HZ BOOKS
华章教育

ELSEVIER

经 典 原 版 书 库

现代嵌入式计算

（美） Peter Barry Intel公司 Patrick Crowley 华盛顿大学圣路易斯分校 著

（英文版）

MODERN EMBEDDED COMPUTING

Designing Connected, Pervasive, Media-Rich Systems

Peter Barry Patrick Crowley

机械工业出版社
China Machine Press

MK
MORGAN KAUFMANN

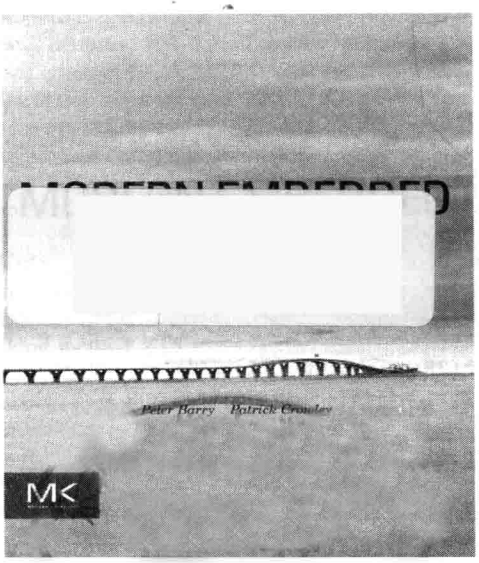
经 典 原 版 书 库

现代嵌入式计算

（英文版）

Modern Embedded Computing

Designing Connected, Pervasive, Media-Rich Systems



（美） Peter Barry Intel公司 Patrick Crowley 华盛顿大学圣路易斯分校 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

现代嵌入式计算（英文版）/（美）巴里（Barry, P.），（美）克罗利（Crowley, P.）著．—北京：机械工业出版社，2013.1
（经典原版书库）
书名原文：Modern Embedded Computing：Designing Connected, Pervasive, Media-Rich Systems

ISBN 978-7-111-41235-9

I. 现… II. ①巴… ②克… III. 微型计算机－系统设计－英文 IV. TP36

中国版本图书馆 CIP 数据核字（2013）第 012564 号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：迟振春
蓟城市京瑞印刷有限公司印刷
2013年2月第1版第1次印刷
186mm × 240mm • 33.75印张
标准书号：ISBN 978-7-111-41235-9
定价：79.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

华章教育
华章科技图书出版中心

To Viviane, Brianna, and Jason
—Peter

To Laura, Sophia, and Alice
—Patrick

Preface

We began writing this textbook with two high-level goals in mind:

1. to introduce students to what we consider to be modern embedded systems, and
2. to show how to understand and program such systems using a concrete platform built around a modern embedded processor like the Intel Atom.

Embedded computing has changed dramatically in recent years, and we have found that academic text books do not reflect this fact. Mindful of the growing importance of embedded systems overall, we saw a need to address the discrepancy between the state of embedded computing and the state of academic textbooks in the area. Hence, this textbook.

Key Insight

How have embedded systems changed dramatically in recent years? Traditionally, embedded systems courses and textbooks take as their foundation microcontroller-based systems used primarily in industrial control applications.

However, many classes of embedded systems today—such as consumer electronics, handsets, and mobile media devices—exhibit the following characteristics.

1. Network connectivity. Nearly all embedded systems include IP networking stacks and enable network connectivity via a combination of wired and wireless network interfaces.
2. Media-rich user interfaces. Many embedded systems include graphical user interfaces built with high-resolution 2D and 3D graphics, as well as displays, inputs, and outputs supporting standard and high-definition video and audio.
3. Aggressive platform integration. For reasons of power efficiency, performance, and size, chips and chipsets for embedded systems are highly integrated, with on-die implementations of memory and I/O controllers, accelerators for computationally intensive tasks such as encryption and compression, and multiple programmable processor cores.

Modern embedded systems are connected, media-rich, and highly integrated. This core insight inspired the development of the book, and distinguishes it from existing texts.

Target Audience

This textbook is designed to form the basis of a semester-long laboratory-based undergraduate embedded design engineering course. The text is also suitable for similar courses offered in the context of continuing education. The primary audience is undergraduate engineering students in computer engineering, electrical engineering, computer science, and embedded system design.

From a technical perspective, embedded systems programmers are the primary audience of the book. Secondary audiences include embedded platform designers and SoC/CPU architects. The chapter outline of the book, and the core content of each chapter, is primarily designed to meet the needs of embedded systems programmers, and therefore emphasize how to develop systems and application software for embedded computing systems.

Our Approach

This undergraduate textbook delves into all aspects of modern embedded systems. It is designed to educate undergraduate engineering students in the principles of embedded system architecture and design. Most principles are further illustrated in a concrete way with examples using the Intel Atom processor.

