

Inhalt

Vorwort.....	V
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
2 Grundkonzepte und einfache Kfz-Bussysteme.....	9
2.1 Grundlagen.....	9
2.1.1 Elektrotechnische Grundlagen	9
2.1.2 Topologie und Kopplung von Bussystemen	12
2.1.3 Botschaften, Protokollstapel, Dienste (Services).....	14
2.1.4 Kommunikationsmodelle, Adressierung	16
2.1.5 Zeichen- und Bitstrom-basierte Übertragung, Nutzdatenrate	20
2.1.6 Buszugriffsverfahren, Fehlererkennung und Fehlerkorrektur	22
2.1.7 Jitter und Latenz bei der Datenübertragung	25
2.2 K-Line nach ISO 9141 und ISO 14230	26
2.2.1 Entwicklung von K-Line und KWP 2000.....	26
2.2.2 K-Line Bus-Topologie und Physical Layer	27
2.2.3 Data Link Layer	29
2.2.4 Einschränkungen für emissionsrelevante Komponenten (OBD)	33
2.2.5 Schnittstelle zwischen Protokoll-Software und Kommunikations- Controller	33
2.2.6 Ältere K-Line-Varianten	34
2.2.7 Zusammenfassung K-Line – Layer 1 und 2	35
2.3 SAE J1850	36
2.4 Sensor-Aktor-Bussysteme.....	39
2.4.1 SENT – Single Edge Nibble Transmission nach SAE J2716	39
2.4.2 PSI 5 – Peripheral Sensor Interface 5	40
2.4.3 ASRB 2.0 – Automotive Safety Restraint Bus (ISO 22898)	41
2.4.4 DSI – Distributed Systems Interface	43
2.5 Normen und Standards	44
3 Kfz-Bussysteme – Physical und Data Link Layer	45
3.1 Controller Area Network CAN nach ISO 11898.....	45
3.1.1 Entwicklung von CAN.....	45
3.1.2 Bus-Topologie und Physical Layer	46
3.1.3 CAN Data Link Layer.....	48
3.1.4 Fehlerbehandlung	50
3.1.5 Einsatz von CAN – Höhere Protokolle	50
3.1.6 Schnittstelle zwischen Protokoll-Software und CAN-Controller.....	52

3.1.7	Zeitverhalten von CAN-Systemen, Wahl der Botschaftspriorität	54
3.1.8	Time-Triggered-CAN (TTCAN) – Deterministischer Buszugriff	60
3.1.9	Zusammenfassung CAN – Layer 1 und 2	62
3.2	Local Interconnect Network LIN	63
3.2.1	Überblick	63
3.2.2	Data Link Layer	64
3.2.3	Neue Botschaftstypen bei LIN V2.0	67
3.2.4	LIN Transportschicht und ISO Diagnose über LIN	68
3.2.5	LIN Configuration Language	69
3.2.6	Dynamische Konfiguration von LIN-Slave-Steuergeräten	73
3.2.7	LIN Application Programming Interface (API).....	74
3.2.8	Zeitverhalten von LIN-Systemen	76
3.2.9	Zusammenfassung LIN – Layer 1 und 2	78
3.3	FlexRay	79
3.3.1	Bus-Topologie und Physical Layer.....	79
3.3.2	Data Link Layer	81
3.3.3	Netzwerk-Start und Takt-Synchronisation	85
3.3.4	Fehlerbehandlung, Bus Guardian	87
3.3.5	Konfiguration und übergeordnete Protokolle.....	88
3.3.6	Zeitverhalten von FlexRay-Systemen, Beispiel-Konfiguration.....	89
3.3.7	Schnittstelle zum FlexRay-Controller	94
3.3.8	Weiterentwicklung FlexRay 3.0	98
3.3.9	Zusammenfassung FlexRay – Layer 1 und 2.....	98
3.4	Media Oriented Systems Transport MOST	99
3.4.1	Bus-Topologie und Physical Layer.....	99
3.4.2	Data Link Layer	100
3.4.3	Kommunikationscontroller	105
3.4.4	Network Services und Funktionsblöcke	106
3.4.5	Netzmanagement	110
3.4.6	Höhere Protokollschichten	112
3.4.7	Beispiel für Systemstart und Audioverbindung.....	112
3.4.8	Neuere Entwicklungen: MOST150	114
3.4.9	Zusammenfassung MOST	116
3.5	Normen und Standards	117
4	Transportprotokolle	119
4.1	Transportprotokoll ISO TP für CAN nach ISO 15765-2	119
4.1.1	Botschaftsaufbau.....	120
4.1.2	Flusssteuerung, Zeitüberwachung und Fehlerbehandlung	121
4.1.3	Dienste für die Anwendungsschicht (Application Layer Services).....	123
4.1.4	Protokoll-Erweiterungen	124
4.1.5	Adressierung bei KWP 2000/UDS – Zuordnung von CAN Identifiern ..	124
4.2	Transportprotokoll für FlexRay nach ISO 10681-2	125
4.2.1	Botschaftsaufbau und Adressierung	125
4.2.2	Verbindungsarten und Übertragungsablauf	126
4.2.3	Bandbreitensteuerung	128
4.2.4	Fehlerbehandlung und Implementierungshinweise	128

