

# Embedded Linux Primer

Second Edition

A Practical Real-World Approach



Christopher Hallinan

---

# Embedded Linux Primer, Second Edition

Download at [www.wowebook.com](http://www.wowebook.com)

---

# Embedded Linux Primer, Second Edition

A Practical, Real-World Approach

Christopher Hallinan



PRENTICE  
HALL

Prentice Hall Professional Technical Reference

Upper Saddle River, NJ · Boston · Indianapolis · San Francisco

New York · Toronto · Montreal · London · Munich · Paris · Madrid

Capetown · Sydney · Tokyo · Singapore · Mexico City

Download at [www.wowebook.com](http://www.wowebook.com)

---

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales  
(800) 382-3419  
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales  
international@pearson.com

Visit us on the Web: [informit.com/aw](http://informit.com/aw)

*Library of Congress Cataloging-in-Publication Data:*

Hallinan, Christopher.

Embedded Linux primer : a practical real-world approach / Christopher Hallinan.

p. cm.

ISBN 978-0-13-701783-6 (hardback : alk. paper) 1. Linux. 2. Operating systems (Computers) 3. Embedded computer systems--Programming. I. Title.

QA76.76.O63H34462 2011  
005.4'32--dc22

2010032891

Copyright © 2011 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax (617) 671-3447

ISBN-13: 978-0-137-01783-6

ISBN-10: 0-137-01783-9

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.

First printing September 2010

Editor-in-Chief: Mark L. Taub

Executive Acquisitions Editor: Debra Williams Cauley

Development Editor: Michael Thurston

Managing Editor: Kristy Hart

Project Editors: Alexandra Maurer and Jovana San Nicolas-Shirley

Copy Editor: Gayle Johnson

Indexer: Heather McNeill

Proofreader: Sarah Kearns

Technical Reviewers: Robert P.J. Day, Kurt Lloyd, Jon Masters, Sandra Terrace, and Mark A. Yoder

Publishing Coordinator: Kim Boedigheimer

Cover Designer: Alan Clements

Compositor: Tricia Bronkella

Download at [www.wowebook.com](http://www.wowebook.com)

---

*To my grandmother Edythe Diorio Ricciuti, who, at one hundred  
and five and counting, continues to provide inspiration to her  
loving family through her deep faith, unshakable moral compass,  
and selfless dedication to others.*

---

*This page intentionally left blank*

---

# Contents

Foreword for the First Edition .....	xxv
Foreword for the Second Edition.....	xxvi
Preface.....	xxvii
Acknowledgments for the First Edition .....	xxxiii
Acknowledgments for the Second Edition.....	xxxv
About the Author.....	xxxvi
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Why Linux?.....	2
1.2 Embedded Linux Today .....	3
1.3 Open Source and the GPL .....	3
1.3.1 Free Versus Freedom .....	4
1.4 Standards and Relevant Bodies .....	5
1.4.1 Linux Standard Base .....	5
1.4.2 Linux Foundation .....	6
1.4.3 Carrier-Grade Linux .....	6
1.4.4 Mobile Linux Initiative: Moblin. ....	7
1.4.5 Service Availability Forum. ....	7
1.5 Summary. ....	8
1.5.1 Suggestions for Additional Reading. ....	8

<b>Chapter 2 The Big Picture . . . . .</b>	<b>9</b>
2.1 Embedded or Not? . . . . .	10
2.1.1 BIOS Versus Bootloader . . . . .	11
2.2 Anatomy of an Embedded System. . . . .	12
2.2.1 Typical Embedded Linux Setup . . . . .	13
2.2.2 Starting the Target Board . . . . .	14
2.2.3 Booting the Kernel. . . . .	16
2.2.4 Kernel Initialization: Overview . . . . .	18
2.2.5 First User Space Process: <code>init</code> . . . . .	19
2.3 Storage Considerations. . . . .	20
2.3.1 Flash Memory. . . . .	20
2.3.2 NAND Flash . . . . .	22
2.3.3 Flash Usage . . . . .	23
2.3.4 Flash File Systems . . . . .	24
2.3.5 Memory Space . . . . .	25
2.3.6 Execution Contexts. . . . .	26
2.3.7 Process Virtual Memory. . . . .	28
2.3.8 Cross-Development Environment. . . . .	30
2.4 Embedded Linux Distributions. . . . .	32
2.4.1 Commercial Linux Distributions . . . . .	33
2.4.2 Do-It-Yourself Linux Distributions. . . . .	33
2.5 Summary. . . . .	34
2.5.1 Suggestions for Additional Reading. . . . .	35
<b>Chapter 3 Processor Basics . . . . .</b>	<b>37</b>
3.1 Stand-Alone Processors. . . . .	38
3.1.1 IBM 970FX. . . . .	39
3.1.2 Intel Pentium M. . . . .	39



