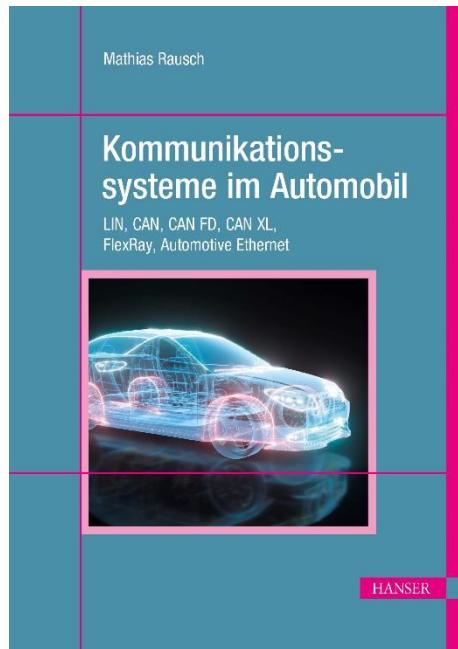


HANSER



Leseprobe

zu

Kommunikationssysteme im Automobil

von Mathias Rausch

Print-ISBN: 978-3-446-47035-4

E-Book-ISBN: 978-3-446-47457-4

Weitere Informationen und Bestellungen unter
<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446470354>
sowie im Buchhandel

© Carl Hanser Verlag, München

Vorwort

Seit über 20 Jahren kommen im Fahrzeugbereich die Bussysteme LIN und CAN zum Einsatz, FlexRay seit ca. 15 Jahren. Um die Funktionsweise dieser Kommunikationssysteme auch im Detail verstehen zu können, ist entsprechende Literatur notwendig. Standards beschreiben zwar das (geforderte) Verhalten der Systeme, erklären aber keine Zusammenhänge und Hintergründe. Artikel in Fachzeitschriften und im Internet sind vom Umfang begrenzt, sodass sie entweder nur einen Überblick geben oder nur einen speziellen Aspekt beleuchten. Hier setzt dieses Buch an. Die Funktionsweise von LIN, CAN und FlexRay soll anschaulich dargestellt werden. Viele Abbildungen sollen das Verständnis erleichtern. Durch die Weiterentwicklung der Drucktechnik ist es mittlerweile möglich, Fachbücher zu vertretbaren Kosten farbig zu drucken. Dies wird in diesem Buch genutzt. Durch den Einsatz der Farben werden Zusammenhänge einfacher verständlich. Für einen Teil der beschriebenen Funktionsweisen und Effekte wurden Messungen mit dem Oszilloskop vorgenommen, um einen Bezug zum praktischen Einsatz herzustellen. Die Farben der Messkurven korrespondieren dabei mit den Farben in den gezeichneten Abbildungen.

Kommunikationssysteme werden ständig weiterentwickelt. Dies ist nicht nur im Mobilfunk oder bei den Internettechnologien der Fall, sondern auch bei den Kommunikationssystemen im Fahrzeug. So wurde CAN weiterentwickelt zu CAN FD und CAN XL. Auf beide Weiterentwicklungen wird in diesem Buch eingegangen.

Die Entwicklung von Automotive Ethernet eröffnet den Fahrzeugherstellern neue Möglichkeiten in der Kommunikation im Fahrzeug und erschließt neue Anwendungen. Auf Ethernet basieren die Netzwerke, mit denen wir täglich arbeiten. Für die Nutzer ist es dabei nicht wichtig, wie die Kommunikation arbeitet. Muss hingegen ein Ingenieur oder eine Ingenieurin abwägen, ob ein CAN-Netzwerk durch ein Ethernet-Netzwerk ersetzt werden kann, muss er bzw. sie die Funktionsweisen beider Systeme genauer verstehen und gegeneinander abwägen können. Hier soll dieses Buch einen Beitrag leisten.

Das Buch geht auf die Funktionsweise der Protokollebene und des Physical Layers ein. Software, die die genannten Protokolle nutzt, ist hingegen nicht Bestandteil dieses Buches.

■ Danksagungen

Ich möchte mich bei allen bedanken, die zum Entstehen und Gelingen dieses Buches beigetragen haben. Dies sind die Studierenden der Elektrotechnik der Hochschule Landshut, die mit ihren Fragen aufgezeigt haben, an welchen Stellen das Verständnis der technischen Zusammenhänge anspruchsvoll ist. Ich möchte mich bei den zahlreichen Studierenden bedanken, die mit ihren Projekt- und Abschlussarbeiten zu Themen in diesem Buch beigetragen haben.

Bei meinen Kolleginnen und Kollegen möchte ich mich für die Diskussionen bedanken, die mir neue Einsichten gebracht und mich zum Nachdenken angeregt haben. Ein besonderer Dank geht an Magnus Hell von der Firma Infineon, der mir insbesondere bei CAN FD und CAN XL geholfen hat, alles fachlich korrekt darzustellen.

Bei Herrn Herzberg und Herrn Borck vom Hanser Verlag möchte ich mich für die sehr gute Zusammenarbeit bedanken. Ganz besonders möchte ich mich bei meiner Frau Eva bedanken, die mir das Schreiben dieses Buches ermöglicht hat.

Landshut, im Juli 2022

Mathias Rausch

Inhalt

Vorwort	V
1 Einleitung	1
1.1 Zum Inhalt dieses Buches	1
1.2 Vorhandene Literatur	2
1.2.1 Monografien	2
1.2.2 Weitere Literatur.....	3
1.3 Aufbau und Nomenklatur	3
1.4 LDF-Editor als Zusatzmaterial.....	4
I Grundlegende Funktionsweisen	5
2 Grundlagen (1)	7
2.1 Einführung in die Kommunikation	7
2.2 Begriffe	8
2.2.1 Elemente eines Kommunikationssystems	8
2.2.2 Begriffe zur Adressierung	9
2.2.2.1 Unicast	9
2.2.2.2 Multicast	9
2.2.2.3 Broadcast	10
2.2.2.4 Simplex.....	10
2.2.2.5 Voll duplex.....	11
2.2.2.6 Halbduplex	11
2.2.3 Bussysteme.....	11
2.2.4 Notwendigkeit einer Vernetzung.....	12
2.2.5 Weitere Begriffe.....	13
2.3 Aufbau	13
2.3.1 Aufbau eines Knotens	13
2.3.2 Aufbau eines Clusters.....	15
2.3.3 Komponenten	15
2.3.4 OSI-Modell	16

2.4	Topologien.....	17
2.4.1	Bustopologie	17
2.4.2	Sterntopologie	19
2.4.3	Ringtopologie	19
2.4.4	Hybride Topologien	20
2.5	Zugriffsverfahren	22
2.5.1	Einführungsbeispiel	22
2.5.2	Master-Slave-Prinzip	23
2.5.3	Zeitgesteuertes Zugriffsverfahren.....	24
2.5.4	Token Passing	24
2.5.5	Ereignisgesteuertes Zugriffsverfahren.....	24
2.5.6	Zusammenfassung Zugriffsverfahren	25
2.6	Codierung	26
2.6.1	Begriffe.....	27
2.6.2	Reihenfolge der Bits.....	28
2.6.3	Anzahl der Leitungen	29
2.6.3.1	Ein-Draht-Leitung	30
2.6.3.2	Zwei-Draht-Leitung	30
2.6.4	Signalübertragung	31
2.6.4.1	Asymmetrische Signalübertragung	31
2.6.4.2	Symmetrische Signalübertragung	32
2.6.5	Bitsynchronisation	32
2.6.5.1	Rahmenbildung für Resynchronisation	33
2.6.5.2	Bit Stuffing für Resynchronisation	34
2.6.6	Leitungscodes	35
2.6.6.1	NRZ-Codierung	35
2.6.6.2	RZ-Codierung	36
2.6.6.3	Manchester-Code.....	37
2.6.6.4	4B5B-Code	38
2.6.6.5	MLT-3-Code	39
2.6.6.6	4B3T-Code	39
2.7	Kenngrößen	40
2.7.1	Datenmenge	40
2.7.2	Datenübertragungsrate	41
2.7.3	Baudrate	42
2.7.4	Verzögerung.....	42
2.7.5	Jitter	43
2.8	Übersicht über Kommunikationsprotokolle im Automobil	44

