EMBEDDED SYSTEMS:

REAL-TIME OPERATING SYSTEMS FOR ARM® CORTEX™-M MICROCONTROLLERS

Volume 3

Fourth Edition,

January 2017

Jonathan W. Valvano

Fourth edition
January 2017

ARM and uVision are registered trademarks of ARM Limited.
Cortex and Keil are trademarks of ARM Limited.
Stellaris and Tiva are registered trademarks Texas Instruments.
Code Composer Studio is a trademark of Texas Instruments.
All other product or service names mentioned herein are the trademarks of their respective owners.

In order to reduce costs, this college textbook has been self-published. For more information about my classes, my research, and my books, see http://users.ece.utexas.edu/~valvano/

For corrections and comments, please contact me at: valvano@mail.utexas.edu. Please cite this book as: J. W. Valvano, Embedded Systems: Real-Time Operating Systems for ARM® Cortex™-M Microcontrollers, Volume 3, http://users.ece.utexas.edu/~valvano/, ISBN: 978-1466468863.

Copyright © 2017 Jonathan W. Valvano
All rights reserved. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, web distribution, information networks, or information storage and retrieval, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

ISBN-13: 978-1466468863
ISBN-10: 1466468866

Table of Contents

Preface to The Fourth Edition xii

Preface to Volume 3 xii

Acknowledgements xiii

1. Computer Architecture 1

1.1. Introduction to Real-Time Operating Systems 1

1.1.1. Real-time operating systems 1

1.1.2. Embedded Systems 3

1.2. Computer Architecture 4

1.2.1. Computers, processors, and microcontrollers 4

1.2.2. Memory 7

1.3. Cortex-M Processor Architecture 9

1.3.1. Registers 9

1.3.2. Stack 11

1.3.3. Operating modes 13

1.3.4. Reset 14

1.3.5. Clock system 14

1.4. Texas Instruments Cortex-M Microcontrollers 15

1.4.1. Introduction to I/O 15

1.4.2. Texas Instruments TM4C123 LaunchPad I/O pins 17

1.4.3. Texas Instruments TM4C1294 Connected LaunchPad I/O pins .. 21

1.4.4. Texas Instruments MSP432 LaunchPad I/O pins 26

1.4.5. Interfacing to a LaunchPad 29

1.5. ARM Cortex-M Assembly Language 30

1.5.1. Syntax 30

1.5.2. Addressing modes and operands 32

1.5.3. List of twelve instructions 32

1.5.4. Accessing memory 34

1.5.5. Functions..37
1.5.6. ARM Cortex Microcontroller Software Interface Standard 39
1.5.7. Conditional execution..40
1.5.8. Stack usage 42
1.5.9. Floating-point math..43
1.5.10. Keil assembler directives..47
1.6. Pointers in C... 48
1.6.1. Pointers 48
1.6.2. Arrays..50
1.6.3. Linked lists...52
1.7. Memory Management 54
1.7.1. Use of the heap...54
1.7.2. Simple fixed-size heap 56
1.7.3. Memory manager: malloc and free.................................57
1.8. Introduction to debugging 61
1.9. Exercises... 65
2. Microcontroller Input/Output 67
2.1. Parallel I/O 67
2.1.1. TM4C I/O programming 68
2.1.2. MSP432 I/O programming 71
2.2. Interrupts.. 76
2.2.1. NVIC..77
2.2.2. SysTick periodic interrupts 80
2.2.3. Periodic timer interrupts 83
2.2.4. Critical sections 86
2.2.5. Executing periodic tasks...90
2.2.6. Software interrupts..91
2.3. First in First Out (FIFO) Queues 92
2.4. Edge-triggered Interrupts... 94
2.4.1. Edge-triggered interrupts on the TM4C123.........................94

