

Kai Qian • David den Haring • Li Cao

Embedded Software Development with C

Kai Qian
Dept. of Computer Science
Southern Polytechnic University
Marietta GA 30060
USA
kqian@spsu.edu

David den Haring
Atronix Engineering Inc.
3100 Medlock Bridge Road
Suite 110
Norcross GA 30071
USA
ddharing@atronixengineering.com

Li Cao
University of Texas at Austin
2500 San Jacinto Blvd.
Austin TX 78705
USA
lcao@x85.com

ISBN 978-1-4419-0605-2

e-ISBN 978-1-4419-0606-9

DOI 10.1007/978-1-4419-0606-9

Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009926312

© Springer Science+Business Media, LLC 2009

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

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

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.

Roadmap

This book basically consists of 4 parts.

Part I: Getting Started with the Microcontroller and Embedded Systems Development

- Chapter 1 provides an overview of microcontrollers, embedded systems and their application.
- Appendix A gives a review of C programming. The reader can skip it if you have experience with C.
- Appendix B and C help the reader get started with the Keil and Silicon Labs development tools:

After the completion of Part 1, the reader will have a better understanding of microcontrollers and embedded systems and will be able to develop simple embedded application using Keil and C8051F005DK with 8051 MCU.

Part II: Embedded System Fundamentals

- Chapter 2 gets serious (briefly) and discusses requirements analysis and software design.
- Chapter 3 takes the reader on a tour of hardware fundamentals that are common to all embedded systems.
- Chapter 4 shifts attention to the C language and how to use it properly in an embedded environment where CPU and memory resources are limited.
- Chapter 5 provides an introduction to the RTOS (Real Time Operating System) and presents a simple RTOS for the 8051 microcontroller.
- Chapter 6 introduces the reader to serial communications, UART, and serial communication programming in the 8051. Software and hardware handshaking are also discussed.

Part III: Popular Microcontrollers and Ethernet Application Development

- Chapter 7 pulls away from all the low level detail and puts the 8051 microcontroller in perspective with other popular microcontrollers like the ARM, PIC and Rabbit.
- Chapter 8 discusses Ethernet applications and the future of the microcontroller. This is not a survey chapter. It introduces TCP/IP and features one case study and a complete example. All source code is provided. To work the example in this chapter, the Silicon Labs ETHERNETDK kit is required. The chapter concludes with a look at the possible future of the microcontroller.

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

Contents

1	Introduction to Embedded Systems	1
1.1	Overview	1
1.2	Categories of Embedded Systems	2
1.3	Application Areas of Embedded Systems	4
1.4	Characteristics of Embedded systems	4
1.5	Hardware Overview	6
1.5.1	Overall Architecture	6
1.5.2	Memory	9
1.5.3	Bus	11
1.5.4	I/O Ports	12
1.5.5	Clock Oscillator, Timer and Watchdog	14
1.5.6	Analog to Digital Converter (ADC) and DAC	15
1.6	Embedded Software Design and Development	15
1.6.1	Software Engineering Life cycle	16
1.6.2	Choose RTOS and Programming Language	17
1.7	Practice Labs	19
1.7.1	Getting Started with the Keil μ Vision3 C51 Development Kit	19
1.7.2	Lab with 8051 Microcontroller Development Kit	27
1.8	Summary	35
1.9	Review questions	36
1.10	Exercises	37
	References	37
2	Embedded Software Design and Development	39
2.1	Overview	39
2.2	Software Requirement Specification	41
2.3	Embedded Software Modeling Analysis and Design	42
2.3.1	Context Diagram	42
2.3.2	Finite State Machine (FSM) and State Chart	42

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

4.4	Functions	104
4.4.1	Interrupt Functions	104
4.4.2	Reentrant Functions	109
4.4.3	Real Time Function	111
4.5	Pointers	111
4.6	Mix C and Assembly Code	112
4.7	Modular Programming in C	113
4.7.1	Scope of Functions and Variables	114
4.7.2	Header Files	116
4.7.3	Multi-module C Programming	116
4.8	Lab: Debug and Test Embedded C Program with the Keil μVision3	120
4.9	Practice Examples	126
4.10	Summary	133
4.11	Review Questions	133
4.12	Programming Exercises	135
	References	135
5	Real-Time Operating Systems	137
5.1	Overview	137
5.2	RTX51-Tiny	140
5.3	Task Scheduling	142
5.3.1	Round Robin Multitasking	142
5.3.2	Cooperative Multitasking	145
5.3.3	Priority-Driven Preemptive Multitasking	147
5.4	RTOS Events	149
5.4.1	Signal Event Based Communication	149
5.4.2	Message Event Based Communication	149
5.4.3	Semaphores	150
5.4.4	Attaching Interrupts to Tasks	150
5.5	When to Use RTOS	151
5.6	Practice Labs	151
5.7	Summary	174
5.8	Review Questions	174
5.9	Exercises	176
	References	178
6	Serial Communications	179
6.1	Introduction	179
6.1.1	Serial vs. Parallel	180
6.1.2	Simplex and Duplex	180
6.2	UARTs and Transceivers	181
6.2.1	UART	181
6.2.2	Transceivers (aka Line Drivers)	182
6.2.3	Asynchronous vs. Synchronous	182

