

HACKING RASPBERRY PI

TIMOTHY L. WARNER

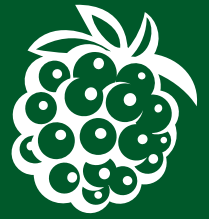


que

FREE SAMPLE CHAPTER



SHARE WITH OTHERS



HACKING RASPBERRY PI[®]

T I M O T H Y L . W A R N E R

800 East 96th Street,
Indianapolis, Indiana 46240 USA

Hacking Raspberry Pi®

Copyright © 2014 by Que Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-5156-0

ISBN-10: 0-7897-5156-9

Library of Congress Control Number: 2013944701

Printed in the United States of America

First Printing: November 2013

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales
international@pearsoned.com

Editor-in-Chief

Greg Wiegand

Executive Editor

Rick Kughen

Development Editor

Todd Brakke

Managing Editor

Kristy Hart

Project Editor

Elaine Wiley

Copy Editor

Chrissy White

Indexer

Brad Herriman

Proofreader

Kathy Ruiz

Technical Editor

Brian McLaughlin

Editorial Assistant

Kristen Watterson

Cover Designer

Chuti Prasertsith

Book Designer

Mark Shirar

Senior Compositor

Gloria Schurick

Graphics

Tammy Graham

Cover Illustration

©Kokander/Shutter Stock

©Yippe/Shutter Stock

Contents at a Glance

Introduction 1

PART I HARDWARE FOUNDATIONS

CHAPTER 1 What Is the Raspberry Pi? 7

CHAPTER 2 Hardware Components Quick Start 17

CHAPTER 3 A Tour of Raspberry Pi Peripheral Devices 33

PART II SOFTWARE FOUNDATIONS

CHAPTER 4 Installing and Configuring an Operating System 47

CHAPTER 5 Debian Linux Fundamentals—Terminal 67

CHAPTER 6 Debian Linux Fundamentals—Graphical User Interface 87

CHAPTER 7 Networking Raspberry Pi 103

CHAPTER 8 Programming Raspberry Pi with Scratch—Beginnings 121

CHAPTER 9 Programming Raspberry Pi with Scratch—Next Steps 137

CHAPTER 10 Programming Raspberry Pi with Python—Beginnings 153

CHAPTER 11 Programming Raspberry Pi with Python—Next Steps 165

PART III BASIC RASPBERRY PI PROJECTS

CHAPTER 12 Raspberry Pi Media Center 179

CHAPTER 13 Raspberry Pi Retro Game Station 199

CHAPTER 14 Raspberry Pi *Minecraft* Server 217

CHAPTER 15 Raspberry Pi Web Server 233

CHAPTER 16 Raspberry Pi Portable Webcam 253

CHAPTER 17 Raspberry Pi Security and Privacy Device 277

PART IV HACKING THE RASPBERRY PI

CHAPTER 18 Raspberry Pi Overclocking 299

CHAPTER 19 Raspberry Pi and Arduino 315

CHAPTER 20 Raspberry Pi and the Gertboard 337

Index 359

Table of Contents

Introduction	1
What's in This Book	2
Who Can Use This Book	4
How to Use This Book	5
<i>Task: This Is a Task</i>	5
There's More Online...	5

PART I HARDWARE FOUNDATIONS

Chapter 1 What Is the Raspberry Pi?	7
Why the Pi?	8
Hardware Versions	9
Is the Raspberry Pi Open Source?	14
Open Source Licenses	15
Open Source and the Raspberry Pi	15
How Can I Purchase a Raspberry Pi?	16
Chapter 2 Hardware Components Quick Start	17
Understanding Pi Hardware Terminology	17
The Hidden Cost of Owning a Raspberry Pi	20
A 5V Power Supply	21
SD Card	23
Powered USB Hub	24
Ethernet Cable	24
Monitor	25
Cables	25
USB Keyboard and Mouse	26
A Tour of the Model B Board	27
Networking	27
Video and Audio	27
Storage	29
Power/Status Information	29
Camera	30

Processing	31
Expansion	31
Next Steps	32

Chapter 3 A Tour of Raspberry Pi Peripheral Devices 33

Circuit Prototyping Equipment	34
Single-Board Microcontrollers	37
The Gertboard	38
Single-Board Computers	40
Relevant Technician Tools	41
<i>Task: Check Raspberry Pi Voltage with a Multimeter</i>	41
Raspberry Pi Starter Kits	43
Next Steps	45

PART II SOFTWARE FOUNDATIONS

Chapter 4 Installing and Configuring an Operating System 47

Common Operating Systems	49
Understanding Linux	49
Linux and Raspberry Pi	52
The Kernel and Firmware	53
Raspberry Pi's Other Operating Systems	54
Installing Raspbian on Your Raspberry Pi	55
<i>Task: Creating a Raspbian SD Card in Windows</i>	57
<i>Task: Creating a Raspbian SD Card in OS X</i>	58
<i>Task: Creating a Raspbian SD Card in Linux</i>	60
An All-in-One Solution	63
<i>Task: Installing an OS on Your Raspberry Pi Using NOOBS</i>	64
Testing Your New Raspbian Image	65

Chapter 5 Debian Linux Fundamentals—Terminal 67

Baby Steps	67
Essential Terminal Commands	69
startx	69
pwd	70
ls	71
cd	72

- sudo 73
- passwd 74
- nano 74
- man 75
- shutdown 76

Updating Your Software 77

- Updating Raspbian 78
- Updating the Pi Firmware 79

Revisiting Raspi-Config 80

- Expand Filesystem 81
- Change User Password 81
- Enable Boot to Desktop 82
- Internationalisation Options 82
- Enable Camera 82
- Add to Rastrack 83
- Overclocking 83
- Advanced Options 84
- Raspi-Config Under the Hood 84

Next Steps 85

Chapter 6 Debian Linux Fundamentals—Graphical User Interface 87

LXDE—The Desktop Environment 87

- Starting and Exiting LXDE 88
 - Task: Using Raspi-Config to Adjust Boot Behavior 88*

- Touring the LXDE Interface 89

- Delving into the Main Menu 92

Installing, Updating, and Removing Software 94

Accessing the Pi Store 95

- Task: Downloading a Free App from the Pi Store 97*

Tweaking the LXDE UI 98

- Task: Change the Desktop Background 98*

- Task: Customize the LXTerminal 99*

- Task: Customize the LXPanel 100*

- Openbox 100

Editing Configuration Files 101

Chapter 7 Networking Raspberry Pi 103

Basic Networking Concepts 103

Configuring Wired Ethernet 104

Task: Setting a Static IP Address on Your Raspberry Pi 106

Configuring Wireless Ethernet 107

Task: Setting Up Wi-Fi on Your Pi 109

Configuring “Headless” Raspberry Pi 110

Secure Shell (SSH) 112

Task: Using SSH to Connect to a Raspberry Pi Remotely 113

Virtual Network Computing (VNC) 114

*Task: Using VNC to Connect to the Raspberry Pi 115**Task: Configure Your Pi to Start VNC Server Automatically 117*

On Browsing the Web with the Pi 118

Chapter 8 Programming Raspberry Pi with Scratch—Beginnings 121

Technical Aspects of Scratch 122

Scratch Version Issues 122

The Scratch Community 123

Getting Comfortable with the Scratch Interface 124

About Blocks 127

Crafting a (Very) Simple Scratch Application 129

Task: Creating a Basic Scratch App 130

The PicoBoard 132

Obtaining a PicoBoard? 134

Using a PicoBoard in Scratch 134

*Task: Using the PicoBoard in Scratch 134***Chapter 9 Programming Raspberry Pi with Scratch—Next Steps 137**

Begin with the End in Mind 137

Let’s Build the Game! 139

*Task: Setting Up the Game Screens 140**Task: Setting Up the Scratch Cat 143*

Setting Up the First Ball 144

Setting Up the Second and Third Balls 146

Debugging and Troubleshooting 147

- Uploading Your Project 149
 - Task: Share Your Scratch Project* 149
- Remixing 151
 - Task: Create a Remix* 151

Chapter 10 Programming Raspberry Pi with Python—Beginnings 153

- Checking Out the Python Environment 154
- Interacting with the Python 3 Interpreter 156
 - Exiting the Interpreter 157
 - Getting Online Help 157
- Writing Your First Python Program 158
 - Task: Creating Your First Python Script* 159
- Running Python Scripts 161
 - Task: Running Python Scripts (command line)* 161
 - Task: Running Python Scripts (IDLE)* 161
- Broadening Your Python Horizons 163

Chapter 11 Programming Raspberry Pi with Python—Next Steps 165

- Getting Comfortable with IDLE 165
 - Task: Creating a New Python Script File* 168
- Writing a Simple Game 168
- Delving into a Bit More Detail 171
 - Variables 171
 - Type Casting 171
 - Concatenation 173
 - Modules 173
 - Task: Loading and then Updating the GPIO Module in Python 3* 176
- Where Do You Go from Here? 177

PART III BASIC RASPBERRY PI PROJECTS

Chapter 12 Raspberry Pi Media Center 179

- A Bit o' History 180
- But Will It Blend? 181

Introducing Raspbmc	182
<i>Task: Installing Raspbmc Under Windows</i>	182
<i>Task: Installing Raspbmc Under OS X</i>	183
Configuring Raspbmc	184
Getting Your Remote Control Working	186
<i>Task: Controlling Your Raspbmc Box from iOS</i>	186
<i>Task: Controlling Your Raspbmc Box from a Web Browser</i>	189
<i>Task: Configuring a GPIO-based IR Receiver</i>	190
Transferring Content to Your Pi	192
<i>Task: Uploading Media Content to Raspberry Pi</i>	192
Scraping Your Media	194
<i>Task: Configuring XBMC Media Detection</i>	194
Playing Your Content	196
Installing Add-Ons	197

Chapter 13 Raspberry Pi Retro Game Station 199

A Word About Console Video Game Emulation	200
Installing RetroPie	200
<i>Task: Setting Up RetroPie</i>	201
Transferring ROMs to RetroPie	203
<i>Task: Transferring ROMs to Your Raspberry Pi</i>	204
Setting Up Your Controls	206
<i>Task: Configure RetroPie Joystick Control Mappings</i>	208
Playing Your Games	209
Installing Useful Add-Ons	210
<i>Task: Install ES-Scraper</i>	211
In Search of the Perfect Joystick	212

Chapter 14 Raspberry Pi Minecraft Server 217

Installing Minecraft Pi	219
<i>Task: Installing Minecraft Pi</i>	219
Accessing the Python API	221
<i>Task: Loading the Minecraft Python Libraries</i>	222
<i>Task: Making Your Player Jump High!</i>	224
<i>Task: Create a Diamond Floor</i>	225

