

Kurs 02423  
Prozeßautomatisierung II

Kurs 21312  
Echtzeitsysteme II

---

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil dieses Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

# Vorwort

In der Prozeßautomatisierung arbeitende Digitalrechner haben die Aufgabe, Programme auszuführen, die mit externen technischen Prozessen assoziiert sind. Auf Anforderung dieser Prozesse hin müssen Erfassung und Auswertung von Prozeßdaten sowie geeignete Reaktionen pünktlich und schritthaltend mit der Prozeßdynamik ausgeführt werden. Dabei steht nicht die Schnelligkeit der Bearbeitung im Vordergrund, sondern die Rechtzeitigkeit der Reaktionen innerhalb vorgegebener und vorhersehbarer Zeitschranken. Die so definierten Realzeitsysteme sind mithin dadurch charakterisiert, daß die funktionale Korrektheit eines Systems nicht nur vom Resultat einer Berechnung, sondern auch von der Zeit abhängt, wann dieses Resultat produziert wird, womit sie sich grundlegend von anderen informationstechnischen Systemen unterscheiden.

Die Begriffe Realzeit-, Echtzeit- und Prozeßdatenverarbeitung sowie rechnergestützte Prozeßautomatisierung werden in diesem Kurs alle synonym für Steuerung und Regelung technischer Prozesse mit programmierbaren elektronischen Systemen verwendet. Unter letzterem, in der internationalen Normung eingeführten Oberbegriff lassen sich programmierbare Logikbausteine, speicherprogrammierbare Steuerungen, Industrie-PCs, Prozeßrechner, Prozeßleitsysteme und ganz allgemein eingebettete (Rechner) Systeme zusammenfassen.

Die technische Entwicklung ist durch das Bestreben gekennzeichnet, technische Vorgänge möglichst kostengünstig durchzuführen. Im Zuge dieser Entwicklung werden immer mehr selbsttätig arbeitende Einrichtungen zur Meßwerterfassung, Steuerung und Regelung eingesetzt. So entstanden in der Vergangenheit die weitgehend selbständigen Fachgebiete Meß-, Steuerungs- und Regelungstechnik. Mit dem Vordringen der Mikroelektronik werden für alle Automatisierungsaufgaben zunehmend digitale Verfahren eingesetzt, Meß-, Steuer- und Regelgeräte wurden weitgehend digitalisiert. Dadurch besteht für entwickelnde Ingenieurinnen und Ingenieure die Notwendigkeit, neben den Methoden der Meß-, Steuerungs- und Regelungstechnik auch Verfahren der Software- und Rechnertechnik zu beherrschen, um Prozeßautomatisierungs- und allgemein eingebettete Systeme zu konstruieren. Um diese letztgenannten Verfahren und um ihre Anwendung zur Entwicklung von Echtzeitsystemen geht es im vorliegenden Kurs.

Ziel des Kurses ist es, Kenntnisse und Fähigkeiten zu vermitteln, die die Studierenden in die Lage versetzen, als Ingenieurin oder Ingenieur auf dem Gebiet der Prozeßautomatisierung tätig zu sein. Darüber hinaus sollen durch Bearbeitung dieses Kurses und insbesondere auch der dazugehörigen Übungsaufgaben die Fähigkeiten erworben werden, geräte- und programmtechnische Fragen im Zusammenhang mit Prozeßautomatisierungsaufgaben fachgerecht zu klären, die Literatur des Gebietes zu verstehen und sich mit ihrer Hilfe weiterzubilden.

