

**Fabian Wolf**

# **Fahrzeuginformatik**

**Eine Einführung in die Software- und  
Elektronikentwicklung aus der Praxis der  
Automobilindustrie**

**ö Springer Vieweg**

# Inhaltsverzeichnis

<b>1</b>	<b>Fahrzeugelektronik</b>	<b>1</b>
1.1	Gesamtfahrzeugarchitektur	1
1.2	Steuergeräte	3
1.3	Bordnetz	5
1.4	Elektrik-/Elektronikarchitektur	6
1.4.1	Funktionale Anforderungen	7
1.4.2	Technologie	8
1.4.3	Topologie	8
1.5	Entwurfsprozess der Elektrik-/Elektronikarchitektur	9
1.6	Digitale Bussysteme	10
1.6.1	Protokolle	13
1.6.2	Topologien	14
1.6.2.1	Sterntopologie	14
1.6.2.2	Bustopologie	14
1.6.2.3	Ringtopologie	15
1.6.2.4	Baumtopologie	15
1.6.3	Digitale Bussysteme – CAN	15
1.6.3.1	Allgemeine Eigenschaften	15
1.6.3.2	Buszugriff: CSMA/CA	16
1.6.3.3	CAN – Bus: Struktur der Nachricht	17
1.6.3.4	Bitstuffing	17
1.6.3.5	Kollisionsvermeidung	18
1.6.3.6	Fehlererkennung	18
1.6.3.7	High Speed CAN Buspegel	19
1.6.3.8	Low Speed CAN Buspegel	20
1.6.3.9	CAN-Bus: Weitere Anwendungen	20
1.6.3.10	CAN-Bus: Grenzen	20
1.6.4	Digitale Bussysteme – Flexray	21
1.6.5	Digitale Bussysteme – LIN	23
1.6.6	Digitale Bussysteme – MOST	25
1.6.7	Digitale Bussysteme – Vergleich	25

1.6.8	Digitale Bussysteme – Kombination	26
1.6.9	Unterschiede der Zugriffsverfahren	26
1.7	Sensoren	27
1.7.1	Wandlerketten	28
1.7.2	Sensorkennlinien	28
1.7.3	Abtastraten	29
1.7.4	Sensorpartitionierung	30
1.7.5	Beispiele für Sensoren	32
1.7.6	Schnittstellenbeschreibung	32
1.8	Aktoren	33
1.8.1	Schnittstellenbeschreibung	33
1.8.2	Beispiele für Aktoren	34
1.9	Mikrocontroller	34
1.10	Schaltungstechnik	35
1.10.1	Programmierung einer Schaltung durch Verdrahtung	35
1.11	Hardwarebeschreibungssprachen	36
1.12	Speicher	38
1.13	Energie	39
1.14	Zusammenfassung	39
1.15	Lernkontrollen	40
1.15.1	Bordnetz	40
1.15.2	Bussysteme	40
1.15.3	Sensoren und Aktoren	40
1.15.4	Steuergeräte	40
<b>2</b>	<b>Software im Fahrzeug</b>	<b>41</b>
2.1	Durchgängigkeit der Anforderungen	42
2.2	Abbildung Funktion auf Architektur	43
2.3	Softwarearchitektur	44
2.4	Echtzeitbetriebssysteme	45
2.4.1	Echtzeitanforderungen	46
2.4.2	Prozess und Prozesszustände	47
2.4.3	Kontextwechsel	47
2.4.4	Scheduling	48
2.5	Diagnose	49
2.5.1	Diagnose in der Automobiltechnik	50
2.5.2	Eigendiagnose: Onboard-Diagnose OBD	51
2.5.2.1	Beispiel Motorsteuergerät	51
2.5.2.2	Fehlererkennung	52
2.5.2.3	Fehlerbehandlung	53
2.5.2.4	Ereignisspeicher	53