- Building a *Minecraft* Server 226
 - Task: Installing Java and the Minecraft Server* 227
- Administering a *Minecraft* Server 230
 - Minecraft Plugins* 232

Chapter 15 Raspberry Pi Web Server 233

- What Is the LAMP Stack? 234
- Installing Your Web Server 235
 - Task: Installing the LAMP Stack on Raspberry Pi* 236
 - Task: Verifying PHP Configuration* 237
 - Task: Verifying MySQL Installation* 239
- Tweaking Up Your Web Server Settings 240
- Transferring Content to Your Web Server 240
 - Task: Using SFTP to Transfer Content to Your Pi* 241
 - Task: Install and Test FTP on Your Raspberry Pi* 243
- Setting Up phpMyAdmin 243
 - Task: Installing phpMyAdmin* 244
- Setting Up Joomla 245
 - Task: Installing Joomla on Your Raspberry Pi* 246
- Putting Your Web Server on the Public Internet 249
 - Task: Making Your Raspberry Pi Publicly Accessible by Using No-IP* 250

Chapter 16 Raspberry Pi Portable Webcam 253

- About the Raspberry Pi Camera Board 253
- Installing and Configuring the Raspberry Pi Camera Board 256
 - Task: Preparing the Raspberry Pi for the Camera Board* 256
 - Task: Installing the Raspberry Pi Camera Board* 257
- Using the Camera Board 259
 - Capturing Still Pictures 259
 - Recording Video with the Camera Board 263
 - Task: Encoding a raspivid Video File* 265
- Using a Third-Party USB Webcam 266
 - Task: Configuring a Third-Party USB Webcam* 267
- Setting Up Your Webcam 269
 - Task: Setting Up a Time-Lapse Webcam* 270

Adding a Rechargeable Battery Pack to Your Pi Camera	273
Python and Your Webcam	274
<i>Task: Setting up SimpleCV</i>	274
Chapter 17 Raspberry Pi Security and Privacy Device	277
Encrypting Your Internet Connection with a VPN	277
Browsing Where You Want via a Proxy Server	278
Building Your Raspberry Pi VPN Gateway	280
<i>Task: Creating Your Hamachi VPN</i>	281
<i>Task: Connecting to Your Hamachi VPN from Your Desktop Computer</i>	283
<i>Task: Installing Hamachi Client on Your Raspberry Pi</i>	284
Building Your Raspberry Pi Proxy Server	286
<i>Task: Installing and Configuring Privoxy Raspberry Pi</i>	286
Testing the Configuration	287
<i>Task: Pointing Your Workstation Computer at Your Pi Proxy</i>	289
Speaking of Geolocation...	292
<i>Task: Adding Your Raspberry Pi to the Rastrack Database</i>	294
Building a Raspberry Pi Tor Proxy	294
PART IV HACKING THE RASPBERRY PI	
Chapter 18 Raspberry Pi Overclocking	299
What Is Overclocking?	299
Warranty Implications of Overclocking	300
Overclocking Your Pi	302
The Raspi-Config Method	302
<i>Task: Verifying Your Pi's CPU, Voltage, and Temperature Status</i>	303
The Manual Method	304
Benchmarking Your Pi	305
<i>Task: Installing and Running nbench</i>	306
Adjusting the Memory Split on Your Pi	307
<i>Task: Tweaking Memory Split with Raspi-Config</i>	308
Tweaking Memory Split by Hand	309
A Historical Footnote	309
Various and Sundry Performance Tweaks	311

Chapter 19 Raspberry Pi and Arduino 315

Introducing the Arduino 315

Digging into the Arduino Uno 319

Connecting the Arduino and the Raspberry Pi 321

Connecting the Raspberry Pi GPIO Pins to the Arduino Serial Pins 322

Connecting the Raspberry Pi GPIO Pins to the Arduino I2C 322

Connecting the Raspberry Pi to the Arduino via USB 322

Connecting the Raspberry Pi to the Arduino via a Shield or Bridge Board 322

Connecting the Raspberry Pi to an Arduino Clone 323

Understanding the Arduino Development Workflow 323

Task: Install and Configure Arduino IDE on the Raspberry Pi 324 *Task: Running the “Blink” Sample Sketch* 327

Fading an LED 330

Task: Running the Fade Sketch 331

Using the AlaMode 332

Task: Getting the AlaMode Up and Running 333**Chapter 20 Raspberry Pi and the Gertboard 337**

Anatomy of the GPIO 339

Connecting Your Gertboard and Raspberry Pi 341

Installing the Testing Software 343

Task: Enabling SPI on Your Raspberry Pi 344 *Task: Installing the Python Gertboard Test Suite* 345

Testing the LEDs 346

Testing Input/Output 349

A Couple Quick Breadboarding Exercises 350

Accessing the GPIO Pins Directly 350

Task: Lighting an LED Directly from the GPIO 351

Accessing the GPIO Pins via the Pi Cobbler 353

Programming the Atmel Microcontroller 354

Task: Preparing Your Arduino Environment 355

Final Thoughts, and Thank You 357

Index 359

About the Author

Timothy L. Warner has helped thousands of beginners succeed with technology in business and in schools. Until recently a digital media specialist with Pearson Technology Group, he worked in various facets of information technology, including systems administration, software architecture, and technical training. He is the author of *Unauthorized Guide to iPhone, iPad and iPhone Repair*.

Dedication

To my father, Lawrence K. Warner, whose fascination with electronic gadgets fueled and inspired my own.

Acknowledgments

Thank you to all my friends at Pearson Technology Group, especially my editors Rick Kughen and Todd Brakke, my publishers Greg Wiegand and Paul Boger, my project managers Elaine Wiley and Kristen Watterson, and my copy editors/proof readers Chrissy White and Kathy Ruiz. You are a group of wonderful people, that's for sure.

With regard to the book's subject matter, thank you to Eben Upton and Gert van Loo of the Raspberry Pi Foundation (Gert, I know you say you aren't a member of the Foundation, but you're foundational to the Pi project at any rate) for your gracious assistance—you guys are brilliant! Thanks to Brian McLaughlin for his thorough technical edit of the manuscript.

Thank you to my family (Susan, Zoey, Fred, Moon, Maya, Stub, Mom, Dad, Trish, Mom H, Rick, Chelle, Don, Alex, Kevin, Kristina, and so on) for your love and support—I treasure you all.

Finally, thank you to you, my reader. Without you I would be teaching and writing into a blank void. Don't get me wrong—I love the sound of my own voice and I'm my own favorite subject. However, discussing this technology and sharing this knowledge just wouldn't be the same without you.

We Want to Hear from You!

As the reader of this book, you are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@quepublishing.com

Mail: Que Publishing
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at quepublishing.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

INTRODUCTION



Hi! I'm Tim Warner, and I am happy to be your guide to the Raspberry Pi. My goal for this Introduction is to get you excited about this \$35 credit card-sized computer that we call the Raspberry Pi.

“What in the world can you do with a \$35 computer?” you might ask. Well let's have a look at a representative smattering of some popular Raspberry Pi projects, and you can answer that question for yourself:

- Picade Arcade Cabinet (<http://is.gd/i4YwQ2>). This is a tiny, fully-functional arcade machine.
- Pi in the Sky (<http://is.gd/4niEMV>). The inventor put a camera-equipped Raspberry Pi into the upper atmosphere via a weather balloon.
- Streaming Music Jukebox (<http://is.gd/oqec3H>). This is an inexpensive wireless music broadcasting machine.
- Raspberry Pi Keyboard Computer (<http://is.gd/tvmgC8>). This is a full computer packed into the form factor of a computer keyboard.
- DeviantArt Picture Frame (<http://is.gd/i7ED9w>). This is an interactive frame that dynamically displays artwork from the DeviantArt website.
- Pi-Powered Bitcoin Miner (<http://is.gd/DrpJ7A>). This tiny rig enables you to participate in Bitcoin mining, which can yield you some hefty monetary returns.
- FM Radio Transmitter (<http://is.gd/tS52Yb>). This is a low-power (albeit mono) portable FM radio.
- KindleBerry Pi (<http://is.gd/73iVz4>). This is a Pi that uses an Amazon Kindle as a monitor and a “dumb” terminal.
- Pi-Powered Motion Detector (<http://is.gd/g4Okb6>). This is a Pi that can detect motion and optionally take action upon that detection.
- 3D Printer (<http://is.gd/Bg83jD>). This is a Raspberry Pi-powered printer that can actually fabricate three-dimensional objects.
- Cheese-Powered Radio-Controlled Car (<http://is.gd/ZExbWO>). This radio-controlled car is controlled with a Nintendo Wii Remote and powered by ordinary slices of cheese.
- Raspberry Pi Robot (<http://is.gd/367FZ5>). This is a multi-tasking robot that uses the Raspberry Pi as its “brain.”

- Automated Chicken Coop Door (<http://is.gd/boZR6F>). This is a Raspberry Pi, a relay, and a motor that opens and closes a farm chicken coop door on a schedule.
- Pi-Powered Weather Station (<http://is.gd/LDbZlr>). This outdoor weather sensing and reporting unit runs on USB power.
- Audio Book Player (<http://is.gd/TnkcoW>). The inventor made this simple, push button-operated audio book player for his grandmother's 90th birthday.
- Android Device (<http://is.gd/9eLPkV>). The true mark of the software/hardware hacker is to get Linux or Android running on any device. This is a Raspberry Pi that runs Android 4.0 Ice Cream Sandwich.

I think you'll agree that the diminutive Raspberry Pi has quite a bit of potential for a tiny little circuit board. Let's take a look at how I've organized this book to provide you with maximum learning in minimum time.

The title of this book contains the inflammatory term *hacking*. This word has several meanings even within the information technology industry. First, hacking means to use or adapt a piece of electronics for a purpose other than which it was originally intended. For instance, you can hack your eReader such that it runs Linux and acts as a web server. Would you want to? Believe it or not, people do exactly that, often just to see if they can.

Second, hacking means to break into someone else's computer system, often illegally and without the owner's permission. Of course, *Hacking Raspberry Pi* embraces the first definition of the term!

What's in This Book

Hacking Raspberry Pi is organized in such a way that I gradually immerse you into software and hardware engineering. Believe it or not, configuring hardware and mastering the Linux command line isn't as difficult as you might have imagined. This book takes you through it, beginning with...

- **Chapter 1—What Is the Raspberry Pi?:** Here I explain exactly what the Raspberry Pi is, how it is important, and why you would want to learn more about the device.
- **Chapter 2—Hardware Components Quick Start:** In this chapter you become familiar with the Raspberry Pi's form factor; in other words, its onboard hardware.
- **Chapter 3—A Tour of Raspberry Pi Peripheral Devices:** What do you plug into the Raspberry Pi and where? This chapter gives you everything you need to know to get your Raspberry Pi up and running.
- **Chapter 4—Installing and Configuring an Operating System:** In this chapter you learn how to install Raspbian, the reference Linux distribution for Raspberry Pi.