Der im letzten Semester vorangegangene erste Kursteil war Fragen des grundsätzlichen strukturellen, geräte- und programmtechnischen Aufbaus von Prozeßautomatisierungssystemen gewidmet. Der vorliegende, zum Sommersemester 2005 völlig neu geschriebene zweite Kursteil vertieft nun die Betrachtungen von Feldbussen (Kurseinheit 12), Echtzeitbetriebssystemen (Kurseinheit 13) und Vorgehensweisen zur Erstellung von Software für die Prozeßautomatisierung. Bei letztgenanntem Themengebiet liegt der Schwerpunkt nicht auf der Echtzeitprogrammierung im engeren Sinne, sondern auf Anforderungsanalyse sowie Entwurf und Strukturierung automatisierungstechnischer Software (Kurseinheit 9), wozu auch graphische Methoden vorgestellt werden (Kurseinheit 8). Da zur Programmverifikation zunehmend formale Methoden an Bedeutung gewinnen, wird in dieses Gebiet unter besonderer Berücksichtigung von Petri-Netzen und temporaler Logik eingeführt (Kurseinheit 14). Weiterhin wird eine Übersicht zu aktiven und Echtzeitdatenbanken gegeben (Kurseinheit 10), die im Rahmen datenintensiver Automatisierungsanwendungen von Interesse sind. In Kurseinheit 11 wird die terrestrische und satellitengestützte Zeitsignalverbreitung über Funk und rechnerinterne Zeitverwaltung eingehend dargestellt – beides Themen, denen in der Informationstechnik nicht die gebührende Beachtung entgegengebracht wird.

Vielleicht werden Sie sich wundern, daß ein neu geschriebener Kurstext weiterhin unter Verwendung der alten Rechtschreibung abgefaßt wurde. Der Grund liegt darin, daß die neue Rechtschreibung häufig unsinnig ist oder den Sinn entstellt. Zwar schreibt das Ministerium für Wissenschaft und Forschung des Landes Nordrhein-Westfalen vor, die neue Rechtschreibung anzuwenden, jedoch kann dies nicht für Lehrtexte gelten, die unter dem besonderen Schutze des Grundgesetzes stehen, das in Artikel 5 Absatz 3 die Freiheit der Lehre garantiert.

Abschließend möchten wir allen jenen ganz herzlich danken, die an der Erstellung des vorliegenden Kurstextes mitgewirkt haben. Da sind an erster Stelle die Herren Dipl.-Math. Wolfgang Hümb's und Dipl.-Math. Klaus Kuzyk als Autoren von Kurseinheit 14 und Herr Dr.-Ing. Thomas Erdner zu nennen, der Kurseinheit 12 geschrieben hat. Frau Dr. Janine Magnussen sowie die Herren Dr.-Ing. Markus Gemeinder, Thorsten Schrinner und Dipl.-Ing. Martin Skambraks haben den Text sprachlich überarbeitet und korrekturgelesen. Frau Dipl.-Ing. Jutta Düring hat ebenfalls korrekturgelesen, die Formatierung verbessert und ebenso wie Herr Georg Schindel Zeichnungen angefertigt.

Hagen, im Winter 2004

*Prof. Dr. Dr. Wolfgang A. Halang  
Jun.-Prof. Dr. Zhong Li*

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>iii</b>
<b>8 Graphische Echtzeitprogrammierung</b>	<b>1</b>
8.1 Strukturierte Echtzeitprogrammierung	2
8.1.1 Anforderungen	2
8.1.2 Grundelemente der Benutzersprache	3
8.1.2.1 Struktur-Task	3
8.1.2.2 Task-Bündel	5
8.1.2.3 Synchronisationsblock	7
8.1.2.4 Integrierte Signalisierung	8
8.1.2.5 Ausweichreaktion	9
8.1.3 Anwendungsbeispiel	9
8.2 Graphischer Entwurf von Echtzeitprogrammen in LACATRE	11
8.2.1 Einführung und Zielsetzungen	11
8.2.2 Elementarobjekte	14
8.2.2.1 Graphische Objektsymbole	14
8.2.2.2 Primitive	15
8.2.2.3 Der Textmodus von LACATRE	16
8.2.2.4 Zustandsprimitive zur Objekterzeugung und -vernichtung	16
8.2.3 Konfigurierbare Objekte	16
8.2.3.1 Semaphor	17
8.2.3.2 Briefkasten, Nachricht und Nachrichtenpuffer	18
8.2.3.3 Betriebsmittel	20
8.2.3.4 Ereignis	22
8.2.4 Programmierbare Objekte	24
8.2.4.1 Task	24
8.2.4.2 Unterbrechungen, Alarme und Ausnahmen	26
Unterbrechungsobjekt	27
Alarmobjekt	28
Ausnahmeobjekt	29
8.2.4.3 Algorithmischen Formen	30
8.2.5 Verwaltung der Ablaufphasen und Anwendungsbeispiele	31
8.2.5.1 Steuerung eines Förderbandes	32
8.2.5.2 Ablaufphasen	34
8.2.5.3 Steuerung der Bewegung eines Roboterarms	36
8.2.6 Entwicklungsumgebung für Echtzeitprogramme	37
<b>9 Programmentwicklungsumgebungen</b>	<b>43</b>
9.1 Systementwicklung	44
9.1.1 Systemlebenszyklus	44

