
ATZ/MTZ-Fachbuch

Die komplexe Technik heutiger Kraftfahrzeuge und Motoren macht einen immer größer werdenden Fundus an Informationen notwendig, um die Funktion und die Arbeitsweise von Komponenten oder Systemen zu verstehen. Den raschen und sicheren Zugriff auf diese Informationen bietet die regelmäßig aktualisierte Reihe ATZ/MTZ-Fachbuch, welche die zum Verständnis erforderlichen Grundlagen, Daten und Erklärungen anschaulich, systematisch und anwendungsorientiert zusammenstellt.

Die Reihe wendet sich an Fahrzeug- und Motoreningenieure sowie Studierende, die Nachschlagebedarf haben und im Zusammenhang Fragestellungen ihres Arbeitsfeldes verstehen müssen und an Professoren und Dozenten an Universitäten und Hochschulen mit Schwerpunkt Kraftfahrzeug- und Motorentechnik. Sie liefert gleichzeitig das theoretische Rüstzeug für das Verständnis wie auch die Anwendungen, wie sie für Gutachter, Forscher und Entwicklungsingenieure in der Automobil- und Zulieferindustrie sowie bei Dienstleistern benötigt werden.

Werner Zimmermann · Ralf Schmidgall

Bussysteme in der Fahrzeugtechnik

Protokolle, Standards und
Softwarearchitektur

5., aktualisierte und erweiterte Auflage

Mit 278 Abbildungen und 103 Tabellen



Springer Vieweg

Prof. Dr.-Ing. Werner Zimmermann
Fakultät Informationstechnik
Hochschule Esslingen
Esslingen, Deutschland

Dr. Ralf Schmidgall
Plochingen, Deutschland

ISBN 978-3-658-02418-5
DOI 10.1007/978-3-658-02419-2

ISBN 978-3-658-02419-2 (eBook)

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

Springer Vieweg

© Springer Fachmedien Wiesbaden 2007, 2008, 2011, 2014

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

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer Vieweg ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media
www.springer-vieweg.de

Vorwort zur 5. Auflage

„Das Schöne an Standards ist, dass es so viele davon gibt.“ Dieses Zitat stammt aus Andrew Tanenbaums Buch über Computernetze, das erstmals 1981 veröffentlicht wurde. Es gilt uneingeschränkt auch für die heutige Kraftfahrzeugelektronik.

Als die erste Auflage des vorliegenden Buches 2006 erschien, hatte die Welt der automobilen Kommunikation eine stürmische Entwicklung hinter sich. Innerhalb weniger Jahre waren die seit den 1990er Jahren erfolgreich eingesetzten CAN- und K-Line Schnittstellen durch die neuen Konzepte LIN, FlexRay und MOST ergänzt worden. Bei den Übertragungsprotokollen hatten sich KWP 2000 und OBD für die Diagnose im PKW und SAE J1939 bei Nutzfahrzeugen etabliert. Im Softwarebereich war OSEK/VDX als Betriebssystem in den Steuergeräten eingeführt und die Flash-Programmierung, mit der die Steuergerätesoftware wie ein Ersatzteil im Laufe der Lebensdauer eines Fahrzeugs ausgetauscht werden kann, wurde Stand der Technik. Weil die Entwicklung sehr rasch verlaufen war, schien es damals an der Zeit, die neuen Konzepte in einem Buch im Zusammenhang darzustellen.

Wer damals allerdings geglaubt hatte, dass die Welt der automobilen Kommunikation konsolidieren würde, sah sich getäuscht. Auch wenn sich die Bussysteme seither eher evolutionär weiterentwickelt haben, so hat sich der Softwarebereich umso stürmischer bewegt. Nicht nur die Kommunikation, sondern die gesamte Architektur der Steuergeräte zu ordnen, ist Ziel von AUTOSAR, das die gesamte Industrie massiv umtreibt. Und parallel dazu wird in der ASAM-Initiative versucht, den Aufwand für die immer umfangreichere Fahrzeugdiagnose nicht explodieren zu lassen. Entsprechend nahm in den folgenden Auflagen dieses Buches der Softwareanteil immer weiter zu. Gekürzt konnte dagegen kaum werden, da die Automobilindustrie bei aller Innovation die schöne Angewohnheit hat, neue Lösungen zügig einzusetzen, bewährte Konzepte aber keineswegs aufs Altenteil zu schicken, sondern weiter zu pflegen.