3.1.3 Intel Atom™ ..... 40

3.1.4 Freescale MPC7448 ..... 40

3.1.5 Companion Chipsets ..... 41

3.2 Integrated Processors: Systems on Chip. .... 43

3.2.1 Power Architecture..... 44

3.2.2 Freescale Power Architecture . .... 44

3.2.3 Freescale PowerQUICC I..... 45

3.2.4 Freescale PowerQUICC II ..... 46

3.2.5 PowerQUICC II Pro..... 47

3.2.6 Freescale PowerQUICC III ..... 48

3.2.7 Freescale QorIQ™ . .... 48

3.2.8 AMCC Power Architecture ..... 50

3.2.9 MIPS ..... 53

3.2.10 Broadcom MIPS. .... 54

3.2.11 Other MIPS..... 55

3.2.12 ARM ..... 55

3.2.13 TI ARM. .... 56

3.2.14 Freescale ARM. .... 58

3.2.15 Other ARM Processors . .... 59

3.3 Other Architectures..... 59

3.4 Hardware Platforms . .... 60

3.4.1 CompactPCI . .... 60

3.4.2 ATCA . .... 60

3.5 Summary. .... 61

3.5.1 Suggestions for Additional Reading. .... 62

<b>Chapter 4 The Linux Kernel: A Different Perspective.....</b>	<b>63</b>
4.1 Background. ....	64
4.1.1 Kernel Versions .....	65
4.1.2 Kernel Source Repositories.....	67
4.1.3 Using <code>git</code> to Download a Kernel .....	68
4.2 Linux Kernel Construction. ....	68
4.2.1 Top-Level Source Directory. ....	69
4.2.2 Compiling the Kernel . ....	69
4.2.3 The Kernel Proper: <code>vmlinux</code> . ....	72
4.2.4 Kernel Image Components .....	73
4.2.5 Subdirectory Layout.....	77
4.3 Kernel Build System. ....	78
4.3.1 The Dot-Config.....	78
4.3.2 Configuration Editor(s) .....	80
4.3.3 Makefile Targets.....	83
4.4 Kernel Configuration . ....	89
4.4.1 Custom Configuration Options . ....	91
4.4.2 Kernel Makefiles .....	95
4.5 Kernel Documentation.....	96
4.6 Obtaining a Custom Linux Kernel . ....	96
4.6.1 What Else Do I Need? .....	97
4.7 Summary. ....	97
4.7.1 Suggestions for Additional Reading.....	98
<b>Chapter 5 Kernel Initialization.....</b>	<b>99</b>
5.1 Composite Kernel Image: Piggy and Friends . ....	100
5.1.1 The Image Object.....	103
5.1.2 Architecture Objects .....	104

---

5.1.3	Bootstrap Loader .....	105
5.1.4	Boot Messages.....	106
5.2	Initialization Flow of Control. ....	109
5.2.1	Kernel Entry Point: <code>head.o</code> . ....	111
5.2.2	Kernel Startup: <code>main.c</code> .....	113
5.2.3	Architecture Setup .....	114
5.3	Kernel Command-Line Processing . ....	115
5.3.1	The <code>__setup</code> Macro . ....	116
5.4	Subsystem Initialization. ....	122
5.4.1	The <code>*__initcall</code> Macros . ....	122
5.5	The <code>init</code> Thread. ....	125
5.5.1	Initialization Via <code>initcalls</code> . ....	126
5.5.2	<code>initcall_debug</code> .....	127
5.5.3	Final Boot Steps.....	127
5.6	Summary. ....	129
5.6.1	Suggestions for Additional Reading.....	130
<b>Chapter 6</b>	<b>User Space Initialization .....</b>	<b>131</b>
6.1	Root File System . ....	132
6.1.1	FHS: File System Hierarchy Standard .....	133
6.1.2	File System Layout. ....	133
6.1.3	Minimal File System .....	134
6.1.4	The Embedded Root FS Challenge .....	136
6.1.5	Trial-and-Error Method .....	137
6.1.6	Automated File System Build Tools.....	137

