmal von der Lichtsteuerung übernommen und das Infotainment-Steuergerät bekommt noch eine CAN- oder LIN-Schnittstelle, weil die Radiobedientasten im Lenkrad nun einmal kein MOST-Interface besitzen. Richtig hinderlich werden diese Domänengrenzen, wenn Fahrerassistenzsysteme integriert werden sollen, die nur durch das Zusammenwirken von Funktionsgruppen mehrerer Domänen realisierbar sind.

- Haupttreiber der Komplexität ist nicht zuletzt das Beharrungsvermögen der Automobilindustrie, das durch die breiteren Modellpaletten und paradoxerweise gerade durch kürzere Entwicklungszyklen erzwungen wird. Beides ist nur möglich, wenn viele Teilsysteme über mehrere Modellbaureihen und Fahrzeuggenerationen wiederverwendet werden. Dies erzwingt Aufwärtskompatibilität und verhindert, dass Neuentwicklungen ältere Lösungen vollständig ablösen. Hatten die ersten Motorsteuergeräte zusätzlich zur ihren proprietären PWM-Schnittstellen ein K-Line-Diagnoseinterface, so kamen später ein, dann mehrere CAN-Busse dazu, ohne dass die K-Line wirklich verschwand. Mittlerweile haben Motorsteuergeräte auch ein FlexRay- und ein LIN-Interface und vielleicht demnächst auch noch eine Ethernet-Schnittstelle. Aber ob die PWM-Schnittstellen wirklich verschwunden sind, ist so sicher nicht. Das alte CAN-Interface, das die Einführung von FlexRay überlebt hat, kann sich freuen, zu CAN FD erweitert zu werden, und muss nicht befürchten, durch Ethernet vollständig ersetzt zu werden. Mit jedem neuen Konzept nimmt die Komplexität zu statt ab. Und so, wie bei CAN, bei LIN und schließlich bei FlexRay wird man sich bei der Einführung von Ethernet vorrangig auf das Übertragungsmedium konzentrieren, statt endlich auch im PKW-Bereich die Anwendungsschnittstelle, d. h. die zu übertragenden Echtzeitdaten, zu standardisieren, wie man das im Nutzfahrzeugbereich mit SAE J1939/71 schon vor 20 Jahren gemacht hat.

Natürlich kennt die Automobilindustrie diese Problembereiche und geht dagegen an:

- Elektrik/Elektronik-(E/E)-Architekturen werden gezielter konzipiert und analysiert.
- AUTOSAR ist ein richtiger, wenn auch selbst sehr komplexer Schritt, die Komplexität durch hierarchische Strukturen, Kapselung und eine stärkere Automatisierung der Softwareerstellung beherrschbar zu machen.
- CAN FD ist der Versuch, durch evolutionäre Änderungen den eher revolutionären Umstieg auf FlexRay in vielen Systemen unnötig zu machen.
- Ethernet könnte langfristig vielleicht wirklich zu einer Bereinigung der Bussystem-Landschaft im Automobil führen.

Bis dahin ist allerdings ein langer Weg, so dass der Entwickler und damit dieses Buch von Ausgabe zu Ausgabe immer mehr statt weniger Themengebiete abdecken muss.

Literatur

[1] Robert Bosch GmbH, K. Reif, K.-H. Dietsche (Hrsg.): Kraftfahrtechnisches Taschenbuch. Springer Vieweg Verlag, 27. Auflage, 2011

[2] R. K. Jurgen (Hrsg.): Automotive Electronics Handbook. McGraw Hill Verlag, 2. Auflage, 1999

[3] G. Conzelmann, U. Kiencke: Mikroelektronik im Kraftfahrzeug. Springer Verlag, 1. Auflage, 1995

[4] J. Schäuffele, T. Zurawka: Automotive Software Engineering. Springer-Vieweg Verlag, 5. Auflage, 2013

[5] K. Reif: Automobilelektronik. Springer-Vieweg Verlag, 4. Auflage, 2012

[6] H. Wallentowitz, K. Reif: Handbuch Kraftfahrzeugelektronik. Springer-Vieweg Verlag, 2. Auflage, 2011

Grundkonzepte und einfache Kfz-Bussysteme

2

In diesem Kapitel werden die elektrotechnischen Grundlagen dargestellt und einfache Kfz-Kommunikationsschnittstellen wie K-Line, SAE J1850 sowie einige Sensor-Aktor-Busse betrachtet.

