Real-Time Embedded Multithreading
Using ThreadX
Real-Time Embedded Multithreading
Using ThreadX

Second Edition

Edward L. Lamie
à mes ancêtres québécois
# Contents

**Preface** ............................................................................................................................... xv

**Chapter 1: Embedded and Real-time Systems** ................................................................... 1

1.1 Introduction .................................................................................................................. 1
1.2 What is an Embedded System? ................................................................................. 1
1.3 Characteristics of Embedded Systems ....................................................................... 2
1.4 Real-Time Systems .................................................................................................... 2
1.5 Real-Time Operating Systems and Real-Time Kernels ......................................... 3
1.6 Processes, Tasks, and Threads .................................................................................. 4
1.7 Architecture of Real-Time Systems .......................................................................... 5
1.8 Embedded Systems Development ............................................................................. 7
1.9 Key Terms and Phrases ............................................................................................. 8

**Chapter 2: First Look At a System Using An RTOS** ...................................................... 9

2.1 Operating Environment ............................................................................................. 9
2.2 Installation of the ThreadX Demonstration System .................................................. 9
2.3 Sample System with Two Threads ............................................................................ 9
2.4 Creating the ThreadX Objects................................................................................. 12
2.5 Compiling and Executing the Sample System ........................................................ 14
2.6 Analysis of the System and the Resulting Output................................................... 14
2.7 Listing of 02_sample_system.c ............................................................................... 14
2.8 Key Terms and Phrases ........................................................................................... 19
2.9 Problems.................................................................................................................. 20

**Chapter 3: RTOS Concepts and Definitions** .................................................................. 21

3.1 Introduction.................................................................................................................. 21
3.2 Priorities ..................................................................................................................... 21
3.3 Ready Threads and Suspended Threads ................................................................... 22
3.4 Preemptive, Priority-Based Scheduling .................................................................. 23
3.5 Round-Robin Scheduling ................................................................. 24
3.6 Determinism ................................................................................ 25
3.7 Kernel .......................................................................................... 26
3.8 RTOS .......................................................................................... 26
3.9 Context Switch ............................................................................. 26
3.10 Time-Slice ................................................................................ 26
3.11 Interrupt Handling ...................................................................... 27
3.12 Thread Starvation ....................................................................... 27
3.13 Priority Inversion ....................................................................... 27
3.14 Priority Inheritance .................................................................... 28
3.15 Preemption-Threshold ................................................................. 29
3.16 Key Terms and Phrases ............................................................. 30
3.17 Problems .................................................................................... 30

Chapter 4: RTOS Building Blocks For System Development ................. 31
4.1 Introduction ................................................................................. 31
4.2 Defining Public Resources ......................................................... 31
4.3 ThreadX Data Types ................................................................. 32
4.4 Thread ....................................................................................... 32
4.5 Memory Pools .......................................................................... 34
4.6 Application Timer ....................................................................... 36
4.7 Mutex ......................................................................................... 37
4.8 Counting Semaphore .................................................................. 38
4.9 Event Flags Group ...................................................................... 39
4.10 Message Queue ......................................................................... 40
4.11 Summary of Thread Synchronization and Communication Components .... 41
4.12 Key Terms and Phrases ............................................................ 42
4.13 Problems .................................................................................. 43

Chapter 5: The Thread—the Essential Component .................................... 45
5.1 Introduction ................................................................................. 45
5.2 Thread Control Block ................................................................. 45
5.3 Summary of Thread Services ...................................................... 47
5.4 Thread Creation .......................................................................... 48
5.5 Thread Deletion ......................................................................... 54
5.6 Identify Thread .......................................................................... 55
5.7 Get Thread Information ..........................................................................................55
5.8 Preemption-Threshold Change ...............................................................................56
5.9 Priority Change .......................................................................................................57
5.10 Relinquish Control ...............................................................................................58
5.11 Resume Thread Execution .....................................................................................58
5.12 Thread Sleep .........................................................................................................59
5.13 Suspend Thread Execution ....................................................................................59
5.14 Terminate Application Thread ...............................................................................59
5.15 Time-Slice Change ...............................................................................................60
5.16 Abort Thread Suspension ....................................................................................61
5.17 Thread Notification Services ...............................................................................61
5.18 Execution Overview .............................................................................................61
5.19 Thread States .......................................................................................................63
5.20 Thread Design ......................................................................................................64
5.21 Thread Internals ..................................................................................................67
5.22 Overview .............................................................................................................69
5.23 Key Terms and Phrases .......................................................................................69
5.24 Problems ..............................................................................................................70