3 LIN (1)	47
3.1 Einleitung.....	47
3.1.1 Anforderungen und Eigenschaften	47
3.1.2 Historie	48
3.1.3 Anwendung von LIN.....	49
3.2 Grundlagen.....	49
3.2.1 Begriffe.....	49
3.2.2 Aufbau eines LIN-Knotens	49
3.2.3 Aufbau eines LIN-Clusters	50
3.2.4 Einordnung ins OSI-Modell.....	51
3.2.5 LIN-Standards	51
3.3 LIN-Frames und LIN-Schedule	53
3.3.1 Aufbau eines LIN-Frames	53
3.3.2 Frame Header.....	53
3.3.2.1 Break-Feld	54
3.3.2.2 Sync-Feld	54
3.3.2.3 Identifier-Feld	55
3.3.3 Frame Response	55
3.3.3.1 Data-Feld	55
3.3.3.2 Checksum-Feld	56
3.3.4 LIN-Zyklus	57
3.3.5 Frame-Typen.....	58
3.3.5.1 Unconditional Frames	58
3.3.5.2 Event Triggered Frames	59
3.3.5.3 Sporadic Frames	61
3.3.5.4 Diagnostic Frames	61
3.3.5.5 Reserved Frames	62
3.4 Wecken eines LIN-Clusters.....	62
3.4.1 Zustände eines Slave-Knotens	62
3.4.2 Wakeup-Signal.....	63
3.4.3 Wiederholtes Wecken.....	63
3.5 Physical Layer	64
3.5.1 LIN-Transceiver	64
3.5.2 Buspegel	65
3.5.3 Bitraten-Toleranzen.....	67
3.5.4 Bit-Synchronisation.....	67
3.5.5 Bit Sampling	68
3.6 Frame-Timing.....	69
3.6.1 Berechnung der Frame-Länge und der Slotlänge	69
3.6.2 Slotlängen für verschiedene Konfigurationen.....	70

4 CAN (1)	73
4.1 Einleitung.....	73
4.1.1 CAN-Eigenschaften	73
4.1.2 Historie	74
4.1.3 Einsatzgebiete	75
4.2 Aufbau	76
4.2.1 Begriffe.....	76
4.2.2 Aufbau eines CAN-Knotens	76
4.2.3 Aufbau eines CAN-Clusters	77
4.2.4 OSI-Modell	77
4.2.5 CAN-Standards	79
4.2.6 Steckverbinder.....	79
4.3 Physical Layer	80
4.3.1 Einteilung	80
4.3.2 Buspegel	81
4.3.3 Transceiver	83
4.3.4 Terminierung	85
4.3.5 Topologie	86
4.3.6 Buslogik.....	87
4.4 Data Link Layer	88
4.4.1 CAN-Kommunikationsprinzip	88
4.4.2 Frame-Format	91
4.4.2.1 Data Frame Format.....	91
4.4.2.2 Remote Frame Format	93
4.4.2.3 Steuerfeld	93
4.4.3 Arbitrierung	94
4.4.4 CAN-Priorisierung	99
4.4.5 Bit Stuffing	100
4.4.6 Frame-Decodierung	100
4.5 Fehlerbehandlung.....	102
4.5.1 Fehlerarten	102
4.5.2 Fehlererkennung	103
4.5.3 Error Frame	104
4.5.4 Overload Frame	106
4.5.5 Fehlerzustände (Error Modes).....	107
4.5.6 Fehlerzähler (Auszug).....	109
4.5.7 Acknowledgement	109
4.5.8 Zusammenfassung Fehlererkennung	111

4.6	Berechnungen zur Busauslastung	112
4.6.1	Berechnung der Frame-Länge	112
4.6.2	Berechnung der maximalen Anzahl an Stuff Bits	112
4.6.3	Berechnung der Frame-Übertragungszeit und der Busauslastung	113
4.6.4	Maximale Anzahl an Frames in einem Cluster	116
4.6.5	Maximale Nutzdatenrate in einem Cluster	116
5	FlexRay (1)	119
5.1	Einleitung	119
5.1.1	Was ist FlexRay?	119
5.1.2	FlexRay-Eigenschaften	119
5.1.3	Historie	120
5.2	Begriffe und Aufbau	121
5.2.1	Begriffe	121
5.2.2	Aufbau	121
5.2.3	OSI-Modell	123
5.2.4	FlexRay-Standards	123
5.3	Physical Layer	125
5.3.1	Medien	125
5.3.2	Buspegel	126
5.3.3	Topologien	126
5.3.3.1	Bustopologie	127
5.3.3.2	Sterntopologie	127
5.3.3.3	Gemischte Topologien	128
5.3.4	Sternkoppler	129
5.3.5	Terminierung	130
5.4	Data Link Layer	132
5.4.1	Aufbau eines Kommunikationszyklus	132
5.4.1.1	Statisches Segment	133
5.4.1.2	Dynamisches Segment	135
5.4.2	Zeithierarchie	137
5.4.3	Frame-Format	138
5.4.3.1	Header	138
5.4.3.2	Payload	139
5.4.3.3	Trailer	139
5.4.4	Codierung	140

5.5 Uhrensynchronisation	141
5.5.1 Uhrenabweichungen	142
5.5.2 Uhrenkorrektur	143
5.5.2.1 Offset-Korrektur	143
5.5.2.2 Raten-Korrektur	143
5.5.2.3 Uhrenkorrektur bei FlexRay	144
5.6 Konfiguration	145
5.6.1 Schreibweisen	146
5.6.2 Berechnung der Zykluszeit	146
5.6.3 Berechnung der statischen Slotlänge	148
5.7 Übungen	151
II Erweiterte Prinzipien und spezielle Mechanismen	155
6 Grundlagen (2)	157
6.1 Methoden zur Berechnung eines Prüfwerts	157
6.1.1 Paritätsprüfung	157
6.1.2 Berechnung der Prüfbits bei LIN	158
6.1.2.1 Bestimmung der Paritätsbits im Identifier-Feld	158
6.1.2.2 Bestimmung der Prüfbits für das Checksum-Feld	159
6.1.3 Kreuzparitätsprüfung	160
6.1.4 Zyklische Redundanzprüfung	162
6.2 Kommunikationsarten	166
6.2.1 Signalorientierte Kommunikation	166
6.2.2 Serviceorientierte Kommunikation	168
6.2.3 Vergleich der beiden Kommunikationsarten	170
6.3 Leitungsreflexionen	171
6.3.1 Entstehung von Reflexionen	171
6.3.2 Leitungsparameter	172
6.3.3 Messung von Reflexionen	173
6.3.4 Ringing	176
7 LIN (2)	179
7.1 Entwicklungsprozess	179
7.1.1 Überblick	179
7.1.2 LIN Description File (LDF)	180
7.1.3 Node Capability File (NCF)	183

