

Inhalt

1	Grußwort des LIN-Konsortiums	17
2	Kommunikationsarchitekturen im Automobil	19
2.1	Entwicklung der Vernetzung im Fahrzeug	19
2.2	Anforderungen an Bussysteme im Fahrzeug	23
2.2.1	Verschiedene Ausprägungen von Topologien (allgemein)	23
2.2.2	Logische Arten der Datenflüsse	26
2.2.3	Client-Server- und Publisher-Subscriber-Modell	27
2.3	Typische Fahrzeugsysteme	28
2.3.1	Typische Fahrzeugtopologie	28
2.3.2	Datenfluss	29
2.3.3	Protokoll-Konversion zwischen Bussen	30
2.3.4	Randbedingungen für die Auswahl eines Bussystems	33
2.3.5	Integration	35
2.3.6	Signaldefinition und Nachrichtendefinition	36
2.3.7	Datenmanagement und Konsistenzsicherung im Netzwerk	36
2.3.8	Software-Entwicklung und Debugging	38
2.3.9	Systemintegration und Inbetriebnahme des Gesamtsystems	39
2.3.10	Einbettung des LIN-Busses in eine Fahrzeugarchitektur	39
3	Bussysteme im Automobil	41
3.1	SAE-Klassifizierung	41
3.2	MOST-Bus	43
3.2.1	Konzept	43
3.2.2	Funktionsblöcke des MOST-Systems	45
3.2.3	Das Protokoll	45
3.2.4	Aufbau der Kanäle	46
3.3	IEEE 1394	48
3.3.1	Das Konzept	49
3.3.2	Protokoll	49
3.4	Bluetooth	52
3.4.1	Geschichte	53
3.4.2	Technologie	53
3.4.3	Vernetzung	53
3.4.4	Bluetooth Protokoll-Stacks	55

3.4.5	Bluetooth Car Profile	57
3.5	Flexray [Meh 04]	58
3.5.1	Geschichte	59
3.5.2	Technologie	59
3.5.3	Vernetzung	62
3.5.4	Ausblick	64
3.6	CAN	64
3.6.1	Das Konzept	65
3.6.2	Arbitrierung	66
3.6.3	Protokoll	67
3.6.4	CAN-B und CAN-C	68
3.7	TTCAN, siehe auch [Eng 02], [Ets 02]	69
3.7.1	Zeitgesteuerte Nachrichtenübertragung	69
3.8	J1850	71
3.8.1	Der Nachrichtenframe	71
3.8.2	Variable Pulsbreitenmodulation (VPW – Variable Pulse Width Modulation)	71
3.8.3	Header	73
3.8.4	Das Bestätigungsfeld (In-Frame Response Field, IFR)	74
3.8.5	Die Bitübertragungsschicht	74
4	LIN – Geschichte und Technologie	75
4.1	Einführung	75
4.2	Geschichte und Motivation	75
4.3	Weiterentwicklung / Evolution	79
4.3.1	Der nächste große Schritt: LIN 2.0	80
4.4	LIN Konzept & Workflow	81
4.5	Off-The-Shelf Knoten	82
4.6	Conformance Test	83
4.7	Ausblick	83
5	LIN-Protokoll	85
5.1	Der LIN-Standard im Überblick	85
5.2	Das Kommunikationskonzept	87
5.3	Die LIN-Datenübertragungsschicht	89
5.3.1	Synchronisationsbreak und Synchronisationsmechanismus	89
5.3.2	Synchronisationsfeld	92
5.3.3	Identifizier	92
5.3.4	Datenfeld	94
5.3.5	Checksumme	94
5.4	Zeitsteuerung	95
5.5	Frametypen	95
5.5.1	Standardframe (Unconditional Frame)	95
5.5.2	Ereignisgesteuerte Frames (Event Triggered Frame)	96
5.5.3	Spontane Frames (Sporadic Frame)	98

