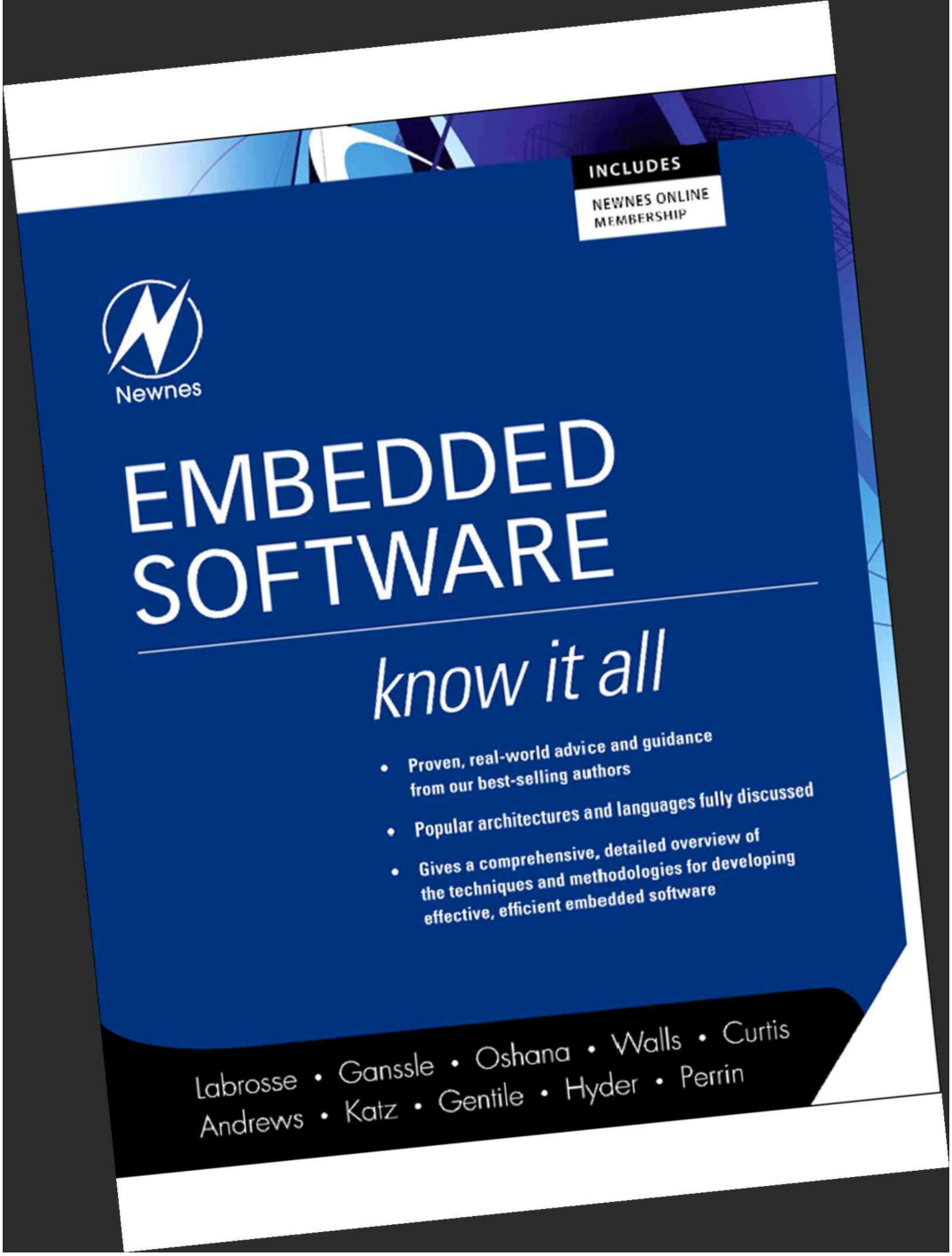
INCLUDES
NEWNES ONLINE MEMBERSHIP
Newnes
EMBEDDED SOFTWARE
know it all
• Proven, real-world advice and guidance from our best-selling authors
• Popular architectures and languages fully discussed
• Gives a comprehensive, detailed overview of the techniques and methodologies for developing effective, efficient embedded software
Labrosse • Ganssle • Oshana • Walls • Curtis
Andrews • Katz • Gentile • Hyder • Perrin

Embedded Software

Newnes Know It All *Series*

PIC Microcontrollers: Know It All

Lucio Di Jasio, Tim Wilmshurst, Dogan Ibrahim, John Morton, Martin Bates, Jack Smith, D.W. Smith, and Chuck Hellebuyck