7.2	Anwendungsbeispiel	186
7.2.1	Beschreibung der Knoten	186
7.2.2	Signalcodierung	189
7.2.3	Signaldefinition	190
7.2.4	Frames	193
7.2.5	Schedules	196
8	CAN (2)	199
8.1	Bit-Timing und Bitsynchronisation	199
8.1.1	Diskretisierung des empfangenen Spannungsverlaufs	199
8.1.2	Bit-Timing	201
8.1.3	Bitsynchronisation	203
8.2	Weiterentwicklungen von CAN	204
8.2.1	Grenzen und Erweiterungen von CAN	204
8.2.2	Erweiterter Identifier	206
8.2.3	Erweiterungen im Physical Layer	207
8.2.3.1	Problembeschreibung	207
8.2.3.2	Lösungsansatz	208
8.2.3.3	Wecken mittels Wakeup-Pattern (WUP)	209
8.2.3.4	Wecken mittels Wakeup-Frame (WUF)	212
8.3	CAN FD	214
8.3.1	Frame-Format und Funktionsweise	214
8.3.2	CAN FD CRC	216
8.3.3	Bit Stuffing	218
8.3.4	Übersicht über CAN-Frame-Typen	219
8.3.5	Hinweise zu Implementationen	219
8.3.6	CAN-FD-Bitraten	220
8.3.7	Effekt der längeren Payload und der höheren Datenbitrate	220
8.3.8	CAN-FD-SIC-Ansatz	221
8.3.9	CAN-FD-Physical-Layer-Spezifikationen	226
8.4	CAN XL	226
8.4.1	Eigenschaften und Anwendung von CAN XL	226
8.4.2	Frame-Aufbau und prinzipielle Funktionsweise	227
8.4.3	Arbitrierungsfeld	227
8.4.4	Steuerfeld	228
8.4.4.1	ADS-Feld	229
8.4.4.2	Service Data Unit Type	229

8.4.4.3	Data Length Code.....	229
8.4.4.4	Stuff-Bit-Zähler	229
8.4.4.5	Preface Cyclic Redundancy Check	230
8.4.4.6	Virtual CAN ID	230
8.4.4.7	Acceptance-Feld	230
8.4.5	Datenfeld	231
8.4.6	CRC-Feld	231
8.4.7	Acknowledge-Feld und EOF	232
8.4.8	Bit Stuffing	232
8.4.9	Bitraten-Umschaltung.....	233
8.4.10	Vergleich der Header von Classical CAN, CAN FD und CAN XL	237
8.5	Realisierung	238
8.5.1	CAN-Controller	238
8.5.2	CAN-Transceiver	239
8.5.3	CAN-Oszilloskope	241
8.5.4	Typische Schaltung eines CAN-Knotens	241
8.6	Berechnung der Busauslastung beim Classical CAN	242
8.7	Übungen	244
9	FlexRay (2).....	247
9.1	Protokollzustände	247
9.2	Wakeup	249
9.2.1	Cluster-Wakeup	249
9.2.2	Wakeup-Pattern	250
9.2.3	Wakeup-Szenario mit sich überlappenden Low-Phasen	252
9.2.4	Wakeup-Szenario mit sich teilweise überlappenden Low-Phasen	253
9.2.5	Wakeup-Szenario mit sich nicht überlappenden Low-Phasen	254
9.3	Startup	255
9.3.1	Startup von zwei Coldstart-Knoten	255
9.3.2	Startup eines Non-Coldstart-Knotens	257
9.3.3	Startup mit nur einem Coldstart-Knoten	258
9.4	Bus Guardian	259
9.4.1	Eigenschaften.....	259
9.4.2	Funktionsweise	259
9.4.3	Realisierung.....	261
9.5	Mechanismen und Effekte	261
9.5.1	Null-Frames	261
9.5.2	Verkürzung der Transmission Start Sequence.....	263

9.5.3	Uhrensynchronisation	265
9.5.3.1	Ablauf der Uhrensynchronisation	265
9.5.3.2	Erfassen der Messwerte	265
9.5.3.3	Berechnung der Offset-Korrekturwerte	266
9.5.3.4	Berechnung der Raten-Korrekturwerte	267
9.5.3.5	Anwendung der Korrekturwerte	268
9.5.4	Bit-Decodierung	268
9.5.5	Asymmetrien	269
9.5.5.1	Was ist eine Bitasymmetrie?	269
9.5.5.2	Auswirkungen von Bitasymmetrien	270
9.5.5.3	Treiberstufe im Protokollcontroller und im Transceiver	270
9.5.5.4	Bitasymmetrie durch Schwellwertverschiebung	271
9.5.5.5	Bitasymmetrie durch Änderung der Flankensteilheit	272
9.5.5.6	Bitasymmetrie durch Unterschiede in der Verzögerungszeit	272
9.6	Applizierung	273
9.6.1	FlexRay im BMW X5	273
9.6.2	FlexRay im 7er BMW	275
9.7	Berechnung der Minislottlänge	276
9.8	Übungen	278
10	Automotive Ethernet	281
10.1	Einleitung	281
10.1.1	Versionen von Automotive Ethernet	281
10.1.2	Entwicklungsgeschichte von (Automotive) Ethernet	282
10.2	Aufbau	282
10.2.1	Aufbau eines Ethernet-Knotens	282
10.2.2	Topologien	283
10.2.3	Interfaces	284
10.2.4	Media Dependent Interface	284
10.2.4.1	ESD-Schutz	285
10.2.4.2	Common Mode Termination	286
10.2.4.3	AC-Kopplung	286
10.2.4.4	Gleichtaktdrossel	287
10.2.4.5	Low-Pass-Filter	287
10.2.5	Media Independent Interface	287
10.2.5.1	MII	287
10.2.5.2	RMII	287
10.2.5.3	RGMII und SGMII	287

10.3 Ethernet-Versionen	289
10.3.1 Ethernet IEEE 100BASE-TX	289
10.3.2 Ethernet IEEE 1000BASE-T	289
10.3.3 Ethernet IEEE 100BASE-T1	290
10.3.4 Versionen von Automotive Ethernet.....	291
10.4 Physical Layer	291
10.4.1 Aufbau des Transceivers	291
10.4.1.1 Bitmanipulationen im Sendepfad des Physical Coding Sublayers	292
10.4.1.2 Bitmanipulationen im Empfangspfad des Physical Coding Sublayers	294
10.4.2 Leitungscodes	295
10.4.2.1 Spannungswerte	295
10.4.2.2 Zuordnungs- und Leitungscodes	295
10.4.2.3 Signalhandling	296
10.4.2.4 Oszilloskopmessungen	298
10.4.3 Startup	299
10.4.3.1 Transceiver-Modi und Startup-Sequenz.....	299
10.4.3.2 Leitungscode 3B2T in der Trainingsphase	301
10.4.3.3 Leitungscode 3B2T in der Sendephase	301
10.4.3.4 Diagnosemöglichkeiten.....	302
10.5 Data Link Layer	303
10.5.1 Ethernet-Controller	303
10.5.2 Ethernet Frame Format	303
10.5.3 Adressierung	306
10.5.3.1 Unicast	306
10.5.3.2 Broadcast	306
10.5.3.3 Multicast	307
10.6 Weitere Aspekte zu Ethernet	308
10.6.1 Höhere Protokolle	308
10.6.2 Kopplungselement für Messungen	309
10.6.3 Implementierungen nach 100BASE-T1	311
10.6.4 OPEN Alliance	311
Literatur	313
Index	315

1

Einleitung

■ 1.1 Zum Inhalt dieses Buches

Bei der Kommunikation kann grundsätzlich in drahtgebundene und in drahtlose Kommunikation unterschieden werden. Des Weiteren kann die Kommunikation danach eingeteilt werden, in welchem Bereich sie zur Anwendung kommt. So gibt es Kommunikationssysteme für Gebäude, für Büros, für die Produktion, für Maschinen, für Fahrzeuge, für Wohnungen, im medizinischen Bereich usw. In diesem Buch geht es ausschließlich um drahtgebundene Kommunikation, wobei der Fokus auf Kommunikationssystemen liegt, die in Fahrzeugen zum Einsatz kommen bzw. dafür entwickelt wurden. Dies bedeutet aber nicht, dass diese Kommunikationssysteme nicht auch in anderen Bereichen zum Einsatz kommen.