2.4.2. Edge-triggered Interrupts on the MSP432 98

2.5. UART Interface 101

2.5.1. Transmitting in asynchronous mode 103

2.5.2. Receiving in asynchronous mode 104

2.5.3. Interrupt-driven UART on the TM4C123 105

2.5.4. Interrupt-driven UART on the MSP432 110

2.6. Synchronous Transmission and Receiving using the SSI 112

2.7. Input Capture or Input Edge Time Mode 114

2.7.1. Basic principles 114

2.7.2. Period measurement on the TM4C123 116

2.7.3. Period measurement on the MSP432 120

2.7.4. Pulse width measurement 125

2.7.5. Ultrasonic distance measurement 125

2.8. Pulse Width Modulation 125

2.8.1. Pulse width modulation on the TM4C123 126

2.8.2. Pulse width modulation on the MSP432 127

2.9. Analog Output 130

2.10. Analog Input 134

2.10.1. ADC Parameters 134

2.10.2. Internal ADC on TM4C 135

2.10.3. Internal ADC on MSP432 139

2.10.4. IR distance measurement 143

2.11. OS Considerations for I/O Devices 144

2.11.1 Board Support Package 144

2.11.2 Path Expression 146

2.12. Debugging 148

2.12.1. Functional Debugging 148

2.12.2. Performance Debugging (FFT analysis) 150

2.12.3. Debugging heartbeat 153

2.12.4. Profiling 154

2.13. Exercises **155**
3. Thread Management **159**
3.1. Introduction to RTOS **159**
3.1.1. Motivation 159
3.1.2. Parallel, distributed and concurrent programming 160
3.1.3. Introduction to threads 162
3.1.4. States of a main thread 163
3.1.5. Real-time systems 164
3.1.6. Producer/Consumer problem using a mailbox 168
3.1.7. Scheduler 169
3.2. Function pointers **172**
3.3. Thread Management **174**
3.3.1. Two types of threads 174
3.3.2. Thread Control Block (TCB) 175
3.3.3. Creation of threads 177
3.3.4. Launching the OS 178
3.3.5. Switching threads 179
3.3.6. Profiling the OS 181
3.3.7. Linking assembly to C 182
3.3.8. Periodic tasks 183
3.4. Semaphores **185**
3.5. Thread Synchronization **188**
3.5.1. Resource sharing, nonreentrant code or mutual exclusion 188
3.5.2. Condition variable 189
3.5.3. Thread communication between two threads using a mailbox.... 190
3.6. Process Management **192**
3.7. Dynamic loading and linking **193**
3.8. Exercises **195**
4. Time Management **197**
4.1. Cooperation **197**

4.1.1. Spin-lock semaphore implementation with cooperation 197
4.1.2. Cooperative Scheduler...199
4.2. Blocking semaphores 200
4.2.1. The need for blocking 200
4.2.2. The blocked state 201
4.2.3. Implementation..203
4.2.4. Thread rendezvous 205
4.3. First In First Out Queue.. 206
4.3.1. Producer/Consumer problem using a FIFO 206
4.3.2. Little's Theorem 207
4.3.3. FIFO implementation..208
4.3.4. Three-semaphore FIFO implementation..209
4.3.5. Two-semaphore FIFO implementation..212
4.3.6. One-semaphore FIFO implementation 214
4.3.7. Kahn Process Networks..215
4.4. Thread sleeping 217
4.5. Deadlocks.. 218
4.6. Monitors 220
4.7. Fixed Scheduling 222
4.8. Exercises... 226
5. Real-time Systems 229
5.1. Data Acquisition Systems.. 229
5.1.1. Approach 229
5.1.2. Performance Metrics 231
5.1.3. Audio Input/Output 237
5.2. Priority scheduler 240
5.2.1. Implementation..240
5.2.2. Multi-level Feedback Queue...241
5.2.3. Starvation and aging 243
5.2.4. Priority inversion and inheritance on Mars Pathfinder................243

5.3. Debouncing a switch 244
5.3.1. Approach to debouncing 244
5.3.2. Debouncing a switch on TM4C123 245
5.3.3. Debouncing a switch on MSP432 248
5.4. Running event threads as high priority main threads 250
5.5. Available RTOS 251
5.5.1. Micrium uC/OS-II 251
5.5.2. Texas Instruments RTOS 254
5.5.3. ARM RTX Real-Time Operating System 255
5.5.4. FreeRTOS 256
5.5.5. Other Real Time Operating Systems 257
5.6. Exercises 258
6. Digital Signal Processing 261
6.1. Basic Principles 262
6.2. Multiple Access Circular Queue 267
6.3. Using the Z-Transform to Derive Filter Response 270
6.4. IIR Filter Design Using the Pole-Zero Plot 274
6.5. Discrete Fourier Transform 279
6.6. FIR Filter Design 281
6.7. Direct-Form Implementations. 283
6.8. Exercises 285
7. High-Speed Interfacing 287
7.1. The Need for Speed 287
7.2. High-Speed I/O Applications 288
7.3. General Approaches to High-Speed Interfaces 290
7.3.1. Hardware FIFO 290
7.3.2. Dual Port Memory 291
7.3.3. Bank-Switched Memory 292
7.4. Fundamental Approach to DMA 293
7.4.1. DMA Cycles 293