4.3	Transportprotokoll TP 2.0 für CAN	130
4.3.1	Adressierung und CAN Message Identifier	130
4.3.2	Broadcast-Botschaften	131
4.3.3	Dynamischer Kanalaufbau und Verbindungsmanagement	132
4.3.4	Datenübertragung	134
4.4	Transportprotokoll TP 1.6 für CAN	136
4.4.1	Botschaftsaufbau	136
4.4.2	Dynamischer Kanalaufbau	136
4.4.3	Datenübertragung und Datenrichtungswechsel	137
4.5	Transportprotokoll SAE J1939/21 für CAN	138
4.5.1	Übertragungsarten, Adressierung und CAN Message Identifier	138
4.5.2	Segmentierte Datenübertragung (Multi Packet)	141
4.6	Transportprotokoll DoIP nach ISO 13400	143
4.7	Normen und Standards	147
5	Diagnoseprotokolle – Application Layer	149
5.1	Diagnoseprotokoll KWP 2000 (ISO 14230-3)	151
5.1.1	Überblick	151
5.1.2	Diagnosesitzungen (Diagnostic Management)	154
5.1.3	Adressierung der Steuergeräte nach KWP 2000 und UDS	156
5.1.4	Bussystem-abhängige Dienste (Network Layer Protocol Control)	158
5.1.5	Fehlerspeicher lesen und löschen (Stored Data Transmission)	159
5.1.6	Daten lesen und schreiben (Data Transmission), Ansteuern von Steuergeräte-Ein- und Ausgängen (Input/Output Control)	160
5.1.7	Speicherblöcke auslesen und speichern (Upload, Download)	161
5.1.8	Start von Programmen im Steuergerät (Remote Routine Activation)	162
5.1.9	Erweiterte Dienste (Extended Services)	163
5.2	Unified Diagnostic Services UDS nach ISO 14229/15765-3	163
5.2.1	Unterschiede zum KWP 2000 Diagnoseprotokoll	163
5.2.2	Überblick über die UDS-Diagnosedienste	164
5.2.3	Response on Event Dienst	171
5.3	On-Board-Diagnose OBD nach ISO 15031 / SAE J1979	173
5.3.1	Überblick OBD-Diagnosedienste	173
5.3.2	Auslesen des Fehlerspeichers und von Steuergerätewerten	175
5.3.3	Abfrage der Testergebnisse für abgasrelevante Komponenten	178
5.3.4	OBD-Fehlercodes	179
5.3.5	Data Link Security	181
5.3.6	Pass-Through-Programmierung	181
5.3.7	Beispiel	182
5.4	Weiterentwicklung der Diagnoseschnittstellen	184
5.5	Normen und Standards	185
6	Anwendungen für Messen, Kalibrieren und Diagnose (ASAM AE MCD)	187
6.1	Einführung	187
6.2	Busprotokolle für Aufgaben in der Applikation (ASAM AE MCD 1MC)	191
6.2.1	CAN Calibration Protocol CCP	192
6.2.2	Extended Calibration Protocol XCP	200