Ziel des Buches ist es, die Funktionsweise der betrachteten Kommunikationssysteme zu erläutern. Es werden folgende Kommunikationssysteme behandelt:

- LIN
- CAN mit den Erweiterungen CAN FD und CAN XL
- FlexRay
- Automotive Ethernet

Das Buch ist in zwei Teilen organisiert. Im ersten Teil wird nach einigen Grundlagen die prinzipielle Funktionsweise von LIN, CAN und FlexRay mit den wichtigsten Eigenschaften dargestellt. Im zweiten Teil wird auf Funktionen und Details eingegangen, die für das allgemeine Verständnis nicht so wichtig sind. Damit ist das Buch sowohl für Leser geeignet, die sich einen Überblick über die behandelten Kommunikationssysteme verschaffen wollen, als auch für Leser, die es technisch genauer wissen möchten.

Die Gliederung des Buches in zwei Teile erleichtert es der ersten Gruppe, sich nicht in technischen Details zu verlieren, und der zweiten Gruppe, sich nicht durch Grundlegendes zu langweilen. Das Gleiche gilt für Studierende. Teil 1 ist für Studierende geeignet, die ein Modul zum Thema Kommunikation im Fahrzeug hören. Teil 2 erfüllt die Erwartungen von Studierenden, deren Studium einen Schwerpunkt im Bereich Kommunikation und Bussysteme hat.

Im ersten Kapitel des ersten Teils des Buches wird auf die Kommunikation allgemein eingegangen. Es werden Begriffe eingeführt und grundlegende technische Prinzipien dargestellt. Basierend auf diesen Prinzipien werden in den drei folgenden Kapiteln die Funktionsweisen von LIN, CAN und FlexRay erläutert.

Der Teil 2 des Buches beginnt wieder mit einem Kapitel *Grundlagen*. Hier wird auf Verfahren zur Berechnung von Prüfsummen eingegangen. Des Weiteren werden die Unterschiede zwischen der signalorientierten und der serviceorientierten Kommunikation dargestellt. Abgeschlossen wird dieses Kapitel mit Ausführungen zu Reflexionen auf Leitungen.

Im [Kapitel 7](#) wird auf den Entwicklungsprozess bei LIN eingegangen und es wird ein ausführliches Beispiel für eine Kommunikation mit LIN gezeigt.

Im [Kapitel 8](#) wird besonders auf die Weiterentwicklungen des CAN-Protokolls eingegangen mit einer ausführlichen Darstellung von CAN FD und CAN XL.

Im [Kapitel 9](#) werden weitere Mechanismen von FlexRay erläutert, wie das Starten der Kommunikation, das Prinzip der Null-Frames, der Bus Guardian und die Uhrensynchronisation.

Ein Kapitel über Automotive Ethernet rundet diesen Teil des Buches ab und gibt einen Einblick, wohin sich die Kommunikation in Fahrzeugen weiterentwickelt.

Die Beispiele, die zum Erläutern bestimmter technischer Sachverhalte benutzt werden, kommen aus dem Bereich der Automobiltechnik. Sie sind bewusst einfach gehalten, um die Beispiele nicht ausführlich erläutern zu müssen. Dem Autor ist bewusst, dass diese Beispiele nicht oder nicht mehr so in einem modernen Fahrzeug eingesetzt werden. Im Buch geht es um Kommunikationssysteme und nicht um den Aufbau eines Kraftfahrzeugs.

■ 1.2 Vorhandene Literatur

Dieser Abschnitt gibt eine Übersicht über bereits vorhandene Literatur zu den Kommunikationssystemen LIN, CAN, FlexRay und Automotive Ethernet.

1.2.1 Monografien

Bücher zu mehreren Kommunikationssystemen

ZIMMERMANN und SCHMIDGALL geben in der 5. Auflage ihres Buches [[Zimmermann](#)] eine Einführung in CAN, LIN, FlexRay, MOST und Automotive Ethernet. Es werden die grundlegenden Prinzipien dargestellt. Der Schwerpunkt des Buches liegt auf den höheren Softwareschichten.

In dem englischsprachigen Buch [[Paret07](#)] behandelt PARET neben CAN auch LIN und FlexRay sowie weitere Bussysteme. Ebenfalls in Englisch ist das Buch von NAVET und SIMONOT-LION, in welchem Aspekte von CAN und FlexRay behandelt werden [[Navet](#)].

Bücher zu LIN

Über LIN gibt es nur ein einziges Buch, welches 2005 von GRZEMBA und VON DER WENSE herausgegeben wurde [[Grzembba](#)]. Von diesem Buch gab es nur eine Auflage, die schon länger vergriffen ist. LIN wurde zwar nicht mehr weiterentwickelt, aber eine Erklärung des Protokolls außer in diesem Buch gibt es nicht.

Bücher zu CAN

Seit vielen Jahren gibt es zwei Werke, die CAN beschreiben [[Etschberger](#), [Lawrenz](#)]. Das Buch von ETSCHBERGER ist 2002 in der 3. Auflage erschienen, das Buch von LAWRENZ 2011 in der 5. Auflage. Die Weiterentwicklung von CAN zu CAN FD erfolgte ab 2011 und ist folglich in keinem der Standardwerke zu CAN enthalten.

Bücher zu FlexRay

Das erste Buch zu FlexRay ist von RAUSCH 2007 erschienen. Es beschreibt detailliert die Funktionsweise von FlexRay [Rausch]. Der Autor geht auch auf die Konfiguration eines FlexRay-Clusters ein und gibt eine Übersicht über die zu dieser Zeit verfügbaren Implementierungen.

2011 erschien zuerst in Französisch und 2012 in Englisch ein Buch von PARET [Paret12]. Er geht auf alle Aspekte von FlexRay ein (Physical Layer, Protokoll, Uhrensynchronisation, Startup) und gibt wie das Buch von RAUSCH eine detaillierte Einführung in FlexRay.

Bücher zu Automotive Ethernet

Das Buch zu Automotive Ethernet ist von MATHEUS und KÖNIGSEDER und gibt eine Einführung in Automotive Ethernet [Matheus]. Dieses Buch ist mittlerweile in der zweiten Auflage erschienen. Zu jedem der sieben Kapitel gibt es ein eigenes Literaturverzeichnis, welches je Kapitel bis zu 100 Quellen enthält. Neben der Funktionsweise werden die Entstehungsgeschichte dieses Protokolls sowie die Entscheidungsprozesse bei BMW und die Hintergründe dazu dargestellt. Da wichtige Mechanismen von Automotive Ethernet auf 100BASE-TX und 1000BASE-T basieren, ist deren Kenntnis notwendig, um Automotive Ethernet verstehen zu können.