5.5.4	Diagnoseframes (Diagnostic Frames)	98
5.5.5	Nutzerdefinierte Frames (User-defined Frame)	98
5.5.6	Reservierte Frames (Extended Frame)	99
5.6	Fehlererkennung und Fehlerbehandlung	99
5.6.1	Fehlererkennung	99
5.6.2	Fehlerbehandlung	100
5.7	Zustandsmodell	101
5.8	Netzwerkmanagement	104
5.8.1	Powermanagement	104
5.8.2	Statusmanagement	105
6	Bitübertragungsschicht	107
6.1	Grundstruktur eines LIN-Transceivers	107
6.2	Signalspezifikation	108
6.3	Die AC-Parameter	108
6.3.1	Berechnung des Signaltastverhältnisses („Duty Cycle“)	109
6.3.2	Berechnung der Schaltschwellen	112
6.4	Signallaufzeit	115
6.5	Charakteristik der Buslinie	115
6.6	Signalstromaufnahme des LIN- Moduls von der Buslinie	116
6.7	Automatische Baudratenerkennung	116
7	LDF und NCF	118
7.1	LIN-Entwurfsprozess, der sogenannte Workflow	118
7.2	Darstellung eines Beispiels: Heizregister	119
7.3	Definition des LIN Description Files	121
7.3.1	Netzwerkdefinition	121
7.3.2	Knotendefinition	121
7.3.3	Signaldefinition	123
7.3.4	Framedefinition	124
7.3.5	Definition des Zeitscheduling	126
7.3.6	Signaldecodierung für die Entwicklungstools	129
7.3.7	Weitere Definitionen der Configuration Language Specification	131
7.4	NCF-Node Capability File	133
7.4.1	Struktur des Node Capability Files	133
7.4.2	Der Bereich „general“	133
7.4.3	Der Bereich „diagnostic“	134
7.4.4	Framedefinition	135
7.4.5	Statusmanagement	136
7.4.6	Freier Text	136
8	Konfiguration und Diagnose	137
8.1	Konfiguration	137
8.1.1	Konfigurationskonzept	137

8.1.2	Konfigurationsprotokoll	139
8.1.3	Konfigurationsfunktionen	141
8.2	Diagnose	146
8.2.1	Diagnosemethoden	146
8.2.2	Diagnose-Transportschicht (PDU-Frames)	149
9	API	152
9.1	Konzept	152
9.1.1	Designprozess	152
9.1.2	Aufbau	153
9.2	Core API	155
9.2.1	Driver und Cluster Management	156
9.2.2	Signalinteraktion	157
9.2.3	Notification	158
9.2.4	Schedule Management	159
9.2.5	Interfacefunktionen	161
9.2.6	Hardwareinterface-Funktionen	163
9.3	Knotenmanagement	165
9.3.1	Statusfunktionen	166
9.3.2	Konfigurationsfunktionen	167
9.4	Diagnose-Transportschicht	168
9.4.1	RAW-API	168
9.4.2	Cooked-API	170
9.5	Implementierung der API	171
9.5.1	Signal-Handle	171
9.5.2	Frame-Handle	172
9.5.3	Interface des Masters	173
9.5.4	Interface des Slaves	174
9.5.5	Schedule Handle	175
9.5.6	Beispiel	175
10	Entwurfsprozess	180
10.1	Zeitverhalten von Signalen in LIN-Systemen	180
10.1.1	Latenzzeiten	180
10.1.2	Signallaufzeiten durch mehrere Netzwerke	182
10.1.3	Reaktionsverhalten von LIN-Clustern	182
10.2	„Off-the-Shelf“-Entwurfskonzept für LIN-Systeme	183
10.3	Entwurfsprozess von vernetzten elektronischen Systemen im Automobil	185
10.3.1	Zweidimensionales Ebenenmodell	185
10.3.2	Abbildung des V-Modells auf den LIN-Systementwurf	187
10.3.3	Globale Dokumente	188
10.4	Entwurf eines LIN-Clusters	189
10.4.1	System- und Subsystemebene	189

10.4.2	Komponentenebene	190
10.4.3	Test- und Integrationsphasen	191
11	Test von LIN-Systemen	192
11.1	Testkonzepte	192
11.1.1	Konformitätstests – Methodik und Architektur	194
11.1.2	Konformitätstests – Ableitung der Testfälle	197
11.1.3	Konformitätstests – Tester-Implementierung	202
11.1.4	Realisierungskonzept	205
11.2	Conformance Test Schicht 1	209
11.2.1	Operational Voltage Range – Test Case 2.1	212
11.2.2	Slope Control – Test Case 2.5	214
11.2.3	Masse-Drift – Test Case 2.7	215
11.2.4	Tester Implementierung	217
11.3	Conformance Test Schicht 2	218
11.3.1	Oscillator tolerance – Data Link Layer, Test 2.8	221
11.3.2	Kommunikation ohne/mit Fehlern	222
11.3.3	Test Implementierung	223
11.4	Modultest	225
11.4.1	Testmethoden	225
11.4.2	Testabdeckung	229
11.4.3	Konzeption von Modultests	229
11.5	Systemtest	232
11.5.1	Systemtest bei der Integration der Komponenten	232
11.5.2	Systemtest als Interoperabilitätstest für Komponenten	233
11.6	EMV-Tests	236
11.6.1	Intension	236
11.6.2	Testphilosophie zur Bewertung von LIN-Transceivern	237
11.6.3	Prinzipieller Testablauf für die EMV- Bewertung von LIN-Transceivern	241
11.7	Ausblick	247
12	Entwurf eines LIN-Türmoduls	250
12.1	Motivation	250
12.2	Systemdesign	252
12.2.1	Funktionsdefinition	252
12.2.2	Bustopologie	253
12.3	Definition der LIN-Kommunikation	255
12.3.1	Signal- und Telegramm-Definition	255
12.3.2	Scheduling	256
12.4	Abstimmung mit Projektpartnern	257
12.5	Systemtest	258
12.6	Erfahrungen und zukünftiges Vorgehen	259