- **Chapter 5—Debian Linux Fundamentals—Terminal:** This chapter is an excellent jump-start for those who always wanted to understand something of the Linux command-line environment.
- **Chapter 6—Debian Linux Fundamentals—Graphical User Interface:** Sometimes it is plain easier to navigate in Linux from a GUI environment. Here you learn how to use LXDE, the reference GUI shell in Raspbian.
- **Chapter 7—Networking Raspberry Pi:** In most cases, you'll want to put your Raspberry Pi on your local area network (LAN), if not the Internet. You learn how to configure both wired and wireless Ethernet by reading the material in this chapter.
- **Chapter 8—Programming Raspberry Pi with Scratch—Beginnings:** Scratch provides an easy-to-learn platform for learning how to develop computer programs. In this chapter you become familiar with what Scratch is and how the platform works.
- **Chapter 9—Programming Raspberry Pi with Scratch—Next Steps:** Here you develop your first real Scratch application, all on the Raspberry Pi!
- **Chapter 10—Programming Raspberry Pi with Python—Beginnings:** The “Pi” in Raspberry Pi is actually a reference to the Python programming language. Therefore, it is imperative that you learn a thing or two about programming in Python.
- **Chapter 11—Programming Raspberry Pi with Python—Next Steps:** Many Raspberry Pi projects involve one or more Python scripts. Therefore, the more experience you obtain with the language, the better.
- **Chapter 12—Raspberry Pi Media Center:** This section of the book is focused on applying your new Raspberry Pi skills to several practical projects. Here you build a Pi-powered media center running Xbox Media Center (XBMC).
- **Chapter 13—Raspberry Pi Retro Game Station:** Who does not love retro video games? In this chapter you learn how to convert your Raspberry Pi into a mobile classic game station. This is my favorite project in the entire book.
- **Chapter 14—Raspberry Pi *Minecraft* Server:** In this chapter you learn how to install, configure, and play *Minecraft* Pi Edition. You also learn how to set up your Pi as a *Minecraft* server.
- **Chapter 15—Raspberry Pi Web Server:** Here you configure your Raspberry Pi to serve up web pages on your own local network and/or the public Internet.
- **Chapter 16—Raspberry Pi Portable Webcam:** Many people are interested in making their own security camera or general-purpose webcam. It is surprisingly easy to do this with a Raspberry Pi; you can use the Raspberry Pi Camera Board or your own webcam.
- **Chapter 17—Raspberry Pi Security and Privacy Device:** You can use the Raspberry Pi to increase your online security and privacy, say, when you access the Internet through a public Wi-Fi hotspot, hotel room, and so forth. You learn a lot of cool stuff in this chapter!

- **Chapter 18—Raspberry Pi Overclocking:** Although the Raspberry Pi is small and inexpensive, the Raspberry Pi Foundation gives users a great deal of flexibility in squeezing every bit of performance from the device. In this chapter you learn what your options are and how to leverage them to customize the behavior of your Pi.
- **Chapter 19—Raspberry Pi and Arduino:** For my money, the combination of the Raspberry Pi and the Arduino is unbeatable with Pi's flexibility and the Arduino's singleness of purpose. In this chapter you use the Arduino Uno and AlamoDe with your Pi and get some great project ideas.
- **Chapter 20—Raspberry Pi and the Gertboard:** I close the book by teaching you how to use the ultimate Raspberry Pi add-on board, the Gertboard. The Gertboard is a kitchen sink expansion board that provides you with many opportunities for experimentation and learning.

This book is chock-full of tasks that give you guided experience at setting up, configuring, troubleshooting, and building projects with your Raspberry Pi. I strongly suggest you work through as many tasks as possible.

I am confident that by the time you finish this volume, you will not only be able to discuss the Raspberry Pi intelligently, but you'll also have an excellent baseline familiarity with practical, applied computer science.

Who Can Use This Book

Ah, now we come to the “Exactly who is this book intended for?” question. Actually, I have a very detailed view of those of you who will benefit most from this book:

- **Students and Teachers:** The Raspberry Pi was developed by educators for educators and their students. Due to its open architecture and low price point, people can use the Pi as a platform for learning how computer hardware works at a low level with minimal risk. After all, if the worst happens and you fry your Pi, your investment loss is limited to \$25 or \$35.
- **Hardware and Software Hackers:** As I discussed earlier in this Introduction, hacking has myriad goals. Here I refer to those who want to leverage the Pi to accomplish some business or personal goals, with or without the addition of third-party extension hardware and software.
- **Tech Enthusiasts:** These people are do-it-yourselfers (DIYers) who are of the mind, “It’s cheaper for me to make it myself,” or better yet, “I can make this better than anything I can buy.” If you are among these individuals, then kudos to you! You are among a small elite.

How to Use This Book

I hope this book is easy enough to read that you don't need instructions. That said, a few elements bear explaining.

First, I love to provide relevant websites, but as you know, some URLs are absurdly long and difficult to transcribe. To that end, I make use of the is.gd (<http://is.gd>) URL shortening service. I hope you find my is.gd “shortie” URLs convenient. One important note about those is.gd URLs: they are case-sensitive, so if you type the URL <http://is.gd/6zwzWT> as <http://is.gd/6ZWZWT> or some other variation, the link will not work correctly.

Second, this book contains several special elements, presented in what we in the publishing business call “margin notes.” There are different types of margin notes for different types of information, as you see here.

NOTE

This is a note that presents information of interest, even if it isn't wholly relevant to the discussion in the main text.

TASK: THIS IS A TASK

This is a step-by-step procedure that gives you practice in whatever technology we're discussing at the moment. Almost every chapter in this book has at least a couple tasks for you to perform that will help you get the most out of your Raspberry Pi.

There's More Online...

When you need a break from reading, feel free to go online and check out my personal website, located at timwarnertech.com. Here you'll find more information about this book as well as other work I do. And if you have any questions or comments, feel free to send me an email at tim@timwarnertech.com. I do my utmost to answer every message I receive from my readers. Thanks very much for reading my book, and I hope that it exceeds your expectations!

This page intentionally left blank

CHAPTER 3

A Tour of Raspberry Pi Peripheral Devices

I got my start learning about electricity and electronics not through school but by horsing around with a Science Fair 160-in-1 electronics project kit my parents bought for me from Radio Shack for my tenth birthday.

As you can see in Figure 3.1, this wooden-framed kit enabled kids like me to prototype electrical circuits without having to solder any components together. The various “doo dads” on the kit’s circuit board kept me engaged and entertained for many, many hours.



FIGURE 3.1 I learned electronics by studying (playing?) with this Radio Shack project kit.

Fast-forward to the twenty-first century—now we have the Raspberry Pi, a \$35 personal computer the size of a credit card! In this chapter, I’d like to pique your curiosity by sharing with you the most popular peripheral devices—which is to say, electronic equipment that is connected to the Pi by means of a cable instead of soldered directly to the board—that exist in today’s marketplace.

If you want to really dig into physical computing and circuit building, you will indeed need to take an iron and braid in hand and learn to solder. I have you covered, though: You learn about all of the most popular starter kits and technician tools at the end of this chapter.

Let's begin!

Circuit Prototyping Equipment

In electronics, *prototyping* refers to mocking up an idea in a way that the circuit can easily be rebuilt. To that end, the breadboard is by far one of the most useful tools you can have in your possession.

A *breadboard* is a plastic block that is perforated with small holes that are connected internally by tin, bronze, or nickel alloy spring clips. Take a look at Figure 3.2 as a reference while I explain how these devices work.

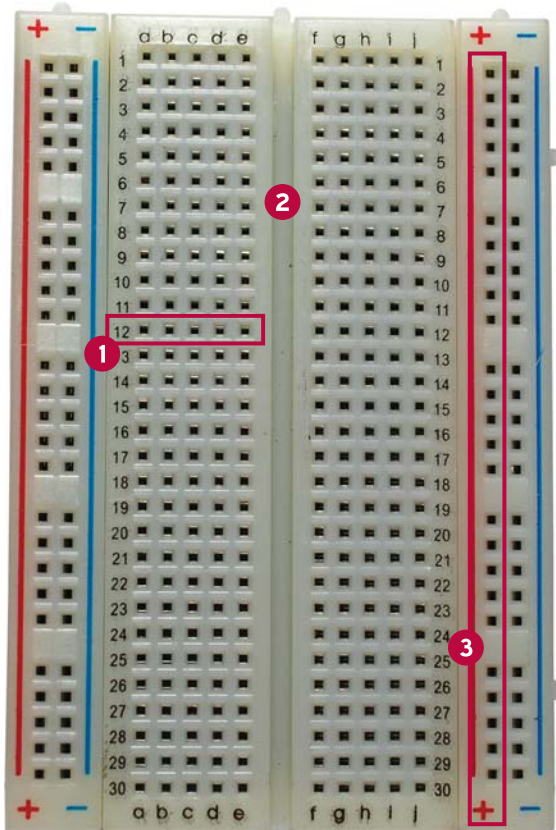


FIGURE 3.2 Anatomy and physiology of a breadboard. A terminal strip is labeled 1, the bridge is labeled 2, and a bus strip is labeled 3.

First of all, see the empty area that runs down the center line of the breadboard? This region is called the *bridge*. It's a physical barrier that prevents current on one side from interacting with current on the other side. Thus, the breadboard is *bilaterally symmetric*, which is a fancy way of saying it consists of two mirror image halves that represent two separate circuits.

When you mount integrated circuit chips that use the dual inline package (DIP) format on a breadboard, be careful to align the opposing sets of pins on opposite sides of the bridge to prevent circuit overflow.

If you are wondering what a DIP looks like, whip out your Raspberry Pi board and look below the GPIO header: the voltage regulators labeled RG1, RG2, and RG3 are DIPs.

The horizontally numbered rows of perforations represent the breadboard's *terminal strips*. Any wires that you connect in a single row share a single electrical circuit. Breadboards come in several different sizes, and each has its own number of terminal strips.

For instance, full-sized breadboards typically include 56 to 65 connector rows, while smaller breadboards normally have 30 rows.

Finally, there are the horizontally aligned perforations that line the outer edges of the breadboard. These are called *bus strips*, and they constitute "power rails" for your prototype circuits. One connector column represents supply voltage (positive), and the other represents ground (negative).

In sum, the breadboard is the perfect platform for prototyping electrical circuits because you don't need to solder anything. Instead, you can simply "plug and play" with ICs, resistors, lead wires, buttons, and other components.

