Mc Graw Hill Education

大学计算机教育国外著名教材系列（影印版）

EMBEDDED SYSTEMS
ARCHITECTURE
PROGRAMMING AND DESIGN

嵌入式系统
体系结构 编程与设计

Raj Kamal 著

Mc Graw Hill

清华大学出版社

大学计算机教育国外著名教材系列（影印版）

Embedded Systems

Architecture, Programming and Design

嵌入式系统

体系结构 编程与设计

Raj Kamal 著

Institute of Computer Sciences and Electronics

Devi Ahilya University

Indore

清华大学出版社

北京

Raj Kamal
Embedded Systems: Architecture, Programming and Design
EISBN: 0-07-049470-3

图书在版编目(CIP)数据

嵌入式系统体系结构 编程与设计/ 卡马尔(Kamal, R.)著. —影印本. —北京：清华大学出版社，2005.2
(大学计算机教育国外著名教材系列)
ISBN 7-302-10297-X

Ⅰ.嵌…　Ⅱ.卡…　Ⅲ.微型计算机—系统设计—高等学校—教材—英文　Ⅳ.TP360.21

中国版本图书馆 CIP 数据核字(2005)第 000671 号

出 版 者：清华大学出版社　　地　　址：北京清华大学学研大厦
　　　　　http://www.tup.com.cn　　邮　　编：100084
　　　　　社 总 机：010-62770175　　客户服务：010-62776969
印 刷 者：北京市世界知识印刷厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×230　印张：40.25
版　　次：2005 年 2 月第 1 版　2006 年 3 月第 2 次印刷
书　　号：ISBN 7-302-10297-X/TP·7016
印　　数：3001～5000
定　　价：59.00 元

出版说明

进入21世纪，世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才，谁就能在竞争中取得优势。高等教育，作为培养高素质人才的事业，必然受到高度重视。目前我国高等教育的教材更新较慢，为了加快教材的更新频率，教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从1996年开始，与国外著名出版公司合作，影印出版了“大学计算机教育丛书(影印版)”等一系列引进图书，受到国内读者的欢迎和支持。跨入21世纪，我们本着为我国高等教育教材建设服务的初衷，在已有的基础上，进一步扩大选题内容，改变图书开本尺寸，一如既往地请有关专家挑选适用于我国高等本科及研究生计算机教育的国外经典教材或著名教材，组成本套“大学计算机教育国外著名教材系列(影印版)”，以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材，以利我们把“大学计算机教育国外著名教材系列(影印版)”做得更好，更适合高校师生的需要。

清华大学出版社

Preface

Children need embedded systems to play *smart video games* and to operate *automatic chocolate-vending machines*! Young people need embedded systems to borrow *smart cards* from parents to see movies! Housewives need embedded systems for smart Internet-compliant home appliances, such as, microwave, television, music system, and so on. The driver needs it for automatic cruise control of car. Organisations require embedded systems for network systems and products. There are uncountable examples of the applications of embedded systems.

Three *classics* that not only influenced this author greatly but also provided a deep insight into the subject and fueled his interest in microprocessor- and microcontroller-based embedded systems are as follows: Foremost is the classic *'Design with Microcontrollers and Microcomputers'* by J.B. Peatman, McGraw-Hill (1988). The second is *'Microcontrollers Architecture, Implementation and Programming - HD44795, MC68HC11, MCS-51, 80960CA'* by Kenneth Hintz and Daniel Tabak, a McGraw-Hill International Edition (1992). The third is *'Advanced Microprocessors'* by Daniel Tabak, a McGraw-Hill International Edition (1995).

In the early days, embedded systems were designed using microprocessors like 8085. Applications were simple, for example, temperature-monitoring systems, data acquisition systems using ADC and DAC, music systems using appropriate interface and simple robotic systems using stepper motor interface. These are now not even referred to as embedded systems.

Since the early eighties, small-scale embedded systems have used microcontrollers from General Instruments Corporation, using their late seventies microcontrollers PIC 16xxx, Motorola microcontrollers 68HC05 and 08, and Intel microcontrollers of 8031 families. The remote of a television, watches, washing machines, ovens, calculators, digital diaries and video games are all examples of the commonly used small scale embedded systems. The advent of microcontrollers of Intel 8051/52, Motorola 68HC11/12, Intel 80196 and 80960 families in the late eighties gave a new dimension to the use of embedded systems hardware.