Chapter 6: Mutual Exclusion Challenges and Considerations ......................................73
6.1 Introduction .........................................................................................................73
6.2 Protecting a Critical Section ...............................................................................73
6.3 Providing Exclusive Access to Shared Resources .................................................74
6.4 Mutex Control Block ............................................................................................75
6.5 Summary of Mutex Services ...............................................................................75
6.6 Creating a Mutex ..................................................................................................76
6.7 Deleting a Mutex ..................................................................................................77
6.8 Obtaining Ownership of a Mutex .........................................................................78
6.9 Retrieving Mutex Information ..............................................................................79
6.10 Prioritizing the Mutex Suspension List .................................................................79
6.11 Releasing Ownership of a Mutex .......................................................................81
6.12 Avoiding the Deadly Embrace ............................................................................81
6.13 Sample System Using a Mutex to Protect Critical Sections ...............................83
6.14 Output Produced by Sample System ..................................................................87
6.15 Listing for 06_sample_system.c .......................................................................90
Contents

6.16 Mutex Internals ................................................................. 96
6.17 Overview .......................................................................... 97
6.18 Key Terms and Phrases .................................................... 98
6.19 Problems .......................................................................... 98

Chapter 7: Memory Management: Byte Pools And Block Pools ....................... 101
7.1 Introduction ...................................................................... 101
7.2 Summary of Memory Byte Pools ........................................... 102
7.3 Memory Byte Pool Control Block ........................................ 103
7.4 Pitfalls of Memory Byte Pools ............................................ 103
7.5 Summary of Memory Byte Pool Services ............................. 104
7.6 Creating a Memory Byte Pool ............................................. 105
7.7 Allocating from a Memory Byte Pool ................................. 105
7.8 Deleting a Memory Byte Pool .............................................. 107
7.9 Retrieving Memory Byte Pool Information .......................... 107
7.10 Prioritizing a Memory Byte Pool Suspension List .................. 108
7.11 Releasing Memory to a Byte Pool ..................................... 109
7.12 Memory Byte Pool Example—Allocating Thread Stacks ............. 110
7.13 Memory Byte Pool Internals ............................................ 111
7.14 Summary of Memory Block Pools ...................................... 112
7.15 Memory Block Pool Control Block .................................... 114
7.16 Summary of Memory Block Pool Services .......................... 115
7.17 Creating a Memory Block Pool .......................................... 116
7.18 Allocating a Memory Block Pool ....................................... 117
7.19 Deleting a Memory Block Pool .......................................... 118
7.20 Retrieving Memory Block Pool Information ........................ 119
7.21 Prioritizing a Memory Block Pool Suspension List ................ 119
7.22 Releasing a Memory Block .............................................. 120
7.23 Memory Block Pool Example—Allocating Thread Stacks ............. 121
7.24 Memory Block Pool Internals ............................................ 122
7.25 Overview and Comparison ................................................ 123
7.26 Key Terms and Phrases .................................................... 124
7.27 Problems .......................................................................... 124

Chapter 8: Internal System Clock and Application Timers ................................. 127
8.1 Introduction ...................................................................... 127
Chapter 9: Event Notification and Synchronization With Counting Semaphores

9.1 Introduction ................................................................. 151
9.2 Counting Semaphore Control Block ............................. 153
9.3 Avoiding Deadly Embrace ............................................ 154
9.4 Avoiding Priority Inversion ........................................... 154
9.5 Summary of Counting Semaphore Services ................ 154
9.6 Creating a Counting Semaphore ................................... 154
9.7 Deleting a Counting Semaphore ................................. 155
9.8 Getting an Instance of a Counting Semaphore ............. 156
9.9 Retrieving Information About Counting Semaphores .... 157
9.10 Prioritizing a Counting Semaphore Suspension List ...... 158
9.11 Placing an Instance in a Counting Semaphore .......... 158
9.12 Placing an Instance in a Semaphore Using a Ceiling ... 159
9.13 Semaphore Notification and Event-Chaining .............. 160
9.14 Comparing a Counting Semaphore with a Mutex ...... 160
9.15 Sample System Using a Binary Semaphore in Place of a Mutex ........................ 161
9.16 Listing for 9a_sample_system.c ................................. 164
9.17 Sample System Using a Counting Semaphore in a Producer-Consumer Application .......... 170
## Contents