1.2.2 Weitere Literatur

Webinars

Von der Firma Vector gibt es zu allen vier Kommunikationssystemen Webinars sowohl in Deutsch als auch in Englisch, die eine Einführung in das jeweilige Kommunikationssystem geben. Abgerundet werden diese Webinars mit einem Online-Test, um das Gelernte zu überprüfen [VectorLIN, VectorCAN, VectorFlexRay, VectorAE].

Standards

Alle in diesem Buch behandelten Protokolle sind international standardisiert. Diese Standards können bei den Standardisierungsorganisationen ISO und IEEE bezogen werden. In den jeweiligen Kapiteln sind die betreffenden Standards genannt.

■ 1.3 Aufbau und Nomenklatur

In der Literatur und in den Standards werden viele Fachtermini verwendet. Da die Standards in Englisch verfasst sind, werden viele der Fachtermini auch in der deutschsprachigen Literatur ohne Übersetzung übernommen. Dieser Ansatz wird auch in diesem Buch verfolgt. Es wurde versucht, die englischen Begriffe weitestgehend aus den Standards zu übernehmen. Insbesondere in Kombination mit deutschen Wörtern wurde sich für eine Schreibweise entschieden. Es wären aber auch andere Schreibweisen möglich.

Wenn im Buch Personen, z. B. ein Anwender oder ein Nutzer, genannt werden, wird nur die männliche Form verwendet. Dies erfolgt, um eine bessere Verständlichkeit der technischen Zusammenhänge zu erreichen. An diesen Stellen ist die konkrete Person nicht wichtig. Es geht um technische Zusammenhänge und nicht um gesellschaftliche Aussagen. Gemeint sind immer Personen jeglichen Geschlechts.

Für eine bessere Anschaulichkeit werden in den Abbildungen Farben verwendet. Für bestimmte Komponenten bzw. Kategorien werden die gleichen Farben immer wieder in verschiedenen Abbildungen verwendet. Eine Zuordnung für einige der Farben zu Komponenten ist in [Tabelle 1.1](#) gegeben.

Tabelle 1.1 Verwendung der Farben im Buch

Farbe	Verwendung
	CAN-H (CAN), BP (FlexRay)
	CAN-L (CAN), BM (FlexRay)
	Anwendung, Mikrocontroller, Host
	Protokollcontroller, Data Link Layer
	Transceiver, Bustreiber, Physical Layer
	Knoten 1
	Knoten 2
	Knoten 3
	Differenzspannung auf dem Bus

■ 1.4 LDF-Editor als Zusatzmaterial

Um eine Kommunikation mit einem Bussystem zu realisieren, ist zum einen die Hardware notwendig, zum anderen muss die Kommunikation definiert und softwaremäßig umgesetzt werden. Zur Unterstützung der Definition der Kommunikation wurden für verschiedene Bussysteme Dateiformate und Tools entwickelt, die den Anwender möglichst gut unterstützen sollen. So wurde für LIN das *LIN Description File (LDF)* spezifiziert, in dem alle Aspekte der Kommunikation in einem standardisierten Format beschrieben werden. Solch ein LDF ist ein Textfile im ASCII-Format, welches mit einem Texteditor erzeugt werden kann. Wesentlich einfacher ist es, einen entsprechenden Editor zu nutzen, der den Anwender unterstützt, sich um die Syntax kümmert und Fehler detektiert.

Solch ein Editor für die Erzeugung von LDFs steht als Zusatzmaterial zu diesem Buch zur Verfügung. Er wurde an der Hochschule Landshut in C# entwickelt und basiert auf dem .NET Framework 4.5.2 von Microsoft. Damit er auf einem Computer läuft, muss die .NET-Umgebung installiert sein.

Der Editor deckt alle notwendigen Schritte für die Erstellung einer signalbasierten Kommunikation ab. Selbst wenn man die Kommunikation nicht mit LIN realisieren möchte oder kann, hilft der Editor, sich über die Struktur der Daten für die Kommunikation klar zu werden. Der Editor steht als Download beim Hanser Verlag zur Verfügung. Er deckt den kompletten Umfang ab, der im LIN-Standard beschrieben ist.

Grundlegende Funktionsweisen

2

Grundlagen (1)

■ 2.1 Einführung in die Kommunikation

Das Wort *Kommunikation* bedeutet laut Duden die Verständigung untereinander, die mit Hilfe von Sprache und Zeichen stattfinden kann. Ursprünglich kommt das Wort aus dem Lateinischen von *communicare* und bedeutet so viel wie *teilen*. Im heutigen Sprachgebrauch wird mit Kommunikation das Teilen bzw. der Austausch von Informationen bezeichnet. Eines der wichtigsten Hilfsmittel für die Kommunikation des Menschen ist die Sprache.

Schon jeder Säugling kommuniziert mit seiner Umwelt. Schreien bedeutet, dass etwas nicht passt (Hunger, Schmerzen, Windel voll), nicht schreien bedeutet, dass alles okay ist. Diese sehr einfache Kommunikation entwickelt sich mit der Zeit weiter. Ein Kind lernt Wörter zu verstehen und zu sprechen, Sätze zu bilden, Grammatik anzuwenden. Später lernt es, wann es sprechen kann und wann es besser den Mund hält. Manche verlernen das auch wieder. Dies alles geschieht spielerisch und unbewusst.

Wie kommunizieren wir?

Da gibt es viele Formen und Aspekte, von denen hier einige genannt sind:

- durch Sprache, Wörter, Sätze
- unter Benutzung von Grammatik
- es gibt einen Sprecher und einen oder mehrere Zuhörer
- direkt, per Telefon oder Computer
- durch gesprochene Sätze, durch geschriebene Sätze
- und zusätzlich durch non-verbale Kommunikation, wie Gesten, Körperhaltung, Handzeichen, Augenzwinkern, Lächeln usw.

Was jeder Mensch als Kind in Bezug auf Kommunikation gelernt hat und, ohne darüber nachdenken zu müssen, anwendet, muss bei einem technischen System beschrieben (spezifiziert) und dann technisch umgesetzt werden. Dies erfolgt in einem Kommunikationsprotokoll. Darin wird u. a. festgelegt:

- wie Informationen technisch dargestellt werden
- wie Informationen technisch übertragen werden
- wie kleine Informationseinheiten zu größeren zusammengesetzt werden
- wann ein Teilnehmer senden darf (und wann nicht)
- wie die übertragene Information gesichert wird
- was gemacht wird, wenn bei der Übertragung ein Fehler auftritt

Ein Kommunikationsprotokoll ist eine Menge von Regeln für die Darstellung der Information, die Nachrichtenübermittlung, die Synchronisation und die Fehlererkennung, welche erforderlich sind, um Informationen über einen Kommunikationskanal auszutauschen.

In diesem Kapitel werden grundlegende Prinzipien, Begriffe und Methoden, die bei der *technischen* Kommunikation eine Rolle spielen, erläutert. In den folgenden Kapiteln werden vier konkrete Kommunikationssysteme ausführlicher beschrieben.