Of course, all of this background information on breadboarding suggests the question, "Why would I, a Raspberry Pi owner, want to prototype anything?"

Great question! Here's the deal: If you want to use your Raspberry Pi to interact with the outside world, whether that interaction is controlling a robot, snapping pictures from 30,000 ft in the air, or creating a solar-powered weather station, you'll need to learn how to use prototyping hardware such as breadboards, resistors, jumpers, and the like.

On the Raspberry Pi, the 26 General Purpose Input/Output (GPIO) pins are used to "break out" the Pi onto a breadboard. You can do this by using two different types of cable:

- **Ribbon cable:** This flat cable connects to all the GPIO pins simultaneously
- **Jumper wire:** This wire connects a single GPIO pin to a terminal on the breadboard. Jumper wires are also called straps, and you'll use several of them when we use the Gertboard expansion board in Chapter 20, "Raspberry Pi and the Gertboard."

A ribbon cable and jumper wires are shown in Figure 3.3.

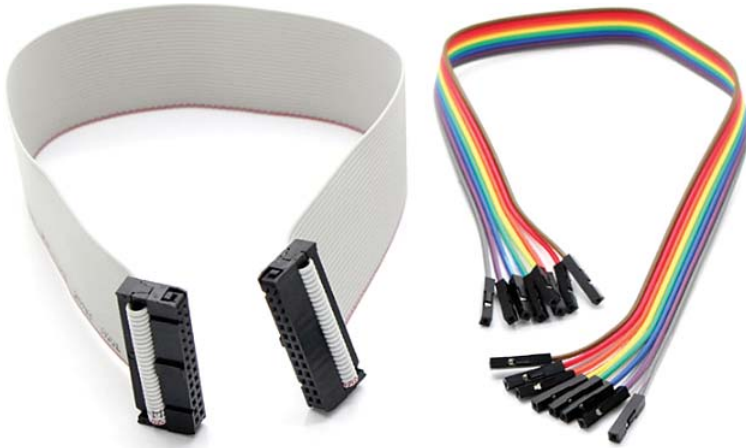


FIGURE 3.3 Ribbon cable at left and female-to-female jumper wires at right.

Breakout boards provide an excellent and convenient way to connect your Raspberry Pi to a solderless breadboard. I recommend the Pi Cobbler kit, sold by Adafruit Industries (<http://is.gd/b4LIQ7>).

As you can see in Figure 3.4, you mount the Pi Cobbler board across the breadboard bridge (do you like my alliteration?). The ribbon cable connects from the Cobbler to the Pi's GPIO header on the other side of the connection.

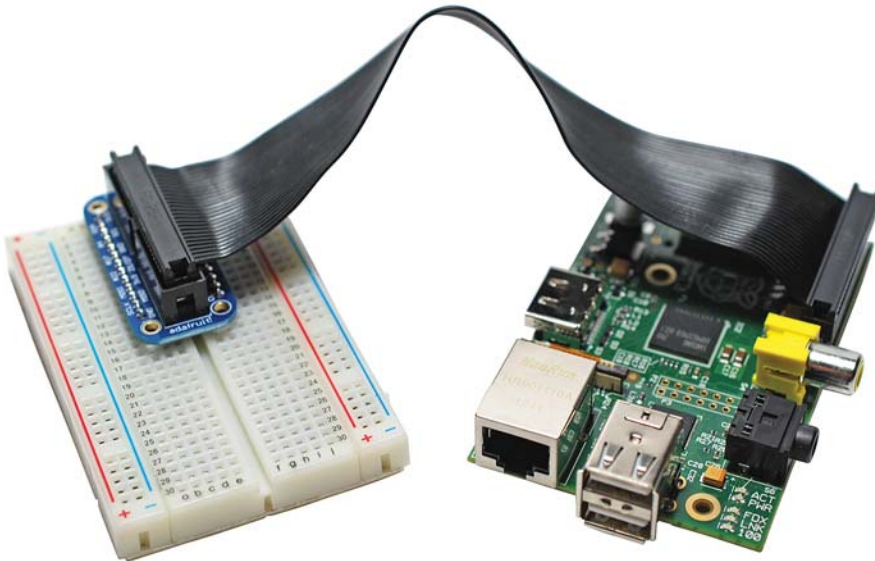


FIGURE 3.4 The Pi Cobbler is a quick and easy way to expand your Raspberry Pi to a breadboard.

Once you've broken out your Pi to the breadboard, you have the proverbial world available to you. In point of fact, the latter part of this book walks you through some real-world projects that take advantage of the Raspberry Pi-breadboard connection.

Single-Board Microcontrollers

Recall from our initial discussion in Chapter 1, “What is the Raspberry Pi?” that a microcontroller is a PCB that is designed primarily for a small number of time-dependent tasks.

The big benefit of integrating your Raspberry Pi with a microcontroller is that you can connect to an almost endless number of analog and digital sensors. This means you can write programs that detect and take action on the following and more:

- Light
- Moisture
- Sound/Volume
- Contact
- Motion

The Arduino platform (www.arduino.cc) is a suite of electronics prototyping PCBs that are dearly loved by artists, designers, inventors, and hobbyists for their ease of use and flexibility.

Hobbyists have developed some pretty cool technology by using Arduino microcontrollers: motion sensors, home automation systems, MIDI controllers, radon detectors...the list of project ideas is seemingly endless.

The Raspberry Pi-Arduino heavenly match is discussed in Chapter 19, “Raspberry Pi and Arduino.” For now, however, let's go over the basic “gotchas” of this electronic marriage:

- **Connection options:** To connect your Raspberry Pi to an Arduino board, you can either use a USB cable or a I²C (pronounced *eye-squared-see*) serial link. You can see the Pi and Arduino UNO lined up side-by-side in Figure 3.5.
- **Voltage differences:** We already know from Chapter 2, “Hardware Components Quick Start,” that the Raspberry Pi accepts 5V inbound power but operates at 3.3V internally. By contrast, the Arduino operates externally and internally at 5V. Consequently, when joining Pi with Arduino you need to invest in an external voltage regulation solution to avoid burning up your Pi.

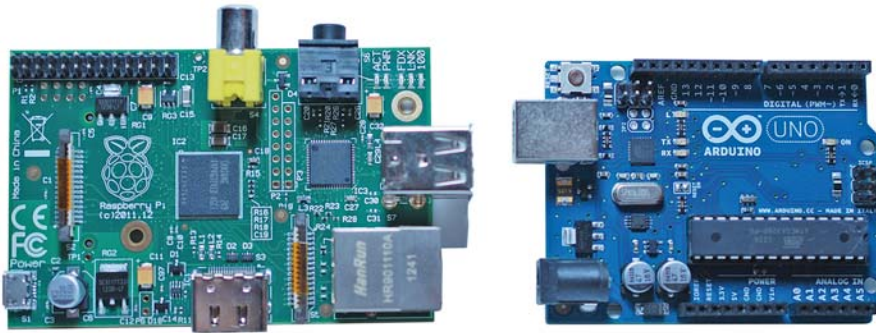


FIGURE 3.5 You can connect an Arduino board directly to the Raspberry Pi by using USB, serial, or GPIO connections.

- **Administration:** Recall that the lack of an operating system is one of the defining characteristics of a single-board microcontroller. Therefore, in an Arduino/Raspberry Pi nexus, all your Arduino programming happens on the Pi, and you upload your Arduino “sketches” to that hardware over the serial or USB connection.

NOTE: BUT WAIT, THERE'S MORE!

Perhaps the most elegant way to connect your Arduino board to your Raspberry Pi is to purchase the Alamode shield (<http://is.gd/4H3aVwV>). The Alamode is an Arduino device that connects directly to the Pi's GPIO header and provides a real-time clock, seamless connectivity to the Arduino microcontroller application programming interface (API), and voltage regulation to the Pi. It's a great deal!

Please note that despite its overwhelming popularity, the Arduino is not the only single-board microcontroller game in town. Here's a quick list of single-board microcontroller vendors that you might find useful:

- Texas Instruments MSP430 LaunchPad (<http://is.gd/xBAjco>)
- Teensy (<http://is.gd/rIElxy>)
- STM32 (<http://is.gd/TscRtp>)
- Pinguino (<http://is.gd/rdEpF5>)

The Gertboard

What the heck is a Gertboard, you ask? Gert van Loo is a computer electronics engineer who was the chief architect of the Raspberry Pi PCB. Gert designed the Gertboard as a Raspberry Pi expansion board, or *daughterboard*, that makes it easy to detect and respond to physical (analog) events such as voltage changes, motor state changes, and the like.

NOTE: WHAT'S A DAUGHTERBOARD?

A *daughterboard* is a printed circuit board that is intended to extend the functionality of mainboard. In this context, the Raspberry Pi is the mainboard, and the Gertboard is the daughterboard. In full-sized PCs, daughterboards, which are also called mezzanine boards or piggyback boards, are often used to enable expansion cards to mount on their side, parallel to the motherboard, in the name of making the PC's form factor as slim as possible. Question: Do you think the world needs a brotherboard? How about a second-cousin-twice-removedboard?

The Gertboard is awesome because it saves you the work of building circuits with a breadboard. The Gertboard PCB is literally covered with useful electrical components like the following:

- Tactile buttons
- LEDs
- Motor controllers
- Digital-to-analog and analog-to-digital converters

The Gertboard and the Pi connect together directly by means of (what else?) the Pi's GPIO header. Figure 3.6 shows you a close-up of the amazing Gertboard.

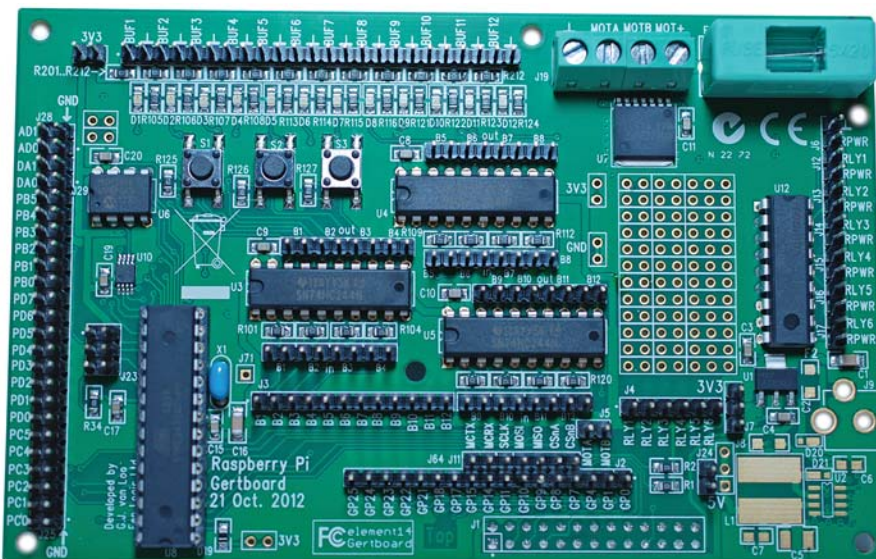


FIGURE 3.6 The Gertboard provides a truly seamless expansion experience for the Raspberry Pi.

