Learn Electronics with Raspberry Pi

Physical Computing with Circuits,
Sensors, Outputs, and Projects

Stewart Watkiss

Apress®

Learn Electronics with Raspberry Pi: Physical Computing with Circuits, Sensors, Outputs, and Projects

Stewart Watkiss
Redditch
United Kingdom

ISBN-13 (pbk): 978-1-4842-1897-6
DOI 10.1007/978-1-4842-1898-3

ISBN-13 (electronic): 978-1-4842-1898-3

Library of Congress Control Number: 2016943841

Copyright © 2016 by Stewart Watkiss

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Steve Anglin

Technical Reviewer: Chaim Krause

Editorial Board: Steve Anglin, Pramila Balan, Louise Corrigan, Jonathan Gennick, Robert Hutchinson,

Celestin Suresh John, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie,

Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing

Coordinating Editor: Mark Powers

Copy Editor: Kezia Endsley

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com/9781484218976. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

For Amelia and Oliver

Contents at a Glance

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
■ Chapter 1: Getting Started with Electronic Circuits.....	1
■ Chapter 2: All About Raspberry Pi.....	13
■ Chapter 3: Starting with the Basics: Programming with Scratch	23
■ Chapter 4: Using Python for Input and Output: GPIO Zero	55
■ Chapter 5: More Input and Output: Infrared Sensors and LCD Displays	89
■ Chapter 6: Adding Control in Python and Linux.....	127
■ Chapter 7: Creating Video with a Pi Camera	167
■ Chapter 8: Rolling Forward: Designing and Building a Robot	189
■ Chapter 9: Customize Your Gameplay: Minecraft Hardware Programming	217
■ Chapter 10: Making Your Circuits Permanent	235
■ Chapter 11: Let the Innovation Begin: Designing Your Own Circuits.....	247
■ Appendix A: Required Tools and Components	263
■ Appendix B: Electronic Components Quick Reference.....	271
■ Appendix C: Component Labeling	275
■ Appendix D: GPIO Quick Reference.....	281
Index.....	285

Contents

About the Authorxv

About the Technical Reviewerxvii

Acknowledgmentsxix

Introductionxxi

■ Chapter 1: Getting Started with Electronic Circuits..... 1

 Voltage, Current, and Resistance..... 2

 Ohm’s Law..... 3

 Electrical Safety 4

 Breadboard..... 5

 A First Breadboard Circuit 8

 Calculating the Resistor Value 10

 Static-Sensitive Devices..... 10

 More Circuits 11

■ Chapter 2: All About Raspberry Pi..... 13

 Introduction to the Raspberry Pi 13

 Raspberry Pi GPIO Ports 14

 Serial Communications/UART..... 16

 I²C: Inter-Integrated Circuit 16

 SPI: Serial Peripheral Interface Bus..... 16

 PWM: Pulse Width Modulation 17

 Getting Started with Raspbian Linux..... 17

 Connecting to the Raspberry Pi Using the Network 18

Secure Shell (ssh).....	19
Remote Desktop Using VNC (TightVNC)	20
More Raspberry Pi.....	21
■ Chapter 3: Starting with the Basics: Programming with Scratch	23
Introduction to Scratch.....	23
Scratch with GPIO Support.....	26
Controlling an LED Using Scratch GPIO Server.....	26
Light Emitting Diode (LED).....	27
Resistor	27
Connecting the LED to the Raspberry Pi	28
Adding an Input to the Scratch GPIO Program	31
Using a Switch as a Digital Input.....	31
Adding the Switch to the Circuit.....	32
Robot Soccer	34
Playing Robot Soccer.....	42
Mars Lander Arcade Game	43
Making an Enclosure	43
Adding Joystick and Switches.....	44
Wiring the Switches.....	45
Creating the Game	46
Playing the Game.....	53
More Games	53
■ Chapter 4: Using Python for Input and Output: GPIO Zero	55
Power Supplies	55
+5V from the Raspberry Pi GPIO.....	56
Another USB Power Supply	56
Other External Power Supplies.....	57
Main Power Supply.....	59
Batteries	59

