

Inhaltsverzeichnis

1	Einleitung	1
2	Feldbus-Grundlagen	3
2.1	Grundlegende Begriffe	3
2.1.1	OSI-Schichtenmodell	3
2.1.2	Klassifizierung von Autobusprotokollen	5
2.1.3	Verkabelungstechnik	6
2.1.4	Übertragungsmedien	7
2.1.5	Topologien	9
2.1.6	Medienzugriffssteuerung	9
2.1.7	Fehlererkennung	11
2.1.8	Echtzeitfähigkeit	12
2.2	Kriterien für die Auswahl von Feldbusprotokollen	13
2.3	Die Entwicklung von CAN	14
2.3.1	Standardisierungen von CAN	16
3	Grundlagen der CAN-Architektur	17
3.1	CAN-Spezifikationen	18
3.1.1	CAN-Telegramme	18
3.1.2	Bit-Codierung	22
3.1.3	Arbitrierung	22
3.1.4	Ausnahmebehandlung	24
3.1.5	Zeitverhalten	30
3.1.6	Bit-Timing und Synchronisation	32
3.1.7	Eigenschaften von CAN-Controllern	38
3.2	Time-Triggered CAN – TTCAN	
	Die zeitgesteuerte Version des CAN-Protokolls	40
3.2.1	Motivation der Weiterentwicklung von CAN zu TTCAN	40
3.2.2	Rahmenbedingungen	41
3.2.3	Prinzipien der Zeitsteuerung	41
3.2.4	Kommunikationsstruktur	42
3.2.5	Kommunikationsablauf	44
3.2.6	Synchronisation, lokale und globale Zeit	44
3.2.7	Redundanz der Zeitgeber	46
3.2.8	Ereignissynchronisierter Zyklus	46
3.2.9	Driftkorrektur	47
3.2.10	Erweiterte Fehlererkennung und -behandlung	48
3.2.11	Zusammenfassung	48

4	Die physikalische Schicht	49
4.1	Der Transceiver	49
4.1.1	Transceiver-Modi	50
4.2	Die CAN-Drossel	52
4.3	Netzwerkkonzepte	52
4.3.1	Der fehlertolerante Low-Speed Physical Layer	53
4.3.2	Der High-Speed Physical Layer	55
4.4	Terminierungskonzepte	58
4.4.1	Standard-Terminierung	58
4.4.2	Terminierung mit Mittelabgriff	58
4.4.3	Terminierung mit Split-Pin	58
4.4.4	Terminierungswiderstände an allen Knoten	59
4.5	Netzwerktopologien	59
4.5.1	Einzelstern	60
4.5.2	Doppelstern	60
4.5.3	Linearer Bus	61
4.5.4	Hybride Topologien	61
4.5.5	Laufzeiten im Netzwerk	62
4.6	Topologieeinflüsse	62
4.6.1	Architektur des CAN-Netzwerks	63
4.6.2	Architektur von CAN-Knoten	67
4.6.3	Wechselwirkungen der Komponenten und analytische Betrachtung der Signalqualität	75
4.7	Simulationsbasierter Entwurf von Fahrzeugvernetzungstopologien	82
4.7.1	Entwicklung automotiver Vernetzungstopologien	82
4.7.2	Systemsimulation als Werkzeug des Netzwerkdesigners	84
4.7.3	Saber – Eine Entwicklungsumgebung zur Simulation und Analyse der physikalischen Schicht von Netzwerktopologien	88
4.8	Elektromagnetische Verträglichkeit – EMV	93
4.8.1	Einleitung	93
4.8.2	EMV-Anforderungen, -Vorschriften und -Richtlinien	94
4.8.3	Einflussfaktoren auf die EMV des CAN-Busses	94
4.8.4	EMV-Bewertung von CAN-Transceivern	96
4.9	Teilnetzbetrieb – Partial Networking	109
4.9.1	Motivation	109
4.9.2	Realisierungsmöglichkeiten	111
4.9.3	Transceiver-Lösung für den Teilnetzbetrieb (Infineon)	112
4.9.4	Teilnetzfähige High-Speed-CAN-Transceiver (ELMOS).....	117
4.9.5	Partial Networking mit dem Mikrocontroller (Infineon)	122
4.9.6	Vergleich der Transceiver- und Mikrocontroller-Lösung (Infineon)	123
4.9.7	Konzept und Realisierung bei NXP	123
4.10	Transceiver-Implementierungen	125
4.10.1	Implementierungsbeispiel TLE 6254-3G (ISO 11898-3)	125
4.10.2	Implementierungsbeispiel TLE 6250 G (ISO 11898-2)	129
4.10.3	Implementierungsbeispiel TLE 6251-2G (ISO 11898-5)	132