Just wait—you get to use the Gertboard in Chapter 20.

Single-Board Computers

The BeagleBone (<http://is.gd/A5m89F>) is perhaps the Raspberry Pi's chief competitor in the single-board computer market. The BeagleBone is, like the Raspberry Pi, an ARM-based, credit card-sized Linux computer.

The BeagleBone is actually the smaller sibling of the BeagleBoard. Both boards are manufactured by the legendary Texas Instruments (TI), which lends immediate credibility to the Beagle projects.

Serious gearheads prefer the BeagleBone because its ARM Cortex A8 processor (running at 720 MHz) supports the ARMv7 instruction set, as opposed to the ARMv6 set included with the Pi.

Because of its support for ARMv7, the BeagleBone's benchmark performance is much better than that of the ARMv6-equipped Raspberry Pi. You also have a wider range of Linux distributions to choose from with ARMv7-compatible devices such as the 'Bone.

The BeagleBone and the Raspberry Pi aren't exactly "finger in glove" partners like the Arduino and Pi. Really, they are competitors in the same or highly similar market space. Table 3.1 compares and contrasts the technical specifications for both systems.

TABLE 3.1 Comparison Between the Raspberry Pi Model B and the BeagleBone

	Raspberry Pi Model B	BeagleBone
List Price	\$35 USD	\$89 USD
Dimensions	3.37" x 2.125"	3.4" x 2.1"
SoC	Broadcom BCM2835	TI AM3359
CPU	700 MHz, single core	720 MHz, single core
GPU	VideoCore IV	PowerVR
RAM	512MB	256MB
USB	2	2
Storage	SD	MicroSD
GPIO	Yes	Yes
Camera	Yes (future development)	No
Development	Squeak/Scratch; Python	Squeak/Scratch; Python
Video Out	HDMI; Composite	None

Here's a noncomprehensive list of other single-board computer manufacturers:

- Cotton Candy (<http://is.gd/quXmJu>)
- CuBox (<http://is.gd/B6hvsZ>)
- Gumstix (<http://is.gd/29EJ4A>)
- PandaBoard (<http://is.gd/5of9yx>)

“Why is it important that I understand the Raspberry Pi’s competition?” you might ask. In my estimation, it is important for you to know that there exist alternatives to the Raspberry Pi. You may find, for example, that the Pi is the best fit for the types of learning goals and projects that you have in mind. By contrast, you may also save yourself time, money, and frustration by concluding at the outset that you should consider an Arduino or a BeagleBone rather than a Pi.

Relevant Technician Tools

To perform all aspects of physical computing with your Raspberry Pi, you’ll need a few electrical tools. Chief among these is the *digital multimeter*, an instrument with which you can measure electrical current, voltage, and resistance.

For instance, you can run a quick verification of the Raspberry Pi’s 5V power supply voltage by using a multimeter and the TP1 and TP2 test points on the Model B board.

To locate the TP1 and TP2 test points, take a closer look at Figure 2.2. You’ll see TP1 located just above and to the right of the C2 capacitor and TP2 just to the left of the RCA video output (assuming you are looking at the PCB with the Raspberry Pi logo facing up).

TASK: CHECK RASPBERRY PI VOLTAGE WITH A MULTIMETER

1. Turn on your multimeter and set the dial to a low voltage threshold (for instance, 20V is good). If you own an autoranging multimeter, you don’t need to worry about this step.
2. Disconnect all peripherals (including the SD card) from your Pi except for the Micro USB power supply cable.
3. Power up your Pi.
4. Place the hot (red) lead to the TP1 test point, and place the ground (black) lead to the TP2 test point. You want to connect the leads simultaneously.
5. Verify that the multimeter shows a net voltage of approximately 5V (see Figure 3.7).

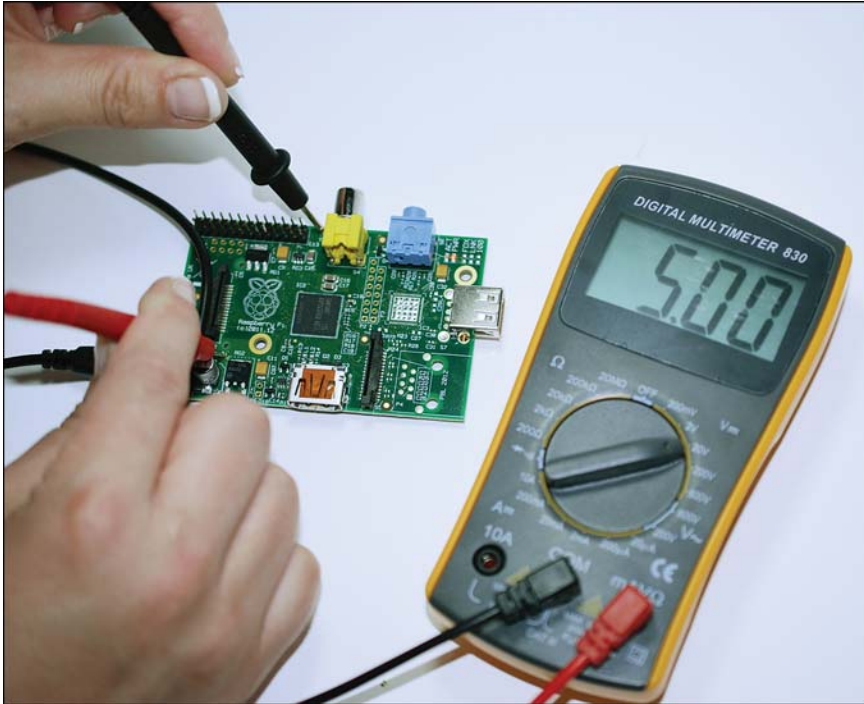


FIGURE 3.7 You can use a multimeter and two PCB test points to quickly verify Raspberry Pi voltage.

You'll also need a *soldering iron*, which is a tool you use to permanently join electrical components and to extend circuits.

The very idea of soldering intimidates some people, but the kernel idea at play is really quite simple. You heat up the soldering iron to 700 degrees Fahrenheit or so and then melt solder into a junction between two other conductive components. When the solder dries, you have a permanent connection that allows electrical current to flow between the soldered components.

In a nutshell, *solder* is a fusible metal alloy that, as a conductor, can transmit electricity.

For Raspberry Pi projects, I recommend you get an adjustable 30W pen-style soldering iron for maximum flexibility. You also should purchase a spool of 60/40 lead rosin-core solder with a 0.031-inch diameter. The 60/40 means that the solder consists of 60 percent tin and 40 percent lead.

To make your soldering experience as user-friendly as possible, you might also want to look into the following relevant soldering accessories:

- **Solder sucker:** This vacuum tool makes short work of removing melted solder particles.
- **Solder wick:** This material, also called *desoldering braid*, is used in conjunction with the solder sucker to remove solder from your components.
- **Soldering stand:** We have only two hands, and if you are operating the iron with one hand and the solder wire with the other, then how the heck can you position your components to be soldered? Because using The Force isn't an option (probably), a soldering stand makes this easy.

Raspberry Pi Starter Kits

With so many hardware options available to Raspberry Pi enthusiasts, figuring out where to start can seem overwhelming to even experienced tech junkies.

For the befuddled (among whose number I once counted myself a member), several third-party vendors assemble so-called *Raspberry Pi starter kits* that include everything you need to start building Raspberry Pi-based projects. Heck, most of the kits even include a Model B board! This section lists a few such kits.

The *Adafruit Raspberry Pi Starter Pack* (<http://is.gd/cf3ply>) lists for \$104.95. Besides the Model B, you also get the following components for your money:

- Adafruit clear acrylic Pi box
- 3' long Micro USB cable
- 5V 1A power adapter
- USB TTL console cable
- 4GB SD card
- Assembled Adafruit Pi Cobbler kit with GPIO cable
- USB microSD card reader
- Large, full-size breadboard
- Breadboarding wires
- 10' long Ethernet cable
- Embroidered Raspberry Pi badge
- 5 x 10K ohm resistors
- 5 x 560 ohm resistors
- 1 Red 10mm diffused LED
- 1 Green 10mm diffused LED
- 1 Blue 10mm diffused LED
- 3 tactile pushbuttons
- Light-sensitive resistor photocell
- 1 x 1uF capacitor

The *Maker Shed Raspberry Pi Starter Kit* (<http://is.gd/YIWK7Z>) costs \$129.99 and also includes a Model B board. In addition, the pack includes the following:

- Adafruit Pi Cobbler Breakout Kit
- MAKE: Pi Enclosure
- 2-port USB wall charger (1A) and USB cable
- 4GB Class 4 SDHC flash memory card
- Deluxe, full-sized breadboard
- Mintronics Survival Pack with 60+ components, including voltage regulators, trim pots, LEDs, pushbuttons, battery snaps, capacitors, diodes, transistors, and resistors
- HDMI high-speed cable, 1.5 feet
- Deluxe breadboard jumper wires

The previous two Raspberry Pi starter kits are decidedly weighted toward electronics experimentation. There are also vendors who will sell you kits that include the Pi as well as core peripherals.

For instance, consider the *CanaKit Raspberry Pi Complete Starter Kit* (\$89.95, <http://is.gd/HnsiOJ>), which includes the following parts:

- Raspberry Pi Model B
- Clear case
- Micro USB power supply
- Preloaded 4GB SD card
- HDMI cable

MCM Electronics (<http://is.gd/6cAfs3>) offers several different Raspberry Pi kits. Their *Raspberry Pi Enhanced Bundle* lists for \$79.99 and includes the following:

- Raspberry Pi Model B
- 5V 1A Micro USB power supply
- Raspberry Pi case
- 4GB SD card preloaded with Raspbian (We discuss Raspbian in exhaustive detail in Chapter 4, “Installing and Configuring an Operating System.”)
- 88-Key USB mini keyboard
- USB optical mouse

I mentioned in the previous chapter that you should consider purchasing a case for your Raspberry Pi. This is important for the following reasons at the least:

- Protection against electrostatic discharge (ESD).
- Protection against physical damage (remember that the Pi PCB is delicate).
- Some of the cases make the Pi look cool!

Although all cases I've seen fit both Model A and Model B (after all, both models use the same PCB), I suggest you purchase cases specific to each model. You'll find that Model B cases with Model A hardware create big gaps into which dust and other detritus can easily reach the delicate inner components.

