

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
1.1	Motivation .....	1
1.2	Klassifikation, Charakteristika.....	3
1.3	Anwendungen, Beispiele und Branchen.....	6
1.4	Begriffsdefinitionen .....	8
1.5	Logischer Aufbau eingebetteter Systeme .....	10
1.5.1	Kontrolleinheit .....	12
1.5.2	Regelstrecke .....	15
1.5.3	Benutzerschnittstelle .....	21
1.6	Softwareentwicklung eingebetteter Systeme.....	22
1.6.1	Motivation .....	22
1.6.2	Begriffsklärung.....	23
1.6.3	Entwurf.....	23
1.7	Besondere Herausforderungen.....	24
1.8	Zusammenfassung.....	25
<b>2</b>	<b>Nebenläufige Systeme .....</b>	<b>27</b>
2.1	Einführung .....	28
2.1.1	Multitasking.....	29
2.1.2	Multithreading .....	29
2.1.3	Prozesssynchronisation und -kommunikation .....	31
2.2	Grundlegende Modelle für die Nebenläufigkeit.....	32
2.3	Verteilte Systeme .....	34
<b>3</b>	<b>Echtzeit, Echtzeitsysteme, Echtzeitbetriebssysteme.....</b>	<b>39</b>
3.1	Echtzeitsysteme .....	39
3.2	Ereignissteuerung versus Zeitsteuerung .....	41
3.3	Echtzeitbetriebssysteme .....	42

3.3.1	Aufbau und Aufgaben von Betriebssystemen .....	43
3.3.2	Betriebssystemarchitekturen .....	44
3.3.3	Echtzeitfähige Betriebssysteme .....	45
3.3.4	Zeitgeber und Zugriffsebenen auf Zeit .....	50
3.3.5	Prozesse .....	53
3.3.6	Multitasking und Scheduling .....	54
3.3.7	Scheduling in Echtzeitbetriebssystemen .....	57
3.3.8	Speicherverwaltung .....	59
3.4	VxWorks als Beispiel eines Echtzeitbetriebssystems .....	61
3.4.1	Das Laufzeitsystem .....	63
3.4.2	Exkurs: Der POSIX Standard .....	63
3.4.3	Das I/O-Subsystem von VxWorks .....	64
3.4.4	Unterstützung verteilter Systeme in VxWorks .....	64
3.4.5	VxWorks Entwicklungswerkzeuge .....	64
3.5	Weitere Beispiele eingebetteter Betriebssysteme .....	66
3.5.1	Symbian OS .....	67
3.5.2	Palm OS .....	68
3.5.3	Windows CE .....	69
3.5.4	QNX .....	70
3.5.5	Embedded Linux .....	72
3.6	Zusammenfassung .....	73
<b>4</b>	<b>Programmierung eingebetteter Systeme .....</b>	<b>75</b>
4.1	Der Einsatz von C/C++ für eingebettete Systeme .....	77
4.2	Embedded C++ .....	78
4.2.1	Einschränkung: Das Schlüsselwort „mutable“ .....	80
4.2.2	Einschränkung: Ausnahmebehandlung .....	80
4.2.3	Typidentifikation zur Laufzeit .....	81
4.2.4	Namenskonflikte .....	81
4.2.5	Templates .....	81
4.2.6	Mehrfachvererbung und virtuelle Vererbung .....	81
4.2.7	Bibliotheken .....	82
4.2.8	EC++ Styleguide .....	82
4.3	Der Einsatz von Java für eingebettete Systeme .....	83
4.3.1	Java 1 .....	85
4.3.2	Java 2 (J2ME) .....	87
4.3.3	JavaCard .....	90
4.3.4	Echtzeiterweiterungen für Java .....	93
4.4	Synchrone Sprachen .....	98

4.5	Ereignisbasierter Ansatz am Beispiel von Esterel.....	99
4.5.1	Historie.....	100
4.5.2	Hypothese der perfekten Synchronie.....	100
4.5.3	Determinismus.....	104
4.5.4	Allgemeines .....	105
4.5.5	Parallelität .....	106
4.5.6	Deklarationen.....	106
4.5.7	Instruktionen.....	109
4.5.8	Beispiel: Die sogenannte ABRO-Spezifikation .....	111
4.5.9	Semantik .....	111
4.5.10	Kausalitätsprobleme.....	112
4.5.11	Codegenerierung und Werkzeuge.....	116
4.6	Synchrone Datenflusssprachen am Beispiel von Lustre .....	118
4.6.1	Datenfluss und Clocks.....	119
4.6.2	Variablen, Konstanten und Gleichungen .....	120
4.6.3	Operatoren und Programmstruktur .....	120
4.6.4	Assertions (Zusicherungen) .....	122
4.6.5	Compilation.....	122
4.6.6	Verifikation und automatisches Testen.....	124
4.6.7	Lustre im Vergleich zu Signal .....	125
4.7	Zeitgesteuerter Ansatz am Beispiel von Giotto.....	125
4.8	Zusammenfassung.....	136
<b>5</b>	<b>Softwareentwurf eingebetteter Systeme .....</b>	<b>139</b>
5.1	Modellierung eingebetteter Systeme .....	140
5.2	Formale Methoden .....	141
5.3	Statecharts .....	142
5.4	Die Unified Modeling Language (UML) .....	145
5.5	Der Ansatz ROOM.....	151
5.5.1	Softwarewerkzeuge und Umgebung .....	151
5.5.2	Einführung.....	152
5.5.3	Echtzeitfähigkeit .....	154
5.6	Hardware/Software-Codesign.....	155
5.7	Die MARMOT-Methode .....	161
5.8	Hybride Systeme und hybride Automaten .....	164
5.8.1	Einleitung.....	164
5.8.2	Spezifikation hybrider Systeme .....	167
5.9	Zusammenfassung.....	171
<b>6</b>	<b>Softwarequalität eingebetteter Systeme .....</b>	<b>173</b>
6.1	Motivation .....	173

6.2	Begriffe .....	174
6.3	Zuverlässigkeit eingebetteter Systeme .....	178
6.3.1	Konstruktive Maßnahmen .....	182
6.3.2	Analytische Verfahren .....	184
6.3.3	Stochastische Abhängigkeit .....	186
6.3.4	Gefahrenanalyse .....	186
6.4	Sicherheit eingebetteter Systeme .....	188
6.4.1	Testen .....	190
6.4.2	Manuelle Prüftechniken .....	195
6.4.3	Formale Verifikation .....	196
6.5	Zusammenfassung .....	199
<b>7</b>	<b>Vorgehensmodelle und Standards der Entwicklung....</b>	<b>201</b>
7.1	Das Wasserfall-Modell .....	201
7.2	Das V-Modell .....	202
7.3	Das V-Modell XT .....	205
7.3.1	Grundlagen .....	206
7.3.2	Anwendung des V-Modell XT .....	207
7.3.3	Zielsetzung und Aufbau des V-Modell XT .....	208
7.3.4	V-Modell XT Produktvorlagen .....	211
7.3.5	V-Modell XT Werkzeuge .....	211
7.4	Die ROPES-Methode .....	212
7.5	Der OSEK-Standard .....	213
7.6	AUTOSAR .....	215
7.7	Zusammenfassung .....	217
<b>8</b>	<b>Schlussbemerkungen.....</b>	<b>219</b>
<b>Literaturverzeichnis .....</b>		<b>223</b>
<b>Sachverzeichnis .....</b>		<b>229</b>