Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Peter Scholz

# Softwareentwicklung eingebetteter Systeme

## Grundlagen, Modellierung, Qualitätssicherung

Mit 30 Abbildungen

Springer

Peter Scholz
Fachhochschule Landshut
Fachbereich Informatik
Am Lurzenhof 1
84036 Landshut
peter.scholz@fh-landshut.de

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
http://dnb.ddb.de abrufbar.

ISSN 1439-5428
ISBN 3-540-23405-5 Springer Berlin Heidelberg New York



Springer ist ein Unternehmen von Springer Science+Business Media
springer.de



Satz: Druckfertige Daten der Autoren
Herstellung: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

# Vorwort

*„Man darf das Schiff nicht an einen einzigen Anker*
*und das Leben nicht an eine einzige Hoffnung binden!“*

(Epiket, griechischer Philosoph, ca. 50 bis 138)

Eingebettete Systeme (engl. embedded systems) sind Computersysteme, die aus Hardware und Software bestehen und die in komplexe technische Umgebungen eingebettet sind. Solche Umgebungen können maschinelle Systeme wie etwa Kraftfahrzeuge, Flugzeuge, Fernsehgeräte, Waschmaschinen, Küchengeräte, Mobilfunktelefone u. a. sein, die der Interaktion eines menschlichen Benutzers bedürfen (z. B. Steer-by-Wire im Kraftfahrzeug) oder vollautomatisch agieren (z. B. Antiblockiersystem). Teilweise ist dieser Übergang auch fließend, so beispielsweise bei einem elektronischen Bremsassistenten.

Schon heute hat z. B. ein Mittelklassewagen Elektrik und Elektronik im Wert von rund 2.200 Euro an Bord. In zehn Jahren wird sich dieser Wert voraussichtlich auf ca. 4.200 Euro fast verdoppelt haben. Eingesetzt wird Informationstechnologie vor allem in den Bereichen Fahrwerksantrieb, Motormanagement, Sicherheit, Komfort, Emissionsreduzierung und Kommunikation/Entertainment bzw. Infotainment.

Eingebettete Systeme übernehmen komplexe Steuerungs-, Regelungs- und Datenverarbeitungsaufgaben für (bzw. in) diese(n) technischen Systeme(n). Ihre Funktionalität wird durch das Zusammenspiel von Spezialhardware, Standardprozessoren, Peripherie und Software realisiert.

Sie unterscheiden sich daher von anderen Systemen, wie beispielsweise Textverarbeitungs-, Buchhaltungs-, Internet- oder Warenwirtschafssystemen grundlegend und in mannigfaltiger Weise: Diese Systeme sind ausschließlich in Software realisiert, arbeiten

ggf. durch Interaktion mit einem menschlichen Benutzer vor einer Tastatur bestimmte Aufgaben ab, d. h. sie transformieren in einer ersten Näherung Eingaben in Ausgaben: Benutzereingaben werden in einem iterativen Prozess gelesen, bearbeitet und schließlich in Ausgaben transformiert. Bei ihrer Beschreibung liegt der Schwerpunkt auf den Transformationsprozessen bzw. -algorithmen selbst. Man spricht hier auch von „(rein) transformationellen" oder „interaktiven" Systemen.

Dem gegenüber stehen die reaktiven Systeme, die als Verallgemeinerung von eingebetteten Systemen fortwährend auf Eingaben ihrer technischen Umgebung bzw. eines technischen Prozesses reagieren und entsprechende, Kontext-abhängige Ausgaben erzeugen. Eingebettete, reaktive Systeme sind in eine – möglicherweise recht feindliche (Schmutz, Hitze, Kälte, schnelle Bewegung etc.) – Umgebung eingebettet. Bei ihnen liegt der Schwerpunkt der Beschreibung daher auf der Beschreibung der Interaktion zwischen System und Umgebung und damit auf dem Ein-/Ausgabeverhalten des Systems. Reaktive Systeme müssen zu jedem Zeitpunkt in einer nicht vom Computersystem selbst getriebenen Weise auf Sensordaten der Umgebung reagieren können und unterliegen hierbei oft sogenannten Echtzeitanforderungen.