2.5.3	Werkstattdiagnose: Offboard-Diagnose	54
2.5.4	Flashen im Kundendienst	54
2.5.5	Flashen im Fahrzeug-Lebenszyklus	55
2.6	Netzwerksoftware	56
2.6.1	Implementierung von Netzwerkprotokollen	56
2.6.2	Kommunikation und funktionale Vernetzung (Konnektivität) . . .	56
2.7	Funktionssoftware	57
2.7.1	Fragmentierung von Steuergeräten	57
2.7.2	Steuergerätedomänen	58
2.7.3	Klimasteuerung	59
2.7.4	Motorsteuerung	60
2.7.5	Lenkungssteuerung	62
2.7.6	Türsteuerung	63
2.7.7	Verteilte Funktionen	63
2.8	Überwachungskonzepte für sicherheitsrelevante Systeme	64
2.8.1	Anforderungen aus den Normen	64
2.8.2	Abgestufte Funktionseinschränkung	65
2.8.3	Diversitäre Programmierung	65
2.8.4	Redundanz in der Elektronik	66
2.8.5	Watchdog und Drei-Ebenen-Konzept	66
2.8.5.1	Funktionsrechner	67
2.8.5.2	Überwachungsrechner	68
2.8.5.3	Ablauf der Überwachung mittels Fehlerzähler	68
2.9	Herstellerübergreifende Softwarestandards	68
2.9.1	Historie	69
2.9.2	Beispiel OSEK/VDX: Betriebssysteme	70
2.9.3	Beispiel ASAM-MDX: Verteilte Softwareentwicklung	71
2.9.4	Beispiel AUTOSAR: Systemarchitektur	72
2.10	Zusammenfassung	76
2.11	Lernkontrollen	76
2.11.1	Architektur	76
2.11.2	Software	77
2.11.3	Echtzeitbetriebssystem	77
2.11.4	Sicherheitskonzept	77
2.11.5	Standards	77
<b>Softwareentwicklung in der Automobilindustrie</b>		<b>79</b>
3.1	Stand der Technik	81
3.2	Anforderungen und Architekturentwurf	81
3.2.1	Anforderungserhebung	82
3.2.2	Systemanforderungsanalyse	83
3.2.3	Systemarchitekturentwurf	85

3.2.4	Komponentenanforderungsanalyse	86
3.2.5	Komponentenarchitekturentwurf	87
3.3	Mechanik und Hardware/Elektronik	88
3.4	Softwareentwicklung	90
3.4.1	Softwareanforderungsanalyse	90
3.4.2	Softwareentwurf	91
3.4.3	Funktionssoftwareerstellung	92
3.4.4	Sicherheitssoftwareerstellung	93
3.4.5	Softwareintegrationstest	95
3.4.6	Softwaretest	95
3.5	Integrationstests für Komponenten und System	96
3.5.1	Komponententest	96
3.5.2	Komponentenintegrationstest	97
3.5.3	Systemintegrationstest	98
3.5.4	Systemtest	98
3.6	Übergreifende Prozesse der Softwareentwicklung	99
3.6.1	Qualitätssicherung	100
3.6.1.1	QM-Planung	101
3.6.1.2	QS-Planung	101
3.6.1.3	QS-Durchführung	102
3.6.2	Funktionale Sicherheit	103
3.6.2.1	Konzeption	103
3.6.2.2	Produktentwicklung	104
3.6.2.3	Funktionale Absicherung	105
3.6.3	Projektmanagement	106
3.6.3.1	Projektstart	106
3.6.3.2	Projektplanung	107
3.6.3.3	Projektdurchführung	107
3.6.3.4	Projektabschluss	108
3.6.4	Risikomanagement	109
3.6.4.1	Identifizieren und Bewerten der einzelnen Risiken . . .	110
3.6.4.2	Analyse von Risiken	110
3.6.4.3	Risikobewertung	111
3.6.4.4	Priorisierung	111
3.6.4.5	Festlegung von Maßnahmen	112
3.6.4.6	Risikoverfolgungsliste	112
3.6.5	Lieferantenmanagement	112
3.6.5.1	Lieferantenauswahl	113
3.6.5.2	Lieferantensteuerung	114
3.6.5.3	Lieferantenausphasung	115
3.6.6	Änderungsmanagement	115
3.6.6.1	Vorbereitung Änderungsmanagement	116

3.6.6.2	Änderung	116
3.6.7	Konfigurationsmanagement	117
3.6.7.1	Konfigurationsmanagement Aufbau	117
3.6.7.2	Konfigurationsmanagement Betrieb	118
3.6.8	Problemlösungsmanagement	119
3.6.8.1	Vorbereitung Problemlösungsmanagement	119
3.6.8.2	Problemlösung	120
3.6.8.3	Trendberichterstellung	121
3.6.9	Freigabemanagement	122
3.6.9.1	Freigabevorbereitung	122
3.6.9.2	Freigabedurchführung	122
3.7	Handcodierung in der Programmiersprache C	123
3.8	Modellbasierte Entwicklung	125
3.8.1	Modelle der Elektronik	125
3.8.2	Modelle der Regelstrecke	126
3.8.3	Modelle der Software	127
3.8.4	Modellbasierte Codegenerierung	127
3.8.5	Generierung von Schnittstellencode	128
3.9	Entwicklungswerkzeuge	131
3.9.1	Sicherheitsanalysen	131
3.9.2	Handcodierung von Steuergerätesoftware	131
3.9.3	Prüfung von Handcode/Codierungsrichtlinien	131
3.9.4	Modellbasierte Entwicklung	131
3.9.5	Test	132
3.9.6	Kommunikation	132
3.9.7	Sonstiges/IT-Infrastruktur	132
3.10	Modulbaukasten für Plattformsoftware	132
3.11	Rechenzeitanalyse von Softwarefunktionen	142
3.11.1	Stand der Technik	143
3.11.2	Rechenzeitbestimmende Faktoren	143
3.11.3	Anforderungen an das Messverfahren	144
3.11.4	Verfahren der hybriden Rechenzeitanalyse	145
3.11.5	Softwareumgebung	146
3.11.6	Analytische Betrachtung der Softwarefunktion	146
3.11.6.1	Kontrollflussgraph	147
3.11.6.2	Codeanalyse	147
3.11.6.3	Softwareabschnitte für die Laufzeitanalyse	147
3.11.7	Instrumentierung	148
3.11.7.1	Instrumentierung des Quellcodes	149
3.11.7.2	Instrumentierung der Funktionen	149
3.11.7.3	Instrumentierung der bedingten Verzweigungen	150
3.11.7.4	Erzeugen von Stimuli-Daten	151