	Systemanforderungsanalyse . . . . .	44
	Systementwurf . . . . .	46
	Entwicklung von Konfigurationseinheiten . . . . .	46
	Systemintegration und -test . . . . .	46
	Bewertung und Abnahme . . . . .	46
9.1.2	Software-Lebenszyklus . . . . .	47
	Anforderungsanalyse und -spezifikation . . . . .	48
	Vorläufiger Entwurf . . . . .	49
	Detaillierter Entwurf . . . . .	49
	Implementierung . . . . .	49
	Integration und Test . . . . .	50
	Formaler Qualifikationstest (Validierung) . . . . .	50
	Wartung . . . . .	50
9.1.3	Klassische Software-Entwicklungsmethoden . . . . .	51
9.1.4	Prototyperstellung . . . . .	56
9.1.5	Objektorientierte Entwicklung . . . . .	57
9.1.6	Transformationelle Implementierung . . . . .	61
9.1.7	Bewertung der Entwicklungsparadigmen . . . . .	65
9.1.8	Rechnergestützte Entwicklungswerkzeuge . . . . .	65
9.2	Das System EPOS . . . . .	67
9.2.1	Überblick . . . . .	67
	Anforderungsanalyse . . . . .	69
	Systementwurf . . . . .	70
	Projekt-, Konfigurations-Management und Qualitätssicherung . . . . .	71
	Bewertung . . . . .	71
9.2.2	Beispiel: Ein Chemischer Prozeß . . . . .	72
9.2.3	Die Sprache EPOS-R zur Anforderungsanalyse . . . . .	72
	9.2.3.1 Übersicht . . . . .	72
	9.2.3.2 Sektionen im Rahmen von Gliederungsschemata . . . . .	80
	9.2.3.3 Identifizierbare Aufgabenkomponenten . . . . .	81
	9.2.3.4 Entscheidungstabellen . . . . .	88
	9.2.3.5 Definition von Begriffen und Benennungen (Lexikon) . . . . .	93
9.2.4	Die Sprache EPOS-S für den Systementwurf . . . . .	93
	9.2.4.1 Übersicht . . . . .	93
	9.2.4.2 Sprachelemente zum Aufbau von EPOS-S-Spezifikationen . . . . .	97
	9.2.4.3 Entwurfsobjekte des Typs Aktion . . . . .	101
	9.2.4.4 Entwurfsobjekte des Typs Modul . . . . .	102
	9.2.4.5 Entwurfsobjekte des Typs Daten . . . . .	104
	9.2.4.6 Entwurfsobjekte des Typs Schnittstelle . . . . .	109
	9.2.4.7 Entwurfsobjekte des Typs Ereignis . . . . .	109
	9.2.4.8 Entwurfsobjekte des Typs Bedingung . . . . .	111
	9.2.4.9 Entwurfsobjekte des Typs Ausführungseinheit . . . . .	114
	9.2.4.10 Vorgehensweise beim Systementwurf mit EPOS-S . . . . .	115
<b>10</b>	<b>Aktive und Echtzeitdatenbanken</b>	<b>119</b>
10.1	Einführung . . . . .	120
10.2	Herkömmliche Datenbanken . . . . .	122
	10.2.1 Datenbankmodelle und Abfragesprachen . . . . .	123
	10.2.2 Transaktionen . . . . .	124
	10.2.2.1 Wiederherstellung nach Systemausfall . . . . .	124
	10.2.2.2 Steuerung der Nebenläufigkeit . . . . .	124