13	Simulation von LIN-Clustern	261
13.1	Einführung in die Simulation von LIN-Clustern	261
13.2	Einsatzgebiet der Simulation	262
13.3	Modellierung der Komponenten	262
13.3.1	Modellanforderungen für die einzelnen Komponenten	263
13.3.2	Master-Terminierungs-Widerstand	264
13.3.3	LIN-Slave- und Master-Kondensator	264
13.3.4	LIN-Leitung	265
13.3.5	LIN-Verpölschutzdioden	265
13.3.6	Supply-Stützkondensatoren	265
13.4	Analysemöglichkeiten mittels Simulation	265
13.4.1	Simulation auf ECU-Niveau	266
13.4.2	Gesamtsimulation eines Netzwerkes	269
13.4.3	EMV-Analysen	270
13.4.4	ESD-Analysen	272
13.4.5	Thermische Simulation	272
13.5	Mixed-Mode-Simulation	274
13.5.1	Methoden der embedded Softwaresimulation	274
13.5.2	Modellierung eines LIN-Systems	275
13.6	Zusammenfassung	278
14	LIN 2.0 und SAE J2602	279
14.1	Einführung	279
14.1.1	Intention dieses Kapitels	279
14.1.2	Wer oder was ist die SAE?	279
14.1.3	Was ist das „Vehicle Architecture For Data Communication Standards“ Komitee?	280
14.1.4	Was ist die SAE J2602 LIN Task Force?	280
14.2	Motivation	281
14.3	Vergleich zwischen SAE J2602 und LIN 2.0	281
14.4	Unterschiede in der Diagnose	283
14.5	Unterschiede in den Konfigurations-Methoden	284
14.5.1	NADs, DNNs & ID Zuweisungen	285
14.6	Betrachtung aus Systemebene	288
14.6.1	Fehlererkennung und Benachrichtigung des Masters	288
14.6.2	Slave zu Slave Kommunikation	289
14.6.3	Event Triggered Frames	289
14.6.4	Nachrichtenübertragungszeit und Scheduling Tabellen	289
14.6.5	Bitübertragungsschicht und Netzwerk-Topologie	290
14.6.6	Elektrische Bus-Eigenschaften und Grenzen der Verdrahtung	290
14.6.7	EMC-Performance	291
14.6.8	Fehlertolerante Betriebsarten	292
14.7	Weitere nützliche Informationen	292
14.8	Zukünftige Arbeiten der SAE Task Force	293

15	LIN-Tools	294
15.1	Systementwurf am Beispiel einer Ampel	294
15.2	Vector CANoe.LIN	296
15.3	LIN-Entwicklungstool von IHR	303
15.4	Volcano	308
16	LIN-Hardware	315
16.1	Basisarchitektur eines LIN-Knotens	315
16.2	Klassifizierung der Hardware	316
16.3	LIN-Transceiver	317
16.4	Mikrocontroller	319
16.4.1	Synchronisationsbreak-Erzeugung	319
16.4.2	Synchronisation des Slaves	320
16.4.3	LIN-spezifische serielle Schnittstellen	321
16.4.4	Auswahl eines Mikrocontrollers	322
16.5	System Basis Chips (SBC)	323
16.6	Single-Package-Lösungen	324
A	Kenngrößen	327
A.1	Protokoll Kenngrößen	327
A.2	Physikalische Kenngrößen	328
B	Hardwareübersicht	330
B.1	LIN-Transceiver	330
B.2	Microcontroller	331
B.3	System Basis Chips	333
B.4	Single Package Lösungen	334
C	Weiterführende Literatur	335
C.1	Weiterführende Literatur zum Thema: LIN	335
C.1.1	1. Internationale LIN Konferenz, Ludwigsburg, 19.–20. September 2002	335
C.1.2	Elektronik Automotive – Sonderheft: LIN-Bus: Produkte, Applikationen, Tests; WEKA-Verlag 2004	336
C.1.3	LIN im Kraftfahrzeug: Aktuelle Einblicke – Herausforderungen – Trends; LIN-Anwender-Symposium; Vector Informatik; Stuttgart; 29. Juni 2004	337
C.1.4	Weitere Vorträge und Artikel	338
C.2	Weiterführende Literatur zum Thema: Datenkommunikation im Automobil	335
Glossar		341
Literaturverzeichnis		346
Sachverzeichnis		349