Also as I said in the last chapter, be sure to select a case that gives easy access to the GPIO pins (and potentially the CSI camera interface) from outside the case. You want to provide yourself with as much flexibility as possible with your Pi.

Next Steps

If you've been reading this book sequentially thus far (and I sincerely hope you have), you should have a crystal-clear picture of what is necessary to get started with Raspberry Pi development.

You've heard the age-old adage, "You've got to learn to crawl before you can walk." Well, what this means for our purposes is that you need to learn how to write computer programs before you can link your Pi to an expansion board and start to perform meaningful work. Before we even get to writing code with Scratch and Python, you need to get comfortable with Linux. That's the subject we tackle in Chapter 4.

This page intentionally left blank

INDEX

A

- Accessories command (LXDE main menu), 93**
- Acorn BBC Micro personal computer, 7-8**
- Acorn RISC Machines (ARM) processors, 8**
- add-ons, media centers, installing, 197-198**
- addresses, static IP, setting, 106**
- advanced options, Raspo-Config utility, 84**
- AlaMode (Arduino), 324-325, 332-335**
- Amтел microcontroller, programming with, 354-357**
- analog versus digital signals, 317**
- analogWrite() function, 331**
- Android Device, 2**
- Apache, 235**
- APIs (application programming interfaces), 10, 221-226**
- Apple OS X, 49**
- Application Launch Bar (LXDE), 92**
- application layer (OS), 48**
- applications**
 - creating, 129-132
 - Dogeball Challenge, 137-149
 - Pi Store, downloading, 97
 - Python, writing, 158-160
 - remixing, 151
 - uploading, 149-151
 - writing simple, 168-170
 - XBMC, 181
- apt-get parameters, 313**
- Arch Linux ARM, 54**
- Arduino, 315, 317-319**
 - AlaMode, 324-325, 332-335
 - connecting, 321-323
 - development workflow, 323-332
 - Esplora, 316
 - IDE, 323-332, 355-357
 - LEDs, fading, 330-331
 - LilyPad, 315-316
 - Mega 2560, 316
 - microcontrollers, 10-11, 14
 - PAPERduino, 315
 - Playground, 321
 - shields, 317
 - starter kits, 319
 - Uno, 315-321
- ARM11 processors, 8, 20**
- ARM (Acorn RISC Machines) processors, 8**
- Atari 2600 Video Computer System (VCS), 199**
- Atmel ATmega 328P AVR microcontroller (Gertboard), 339**
- Atmel RISC chip, 10**
- audio**
 - HDMI cables, 26
 - Model B boards, 27-29
- Audio Book Player, 2**
- autoclean parameter, 313**
- Automated Chicken Coop Door, 2**
- autoremove parameter, 313**

B

- Bald Wisdom blog, 324**
- Banzi, Massimo, 315**
- BASIC programming, 7**
- batteries, webcams, adding to, 273-274**
- BBC Micro personal computers, 7-8, 12**
- BeagleBone, 40-41**
- benchmarking, 305-307**
- bilaterally symmetric breadboards, 35**
- binary large object (BLOB) driver, 15**
- bin directory, 70**
- black boxes, 9, 17**
- Blendtec, 181**
- BLOB (binary large object) driver, 15**
- blocks**
 - Scratch, 127-129
 - unlinking, 131
- Blocks palette (Scratch), 125, 127**
- Blum, Richard, 154**

boards

- audio, 27-29
- breadboards, 34-35
- breakout, 36
- daughterboards, 38-39
- Gertboards, 38-39
- networking, 27
- single-board microcontrollers, 37-38
- video, 27-29

Boolean blocks, 129**Boolean data type (Python), 172****boot behavior, adjusting, Rasp-Config, 88****boot directory, 70****boot partition contents, 310****breadboarding, 350-354****breadboards, 34-35****breakout boards, 36****Bresnahan, Christine, 154****bridges, 35****Broadcom BCM2835 system-on-a-chip, 10****Broadcom VideoCore IV GPU, 179****browsers, 289****browsing Internet, proxy servers, 278-280****bs command, 60-63****building Minecraft servers, 226-231****bus strips, 35****BYTEmark (nbench) benchmark, 307****Bytes data type (Python), 172****C****cables, 25-26**

- Ethernet, 24-25
- ribbon, 35

camera

- enabling, 82
- Model B board, 30

Camera Board, 253-256, 259

- capturing still pictures, 259-263
- installing, 256-258
- interface, 254
- recording video, 263-266
- specifications, 255

Camera Serial Interface (CSI) connectors, 30**Campidoglio, Enrico, 302****CanaKit Raspberry Pi Complete Starter Kit, 44****capacitors, PCBs (printed circuit boards), 17****Cap blocks, 129****Cartesian coordinate system, 130****cases, 32****C blocks, 129****cd command, 63, 72-73****Cheese-Powered Radio-Controlled Car, 1****chipssets, 31****Chrome web browser, 289****Chromium web browser, 120****Chun, Wesley, 178****circuit prototyping equipment, 34-37****classes, 221****classes, SD cards, 23****class libraries, 221****clients, Hamachi, installing, 284-285****codecs, XBMC, 181****command prompt, Raspbian, 67-69****commands**

- bs, 60-63
- cd, 63, 72-73
- cp, 77
- dd, 60-62
- if, 60-63
- ifconfig, 104-105
- ls, 71-72
- LXDE main menu, 93
- man, 75-76
- mkdir, 77
- mv, 77
- nano, 74-75
- of, 60-63
- passwd, 74, 81
- pwd, 70-73
- Rasp-Config, 80-85
- rm, 77
- rmdir, 77
- shutdown, 76-77
- startx, 69
- sudo, 60-62, 73-74
- Terminal, 69-77

community

- Raspberry Pi, 357
- Scratch, 123-124

computer science, 9**concatenation, Python, 173****configuration**

- Arduino IDE, 324-327
- GPIO-based IR receivers, 190-192
- headless (q) Raspberry Pi, 110-117
- Joomla, 245-248
- LXDE UI, 98-102
- phpMyAdmin, 243-245
- Raspbmc, 184-185
- RetroPie, 201-202, 208-209
- SimpleCV vision library, 274-276

- third-party USB webcams, 267-274
 - web servers, 240
 - wired Ethernet, 104-107
 - wireless Ethernet, 107-110
 - XBMC, media detection, 194-196
 - connecting Arduino**
 - Arduino, 321-323
 - Gertboard, 341-343
 - connectors**
 - CSI connectors, 30
 - RCA, 28
 - versus ports, 25
 - console video game emulation, RetroPie, 199-200**
 - configuring, 201-202
 - installing, 200-202
 - joysticks, 208-209, 212-215
 - playing games, 209-210
 - setting up controls, 206-207
 - transferring ROMs to, 203-206
 - content, web servers, transferring to, 240-243**
 - Control blocks (Scratch), 127**
 - controls, RetroPie, setting up, 206-207**
 - cp command, 77**
 - CPUs (central processing units)**
 - ARM11, 8
 - verifying status, 303
 - CPU Usage Monitor icon (LXDE), 92**
 - CPU/GPU memory splits, adjusting, 307-311**
 - CraftBukkit, 227**
 - CrazyWorms2, 97**
 - Creative Commons, 151**
 - Creative mode (Minecraft), 218**
 - cron utility, 271**
 - CSI (Camera Serial Interface) connectors, 30**
 - Cyberduck, 204**
- D**
- data processing tasks, personal computers, 9**
 - data types, Python, 172**
 - daughterboards, 38-39**
 - Dawson, Anne, 163**
 - dd command, 60-62**
 - Debian, 52**
 - Debian Linux**
 - command prompt, 67-69
 - GUI. *See* GUI (graphical user interface)
 - Raspi-Config utility, 80-85
 - Terminal, 68-77
 - updating software, 77-80
 - Debian Reference icon (LXDE), 89**
 - debugging Dodgeball Challenge, 147-149**
 - dedicated audio, HDMI cables, 26**
 - default gateways, 106**
 - deleting software, 94-95**
 - desktop**
 - changing background, 98
 - enabling boot to, 82
 - LXDE (Linux X11 Desktop Environment), 87-94, 98-102
 - desoldering braids, 43**
 - dev directory, 70**
 - development workflow, Arduino, 323-332**
 - DeviantArt Picture Frame, 1**
 - DHCP (Dynamic Host Configuration Protocol), 104**
 - dialog boxes**
 - Import Background, 130
 - Internet Properties, 289
 - Panel Preferences, 100
 - Dictionaries data type (Python), 172**
 - Digital Clock icon (LXDE), 92**
 - digital multimeters, 41-43**
 - digital versus analog signals, 317**
 - Digital Video Interface (DVI-D), 27**
 - digitalWrite() function, 329**
 - diodes, PCBs (printed circuit boards), 18**
 - directories, 70-71**
 - discussion forums, 102**
 - displays, 25**
 - Display Serial Interface (DSI), 28**
 - Dive into Python 3, 178**
 - DIY Arduino, 319**
 - Dodgeball Challenge, 137-138**
 - building, 139-149
 - debugging, 147-149
 - setting up screens, 140-143
 - troubleshooting, 147-149
 - uploading, 149-151
 - dongles, 108**
 - Doom, 217**
 - downloading apps, Pi Store, 97**
 - drivers, 15, 48**
 - DSI (Display Serial Interface), 28**
 - DVI-D (Digital Video Interface), 27**
 - Dynamic Host Configuration Protocol (DHCP), 104**

E

Eames, Alex, 344
 Education command (LXDE main menu), 93
 electrostatic discharge (ESD), 27
 emulation, 200
 Emulation Station, 200-202
 encoding raspivid video files, 265-266
 ESD (electrostatic discharge), 27
 Esplora, 316
 Esplora (Arduino), 316
 ES-Scraper, installing, 211-212
 etc directory, 70
 Ethernet, 103

- configuring wired, 104-107
- configuring wireless, 107-110

 Ethernet cable, 24-25
 Ethernet Shield, 318
 exiting LXDE, 88
 expansion, Model B board, 31-32

F

Fedora Remix, 54
 file extensions, Windows, showing in, 206
 file system, expanding, 81
 FileZilla, 204, 241-242
 Firefox web browser, 289
 firmware, Linux, 53
 FM Radio Transmitter, 1
 forked distributions, 52
 forums, 102, 357
 FTP (File Transfer Protocol), installing and testing, 243
 functions, 329-331

G

Gaddis, Tony, 178
 gamepads versus joysticks, 212
 games. *See* Minecraft; Minecraft Pi; RetroPie
 game screens, setting up, 140-143
 gateways, VPN, building, 280-285
 General Purpose Input/Output (GPIO) pin-out (Gertboard), 339-341