Deutet sich heute eine Konsolidierung an oder geht die stürmische Entwicklung weiter? Letzteres. Bei den Bussystemen sind mit *CAN Flexible Data-Rate* (CAN FD) und *Automotive Ethernet/IP* zwei neue Spieler aufgetaucht, die für Furore sorgen könnten. CAN FD schickt sich an, FlexRay bei höheren Datenraten nicht kampfflos das Feld zu überlassen und Ethernet könnte, vom High-End her kommend, zunächst MOST verdrängen und später vielleicht sogar die Echtzeitdomäne von FlexRay erobern.

Beide Neuentwicklungen werden in dieser Auflage genauso ausführlich dargestellt wie die Neuerungen bei *AUTOSAR 4.x*, das jetzt ein ganzes Kapitel einnimmt, und *Open Test Sequence Exchange OTX*, das die Test-Entwicklung weiter vereinfachen soll. Wie in den vergangenen Auflagen wurden wieder die Änderungen bei den bekannten Systemen MOST 150, FlexRay 3.0 und den Sensor-Aktor-Bussen wie PSI5 sowie die seit der letzten Auflage erschienen ISO-Versionen der ASAM-Standards, World-Wide Harmonized OBD und die Fortschritte bei der Car-to-Car-Kommunikation eingearbeitet. Die Zeitanalyse wurde auf die Transportprotokolle ausgeweitet.

Unverändert geblieben ist unser Grundansatz: Das Buch soll einen Überblick aus Sicht des Anwenders geben, der als Entwickler von Fahrzeugen und Steuergeräten solche Bussysteme einsetzen oder Automobilsoftware entwickeln will. Dabei stehen die Konzepte und Zusammenhänge im Vordergrund, die sich aus den Standard-Dokumenten allein nur selten direkt erschließen. Wer Protokolle oder Software selbst implementieren muss, kommt am Ende nicht umhin, die eigentlichen Standard-Dokumente im Detail zu studieren, sollte mit diesem Buch aber einen deutlich leichteren Einstieg finden. Das Buch konzentriert sich auf Systemaspekte, Hardwarethemen und EMV wurden weitgehend ausgeklammert.

Im Text werden die Normen und Standardschriften naturgemäß häufig zitiert. Aufgrund der Vielzahl wurde darauf verzichtet, an all diesen Stellen explizit Quellen anzugeben. Vielmehr finden sich am Ende jedes Kapitels Tabellen mit den jeweils einschlägigen Literaturstellen. Hersteller- und Produktnamen werden ohne ausdrückliche Erwähnung von eingetragenen Markennamen und Markenrechten verwendet.

Ein steter Diskussionspunkt bei deutschsprachiger Technikkultur ist die Verwendung englischsprachiger Begriffe. Da die Originaldokumente praktisch ausschließlich in Englisch verfügbar sind, haben wir bewusst darauf verzichtet, die normspezifischen Fachbegriffe ins Deutsche zu übersetzen. In der Regel wird bei der ersten Erwähnung die deutsche Übersetzung angegeben. Anschließend wird dann der Originalbegriff verwendet, um den *Wiedererkennungswert* beim Lesen der englischsprachigen Originale im Anschluss an dieses Buch zu erhöhen. Aufmerksamen Lesern wird auffallen, dass die Begriffe oft von den in der IT-Welt etablierten Bezeichnungen abweichen. Die Standards der Automobilelektronik sind über viele Jahre gewachsen, wurden häufig von Ingenieuren aus unterschiedlichsten Fachgebieten ohne Rücksicht auf andere Normen erstellt. Während ein IT-Ingenieur kein Verständnisproblem hat, wundert sich der Fahrzeugingenieur möglicherweise, wenn Steuergeräte als Server und Diagnosetester als Client bezeichnet werden und die darauf laufende Software Instanzen von Objekten generiert, die Diagnoseservices aufrufen. Um Zusammenhänge darzustellen, haben wir uns um Durchgängigkeit bemüht, kommen aber letztlich nicht umhin, die Originalbegriffe zu verwenden, auch wenn diese den akademischen Ansprüchen eindeutiger, einheitlicher Bezeichnungen nicht immer genügen.