6.2 Kernel's Last Boot Steps. .... 137

    6.2.1 First User Space Program ..... 139

    6.2.2 Resolving Dependencies..... 139

    6.2.3 Customized Initial Process. .... 140

6.3 The `init` Process. .... 140

    6.3.1 `inittab`..... 143

    6.3.2 Sample Web Server Startup Script. .... 145

6.4 Initial RAM Disk. .... 146

    6.4.1 Booting with `initrd`..... 147

    6.4.2 Bootloader Support for `initrd`..... 148

    6.4.3 `initrd` Magic: `linuxrc` ..... 150

    6.4.4 The `initrd` Plumbing..... 151

    6.4.5 Building an `initrd` Image..... 152

6.5 Using `initramfs`. .... 153

    6.5.1 Customizing `initramfs`. .... 154

6.6 Shutdown..... 156

6.7 Summary. .... 156

    6.7.1 Suggestions for Additional Reading. .... 157

**Chapter 7 Bootloaders.....159**

7.1 Role of a Bootloader..... 160

7.2 Bootloader Challenges. .... 161

    7.2.1 DRAM Controller ..... 161

    7.2.2 Flash Versus RAM..... 162

    7.2.3 Image Complexity. .... 162

    7.2.4 Execution Context ..... 165

---

7.3	A Universal Bootloader: Das U-Boot .....	166
7.3.1	Obtaining U-Boot .....	166
7.3.2	Configuring U-Boot .....	167
7.3.3	U-Boot Monitor Commands .....	169
7.3.4	Network Operations .....	170
7.3.5	Storage Subsystems .....	173
7.3.6	Booting from Disk.....	174
7.4	Porting U-Boot . .....	174
7.4.1	EP405 U-Boot Port .....	175
7.4.2	U-Boot Makefile Configuration Target .....	176
7.4.3	EP405 First Build .....	177
7.4.4	EP405 Processor Initialization.....	178
7.4.5	Board-Specific Initialization. ....	181
7.4.6	Porting Summary. ....	184
7.4.7	U-Boot Image Format.....	185
7.5	Device Tree Blob (Flat Device Tree). ....	187
7.5.1	Device Tree Source.....	189
7.5.2	Device Tree Compiler .....	192
7.5.3	Alternative Kernel Images Using DTB .....	193
7.6	Other Bootloaders. ....	194
7.6.1	Lilo .....	194
7.6.2	GRUB .....	195
7.6.3	Still More Bootloaders . ....	197
7.7	Summary. ....	197
7.7.1	Suggestions for Additional Reading. ....	198

**Chapter 8 Device Driver Basics .....201**

- 8.1 Device Driver Concepts . ..... 202
  - 8.1.1 Loadable Modules..... 203
  - 8.1.2 Device Driver Architecture ..... 204
  - 8.1.3 Minimal Device Driver Example..... 204
  - 8.1.4 Module Build Infrastructure . ..... 205
  - 8.1.5 Installing a Device Driver ..... 209
  - 8.1.6 Loading a Module..... 210
  - 8.1.7 Module Parameters ..... 211
- 8.2 Module Utilities . ..... 212
  - 8.2.1 insmod..... 212
  - 8.2.2 lsmod ..... 213
  - 8.2.3 modprobe. .... 213
  - 8.2.4 depmod..... 214
  - 8.2.5 rmmod ..... 215
  - 8.2.6 modinfo..... 216
- 8.3 Driver Methods. .... 217
  - 8.3.1 Driver File System Operations . ..... 217
  - 8.3.2 Allocation of Device Numbers ..... 220
  - 8.3.3 Device Nodes and mknod ..... 220
- 8.4 Bringing It All Together. .... 222
- 8.5 Building Out-of-Tree Drivers ..... 223
- 8.6 Device Drivers and the GPL. .... 224
- 8.7 Summary..... 225
  - 8.7.1 Suggestions for Additional Reading. .... 226