■ 2.2 Begriffe

Um Systeme zu beschreiben, werden Begriffe verwendet, die Teilsysteme und Komponenten, aber auch Methoden und Prinzipien benennen. Dabei ist es nicht ungewöhnlich, dass in unterschiedlichen Fachbereichen verschiedene Begriffe für die gleiche Sache verwendet werden. So werden z. B. für Informationsentities die Begriffe Nachricht, Botschaft, Telegramm, Message, Frame, Daten, Datei und File verwendet, wobei die Liste keinen Anspruch auf Vollständigkeit erhebt.

Ebenso kommt es vor, dass der gleiche Begriff für unterschiedliche Komponenten oder Methoden verwendet wird. Bei Abkürzungen tritt dieses Problem noch wesentlich stärker auf. So wird z. B. die Abkürzung *IP* für *Internet Protocol*, aber auch für *Intellectual Property* verwendet.

In diesem Abschnitt werden die wichtigsten Begriffe für die Erläuterung der Kommunikationssysteme eingeführt. Gibt es mehrere Begriffe, wird einer ausgewählt, der dann durchgängig bei der Beschreibung der verschiedenen Kommunikationssysteme verwendet wird. Dies soll der Leserin und dem Leser das Verständnis erleichtern. Ein einmal verstandenes Prinzip muss nicht erneut durchdacht werden, nur weil plötzlich ganz andere Begriffe verwendet werden. Dies ermöglicht es auch, die verschiedenen Systeme besser miteinander zu vergleichen.

2.2.1 Elemente eines Kommunikationssystems

Um kommunizieren zu können, sind Sender (engl.: Transmitter) und Empfänger (engl.: Receiver) notwendig. Ein Kommunikationsteilnehmer kann gewöhnlich senden und empfangen. Er wird Teilnehmer oder *Knoten* genannt.

Die Daten, die die Knoten miteinander austauschen, werden als Nachrichten, Botschaften, Telegramme oder *Frames* bezeichnet.

Mehrere Knoten bilden ein *Cluster*, wenn sie über *ein* Kommunikationssystem miteinander verbunden sind.

Aus mehreren Clustern kann über *Gateways* ein *Netzwerk* gebildet werden. In einem Netzwerk sind *mehrere* gleiche oder verschiedene Kommunikationssysteme miteinander verbunden.

Diese gedankliche Hierarchie sowie diese Begriffe könnten genauso gut auf einen Ort, z. B. ein Dorf, angewendet werden. Dabei ergibt sich folgende Zuordnung:

- Die Häuser würden den *Knoten* entsprechen.
- Alle Häuser an einer Straße würden ein *Cluster* bilden.

- Die einzelnen Straßen sind durch Kreuzungen miteinander verbunden. Eine Kreuzung würde einem *Gateway* entsprechen.
- Alle Häuser, Straßen und Kreuzungen bilden das Dorf. Dies würde dem *Netzwerk* entsprechen.
- Auf den Straßen bewegen sich Menschen und Fahrzeuge. Dies wären die *Frames*.

2.2.2 Begriffe zur Adressierung

Bei den ersten drei Begriffen geht es darum, wie viele Empfänger die Informationen des Senders empfangen können. Bei den weiteren Begriffen geht es darum, in welcher Richtung Informationen zwischen zwei Teilnehmern ausgetauscht werden können.

2.2.2.1 Unicast

Beim Unicast kommuniziert ein Teilnehmer einer Gruppe mit genau einem anderen Teilnehmer der Gruppe. Es bezeichnet eine 1:1-Kommunikation (Bild 2.1). Es sendet also immer nur *ein* Teilnehmer eine Nachricht an *einen* anderen Teilnehmer zu einem Zeitpunkt. Nach dieser Nachricht kann er eine weitere Nachricht an diesen oder einen anderen Teilnehmer senden oder ein anderer Teilnehmer sendet.

Beispiele sind das Telefonieren, ein 1:1-Chat und ein Vier-Augen-Gespräch.

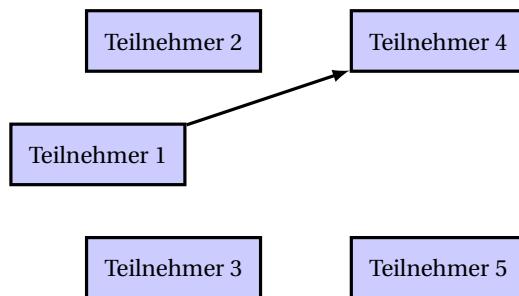
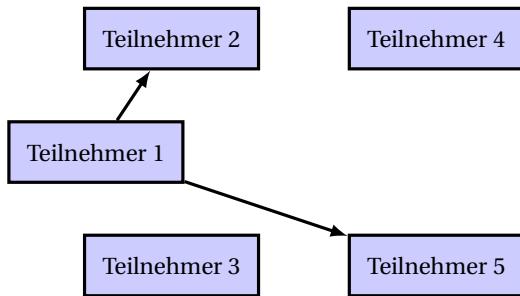


Bild 2.1 Kommunikation bei Unicast

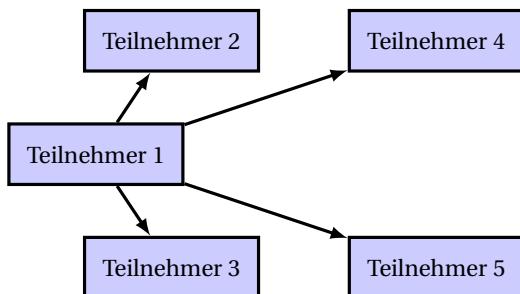
2.2.2.2 Multicast

Ein Teilnehmer sendet beim Multicast eine Nachricht an eine Teilmenge einer Gruppe (Bild 2.2). Nach dieser Nachricht kann er einer anderen Teilmenge der Gruppe eine Nachricht schicken oder ein anderer Teilnehmer sendet. Es kann also immer nur *ein* Teilnehmer eine Nachricht an eine *Teilgruppe* zu einem Zeitpunkt schicken. Ein Beispiel für Multicast ist eine Videokonferenz, zu der nicht alle Mitglieder der Gruppe den Zugangscode erhalten haben. Ein weiteres Beispiel wären Gespräche an einer langen Tafel. Nur die Gäste, die nicht zu weit weg von einem Sprecher sitzen, können den Ausführungen folgen. Alle anderen Gäste können den Sprecher akustisch nicht verstehen und hören wahrscheinlich einer anderen Sprecherin zu.

**Bild 2.2** Kommunikation bei Multicast

2.2.2.3 Broadcast

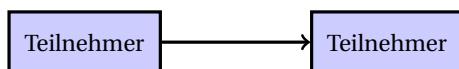
Ein Teilnehmer einer Gruppe kommuniziert mit allen anderen Teilnehmern der Gruppe. Es sendet also *ein* Teilnehmer eine Nachricht an *alle* anderen Teilnehmer der Gruppe zu einem Zeitpunkt (Bild 2.3). Bei vielen Kommunikationssystemen (z. B. LIN, CAN, FlexRay) ist Broadcast die übliche und einzige Adressierungsart. Beispiele für Broadcast sind das Radio, das Fernsehen und ein Vortrag.

**Bild 2.3** Kommunikation bei Broadcast

Die Unterscheidung zwischen Multicast und Broadcast ist nicht immer eindeutig. Hier kommt es darauf an, wie die Gruppe, die empfängt, definiert ist. So kann das Radioprogramm technisch zwar prinzipiell von allen Radiogeräten empfangen werden, aber nur die Geräte, die an sind und auf den betrachteten Sender eingestellt sind, können das Programm empfangen.