Brighter LEDs with a Transistor	59
Transistor	59
Calculating the Resistor Sizes	61
Introduction to Python	63
Getting Started with Python GPIO Zero.....	66
Using a While Loop	68
Circuit Diagram and Schematics.....	68
Brighter LEDs with Darlington Transistors.....	73
Reading a Switch Input with Python GPIO Zero	77
Disco Lights with MOSFETs.....	79
Light Sequence Using a Python List (Array)	84
Switching AC Lights Using a Thyristor and a TRIAC.....	85
More GPIO Zero	87
■ Chapter 5: More Input and Output: Infrared Sensors and LCD Displays	89
PIR Sensor and Pi Camera.....	89
Controlling a Raspberry Pi Camera with picamera.....	89
Using a PIR Sensor as a Motion Sensor.....	92
Using the PIR Motion Sensor to Trigger the Camera.....	95
Infrared Transmitter and Receiver	96
Infrared Receiver	96
Infrared Transmitter	97
Infrared Transmitter and Receiver Circuit.....	98
Configuring the Infrared Transmitter and Receiver Using LIRC.....	99
Receiving Infrared Commands Using python-lirc	103
Sending an Infrared Code Using Python	105
More Infrared Devices	105
Changing Voltage with a Level Shifter.....	106
Voltage Divider Circuit to Reduce Input Voltage	106
Unidirectional Level-Shift Buffer	107
Bidirectional Level-Shifters	108

I ² C LCD Display: True or False Game.....	110
LCD Character Display	110
I ² C	110
I ² C Backpack for the LCD Character Display.....	111
True or False Game Circuit	112
Setting Up the I ² C Connection and Adding the Code.....	114
SPI Analog to Digital Input.....	120
Creating an Analog Value Using a Potentiometer	120
Analog-to-Digital Conversion.....	121
SPI (Serial Peripheral Interface Bus)	121
Potentiometer SPI Circuit.....	123
Accessing the ADC Using Python.....	124
More Input and Output	126
■ Chapter 6: Adding Control in Python and Linux.....	127
Taking the Next Steps in Python Programming	127
Creating Python Functions	130
Adding Disco Light Sequences Using Python Functions.....	130
Using a Python Main Function	133
Making Python Programs Executable.....	133
Handling Command-Line Arguments.....	134
Running Python Programs as a Service	136
Running Programs at Regular Intervals with Cron	137
Creating an Automated Lego Train Using Infrared	138
Dealing with Conflicts.....	142
Software to Control the Lego Train Using LIRC and GPIO Zero.....	143
Using the Internet of Things to Control the Model Train	144
Controlling Color Light Strips Using NeoPixels.....	151
Powering RGB LEDs.....	153
How the RGB LEDs Work.....	153
Installing the Python Module	153

Controlling RGB LEDs from Python	154
Creating a Graphical Application Using Pygame Zero.....	155
Adding an Icon to the Raspbian Desktop.....	163
More Linux and Programming	165
■ Chapter 7: Creating Video with a Pi Camera	167
Infrared Shutter Release	167
Designing the Film	172
Filming the Scenes.....	174
Editing the Video	175
Creating the Video on a Raspberry Pi	175
Editing the Video on a PC Using OpenShot	176
Adding Effects to the Video	178
Adding Special Effects Using GIMP	179
Making Changes to Video Frames	180
Using Green Screen Special Effects	181
Adding Sounds to the Video	185
Recording Sounds with Audacity	186
Making Your Own Background Music with Sonic Pi	187
Adding Sounds to OpenShot	187
More Video Editing	188
■ Chapter 8: Rolling Forward: Designing and Building a Robot	189
Selecting or Making a Robot Chassis.....	189
Two Motorized Wheels and Omnidirectional Wheel	190
Four Motorized Wheels	190
Caterpillar Tracks	190
Wheels that Steer	190
Buying a Kit or Making Your Own	190
Choosing a Raspberry Pi.....	191

Controlling the Motors.....	192
DC Motors and Stepper Motors	192
H-Bridge Motor Control.....	194
Controlling the Speed with Pulse Width Modulation (PWM)	198
Powering the Motors and the Raspberry Pi.....	199
Building the Circuit on a Breadboard.....	200
Add-On Motor Controller Boards	201
Controlling the Robot Using Python.....	202
Measuring the Distance Using an Ultrasonic Range Sensor	206
Controlling the Robot Using a Wii Remote.....	210
More Robotics	216
■ Chapter 9: Customize Your Gameplay: Minecraft Hardware Programming	217
Connecting to Minecraft with Python	217
Moving Around Using a Joystick	220
Building a House in Minecraft	223
Adding Status LEDs	227
Find the Glowstone Game	232
More Minecraft Hardware Programming.....	234
■ Chapter 10: Making Your Circuits Permanent	235
Soldering Basics.....	235
Gathering the Essential Tools	235
Choosing Solder.....	237
Safety Tips When Soldering.....	238
Soldering to a Printed Circuit Board	239
Soldering Direct to Leads	240
Stripboard.....	241
Perfboard	241
Raspberry Pi Prototyping Boards.....	241
Cases and Enclosures	243
Testing Tools.....	243