10.3	Zeitliche Gültigkeit . . . . .	125
10.4	Transaktionsmodell . . . . .	125
10.5	Notfallpläne . . . . .	129
10.6	Zuteilung von Transaktionen . . . . .	129
10.7	Nebenläufigkeitssteuerung . . . . .	132
10.8	Zuteilung von Ein-/Ausgabeoperationen . . . . .	136
10.9	Ursachen von Unvorhersagbarkeit . . . . .	138
10.9.1	Abhängigkeit von Daten . . . . .	138
10.9.2	Daten- und Betriebsmittelkonflikte . . . . .	138
10.9.3	Dynamische Seitenwechsel und Ein-/Ausgabe . . . . .	139
10.9.4	Transaktionsabbruch mit Rücksetzen und Neustarten . . . . .	140
10.10	Pufferverwaltung und hauptspeicherresidente Datenbanken . . . . .	140
10.10.1	Pufferverwaltung . . . . .	141
10.10.2	Hauptspeicherresidente Datenbanksysteme . . . . .	142
10.11	Wiederherstellung in Echtzeitdatenbanken . . . . .	143
10.12	Aktive Datenbanken . . . . .	144
10.12.1	Regelbasierte Systeme . . . . .	144
10.12.1.1	Ereignisse . . . . .	146
	Primitive Ereignisse . . . . .	146
	Zusammengesetzte Ereignisse . . . . .	147
10.12.1.2	Bedingungen . . . . .	148
10.12.1.3	Aktionen . . . . .	149
10.12.2	Ausführungssemantik . . . . .	149
10.12.2.1	Kopplungsmodi . . . . .	149
10.12.2.2	Dynamische Regeln . . . . .	150
10.12.2.3	Kaskadenauslösung . . . . .	151
10.12.2.4	Konfliktlösung . . . . .	151
10.12.2.5	Prioritäten . . . . .	151
10.12.3	Aktive im Vergleich zu passiven Datenbanken . . . . .	152
10.13	Aktive Echtzeitdatenbanken . . . . .	152
10.13.1	Regeln und Vorhersagbarkeit . . . . .	153
10.13.1.1	Kopplungsmodi . . . . .	153
10.13.1.2	Kaskadenauslösung . . . . .	153
10.13.1.3	Zusammengesetzte Ereignisse . . . . .	153
10.13.2	Echtzeitregelverwaltung . . . . .	154
10.13.2.1	Notfallpläne . . . . .	154
10.13.2.2	Zeitkennende Regeln und ihre Definitionssprache . . . . .	154
10.13.3	Diskussion . . . . .	155
<b>11</b>	<b>Zeitsignalverbreitung und rechnerinterne Zeitverwaltung</b>	<b>157</b>
11.1	Terrestrische Zeitsignalverbreitung . . . . .	158
11.1.1	Zeitübertragung . . . . .	158
11.1.1.1	Verfügbare Zeitinformationen . . . . .	158
11.1.1.2	Kontinuität der Zeitskala . . . . .	159
11.1.1.3	Signalverfügbarkeit . . . . .	160
11.1.1.4	Stabilität und Ausbreitungsverzögerung . . . . .	160
11.1.1.5	Überlagerung von Raum- und Bodenwelle . . . . .	161
11.1.2	Empfang von Zeitsignalen . . . . .	161
11.1.2.1	Empfang innerhalb von Gebäuden . . . . .	161
11.1.2.2	Sicherheit der Zeitinformatiionsdekodierung . . . . .	162
11.1.2.3	Synchronisationsgenauigkeit . . . . .	162