Dieses Buch geht auf eine Anregung von *Wolfgang Schmid* zurück, ohne dessen beharrliches Drängen wir diese herausfordernde Aufgabe kaum angegangen wären.

Unser besonderer Dank gilt allen Mitautoren an dieser und an früheren Auflagen, *Reinhard Dapper* und *Ewald Schmitt* vom Springer-Vieweg Verlag und ihren Mitarbeitern sowie allen ungenannten Helfern, die zu diesem Buch beigetragen haben.

Den Geschäftsführern *Joachim Tauscher* (Smart In Ovation), *Wolfgang Neu* (Smart Test-solutions), *Ewald Hartmann* (Samtec Automotive Software & Electronics), *Dieter Schaller*, *Hans-Dieter Kübler* (vormals ebenfalls Samtec Automotive), *Thomas Riegraf* (Vector Informatik), *Jörg Supke* (emotive) und *Michael Kirschner* (Bereichsleiter bei Bosch Engineering) sind wir für viele Jahre freundschaftlicher Unterstützung verbunden, in denen wir auf ihr Know-How und das ihrer Mitarbeiter zurückgreifen durften. Dank unseren Kollegen bei der Daimler AG, der Robert Bosch GmbH, dem STZ Rechnereinsatz und der Hochschule Esslingen. Und nicht zuletzt unseren Familien.

Stuttgart, im Frühjahr 2014

Werner Zimmermann, Ralf Schmidgall

Autoren der 5. Auflage

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

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

Unter Mitwirkung von:

Oliver Falkner, Produktmanager CANoe/CANalyzer, *Matthias Wernicke*, Produktmanager DaVinci, und *Andreas Patzer*, Produktmanagement Measurement and Calibration bei der Vector Informatik GmbH, verfassten die Abschn. 9.1, 9.2 und 9.3.

Ewald Hartmann und *René Brzezinski*, Geschäftsführer der Samtec Automotive Software & Electronics GmbH, sowie *Sascha Riexinger*, Teamleiter Softwareentwicklung, und ihre Mitarbeiter haben den Abschn. 9.5 gestaltet.

Steffen Lang, vormals Entwicklungsingenieur bei der Bose Automotive GmbH, hat wesentlichen Anteil an Abschn. 3.4.

Dr. Kai Richter und *Dr. Marek Jersak*, Geschäftsführer der Symtavision GmbH, sind wir für den Abschn. 9.8 dankbar.

Dr. Jörg Supke, Geschäftsführer der emotive GmbH, zeichnete für Abschn. 9.7 verantwortlich und lieferte einen Beitrag zum Abschn. 6.9.

Joachim Tauscher, Geschäftsführer der Smart In Ovation GmbH, und *Nicole Peters* trugen Abschn. 9.4 zu diesem Buch bei.

Hinweis Sicher enthält auch dieses Buch kleinere und größere Fehler und Ungenauigkeiten. Angesichts der Vielzahl an Protokollen, Spezifikationen und Normen, von denen in der Regel mehrere Varianten, Vorabversionen und Revisionen existieren, ist dies auch bei größter Sorgfalt leider nicht auszuschließen. Für Hinweise und Korrekturvorschläge an <http://www.hs-esslingen.de/~zimmerma/automotive> sind wir dankbar.