Bei Echtzeitanforderungen unterscheidet man zwischen harten und weichen. „Harte" Echtzeitanforderungen müssen zur Erfüllung der Funktion eingehalten werden (Beispiel: Ein Airbag muss innerhalb von wenigen hundertstel Sekunden voll aufgeblasen werden, sonst verfehlt er seine Wirkung). „Weiche" Echtzeitanforderungen dagegen erhöhen den Komfort, man spricht hier gerade im Automobilbau auch von „Komfortelektronik" (Beispiele sind elektrische Fensterheber, Zentralverriegelung usw.). Wird das Thema „Embedded Systems" aus dem Blickwinkel der Ingenieursdisziplinen betrachtet, so wird hier naturgemäß gerne der Schwerpunkt auf Hardware-Gesichtspunkte (Mikrocontroller, Signalprozessoren, Sensoren, Aktoren, Analog-Digital-Wandler usw.) gelegt. Daraus könnte leicht der falsche Eindruck erwachsen, bei der Entwicklung eingebetteter Systeme handele es sich um eine reine Hardwareentwicklungsaufgabe. Dies ist aber mitnichten so. Vielmehr handelt es sich beim Entwurf eingebetteter Systeme um eine mindestens genauso wichtige Softwareentwurfsaufgabe. Gerade letztere ist Kern regen wissenschaftlichen Interesses (Rosenstiel, 2003), was es in überschaubarer Zukunft wohl auch noch bleiben wird. Ein aktueller Wegweiser für die Forschung und Lehre für das Software Engineering im Bereich eingebetteter Systeme findet sich in (Broy und Pree, 2003).

In diesem Buch wollen wir einen ersten Überblick über das Thema geben. Wir starten dabei nach einer ausführlichen Einleitung und Hinführung zum Thema mit der Diskussion von nebenläufigen Systemen. Danach widmen wir uns den Gebieten Echtzeit, Echtzeitsysteme und Echtzeitbetriebssysteme. Im Anschluss werden wir dann einen Überblick zur Entwicklung eingebetteter Systeme geben. An eingebettete Systeme werden oft Echtzeitanforderungen gestellt. Ein Echtzeitsystem ist ein System, bei dem der Zeitpunkt, zu dem Ausgaben erzeugt werden, bedeutend ist. Programme, die auf einer (fast) beliebigen Hardware ablaufen, die Grundfunktionen von Betriebssystemen erfüllen und Echtzeitverhalten aufweisen, nennt man Echtzeitbetriebssysteme. Wir beginnen daher mit einer Beschreibung von Echtzeitbetriebssystemen und widmen uns dann in den folgenden Kapiteln der Programmierung von eingebetteten Systemen, bevor wir auf ausgewählte Techniken zum Softwareentwurf für diese Systeme eingehen. Da es sich bei eingebetteten Systemen oft um sicherheitskritische Systeme handelt, deren Fehlfunktion ihre Umgebung massiv beeinträchtigen kann und letztendlich sogar zur Gefährdung von Menschenleben führen kann, ist die Qualität solcher Systeme von zentraler Bedeutung. Dieses Buch enthält daher ebenfalls eine überblicksartige Betrachtung des Themas Softwarequalität und schließt mit einer Zusammenfassung verschiedener für Embedded Systeme geeigneter Vorgehensmodelle.

Ein kompaktes Buch wie das vorliegende kann naturgemäß ein derart komplexes und umfangreiches Thema wie die Softwareentwicklung eingebetteter Systeme nicht auch nur annähernd erschöpfend in allen Details behandeln. Dieser Anspruch wird demnach selbstverständlich gar nicht erst erhoben. Vielmehr soll es einen Zugang zu diesem – gerade für die deutsche Softwareindustrie in den kommenden Jahren wohl sehr zentralen – Thema schaffen und „Lust auf mehr“ generieren. Ein besonderes aber nicht ausschließliches Augenmerk gilt dabei der automobilen Softwareentwicklung.