Multimeter	244
Oscilloscope	245
More Project Making	246
■ Chapter 11: Let the Innovation Begin: Designing Your Own Circuits	247
The Design Process in a Nutshell	247
Looking at Manufacturer Datasheets	248
Designing with Fritzing	251
Designing a Circuit Diagram/Schematic	251
Design Conventions	254
Creating a Breadboard Layout	255
Creating a Stripboard Layout	256
Designing a Printed Circuit Board	256
Powering the Raspberry Pi	260
78xx Linear Voltage Regulator	260
Buck Converter	260
Designing More Circuits	261
■ Appendix A: Required Tools and Components	263
Tools Required	263
Basic Breadboard Circuits	263
Crimping and Soldering Tools	263
Manufacturing Tools for Enclosures	264
Meters and Test Equipment	264
Components for Each Project	264
■ Appendix B: Electronic Components Quick Reference	271
Resistors	271
Variable Resistors	271
Switches	271
Diode	272
Light Emitting Diode (LED)	272

Multi-Colored LEDs.....	272
Bipolar Transistor	272
Darlington Transistor	273
MOSFET Transistor	273
Capacitor	273
Thyristor	273
Triac.....	273
■ Appendix C: Component Labeling	275
Resistor Color Codes	275
SMD Resistors	277
Electrolytic Capacitors.....	277
Polyester Capacitors	278
Ceramic Capacitors	279
■ Appendix D: GPIO Quick Reference.....	281
GPIO Pin Layouts	281
GPIO Ports with Alternative Functions.....	282
Index.....	285

About the Author

Stewart Watkiss has been a keen electronics hobbyist since the early 1990s. He first studied electronic engineering at Huddersfield Technical College and subsequently at the University of Hull, where he earned a master's degree in electronic engineering.

He then became more involved in the software side, studying the Linux operating system and programming. During this time he launched a web site (www.penguintutor.com) to teach Linux and to help those working toward Linux certification.

His interest in electronics was revitalized thanks in part to the Raspberry Pi. Stewart created a number of projects, some of which have been featured on the Raspberry Pi blog and *The MagPi* magazine. Stewart also volunteers as a STEM Ambassador, going into local schools to help support teachers and teach programming and physical computing to teachers and children.

About the Technical Reviewer

Chaim Krause presently lives in Leavenworth, Kansas, where the U.S. Army employs him as a simulation specialist. In his spare time, he likes to play PC games and occasionally develops his own. He has recently taken up the sport of golf to spend more time with his significant other, Ivana. Although he holds a bachelor's in political science from the University of Chicago, Chaim is an autodidact when it comes to computers, programming, and electronics. He wrote his first computer game in BASIC on a Tandy Model I Level I and stored the program on a cassette tape. Amateur radio introduced him to electronics, while the Arduino and the Raspberry Pi provided a medium to combine computing, programming, and electronics into one hobby.

Acknowledgments

My family has been very supportive during the writing of this book. Thank you to my wife Sarah for her support and especially to my children, Amelia and Oliver, who have been both a source of inspiration and enthusiastic testers of the games and activities.

I'd also like to thank the team behind the Raspberry Pi, including the Raspberry Pi foundation and the community that has grown around it. The Raspberry Pi has reinvigorated my love of electronics, making it possible to interact with hobby electronics projects in a way that I'd only dreamed about before. Raspberry Jams and community events have been a great way to meet the team behind the Raspberry Pi as well as other members of the community. All this has encouraged me to pursue it further.

There have been many other people who have helped in the making of this book. Special thanks to the technical reviewer, Chaim Krause, who has tested all the projects. Michelle Lowman encouraged me to write this book and Mark Powers ensured that the book moved forward. I'd also like to thank Corbin Collins for his comments and suggestions and to the rest of the team at Apress.

Introduction

Learning computer programming is fun in itself, but when the computer is connected to external sensors and outputs, your programs can interact with the real world. This is known as *physical computing* and it opens up the opportunity to create some fun projects.

I am a big fan of learning by doing. It's much easier to learn when you get to make the projects rather than just read about what other people have been done. It's even better when those projects are fun. This book covers simple projects that you can do at home to make games, control toys, create your own films, or just have fun.

The Raspberry Pi computer is great for learning physical computing, thanks to special pins that provide access to ports on the processor. These 40 pins (26 on earlier versions) provide a simple way to extend computing into the physical world. The circuits in this book are adding sensors, outputs, and electronic circuits to a Raspberry Pi. With a little bit of programming, these can do some pretty amazing things.