2.1 Grundlagen

Zunächst werden die für die Anwendung im Kfz wichtigen Grundeigenschaften von Bussystemen vorgestellt. Dabei wird davon ausgegangen, dass der Leser mit den allgemeinen Grundlagen und Grundbegriffen von Datennetzen vertraut ist [1–3].

2.1.1 Elektrotechnische Grundlagen

Die Datenübertragung im Kfz erfolgt in der Regel bitweise seriell. Im einfachsten Fall sind zwei Geräte direkt miteinander verbunden (Abb. 2.1).

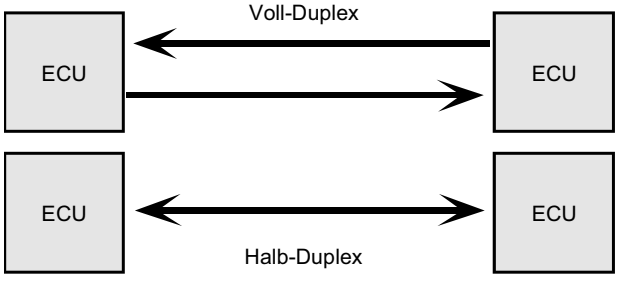

Abb. 2.1 Punkt-zu-Punkt-Verbindung

W. Zimmermann und R. Schmidgall, *Bussysteme in der Fahrzeugtechnik*, ATZ/MTZ-Fachbuch, DOI 10.1007/978-3-658-02419-2_2, © Springer Fachmedien Wiesbaden 2014

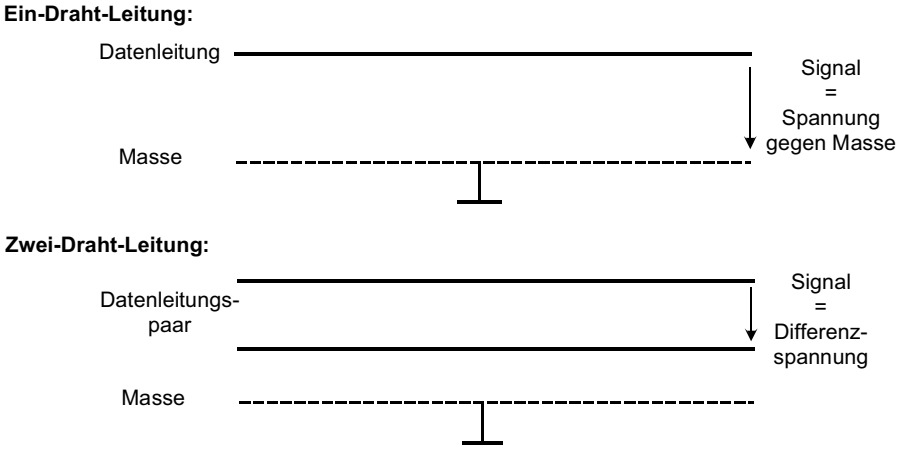

Abb. 2.2 Ein-Draht- und Zwei-Draht-Leitung

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 ist möglich.
Halb-Duplex: Jedes Steuergerät kann nur abwechselnd senden oder empfangen. Standardverfahren bei Kfz-Bussystemen.

Die elektrischen Verbindungen können als Ein-Draht- oder als Zwei-Draht-Leitung ausgeführt werden (Abb. 2.2). Bei der kostengünstigen Ein-Draht-Leitung erfolgt die Signalrückführung 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 bei ISO 9141) als Halb-Duplex-Verbindung, seltener mit zwei unidirektionalen Ein-Draht-Leitungen als Voll-Duplex-Verbindung.