11.1.2.4	Verbesserte Genauigkeit durch Verwendung binärer Phasenschlüssel . . . . .	163
11.1.2.5	Phasengekoppelte Quarzkristalloszillatoren . . . . .	164
11.1.3	Zeitsignalgebende Stationen und Protokolle . . . . .	165
11.1.3.1	DCF 77 . . . . .	166
11.1.3.2	MSF . . . . .	167
11.1.3.3	OMA . . . . .	168
11.1.3.4	France Inter . . . . .	169
11.1.3.5	WWV/WWVH und WWVB . . . . .	170
11.1.3.6	Standardisierung der Zeitcodes . . . . .	172
11.1.4	Verfügbare Funkuhren . . . . .	173
11.1.5	Zusammenfassung . . . . .	173
11.2	Zeitverbreitung mit GPS . . . . .	175
11.2.1	Überblick über GPS . . . . .	175
11.2.1.1	Geschichte von GPS . . . . .	175
11.2.1.2	Zivil- und Militär-GPS . . . . .	176
11.2.1.3	GPS-Segmente . . . . .	176
11.2.1.4	GPS-Zeit . . . . .	178
11.2.1.5	GPS-Navigationsnachrichten . . . . .	178
11.2.1.6	Inhalt der Teilrahmen . . . . .	179
11.2.1.7	Satellitencodierung im GPS-Empfänger . . . . .	180
11.2.1.8	Grundlegende Aufgaben von Empfängern . . . . .	181
11.2.1.9	Selektive Verfügbarkeit . . . . .	181
11.2.2	Fehlerquellen . . . . .	182
11.2.3	GPS-Empfänger . . . . .	185
11.2.3.1	Satellitenauswahl und Signalempfang bei Inbetriebnahme . . . . .	185
11.2.3.2	Zeit- und Frequenzmessungen . . . . .	186
11.2.3.3	Differentielle Verfahren . . . . .	187
11.2.3.4	Kopplung mit Echtzeitanwendungen . . . . .	188
11.2.3.5	Empfängerstatus und Alarme . . . . .	188
11.2.4	Ausfallmodi bei Verbreitung des Zeitsignals mit GPS . . . . .	189
11.2.4.1	Ausfälle im Kontrollsegment . . . . .	189
11.2.4.2	Ausfälle im Raumsegment . . . . .	189
11.2.4.3	Ausfälle im Anwendersegment . . . . .	190
11.2.5	Zusammenfassung . . . . .	190
11.3	Hochgenaue simultane Ereignisverarbeitung . . . . .	192
11.3.1	Prozessor zur Bearbeitung gleichzeitiger Ereignisse . . . . .	193
11.3.2	GPS-basierte hochgenaue Zeitsteuereinheit . . . . .	195
11.3.3	Formale Funktionsbeschreibung des Ereignisprozessors . . . . .	200
11.3.4	Algorithmen der Task-Verwaltung . . . . .	203
11.3.5	Bewertung . . . . .	206
<b>12</b>	<b>Feldbusse</b>	<b>209</b>
12.1	Einführung . . . . .	210
12.2	Signalcodierung . . . . .	213
12.2.1	Fehlerreduktion durch Signalcodierung . . . . .	213
12.2.2	Signalcodierung mit modifiziertem FSK-Verfahren . . . . .	213
12.2.2.1	Vergleich der Signalformen . . . . .	213
	Sinusschwingung . . . . .	214
	Sinusquadratschwingung . . . . .	214
	Rechteckschwingung . . . . .	215