---

<b>Chapter 9 File Systems</b> .....	<b>227</b>
9.1 Linux File System Concepts . . . . .	228
9.1.1 Partitions . . . . .	229
9.2 ext2 . . . . .	230
9.2.1 Mounting a File System . . . . .	232
9.2.2 Checking File System Integrity . . . . .	233
9.3 ext3 . . . . .	235
9.4 ext4 . . . . .	237
9.5 ReiserFS . . . . .	238
9.6 JFFS2 . . . . .	239
9.6.1 Building a JFFS2 Image . . . . .	240
9.7 <i>cramfs</i> . . . . .	242
9.8 Network File System . . . . .	244
9.8.1 Root File System on NFS . . . . .	246
9.9 Pseudo File Systems . . . . .	248
9.9.1 <i>/proc</i> File System . . . . .	249
9.9.2 <i>sysfs</i> . . . . .	252
9.10 Other File Systems . . . . .	255
9.11 Building a Simple File System . . . . .	256
9.12 Summary . . . . .	258
9.12.1 Suggestions for Additional Reading . . . . .	259
<b>Chapter 10 MTD Subsystem</b> . . . . .	<b>261</b>
10.1 MTD Overview . . . . .	262
10.1.1 Enabling MTD Services . . . . .	263
10.1.2 MTD Basics . . . . .	265
10.1.3 Configuring MTD on Your Target . . . . .	267

10.2	MTD Partitions . . . . .	267
10.2.1	Redboot Partition Table Partitioning . . . . .	269
10.2.2	Kernel Command-Line Partitioning . . . . .	273
10.2.3	Mapping Driver . . . . .	274
10.2.4	Flash Chip Drivers . . . . .	276
10.2.5	Board-Specific Initialization . . . . .	276
10.3	MTD Utilities. . . . .	279
10.3.1	JFFS2 Root File System . . . . .	281
10.4	UBI File System . . . . .	284
10.4.1	Configuring for UBIFS . . . . .	284
10.4.2	Building a UBIFS Image . . . . .	284
10.4.3	Using UBIFS as the Root File System . . . . .	287
10.5	Summary. . . . .	287
10.5.1	Suggestions for Additional Reading . . . . .	288
<b>Chapter 11</b>	<b>BusyBox . . . . .</b>	<b>289</b>
11.1	Introduction to BusyBox . . . . .	290
11.1.1	BusyBox Is Easy . . . . .	291
11.2	BusyBox Configuration . . . . .	291
11.2.1	Cross-Compiling BusyBox . . . . .	293
11.3	BusyBox Operation . . . . .	293
11.3.1	BusyBox <code>init</code> . . . . .	297
11.3.2	Sample <code>rcS</code> Initialization Script . . . . .	299
11.3.3	BusyBox Target Installation. . . . .	300
11.3.4	BusyBox Applets. . . . .	302
11.4	Summary. . . . .	303
11.4.1	Suggestions for Additional Reading. . . . .	304



---

<b>Chapter 12 Embedded Development Environment .....</b>	<b>305</b>
12.1 Cross-Development Environment .....	306
12.1.1 “Hello World” Embedded .....	307
12.2 Host System Requirements .....	311
12.2.1 Hardware Debug Probe .....	311
12.3 Hosting Target Boards .....	312
12.3.1 TFTP Server .....	312
12.3.2 BOOTP/DHCP Server .....	313
12.3.3 NFS Server .....	316
12.3.4 Target NFS Root Mount .....	318
12.3.5 U-Boot NFS Root Mount Example .....	320
12.4 Summary .....	322
12.4.1 Suggestions for Additional Reading .....	323
<b>Chapter 13 Development Tools .....</b>	<b>325</b>
13.1 GNU Debugger (GDB) .....	326
13.1.1 Debugging a Core Dump .....	327
13.1.2 Invoking GDB.....	329
13.1.3 Debug Session in GDB.....	331
13.2 Data Display Debugger.....	333
13.3 cbrowser/cscope.....	335
13.4 Tracing and Profiling Tools .....	337
13.4.1 strace.....	337
13.4.2 strace Variations.....	341
13.4.3 ltrace.....	343
13.4.4 ps.....	344
13.4.5 top.....	346