Zwei-Draht-Leitungen, deren Adern oft noch zusätzlich verdrillt werden (*Twisted Pair*) sind weniger anfällig für EMV-Probleme (Elektromagnetische Verträglichkeit). Dabei wird die Spannung zwischen den beiden Leitungen zur Signalübertragung verwendet (Differenzsignal). Daher kann mit kleineren Signalpegeln und höheren Bitraten gearbeitet werden. Im Kfz werden für Class B und C Netze, z. B. CAN oder FlexRay, 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 Infotainment-Bussystems MOST im Kfz bisher nicht eingesetzt wer-

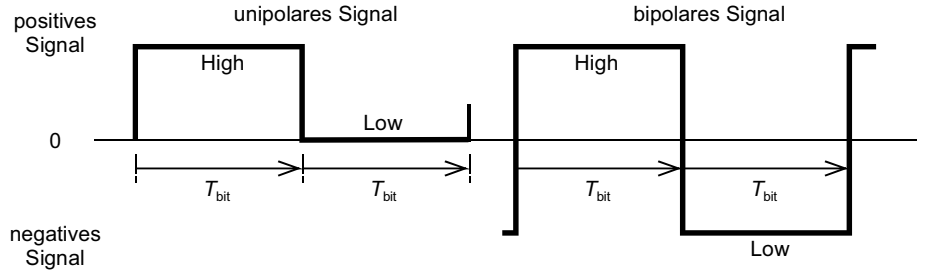

Abb. 2.3 Unipolare und bipolare Signale

den. 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*, das Differenzsignal bei den Zwei-Draht-Verbindungen, z. B. CAN, meist *bipolar* (Abb. 2.3). Die digitale Übertragung erfolgt üblicherweise mit positiver Logik, d. h. die höhere Spannung (*High*) entspricht der *logischen 1*, die kleinere Spannung (*Low*) der *logischen 0* (*binäre* Signale). Verwendet man wie bei FlexRay einen dritten, zwischen *High* und *Low* liegenden Spannungspegel um z. B. den Ruhezustand des Bussystems abzubilden, spricht man von *ternären* Signalen.

Da die EMV-Störstrahlung umso größer ist, je häufiger Signalflanken auftreten, arbeitet man in der Regel mit Non-Return-to-Zero-(NRZ)-Codierung, d. h. das Signal bleibt für die gesamte Dauer eines Bits T_{bit} konstant auf Low oder High. Lediglich das veraltete SAE J1850 arbeitet mit pulsweitenmodulierten Signalen.

Der Wert

$$f_{bit} = \frac{1}{T_{bit}}$$

wird als *Bitrate* bezeichnet. Heute übliche Bitraten liegen im Bereich von 10 kbit/s (Class A-Netz, z. B. K-Line), d. h. *Bitdauer* $T_{bit} = 100$ µs, bis zu 500 kbit/s (Class C-Netz, z. B. CAN), d. h. $T_{bit} = 2$ µs. Bei Class D-Netzen (z. B. FlexRay) werden heute bis zu 10 Mbit/s, d. h. $T_{bit} = 100$ ns, verwendet und im Infotainmentbereich (z. B. MOST) sind schon heute 25 Mbit/s, d. h. $T_{bit} = 40$ ns, im Einsatz, zukünftig 150 Mbit/s.

Ebenfalls aus EMV-Gründen versucht man, die Steilheit (*Slew Rate*) der Signalflanken 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.

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

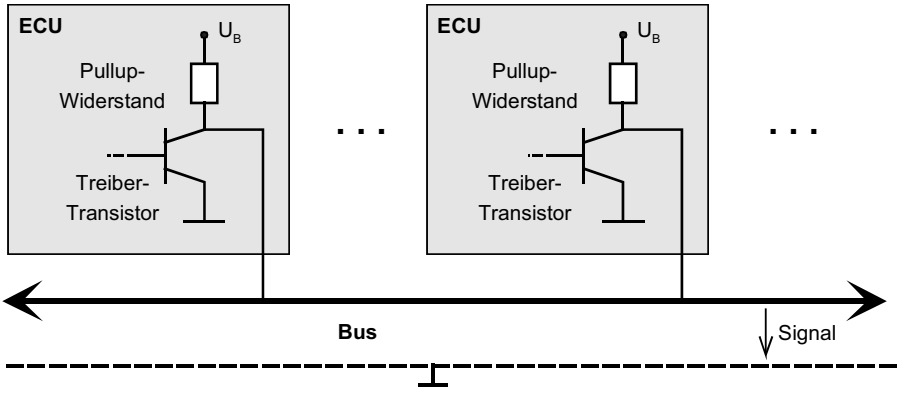

Abb. 2.4 Linien-Bus

Bei einer $l = 20$ 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

$$T_\mathrm{d} = \frac{I}{c} = 125 \ldots 200\mathrm{ns} \; .$$

Zusätzliche Verzögerungen entstehen in den Leitungstreibern und -empfängern in den Steuergeräten. Für $T_\mathrm{d} > 0{,}1 \cdot T_\mathrm{bit}$ 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, oder sie werden in das Kabel bzw. die Steckverbinder integriert. Steuergeräte, die bei Bussystemen nicht an den Leitungsenden sitzen sondern über Stichleitungen dazwischen angeschlossen sind (Abb. 2.4), dürfen keine Abschlusswiderstände enthalten.