6.2.3	AML-Konfigurationsdateien für XCP und CCP	215
6.2.4	Interface zwischen Busprotokolltreiber und Applikationssystem ASAM MCD 1b	217
6.3	Field Bus Exchange Format FIBEX	221
6.4	Überblick über ASAM AE MCD 2 und MCD 3	231
6.5	Applikationsdatensätze nach ASAM MCD 2 MC	232
6.5.1	ASAP2/A2L-Applikationsdatensätze	232
6.5.2	Calibration Data Format CDF und Meta Data Exchange Format MDX236	
6.6	ODX-Diagnosedatensätze nach ASAM AE MCD 2D	238
6.6.1	Aufbau des ODX-Datenmodells	239
6.6.2	DIAG-LAYER: Hierarchische Diagnosebeschreibung	241
6.6.3	VEHICLE-INFO-SPEC: Fahrzeugzugang und Bustopologie	244
6.6.4	COMPARAM-SPEC und COMPARAM-SUBSET: Busprotokoll	247
6.6.5	DIAG-COMM und DIAG-SERVICE: Diagnosedienste	249
6.6.6	Einfache und komplexe Datenobjekte	254
6.6.7	SINGLE-ECU-JOB und MULTIPLE-ECU-JOB: Diagnoseabläufe	262
6.6.8	STATE-CHART: Diagnosesitzungen	264
6.6.9	ECU-CONFIG: Beschreibung der Steuergeräte-Konfiguration	265
6.6.10	ECU-MEM: Beschreibung der Flash-Programmierung	266
6.6.11	FUNCTION-DICTIONARY: Funktionsorientierte Diagnose	268
6.6.12	Packaged ODX und ODX-Autorenwerkzeuge	269
6.6.13	ODX Version 2.2	269
6.7	ASAM AE MCD 3-Server	270
6.7.1	Funktionsgruppe M – Messen	271
6.7.2	Funktionsgruppe C – Kalibrieren	273
6.7.3	Funktionsgruppe D – Diagnose	273
6.8	MVCI-Schnittstelle für Diagnostester nach ISO 22900	275
6.9	OTX-Beschreibung von Testabläufen nach ISO 13209	278
6.10	Normen und Standards	279
7	Software-Standards: OSEK, HIS, AUTOSAR	281
7.1	Einführung	281
7.2	OSEK/VDX	284
7.2.1	Ereignisgesteuerter Betriebssystemkern OSEK/VDX OS	285
7.2.2	Kommunikation in OSEK/VDX COM	295
7.2.3	Netzmanagement mit OSEK/VDX NM	299
7.2.4	Zeitgesteuerter Betriebssystemkern OSEK Time, Fehlertoleranz OSEK FTCOM und Schutzmechanismen Protected OSEK	304
7.2.5	Scheduling, Taskprioritäten und Zeitverhalten bei OSEK OS und AUTOSAR OS	307
7.3	Hardware-Ein- und Ausgabe (HIS IO Library, IO Driver)	310
7.4	HIS Hardwaretreiber für CAN-Kommunikationscontroller (HIS CAN Driver)	312
7.5	HIS Flash-Lader	312
7.6	AUTOSAR	313
7.6.1	Überblick über die AUTOSAR-Basissoftware	315

7.6.2	Betriebssystem AUTOSAR OS	326
7.6.3	Kommunikationsstack AUTOSAR COM und Diagnose DCM	330
7.6.4	Netzmanagement AUTOSAR NM	342
7.6.5	Virtual Function Bus VFB, Runtime Environment AUTOSAR RTE und AUTOSAR-Softwarekomponenten	346
7.6.6	Ausblick	352
7.7	Normen und Standards	353
8	Werkzeuge, Anwendungen und Einsatzgebiete	355
8.1	Softwarekomponenten für Steuergeräte	355
8.2	Entwurf und Test der On-Board-Kommunikation	355
8.2.1	Entwicklungsprozess mit <i>CANoe</i> von Vector Informatik	356
8.2.2	Netzwerkdesign mit dem <i>Network Designer</i>	356
8.2.3	Simulation des Gesamtsystems in <i>CANoe</i>	360
8.2.4	Restbussimulation als Entwicklungsumgebung für Steuergeräte	361
8.2.5	Integration des Gesamtsystems	363
8.2.6	Einfluss von AUTOSAR auf das Netzwerkdesign	363
8.2.7	Test von AUTOSAR Softwarekomponenten	365
8.3	Werkzeuge zur Applikation von Steuergeräten	365
8.3.1	Steuergeräte-Applikation mit <i>CANape</i> von Vector Informatik	367
8.4	Flash-Programmierung von Steuergeräten	370
8.4.1	Rahmenbedingungen	371
8.4.2	Flash-Speicher	374
8.4.3	Flash-Programmierprozess	376
8.4.4	Beispiel eines Flash-Laders: <i>ADLATUS</i> von SMART Electronic	384
8.4.5	Test und Freigabe von Flash-Ladern und Busprotokollen	390
8.5	Diagnosewerkzeuge in Entwicklung und Fertigung	394
8.5.1	Beispiel für Diagnosewerkzeuge: <i>samDia</i> von Samtec Automotive ...	395
8.6	Autorenwerkzeuge für Diagnosedaten	405
8.7	Diagnose-Laufzeitsysteme und OTX Diagnose-Sequenzen	407
8.7.1	Einsatz des Open Test Sequence Exchange Datenformats OTX	408
8.7.2	<i>Open Diagnostic Framework</i> von Emotive als OTX Werkzeug	410
8.8	Echtzeitverhalten von Steuergeräten mit Bussystemen	412
8.8.1	Echtzeitanalyse mit <i>SymTA/S</i> von Syntavision	413
9	Kommunikation zwischen Fahrzeugen	415
9.1	Mautsysteme	415
9.2	Car2Car-Konsortium	416
9.3	Normen und Standards	418
	Literaturverzeichnis	419
	Webseite zu diesem Buch	420
	Web-Adressen	421
	Abkürzungen	423
	Sachwortverzeichnis	429