Embedded Systems

For other titles published in this series, go to
www.springer.com/series/8563

Peter Marwedel

Embedded System Design

Embedded Systems Foundations of Cyber-Physical Systems

2nd Edition

Springer

Dr. Peter Marwedel
TU Dortmund
Informatik 12
Otto-Hahn-Str. 16
44221 Dortmund
Germany
peter.marwedel@tu-dortmund.de

ISBN 978-94-007-0256-1 e-ISBN 978-94-007-0257-8
DOI 10.1007/978-94-007-0257-8
Springer Dordrecht Heidelberg London New York

Cover design: VTEX, Vilnius

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Preface

Definitions and scope

Until the late 1980s, information processing was associated with large mainframe computers and huge tape drives. During the 1990s, this shifted towards information processing being associated with personal computers, PCs. The trend towards miniaturization continues and the majority of information processing devices will be small portable computers, many of which will be integrated into larger products. Their presence in these larger products, such as telecommunication equipment, will be less obvious than for the PC. Usually, technical products must be technologically advanced to attract customers' interest. Cars, cameras, TV sets, mobile phones, etc. can hardly be sold any more in technologically advanced countries unless they come with built-in computers. Hence and according to several forecasts (see, for example [National Research Council, 2001]), the future of information and communication technologies (ICT) is characterized by terms such as

1 ubiquitous computing [Weiser, 2003],

2 pervasive computing [Hansmann, 2001], [Burkhardt, 2001],

3 ambient intelligence [Koninklijke Philips Electronics N.V., 2003], [Marzano and Aarts, 2003],

4 the disappearing computer [Weiser, 2003],

5 and the post-PC era.

The first term reflects the fact that computing (and communication) will be everywhere. The expectation is that *information* will be available *anytime, anywhere*. The predicted penetration of our day-to-day life with computing

devices led to the term "pervasive computing". For ambient intelligence, there is some emphasis on communication technology in future homes and smart buildings. These three terms focus on only slightly different aspects of future information technology. Ubiquitous computing focuses more on the long term goal of providing information anytime, anywhere, whereas pervasive computing focuses more on practical aspects and the exploitation of already available technology. The fourth term refers to the expectation that processors and software will be used in much smaller systems and will in many cases even be invisible. The term **post-PC era** denotes the fact that in the future, standard-PCs will be less dominant hardware platforms.

Two basic technologies are needed for next-generation ICT systems:

- **embedded systems**,
- and **communication technologies**.

Fig. 0.1 shows a graphical representation of how ubiquitous computing is influenced by embedded systems and by communication technology.

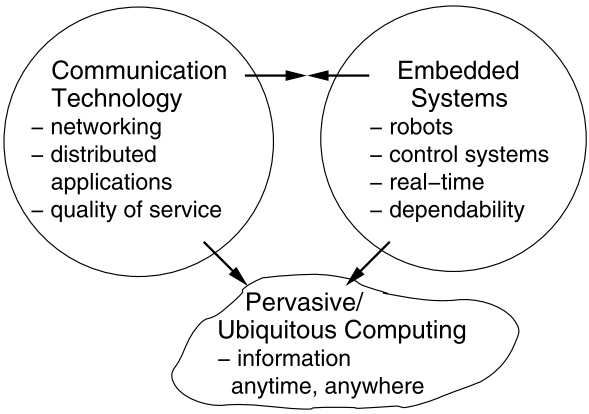

Figure 0.1. Influence of embedded systems on ubiquitous computing (©European Commission)

For example, ubiquitous computing devices -like embedded systems- must meet real-time and dependability requirements of embedded systems while using fundamental techniques of communication technology, such as networking.

A comprehensive coverage of communication technologies would require a separate book. Therefore, this book does not cover communication technologies, except as a minor topic in few subsections. What are "embedded systems" anyway? They can be defined as follows [Marwedel, 2003]:

Definition: Embedded systems are information processing systems embedded into enclosing products.

Examples include embedded systems in cars, trains, planes, and telecommunication or fabrication equipment. Such systems come with a large number of common characteristics, including real-time constraints, and dependability as well as efficiency requirements. For such systems, the link to physics and physical systems is rather important. This link is emphasized in the following citation [Lee, 2006]:

"Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems".

This citation could be used as a definition of the term "embedded software" and could be extended into a definition of "embedded systems" by just replacing "software" by "system". However, the strong link to physics has recently been stressed even more by the introduction of the term "cyber-physical systems" (CPS or "cy-phy" systems for short). Cy-phy systems can be defined as follows:

Definition: "*Cyber-Physical Systems (CPS) are integrations of computation and physical processes*" [Lee, 2007].

The new term emphasizes the link to physical quantities such as time, energy and space. Emphasizing this link makes a lot of sense, since it is frequently ignored in a world of applications running on PCs. For cy-phy systems, we may be expecting models to include models of the physical environment as well. **In this sense, we may think of cy-phy systems to comprise embedded systems (the information processing part) and the physical environment.** We will refer to the new term whenever we want to emphasize the link to physics and the environment. In the future, links to chemistry and biology are likely to be important as well.

This book provides an overview of key concepts for embedded systems as they are needed for cyber-physical systems. The scope includes specification techniques, hardware components, system software, application mapping, evaluation and validation, as well as exemplary optimizations and test methods.

Importance of embedded and cyber-physical systems

Following the success of ICT for office and work flow applications, embedded and cyber-physical systems are considered to be **the** most important application area of ICT during the coming years. The number of processors in embedded systems already exceeds the number of processors in PCs, and this trend is expected to continue. According to forecasts, the size of embedded software will also increase at a large rate. Another kind of Moore's law was predicted:

For many products in the area of consumer electronics the amount of code is doubling every two years [Vaandrager, 1998]. The increasing importance of embedded systems is also reflected in a report of the National Research Council in the United States [National Research Council, 2001]. According to the introduction of this report, "*Information technology (IT) is on the verge of another revolution. ... networked systems of embedded computers ... have the potential to change radically the way people interact with their environment by linking together a range of devices and sensors that will allow information to be collected, shared, and processed in unprecedented ways. ... The use ... throughout society could well dwarf previous milestones in the information revolution*."

Statistics regarding the size of the embedded systems market can be found on relevant web sites. Sites such as "IT facts" [IT Facts, 2010] demonstrate the importance of the embedded system market. The size of the embedded system market can be analyzed from a variety of perspectives. Many of the embedded processors are 8-bit processors, but despite this, even the majority of all 32-bit processors are integrated into embedded systems [Stiller, 2000]. Already in 1996, it was estimated that the average American came into contact with 60 microprocessors per day [Camposano and Wolf, 1996]. Some high-end cars contain more than 100 processors[1]. These numbers are much larger than what is typically expected, since most people do not realize that they are using processors. The importance of embedded systems was also stated by journalist Margaret Ryan [Ryan, 1995]:

"... embedded chips form the backbone of the electronics driven world in which we live. ... they are part of almost everything that runs on electricity".

According to quite a number of forecasts, the embedded system market will be much larger than the market for PC-like systems.

In the United States, the National Science Foundation is supporting research on cyber-physical systems [National Science Foundation, 2010]. In Europe, the Sixth and the Seventh Framework Programme [European Commission Cordis, 2010] support research and development of embedded systems. Also, the ARTEMIS joint undertaking [ARTEMIS Joint Undertaking, 2010] was created as a public/private partnership between government institutions and companies in order to move research and development in embedded computing ahead. This initiative demonstrates the huge interest of the European commercial sector in this technology. Similar initiatives exist on other continents as well.

[1] According to personal communication.

This importance of embedded/cyber-physical systems is so far not well reflected in many of the current curricula. This book is intended as an aid for changing this situation. It provides the material for a first course on such systems. Therefore, it has been designed as a textbook. However, it provides more references than typical textbooks and also helps to structure the area. Hence, this book should also be useful for faculty members and engineers. For students, the inclusion of a rich set of references facilitates access to relevant sources of information.

Audience for this book

This book is intended for the following audience:

- Computer science (CS), computer engineering (CE), and electrical engineering (EE) students as well as students in other ICT-related areas who would like to specialize in embedded/cyber-physical systems. The book should be appropriate for third year students who do have a basic knowledge of computer hardware and software. This means that the book primarily targets senior undergraduate students. However, it can also be used at the graduate level if embedded system design is not part of the undergraduate program. This book is intended to pave the way for **more advanced topics** that should be **covered in follow-up courses**. The book assumes a basic knowledge of computer science. EE students may have to read some additional material in order to fully understand the topics of this book. This should be compensated by the fact that some material covered in this book may already be known to EE students.
- Engineers who have so far worked on systems hardware and who have to move more towards software of embedded systems. This book should provide enough background to understand the relevant technical publications.
- PhD students who would like to get a quick, broad overview of key concepts in embedded system technology before focusing on a specific research area.
- Professors designing a new curriculum for embedded systems.

Curriculum integration of embedded systems

Unfortunately, embedded systems are hardly covered in the latest edition of the Computer Science Curriculum, as published by ACM and the IEEE Computer Society [ACM/IEEE, 2008]. However, the growing number of applications results in the need for more education in this area. This education should help to overcome the limitations of currently available design technologies for embedded systems. For example, there is still a need for better specification

languages, models, tools generating implementations from specifications, timing verifiers, system software, real-time operating systems, low-power design techniques, and design techniques for dependable systems. This book should help teaching the essential issues and should be a stepping stone for starting more research in the area.

Areas covered in this book

This book covers hardware as well as software aspects of embedded systems. This is in-line with the ARTIST guidelines for curricula: "*The development of embedded systems cannot ignore the underlying hardware characteristics. Timing, memory usage, power consumption, and physical failures are important.*" [Caspi et al., 2005].

The book focuses on the fundamental bases of software and hardware. Specific products and tools are mentioned only if they have outstanding characteristics. Again, this is in-line with the ARTIST guidelines: "*It seems that fundamental bases are really difficult to acquire during continuous training if they haven't been initially learned, and we must focus on them.*" [Caspi et al., 2005]. As a consequence, this book goes beyond teaching embedded system design by programming micro-controllers. With this approach, we would like to make sure that the material taught will not be outdated too soon. The concepts covered in this book should be relevant for a number of years to come.

The proposed positioning of the current textbook in computer science and computer engineering curricula is explained in a paper [Marwedel, 2005]. A key goal of this book is to provide an overview of embedded system design and to relate the most important topics in embedded system design to each other. This way, we avoid a problem mentioned in the ARTIST guidelines: "*The lack of maturity of the domain results in a large variety of industrial practices, often due to cultural habits. ... curricula ... concentrate on one technique and do not present a sufficiently wide perspective. .. As a result, industry has difficulty finding adequately trained engineers, fully aware of design choices*" [Caspi et al., 2005].

The book should also help to bridge the gap between practical experiences with programming micro-controllers and more theoretical issues. Furthermore, it should help to motivate students and teachers to look at more details. While the book covers a number of topics in detail, others are covered only briefly. These brief sections have been included in order to put a number of related issues into perspective. Furthermore, this approach allows lecturers to have appropriate links in the book for adding complementary material of their choice. The book includes more references than textbooks would normally do. This way, the book can also be used as a comprehensive tutorial, providing pointers

for additional reading. Such references can also stimulate taking benefit of the book during labs, projects, and independent studies as well as a starting point for research.

Additional information related to the book can be obtained from the following web page:

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

This page includes links to slides, simulation tools, error corrections, and other related material. Readers who discover errors or who would like to make comments on how to improve the book should send an e-mail to:

peter.marwedel at tu-dortmund.de

Assignments could also use the information in complementary books (e.g. [Wolf, 2001], [Buttazzo, 2002], and [Gajski et al., 2009]).

Prerequisites

The book assumes a basic understanding in several areas:

- electrical networks at the high-school level (e.g. Kirchhoff's laws),
- operational amplifiers (optional),
- computer organization, for example at the level of the introductory book by J.L. Hennessy and D.A. Patterson [Hennessy and Patterson, 2008],
- fundamental digital circuits such as gates and registers,
- computer programming (including foundations of software engineering),
- fundamentals of operating systems,
- fundamentals of computer networks,
- finite state machines,
- some first experience with programming micro-controllers,
- fundamental mathematical concepts (such as tuples, integrals, and linear equations), and welcome knowledge in statistics and Fourier series,
- algorithms (graph algorithms and optimization algorithms such as branch and bound),

- the concept of NP-completeness.

These prerequisites can be grouped into courses as shown in the top row in fig. 0.2.

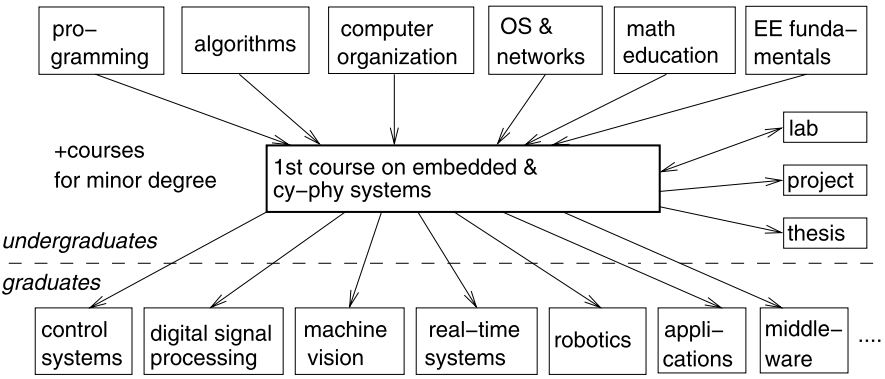

Figure 0.2. Positioning of the topics of this book

Recommended additional teaching

A course using this textbook should be complemented by an exciting lab, using, for example, small robots, such as Lego Mindstorms^{TM} or similar robots. Another option is to let students gain some practical experience with finite state machine tools.

The book should be complemented by follow-up courses providing a more specialized knowledge in some of the following areas (see the bottom row in fig. 0.2)[2]:

- control systems,
- digital signal processing,
- machine vision,
- real-time systems, real-time operating systems, and scheduling,
- middleware,
- application areas such as telecommunications, automotive, medical equipment, and smart homes,

[2]The partitioning between undergraduate courses and graduate courses may differ between universities.

- robotics,
- sensors and actuators,
- specification languages for embedded systems,
- computer-aided design tools for application-specific hardware,
- formal verification of hardware systems,
- testing of hardware and software systems,
- performance evaluation of computer systems,
- low-power design techniques,
- security and dependability of computer systems,
- ubiquitous computing,
- impact of embedded systems.

History of the book

The first edition of this book was published in 2003. The field of embedded systems is moving fast and many new results have become available since then. Also, there are areas for which the emphasis has shifted. In some cases, a more detailed treatment of the topic became desirable. New developments have been taken up when the first German edition of the book was published in 2007. Therefore it became necessary to publish a major new English release, the current second edition.

Names used in this book without any reference to copyrights or trademarks may still be legally protected.

Please enjoy reading the book!

Dortmund (Germany), August 2010

Peter Marwedel

This book is dedicated
to my family members
Veronika, Malte,
Gesine, and Ronja.

Acknowledgments

My PhD students, in particular Lars Wehmeyer, did an excellent job in proofreading a preliminary version of this book. Also, the students attending my courses provided valuable help. Corrections were contributed by David Hec, Thomas Wiederkehr, Thorsten Wilmer and Henning Garus. In addition, the following colleagues and students gave comments or hints which were incorporated into this book: R. Dömer, N. Dutt (UC Irvine), A. B. Kahng (UC San Diego), W. Kluge, R. von Hanxleden (U. Kiel), P. Buchholz, M. Engel, H. Krumm, O. Spinczyk (TU Dortmund), W. Müller, F. Rammig (U. Paderborn), W. Rosenstiel (U. Tübingen), L. Thiele (ETH Zürich), and R. Wilhelm (Saarland University). Material from the following persons was used to prepare this book: G. C. Buttazzo, D. Gajski, R. Gupta, J. P. Hayes, H. Kopetz, R. Leupers, R. Niemann, W. Rosenstiel, H. Takada, L. Thiele, and R. Wilhelm. PhD students of my group contributed to the assignments included in this book. Of course, the author is responsible for all errors and mistakes.

I do acknowledge the support of the European Commission through projects MORE, Artist2, ArtistDesign, Hipeac(2), PREDATOR, MNEMEE and MADNESS, which provided an excellent context for writing the second edition of this book.

The book has been produced using the LaTeXtype setting system from the TeXnicCenter user interface. I would like to thank the authors of this software for their contribution to this work.

Acknowledgments also go to all those who have patiently accepted the author's additional workload during the writing of this book and his resulting reduced availability for professional as well as personal partners.

Finally, it should be mentioned that the Springer company has supported the publication of the book. Their support has been stimulating during the work on this book.