Inhaltsverzeichnis

1	Anwendung von Bussystemen und Protokollen	1
1.1	Überblick	2
1.2	Kfz-Bussysteme, Protokolle und Standards	5
1.3	Standardisierung bei Kfz-Bussystemen und Software	7
1.4	Neuere Entwicklungen	9
	Literatur	12
2	Grundkonzepte und einfache Kfz-Bussysteme	13
2.1	Grundlagen	13
2.1.1	Elektrotechnische Grundlagen	13
2.1.2	Topologie und Kopplung von Bussystemen	17
2.1.3	Botschaften, Protokollstapel, Dienste (Services)	18
2.1.4	Kommunikationsmodelle, Adressierung	20
2.1.5	Zeichen- und Bitstrom-basierte Übertragung, Nutzdatenrate	25
2.1.6	Buszugriffsverfahren, Fehlererkennung und Fehlerkorrektur	27
2.1.7	Jitter und Latenz bei der Datenübertragung	29
2.1.8	Elektrik/Elektronik-(E/E)-Architekturen	30
2.2	K-Line nach ISO 9141 und ISO 14230	32
2.2.1	Entwicklung von K-Line und KWP 2000	33
2.2.2	K-Line Bus-Topologie und Physical Layer	34
2.2.3	Data Link Layer	36
2.2.4	Einschränkungen für emissionsrelevante Komponenten (OBD)	40
2.2.5	Schnittstelle zwischen Software und Kommunikations-Controller	40
2.2.6	Ältere K-Line-Varianten	41
2.2.7	Zusammenfassung K-Line – Layer 1 und 2	41
2.3	SAE J1850	42
2.4	Sensor-Aktor-Bussysteme	45
2.4.1	SENT – Single Edge Nibble Transmission nach SAE J2716	45
2.4.2	PSI 5 – Peripheral Sensor Interface 5	46
2.4.3	ASRB 2.0 – Automotive Safety Restraint Bus (ISO 22898)	49
2.4.4	DSI – Distributed Systems Interface	51

2.5	Normen und Standards zu Kap. 2	54
	Literatur	55
3	Kfz-Bussysteme – Physical und Data Link Layer	57
3.1	Controller Area Network CAN nach ISO 11898	57
3.1.1	Entwicklung von CAN	57
3.1.2	Bus-Topologie und Physical Layer	58
3.1.3	CAN Data Link Layer	61
3.1.4	Fehlerbehandlung	63
3.1.5	Einsatz von CAN – Höhere Protokolle	63
3.1.6	Schnittstelle zwischen Protokoll-Software und CAN-Controller	64
3.1.7	Zeitverhalten von CAN-Systemen, Wahl der Botschaftspriorität	67
3.1.8	Time-Triggered-CAN (TTCAN) – Deterministischer Buszugriff	72
3.1.9	Energiesparmaßnahmen: Wakeup und Partial Networking	75
3.1.10	Höhere Datenraten: CAN Flexible Data-Rate CAN FD	76
3.1.11	Zusammenfassung CAN – Layer 1 und 2	78
3.2	Local Interconnect Network LIN	79
3.2.1	Überblick	79
3.2.2	Data Link Layer	81
3.2.3	Neue Botschaftstypen bei LIN V2.0	84
3.2.4	LIN Transportschicht und ISO Diagnose über LIN	85
3.2.5	LIN Configuration Language	86
3.2.6	Dynamische Konfiguration von LIN-Slave-Steuergeräten	90
3.2.7	LIN Application Programming Interface (API)	92
3.2.8	Zeitverhalten von LIN-Systemen	94
3.2.9	Zusammenfassung LIN – Layer 1 und 2	95
3.3	FlexRay	96
3.3.1	Bus-Topologie und Physical Layer	97
3.3.2	Data Link Layer	99
3.3.3	Netzwerk-Start und Takt-Synchronisation	102
3.3.4	Fehlerbehandlung, Bus Guardian	105
3.3.5	Konfiguration und übergeordnete Protokolle	106
3.3.6	Zeitverhalten von FlexRay-Systemen, Beispiel-Konfiguration	107
3.3.7	Schnittstelle zum FlexRay-Controller	112
3.3.8	Weiterentwicklung FlexRay 3.x	116
3.3.9	Zusammenfassung FlexRay – Layer 1 und 2	118
3.4	Media Oriented Systems Transport MOST	119
3.4.1	Bus-Topologie und Physical Layer	121
3.4.2	Data Link Layer	121
3.4.3	Kommunikationscontroller	128
3.4.4	Network Services und Funktionsblöcke	129
3.4.5	Netzmanagement	133