6.2.4	Data Transfer Rate	183
6.3	Configuring the Serial Port	184
6.4	Setting the Baud Rate	185
6.5	Reading and Writing	187
6.6	Handshaking	188
6.6.1	Software Handshaking	188
6.6.2	Hardware Handshaking	188
6.7	Examples	190
6.7.1	Example 1 – Hello World (Using STDIO)	190
6.7.2	Example 2 – Hello World (Beyond the Serial Port)	192
6.7.3	Example 3 – Interactive Console Using the Serial Port	196
6.8	Summary	200
6.9	Review Questions	201
6.9.1	Key to the Review Questions:	201
6.10	Exercises	202
7	Survey of Popular Microcontrollers	203
7.1	Introduction	203
7.2	Features for Comparison	204
7.2.1	Packages	205
7.3	The Large 8051 Family	206
7.3.1	8051 Development Tools	206
7.3.2	Silicon Laboratories Family	207
7.3.3	Atmel Family	207
7.3.4	Maxim (i.e. Dallas) Family	207
7.4	PIC Microcontrollers	209
7.4.1	PIC Development Tools	210
7.5	Rabbit Microprocessors	211
7.5.1	Rabbit Development Tools	212
7.6	Atmel AVR Microcontrollers	213
7.6.1	AVR 8-bit MCU	213
7.6.2	AVR 8-bit Development Tools	215
7.6.3	AVR 32-bit MCU	215
7.6.4	AVR 32-bit Development Tools	216
7.7	ARM Microprocessors	217
7.8	ColdFire Microprocessors (and Microcontrollers)	217
7.8.1	ColdFire Development Tools	218
7.9	Basic Stamp	219
7.9.1	Basic Stamp Development Tools	220
7.10	Review Questions	221
7.11	Exercises	221

8 Ethernet Applications and the Future of the Microcontroller	223
8.1 Objectives	223
8.2 Introduction	223
8.3 Introduction to TCP/IP	224
8.3.1 Link Layer	225
8.3.2 Internet Layer	225
8.3.3 Transport Layer	226
8.3.4 Application Layer	226
8.4 Circuit Cellar® Project	227
8.4.1 The Role of the Ethernet Controller	227
8.4.2 The Role of the 8051	228
8.4.3 Code Organization	229
8.5 Silicon Labs Ethernet Development Kit	230
8.5.1 Stage 1: Setup the Hardware	231
8.5.2 Stage 2: Setup the Software and Checkout the Hardware	231
8.5.3 Stage 3: Creating an Embedded Ethernet Application using the Wizard	234
8.5.4 Stage 4: The uWeb Embedded Web Server	245
8.6 The Future of the Microcontroller	246
8.6.1 Dominant Market Share	247
8.6.2 Networking, Networking, Networking	248
8.6.3 New Applications and Role Blurring	248
8.6.4 Better and Cheaper Development Tools	248
8.6.5 Roll your Own Chips	248
8.7 Review Questions	249
8.7.1 Key to the review questions:	249
8.8 Exercises	250
8.8.1 Note: The ETHERNETDK kit is needed for these exercises.	250
9 Hands-On Labs	251
9.1 Lab 1: Getting Started with The 8051 – Reading and Writing Ports	252
9.2 Lab 2: Intro to the 7-segment LED (Direct Driven and Decoded)	258
9.3 Lab 3: Intro to A/D Conversion Using a Light Sensor	266
9.4 Lab 4: Intro to the Keypad	276
9.5 Lab 5: Intro to D/A Conversion Using a Speaker	284
9.6 Lab 6: 3-Wire RS232 Serial Communication	295
9.7 Lab 7: Intro to the LCD Display	303
9.8 Lab 8: Advanced RS232 Communications (Handshaking)	314
A C programming Primer	327
A.1 Overview	327
A.2 Data Types in C	328
A.2.1 Primitive Data Types	328
A.2.2 Enumerated Type	329

A.2.3	Pointer Type	330
A.2.4	Arrays	330
A.2.5	Char and Strings	332
A.2.6	Structures and Unions	333
A.2.7	User Defined Type with typedef	334
A.2.8	Variable Scope Specifiers	335
A.3	Operators	335
A.3.1	Arithmetic and Shift Operators	335
A.3.2	Bitwise Operators	336
A.3.3	Relational Operators	336
A.3.4	Logical Operators	337
A.4	Functions	337
A.4.1	Function Definition	337
A.4.2	Function Parameters	339
A.5	Control Structures	340
A.5.1	Selection Statements	341
A.5.2	Loop Statements	343
A.5.3	break and continue Statement in a Loop	347
A.5.4	goto Statements	348
A.6	Reserved Keywords	348
A.7	Command-line Arguments	349
B	Getting Started with The Keil μVision IDE	351
B.1	Introduction to the Keil μ Vision IDE	351
B.2	Run a Keil μ Vision IDE Project	351
B.3	Build your First Keil Project	354
C	Getting Started with Silicon Laboratories C8051F005DK Development	363
D	Counter with 7-Segment Numeric Display	373
D.1	Overview	373
D.2	Equipment	374
D.3	Schematic	375
D.4	Building the Circuit	375
D.5	Code	376
	References	380
	Index	383