



Embedded Software Development with C

Kai Qian
David den Haring
Li Cao

 Springer

Kai Qian • David den Haring • Li Cao

Embedded Software Development with C

 Springer

Preface

Welcome to the world of embedded systems software development with the C language. This book is an introductory text intended primarily for undergraduate students in computer science, electrical engineering, software engineering and other related engineering majors. If you are not a student, however, please don't let that stop you from reading and using this book to help satisfy your curiosity about embedded systems or to help move your career forward.

Overview

This is a practical book. It won't waste your time. Here are three reasons why:

1. **The C programming language.** C has a long history and is the defacto high-level language in the embedded systems world. It is also used widely for operating system and application development for all important platforms like Windows, UNIX, Macintosh and Linux. Learning C is definitely not a waste of time – it's used everywhere.
This book assumes that you have a basic knowledge of C, but not much more than that. If you need a quick review, check out Appendix A and B.
2. **The 8051 microcontroller.** Like C, the 8051 microcontroller (originally developed by Intel) has a long history and remains a popular and relevant platform for designing embedded systems.
3. **Bundled hardware and software.** Learn best by doing! This book has complete, self-contained labs. The book can be bundled with the Silicon Labs 8051 development kit (C8051F005DK), all lab materials (e.g. breadboard, LEDs, resistors, etc) and the Keil C software. This optional bundling is convenient for both the university and the individual.

Whether you are a student, educator or working professional, this book will give you a solid foundation in the hardware and software aspects of embedded systems development.

Part IV: Hands-on Project Labs

- Chapter 9 is the lab chapter. There are eight labs with step-by-step instructions, circuit diagrams, finished lab photos and complete source code.
- Appendix D is an introductory lab that explains in more detail than chapter 9 how to breadboard circuits. If you find the first few labs in Chapter 9 difficult, check out Appendix D for a softer introduction.
- Appendix E is only available at <http://embeddedbook.x85.com>. It features an advanced project that demonstrates PC to 8051 communications over a Bluetooth wireless network.

The authors firmly believe that people learn by studying the example of others and then learn by doing. With that said, source code is included with each lab for study and modification. That's the "*learn by example*" part.

The student must wire the circuits and use the IDE tools to download and debug the software. There are also lab questions and optional exercises which will require students to modify the code, write new functions and add additional hardware components. That's the "*learn by doing*" part. This two-step approach will allow instructors to provide students with labs of varying length and difficulty.

The authors designed, built and tested all of the labs. The labs are available for download at <http://embeddedbook.x85.com/>. Additionally, full source code is included in the text. Instructors can get access to testbank material based on the topics brought up in this text by sending E-mail to kqian@spsu.edu.

We enjoyed writing this book and hope that you learn and benefit from it in your academic and professional career.

Acknowledgements

Thanks to all those who reviewed this book for their constructive comments, suggestions, and encouragements. Thanks to Iqbal Ahmed for his help in development of the uWeb Embedded Web Server. We appreciate the hard work and support of the editorial and production teams at the Springer publishing.

June, 2009

Kai Qian
David den Haring
Li Cao

2.4	Time Requirement Analysis for Real-Time Systems	49
2.4.1	Non-Preemptive Scheduling	51
2.4.2	Pre-emptive Scheduling	52
2.4.3	RMS	53
2.4.4	Dynamic scheduling with EDF	55
2.5	Multi-Tasking Design Methodology	56
2.5.1	Polling	57
2.5.2	Interrupts	58
2.5.3	RTOS	60
2.6	Software Design Issues	62
2.6.1	Task Interactions	62
2.6.2	Resource Sharing	63
2.7	Lab Practice: A Traffic Light Control System Modeling and Design	66
2.8	Summary	68
2.9	Review Questions	69
2.10	Exercises	70
	References	71
3	8051 Microcontroller	73
3.1	Overview	73
3.1.1	8051 Chip Pins	74
3.1.2	System Clock and Oscillator Circuits	75
3.1.3	8051 Internal Architecture	76
3.2	Ports	78
3.2.1	Port Reading and Writing	78
3.2.2	The Port Alternate Functions	79
3.3	Memory and SFR	80
3.3.1	Memory	80
3.3.2	Special Function Registers (SFRs)	82
3.4	SFRs and Interrupts	90
3.4.1	External Interrupts	91
3.4.2	Timer/Counter Interrupts	92
3.4.3	Serial Interrupts	92
3.5	Summary	94
3.6	Review Questions	94
3.7	Exercises	96
	References	96
4	Embedded C Programming with 8051	97
4.1	Overview	97
4.2	Memory in 8051 C Programming	98
4.2.1	8051 Memory Types	98
4.2.2	Memory Models	99
4.3	Data Types of 8051 C	100