7.4.2. DMA Initiation..294
7.4.3. Burst versus Single Cycle DMA 295
7.4.4. Single Address versus Dual Address DMA 296
7.4.5. DMA programming on the TM4C123 298
7.6. Exercises.. 305
8. File system management 307
8.1. Performance Metrics 307
8.1.1. Usage 307
8.1.2. Specifications...308
8.1.3. Fragmentation..309
8.2. File System Allocation 310
8.2.1. Contiguous allocation..311
8.2.2. Linked allocation 312
8.2.3. Indexed allocation 313
8.2.4. File allocation table (FAT) 314
8.3. Solid State Disk 316
8.3.1. Flash memory...316
8.3.2. Flash device driver 318
8.3.3. eDisk device driver 319
8.3.4. Secure digital card interface...321
8.4. Simple File System.. 326
8.4.1. Directory 327
8.4.2. Allocation...327
8.4.3. Free space management 328
8.5. Write-once File System 329
8.5.1. Usage 329
8.5.2. Allocation...331
8.5.3. Directory 334
8.5.4. Append 335
8.5.5. Free space management 337

8.6. Readers-Writers Problem 338
8.7. Exercises 339
9. Communication Systems 343
9.1. Fundamentals 343
9.1.1. The network 343
9.1.2. Physical Channel 346
9.1.3. Wireless Communication 349
9.1.4. Radio 350
9.2. Controller Area Network (CAN) 352
9.2.1. The Fundamentals of CAN 352
9.2.2. Texas Instruments TM4C CAN 355
9.3. Embedded Internet 359
9.3.1. Abstraction 360
9.3.2. Message Protocols 362
9.3.3. Ethernet Physical Layer 363
9.3.4. Ethernet on the TM4C1294 365
9.4. Internet of Things 367
9.4.1. Basic Concepts 367
9.4.2. UDP and TCP Packets 369
9.4.3. Web server 370
9.4.4. UDP communication over WiFi 372
9.4.5. Other CC3100 Applications 376
9.4. Bluetooth Fundamentals 378
9.4.1. Bluetooth Protocol Stack 380
9.4.2. Client-server Paradigm 383
9.5. CC2650 Solutions 384
9.5.1. CC2650 Microcontroller 384
9.5.2. Single Chip Solution, CC2650 LaunchPad 387
9.6. Network Processor Interface (NPI) 387
9.6.1. Overview 387

9.6.2. Services and Characteristics....................390

9.6.3. Advertising....................392

9.6.4. Read and Write Indications....................393

9.7. Application Layer Protocols for Embedded Systems 394

9.7.1. CoAP....................394

9.7.2 MQTT....................396

9.8. Exercises.................... 397

10. Robotic Systems.................... 399

10.1. Introduction to Digital Control Systems.................... 399

10.2. Binary Actuators.................... 400

10.2.1. Electrical Interface....................400

10.2.2. DC Motor Interface with PWM....................404

10.3. Sensors 408

10.4. Odometry.................... 409

10.5. Simple Closed-Loop Control Systems.................... 412

10.6. PID Controllers 415

10.6.1. General Approach to a PID Controller....................415

10.6.2. Design Process for a PID Controller....................418

10.7. Fuzzy Logic Control 422

10.8. Exercises.................... 431

Appendix 1. Glossary 433

Appendix 2. Solutions to Checkpoints 451

Index 457

Reference Material.................... 466

Preface to The Fourth Edition

There are two major additions to this fourth edition. First, this version supports both the TM4C and the MSP432 architectures. The material for the LM3S series has been removed. Volumes 1 and 2 focused on the hardware and software aspects I/O interfacing. In this volume we provide a set of low level device drivers allowing this volume to focus on real-time operating systems, digital signal processing, control systems, and the internet of things. The second addition is Bluetooth Low Energy (BLE), which will be implemented by interfacing a CC2650, in a similar manner with which IEEE802.11b wifi is implemented in this book using the CC3100. Running on the CC2650 will be an application programmer interface called Simple Network Processor (SNP). SNP allows the TM4C123/MSP432 microcontroller to implement BLE using a simple set of UART messaging. Off-loading the BLE functions to the CC2650 allows the target microcontroller to implement system level functions without the burden of satisfying the real-time communication required by Bluetooth.

Preface to Volume 3

Embedded systems are a ubiquitous component of our everyday lives. We interact with hundreds of tiny computers every day that are embedded into our houses, our cars, our toys, and our work. As our world has become more complex, so have the capabilities of the microcontrollers embedded into our devices. The ARM Cortex-M family represents the new class of microcontrollers much more powerful than the devices available ten years ago. The purpose of this book is to present the design methodology to train young engineers to understand the basic building blocks that comprise devices like a cell phone, an MP3 player, a pacemaker, antilock brakes, and an engine controller.