Im Gegensatz zu einer Punkt-zu-Punkt-Verbindung sind bei einem echten Bus mehrere Steuergeräteausgänge an dieselbe Leitung angeschlossen. Die Signalverknüpfung erfolgt dabei nach dem Wired-OR-Grundprinzip, wie es im Abb. 2.5 für eine Ein-Draht-Busleitung dargestellt ist.

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 Steuer-

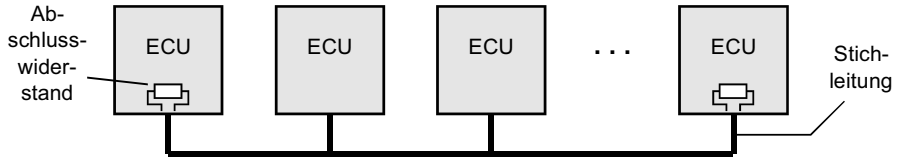

Abb. 2.5 Wired-OR-Prinzip

geräte gleichzeitig zu senden versuchen und ein Gerät *Low*, das andere Gerät *High* sendet, „gewinnt" dasjenige 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.

2.1.2 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 existiert 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 (Abb. 2.4). Durch ein Buszugriffsverfahren muss geregelt werden, welches Gerät zu welchem Zeitpunkt senden darf, um Kollisionen zu verhindern bzw. wie Kollisionen erkannt und aufgelöst werden.

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.

FlexRay unterstützt zwar auch Linien-Busse, bevorzugt aber Stern-Strukturen (Abb. 2.6). Ringe sind beim Infotainment-Bus MOST üblich.

Baum-ähnliche Netze findet man heute als Zusammenschaltung mehrerer Linien-Bussysteme. So kann man aus Sicht des Diagnosetesters das Fahrzeugnetz in Abb. 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 denjenigen Stellen, an denen Steuergerät und Bus bzw. mehrere Bussysteme zusammengeschaltet werden, muss die Koppelstelle bestimmte Umsetzungsaufgaben erledigen (Tab. 2.1).

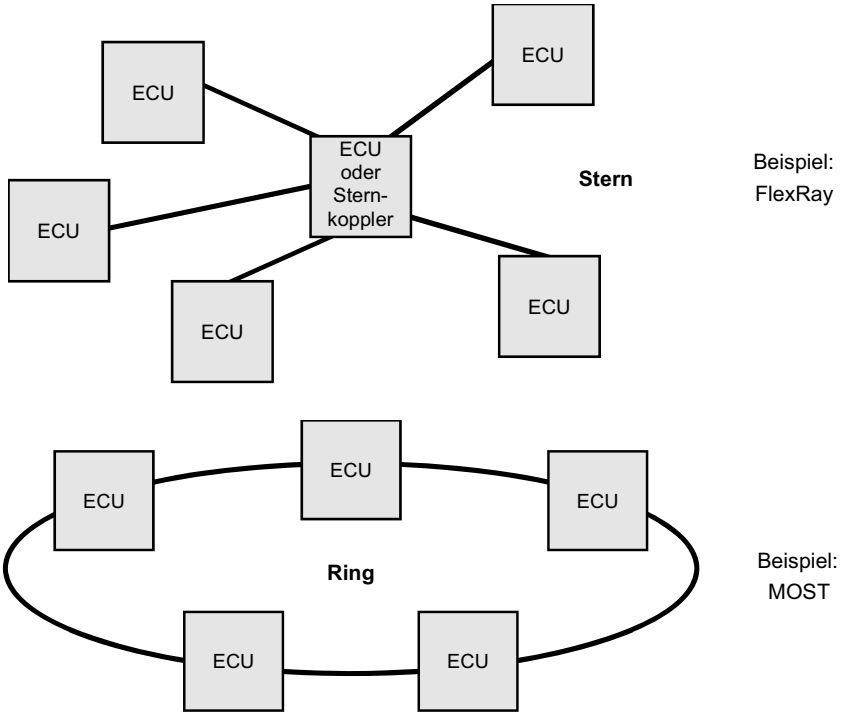

Abb. 2.6 Weitere Netz-Topologien

2.1.3 Botschaften, Protokollstapel, Dienste (Services)

Bei der Übertragung werden die Nutzdaten mit zusätzlichen Informationen versehen (Abb. 2.7). 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.