2.2.2.4 Simplex

Beim Simplex sendet ein Teilnehmer ausschließlich, der andere empfängt nur (Bild 2.4). Die Kommunikation kann also nur in einer Richtung erfolgen. Es ist technisch nicht möglich, dass die Kommunikation auch in der anderen Richtung erfolgen kann, weil es keinen Rückkanal gibt. Beispiele für eine Simplex-Kommunikation sind Radio, Fernseher, Pager und Diktatoren.

**Bild 2.4** Kommunikation bei Simplex

2.2.2.5 Vollduplex

Beim Vollduplex können die Teilnehmer gleichzeitig Senden und Empfangen. Es ist also möglich, Daten in beide Richtungen zur gleichen Zeit zu übertragen (Bild 2.5). Beispiele für dieses Prinzip sind das Telefon und die Demokratie.

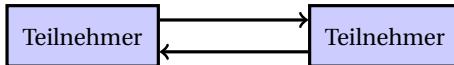


Bild 2.5 Kommunikation bei Vollduplex

2.2.2.6 Halbduplex

Beim Halbduplex kann auch jeder Teilnehmer senden und empfangen, aber beides gleichzeitig ist nicht möglich (Bild 2.6). Senden und Empfangen kann also nur nacheinander erfolgen. Ein Beispiel sind Funkgeräte, bei denen die Sendetaste gedrückt werden muss, um zu senden. Während des Sendens ist kein Empfang möglich.

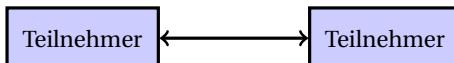


Bild 2.6 Kommunikation bei Halbduplex

Es gibt noch das Prinzip des *Dual-Simplex*. Das Ergebnis ist das Gleiche wie beim Vollduplex. Allerdings gibt es getrennte Sende- und Empfangswege. Ein Beispiel wäre eine Internetverbindung im Downstream über Satellit und im Upstream über Telefon. Um Vollduplex von Dual-Simplex unterscheiden zu können, müssen Details über die technische Realisierung bekannt sein.

2.2.3 Bussysteme

Bisher wurde der Begriff *Kommunikationssystem* verwendet, um zu kennzeichnen, dass zwei oder mehr Teilnehmer miteinander Informationen austauschen. Im Fahrzeug, der Automatisierungstechnik und anderen Bereichen wird häufig der Begriff *Bussystem* verwendet. Damit wird gewöhnlich ein Kommunikationssystem mit bestimmten Eigenschaften bezeichnet. Eine Eigenschaft ist, dass ein solches System drahtgebunden ist. Der Begriff *Bus* wird verwendet für Bussysteme und für die Bustopologie.

Bussystem: Ein Bus ist ein System zur Datenübertragung zwischen mehreren Teilnehmern über einen gemeinsamen Übertragungsweg, bei dem die Teilnehmer nicht an der Datenübertragung zwischen anderen Teilnehmern beteiligt sind. Die Abkürzung Bus steht dabei für Binary Unit System.

Bustopologie: Bei einer Bustopologie (Linien- oder Strangtopologie) sind alle Geräte direkt mit demselben Übertragungsmedium, dem Bus, verbunden (siehe Bustopologie, Abschnitt 2.4.1). Es gibt keine aktiven Komponenten zwischen den Geräten und dem Medium. Das Übertragungsmedium ist bei Systemen mit einer kleineren physikalischen Ausdehnung oft direkt auf einer Leiterplatte realisiert, ansonsten als Kabel oder Kabelbündel.

Nach dieser Definition sind LIN, CAN und FlexRay Bussysteme, Automotive Ethernet hingegen nicht. Trotzdem wird Automotive Ethernet oft zu den Bussystemen hinzugezählt (siehe [Zimmermann]).

2.2.4 Notwendigkeit einer Vernetzung

Bis in die 80er-Jahre gab es nur sehr wenige Steuergeräte, auch als Electronic Control Unit (ECU) bezeichnet, in den Fahrzeugen. Um Informationen miteinander auszutauschen, gab es separate Leitungen zwischen den Steuergeräten (Bild 2.7a). Wurden mehrere Signale miteinander ausgetauscht, gab es mitunter auch mehrere Leitungen.

Bei einer steigenden Anzahl von Steuergeräten und auszutauschenden Signalen wuchs der Aufwand sehr schnell. Auch war es sehr unhandlich, für gleiche Signale mehrere Leitungen verwenden zu müssen. Änderungen und Erweiterungen erforderten immer Änderungen an der Hardware und der Verdrahtung.

Mit der Einführung von Bussystemen, an die jedes Steuergerät angeschlossen wird, wurde die Kommunikation viel flexibler. Jeder Teilnehmer kann von jedem anderen Teilnehmer Nachrichten empfangen, ohne dass neue Leitungen notwendig sind. Ein solches System kann sehr leicht um weitere Nachrichten erweitert werden, ohne dass etwas an der Verdrahtung geändert werden muss (Bild 2.7b).

Die Automobilhersteller begannen mit proprietären Systemen, d. h., jeder Hersteller hatte sein eigenes Bussystem. Später setzten sich standardisierte Bussysteme durch, die von mehreren Automobilherstellern genutzt wurden und deshalb von den Halbleiterherstellern in ihre Produkte integriert wurden. Dies führte zu einer Kostensenkung und zu weiterer Verbreitung.