This book is the third in a series of three books that teach the fundamentals of embedded systems as applied to the ARM Cortex-M family of microcontrollers. This third volume is primarily written for senior undergraduate or first-year graduate electrical and computer engineering students. It could also be used for professionals wishing to design or deploy a real-time operating system onto an ARM platform. The first book Embedded Systems: Introduction to ARM Cortex-M Microcontrollers is an introduction to computers and interfacing focusing on assembly language and C programming. The second book Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers focuses on interfacing and the design of embedded systems. This third book is an advanced book focusing on operating systems, high-speed interfacing, control systems, and robotics.

An embedded system is a system that performs a specific task and has a computer embedded inside. A system is comprised of components and interfaces connected together for a common purpose. This book presents components, interfaces and methodologies for building systems. Specific topics include microcontrollers, design, verification, hardware/software synchronization, interfacing devices to the computer, timing diagrams, real-time operating systems, data collection and processing, motor control, analog filters, digital filters, and real-time signal processing.

In general, the area of embedded systems is an important and growing discipline within electrical and computer engineering. In the past, the educational market of embedded systems has been dominated by simple microcontrollers like the PIC, the 9S12, and the 8051. This is because of their market share, low cost, and historical dominance. However, as problems become more complex, so must the systems that solve them. A number of embedded system paradigms must shift in order to accommodate this growth in complexity. First, the number of calculations per second will increase from millions/sec to billions/sec. Similarly, the number of lines of software code will also increase from thousands to millions. Thirdly, systems will involve multiple microcontrollers supporting many simultaneous operations. Lastly, the need for system verification will continue to grow as these systems are deployed into safety critical applications. These changes are more than a simple growth in size and bandwidth. These systems must employ parallel programming, high-speed synchronization, real-time operating systems, fault tolerant design, priority interrupt handling, and networking. Consequently, it will be important to provide our students with these types of design experiences. The ARM platform is both low cost and provides the high performance features required in future embedded systems. Although the ARM market share is large and will continue to grow. Furthermore, students trained on the ARM will be equipped to design systems across the complete spectrum from simple to complex. The purpose of writing these three books at this time is to bring engineering education into the 21st century.

This book employs many approaches to learning. It will not include an exhaustive recapitulation of the information in data sheets. First, it begins with basic fundamentals, which allows the reader to solve new problems with new technology. Second, the book presents many detailed design examples. These examples illustrate the process of design. There are multiple structural components that assist learning. Checkpoints, with answers in the back, are short easy to answer questions providing immediate feedback while reading. Homework problems, which typically are simpler than labs, provide more learning opportunities. The book includes an index and a glossary so that information can be searched. The most important learning experiences in a class like this are of course the laboratories. More detailed lab descriptions are available on the web. Specifically for Volume 1, look at the lab assignments for EE319K. For Volume 2 refer to the EE445L labs, and for this volume, look at the lab assignments for EE445M/EE380L.6.

There is a web site accompanying this book **http://users.ece.utexas.edu/~valvano/arm**. Posted here are ARM Keil™ uVision® and Texas Instruments Code Composer Studio™ projects for each of the example programs in the book. You will also find data sheets and Excel spreadsheets relevant to the material in this book.

The book will cover embedded systems for ARM® Cortex™-M microcontrollers with specific details on the TM4C123, TM4C1294, and MSP432. Most of the topics can be run on any Texas Instruments Cortex M microcontroller. In these books the terms **MSP432** and **TM4C** will refer to any of the Texas Instruments ARM Cortex-M based microcontrollers. Although the solutions are specific for the **MSP432** and **TM4C** families, it will be possible to use these books for other ARM derivatives.

Acknowledgements

I owe a wonderful debt of gratitude to Daniel Valvano. He wrote and tested most of the software examples found in these books. Secondly, he maintains the example web site, **http://users.ece.utexas.edu/~valvano/arm.** Lastly, he meticulously proofread this manuscript.

Many shared experiences contributed to the development of this book. First I would like to acknowledge the many excellent teaching assistants I have had the pleasure of working with. Some of these hard-working, underpaid warriors include Pankaj Bishnoi, Rajeev Sethia, Adson da Rocha, Bao Hua, Raj Randeri, Santosh Jodh, Naresh Bhavaraju, Ashutosh Kulkarni, Bryan