- based IR receivers, configuring, 190-192
- interface, 31-32
- lighting LED directly from, 351-352
- module, loading in Python, 176-177
- pins, 35, 350-351
- schematic, 215

Gertboard, 38-39, 337-339

Amtel microcontroller, programming with, 354-357
 breadboarding, 350-354
 connecting, 341-343
 GPIO pinout. *See* General Purpose Input/Output (GPIO) pinout
 input/output (I/O), testing, 349-350
 installing and testing software, 343-346
 LEDs, testing, 346-349

Git, 311**Gnash player, 119****GNOME Linux GUI, 88****GNU General Public License, 15****GNU (GNU's Not Unix), 15****GPIO pinout. *See* General Purpose Input/Output (GPIO) pinout (Gertboard)****GPU/CPU memory splits, adjusting, 307-311****Gralla, Preston, 234****Graphics command (LXDE main menu), 93****GSM Shield, 318****GtkPerf benchmarking utility, 305****GUI (graphical user interface)**

- accessing Pi Store, 95-97
- configuring, 98-102
- installing, updating and removing software, 94-95
- LXDE, 87-94
- Raspbian, 87

H**Hamachi clients, installing, 284-285****Hamachi VPNs, building, 280-284****HardInfo benchmarking utility, 305****hardware**

- cables, 24-26
- circuit prototyping equipment, 34-37
- digital multimeters, 41-43
- Gertboards, 38-39
- Model B board, 27-31
- monitors, 25
- open source, 14-16
- power supply, 21-23
- printed circuit boards (PCBs), 17-20
- Secure Digital (SD) card, 23-24
- single-board computers, 40-41
- single-board microcontrollers, 37-38

starter kits, 43-45
 USB hubs, 24
 USB keyboards and mice, 26-27
hardware layer (OS), 48
Hat blocks, 128
HDMI (High Definition Multimedia Interface) cables, 26
HDMI (High Definition Multimedia Interface) ports, 27
“headless” Raspberry Pi, configuring, 110-117
hidden costs, 20-27
home directory, 70
Hostname option (Raspi-Config), 84
How the Internet Works, 234
HTTP servers, 233
Hypertext Pre Processor (PHP), 235

I

icons, LXDE, 89-92
ICs (integrated circuits), 18
IDLE 3 icon (LXDE), 90
IDLE icon (LXDE), 90
IDLE (Integrated Development Environment), Python, 154-156, 165-168
if command, 60, 63
ifconfig command, 104-105
Import Background dialog box, 130
input, 9
input/output (I/O), Gertboard testing, 349-350
installation
 Arduino IDE, 324-327
 Camera Board, 256-258
 ES-Scraper, 211-212
 FTP, 243
 Hamachi clients, 284-285
 Java, 227-230
 Joomla, 246
 LAMP stack, 236
 media center add-ons, 197-198
 Minecraft Pi, 219-221
 nbench, 306-307
 operating systems, 63-65
 phpMyAdmin, 244
 Raspbmc, 182-184
 RetroPie, 200-202
 software, 94-95
 web servers, 235-240

integrated circuits (ICs), 18
internationalization options, Raspi-Config utility, 82
Internet command (LXDE main menu), 93
Internet connections, encrypting, 277-278
Internet Properties dialog box, 289
Internet Protocol (IP), 103
interpreter, Python 3, interacting with, 156-158
I/O (input/output), Gertboard, testing, 349-350
IP (Internet Protocol), 103
ls command, 71-72

J-K

jacks, 25
Java, installing, 227-230
joystick control mappings, RetroPie, configuring, 208-209
joysticks, 212-215
jumper wires, 35

KDE Linux GUI, 88
kernel, Linux, 53
keyboards, USB, 26-27
key words, Python, 171
KindleBerry Pi, 1

L

L6203 motor controller (Gertboard), 339
LAMP stack, 234-236
layers, operating systems, 48
Learning Python, 163
LEDs (light emitting diodes), 29-30
 fading, Arduino, 330-331
 Gertboard, testing, 346-349
 GPIO, lighting directly from, 351-352
lib directory, 70
LibreOffice, 51
licenses, open source, 15
light emitting diodes (LEDs), 29-30
LilyPad (Arduino), 315-316
LinApple Apple II emulator, 206
Linux, 49-52, 235
 Arch Linux ARM, 54
 disadvantages, 51
 Fedora Remix, 54

- firmware, 53
- Occidentalis, 54
- Raspbian. *See* Raspbian
- Red Hat, 49
- runlevels, 312
- Ubuntu, 49-50

Lists data type (Python), 172

Logitech Rumblepad 2 joystick, 212

Logout command (LXDE main menu), 94

Logout icon (LXDE), 92

Looks blocks (Scratch), 127

loop() function, 329

lost+found directory, 70

Lutz, Rick, 163

LXDE (Linux X11 Desktop Environment), 68, 87-94

- configuring, 98-102
- icons, 89-92
- main menu, 92-94
- starting and exiting, 88

LXPanel, customizing, 100

LXPanel icon (LXDE), 92

LXTerminal, customizing, 99

LXTerminal icon (LXDE), 90

M

Main Menu icon (LXDE), 91

main menu (LXDE), 92-94

Maker Shed Raspberry Pi Starter Kit, 44

man command, 75-76

manually adjusting memory splits, 309-311

manual method, overclocking, 304

Marriott, Jennifer, 246

MCP3002 Digital-to-Analog (D2A) converter (Gertboard), 339

MCP4801 Analog-to-Digital (A2D) converter (Gertboard), 339

media centers

- add-ons, installing, 197-198
- Raspbmc. *See* Raspbmc
- scraping media, 194-196
- XBMC, 179-181, 196-197

Media Companion, 195

media detection, XBMC, configuring, 194-196

media directory, 70

media scrapers, XBMC, 181

Mega 2560 (Arduino), 316

Memory Split option (Raspi-Config), 84

memory splits, adjusting, 307-311

Menu Bar (Scratch), 125

mice, USB, 26-27

microcomputers, Arduino, 14

microcontrollers

- Raspberry Pi-compatible, 11
- single-board, 37-38
- versus personal computers, 9-11

Microsoft Office, 51

Micro USB power ports, 29

Midori icon (LXDE), 89-91

Midori web browser, 118-120

Minecraft, 217-219

- building servers, 226-231
- game modes, 218
- learning to play, 218
- platforms, 219
- plugins, 232
- popularity, 218-219

Minecraft Pi

- accessing API, 221-226
- installing, 219-221

Minimize All icon (LXDE), 91

mkdir command, 77

mnt directory, 70

Model B board, 27

- audio, 27-29
- camera, 30
- expansion, 31-32
- networking, 27
- power/status information, 29-30
- processing, 31
- storage, 29
- video, 27-29

models, 11-14

modes

- Minecraft, 218
- overclocking, 302

ModMyPi cases, 32

modules, Python, 173-177

monitors, 25

motherboards. *See* boards

Motion blocks (Scratch), 127

multimeters, 41-43

mv command, 77

MySQL, 235, 239

N

nano command, 74-75

Nazarko, Sam, 182

nbench utility, 305-307

NES (Nintendo Entertainment System), 199
networking, 103-104

- Ethernet, 103-110
- headless Raspberry Pi, configuring, 110-117
- Model B board, 27
- virtual private networks (VPNs)
 - building gateways, 280-285*
 - encrypting Internet connections, 277-278*

New Out Of Box Software (NOOBS), 63-65

NICs (network interface cards), 47

Nintendo Entertainment System (NES), 199

nodes (Tor), 295

NOOBS (New Out Of Box Software), 63-65

Numbers data type (Python), 172

O

O'Brien, Stephen, 218

Occidentalis, 54

OCR Resources icon (LXDE), 90

of command, 60, 63

Office, 51

Office command (LXDE main menu), 93

Official Joomla! Book, The, 246

Openbox Configuration Manager, 100-101

OpenELEC, 54, 179

OpenOffice, 51

open source, 14-16

operating systems, 47-49, 54

- Arch Linux ARM, 54
- Fedora Remix, 54
- installing, 63-65
- layers, 48
- Linux, 49-53
- Occidentalis, 54
- OpenELEC, 54
- Raspbian. *See* Raspbian
- RaspBMC, 54
- RISC OS, 54

Operators blocks (Scratch), 127

Opera web browser, 289

opt directory, 71

Osborn, Kevin, 324

OS layer, 48

Other command (LXDE main menu), 93

output, 9

overclocking, 83, 299-300

- manual method, 304
- modes, 302
- Raspi-Config method, 302-303
- warranty implications, 300-301

Overscan option (Raspi-Config), 84

P

Panel Preferences dialog box, 100

PAPERduino, 315

parameters, apt-get, 313

passwd command, 74, 81

passwords, changing, 81

PCBs (printed circuit boards), 17-20

PCManFM icon (LXDE), 91

Pen blocks (Scratch), 127

PEP (Python Enhancement Proposal), 163

peripheral devices, 33

- circuit prototyping equipment, 34-37
- digital multimeters, 41-43
- Gertboards, 38-39
- single-board computers, 40-41
- single-board microcontrollers, 37-38
- starter kits, 43-45

Persson, Markkus PNotchN, 217

PetRockBlog RetroPie GPIO Adapter, 214

PHP (Hypertext Pre Processor), 235-238

phpMyAdmin, configuring, 243-245

Picade Arcade Cabinet, 1

Pi Cobbler board, 36

PicoBoard, 132-136

pictures, capturing, Camera Board, 259-263

Pi Hut cases, 32

Pi in the Sky, 1

Pilgrim, Mark, 178

pinMode() function, 329

Pi-Powered Bitcoin Miner, 1

Pi-Powered Motion Detector, 1

Pi-Powered Weather Station, 2

Pi Store

- accessing, 95-97
- downloading apps, 97

Pi Store icon (LXDE), 89

platforms, Minecraft, 219

Playground (Arduino), 321

playing content, XBMC, 196-197

playing games, RetroPie, 209-210

plugins

- Minecraft servers, 232
- XBMC, 181

plugs, 25

ports, 25-29

power/status information, Model B board, 29-30

power supply, 21-23

Practical Guide to Linux Commands, Editors, and Shell Programming, A, 74

Preferences command (LXDE main menu), 94

printed circuit boards (PCBs), 17-20

Privoxy

installing and configuring, 286-287
testing configuration, 287-291

proc directory, 71

processing, 9

processing, Model B board, 31

programming

Amtel microcontroller, 354-357
Python. *See* Python
Scratch. *See* Scratch

Programming Arduino: Getting Started with Sketches, 327

Programming command (LXDE main menu), 93

projects

Dodgeball Challenge, 137-149
remixing, 151
uploading, 149-151

Projects website, Scratch, 126

protocols

SSH (Secure Shell), 112-114
VNC (Virtual Networking Computing),
110-117
stacks, LAMP, 234