3.4.6	Höhere Protokollschichten	135
3.4.7	Beispiel für Systemstart und Audioverbindung	135
3.4.8	Zusammenfassung MOST	138
3.5	Automotive Ethernet	138
3.5.1	Ethernet nach IEEE 802.3	138
3.5.2	Autotauglicher Physical Layer BroadR-Reach	141
3.5.3	Echtzeitfähigkeit mit IEEE 802.1 Audio-Video-Bridging AVB	142
3.5.4	Höhere Protokollschichten IP, TCP und UDP	145
3.6	Normen und Standards zu Kapitel 3	149
	Literatur	150
4	Transportprotokolle	153
4.1	Transportprotokoll ISO TP für CAN nach ISO 15765-2	154
4.1.1	Botschaftsaufbau	154
4.1.2	Flusssteuerung, Zeitüberwachung und Fehlerbehandlung	156
4.1.3	Dienste für die Anwendungsschicht (Application Layer Services)	157
4.1.4	Protokoll-Erweiterungen	159
4.1.5	Adressierung bei KWP 2000/UDS – Zuordnung von CAN Identifiern	159
4.1.6	Bandbreite des ISO TP für CAN	159
4.2	Transportprotokoll für FlexRay nach ISO 10681-2	162
4.2.1	Botschaftsaufbau und Adressierung	162
4.2.2	Verbindungsarten und Übertragungsablauf	163
4.2.3	Bandbreitensteuerung	165
4.2.4	Fehlerbehandlung und Implementierungshinweise	166
4.2.5	Bandbreite des FlexRay Transportprotokolls	166
4.3	Transportprotokoll TP 2.0 für CAN	169
4.3.1	Adressierung und CAN Message Identifier	170
4.3.2	Broadcast-Botschaften	170
4.3.3	Dynamischer Kanalaufbau und Verbindungsmanagement	171
4.3.4	Datenübertragung	174
4.4	Transportprotokoll TP 1.6 für CAN	175
4.4.1	Botschaftsaufbau	176
4.4.2	Dynamischer Kanalaufbau	176
4.4.3	Datenübertragung und Datenrichtungswechsel	177
4.5	Transportprotokoll SAE J1939/21 für CAN	178
4.5.1	Übertragungsarten, Adressierung und CAN Message Identifier	179
4.5.2	Segmentierte Datenübertragung (Multi Packet)	182
4.6	Transportprotokoll DoIP nach ISO 13400	183
4.7	Transportprotokoll für CAN FD	187
4.8	Normen und Standards zu Kapitel 4	187
	Literatur	188

