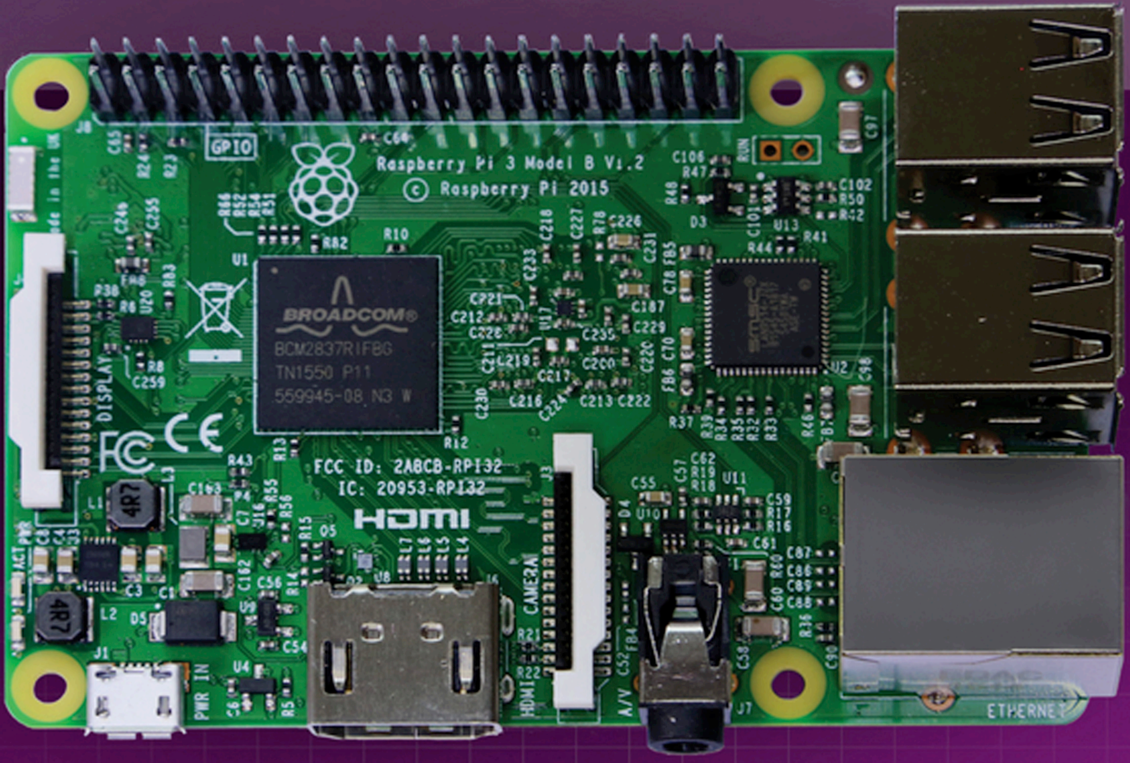
"*Exploring Raspberry Pi* is THE book to go to if you are interested in learning about the impressive physical computing capabilities of the Raspberry Pi platform. Derek Molloy imparts the electronics, programming, and embedded Linux skills that are vital to today's innovators in building the next generation of Internet of Things applications."

—**Eben Upton,** *Co-creator of the Raspberry Pi*

DEREK MOLLOY

EXPLORING RASPBERRY PI®

INTERFACING TO THE REAL WORLD WITH EMBEDDED LINUX®



WILEY

Exploring Raspberry Pi®

Interfacing to the Real World with Embedded Linux®

Derek Molloy

WILEY

Exploring Raspberry Pi®

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-1-119-18868-1
ISBN: 978-1-119-18870-4 (ebk)
ISBN: 978-1-119-18869-8 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. If improperly wired, circuits described in this work may possibly cause damage to the device or physical injury. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016933853

To Sally, Daragh, Eoghan, Aidan, and Sarah

(still in order of age, not preference!)

About the Author

Dr. Derek Molloy is a senior lecturer in the School of Electronic Engineering, Faculty of Engineering and Computing, Dublin City University, Ireland. He lectures at undergraduate and postgraduate levels in object-oriented programming with embedded systems, digital and analog electronics, and the Internet of Things. His research contributions have largely been in the fields of computer and machine vision, 3D graphics/visualization, and e-Learning.

Derek produces a popular YouTube video series that has introduced millions of people to embedded Linux and digital electronics topics. In 2013, he launched a personal web/blog site that is visited by thousands of people every day, and which integrates his YouTube videos with support materials, source code, and user discussion. In 2015, he published a book on the BeagleBone platform, *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*, which has been very well received.

Derek has received several awards for teaching and learning. He was the winner of the 2012 Irish Learning Technology Association (ILTA) national award for Innovation in Teaching and Learning. The award recognizes his learning-by-doing approach to undergraduate engineering education, which utilizes electronic kits and online video content. In 2012, as a result of fervent nominations from his students and peers, he was also awarded the Dublin City University President's Award for Excellence in Teaching and Learning.

You can learn more about Derek, his work, and his other publications at his personal website: `www.derekmolloy.ie`.