9.18 Listing for 9b_sample_system.c .................................................................173
9.19 Counting Semaphore Internals .................................................................179
9.20 Overview ....................................................................................................180
9.21 Key Terms and Phrases .........................................................................180
9.22 Problems .................................................................................................181

### Chapter 10: Synchronization of Threads Using Event Flags Groups ..........183
10.1 Introduction ..............................................................................................183
10.2 Event Flags Group Control Block ..........................................................184
10.3 Summary of Event Flags Group Control Services ..................................185
10.4 Creating an Event Flags Group ...............................................................185
10.5 Deleting an Event Flags Group ...............................................................187
10.6 Getting Event Flags from an Event Flags Group .....................................187
10.7 Retrieving Information about an Event Flags Group .............................191
10.8 Setting Event Flags in an Event Flags Group ..........................................192
10.9 Event Flags Group Notification and Event-Chaining .............................194
10.10 Sample System Using an Event Flags Group to Synchronize Two Threads 194
10.11 Listing for 10_sample_system.c ............................................................198
10.12 Event Flags Group Internals .................................................................204
10.13 Overview ...............................................................................................205
10.14 Key Terms and Phrases .......................................................................206
10.15 Problems ..............................................................................................206

### Chapter 11: Thread Communication With Message Queues ..................209
11.1 Introduction .............................................................................................209
11.2 Message Queue Control Block ..............................................................211
11.3 Summary of Message Queue Services ..................................................212
11.4 Creating a Message Queue ....................................................................212
11.5 Sending a Message to a Message Queue ..............................................213
11.6 Receiving a Message from a Message Queue ......................................214
11.7 Deleting a Message Queue ....................................................................215
11.8 Flushing the Contents of a Message Queue ..........................................216
11.9 Sending a Message to the Front of a Message Queue ..........................216
11.10 Retrieving Message Queue Information .............................................217
11.11 Prioritizing a Message Queue Suspension List ..................................218
11.12 Message Queue Notification and Event-Chaining ...............................219
Contents  xiii

11.13 Sample System Using a Message Queue for Interthread Communication ........... 219
11.14 Listing for 11_sample_system.c ................................................................. 222
11.15 Message Queue Internals .............................................................................. 229
11.16 Overview ....................................................................................................... 229
11.17 Key Terms and Phrases ............................................................................... 230
11.18 Problems......................................................................................................... 231

Chapter 12: Advanced Topics ................................................................................. 233
12.1 Introduction ..................................................................................................... 233
12.2 Event-Chaining ................................................................................................ 233
12.3 Sample System Using Event-Chaining ......................................................... 237
12.4 Listing for 12_sample_system.c ................................................................. 240
12.5 Run-Time Performance Metrics .................................................................... 247
12.6 Key Terms and Phrases ............................................................................... 259

Chapter 13: Case Study: Designing A Multithreaded System................................. 261
13.1 Introduction ..................................................................................................... 261
13.2 Statement of Problem .................................................................................... 263
13.3 Analysis of the Problem ............................................................................... 266
13.4 Design of the System .................................................................................... 268
13.5 Implementation .............................................................................................. 273
13.6 Listing of VAM System ................................................................................. 289
13.7 Overview ......................................................................................................... 300

Appendix A: Memory Block Pool Services ................................................................ A1
Appendix B: Memory Byte Pool Services ................................................................. B1
Appendix C: Event Flags Group Services ................................................................. C1
Appendix D: Interrupt Control Service ................................................................... D1
Appendix E: Mutex Services ................................................................................... E1
Appendix F: Message Queue Services .................................................................... F1
Appendix G: Counting Semaphore Services ......................................................... G1
Appendix H: Thread Services ................................................................................. H1
Appendix I: Internal System Clock Services ................................................................. I1
Appendix J: Application Timer Services .................................................................... J1
Appendix K: ThreadX API ........................................................................................... K1
Appendix L: ThreadX and the ARM Microprocessor .................................................. L1
Appendix M: ThreadX and the ColdFire Microprocessor .......................................... M1
Appendix N: ThreadX and the MIPS Microprocessor ................................................. N1
Appendix O: ThreadX and the PowerPC Microprocessor .......................................... O1
Index ......................................................................................................................... Index-1
Preface