The advent in last few years of technology that embeds low level and high level processing hardware elements and application-specific processors into a chip has given an added dimension to the embedded systems that are multiprocessor systems, on a single VLSI chip (called System-on-Chip) and are smart as well as highly sophisticated. A simple example is a smart card, and the latest, typical example of a sophisticated system is "Smart Cameras as Embedded Systems" developed at the

Princeton University Embedded System Group and reported recently by Wayne Wolf and his team in *IEEE Micro*, IEEE Computer Society, September 2002 issue.

Embedded systems have definitions that vary with time as well as perception! ***An embedded system can be defined as one that has computer hardware with software embedded in it as one of its most important components.*** An embedded system is a dedicated, computer-based system for an application or product. *It addresses the issue of the response time constraints of various tasks of the system.* An embedded system may either be an independent system or a part of a larger system. Its software usually embeds in ROM(s) (Read Only Memory). It, thus, does not need secondary memories as in a computer.

While writing this book, the author had in his mind the graduate engineering students, readers and keen learners who are budding embedded system engineers of today and perhaps the master designers of these systems tomorrow. Also in mind were the young software engineers interested in working on embedded software and real-time programming projects. *This book is intended as a textbook for the students and a reference for the engineers to explain the concepts necessary for designing high-performance response-time constrained sophisticated systems.*

The author expects readers of this book to first learn about the embedded system architecture, its basic hardware and software elements, programming models and software engineering practices that are used during the system development process and then learn the software techniques to embed codes into the systems. The author expects them to develop systems that make optimum use of the available system resources*: processor*, *memory*, *ports*, *devices* and *power*. This book has been written to fulfill these expectations.

What is the range of topics, innovative technologies and tools for designing an embedded system, simple or complex? This book will help the readers to understand these with ease. Next, the students must develop useful projects on applications in the chosen area of their choice. Exemplary areas are networking, communication, automobile electronics, data acquisition and storing, serving, processing and securing information, smart robots, real-time control and tracking systems, biomedical systems and sounds, images and video real-time processing, filtering, compressing and encrypting systems.

The organization of the chapters is as follows:

Chapter 1 gives a detailed introduction to embedded systems. Embedded systems hardware consists of a processor, memory devices, I/O devices and basic hardware units—power supply, clock and reset circuit, I/O ports to access peripheral and other on-chip or off-chip units. Examples of physical devices are UART, modem, transceiver, timer-counter, keypad, keyboard, LED display unit, LCD display unit, DAC and ADC and pulse dialer. This chapter introduces these hardware units, embedded software, state-of-the-art embedded systems and RTOS. It also provides a number of exemplary applications.

Chapter 2 explains embedded system architecture by its processor and memory organization. The readers will learn about the structural units in a processor that provides processing power in an embedded system. They will also learn about memory devices. This chapter explains processor and memory selection methods for a given embedded system. The basis on which the memory blocks and segments are allocated to the data structures is also explained. Memory map concepts and DMA concepts are described. How the memories, devices, IO devices and processor interface is also answered.

Chapter 3 describes the devices—parallel and serial port devices, timing devices, devices for synchronous, iso-synchronous and asynchronous communications and important buses for networking these. Also described are the sophisticated interfacing features in device ports. This chapter also gives a description of I^2C, CAN, USB, advanced serial high speed buses, ISA, PCI, PCIX, advanced parallel high-speed buses.

Chapter 4 concentrates on device drivers. These are important service routines in an embedded system. Use of Linux internals as device drivers and for network functions is also described. Device drivers are explained with examples. An understanding of interrupt servicing and handling mechanism is essential for an embedded system designer. This chapter fulfills that need. It explains thoroughly the concept of interrupt latencies and deadlines. This concept is helpful for real-time programming for an embedded system.

Chapter 5 explains the programming concepts and source code engineering tools for embedded system programming in embedded C/C++/Java. It thoroughly explains the use of pointers and data structures in embedded software. Important concepts explained are—use of multiple function calls in cyclic order and use of the function pointers, function queues and queues of interrupt service routines (thus also device drivers), and data structures; queues, stacks and lists. Object oriented programming concepts in C++ and Java are also described in this chapter. Memory optimization is critical in an embedded system. How is it done? This chapter has the answer.

Chapter 6 teaches program modeling concepts during the single and multi-processor systems software development process. The uses of data flow and control data flow graphs are explained. Program models during real-time programming and uses of FSM and Petri Nets are described. This chapter answers the important question: How do we model multiprocessors, and schedule and synchronize the processing of instructions on these?