Bei der Auswahl der Inhalte habe ich mich dabei in erster Linie davon leiten lassen, wo ich vor allem aufgrund meines persönlichen Hintergrunds aus Lehre, anwendungsnaher Forschung und Praxiserfahrung aus Industrietätigkeiten Handlungsbedarf in den softwareerstellenden Unternehmen sehe. Viele dieser Inhalte konnte ich mit zahlreichen Teilnehmern aus meiner Weiterbildungsreihe „IT Update“ für Fach- und Führungskräfte persönlich und ausführlich besprechen. Insbesondere fanden dabei von mir angebotene Tagesseminare zu Themenkomplexen wie „Software Engineering“,

„Softwarequalität“ und „Software Engineering eingebetteter Systeme“ großen Zuspruch. Inhaltlich geht das Buch an jenen Stellen in die Tiefe, wo ich vor allem in den industriellen Forschungs- und Entwicklungsbereichen Entwicklungspotential sehe. Mit meinem Buch möchte ich daher den Praktiker genauso ansprechen wie Studenten der Informatik, Elektrotechnik oder Mechatronik im Hauptstudium, die erstmals Zugang zu diesem Thema suchen.

Eingebettete Systeme sind eine der schnellstwachsenden Branchen unter den Informatikanwendungen. In Zukunft darf sogar eher noch mit einer Zunahme dieses Ungleichgewichts gerechnet werden. Das Buch zeigt auch deutlich auf, wo in Zukunft interessante Aufgabengebiete und berufliche Chancen für Informatiker und Informationstechniker liegen werden. Die Darstellungsweise der Inhalte orientiert sich dabei gezielt an der Sprachwelt der Informatik.

Dieses Buch wurde ganz bewusst in Deutsch verfasst, ist aber stets bemüht, englische Fachbegriffe weitestgehend ebenfalls einzuführen. Da viele der tangierten Themenbereiche überwiegend auf einer englischsprachigen Terminologie basieren, wird gar nicht erst der Versuch unternommen, Anglizismen zu vermeiden.

Lesehinweis: Nach der Lektüre der Kapitel 1 bis 3 können die nachfolgenden Kapitel in beliebiger Reihenfolge gelesen werden.

Die Erstellung dieses Werkes wäre niemals ohne die tatkräftige Unterstützung Anderer möglich gewesen. Zunächst möchte ich meinen Studenten des Studienprojekts „Embedded Systems“ aus dem Jahre 2004 (den Herren Philipp Konradi, Matthias Ecker, Christian Könik, André Hofmann und Florian Kandlinger) am Fachbereich für Informatik der Fachhochschule Landshut danken, die mit ihrer Literaturrecherche wichtige flankierende Arbeiten geleistet haben. Mit den Ergebnissen aus ihrem, von mir betreuten Studienprojekt konnten sie den ersten Preis bei den „Audi IT Tagen“ im Herbst 2004 gewinnen, was mich letztendlich zum Verfassen dieses Buches beflügelt hat. Besonders bedanken möchte ich mich auch bei Frau Stephanie Hahn für die Übernahme des Erstlektorates. Weiterhin gilt mein spezieller Dank Frau Jutta Maria Fleschutz, Frau Gabi Fischer und Frau Dorothea Glaunsinger vom Springer-Verlag, die bei redaktionellen Fragen stets mit Rat und Tat zur Seite standen.

Ich widme dieses Buch in Dankbarkeit meinen Eltern.

Peter Scholz Februar 2005

# Inhaltsverzeichnis


**1 Einleitung ..... 1**

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

**2 Nebenläufige Systeme ..... 27**

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

**3 Echtzeit, Echtzeitsysteme, Echtzeitbetriebssysteme ..... 39**

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

**4 Programmierung eingebetteter Systeme ..... 75**

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

**5 Softwareentwurf eingebetteter Systeme.....139**

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

**6 Softwarequalität eingebetteter Systeme.....173**

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

**7 Vorgehensmodelle und Standards der Entwicklung ..... 201**

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

**8 Schlussbemerkungen ..... 219**

**Literaturverzeichnis ..... 223**

**Sachverzeichnis ..... 229**