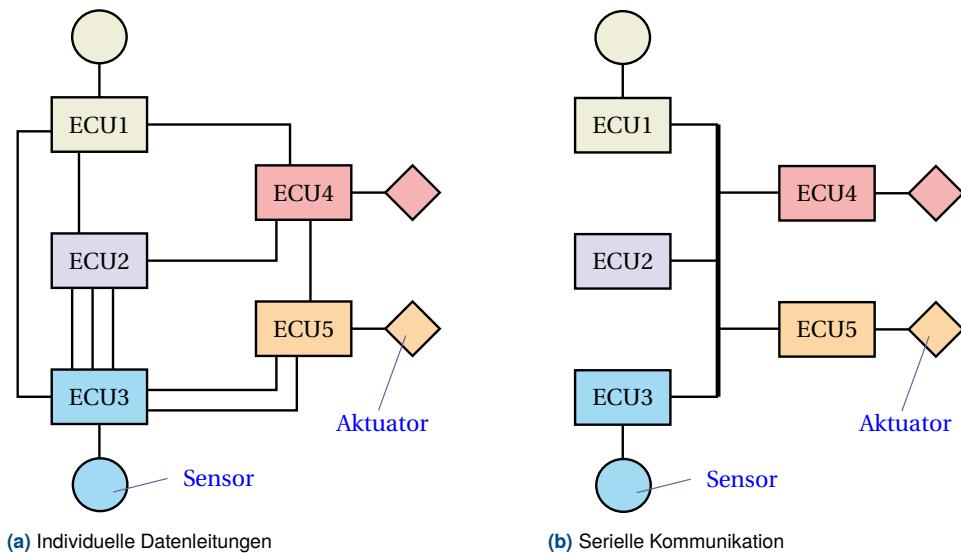


Bild 2.7 Vernetzung von Steuergeräten

Index

- Acknowledge Delimiter [93](#), [110](#)
- Acknowledge Slot [110](#)
- Acknowledge-Bit [93](#)
- Acknowledge-Feld [93](#), [232](#)
- Acknowledgement [109](#)
- Action Point [137](#), [265](#), [276](#)
- Adressierung [9](#), [306](#)
- Akzeptanzfilter [89](#)
- Akzeptanzfilterung [97](#)
- Application Layer [16](#)
- Arbitrierung [94](#)
- Arbitrierungsbitrate [214](#), [227](#)
-
- Baudrate [13](#), [42](#)
- big-endian [29](#)
- Bit Sampling [68](#)
- Bit Stuffing [33](#), [34](#), [100](#), [218](#), [232](#)
- Bitmonitoring [103](#)
- Bitrate [13](#), [41](#)
- Bitsynchronisation [32](#), [100](#), [203](#)
- Bit-Timing [201](#)
- Broadcast [10](#), [73](#), [89](#), [90](#), [306](#)
- Bus Guardian [259](#)
- Buspegel
 - CAN [81](#)
 - FlexRay [126](#)
 - LIN [65](#)
- Bustreiber → Transceiver
- Byte Start Sequence [140](#)
-
- CAN in Automation [74](#), [75](#), [79](#)
- CANopen [75](#), [202](#), [220](#)
- CAN-Takt [199](#), [201](#)
- Carrier Sense Multiple Access [25](#), [73](#), [303](#)
- CiA → CAN in Automation
- Cluster [8](#), [15](#)
- Coldstarter [255](#)
- Collision Avoidance Symbol [255](#)
- Common Mode Choke → Gleichtaktdrossel
-
- Control Area Network [73](#)
- Controller Host Interface [262](#)
- CRC → Cyclic Redundancy Check
- CRC-Feld [92](#), [231](#)
- CSMA → Carrier Sense Multiple Access
- Cycle Count [139](#)
- Cyclic Redundancy Check [157](#), [162](#), [230](#)
-
- Daisy Chain [18](#)
- Data Frame [91](#)
- Data Length Code [92](#), [215](#), [229](#)
- Data Link Layer [77](#), [123](#)
 - Automotive Ethernet [303](#)
 - CAN [88](#)
 - FlexRay [132](#)
- Data-Feld [55](#)
- Datenfeld [92](#), [231](#)
- Datenrate [13](#), [41](#)
- Diagnostic Frame [61](#)
- Dual-Simplex [11](#)
- Dynamic Trailing Sequence [276](#)
- dynamischer Slot [276](#)
- dynamisches Segment [132](#), [135](#), [136](#)
-
- ECU → Electronic Control Unit
- Ein-Draht-Bus [30](#)
- Electronic Control Unit [12](#)
- Empfangsfehlerzähler [107](#), [109](#)
- EOF [93](#), [232](#)
- Error Active [107](#), [108](#)
- Error Delimiter [104](#)
- Error Flag [104](#)
- Error Frame [91](#), [103](#), [104](#), [108](#)
- Error Passive [107](#), [108](#)
- Error State Indicator [215](#)
- Ethernet-Controller [282](#), [303](#)
- Ethernet-Frame [303](#)
- Ethertype [304](#)

Event Triggered Frame 59
Extended-Frame 207

Fehlerflag → Error Flag
Fehlersignalisierung 104
Fehlerzähler 107
Flankensteilheit 272
Frame Check Sequence 305
Frame Header 53
Frame Response 53, 55
Frame-ID 138
Frame-Timing 69

Gateway 16
Generatorpolynom 162
Gleichakttdrossel 83, 287

Halbduplex 11
Hamming-Distanz 93, 139, 164
Header CRC 139, 165
High-speed CAN 80, 81

IDE-Bit 94, 207
Identifier 55, 92
Interframe Space 93
IP-Adresse 304, 309

Jitter 43, 70, 181

Kommunikationszyklus → Zyklus
Kopplungselement 309
Kreuzparitätsprüfung 160

Leitungsbelag 173
Leitungscode 13, 35, 295
Leitungsparameter 172
Leitungsreflexion 171
LIN Description File 180
little-endian 29
low-power Mode 207, 247, 249
Low-speed CAN 80, 81

MAC-Adresse 304
Macrotick 137

Manchester-Code 37
Master 13
Master-Slave-Prinzip 23, 57
Media Access Control 303
Media Dependent Interface 284
Media Independent Interface 287
Microtick 137
Minislot 135, 136, 137, 276
Minislotlänge 276
Multicast 9, 307

Network Idle Time 132, 146
NIT → Network Idle Time
Node Capability File 183
Normalmodus 209
NRZ-Codierung 35
Null-Frame 138, 261

Offsetkorrektur 268
Offset-Korrekturwerte 265, 266, 268
OPEN Alliance 302, 311
OSI-Modell 16, 51, 77, 123, 303
Oszillatortoleranz 202
Oszilloskop 57, 81, 241, 263, 298
Overload Frame 91, 106, 107

Paritätsprüfung 157
Partial Networking 212, 226, 240
Peer-to-Peer 13
Phase Buffer Segment 201, 203
Phasenfehler 203
Physical Layer 17
– Automotive Ethernet 291
– CAN 77, 80
– FlexRay 123, 125
– LIN 51, 64
Point-to-Point 283, → Punkt-zu-Punkt
Propagation Segment 201
Protocol Engine 262
Protokollcontroller 14
Protokollzustandsmaschine 247
Punkt-zu-Punkt 13
PWM-Codierung 234

Raten-Korrekturwerte 265, 267, 268
Reflexionsfaktor 172

Remote Frame 91, 93, 97
 Remote-Transmission-Request-Bit 92
 Repeater 16
 Reserved Bit 138
 Reserved Frame 62
 Resynchronisationssprungweite 201, 203
 Ringing 176
 Ringtopologie 19
 RZ-Codierung 36

 SAE 52, 79
 Sendefehlerzähler 107, 109
 Signallaufzeit 175, 201
 Simplex 10
 Single Wire → Ein-Draht-Leitung
 Slave 13
 Sleep Mode 62
 Slotlänge 69, 148
 SOF-Bit 91, 215
 Sporadic Frame 61
 Stand-alone Controller 14
 Startup 249, 255, 299
 statischer Slot 133, 148
 statisches Segment 132, 133, 146
 Sternkoppler 16, 19, 20, 127, 129
 Steuerfeld 93, 228
 Stichleitung 17, 18, 176, 224
 Stuff Bit 100, 101, 103, 112, 216, 218, 229, 232
 Symbol 13, 27
 Symbol Window 132, 137, 146
 Synchronisationsframe 266
 Synchronisationssegment 201

 TCP 309
 Teilnetzbetrieb → Partial Networking

Terminierung 85, 130, 286
 Time Quanta 201, 222
 Time Triggered CAN → TTCAN
 Topologie 11, 17, 126, 283
 Transceiver
 – CAN 76, 199, 209, 221–226, 233–237, 239
 – Ethernet 284
 – FlexRay 121
 – LIN 64, 65
 Transmission Start Sequence 140, 263
 TTCAN 205
 Twisted-Pair-Leitung 31, 125

 Uhrensynchronisation 141, 145, 265
 Unconditional Frame 58
 Unicast 9, 306

 Virtual Local Area Network 305
 Vollduplex 11

 Wakeup 248, 249
 – selektiv 212, 239, 240
 Wakeup-Frame 212
 Wakeup-Pattern 209, 250
 Wakeup-Symbol 250
 Wellenwiderstand 85, 130, 172, 173

 Zwei-Draht-Leitung 30, 80, 171, 281
 Zyklische Redundanzprüfung → Cyclic
 Redundancy Check
 Zyklus 57, 132, 146
 Zyklusnummer 139
 Zykluszeit 146, 184