We will start with some simple circuits, which can be controlled from Scratch, and then move up to Python and some more complicated circuits. By the end of the book, we will have covered enough so that you can start designing your own circuits.

Most of the circuits can be created by plugging wires into a solderless breadboard, but there are tips on how to solder, which opens the possibilities further. You'll learn how to design custom circuit boards and look at how you can use some of the common Raspberry Pi add-on boards and HATs.

Who Is This Book For?

This book is for anyone who wants to learn about electronics and have fun in the process. This book focuses on fun projects so is great for older children and young adults. My eight-year-old son helped with some of the easier projects so young children could have a go with adult help. While the fun aspect appeals to younger adults, there's no maximum age for having fun, so this is just as useful to adults of any age who want to learn about electronic circuits and connecting to the Raspberry Pi.

You don't need to know anything about electronic circuits before you start. Having a basic knowledge about computers and computer programming will be useful, but is not required and will be explained as we go. We'll be using Scratch and Python, as they are good programming languages for those learning programming, but the electronic circuits can be controlled using any programming language that can communicate with the GPIO ports.

How to Use This Book

As with many books, you can read this book from cover to cover, or you can jump straight to the project that you find most interesting. The book introduces new concepts and components in each chapter. The first few chapters start with simple circuits and work through to more complicated circuits. There is an explanation of each circuit as you go, so for the first few chapters you will find it most useful to follow in order, making the circuits as you go. Most of the circuits in these first few chapters are based on low-cost components that should be within the reach of most readers.

Most of the projects are within a single chapter, but a few need some of the concepts explained in later chapters and so are split between the chapters. The notes explain which chapter you will need to refer to for the rest of the project.

Some of the later projects do use more expensive components or add-on boards for the Raspberry Pi or are designed to interact with more expensive toys. Where possible, boards have been chosen to keep costs as low as possible. You may want to just read about how the circuit works or look for the suggestions on how these can be adapted for use with cheaper components or items you already own.

Creating the examples in this book should not be considered the end. I hope that the information in this book will provide you with the inspiration and knowledge to go on to learn more about electronics and design your own circuits.

After working through the projects, it's useful to have a summary of the components so that you can refer to the book when designing your own circuits. To make it easier to refer back, I've added a summary of some of the components in the appendix along with some extra technical details that are useful when designing your own circuits.

Is Soldering Required?

When I've talked to teachers and students about electronic projects, I often get asked the question of whether soldering is required. Unfortunately, I think that this is something that puts some people off from learning electronics. I don't think it should.

First, a lot can be learned through creating circuits that don't need any soldering. Many of the projects in the first few chapters, and in some of the subsequent chapters, are designed to be made without any soldering. These are typically using solderless breadboards, but some can also be made using crocodile clips or with an inexpensive crimp tool. There are, however, some components that are not suitable for use on a breadboard, or that need a small amount of soldering so that they can be used with a breadboard. In fact many "breadboard friendly" components may need headers to be soldered on to them first.

The second point I'd like to make is that soldering is not as hard, expensive, or dangerous as some people have been led to believe. Chapter 10 explains soldering and will hopefully dispel some of the myths surrounding it. If you are still uncertain, see if you have a local maker club or Hackspace where you can speak to someone experienced in soldering.

Buying a Raspberry Pi

If you are reading this book then there is a good chance that you already have a Raspberry Pi. Since the Raspberry Pi was first released in 2012 there have been several versions. Some have had only minor changes, but one of the bigger changes was increasing the size of the GPIO connectors from 26 to 40. This was introduced in the Raspberry Pi B+ and the larger connector has been used on all new models since, including the Pi Zero and Raspberry Pi 2 and 3 models. While most of the projects in this book use only the first 26 pins, some of them do need the additional pins. If you don't have a Raspberry Pi or only have an original version with 26 pins on the connector, I recommend buying a Raspberry Pi 2, which includes a quad-core processor, or a Raspberry Pi 3 with the 64-bit processor and built-in Wi-Fi and Bluetooth. You don't need the extra processor power for the projects in this book, but it does make it possible to use the computer for other things. After all, nobody ever says, "This computer is too fast!"

The official Raspberry Pi suppliers are listed on the Raspberry Pi web site, but they are also widely stocked by various electronics and hobbyist suppliers, so you shouldn't have a problem finding a supplier.