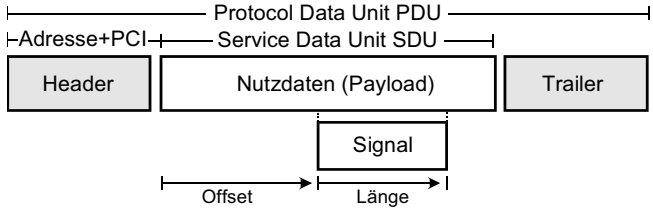

Abb. 2.7 Aufbau einer Botschaft

Tab. 2.1 Kopplung von Bussystemen

Bus-Koppler	Aufgabe
Transceiver (Abb. 2.11)	Umsetzung der Bussignale auf die Steuergeräte-internen Spannungspegel für die Sendesignale (Transmitter) und die Empfangssignale (Receiver), andere Bezeichnung *Bustreiber* oder *Leitungstreiber*, oft auch *physikalisches Businterface* (PHY). Findet sich in jedem Steuergerät.
Repeater	Signalverstärker zwischen Leitungsabschnitten ohne *Eigenintelligenz*. Erlaubt aus elektrotechnischer Sicht längere Leitungen, führt aber zu zusätzlichen Signallaufzeiten. Logisch bilden die beiden über einen Repeater gekoppelten Leitungsstücke einen einzigen Bus. Üblich im aktiven Sternpunkt von FlexRay; möglich bei CAN zur Kopplung von Zugmaschine und Anhänger, wird aber in der Regel als *Gateway* realisiert.
Gateway (Abb. 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. Router-Funktion oft im Gateway.

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 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 (vgl. Tab. 1.3), 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 (Abb. 2.8). Beim Empfangen 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*. 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*.

In der Regel ist die Länge einer Botschaft auf dem Bussystem begrenzt. So erlaubt z. B. CAN nur maximal 8 Byte, d. h. 64 bit Nutzdaten in einer Botschaft. In diesem Fall wird eine

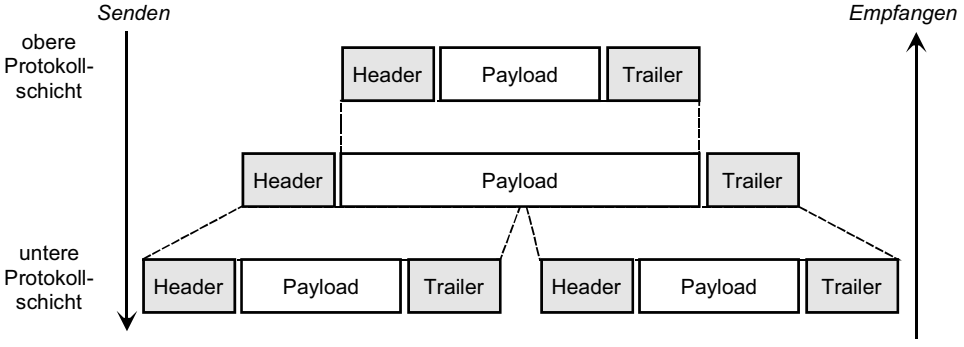

Abb. 2.8 Protokollstapel (*Protocol Stack*)

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, ist bei Kfz-Bussen derzeit noch selten, kommt aber z. B. bei FlexRay in Betracht.

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, Paket* und *Rahmen* (*Frame*).

2.1.4 Kommunikationsmodelle, Adressierung

Der Datenaustausch zwischen zwei Steuergeräten über einen Protokollstapel vermittelt den Anwendungen im Idealfall die Illusion, ohne zwischengeschaltetes Bussystem direkt miteinander zu kommunizieren (Abb. 2.9). Der Protokollstapel kapselt die Details der Kommunikation, wobei auch innerhalb des Protokollstapels jede Ebene den Eindruck haben sollte, direkt mit der entsprechenden Ebene des anderen Kommunikationspartners zu interagieren.