5	CAN-Protokollkerne und CAN-Module	137
5.1	M_CAN – Modularer CAN-Controller	137
5.2	IFI Advanced CAN	140
5.2.1	Sendepuffer	141
5.2.2	Masken und Filter	142
5.2.3	Empfangspuffer	142
5.2.4	Zeitstempel	143
5.2.5	Fazit	143
5.3	Renesas (D)AFCAN	144
5.3.1	Eigenschaften des AFCAN	145
5.3.2	Erweiterte Eigenschaften des DAFCAN	146
5.3.3	Funktionsbeschreibung des AFCAN	146
5.3.4	Zusätzliche Funktionen des DAFCAN	150
5.3.5	AFCAN-Module im V850/Fx3 (μ PD70F337x, μ PD70F338x)	151
5.3.6	(D)AFCAN-Module im V850/CG4 (μ PD70F3433)	152
5.4	Infineons CAN-Module der XC16x- und XC2000/XE16x-Familie	153
5.4.1	TwinCAN und MultiCAN von Infineon	153
5.4.2	TwinCAN	153
5.4.3	MultiCAN	158
5.4.4	Die XC2000-Familie in Funktion eines Body-Gateways – Ein Applikationsbeispiel für den MultiCAN	163
5.5	Microchip – Produkte für CAN-Applikationen	165
5.5.1	8-bit-PIC®-Mikrocontroller mit On-Chip-CAN-Controller	167
5.5.2	16-bit-PIC®-Mikrocontroller und dsPIC® Digital Signal Controllers (DSCs) mit On-Chip-CAN-Controller	170
5.5.3	32-bit-PIC®-Mikrocontroller-Lösungen mit On-Chip-CAN	176
5.5.4	Microchips Stand-alone-CAN-Peripherie-Produkte	177
5.6	Basic CAN+: Das Freescale MSCAN-Modul	178
5.6.1	Funktionsweise des MSCAN-Interfaces	180
5.6.2	Ablauf des Nachrichtenempfangs	182
5.6.3	Sendebetrieb der Nachrichten	183
5.7	Xilinx CAN-Controller LogiCORE™ IP	185
5.7.1	Benutzerschnittstelle	186
5.7.2	Objektschicht	188
5.7.3	Transferschicht	190
5.7.4	Bitstromprozessor	191
5.7.5	Konfiguration des CAN-Controllers	191
5.7.6	Bestellung des LogiCORE CAN-Controllers	196
6	High-Level-Protokolle	197
6.1	CANopen	197
6.1.1	Profil für die unteren Schichten	197
6.1.2	Gerätemodell	199
6.1.3	Netzwerkmanagement	202
6.1.4	Transportprotokolle	204

