

eXamen.press

eXamen.press ist eine Reihe, die Theorie und Praxis aus allen Bereichen der Informatik für die Hochschulausbildung vermittelt.

Peter Marwedel

Eingebettete Systeme

Übersetzt aus dem Englischen von Lars Wehmeyer

Peter Marwedel
Fakultät für Informatik
Technische Universität Dortmund
Otto-Hahn-Str. 16
44221 Dortmund
peter.marwedel@tu-dortmund.de

Übersetzer

Lars Wehmeyer

korrigierter Nachdruck 2008

ISBN 978-3-540-34048-5

e-ISBN 978-3-540-34049-2

DOI 10.1007/978-3-540-34049-2

ISSN 1614-5216

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2008 Springer-Verlag Berlin Heidelberg

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandgestaltung: KünkelLopka Werbeagentur, Heidelberg

Gedruckt auf säurefreiem Papier

9 8 7 6 5 4 3 2 1

springer.com

Dieses Buch ist meiner Familie gewidmet.

Vorwort

Relevanz eingebetteter Systeme

Eingebettete Systeme können definiert werden als informationsverarbeitende Systeme, die in ein umgebendes Produkt wie z.B. Autos, Telekommunikationsgeräte oder Produktionsmaschinen eingebettet sind. Solche Systeme haben viele gemeinsame Charakteristiken, u.a. das Einhalten von Zeitbedingungen, Zuverlässigkeit und Effizienz. Die Technologie eingebetteter Systeme ist eine Grundvoraussetzung, um das Paradigma der allgegenwärtigen Information (*ubiquitous information*) zu gewährleisten – eines der Hauptziele der modernen Informationstechnik (IT).

Nach dem Erfolg der IT im Bereich der Büroanwendungen und Geschäftsprozesse werden eingebettete Systeme in den kommenden Jahren **das** wichtigste Anwendungsgebiet für Informationstechnik sein. Aufgrund dieser Erwartungen wurde bereits der Begriff **Post-PC Era** geprägt. Dieser Begriff drückt aus, dass Standard-PCs in Zukunft weniger dominant sein werden. Prozessoren und Software werden zunehmend in viel kleineren Systemen verwendet werden, und in vielen Fällen werden sie sogar völlig unsichtbar sein, was zum Begriff des **verschwindenden Computers** geführt hat. Es ist heute bereits offensichtlich, dass technische Produkte sehr hochentwickelt sein müssen, um das Interesse der Kunden zu wecken. Autos, Kameras, Fernseher, Handys usw. lassen sich heute ohne clevere integrierte Softwarelösungen kaum noch verkaufen. Bereits heute übersteigt die Anzahl von Prozessoren in eingebetteten Systemen die Anzahl von Prozessoren, die in PCs verbaut sind – und dieser Trend wird sich weiter fortsetzen. Nach aktuellen Vorhersagen wird die Größe und Komplexität eingebetteter Software in den kommenden Jahren sehr stark wachsen. Es gibt auf diesem Gebiet eine neue Form des „Moore'schen Gesetzes“: *In vielen Produkten im Bereich der Konsumelektronik verdoppelt sich die Größe der Software alle zwei Jahre* [Vaandrager, 1998].

Diese Relevanz eingebetteter Systeme wird in den aktuellen Studienplänen nicht ausreichend berücksichtigt. Dieses Buch soll dazu beitragen, die Situation zu verbessern. Es stellt Materialien für einen einführenden Kurs zu eingebetteten Systemen zur Verfügung, kann aber auch außerhalb von Vorlesungen eingesetzt werden.