13.4.6	mtrace.	348
13.4.7	dmalloc.	350
13.4.8	Kernel Oops .	353
13.5	Binary Utilities .	355
13.5.1	readelf.	355
13.5.2	Examining Debug Information Using readelf.	357
13.5.3	objdump.	359
13.5.4	objcopy.	360
13.6	Miscellaneous Binary Utilities .	361
13.6.1	strip	361
13.6.2	addr2line.	361
13.6.3	strings.	362
13.6.4	ldd.	362
13.6.5	nm.	363
13.6.6	prelink.	364
13.7	Summary.	364
13.7.1	Suggestions for Additional Reading.	365
<b>Chapter 14</b>	<b>Kernel Debugging Techniques</b>	<b>367</b>
14.1	Challenges to Kernel Debugging	368
14.2	Using KGDB for Kernel Debugging .	369
14.2.1	KGDB Kernel Configuration.	371
14.2.2	Target Boot with KGDB Support.	372
14.2.3	Useful Kernel Breakpoints.....	376
14.2.4	Sharing a Console Serial Port with KGDB	377
14.2.5	Debugging Very Early Kernel Code	379
14.2.6	KGDB Support in the Mainline Kernel	380

---

14.3	Kernel Debugging Techniques. ....	381
14.3.1	gdb Remote Serial Protocol.....	382
14.3.2	Debugging Optimized Kernel Code.....	385
14.3.3	GDB User-Defined Commands.....	392
14.3.4	Useful Kernel GDB Macros .....	393
14.3.5	Debugging Loadable Modules. ....	402
14.3.6	printk Debugging.....	407
14.3.7	Magic SysReq Key .....	409
14.4	Hardware-Assisted Debugging. ....	410
14.4.1	Programming Flash Using a JTAG Probe.....	411
14.4.2	Debugging with a JTAG Probe .....	413
14.5	When It Doesn't Boot . ....	417
14.5.1	Early Serial Debug Output .....	417
14.5.2	Dumping the printk Log Buffer .....	417
14.5.3	KGDB on Panic. ....	420
14.6	Summary. ....	421
14.6.1	Suggestions for Additional Reading.....	422
<b>Chapter 15</b>	<b>Debugging Embedded Linux Applications .....</b>	<b>423</b>
15.1	Target Debugging.....	424
15.2	Remote (Cross) Debugging. ....	424
15.2.1	gdbserver.....	427
15.3	Debugging with Shared Libraries . ....	429
15.3.1	Shared Library Events in GDB.....	431
15.4	Debugging Multiple Tasks. ....	435
15.4.1	Debugging Multiple Processes.....	435
15.4.2	Debugging Multithreaded Applications .....	438
15.4.3	Debugging Bootloader/Flash Code .....	441

15.5	Additional Remote Debug Options. ....	442
15.5.1	Debugging Using a Serial Port . ....	442
15.5.2	Attaching to a Running Process . ....	442
15.6	Summary. ....	443
15.6.1	Suggestions for Additional Reading. ....	444
<b>Chapter 16</b>	<b>Open Source Build Systems .....</b>	<b>445</b>
16.1	Why Use a Build System? .....	446
16.2	Scratchbox. ....	447
16.2.1	Installing Scratchbox. ....	447
16.2.2	Creating a Cross-Compilation Target. ....	448
16.3	Buildroot. ....	451
16.3.1	Buildroot Installation. ....	451
16.3.2	Buildroot Configuration .....	451
16.3.3	Buildroot Build.....	452
16.4	OpenEmbedded. ....	454
16.4.1	OpenEmbedded Composition .....	455
16.4.2	BitBake Metadata.....	456
16.4.3	Recipe Basics.....	456
16.4.4	Metadata Tasks.....	460
16.4.5	Metadata Classes.....	461
16.4.6	Configuring OpenEmbedded .....	462
16.4.7	Building Images .....	463
16.5	Summary. ....	464
16.5.1	Suggestions for Additional Reading. ....	464

---

<b>Chapter 17 Linux and Real Time.....</b>	<b>465</b>
17.1 What Is Real Time? . . . . .	466
17.1.1 Soft Real Time . . . . .	466
17.1.2 Hard Real Time . . . . .	467
17.1.3 Linux Scheduling.....	467
17.1.4 Latency . . . . .	467
17.2 Kernel Preemption . . . . .	469
17.2.1 Impediments to Preemption.....	469
17.2.2 Preemption Models.....	471
17.2.3 SMP Kernel . . . . .	472
17.2.4 Sources of Preemption Latency . . . . .	473
17.3 Real-Time Kernel Patch. . . . .	473
17.3.1 Real-Time Features . . . . .	475
17.3.2 O(1) Scheduler . . . . .	476
17.3.3 Creating a Real-Time Process.....	477
17.4 Real-Time Kernel Performance Analysis. . . . .	478
17.4.1 Using Ftrace for Tracing. . . . .	478
17.4.2 Preemption Off Latency Measurement.....	479
17.4.3 Wakeup Latency Measurement . . . . .	481
17.4.4 Interrupt Off Timing . . . . .	483
17.4.5 Soft Lockup Detection.....	484
17.5 Summary. . . . .	485
17.5.1 Suggestion for Additional Reading. . . . .	485
<b>Chapter 18 Universal Serial Bus .....</b>	<b>487</b>
18.1 USB Overview . . . . .	488
18.1.1 USB Physical Topology . . . . .	488
18.1.2 USB Logical Topology . . . . .	490