Proto Shield, 318

prototyping, 34-37

proxy servers

browsing Internet, 278-280
building, 286-287
pointing workstation computers at, 289-291
Tor, building, 294-297

pwd command, 70-73

Python, 121, 153-154, 165, 171-178

concatenation, 173
data types, 172
IDLE, 154-156, 165-168
interacting with interpreter, 156-158
keywords, 171
modules, 173-177
programs, writing, 158-160, 168-170
running scripts, 161-162
sample code repositories, 163
type casting, 171-173
variables, 171

Python API, Minecraft, accessing, 221-226

Python Enhancement Proposal (PEP), 163

Python Fundamentals LiveLessons, 178

Python Games icon (LXDE), 90

Python programming language, 9

Python test suite, installing and testing,
344-346

Python Tutorial, 178

Q-R

Quake III Arena timedemo, 305

Raspberry Pi, 7

benefits, 8-9
configuring headless, 110-117
hidden costs, 20-27
models, 11-14
naming of, 9
open source, 14-16
orgins, 7
purchasing, 16

Raspberry Pi Enhanced Bundle, 44

Raspberry Pi Foundation, 7

Raspberry Pi Keyboard Computer, 1

Raspberry Pi Media Center, 182

Raspberry Pi Robot, 1

Raspbian, 52, 55-57, 200

command prompt, 67-69
GUI. *See* GUI (graphical user interface)
Raspi-Config utility, 80-85
SD cards, 55-63
Terminal, 68-77
testing image, 65
updating software, 77-80
XBMC media center, 180-181

Raspbmc, 54, 179, 182-185

configuring, 184-185
controlling from web browser, 189
installing, 182-184
installing add-ons, 197-198
remote controls, 186-192
requirements, 182
scraping media, 194-196
transferring content to PI, 192-193

Raspi-Config utility, 80-85

adjusting boot behavior, 88
adjusting memory splits, 308
advanced options, 84
internationalization options, 82
overclocking, 302-303

raspistill, capturing still pictures, 259-262

raspidvid, 259, 263-266

RasPlex, 179

Rastrack database, 83, 292-294

RCA connectors, 28

RCA video cables, 26

rechargeable batteries, webcams, adding to,
273-274

recording video, Camera Board, 263-266

Red Hat Linux, 49

Reduced Instruction Set Computing (RISC)
 processing architecture, 8
Relay Shield, 318
remixed distributions, 52
remixing projects, 151
remote controls, Raspbmc-compliant, 186-192
remotely connecting to Raspberry Pi, 110-117
Reporter blocks, 129
resistors, PCBs (printed circuit boards), 17
RetroArch, 200
retroarch.cfg file, 207
RetroPie, 199-200
 configuring, 201-202
 ES-Scraper, installing, 211-212
 installing, 200-202
 joystick control mappings, configuring, 208-209
 joysticks, 212-215
 playing games, 209-210
 SD card image, 200
 setting up controls, 206-207
 transferring ROMs to, 203-206
RetroZone, USB adapters, 213
ribbon cables, 35
RISC OS, 54
RISC (Reduced Instruction Set Computing)
 processing architecture, 8
RJ-45 Ethernet jacks, 24-25
RJ-45 Ethernet interface, 27
rm command, 77
rmdir command, 77
ROM (read-only memory), RetroPie, transferring to, 203-206
root directory, 71
Rossum, Guido van, 154
RPi.GPIO library, 344
rpix86 DOS 5.0 emulator, 206
Run command (LXDE main menu), 94
run directory, 71
runlevels, Linux, 312

S

Safari web browser, 289
sample code repositories, Python, 163
Sams Teach Yourself Python Programming for Raspberry Pi, 154
sbin directory, 71
scraping media, 194-196
Scratch, 122-124, 137-138
 applications, creating, 129-132
 blocks, 127-129, 131
 Cat, setting up, 143-144
 community, 123-124
 Dodgeball Challenge, 137-149
 PicoBoard, 132-136
 Projects website, 126
 remixing projects, 151
 uploading projects, 149-151
 user interface, 124-129
 versions, 122-123
Scratch Cat, 130
Scratch icon (LXDE), 89
Scratch programming language, 121
Screenlock icon (LXDE), 92
Scripts Area (Scratch), 125
scripts, Python, running, 161-162
SD (Secure Digital) cards, 23-24, 55-56
SD card slots, 29
SDHC (Secure Digital High Capacity) cards, 24
SDXC (Secure Digital eXtended Capacity) cards, 24
Secure Shell (SSH), 110-114
security and privacy device, 277
 browsing via proxy server, 278-280
 building proxy servers, 286-287
 Privoxy, 287-291
 proxy servers, Tor, 294-297
 Rastrack, 292-294
 virtual private networks, building gateways, 280-285
 virtual private networks (VPNs), encrypting Internet, 277-278
selinux directory, 71
Sensing blocks (Scratch), 127
Serial Peripheral Interface (SPI), enabling, 344-345
Server Outdated, 229
servers
 Minecraft, 226-232
 proxy. *See* proxy servers
 web. *See* web servers
Sets data type (Python), 172
setup() function, 329
SFTP, transferring content to web servers, 241
sharing projects, 149-151
shields (Arduino), 317
shutdown command, 76-77
SimpleCV vision library, 274-276

- single-board computers, 40-41
- single-board microcontrollers, 37-38
- SMSC LAN9512 Ethernet controller, 21
- Sobell, Mark, 74
- software
 - installing, updating and removing, 94-95
 - open source, 14-16
 - updating, 77-80
- soldering irons, 42
- Sound blocks (Scratch), 127
- Sound & Video command (LXDE main menu), 93
- Spigot, 227
- SPI (Serial Peripheral Interface), enabling, 344-345
- Sprite Header Pane (Scratch), 125
- Sprites Pane (Scratch), 125
- srv directory, 71
- SSH option (Raspi-Config), 84
- SSH (Secure Shell), 112-114
- Stack blocks, 129
- Stage (Scratch), 125
- starter kits, 43-45
- starter kits, Arduino, 319
- starting LXDE, 88
- Starting Out with Python, 178
- startx command, 69
- static IP addresses, setting, 106
- still pictures, capturing, Camera Board, 259-263
- storage, Model B board, 29
- Streaming Music Jukebox, 1
- Strings data type (Python), 172
- Stuff About Code website, 252
- “Style Guide for Python Code,” 163
- subnet masks, 103
- sudo command, 60-62, 73-74
- Super Mario Brothers, 210, 217
- Survival mode (Minecraft), 218
- sys directory, 71
- System Tools command (LXDE main menu), 93
- sysv-rc-conf utility, 311-312

T

- temperature, verifying status, 303
- Terminal
 - commands, 69-77
 - starting sessions, 68-69
 - updating software, 77-80

- terminal strips, breadboards, 35
- testing
 - FTP, 243
 - voltage, 41
- third-party USB webcams, 266
 - configuring, 267-274
 - SimpleCV vision library, 274-276
- time-lapse webcams, setting up, 270-272
- tmp directory, 71
- tools
 - multimeters, 41-43
 - soldering irons, 42
- Tor proxy servers, building, 294-297
- Tor routers, 295
- Torvalds, Linus, 49
- traces, PCBs (printed circuit boards), 17
- transistors, PCBs, 18
- troubleshooting Dodgeball Challenge, 147-149
- Tuples data type (Python), 172
- typecasting Python, 171-173

U

- Ubuntu Linux, 49-50
- UI (user interface) Scratch, 124-129
- ULN2803a open collector drivers (Gertboard), 339
- Ultimate Player’s Guide to Minecraft, The, 218
- unlinking blocks, Scratch, 131
- Uno (Arduino), 315-321
- Update option (Raspi-Config), 84
- updating software, 77-80, 94-95
- uploading
 - media content, 192-193
 - projects, 149-151
- Upton, Eben, 7, 16
- USB hubs, 24
- USB keyboards, 26-27
- USB mice, 26-27
- USB webcams, 266, 274-276
- Useful Modules list (Python Wiki), 175
- user layer (OS), 48
- user passwords, changing, 81
- usr directory, 71

V

- van Loo, Gert, 38, 337
- var directory, 71
- variables, 128, 171
- Variables blocks (Scratch), 127
- versions, Scratch, 122-123
- VGA (Video Graphics Array) ports, 28
- vias, PCBs (printed circuit boards), 17
- video
 - Model B boards, 27-29
 - recording, Camera Board, 263-266
- VideoCore graphics processing unit (GPU), 15
- video game emulation, RetroPie, 199-200
 - configuring, 201-202
 - configuring joystick control mappings, 208-209
 - installing, 200-202
 - installing ES-Scraper, 211-212
 - joysticks, 212-215
 - playing games, 209-210
 - setting up controls, 206-207
 - transferring ROMs to, 203-206
- Video Graphics Array (VGA) ports, 28
- Virtual Desktops icon (LXDE), 91
- Virtual Networking Computing (VNC), 110-117
- virtual private networks. *See* VPNs
- VNC (Virtual Networking Computing), 110-117
- voltage
 - testing, 41
 - verifying status, 303
- VPNs (virtual private networks)
 - building gateways, 280-285
 - encrypting Internet connections, 277-278

W

- Walmsley, Ryan, 292
- Waring, Elin, 246
- warranty implications, overclocking, 300-301
- web browsers, 289
 - Chromium, 120
 - Midori, 118-119
- webcams
 - adding rechargeable batteries to, 273-274
 - Camera Board. *See* Camera Board

- setting up, 269
 - SimpleCV vision library, 274-276
 - third-party USB, 266-274
 - time-lapse, setting up, 270-272
- web pages, 233
- web servers, 233-234
 - configuring, 240
 - content, 233, 240-243
 - installing, 235-240
 - Joomla, setting up, 245-248
 - LAMP stack, 234-236
 - phpMyAdmin, configuring, 243-245
 - placing on public Internet, 249-252
- WiFi Config icon (LXDE), 89
- Wi-Fi Shield, 318
- Windows, showing file extensions, 206
- wired Ethernet, configuring, 104-107
- wireless Ethernet, configuring, 107-110
- WiringPi library, 344
- workstation computers, pointing at proxy servers, 289-291

X-Z

- Xbian, 179
- XBMC, 179-181
 - applications, 181
 - codecs, 181
 - configuring media detection, 194-196
 - installing add-ons, 197-198
 - media scrapers, 181
 - playing contents, 196-197
 - plugins, 181
 - scraping media, 194-196
- XBMC Media Center, 181
- XFCE Linux GUI, 88
- “Zen of Python, The”, 163
- Zeroconf (Zero-Configuration Networking), 186-187