The book is organized into three parts. The chapters in Part 1 introduce the principles of modern embedded systems. Part 2, which comprises the majority of the text, explores in detail the architecture and operation of embedded systems. The chapters in Part 3 explore detailed, real-world aspects of developing embedded systems, including sample platform descriptions, debugging, and performance tuning.

The text describes example embedded platforms, interfaces, peripherals, processors and operating systems associated with embedded systems, a comprehensive view of the software framework being developed around embedded SOCs, including open source firmware, power management, networking, multimedia and middleware. The text is replete with examples, which are provided to facilitate a comprehensive understanding of the overall platform architecture of modern embedded computing systems.

Beginning with a discussion of embedded platform architecture and Intel Atom-specific architecture, modular chapters cover system boot-up, operating systems, power optimization, graphics and multi-media, connectivity, and platform tuning. A companion reference design platform and an embedded design example case study will enable a laboratory component complimenting the chapters

Online Resources

The book's companion web site provides instructor-controlled access to chapter-specific homework problems and laboratory exercises. These, in combination with the use of the reference hardware platform, enable hands-on embedded design experiences.

Visit booksite.mkp.com/9780123914903 to access ancillaries, and textbooks.elsevier.com/9780123914903 for instructor's materials.

Foreword

While I am not sure where I first heard it, I have increasingly grown fond of this quote: "Architecture is art in structure." It inspires an arc one can follow that is characterized by a fascinating combination of creativity (art) and engineering discipline (structure). The latter creates bounds, the former looks for ways to fly away from it. One focuses on the function, the other is interested in creating a differentiating form. Just when one wonders if the focus on usability or function may enforce a break-no-rule regime, desire to achieve a distinguishing form opens the door for the innovation to step right in! Achieving the right balance between the seemingly contradictory tendencies of form and function requires the imagination and courage to break the mold while understanding and appreciating the rationale behind it.

The term "architecture" entered the computer vernacular in the early 1960s as the developers of mainframe computers at IBM attempted to describe data formats, instruction types, and other hardware parameters and enhancements. However, I would argue that it was probably not until the early 1980s—after the arrival of Intel's 8086 microprocessor and the IBM PC built around it—that the interesting interplay of form and function (art and structure) mentioned above began in earnest in the world of computing. The personal computer (PC) brought the power of computing to the masses, ushering in a new era of innovation and productivity improvements unimaginable in the past. Developers around the world started developing hardware platforms and software applications on top of the foundation the PC laid. As businesses around the world recognized the value of the PC toward improving productivity and eventually their bottom line, there was an obvious desire to do more with the PC. There was just one problem. The PC was not designed with such widespread applicability and usage in mind! Thus, while people explored how to extend its value, some of the fundamental tenets of the platform were already in place. Changing this foundation to allow for a more purposeful build-out would have not just risked slowing down the pace of innovation, but could possibly have undermined the value that was already realized.

In my view, this is where Moore's Law and Intel's relentless pursuit of it came to the rescue. As a lifelong Intel technologist, my view on the value of Moore's Law may appear tainted, but this is not a new claim I am putting forth. How the computer industry has ridden Moore's Law to its phenomenal expansion since the late 1980s has been well documented in trade journals, books, and academic papers. I want to connect it to the interplay of art and structure I alluded to earlier. With the key tenets of the PC architecture in place—the instruction set architecture (ISA) of the IA-32 CPU, key system-level attributes of how CPU, memory, and I/O subsystems interacted, and several platform details in terms of devices, memory map, etc.—the challenge was how to enhance the platform value without undermining these foundational pillars. In Moore's Law, what the developers found was an obliging benefactor that doubled the resources (transistors) at their disposal roughly every two years without incurring a substantial financial downside. So, the stage was set for the creativity to flourish within the guardrails established by the PC platform architecture.

The Intel386™ processor marked the first major breakthrough in this quest to innovate within the confines of the established architecture. With its 32-bit architecture, memory protection schemes, and paging architecture for virtual memory, it paved the way for supporting more sophisticated operating systems. And this was done while retaining backward compatibility with the 8086 architecture. The Intel386 processor established the playbook on how to bring new innovations to the CPU and platform architecture while protecting the investments developers had made in the prior generation platforms.

Through the various generations of Intel's IA-32 processors (later renamed to Intel® Architecture, or IA), more and more sophisticated architectural constructs—various hierarchies of caches, superscalar pipelines, out of order execution with increasingly more efficient branch prediction, hyper-threading and on-chip multicore, virtualization, power saving states, etc.—have been added. Not only that, IA product roadmap evolved to include not just CPUs optimized for the desktop and notebook computers, but also high-end servers (Intel Xeon® processors) as well as lower-power, battery-operated hand-held devices (Intel Atom™ processor). Similarly, there were several enhancements to the platform architecture as well; Advanced Peripheral Interrupt Controllers (APIC) and higher-precision event timers, Peripheral Component Interface (PCI) and subsequent PCI express, Universal Serial Bus (USB) interface, various memory interfaces to support higher bandwidths, and platform-level power management, to just name a few. Over the years, the term Intel Architecture has somewhat become synonymous with a certain platform architecture that ensures the backward compatibility mentioned above. It provided the foundation on top of which a plethora of operating systems, middleware, tool chains, and of course, applications were successfully developed and deployed with relative ease. The history of the evolution of IA is a testament to the creativity of several generations of innovators across the industry that blossomed to embrace and enable unimaginable usage scenarios, the constraints of its foundation notwithstanding!