5	Diagnoseprotokolle – Application Layer	189
5.1	Diagnoseprotokoll KWP 2000 (ISO 14230-3)	192
5.1.1	Überblick	192
5.1.2	Diagnosesitzungen (Diagnostic Management)	194
5.1.3	Adressierung der Steuergeräte nach KWP 2000 und UDS	197
5.1.4	Bussystem-abhängige Dienste (Network Layer Protocol Control)	199
5.1.5	Fehlerspeicher lesen und löschen (Stored Data Transmission)	200
5.1.6	Daten lesen und schreiben (Data Transmission), Ansteuern von Steuergeräte-Ein- und Ausgängen (Input/Output Control)	200
5.1.7	Speicherblöcke auslesen und speichern (Upload, Download)	202
5.1.8	Start von Programmen im Steuergerät (Remote Routine Activation)	202
5.1.9	Erweiterte Dienste (Extended Services)	203
5.2	Unified Diagnostic Services UDS nach ISO 14229/15765-3	203
5.2.1	Unterschiede zum KWP 2000 Diagnoseprotokoll	204
5.2.2	Überblick über die UDS-Diagnosedienste	204
5.2.3	Response on Event Dienst	208
5.3	On-Board-Diagnose OBD nach ISO 15031/SAE J1979	212
5.3.1	Überblick OBD-Diagnosedienste	213
5.3.2	Auslesen des Fehlerspeichers und von Steuergerätewerten	214
5.3.3	Abfrage der Testergebnisse für abgasrelevante Komponenten	217
5.3.4	OBD-Fehlercodes	218
5.3.5	Data Link Security	221
5.3.6	Pass-Through-Programmierung	221
5.3.7	Beispiel	222
5.4	Weiterentwicklung der Diagnose	224
5.4.1	World-Wide Harmonized On-Board Diagnose nach ISO 27145	225
5.5	Normen und Standards zu Kapitel 5	227
	Literatur	228
6	Anwendungen für Messen, Kalibrieren und Diagnose (ASAM AE MCD)	229
6.1	Einführung	229
6.2	Busprotokolle für Aufgaben in der Applikation (ASAM AE MCD 1MC)	232
6.2.1	CAN Calibration Protocol CCP	234
6.2.2	Extended Calibration Protocol XCP	241
6.2.3	AML-Konfigurationsdateien für XCP und CCP	254
6.2.4	Interface zwischen Bus und Applikationssystem ASAM MCD 1b	256
6.3	Field Bus Exchange Format FIBEX	261
6.4	Überblick über ASAM AE MCD 2 und MCD 3	270
6.5	Applikationsdatensätze nach ASAM MCD 2 MC	272
6.5.1	ASAP2/A2L-Applikationsdatensätze	272
6.5.2	Calibration Data Format CDF und Meta Data Exchange MDX	275
6.6	ODX-Diagnosedatensätze nach ASAM AE MCD 2D	277

6.6.1	Aufbau des ODX-Datenmodells	278
6.6.2	DIAG-LAYER: Hierarchische Diagnosebeschreibung	280
6.6.3	VEHICLE-INFO-SPEC: Fahrzeugzugang und Bustopologie	283
6.6.4	COMPARAM-SPEC und COMPARAM-SUBSET: Busprotokoll	286
6.6.5	DIAG-COMM und DIAG-SERVICE: Diagnosedienste	288
6.6.6	Einfache und komplexe Datenobjekte	292
6.6.7	SINGLE-ECU-JOB und MULTIPLE-ECU-JOB: Diagnoseabläufe	302
6.6.8	STATE-CHART: Diagnosesitzungen	304
6.6.9	ECU-CONFIG: Beschreibung der Steuergeräte-Konfiguration	305
6.6.10	ECU-MEM: Beschreibung der Flash-Programmierung	306
6.6.11	FUNCTION-DICTIONARY: Funktionsorientierte Diagnose	308
6.6.12	Packaged ODX und ODX-Autorenwerkzeuge	309
6.6.13	ODX Version 2.2 und ISO 22901	309
6.7	ASAM AE MCD 3-Server	310
6.7.1	Funktionsgruppe M – Messen	312
6.7.2	Funktionsgruppe C – Kalibrieren	313
6.7.3	Funktionsgruppe D – Diagnose	314
6.8	MVCI-Schnittstelle für Diagnosetester nach ISO 22900	316
6.9	OTX-Beschreibung von Testabläufen nach ISO 13209	319
6.9.1	Grundkonzepte und Aufbau von OTX	319
6.9.2	OTX-Core-Datenmodell	321
6.9.3	OTX-Core-Programmelemente	322
6.9.4	OTX-Erweiterungen	323
6.10	Normen und Standards zu Kapitel 6	328
	Literatur	329
7	Software-Standards: OSEK und HIS	331
7.1	Einführung	331
7.2	OSEK/VDX	334
7.2.1	Ereignisgesteuerter Betriebssystemkern OSEK/VDX OS	336
7.2.2	Kommunikation in OSEK/VDX COM	346
7.2.3	Netzmanagement mit OSEK/VDX NM	350
7.2.4	Zeitgesteuerter Betriebssystemkern OSEK Time, Fehlertoleranz OSEK FTCOM und Schutzmechanismen Protected OSEK	355
7.2.5	Scheduling, Taskprioritäten und Zeitverhalten bei OSEK OS und AUTOSAR OS	358
7.3	Hardware-Ein- und Ausgabe (HIS IO Library, IO Driver)	361
7.4	HIS Hardwaretreiber für CAN-Kommunikationscontroller (HIS CAN Driver)	363
7.5	HIS Flash-Lader	363
7.6	Normen und Standards zu Kapitel 7	364
	Literatur	365