The first edition of this book covered ThreadX\textsuperscript{1} (version 4) as well as information about the ARM® processor relative to ThreadX. The second edition of this book has been enhanced to address the features of ThreadX (version 5) and it includes a variety of concepts including real-time event-chaining\textsuperscript{2} and real-time performance metrics. Chapters 1 through 4 cover fundamental terminology and concepts of embedded and real-time systems. Chapters 5 through 11 investigate major ThreadX services and analyze several sample systems as well as solutions to classical problem areas. Chapter 12 is devoted to a study of advanced topics that include event-chaining and performance metrics. Chapter 13 contains a case study that illustrates how a system could be developed and implemented. Appendices A through K contain details of the ThreadX API and these appendices serve as a compact guide to all the available services. Appendices L through O contain information about the ARM\textsuperscript{3}, Coldfire\textsuperscript{4}, MIPS\textsuperscript{5}, and PowerPC\textsuperscript{6} processors as used with ThreadX. Each of these appendices contains technical information, register set information, processor modes, exception and interrupt handling, thread scheduling, and context switching.

Embedded systems are ubiquitous. These systems are found in most consumer electronics, automotive, government, military, communications, and medical equipment. Most individuals in developed countries have many such systems and use them daily, but relatively few people realize that these systems actually contain embedded computer systems. Although the field of embedded systems is young, the use and importance of these systems is increasing, and the field is rapidly growing and maturing.

\textsuperscript{1}ThreadX is a registered trademark of Express Logic, Inc. The ThreadX API, associated data structures, and data types are copyrights of Express Logic, Inc.
\textsuperscript{2}Event-chaining is a registered trademark of Express Logic, Inc.
\textsuperscript{3}ARM is a registered trademark of ARM Limited
\textsuperscript{4}ColdFire is a registered trademark of Freescale, Inc.
\textsuperscript{5}MIPS is a registered trademark of MIPS Processors, Inc.
\textsuperscript{6}PowerPC is a registered trademark of IBM Corporation
This book is intended for persons who develop embedded systems, or for those who would like to know more about the process of developing such systems. Although embedded systems developers are typically software engineers or electrical engineers, many people from other disciplines have made significant contributions to this field. This book is specifically targeted toward embedded applications that must be small, fast, reliable, and deterministic.\(^7\)

I assume the reader has a programming background in C or C++, so we won’t devote any time to programming fundamentals. Depending on the background of the reader, the chapters of the book may be read independently.

There are several excellent books written about embedded systems. However, most of these books are written from a generalist point of view. This book is unique because it is based on embedded systems development using a typical commercial RTOS, as well as widely used microprocessors. This approach has the advantage of providing specific knowledge and techniques, rather than generic concepts that must be converted to your specific system. Thus, you can immediately apply the topics in this book to your development efforts.

Because an actual RTOS is used as the primary tool for embedded application development, there is no discussion about the merits of building your own RTOS or forgoing an RTOS altogether. I believe that the relatively modest cost of a commercial RTOS provides a number of significant advantages over attempts to “build your own.” For example, most commercial RTOS companies have spent years refining and optimizing their systems. Their expertise and product support may play an important role in the successful development of your system.

As noted previously, the RTOS chosen for use in this book is ThreadX (version 5). This RTOS was selected for a variety of reasons including reliability, ease of use, low cost, widespread use, and the maturity of the product due to the extensive experience of its developers. This RTOS contains most of the features found in contemporary RTOSes, as well as several advanced features that are not. Another notable feature of this RTOS is the consistent and readable coding convention used within its application programming interface (API). Developing applications is highly intuitive because of the logical approach of the API.

\(^7\)Such systems are sometimes called deeply embedded systems.
Although I chose the C programming language for this book, you could use C++ instead for any of the applications described in this book.

There is a CD included with this book that contains a limited ThreadX\textsuperscript{8} system. You may use this system to perform your own experiments, run the included demonstration system, and experiment with the projects described throughout the book.

Typographical conventions are used throughout this book so that key concepts are communicated easily and unambiguously. For example, keywords such as `main` or `int` are displayed in a distinctive typeface, whether these keywords are in a program or appear in the discussion about a program. This typeface is also used for all program segment listings or when actual input or output is illustrated. When an identifier name such as `MyVar` is used in the narrative portion of the book, it will appear in italics. The italics typeface will also be used when new topics are introduced or to provide emphasis.

\textsuperscript{8}Express Logic, Inc. has granted permission to use this demonstration version of ThreadX for the sample systems and the case study in this book.