Zielgruppe dieses Buches

Dieses Buch wurde für die folgenden Leser konzipiert:

- Informatik-, Ingenieur-Informatik- und Elektrotechnik-Studenten, die sich im Bereich der eingebetteten Systeme spezialisieren wollen. Das Buch ist geeignet für Studierende ab dem dritten Studienjahr, die bereits ein grundlegendes Verständnis von Rechner-Hard- und Software haben. Es ebnet den Weg zu fortgeschrittenen Themen, die in Folgekursen vertieft werden sollten.
- Ingenieure, die bislang im Bereich der Hardware für Rechensysteme gearbeitet haben und die sich intensiver mit dem Bereich der Software für eingebettete Systeme beschäftigen wollen. Dieses Buch sollte ein ausreichendes Hintergrundwissen zur Verfügung stellen, um aktuelle relevante Publikationen auf dem Gebiet zu verstehen.
- Professoren, die einen Studiengang oder eine Studienrichtung für eingebettete Systeme konzipieren.

Eingliederung eingebetteter Systeme in einen Studienplan

Dieses Buch setzt Grundwissen in den folgenden Gebieten voraus (s. Abb. 0.1);

- Elektrische Schaltnetze auf Oberstufen-Niveau (z.B. Kirchhoffsche Gesetze),
- Operationsverstärker (optional),
- Rechner-Hardware, z.B. anhand des einführenden Buches von J.L. Hennessy und D.A. Patterson [Hennessy und Patterson, 1995],
- Grundwissen digitaler Schaltungen wie etwa Gatter und Register,
- Programmierung,
- Endliche Automaten,
- Fundamentale mathematische Konzepte wie Tupel, Integrale und lineare Gleichungssysteme, möglichst auch die Grundlagen der Wahrscheinlichkeitstheorie,

- Algorithmen (Graphenalgorithmen und Optimierungsalgorithmen wie z.B. *Branch-and-Bound*),
- NP-Vollständigkeit und Berechenbarkeit.

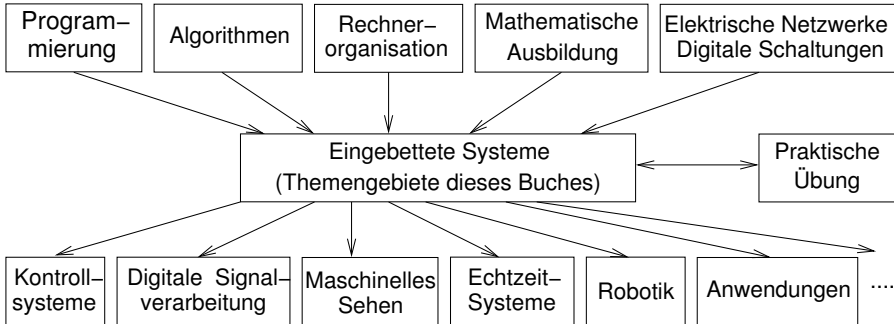


Abb. 0.1. Positionierung der Themen dieses Buches

Eines der Hauptziele dieses Buches ist es, einen Überblick über den Entwurf eingebetteter Systeme zu geben und die wichtigsten Themen im Bereich der eingebetteten Systeme zueinander in Beziehung zu setzen. Es soll sowohl Studierende als auch Lehrende dazu motivieren, sich weitere Themen zu erarbeiten. Einige Themen werden in diesem Buch erschöpfend behandelt, andere nur kurz angerissen. Diese kürzeren Abschnitte sollen helfen, verwandte Themengebiete in den Gesamtkontext einordnen zu können. Dadurch können Lehrende entsprechend ihrer eigenen Vorlieben Zusatzmaterial in den Kurs einbauen. Das Buch sollte durch weiterführende Kurse ergänzt werden, die vertiefendes Wissen in den folgenden Themengebieten vermitteln:

- Digitale Signalverarbeitung,
- Robotik,
- Maschinelles Sehen,
- Sensoren und Aktuatoren,
- Echtzeitsysteme, Echtzeitbetriebssysteme und Echtzeit-Scheduling,
- Regelungstechnik,
- Spezifikationssprachen für eingebettete Systeme,
- Rechnergestützte Entwurfsumgebungen für applikationsspezifische Hardware,
- Formale Verifikation von Hardware,
- Systematisches Testen von Hardware und Software,
- Leistungsbewertung von Rechensystemen,

- Entwurfsmethoden für energiesparende Systeme,
- Sicherheit und Verlässlichkeit von Computersystemen,
- *Ubiquitous Computing* (allgegenwärtiges Rechnen),
- Anwendungsgebiete in der Telekommunikation, im Automobilbereich, in medizinischen Geräten und intelligenten Häusern,
- Konsequenzen des Einsatzes eingebetteter Systeme.