12.2.2.2	Kombination der Signalformen . . . . .	216
	Sinusquadratimpulse für die Datenübertragung . . . . .	216
	Rechteckimpulse zur Übertragung von Synchron- und Status- zeichen . . . . .	217
12.2.3	Detektierung signalcodierter Datenbits . . . . .	219
12.2.3.1	Auswahl geeigneter Signalparameter . . . . .	219
12.2.3.2	Simulation der FFT-Frequenzdetektierung . . . . .	221
	Simulationsergebnisse ohne Störungen . . . . .	222
	Simulationsergebnisse mit Störungen . . . . .	223
	Auswertung der Ergebnisse . . . . .	223
12.2.3.3	Goertzel-Frequenzdetektierung . . . . .	225
	Aufbau des Goertzel-Algorithmus . . . . .	225
	Bewertung . . . . .	226
12.2.3.4	Mittelwertbilder und Auswertelogik . . . . .	226
	Mittelwertbilder . . . . .	226
	Auswertelogik . . . . .	226
12.2.4	Detektierung von Synchron- und Statussignalen . . . . .	226
12.2.4.1	Detektion mit Flankendetektor . . . . .	227
12.2.4.2	Detektion mit signalangepasstem Filter . . . . .	227
12.2.5	Aufbau des signalangepassten Filters . . . . .	228
12.2.5.1	Vergleich der Synchronsignaldetektoren . . . . .	231
12.2.6	Aufbau der Ein- und Ausgangsstufen . . . . .	231
12.3	Datencodierung . . . . .	236
12.3.1	Datencodierung von Nibbles . . . . .	236
12.3.2	Berechnung der Korrekturstellen . . . . .	236
12.3.3	Hamming-Codierung von Daten-Nibbles . . . . .	239
12.3.4	Decodierung gesicherter Daten-Nibbles . . . . .	241
12.3.4.1	7-Bit-Decodierung auf dem Übertragungsweg . . . . .	241
12.3.4.2	8-Bit-Decodierung im Empfänger . . . . .	243
12.3.5	Fehlerwahrscheinlichkeiten . . . . .	243
12.3.5.1	Berechnung der Fehlerwahrscheinlichkeit . . . . .	246
12.3.5.2	Einteilung in Fehlerkategorien . . . . .	247
12.3.5.3	Fehlerwahrscheinlichkeiten im Doppelringbus . . . . .	249
12.3.6	Vergleich mit marktüblichen Feldbussystemen . . . . .	250
12.4	Zeitsynchronisierung auf Ringbussen . . . . .	252
12.4.1	Zeitmessung auf einem Doppelringbus . . . . .	252
12.4.1.1	Berechnung der Telegrammlaufzeiten im Doppelringbus . . . . .	253
12.4.1.2	Synchronisationszeitpunkt zur Zeitmessung . . . . .	255
12.4.2	Zeitmessung auf einem Einzelringbus . . . . .	255
12.4.2.1	Bidirektionaler Datentransfer . . . . .	256
12.4.2.2	Berechnung der Telegrammlaufzeiten im Einzelringbus . . . . .	256
12.4.3	Bearbeitung zeitabhängiger Aufträge . . . . .	258
12.4.4	Synchronisierung der Slave-Bausteine . . . . .	259
12.5	Doppelringbus . . . . .	263
12.5.1	Aufbau . . . . .	263
12.5.2	Steuerung . . . . .	263
12.5.3	Kommunikationskanäle . . . . .	265
12.6	Summenrahmentelegramm . . . . .	267
12.6.1	Beschreibung des Summenrahmentelegramms . . . . .	267
12.6.2	Signalcodierung bestimmter Telegrammabschnitte . . . . .	269
12.6.3	Aufbau der Slave-Bausteine . . . . .	269