About the Technical Editor

Dr. Tom Betka came to the world of embedded systems development by way of a previous career in the aviation industry, and then as a physician practicing clinical medicine for well over a decade. During this time his love of computers and software development evolved toward the field of embedded systems, and his training in computer science culminated in a second undergraduate-level degree. After leaving clinical medicine, Dr. Betka began working in the world of software development and has served as a subject-matter expert in both medicine and embedded systems for various companies in the industry. His recent work has included projects at the NASA Kennedy Space Center and the Sierra Nevada Corporation. Tom's first love is the C-family of programming languages and using these languages to program 8-bit microcontrollers. As a Linux user for the past decade, he has also been working with the BeagleBone, BeagleBone Black, and Raspberry Pi devices for the last several years as well. His hobbies include advanced mathematics, aviation, high-powered model rocketry, and robotics. Also, he can often be found building prototype devices in his home-based machine shop. In a previous life, Tom worked for several years as a professional drummer—and was one of the first in his area to embrace the use of electronic percussion devices in live music scenarios.

Credits

Senior Acquisitions Editor
Aaron Black

Project Editor
Adaobi Obi Tulton

Technical Editor
Tom Betka

Production Editor
Barath Kumar Rajasekaran

Copy Editors
Keith Cline
Marylouise Wiack

Production Manager
Kathleen Wisor

Manager of Content Development and Assembly
Mary Beth Wakefield

Marketing Manager
Carrie Sherrill

Professional Technology & Strategy Director
Barry Pruett

Business Manager
Amy Knies

Executive Editor
Jody Lefevere

Project Coordinator, Cover
Brent Savage

Proofreader
Nancy Bell

Indexer
Nancy Guenther

Cover Designer
Wiley

Cover Image
Courtesy of Derek Molloy

Acknowledgments

Many thanks to everyone at Wiley Publishing once again for their outstanding work on this project: to Jim Minatel for encouraging me to take this book concept forward and for yet again supporting the realization of a book that engages in deeper learning; to Aaron Black and Jody Lefevere, for guiding the project forward, and for their support and help throughout the development of this book; to Jennifer Lynn, for keeping me on schedule and for always being available to answer my questions; to Adaobi Obi Tulton, the project editor, for driving this project to completion in the most efficient way possible—it was a real pleasure to work with such an accomplished and adept editor once again; to Keith Cline and Marylouise Wiack the copy editors, for translating this book into readable U.S. English; to Barath Kumar Rajasekaran, the production editor, and Nancy Bell, the proofreader, for bringing everything together to create a final, polished product.

Sincere thanks to Tom Betka, the technical editor, for the incredible amount of work and personal time he selflessly put into ensuring that the content in this book can be utilized seamlessly by readers. Following the publication of my book on the BeagleBone, Tom of this own volition provided valuable comment and feedback via the book website that further strengthened the title. I immediately thought of Tom when I took on this project, and I was delighted when he agreed to take on the role of technical editor. Tom is a scholar, a polymath, and indeed an inspiration, who was always available when I needed to talk through technical issues. This book has benefited hugely from his technical knowledge, world experience, and immense capabilities—I believe there could be no better technical editor for this topic!

Thanks to the thousands of people who take the time to comment on my YouTube videos, blog, and website articles. I truly appreciate all of the feedback,

advice, and comments—it has really helped in the development of the topics in my books.

The School of Electronic Engineering, Dublin City University, is a great place to work, largely because of its esprit de corps, and its commitment to rigorous, innovative, and accessible engineering education. Thanks again to all of my colleagues in the School for supporting, encouraging, and tolerating me in the development of this book. Thanks in particular must go to Noel Murphy and Conor Brennan for sharing the workload of the School Executive with me while I was so absorbed in the writing of this book. Thanks again to (my brother) David Molloy for his expert software advice and support. Thanks to Jennifer Bruton for her meticulous and expert review of circuits, software, and content that is used in this book. Thanks also to Martin Collier, Pascal Landais, Michele Pringle, Robert Sadleir, Ronan Scaife, and John Whelan for their ongoing expertise, support, and advice.

The biggest Thank You must of course go to my own family. This book was written over six months, predominantly at night and on weekends. Thanks to my wife Sally and our children Daragh, Eoghan, Aidan, and Sarah for putting up with me (again) while I was writing this book. Thank you Mam, Dad, David, and Catriona for your continued lifelong inspiration, support, and encouragement. Finally, thank you to my extended family for graciously excusing my absence at family events for another six months—I definitely have no excuses now (unless I write another book!).

Contents at a Glance

Contents

Introduction

The core idea behind the Raspberry Pi (RPi) project was the development of a small and affordable computing platform that could be used to stimulate the interest of children in core information and communications technology (ICT) education. The rapid evolution of low-cost system on a chip (SoC) devices for mobile applications made it possible to widely deliver the affordable RPi platform in early 2012. The impact was immediate; by February 2015, more than five million Raspberry Pi boards were sold. Given the proliferation of smartphones, the idea of holding in one hand computers that are capable of performing billions of instructions per second is easy to take for granted, but the fact that you can modify the hardware and software of such small yet powerful devices and adapt them to suit your own needs and create your own inventions is nothing short of amazing. