

Preface

What Does This Book Present?

“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.”

This citation from a report of the National Research Council in the USA [392] describes very nicely the dramatic impact of information technology in embedded systems. Such systems can be understood as information processing embedded into an enclosing product [355], such as a car or an airplane. This revolution has already had a major impact and is still continuing. For example, the availability of mobile devices has had an impact on societies. Due to the increasing integration of computing and physical objects, the term cyber-physical systems (CPS) has been introduced. Such systems *“are engineered systems that are built from and depend upon the synergy of computational and physical components”* [394]. Objects or “things” play a key rule in the definition of the popular term “Internet of Things” (IoT). IoT *“... describes ... a variety of devices ... able to interact and cooperate with each other to reach common goals”* [179]. Terms such as pervasive and ubiquitous computing, ambient intelligence, and “Industry 4.0” are also referring to the dramatic impact of changes caused by information technology. More changes like the use of self-driving cars and more remotely controlled airborne devices are on the horizon.

This importance of embedded/cyber-physical systems and IoT is so far not well reflected in many of the current curricula. Designing the mentioned systems requires knowledge and skills from areas beyond traditional programming and algorithm design. Obtaining an overview of such knowledge is very difficult, due to the wide range of relevant areas. This book aims at facilitating the acquisition of knowledge from the relevant areas. It provides material for a first course on such

systems and includes an overview of key concepts for the integration of information technology with physical objects. It covers hardware as well as software aspects. This is in-line with the ARTIST guidelines for curricula of embedded systems: *“The development of embedded systems cannot ignore the underlying hardware characteristics. Timing, memory usage, power consumption, and physical failures are important”* [86].

This book has been designed as a textbook. However, this book provides more references than typical textbooks do 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.

This 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”* [86]. As a consequence, this book goes beyond teaching embedded system design by programming microcontrollers. This book presents the **fundamentals of embedded systems design which are needed for the design of CPS and IoT systems**. 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 [356]. We want to relate the most important topics in this area 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”* [86].

This book should also help to bridge the gap between practical experiences with programming microcontrollers and more theoretical issues. Furthermore, it should help to motivate students and teachers to look at more details. While this 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 this book for adding complementary material of their choice. Due to the rich set of references, this book can also be used as a comprehensive tutorial, providing pointers for additional reading. Such references can also stimulate taking benefit of this book during laboratories, projects, and independent studies as well as a starting point for research.

The scope of this book includes specification techniques, hardware components, system software, application mapping, evaluation and validation, as well as exemplary optimizations and test methods. This book covers embedded systems and their interface to the physical environment from a wide perspective, but cannot cover every related area. Legal and socioeconomic aspects, human interfaces, data

analysis, application-specific aspects, and a detailed presentation of physics and communication are beyond the scope of this book. The coverage of the Internet of Things is limited to areas linked to embedded systems.

Who Should Read the 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 information and communication technology (ICT)-related areas who would like to specialize in embedded/cyber-physical systems or IoT. This book should be appropriate for third-year students who do have a basic knowledge of computer hardware and software. This means that this book primarily targets senior undergraduate students¹. However, it can also be used at the graduate level if embedded system design is not part of the undergraduate program or if the discussion of some topics is postponed. This book is intended to pave the way for **more advanced topics** that should be **covered in follow-up courses**. This 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 toward software of embedded systems. This book should provide enough background to understand the relevant technical publications.
- Ph.D. 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 the mentioned areas.

How Is This Book Different from Earlier Editions?

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 this book was published in 2007. Therefore, it became necessary to publish a second English edition in late 2010/early 2011. Since

¹This is consistent with the curriculum described by T. Abdelzaher in a recent report on CPS education [393].

then, some technological changes occurred. There was a clear shift from single core systems toward multi-core systems. Cyber-physical systems (CPS) and the Internet of Things (IoT) gained more attention. Power consumption and thermal issues have become more important, and many designs now have tight power and thermal constraints. Furthermore, safety and security have gained importance as well. Overall, it became necessary to publish the current third edition.

The changes just described had an impact on several chapters. We are now including and linking those aspects of embedded systems that provide foundations for the design of CPS and IoT systems. The preface and the introduction have been rewritten to reflect these changes. (a) The chapter on specifications and modeling now includes partial differential equations and transaction-level modeling (TLM). The use of this book in flipped-classroom-based teaching led to the consideration of more details, in particular of specification techniques. (b) The chapter on embedded system hardware now includes multi-cores, a rewritten section on memories, more information on the conversion between the analog and the digital domain, including pulse-width modulation (PWM). There are updated descriptions of field programmable gate arrays (FPGAs) and a brief section on security issues in embedded systems. (c) The chapter on system software now contains a section on Linux in embedded systems and more information on resource access protocols. (d) In the context of system evaluation, there are new subsections on quality metrics, on safety/security, on energy models, and on thermal issues. (e) The chapter on mapping to execution platforms has been restructured: A standard classification of scheduling problems has been introduced, and multi-core scheduling algorithms have been added. There is now a clearer distinction between jobs and tasks. The description of hardware/software codesign has been dropped. (f) The chapter on optimizations has been updated. Some information on compilation for specialized processors has been dropped.

All chapters have been carefully reviewed and updated if required. There are also an improved integration of (color) graphics, more assignments (problems), as well as a clearer distinction between definitions, theorems, proofs, code, and examples. We adopted the Springer book layout style.

For this edition, it is typically not feasible to cover the complete book in a single course for undergraduates and lecturers can select a subset which fits the local needs and preferences.

Dortmund, Germany
February 2017

Peter Marwedel

<http://www.springer.com/978-3-319-56043-4>

Embedded System Design

Embedded Systems Foundations of Cyber-Physical
Systems, and the Internet of Things

Marwedel, P.

2018, XXIV, 423 p. 284 illus., 121 illus. in color.,

Hardcover

ISBN: 978-3-319-56043-4