Buying the Components

In order to follow the instructions, you will need some electronic components. There is no single kit that will provide all the items required; it will depend on which circuits you decide to make as well as the suggested variations. Details of the main components required for each project are listed in Appendix A. One thing that is worth stocking up on is a variety of different resistors. You may want to buy either an E6 or E12 series resistor multi-pack (see Appendix C for an explanation of the resistor series).

Most of the components are fairly common and can be bought from any good electronics retailer. There are several retailers that are specifically geared toward makers. In the United States, there are companies such as Adafruit and Sparkfun, and in the UK, two popular suppliers are Maplin and CPC Farnell. You may also want to look at Raspberry Pi retailers, many of whom have an increasing range of electronic sensors and other components. In particular, Pimoroni has created several add-on boards and HATs specifically for the Raspberry Pi. There are also international electronic retailers such as RS and Farnell or many smaller independent suppliers located around the world.

One thing about electronic components in particular is that a device with an almost identical name may work differently. It may be possible to substitute a similar product, but when a specific component is required, I've tried to list the specific part number to help find the correct one. Watch out for codes that are almost the same but have different electrical properties.

Installing Raspbian

The official operating system for the Raspberry Pi is *Raspbian*. It is based on Debian Linux, but has been customized for the Raspberry Pi and comes with some additional software. The operating system needs to be loaded onto an SD card before it can run. The easiest way to install Raspbian is through the NOOBs installer. More details on installing NOOBs are available on the Raspberry Pi web site at <https://www.raspberrypi.org/help/noobs-setup/>.

The Raspbian image has been updated on a regular basis since the Raspberry Pi was launched. If you've had your Raspberry Pi for some time and haven't updated it for a while, now is a good time. For many of the projects in this book, you need a version that was released from at least November 2015. If you downloaded the image after November 2015, that is sufficient, but if you installed prior to that or used a pre-installed image that may not have been updated to that version then you will need to perform an update.

In the event of minor updates to the operating system, it is normally sufficient to run a `dist-upgrade` to update to the latest version of the installed software. To do this, launch the terminal under the Accessories category of the Start menu. Once you are in the terminal shell, enter the following commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

If the version is much older, or it has been a long time since your last update, then it is recommended that you download a new version of NOOBs from the Raspberry Pi web site. The new image can then be installed onto the SD card (this will involve formatting the SD card, so wipe off any data that is stored on it).

If you are unsure of your version, you can run the following command from the terminal:

```
uname -a
```

This provides a date that indicates the date of the kernel. If no date is shown, assume it is an old version and needs to be upgraded. To upgrade from an old version, you will first need to wipe the SD card using the SD formatter tool https://www.sdcard.org/downloads/formatter_4/ and then copy the latest NOOBs files to the SD card.

Software Required

All the software required in this book is available for free and mostly available as open source software. This does include some libraries that have been created by others, and in those cases, links have been provided to the original location.

The source code used in the book can be typed in manually or downloaded from the Internet. For most short examples, I believe you can learn more through typing and experimentation, although downloads can be useful when trying to get a circuit working or when the amount of typing is enough to make your fingers ache. You can download the source code and media files as a ZIP file at:

<https://github.com/penguintutor/learnelectronics/archive/master.zip>

Unzip the `master.zip` file using the following commands:

```
unzip master.zip
mv learnelectronics-master learnelectronics
```

The first command unzips the file and the second command renames the directory to `learnelectronics`. The files are then contained in the relevant subdirectories.

You are free to use the software source code and circuit designs that I have provided in your own projects, but some of the accompanying files or suggested downloads may come under a different license. In particular, some of the files are provided under a Creative Commons license. Any additional copyright information is provided in the `LICENSE` file in the appropriate directory.

Safety Information

All the circuits in this book are designed to run at a low voltage and, as long as an appropriate power supply is used, they are safe to touch. To make these projects permanent may involve the use of power tools where the safety information related to the power tools needs to be followed.

Some of the projects use bright LED lights. Some people may be sensitive to high-frequency flashing lights, which in some circumstances may cause seizures. This is more likely if the flash rate is increased beyond that used in the supplied code. Please take this into consideration when modifying the source code, especially if you're using the lights in a public space. You should also avoid looking directly into any bright light, including the LED lights.

More Electronics

The projects in this book should be considered a starting point toward achieving more using electronics and the Raspberry Pi. At the end of each chapter there is a section that provides a summary of the key points in the chapter and suggestions on how the projects could be improved or ideas for related projects. These have been left as an exercise for the reader. I hope that this will result in future projects inspired from this book and I look forward to seeing some on the Internet in the future.