Chapter 7 is for learning software engineering practices and approaches in system development processes. Explained are the concepts of uses of linear sequential model, RAD (Rapid Development Phase) model and other important models including the use of component-based (object-oriented) software development process models. Software requirement analysis, design, implementation, testing, debugging and validating strategies are described in this chapter. Also described is the UML language, which has emerged as an important design language.

Chapter 8 covers the most important aspect of real-time programming, namely, inter-process communications. It first explains the concept of the processes, tasks and threads. Next, it describes the use of semaphores. This chapter also explains thoroughly the use of the signals, mutex, message queues, mailboxes, pipes, virtual (logical) sockets and remote procedure calls.

Chapter 9 describes RTOS concepts. It first describes the OS structure and kernel functions. It then explains the process, memory, devices, files and IO subsystem management functions. It describes schedule management for multiple tasks by an RTOS. It further explains how the scheduling is done for multiple tasks in real time by periodic, cyclic, preemptive, time slicing and other scheduling models. IEEE standards are described for the RTOS functions. A highlight of this chapter is description of a fifteen-point strategy for synchronization between the processes.

Chapter 10 describes the two most important RTOS tools, µC/OS-II and VxWorks thoroughly with examples of the use of functions in these RTOSs.

Chapter 11 describes four case studies of programming with RTOS. These are *automatic chocolate vending machin*e system, *TCP/IP network system*, *adaptive cruise control system* in a car and *smart card*.

Chapter 12 details hardware and software designing and integration methods and tools. It explains the embedded system development process action plan. Uses of target system, emulator, ICE, use of device programmer for downloading the finalized codes into ROM, uses of code generation tools (assembler, compiler, loader and linker), simulator, exemplary prototype development tools and IDE are explained. Use of hardware testing tools is also described in this chapter.

Appendices give in brief the gist of CISC and RISC processor architectures, addressing modes and instruction sets, embedded high performance processors, ARM7, ARM9, ARM11 and IBM PowerPC 750. Also given is the overview of microcontroller architectures. DSPs are used in systems for imaging, video and convergent technology products. Triple media (data, voice and video) processors are new innovative processors for real-time video, streaming networks and data networks. Appendix covers these too. Serial and parallel buses for interconnecting distributed devices and device hardware units in the embedded systems are also covered in brief. Embedded system topics are being introduced in the graduate and undergraduate programs. To guide the course designers and teachers, the suggested units in these courses are given in appendix.

Also given are over one hundred references of published books, websites and journal papers. This will enable the readers to carry out further in-depth study of the topics related to Embedded Systems.

The seven *Salient Features* of the book are:

1. Well structured, systematic coverage, and logical sequencing of topics.
2. Thorough explanation of embedded system programming concepts, OS, RTOS functions and inter-process synchronization.
3. Special coverage of modeling of programs and use of software engineering practices during software development process for single as well as multiprocessor systems.
4. Thorough explanation of ports, devices, buses for networking devices and device drivers
5. Innovative case studies of RTOS programming in consumer electronics, communications, automobile electronics and secure transaction systems-on-chip.
6. Simultaneous coverage of two RTOSs, μC/OS-II and VxWorks, together to focus on finer points in applications of functions in the RTOSs.
7. Lucid presentation with strong emphasis on examples, well-designed figures and tables, list of keywords and their definitions followed by chapter-end review questions and practice exercises.

Every effort has been made to give correct information and exemplary codes and tools in the book. However, errors might have crept in despite utmost care. The author will be grateful to the readers for pointing out these to him.

Learned teachers, scholars, software and system engineers will be obliging the author if they send their valuable suggestions for further improvement of this textbook. Suggestions and student queries will be heartily welcomed at the author's e-mail, `professor@rajkamal.org` and website `http://www.rajkamal.org`.

Raj Kamal

Acknowledgements

The author is grateful to his teachers at the Indian Institute of Technology, New Delhi, (1966–1972), and to the teachers at the University of Upscale, Upscale, Sweden, (1978–79, 1984). From these teachers, he learned about the great role of self-learning and the constant drive for understanding emerging technologies, and he developed a passion for knowledge. He also learned to keep abreast with the latest high technology areas.

Author is grateful to Professor M. S. Sodha, F.N.A., an eminent scientist, educationist and administrator. He gave him during 1989–92 an opportunity to plan and set up from scratch the Electronics and Computer Laboratories at Devi Ahilya University. He has blessed the author all through his academic life since 1966.