3.11.7.5	Messaufbau	151
3.11.7.6	Durchführung der Messung	152
3.11.7.7	Genauigkeit der Messmethodik	152
3.11.8	Statische Analyse für Worst Case und Best Case	154
3.11.9	Zusammenfassung des Messverfahrens	155
3.12	Zusammenfassung	156
3.13	Lernkontrollen	156
3.13.1	Softwareentwicklung	156
3.13.2	Programmierung	157
3.13.3	Modulbaukasten	157
3.13.4	Rechenzeitanalyse	157
<b>4</b>	<b>Softwaretest</b>	<b>159</b>
4.1	Softwarefehler	159
4.1.1	Ursachen von Softwarefehlern	159
4.1.2	Schäden durch Softwarefehler	160
4.1.3	Prominente Beispiele	161
4.2	Grundlagen des Softwaretests	162
4.2.1	Testbegriff	162
4.2.2	Ablauf von Tests	162
4.2.3	Fehlerbegriff	163
4.2.4	Ziel des Tests	163
4.2.5	Maße für Software	164
4.2.5.1	Zyklomatische Komplexität	165
4.2.5.2	Halstead-Maße	165
4.2.5.3	Live-Variables	167
4.2.6	Vorgehen und Strategie im Softwaretest	167
4.2.7	Testaufwand	168
4.2.8	Weitere Begriffe und Definitionen	169
4.2.8.1	Verifikation und Validierung	169
4.2.8.2	Analysemethoden	169
4.2.8.3	Statische und dynamische Analyse	170
4.2.8.4	Testumgebung und Testfall	170
4.2.8.5	Testarten in der Automobilindustrie	171
4.3	Merkmalsräume von Softwaretests	172
4.3.1	Prüfebene	172
4.3.1.1	Unit-Tests und Modultests	175
4.3.1.2	Softwareintegrationstests	176
4.3.1.3	Systemintegrationstest	178
4.3.1.4	Abnahmetest	179
4.3.2	Prüfkriterium	180
4.3.2.1	Funktionstest	180

4.3.2.2	Trivialtest	181
4.3.2.3	Crashtest	181
4.3.2.4	Kompatibilitätstest	181
4.3.2.5	Zufallstest	182
4.3.2.6	Installationstest	182
4.3.2.7	Ergonomietest	182
4.3.2.8	Sicherheitstests	183
4.3.2.9	Komplexitätstest	183
4.3.2.10	Laufzeittests	184
4.3.2.11	Lasttest	184
4.3.2.12	Stresstest	185
4.3.3	Prüfmethodik und Prüftechniken: Black Box Tests	185
4.3.3.1	Äquivalenzklassentest	186
4.3.3.2	Zustandsbasierter Softwaretest	188
4.3.3.3	UseCaseTest	190
4.3.3.4	EntscheidungstabellenbasierterTest	190
4.3.3.5	Back-to-Back-Test	191
4.3.3.6	Regressionstest	192
4.3.3.7	Mutationstest	193
4.3.4	Prüfmethodik und Prüftechniken: White Box Tests	193
4.3.4.1	Anweisungüberdeckung (CO)	195
4.3.4.2	Kantenüberdeckung (C1)	196
4.3.4.3	Bedingungsüberdeckung (C2, C3)	197
4.3.4.4	Pfadüberdeckung (C4)	198
4.3.5	Testmetriken und Grenzen des Softwaretests	199
4.4	Hardware-in-the-Loop in der Automobilelektronik	201
4.4.1	Historie	202
4.4.2	Automatisierung der Tests	202
4.5	Zusammenfassung	204
4.6	Lernkontrollen	204
4.6.1	Fehlerbegriff	204
4.6.2	Software-Maße	205
4.6.3	Testverfahren	205
<b>5</b>	<b>Prozessmodellierung</b>	<b>207</b>
5.1	Der Softwareentwicklungsprozess	209
5.2	Das Wasserfallmodell	211
5.3	Das V-Modell	212
5.4	Anwendungsbezogene Prozessmodelle	214
5.4.1	Das Prototypenmodell	214
5.4.2	Das evolutionäre Modell	215
5.4.3	Das inkrementelle Modell	216