Ein auf diesem Buch aufbauender Kurs sollte durch eine Übung vervollständigt werden, in der beispielsweise programmierbare Roboter wie Lego MindstormsTM oder ähnliche verwendet werden. Außerdem sollten Studierende praktische Erfahrungen mit StateCharts-basierten Werkzeugen machen können.

Zusatzinformationen zu diesem Buch sind von der folgenden Webseite erhältlich:

<http://ls12-www.cs.tu-dortmund.de/~marwedel/es-book>

Diese Seite beinhaltet Verweise zu Präsentationsfolien, Übungen, Hinweise zum Übungsbetrieb, Referenzen zu einer Auswahl von aktuellen Publikationen sowie Fehlerkorrekturen. Leser, die Fehler entdeckt haben oder die Vorschläge zur Verbesserung des Buches beitragen möchte, wenden sich bitte per E-Mail an Peter.Marwedel@tu-dortmund.de.

Übungen können auch anhand von Informationen aus der folgenden Auswahl von Büchern konzipiert werden: [Ganssle, 1992], [Ball, 1996], [Ball, 1998], [Barr, 1999], [Ganssle, 2000], [Wolf, 2001], [Buttazzo, 2002].

Die Verwendung von Namen in diesem Buch ohne Hinweise auf Urheberrechte oder eingetragene Warenzeichen bedeutet nicht, dass diese Namen nicht evtl. durch Dritte geschützt sind.

Viel Spaß beim Lesen dieses Buches!

Dortmund, September 2003

P. Marwedel

Herzlich willkommen zur aktualisierten Version dieses Buches! Die Verschmelzung des Kluwer- und Springer-Verlags macht es möglich, weniger als zwei Jahre nach dem Erscheinen der Erstausgabe 2003 eine neue Version dieses Buches herauszubringen. In dieser aktualisierten Version wurden alle gefundenen Tipp- und sonstigen Fehler der ersten Ausgabe korrigiert. Außerdem wurden alle Internet-Referenzen überprüft und aktualisiert. Abgesehen von diesen Änderungen ist der Inhalt des Buches gleich geblieben. Eine Liste von korrigierten Fehlern ist auf der oben genannten Webseite verfügbar.

Viel Spaß beim Lesen dieses aktualisierten Buches!

Dortmund, August 2005

P. Marwedel

Die vorliegende deutsche Fassung des Buches basiert auf der englischen Version des Buches aus dem Jahr 2005. Gegenüber dem englischen Original wurden mehrere Themen etwas ergänzt. Insbesondere wurde die Beschreibung von Scheduling-Verfahren um Beweise ergänzt, die Darstellung von Berechnungsmodellen (*models of computation*) überarbeitet und das Kapitel über Validierung zu einem Kapitel über Evaluation und Validierung erweitert. Dennoch sollten in der Regel die deutsche und die englische Fassung nebeneinander benutzt werden können.

Dortmund, Januar 2007

P. Marwedel (Autor des englischen Originals)

L. Wehmeyer (Übersetzer der deutschen Ausgabe)

Die vorliegende korrigierte Fassung der ersten Auflage der deutschen Ausgabe unterscheidet sich von der ursprünglichen Fassung der ersten Auflage nur durch die Beseitigung von Druckfehlern und die Überprüfung aller Referenzen auf das *World Wide Web*. Fehlermeldungen von Melanie Schmidt, Jan-Philipp Kappmeier und Matthias Steinkamp wurden berücksichtigt. Reinhard von Hanxleden (Universität Kiel) schlug neue Formulierungen im Abschnitt über Esterel vor. Beide Fassungen können nebeneinander benutzt werden.