12.6.4	Synchronisation der Slave-Bausteine . . . . .	270
12.6.5	Prioritätssteuerung der Slave-Bausteine . . . . .	271
12.6.6	Übertragungszeit des Summenrahmentelegramms . . . . .	272
12.6.7	Vergleich mit marktüblichen Feldbussystemen . . . . .	273
<b>13</b>	<b>Echtzeitbetriebssysteme</b>	<b>279</b>
13.1	Anwenderorientierte Erwägungen . . . . .	280
13.1.1	Allgemeine Anforderungen an Echtzeitbetriebssysteme . . . . .	280
13.1.2	Gestaltungsrichtlinien für und spezielle Forderungen an Echtzeitbetriebssysteme . . . . .	282
13.1.2.1	Anforderungen an den Leistungsumfang . . . . .	282
13.1.2.2	Architektur und Konfigurierbarkeit . . . . .	283
13.1.2.3	Dokumentation . . . . .	284
13.1.2.4	Vertragsrechtliche Anforderungen . . . . .	284
13.1.2.5	Anforderungen an die Dateiverwaltung . . . . .	284
13.1.2.6	Anforderungen an die Ein- und Ausgabe . . . . .	285
13.1.2.7	Anforderungen an die Fehlerbehandlung . . . . .	285
13.1.2.8	Anforderungen an die Überwachungsfunktionen . . . . .	285
13.1.2.9	Zusätzliche Anforderungen bei Mehrprozessorsystemen . . . . .	286
13.1.3	Technische Entwicklungen . . . . .	286
13.1.4	Wirtschaftlichkeit der Programmentwicklung . . . . .	287
13.1.5	Wirtschaftlichkeit der Programmwartung . . . . .	290
13.1.6	Struktur eines portierbaren Echtzeitbetriebssystems . . . . .	291
13.2	Termingesteuerte Prozessorzuteilung . . . . .	293
13.2.1	Prioritätensteuerung . . . . .	293
13.2.2	Prinzip antwortzeitgesteuerter Prozessorzuteilung . . . . .	293
13.2.3	Vergleich prioritäten- und antwortzeitgesteuerter Prozessorzuteilung . . . . .	295
13.2.4	Vorteile der antwortzeitgesteuerten Prozessorzuteilung . . . . .	298
13.2.5	Erweiterte Task-Steuerung . . . . .	298
13.3	Zeitgerechte Prozessorzuteilung in harten Echtzeitumgebungen . . . . .	299
13.3.1	Modell . . . . .	299
13.3.2	Zeitgerechte Verarbeitbarkeit . . . . .	300
13.3.2.1	Prozessmengen . . . . .	300
13.3.2.2	Präzedenzsysteme . . . . .	301
13.3.3	Zeitgerechte Strategien . . . . .	302
13.3.3.1	Kriterien für zeitgerechte Strategien . . . . .	302
	Gültigkeit der zeitgerechten Verarbeitbarkeit in der Verarbeitungsspanne . . . . .	302
	Ablaufmatrix . . . . .	302
13.3.3.2	Strategie der nächsten ablaufenden Antwortzeit . . . . .	305
	Einprozessorsysteme . . . . .	305
	Mehrprozessorsysteme . . . . .	306
13.3.3.3	Strategie des minimalen Spielraumes . . . . .	309
	Einprozessorsysteme . . . . .	309
	Mehrprozessorsysteme . . . . .	309
13.3.3.4	Vorhaltestrategie . . . . .	310
13.3.3.5	Strategie der präemptiven Antwortzeit . . . . .	311
13.3.4	Zusammenfassung . . . . .	313
13.4	Sprachunabhängige Zuteilbarkeitsanalyse task-basierter Echtzeitprogramme . . . . .	313
13.4.1	Einführung . . . . .	313
13.4.2	Anforderungen an Echtzeitsysteme . . . . .	314

---

13.4.2.1	Programmbezogene Annahmen . . . . .	314
13.4.2.2	Gerätebezogene Annahmen . . . . .	315
13.4.3	Verfahren der Zuteilbarkeitsanalyse . . . . .	316
13.4.3.1	Erste Phase der Zuteilbarkeitsanalyse . . . . .	316
	Bedingungen, Klammern, Unterprogramme und Prozesse . . .	319
	Ergebnisse der ersten Phase . . . . .	319
13.4.3.2	Zweite Phase der Zuteilbarkeitsanalyse . . . . .	321
	Auflösung der Segmentbäume . . . . .	321
	Umwandlung zu Prozeßbäumen . . . . .	321
	Ein Echtzeitmodell . . . . .	323
	Die Zuteilbarkeitsbedingung . . . . .	323
	Rahmenüberlagerung . . . . .	324
	Formale Beschreibung der Rahmenüberlagerung . . . . .	325
	Ein Beispiel . . . . .	327
	Verzögerungen . . . . .	329
	Unterbrechbare Verzögerungen . . . . .	330
	Algorithmus zur Gesamtlösung . . . . .	330
13.4.4	Ergänzende Informationen . . . . .	332
<b>14</b>	<b>Methoden formaler Spezifikation und Verifikation</b>	<b>335</b>
14.1	Einführung . . . . .	336
14.2	Sequentielle und parallele Beschreibung . . . . .	336
14.3	Petri-Netze . . . . .	352
14.4	Eigenschaften von Petri-Netzen . . . . .	371
14.5	Temporale Logik . . . . .	382
14.6	Korrektheitsnachweis mittels temporaler Logik . . . . .	386
	<b>Literaturverzeichnis</b>	<b>391</b>
	<b>A Anhang</b>	<b>397</b>