5.4.4	Das nebenläufige Modell	216
5.4.5	Das Spiralmodell	217
5.5	Zusammenfassung der klassischen Prozessmodelle	218
5.6	Alternativen zu klassischen Prozessmodellen	219
5.6.1	Rational Unified Process	219
5.6.2	Extreme Programming	221
5.6.3	ROPES	222
5.7	Einsatz der vorgestellten Prozessmodelle	222
5.8	Verbesserung von Entwicklungsprozessen: Reifegrade	223
5.9	CMMI	223
5.9.1	Reifegrade in CMMI	224
5.9.2	Prozessgebiete	225
5.9.3	Spezifische und generische Ziele	226
5.9.4	Zertifizierung der Organisation	227
5.10	SPICE	228
5.10.1	Referenz-und Assessmentmodell	228
5.10.2	Prozessdimension	230
5.10.3	Beispiel zur Prozessdimension:	231
5.10.4	Reifegraddimension und Reifegradstufen	232
5.10.5	Beispiel zu Reifegraddimensionen:	234
5.11	Vorteile und Nachteile von SPICE	234
5.12	Automotive SPICE	235
5.12.1	Assessments – Prinzipien	238
5.12.2	Assessments – Ablauf	242
5.13	Funktionale Sicherheit	244
5.13.1	IEC 61508	244
5.13.2	Safety Integrity Level	246
5.13.3	Failure in Time	246
5.13.4	Gefahren-und Risikoanalyse: Gefahrenanalyse	247
5.13.5	Gefahren-und Risikoanalyse: Risikoanalyse	252
5.13.6	IEC 61508 Assessment	253
5.13.7	ISO 26262	254
5.14	Agile Methoden	255
5.15	Reifegrade und Prozessmodellierung in der Praxis	257
5.16	Zusammenfassung	258
5.17	Lernkontrollen	258
5.17.1	Prozessmodelle	258
5.17.2	Reifegradmodelle	259
5.17.3	Funktionale Sicherheit	259

<b>6</b>	<b>Software-Variabilität in der Automobilindustrie</b>	261
6.1	Das Body Comfort System mit variabler Fenstersteuerung als laufendes Beispiel	263
6.2	Grundlagen der Software-Produktlinien	266
6.2.1	Erstellung einer Software-Produktlinie	267
6.2.2	Verwendung einer Software-Produktlinie	270
6.3	Variabilitätsmodelle	273
6.3.1	Feature Models mit Cross-Tree Constraints	273
6.3.2	Decision Models mit Resolution-Constraints	276
6.3.3	Orthogonal Variability Models (OVMs) mit Constraints	279
6.4	Variabilitätsrealisierungsmechanismen	281
6.4.1	Annotativer Variabilitätsrealisierungsmechanismus	281
6.4.2	Kompositionaler Variabilitätsrealisierungsmechanismus	284
6.4.3	Transformativer Variabilitätsrealisierungsmechanismus	287
6.5	Wahl und Verwendung geeigneter Technologien für den Praxiseinsatz von Software-Produktlinien	290
6.5.1	Wahl und Verwendung des Variabilitätsmodells	290
6.5.2	Wahl und Verwendung des Variabilitätsrealisierungsmechanismus	291
6.6	Weiterführende SPL Konzepte und Techniken	295
6.6.1	Prozesse zur Aufnahme der Entwicklung von Software- Produktlinien	295
6.6.1.1	Proaktive Entwicklung von Software-Produktlinien . . .	295
6.6.1.2	Reaktive Entwicklung von Software-Produktlinien . . .	296
6.6.1.3	Extraktive Entwicklung von Software-Produktlinien . .	297
6.6.2	Spezialisierte Notationen für Feature Models	298
6.6.2.1	Kardinalitätsbasierte Feature Models	298
6.6.2.2	Attributierte Feature Models	299
6.6.2.3	Hyper-Feature Models	300
6.6.2.4	Temporal Feature Models	300
6.6.3	Multi-Software-Produktlinien (MSPLs)	301
6.6.4	Staged Configuration	304
6.7	Zusammenfassung	306
<b>7</b>	<b>Zusammenfassung und Abschluss</b>	307
	<b>Literatur</b>	309
	<b>Sachverzeichnis</b>	311