ISBN: 978-0-7506-8615-0

Embedded Software: Know It All

Jean Labrosse, Jack Ganssle, Tammy Noergaard, Robert Oshana, Colin Walls, Keith Curtis, Jason Andrews, David J. Katz, Rick Gentile, Kamal Hyder, and Bob Perrin

ISBN: 978-0-7506-8583-2

Embedded Hardware: Know It All

Jack Ganssle, Tammy Noergaard, Fred Eady, Lewin Edwards, David J. Katz, Rick Gentile, Ken Arnold, Kamal Hyder, and Bob Perrin

ISBN: 978-0-7506-8584-9

Wireless Networking: Know It All

Praphul Chandra, Daniel M. Dobkin, Alan Bensky, Ron Olexa, David Lide, and Farid Dowla

ISBN: 978-0-7506-8582-5

RF & Wireless Technologies: Know It All

Bruce Fette, Roberto Aiello, Praphul Chandra, Daniel Dobkin, Alan Bensky, Douglas Miron, David Lide, Farid Dowla, and Ron Olexa

ISBN: 978-0-7506-8581-8

For more information on these and other Newnes titles visit: ***www.newnespress.com***

Embedded Software

Jean Labrosse
Jack Ganssle
Tammy Noergaard
Robert Oshana
Colin Walls
Keith Curtis
Jason Andrews
David J. Katz
Rick Gentile
Kamal Hyder
Bob Perrin

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Newnes is an imprint of Elsevier

Newnes is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Library of Congress Cataloging-in-Publication Data
Embedded software/Jean Labrosse . . . [et al.]. – 1st ed.
p. cm. – (Newnes know it all series)
Includes bibliographical references and index.
ISBN-13: 978-0-7506-8583-2 (pbk. : alk. paper) 1. Embedded computer systems–Programming. 2. Computer software–Development. I. Labrosse, Jean J.
TK7895.E42E588 2008
005.26–dc22

2007023369

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-0-7506-8583-2

For information on all Newnes publications
visit our Web site at www.books.elsevier.com

07 08 09 10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contents

About the Authors ..... *x*
Introduction ..... *xiii*

Chapter 1: Basic Embedded Programming Concepts *1*
1.1 Numbering Systems 2
1.2 Signed Binary Numbers 5
1.3 Data Structures 13
1.4 Communications Protocols 29
1.5 Mathematics 37
1.6 Numeric Comparison 46
1.7 State Machines 59
1.8 Multitasking 74

Chapter 2: Device Drivers *85*
2.1 In This Chapter 85
2.2 Example 1: Device Drivers for Interrupt-Handling 89
2.3 Example 2: Memory Device Drivers 110
2.4 Example 3: Onboard Bus Device Drivers 134
2.5 Board I/O Driver Examples 143
2.6 Summary 168

Chapter 3: Embedded Operating Systems *169*
3.1 In This Chapter 169
3.2 What Is a Process? 175
3.3 Multitasking and Process Management 177
3.4 Memory Management 213
3.5 I/O and File System Management 230
3.6 OS Standards Example: POSIX (Portable Operating System Interface) 232
3.7 OS Performance Guidelines 235
3.8 OSes and Board Support Packages (BSPs) 237
3.9 Summary 239

Chapter 4: Networking *241*
4.1 Introduction to the RCM3200 Rabbit Core 243
4.2 Introduction to the Dynamic C Development Environment 244
4.3 Brief Introduction to Dynamic C Libraries 246
4.4 Memory Spaces in Dynamic C 247
4.5 How Code Is Compiled and Run 256
4.6 Setting Up a PC as an RCM3200 Development System 259
4.7 Time to Start Writing Code! 259
4.8 Embedded Networks 274
4.9 Dynamic C Support for Networking Protocols 275
4.10 Typical Network Setup 279
4.11 Setting Up a Core Module's Network Configuration 282
4.12 Project 1: Bringing Up a Rabbit Core Module for Networking 288
4.13 The Client Server Paradigm 293
4.14 The Berkeley Sockets Interface 294
4.15 Using TCP versus UDP in an Embedded Application 298
4.16 Important Dynamic C Library Functions for Socket Programming 300
4.17 Project 2: Implementing a Rabbit TCP/IP Server 303
4.18 Project 3: Implementing a Rabbit TCP/IP Client 311
4.19 Project 4: Implementing a Rabbit UDP Server 322
4.20 Some Useful (and Free!) Networking Utilities 328
4.21 Final Thought 331