In der Regel wird die Kommunikation zwischen den Steuergeräten im Fahrzeug während der Entwicklungsphase festgelegt, da die verbauten Geräte und die Struktur der auszutauschenden Informationen vorab bekannt sind und damit keine Laufzeitkonfiguration mehr nötig ist. Bei manchen Protokollen ist trotzdem am Beginn der Kommunikation eine gegenseitige Verständigung vorgesehen, ähnlich der Wahl der Telefonnummer und

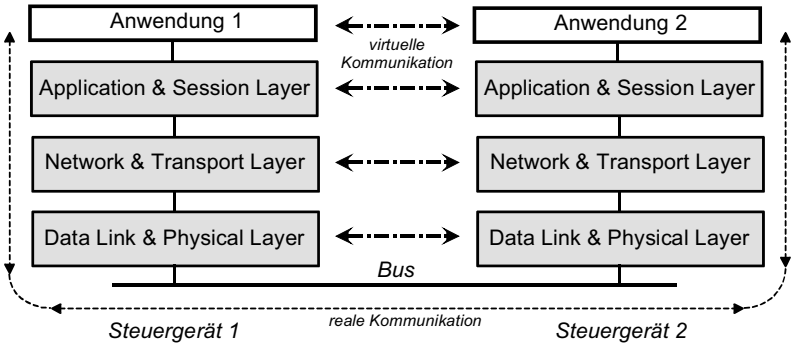

Abb. 2.9 Kommunikation zwischen Anwendungen über einen Protokollstapel

dem folgenden Verbindungsaufbau bei Telefongesprächen. Derartige Protokolle werden als *verbindungsorientiert* bezeichnet und bestehen üblicherweise aus den drei Phasen *Verbindungsaufbau*, *Datenaustausch* (Kommunikationsphase) und *Verbindungsabbau*. Können Geräte dagegen spontan ohne vorherige Absprache kommunizieren, bezeichnet man dies als *verbindungslose* Kommunikation.

Bei Kfz-Bussystemen werden Steuer- und Regeldaten praktisch immer im *Broadcast*-Verfahren versendet (Abb. 2.10). Das Steuergerät sendet seine Botschaften unaufgefordert. Jeder Empfänger entscheidet, ob er die Botschaften ignoriert oder den Dateninhalt auswertet (*Empfangs- oder Akzeptanzfilterung*). In der Regel besitzen die Protokolle zusätzlich *Adressierungsmechanismen*, mit denen ein einzelner (*Unicast*) oder mehrere Empfänger (*Multicast*) gezielt angesprochen werden können.

Für die Adressierung von Botschaften gibt es zwei gängige Verfahren:

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

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

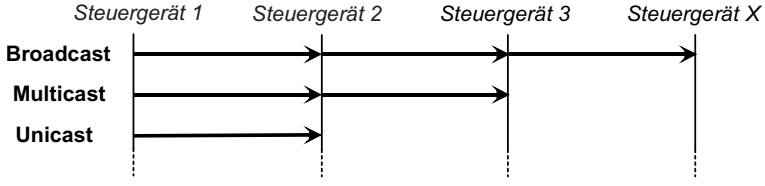

Abb. 2.10 Botschaftsversand an alle, eine Gruppe oder einen einzelnen Empfänger

(*Multicast*) oder alle Empfänger (*Broadcast*) existieren spezielle Adressen, die gelegentlich als funktionale Adressen bezeichnet werden. Da die Anzahl der Steuergeräte in einem Fahrzeug relativ klein ist, genügen für 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 (Identifier) 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.