- 18.1.3 USB Revisions ..... 491
- 18.1.4 USB Connectors..... 492
- 18.1.5 USB Cable Assemblies ..... 494
- 18.1.6 USB Modes ..... 494
- 18.2 Configuring USB. .... 495
  - 18.2.1 USB Initialization ..... 497
- 18.3 sysfs and USB Device Naming ..... 500
- 18.4 Useful USB Tools. .... 502
  - 18.4.1 USB File System ..... 502
  - 18.4.2 Using `usbview`..... 504
  - 18.4.3 USB Utils (`lsusb`)..... 507
- 18.5 Common USB Subsystems. .... 508
  - 18.5.1 USB Mass Storage Class..... 508
  - 18.5.2 USB HID Class ..... 511
  - 18.5.3 USB CDC Class Drivers..... 512
  - 18.5.4 USB Network Support. .... 515
- 18.6 USB Debug. .... 516
  - 18.6.1 `usbmon` ..... 517
  - 18.6.2 Useful USB Miscellanea. .... 518
- 18.7 Summary. .... 519
  - 18.7.1 Suggestions for Additional Reading. .... 519
- Chapter 19 udev .....521**
  - 19.1 What Is udev?..... 522
  - 19.2 Device Discovery..... 523
  - 19.3 Default udev Behavior..... 525

19.4 Understanding udev Rules. .... 527

    19.4.1 Modalias. .... 530

    19.4.2 Typical udev Rules Configuration. .... 533

    19.4.3 Initial System Setup for udev. .... 535

19.5 Loading Platform Device Drivers ..... 538

19.6 Customizing udev Behavior. .... 540

    19.6.1 udev Customization Example: USB Automounting. .... 540

19.7 Persistent Device Naming. .... 541

    19.7.1 udev Helper Utilities. .... 542

19.8 Using udev with busybox. .... 545

    19.8.1 busybox mdev. .... 545

    19.8.2 Configuring mdev. .... 547

19.9 Summary. .... 548

    19.9.1 Suggestions for Additional Reading. .... 548

**Appendix A GNU Public License.....549**

Preamble ..... 550

Terms and Conditions for Copying, Distribution, and Modification. .... 551

No Warranty. .... 555

**Appendix B U-Boot Configurable Commands .....557**

**Appendix C BusyBox Commands.....561**

**Appendix D SDRAM Interface Considerations ..... 571**

D.1 SDRAM Basics. .... 572

    D.1.1 SDRAM Refresh. .... 573

D.2 Clocking ..... 574

D.3 SDRAM Setup..... 575

D.4 Summary . ..... 580

    D.4.1 Suggestions for Additional Reading. .... 580

**Appendix E Open Source Resources .....581**

Source Repositories and Developer Information..... 582

Mailing Lists ..... 582

Linux News and Developments..... 583

Open Source Legal Insight and Discussion..... 583

**Appendix F Sample BDI-2000 Configuration File .....585**

**Index.....593**



- [\*\*The Walking Dead, Volume 6: This Sorrowful Life pdf, azw \(kindle\), epub\*\*](#)
- [download The Invention of Solitude book](#)
- [click Trends and Issues in African Philosophy](#)
- [click Blood River \(Amtrak Wars, Book 4\)](#)
  
- <http://thewun.org/?library/The-Walking-Dead--Volume-6--This-Sorrowful-Life.pdf>
- <http://creativebeard.ru/freebooks/The-Invention-of-Solitude.pdf>
- <http://tuscalaural.com/library/Mia-in-the-Mix--Cupcake-Diaries--Book-2-.pdf>
- <http://studystrategically.com/freebooks/Blood-River--Amtrak-Wars--Book-4-.pdf>