Chapter 5: Error Handling and Debugging *333*
5.1 The Zen of Embedded Systems Development and Troubleshooting 333
5.2 Avoid Debugging Altogether—Code Smart 340
5.3 Proactive Debugging 341
5.4 Stacks and Heaps 342
5.5 Seeding Memory 344
5.6 Wandering Code 346
5.7 Special Decoders 347
5.8 MMUs 348
5.9 Conclusion 349
5.10 Implementing Downloadable Firmware with Flash Memory 350
5.11 The Microprogrammer 350
5.12 Advantages of Microprogrammers 351
5.13 Disadvantages of Microprogrammers 351
5.14 Receiving a Microprogrammer 352
5.15 A Basic Microprogrammer 354
5.16 Common Problems and Their Solutions 355
5.17 Hardware Alternatives 362
5.18 Memory Diagnostics 364
5.19 ROM Tests 365

5.20 RAM Tests 367
5.21 Nonvolatile Memory 372
5.22 Supervisory Circuits 372
5.23 Multibyte Writes 374
5.24 Testing 378
5.25 Conclusion 378
5.26 Building a Great Watchdog 379
5.27 Internal WDTs 382
5.28 External WDTs 384
5.29 Characteristics of Great WDTs 386
5.30 Using an Internal WDT 389
5.31 An External WDT 391
5.32 WDTs for Multitasking 393
5.33 Summary and Other Thoughts 395

Chapter 6: Hardware/Software Co-Verification 399
6.1 Embedded System Design Process 399
6.2 Verification and Validation 401
6.3 Human Interaction 403
6.4 Co-Verification 405

Chapter 7: Techniques for Embedded Media Processing 443
7.1 A Simplified Look at a Media Processing System 445
7.2 System Resource Partitioning and Code Optimization 451
7.3 Event Generation and Handling 452
7.4 Programming Methodology 455
7.5 Architectural Features for Efficient Programming 456
7.6 Compiler Considerations for Efficient Programming 465
7.7 System and Core Synchronization 472
7.8 Memory Architecture—the Need for Management 476
7.9 Physics of Data Movement 488
7.10 Media Processing Frameworks 495
7.11 Defining Your Framework 497
7.12 Asymmetric and Symmetric Dual-Core Processors 505
7.13 Programming Models 507
7.14 Strategies for Architecting a Framework 510
7.15 Other Topics in Media Frameworks 523

Chapter 8: DSP in Embedded Systems 529
8.1 Overview of Embedded Systems and Real-Time Systems 536
8.2 Real-Time Systems 536
8.3 Hard Real-Time and Soft Real-Time Systems 537
8.4 Efficient Execution and the Execution Environment 541

8.5 Challenges in Real-Time System Design 542
8.6 Summary 553
8.7 Overview of Embedded Systems Development Life Cycle Using DSP 554
8.8 The Embedded System Life Cycle Using DSP 554
8.9 Optimizing DSP Software 580
8.10 What Is Optimization? 580
8.11 The Process 581
8.12 Make the Common Case Fast 584
8.13 Make the Common Case Fast—DSP Architectures 584
8.14 Make the Common Case Fast—DSP Algorithms 587
8.15 Make the Common Case Fast—DSP Compilers 588
8.16 An In-Depth Discussion of DSP Optimization 595
8.17 Direct Memory Access 595
8.18 Using DMA 596
8.19 Loop Unrolling 604
8.20 Software Pipelining 610
8.21 More on DSP Compilers and Optimization 620
8.22 Programmer Helping Out the Compiler 633
8.23 Profile-Based Compilation 646
8.24 References 653

Chapter 9: Practical Embedded Coding Techniques 655

9.1 Reentrancy 655
9.2 Atomic Variables 656
9.3 Two More Rules 658
9.4 Keeping Code Reentrant 659
9.5 Recursion 661
9.6 Asynchronous Hardware/Firmware 661
9.7 Race Conditions 662
9.8 Options 664
9.9 Other RTOSes 665
9.10 Metastable States 666
9.11 Firmware, Not Hardware 668
9.12 Interrupt Latency 671
9.13 Taking Data 674
9.14 Understanding Your C Compiler: How to Minimize Code Size 677
9.15 Modern C Compilers 677
9.16 Tips on Programming 687
9.17 Final Notes 695
9.18 Acknowledgments 696

Chapter 10: Development Technologies and Trends *697*
10.1 How to Choose a CPU for Your System on Chip Design 697
10.2 Emerging Technology for Embedded Systems Software Development 700
10.3 Making Development Tool Choices 707
10.4 Eclipse—Bringing Embedded Tools Together 721
10.5 Embedded Software and UML 725
10.6 Model-based Systems Development with xtUML 739
10.7 The Future 743
Index *745*