Dortmund, April 2008

P. Marwedel

Danksagung

Meine Doktoranden, besonders Lars Wehmeyer, haben eine Vorabversion des Buches gründlich Korrektur gelesen. Die Teilnehmer des Kurses „Introduction to Embedded Systems“ im Sommer 2003, insbesondere Lars Bensmann, haben ebenfalls zum Buch beigetragen. Außerdem haben die folgenden Kollegen und Studierenden Kommentare und Hinweise gegeben, die in das Buch eingearbeitet wurden: W. Müller, F. Rammig (U. Paderborn), W. Rosenstiel (U. Tübingen), R. Dömer (UC Irvine) und W. Kluge (U. Kiel). Zur Vorbereitung und Zusammenstellung des Buches wurden Materialien der folgenden Personen verwendet: G. C. Buttazzo, D. Gajski, R. Gupta, J. P. Hayes, H. Kopetz, R. Leupers, R. Niemann, W. Rosenstiel und H. Takada. Korrekturen zur ersten Ausgabe dieses Buches wurden beigetragen von David Hec, Thomas Wiederkehr und Thorsten Wilmer. Eine Vorabversion der deutschen Fassung des Buches wurde von Martin Faßbach, Fatih Gedikli, Gordon Schlechter, Karl-Heinz Temme und Matthias Wiedenhorst Korrektur gelesen. Studierende des Kurses des Wintersemesters 2007/2008 haben ebenfalls zu den Korrekturen beigetragen. Selbstverständlich ist der Autor selbst verantwortlich für alle verbleibenden Fehler.

Dank gilt all denen, die die erhöhte Arbeitsbelastung des Autors während der Entstehung dieses Buches geduldig akzeptiert haben, insbesondere die dadurch oft fehlende Zeit sowohl für berufliche als auch persönliche Partner.

Schließlich sollte noch erwähnt werden, dass Kluwer Academic Publishers die Veröffentlichung dieses Buches vom ersten Konzept an unterstützt hat. Durch die Unterstützung des Verlags wurde die Arbeit an diesem Buch vorangetrieben.

Nach der Verschmelzung von Kluwer mit dem Springer-Verlag hat der Springer-Verlag großes Interesse an der Veröffentlichung einer deutschen Ausgabe gezeigt.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Begriffe und Abgrenzung	1
1.2	Anwendungsgebiete	6
1.3	Wachsende Relevanz von eingebetteten Systemen	8
1.4	Struktur dieses Buches	10
2	Spezifikationssprachen	13
2.1	Anforderungen	13
2.2	Berechnungsmodelle	17
2.3	StateCharts	18
2.3.1	Modellierung von Hierarchie	19
2.3.2	Zeitbedingungen	24
2.3.3	Kantenbeschriftungen und StateMate-Semantik	25
2.3.4	Bewertung und Erweiterungen	28
2.4	Allgemeine Spracheigenschaften	29
2.4.1	Synchrone und asynchrone Sprachen	29
2.4.2	Prozess-Konzepte	30
2.4.3	Synchronisation und Kommunkation	30
2.4.4	Spezifikation von Zeitbedingungen	31
2.4.5	Nicht-Standard-Ein-/Ausgabegeräte	32
2.5	SDL	32
2.6	Petrinetze	39
2.6.1	Einführung	39
2.6.2	Bedingungs-/Ereignisnetze	42
2.6.3	Stellen-/Transitionen-Netze	43