8	AUTOSAR-Softwarearchitektur für Kfz-Systeme	367
8.1	Einführung	367
8.2	Überblick über die AUTOSAR-Basissoftware	369
8.2.1	Funktionale Sicherheit	380
8.3	Betriebssystem AUTOSAR OS	381
8.4	Kommunikationsstack AUTOSAR COM, Diagnose DCM	385
8.5	Netzmanagement AUTOSAR NM	397
8.6	Virtual Function Bus VFB, Runtime Environment RTE und Softwarekomponenten	403
8.7	Beispiel einer einfachen Anwendung	408
8.8	Ausblick	410
8.9	Normen und Standards zu Kapitel 8	412
	Literatur	413
9	Werkzeuge, Anwendungen und Einsatzgebiete	415
9.1	Entwurf und Test der On-Board-Kommunikation	415
9.1.1	Entwicklungsprozess mit <i>CANoe</i> von Vector Informatik	415
9.1.2	Netzwerkdesign mit dem Network Designer	416
9.1.3	Simulation des Gesamtsystems in <i>CANoe</i>	420
9.1.4	Restbussimulation als Entwicklungsumgebung für Steuergeräte	422
9.1.5	Integration des Gesamtsystems	424
9.2	System- und Softwareentwurf für Steuergeräte	424
9.2.1	Systementwurf mit <i>PREEvision</i> von Vector Informatik	426
9.2.2	Entwicklung der Anwendungssoftware im AUTOSAR-Prozess	427
9.2.3	Systemtest und Applikation	428
9.3	Werkzeuge zur Applikation von Steuergeräten	429
9.3.1	Steuergeräte-Applikation mit <i>CANape</i> von Vector Informatik	430
9.4	Flash-Programmierung von Steuergeräten	433
9.4.1	Rahmenbedingungen	434
9.4.2	Flash-Speicher	437
9.4.3	Flash-Programmierprozess	440
9.4.4	Beispiel eines Flash-Laders: ADLATUS von SMART IN OVATION	448
9.4.5	Softwaretest von Flash-Ladern und Busprotokollen	453
9.5	Diagnosewerkzeuge in Entwicklung und Fertigung	457
9.5.1	Beispiel für Diagnosewerkzeuge: samDia von Samtec Automotive	459
9.6	Autorenwerkzeuge für Diagnosedaten	469
9.7	Diagnose-Laufzeitsysteme und OTX Diagnose-Sequenzen	471
9.7.1	Open Test Framework von emotive als OTX Werkzeug	473
9.8	Echtzeitverhalten der Steuergeräte-Kommunikation	478
9.8.1	Kennwerte für das Echtzeitverhalten	478
9.8.2	Echtzeitanalyse mit SymTA/S von Syntavision	480
	Literatur	482

10	Kommunikation zwischen Fahrzeugen	483
10.1	Mautsysteme	483
10.2	Car2Car-Konsortium und Vehicle2X-Kommunikation	484
10.3	Normen und Standards zu Kapitel 10	488
	Literatur	489
	Web-Adressen	491
	Abkürzungen	495
	Sachverzeichnis	501