Sowohl bei gerätebasierter als auch bei inhaltsbasierter Adressierung sehen bei einem Bussystem alle angeschlossenen Steuergeräte jede Botschaft und müssen auf Basis der Adresse entscheiden, ob sie die Botschaft tatsächlich empfangen und weiterverarbeiten oder einfach ignorieren wollen (*Akzeptanzfilterung*). Bei Bussystemen mit niedriger Bitrate findet die Akzeptanzprüfung in der Regel in der Protokollsoftware statt. Bei Protokollen mit höheren Bitraten findet die Protokollverarbeitung für die Schichten 1 und 2 und damit nach Möglichkeit auch die Akzeptanzfilterung dagegen im Kommunikationscontroller statt, einem speziellen IC oder einem auf dem Mikrocontroller des Steuergerätes integrierten Schaltungsteil, häufig auch als *Media Access Controller* (MAC) bezeichnet (Abb. 2.11). Bei der gerätebasierten Adressierung ist die Akzeptanzfilterung verhältnismäßig einfach, da nur die eigene Zieladresse sowie ggf. die Multicast- und Broadcast-Adressen erkannt werden müssen, während bei der inhaltsbasierten Adressierung oft mehrere Dutzend oder gar Hunderte *Adressen* (Botschaftskennungen) erkannt werden müssen. Gängige Kommunikationscontroller können typischerweise so programmiert werden, dass sie die Kennung von 8 bis 16 Botschaften hardwaremäßig erkennen, wobei mit Hilfe von Bitmasken auch ganze Bereiche von Adressen festgelegt werden können. Für die übrigen Botschaftskennungen, die ebenfalls empfangen werden sollen, muss eine Multicast-Kennung eingerichtet werden, für die die Akzeptanzfilterung dann softwaremäßig erfolgt.

Während der Sender bei *Broadcast*-Kommunikation in der Regel keine Bestätigung der Empfänger erhält (*Unacknowledged Communication*), erwartet der Sender bei *Unicast*-Sendungen oft eine positive Bestätigung (*Acknowledged Communication*) des Empfängers, dass die Botschaft korrekt empfangen wurde (Abb. 2.12). Verzichtet man, um das Bussystem zu entlasten, auf die positive Bestätigung, so wird zumindest eine Fehlermeldung (*Not Acknowledged*) erwartet, wenn einer der Empfänger einen Übertragungsfehler erkannt hat. In der Regel versendet der Sender die Botschaft darauf erneut. Ein Kommunikationsprotokoll, bei dem sich die Anwendung darauf verlassen kann, dass der *Network and*

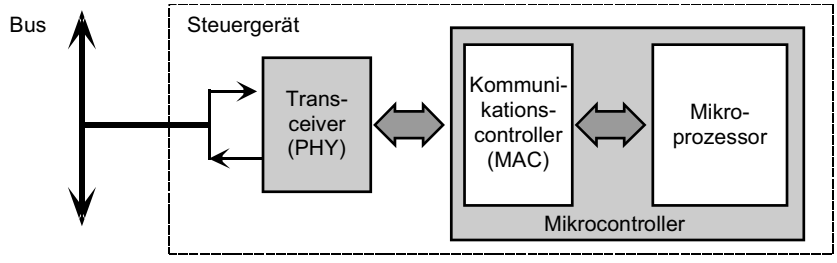

Abb. 2.11 Typisches Bus-Interface eines Steuergerätes (PHY Physical Interface, MAC Media Access Control)

Transport bzw. *Data Link Layer* selbstständig gewährleisten, dass die Information beim Empfänger korrekt ankommt, bezeichnet man als *gesicherte* Übertragung, das Gegenteil als *ungesichert*.

Das *Broadcast*-Verfahren, bei dem der Sender „ungefragt" Informationen versendet, ähnelt der Ausstrahlung von Rundfunk- und Fernsehprogrammen. Im Gegensatz dazu fordert ein Gerät beim *Request-Response*-Verfahren (Abb. 2.13) ein anderes Gerät durch eine Anfrage-Botschaft (*Request*) gezielt auf, Informationen (*Response*) zu liefern. Dieses Kommunikationsmodell wird unter dem Namen *Client-Server*-Verfahren auch im Internet verwendet, bei dem ein Web-Browser (*Client*) eine Anfrage an einen Web-*Server* stellt und von dort eine Antwort, z. B. eine HTML-Seite, erhält. Wird die Anfrage mit *Multicast*-Adressierung gesendet, muss der Sender mit Antworten von mehreren Empfängern rechnen. Benötigt man dieselbe Information wiederholt, entlastet das *Publisher-Subscriber*-Verfahren das Bussystem. Dabei meldet der an der Information interessierte Partner sein Interesse an der Information einmalig an (*Subscribe*), worauf der andere Partner die Information dann ohne weitere Anfragen regelmäßig (*Periodic*) oder bei Änderungen der Daten (*On Event*) liefert.