Not surprisingly, the Intel Architecture, with all the attractive CPU and platform architecture features, found its way into embedded systems over the last three decades. Whether it was a point of sale terminal in a retail segment, an industrial PC in an industrial control segment, a firewall or security appliance in an enterprise segment, or a gaming kiosk, IA provided a ready-to-deploy platform with the most varied software ecosystem to suit different needs of developers in these segments, not to mention the guarantee of the Moore's Law cadence that would sustain predictable and straightforward performance upgrade cycles. Over the last decade the expansion of the IA product portfolio has helped extend its reach within the embedded space. For example, the advent of multi-core Intel Xeon processors has strengthened the IA position in the ever-performance-hungry communications infrastructure sector. On the other hand, the introduction of the Intel Atom processor, with its lower power and lower cost envelopes, has generated tremendous interest in IA in embedded segments—like print imaging, industrial PLC controllers, and in-vehicle infotainment—that were previously out of reach for IA.

It is against this rich and immensely productive backdrop of IA history in embedded systems that I watch with fascination the emerging phenomenon of Internet of Things (IOT)—a world of intelligent systems, including remote wireless sensor nodes, multi-protocol gateways, and smaller ("edge") clouds at the periphery, working seamlessly and in real time with the cloud infrastructure to collect, organize, and analyze massive data and derive knowledge, which then leads to a wise action. While embedded devices connecting to the Internet undoubtedly opens up exciting and enticing new opportunities to fundamentally change the way we live, work, and interact with our surroundings and each other, it also poses a slew of challenges in terms of how to secure and protect these devices as well as how to manage them through their life cycle. If one were to believe the enormous numbers projected for the "connected devices" (a majority of which will be embedded), one could easily surmise that, given the scale, deployment and management of these devices need to become much more simplified than what historically has been the case with PCs and cellular phones. This is where I believe the maturity of IA platforms with respect to the connected paradigm (IA-based clients and servers not only have been at the heart of the Internet build out, but also serve as the development platforms for

practically all other platforms) and the richness of its software ecosystem, coupled with lower power and lower cost envelopes of Intel Atom architecture, can facilitate a easier and technically sound migration to the vision of the IOT.

This book is aimed at providing foundational instruction in the key technology building blocks one needs to master in order to contribute to this new world of embedded systems. I am delighted that the authors of this book not only bring many years of experience in embedded systems, networking, Intel Architecture, and systems engineering, but also represent the visionary technical leadership we will require if we were to successfully take on the challenges of the changing landscape in embedded systems. I know in my heart that this book will greatly serve the young and curious minds who we collectively will count on to change the face of embedded computing in profound ways, ultimately resulting in a better and more meaningful human experience!

—Pranav Mehta
Sr. Principal Engineer & CTO
Embedded & Communications Group
Intel

Acknowledgments

We owe a debt of gratitude to a large number of people for their help and encouragement while writing this book. We would like to thank Pranav Mehta for conceiving the original concept of the book and supporting our work.

We would like to thank the following people for their significant contribution to specific chapters: Max Domeika and Lori Matassa—Embedded Processor Architecture supplement. Pete Dice, Jenny Pelner, and James Pelner—Embedded Platform Boot Sequence. Vasco Santos, David Martinez-Nieto, Julien Carreno, and Ken Reynolds—Digital Signal Processing. Ee Khoon Yap Ee—Graphics and Multimedia. Kevin Bross—Platform Examples. Ger Hartnett—Platform Tuning.

We'd like to thank all of the people who reviewed this book and provided us with much valuable feedback: Sunil Agrawal, Steve Doyle, Ramesh Abel Edgar, Linda Fellingham, Susan Foster, Celeste Fralick, Kevin Gambill, Mylinh Gillen, William Handley, Nee Shen Ho, Muthurajan Jayakumar, Erin Jones, Asad Khan, Tomasz Kilarski, Ray Kinsella, Scott Krig, Micheal Langman, Todd Langley, Matthew Lee, Suresh Marisetty, Krishna Murthy, Bryan O'Donoghue, TJ O'Dwyer, Subhankar Panda, Marc Pepin, Richard Purdie, Mo Rooholamini, Joy Shetler, Benjamin Silva, Jay Simonetta, Brian Skerry, David Stewart, Tat Kin Tan, Tian Tian, Stephanie Wang, Brian Will, and Chris Wojslaw.

We would also like to thank Shrikant Shah, Edward Pullin, and David Clark for their significant attention to detail and whose reviews and edits greatly improved the accuracy and readability of the book.

Contents