2.6.4	Prädikat-/Ereignis-Netze	47
2.6.5	Bewertung	49
2.7	<i>Message Sequence Charts</i>	49
2.8	UML	51
2.9	Prozessnetze	55
2.9.1	Taskgraphen	55
2.9.2	Asynchroner Nachrichtenaustausch	58
2.9.3	Synchroner Nachrichtenaustausch	60
2.10	Java	63
2.11	VHDL	64
2.11.1	Einführung	64
2.11.2	<i>Entities</i> und <i>Architectures</i>	66
2.11.3	Mehrwertige Logik und IEEE 1164	68
2.11.4	VHDL-Prozesse und Simulations-Semantik	75
2.12	SystemC	80
2.13	Verilog und SystemVerilog	81
2.14	SpecC	83
2.15	Weitere Sprachen	84
2.16	Ebenen der Hardware-Modellierung	87
2.17	Vergleich der Sprachen	90
2.18	Verlässlichkeitsanforderungen	92
3	Hardware eingebetteter Systeme	95
3.1	Einführung	95
3.2	Eingabe	96
3.2.1	Sensoren	96
3.2.2	Sample-and-Hold-Schaltungen	99
3.2.3	A/D-Wandler	100
3.3	Kommunikation	102
3.3.1	Anforderungen	103
3.3.2	Elektrische Robustheit	104
3.3.3	Garantieren von Echtzeitverhalten	105
3.3.4	Beispiele	106
3.4	Verarbeitungseinheiten	108

3.4.1	Überblick	108
3.4.2	Anwendungsspezifische integrierte Schaltkreise (ASICs)	109
3.4.3	Prozessoren	110
3.4.4	Rekonfigurierbare Logik	126
3.5	Speicher	129
3.6	Ausgabe	131
3.6.1	D/A-Wandler	132
3.6.2	Aktuatoren	133
4	Eingebettete Betriebssysteme, <i>Middleware</i> und Scheduling	135
4.1	Schranken von Ausführungszeiten	136
4.2	Scheduling in Echtzeitsystemen	137
4.2.1	Klassifikation von Scheduling-Algorithmen	138
4.2.2	Aperiodisches Scheduling	141
4.2.3	Periodisches Scheduling	149
4.2.4	Ressourcen-Zugriffs-Protokolle	155
4.3	Eingebettete Betriebssysteme	158
4.3.1	Allgemeine Anforderungen	158
4.3.2	Echtzeitbetriebssysteme	160
4.4	<i>Middleware</i>	164
4.4.1	Echtzeit-Datenbanken	164
4.4.2	Zugriff auf entfernte Objekte	165
5	Implementierung eingebetteter Systeme:	
	Hardware-/Software-Codesign	169
5.1	Organisation der Nebenläufigkeit auf Task-Ebene	172
5.2	<i>High-Level</i> -Optimierungen	176
5.2.1	Umwandlung von Gleitkomma- in Festkomma-Darstellung	176
5.2.2	Einfache Schleifentransformationen	177
5.2.3	Kachel-/Blockweise Verarbeitung von Schleifen	179
5.2.4	Aufteilen von Schleifen	182
5.2.5	Falten von Feldern	184
5.3	Hardware-/Software-Partitionierung	186
5.3.1	Einleitung	186

5.3.2	COOL	187
5.4	Compiler für eingebettete Systeme	197
5.4.1	Einführung	197
5.4.2	Energieoptimierende Compiler	198
5.4.3	Compiler für digitale Signalverarbeitung	202
5.4.4	Compiler für Multimedia-Prozessoren	205
5.4.5	Compiler für VLIW-Prozessoren	205
5.4.6	Compiler für Netzwerkprozessoren	206
5.4.7	Compiler-Erzeugung, retargierbare Compiler und Untersuchung des Entwurfsraums	207
5.5	Versorgungsspannungs-Anpassung und Energie-Management ..	207
5.5.1	Dynamische Anpassung der Versorgungsspannung	207
5.5.2	Dynamisches Power-Management	211
6	Evaluierung und Validierung	213
6.1	Einleitung	213
6.2	Pareto-Optimalität	215
6.3	Simulation	217
6.4	Rapid Prototyping und Emulation	218
6.5	Leistungsbewertung	219
6.6	Bewertung des Energieverbrauchs	226
6.7	Risiko- und Verlässlichkeits-Analyse	227
6.8	Formale Verifikation	233
6.9	Testen	236
6.9.1	Betrachteter Bereich	236
6.9.2	Testfreundlicher Entwurf	237
6.9.3	Selbsttestprogramme	240
6.10	Fehlersimulation	241
6.11	Fehlerinjektion	242
	Literaturverzeichnis	245
	Sachverzeichnis	259