Ist die Informationsmenge größer als die Nutzdatenlänge einer einzelnen Botschaft, muss die Information vor der Übertragung auf der Senderseite zerstückelt (*Segmentierung*) und auf der Empfängerseite wieder zusammengesetzt (*Desegmentierung*) werden. Dies erfolgt in der Regel im *Network and Transport Layer* des Protokollstapels, dem sogenannten *Transportprotokoll* (siehe Kap. 4). Um den Empfänger vor zu großen Datenmengen oder zu schnell aufeinanderfolgenden Botschaften zu schützen, verfügen diese Protokolle in der

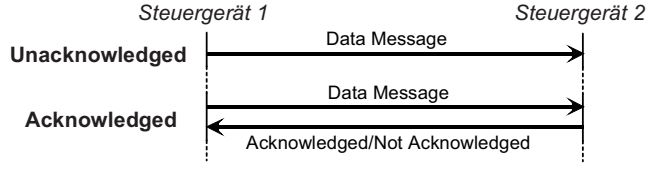

Abb. 2.12 Unbestätigte und bestätigte Kommunikation

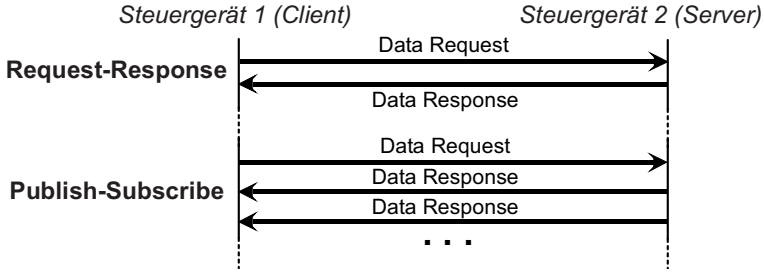

Abb. 2.13 Abfrage von Informationen

Regel auch über eine *Flusssteuerung*, bei der der Empfänger dem Sender mitteilt, wie viele Botschaften in welchem Abstand er empfangen bzw. den Sender zu einer Sendepause veranlassen kann.

Die Schnittstelle zwischen Anwendung und Protokollstapel bzw. den verschiedenen Schichten eines Protokollstapels verwendet üblicherweise die in Abb. 2.14 dargestellte Ablaufreihenfolge. Die Anwendung beauftragt den Protokollstapel mit dem Versenden einer Botschaft (*Transmit Request*) und lässt sich optional die durchgeführte Übertragung bestätigen (*Transmit Confirmation*). Danach wartet die Anwendung im einfachsten Fall, bis die Antwort eintrifft (*synchroner* oder *blockierender Betrieb*). Falls die Anwendung dagegen weiterarbeitet (*asynchroner* oder *nicht-blockierender* Betrieb), muss der Protokollstapel die Anwendung informieren, wenn eine Antwort eingetroffen ist. Diese Rückmeldung kann durch Setzen eines Flags erfolgen, das die Anwendung von Zeit zu Zeit abfragt (*Polling*), oder indem der Protokollstack eine Interrupt- oder Rückruffunktion der Anwendung aufruft (*Receive Indication*, *Notification* oder *Callback*).

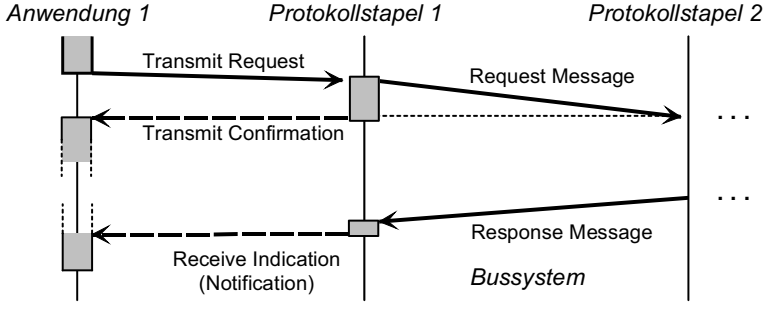

Abb. 2.14 Ablaufreihenfolge beim Senden und Empfangen von Botschaften