6.1.5	Anwendungsprotokolle	206
6.1.6	Geräteprofile	209
6.1.7	Anwendungsprofile	214
6.2	AUTOSAR	216
6.2.1	Einleitung	216
6.2.2	Die AUTOSAR-Plattform	219
6.2.3	AUTOSAR-Kommunikation – Ein Beispiel	224
6.3	Fahrzeugdiagnose-Realisierungen auf CAN	229
6.3.1	Überblick	229
6.3.2	OBDonCAN – ISO 15031 emissionsbezogene OBD auf ISO 15765-4 DoCAN	232
6.3.3	UDSonCAN – ISO 14229-3	239
6.3.4	Entwicklungstrends	243
6.4	SAE J1939	252
6.4.1	Hintergrund der SAE J1939	252
6.4.2	Struktur der SAE J1939	253
6.4.3	SAE J1939-21 – Sicherungsschicht	254
6.4.4	SAE J1939-31 – Netzwerkschicht	257
6.4.5	SAE J1939-71 – Fahrzeugapplikationsschicht	257
6.4.6	SAE J1939-73 – Diagnose	259
6.4.7	SAE J1939-81 – Netzwerkmanagement	262
6.4.8	SAE J1939-84 – Konformitätstest	263
6.5	CANaerospace	263
6.5.1	Die Netzwerkschichten von CANaerospace	264
6.5.2	Datenrepräsentation in CANaerospace	266
6.5.3	Zeitverhalten in CANaerospace-Netzwerken	268
6.6	Die ARINC-Spezifikation 825	270
6.7	Das DeviceNet-Kommunikationsnetzwerk	274
6.7.1	Einleitung	274
6.7.2	Die DeviceNet-Architektur	274
6.7.3	Die physikalische Ebene	275
6.7.4	Buszugriff	275
6.7.5	Die Anwendungsebene	276
6.7.6	Das CIP-Objektmodell	278
6.7.7	Vordefinierte Einstellung einer Master/Slave-Verbindung	280
6.7.8	Geräteprofile	282
6.7.9	DeviceNet Safety	282
6.7.10	Hauptunterscheidungsmerkmale von DeviceNet	283
6.7.11	Zusammenfassung	285
6.7.12	Die ODVA (Herstellerorganisation für DeviceNet, ControlNet, EtherNet/IP und CompoNet)	286
6.8	CanKingdom	286
6.8.1	Übersicht	286
6.8.2	Hintergrund	287
6.8.3	Das Konzept hinter CanKingdom	288

6.8.4	Eine kurze Übersicht	289
6.8.5	Vokabular von CanKingdom	290
6.8.6	Die Seite des Königs	291
6.8.7	Die Seite des Bürgermeisters	293
6.8.8	Organisation der Stadt	294
6.8.9	Der Ordner	295
6.8.10	Ordnerkennzeichnung	295
6.8.11	Komponierbarkeit und Zugehörigkeit	298
6.9	SocketCAN	
	CAN-Programmierschnittstellen in Mehrbenutzerbetriebssystemen	299
6.9.1	Einführung	299
6.9.2	CAN als Netzwerkschnittstelle	300
6.9.3	Die Protokollfamilie PF_CAN	302
6.9.4	Integration in das Betriebssystem	304
6.9.5	Fazit	305
7	Entwurf deterministischer Netzwerke	307
7.1	Einleitung	307
7.2	Hintergrund	307
7.3	Problemanalyse	309
7.4	Timing-Modell	311
7.5	Netzwerkentwurf	314
7.6	Auswirkungen	315
7.6.1	Geringerer Testaufwand	315
7.6.2	Bessere Auslastung	315
7.6.3	Flexible Rekonfiguration	316
7.6.4	Vereinfachter Gateway-Entwurf	316
7.7	Fazit	317
8	CAN-Applikationen	319
8.1	Elektronik-Systemarchitektur von Kraftfahrzeugen	
	Anwendung des CAN-Busses – Status und Perspektiven	319
8.1.1	Bussysteme im Kraftfahrzeug	319
8.1.2	Die Anwendung von CAN in heutigen Bordnetzen	321
8.1.3	CAN und AUTOSAR	321
8.2	Vernetzte Systeme bei PSA Peugeot Citroën	323
8.2.1	Plattform-Strategie	323
8.2.2	Backbone der Elektronikarchitektur	324
8.2.3	Die Komplexität der Architektur	325
8.2.4	Topologie des Netzwerks	327
8.2.5	Softwarearchitektur	328
8.2.6	Zukünftige Kommunikationsnetzwerke	329
8.3	TTCAN – Applikationen	329
8.3.1	Softwareimplementierungen von TTCAN-X-by-Wire	329