Encouragement of Dr. Bharat Chapperwal, Vice Chancellor, colleagues and friends at Devi Ahilya University in the first phase is unforgettable and the author is thankful to them. Author is especially thankful to Ms. Preeti Saxena (for checking and minutely tracing the errors in the C/C++ codes in the examples given in the book) and for the support of Dr. Sanjiv Tokekar, Ms. Vrinda Tokekar, Dr. Sumant Katyal, Mr. Sanjeev Shrivastava, of the University faculty and Ms. Vasanti G. Parulkar and Ms. Aparna Dev, his laboratory colleagues, during this phase.

Blessings of Kalvivallal T. Kalasalingam, chairman, A K College of Engineering and college Principal Dr. Chelliah Thangaraj's continuous support in the second but no-less important phase of this book are also unforgettable and the author is grateful to them for these. The help of his college colleagues, particularly Dr. S. Radhakrishna and Dr. M. Shanmugasundram at various stages is gratefully acknowledged.

The author is thankful to Mr. Sandeep Bansal (Indore) and Mr. Annadurai (AKCE) for making the AutoCad drawings. The author is grateful to the editorial team at Tata McGraw-Hill for reviews and suggestions. Finally, he is grateful to his understanding spouse (Sushil), daughter (Shilpi) and son (Shalin).

Raj Kamal

Contents

Appendix D. Embedded Digital Signal Processors 575

Appendix E. New Innovative Processors for Embedded Systems 587

Appendix F. Serial and Parallel Buses 592

Appendix G. Devices in Embedded Systems 594

Appendix H. Important Topics in Embedded Systems Architecture, Programming and Design 611

Chapter

1

Introduction to Embedded Systems

What We Will Learn

In this chapter, we will learn the following:

Section 1.1

1. Definitions of ***system*** and ***embedded system***.
2. Classification of embedded systems into three types.
3. Skills needed in designing an embedded system.

Section 1.2 The processing unit (s) of the embedded system

1. *Processor in an Embedded System* A processor is an important unit in the embedded system hardware. A ***microcontroller*** is an integrated chip that has the processor, memory and several other hardware units in it; these form the microcomputer part of the embedded system. An ***embedded processor*** is a processor with special features that allow it to be embedded into a system. A ***digital signal processor (DSP)*** is a processor meant for applications that process digital signals. [For example, filtering, noise cancellation, echo elimination, compression and encryption].
2. Commonly used microprocessors, microcontrollers and DSPs in the small-, medium-and large-scale embedded systems.
3. A recently introduced technology that additionally incorporates the ***application-specific system processors*** (*ASSP*s) in the embedded systems.
4. Multiple processors in a system.

Section 1.3

1. Embedded system ***power source(s)*** and the need for controlled power-dissipation.
2. Embedded system ***Clock*** oscillator circuit and clocking unit. It lets a processor execute and process instructions.
3. ***Real time clock (RTC),*** timers and various timing needs of the system.
4. ***Reset circuit*** and ***watchdog timer***.
5. ***System memories***. [In the second part of Chapter 2, we will learn the system memories in detail].
6. System Input Output (***IO***) ports, serial Universal Asynchronous Receiver and Transmitter ***(UART)*** and other ports, multiplexers and demultiplexers and interfacing buses. [These will be dealt with in detail in Chapter 3].
7. Interrupt Handler [The latter part of Chapter 4 has the details].
8. Interfacing units- DAC (Digital to Analog Converter) using PWM (Pulse Width Modulation), ADC (Analog to Digital Converter), LED and LCD display units, keypad and keyboard, pulse dialer, modem and transceiver
9. Hardware required for exemplary embedded systems.

Section 1.4 Different levels of languages that are used to develop the ***embedded software*** for a system [Chapter 5 details the high-level programming aspects, 'C' and 'C++' language structures and the application of these for coding embedded software].

1. System device drivers, device management and multitasking using an operating system (OS) and ***real time operating system*** (RTOS). [RTOS (s) are dealt with in detail in Chapters 9 and 10, and case studies using RTOSs in Chapter 11].
2. Software tools in system designing.
3. Software tools required in six exemplary cases.
4. Programming models for software designing. [Software designing models are detailed in Chapter 6].

Section 1.5 ***Exemplary applications*** of each type of embedded system.

Section 1.6 Designing an embedded system on a VLSI chip.

1. Embedded ***SoC*** (System on Chip) and ***ASIC*** (Application Specific Integrated Circuit) and examples of their applications. These use (i) Application Specific Instruction Processor (ASIP), (ii) Intellectual Property (IP) core, (iii) Field Programmable Gate Arrays (FPGA) core with single or multiple processor units on an ASIC chip.
2. ***Smart card,*** an example of the units of an embedded system on a chip (SoC).