8.3.2	Implementierung des TTCAN-Netzwerks	329
8.3.3	Umsetzung der Lenkung	330
8.3.4	Umsetzung der Bremse	332
8.3.5	Rückmeldungsnachrichten	332
8.3.6	Das endgültige System	332
8.4	Entwicklung von CAN in der Luftfahrt	333
8.5	CAN in der Welt der Flugzeuge	339
8.5.1	Warum CAN?	339
8.5.2	Luftfahrzeugspezifische Bedingungen an die physikalische Schicht	342
8.5.3	DAL, Sicherheit, Zertifizierung	343
8.5.4	Beispiel: Rauchmelder an einem sicherheitsrelevanten CAN-Netzwerk	344
8.6	Das dezentrale Heizungspumpen-System GENIAX	352
9	Test-Techniken	357
9.1	Ziele und Methoden von Konformitätstests	357
9.1.1	Wichtige Begriffe	359
9.1.2	Zweck, Nutzen und Bedeutung von Konformitätstests	360
9.1.3	Test-Methoden, -Standards und -Regeln	363
9.1.4	Funktionale Tests, Zeitverhalten und Performance	366
9.1.5	Verifikation und Validierung von Testsystemen	367
9.2	CAN-Transceiver-Konformitätstests	369
9.2.1	Standardisierung der physikalischen Schicht von CAN	369
9.2.2	Die Notwendigkeit von CAN-Transceiver-Tests	370
9.2.3	Die Testidee der CAN-Transceiver-Tests	372
9.2.4	Das Testsystem für CAN-Transceiver	376
9.2.5	Die Testfälle der CAN-Transceiver-Tests	379
9.2.6	Ausblick	383
9.3	CAN-Controller-Konformitätstests	384
9.3.1	Architektur und Implementierung der Testumgebung	384
9.3.2	Grundtestarten	387
9.3.3	Testtypen, -klassen und -fälle	387
9.3.4	Ergebnisse der Konformitätstests	391
9.4	CAN-Software-Tests	392
9.4.1	Anforderungen an Tests und Testsysteme	394
9.4.2	Testsystem-Strukturen für Embedded-Software-Tests	397
9.5	Modellbasiertes Testen	401
9.5.1	VHDL-AMS	402
9.5.2	Testmethodik	403
10	Emulation von CAN-Netzwerken	407
10.1	Test von CAN-Steuergeräten	407
10.1.1	Problemstellung und Anforderungen	407
10.1.2	Testmethodik	408
10.1.3	Simulation des CAN-Netzwerks	412
10.1.4	Testen mit Vector CANoe	413

10.2	Simulation und Test der CAN-Kommunikation	415
10.2.1	Entwicklung und Validierung der Kommunikation durch Restbussimulation	416
10.2.2	Implementierungssoftware CAN MM	417
10.2.3	Test und Experimentierwerkzeuge	421
10.2.4	CAN-Werkzeugkette	425
11	Weitere Bussysteme	427
11.1	LIN	427
11.1.1	OSI-Modell	428
11.1.2	Implementierungen	429
11.2	FlexRay	430
11.2.1	OSI-Modell	431
11.2.2	Implementierungen	434
11.3	Ethernet	434
11.3.1	OSI-Modell	435
11.3.2	Einige Beispiele für Ethernet-Implementierungen	437
11.3.3	Power over Ethernet (PoE)	437
11.3.4	Industrial Ethernet	438
11.4	MOST	438
11.4.1	OSI-Modell	438
11.4.2	Implementierungen	440
11.5	USB (Universal Serial Bus)	441
11.6	FireWire	442
11.7	Profibus	443
11.8	Interbus-S	444
11.9	AS-i (Actuator-Sensor-Interface)	445
11.10	eBUS (Energy Bus)	446
	Literaturverzeichnis	447
	Verweise auf nationale und internationale Standards	449
	Abkürzungsverzeichnis	455
	Autorenverzeichnis	473
	Sachwörterverzeichnis	479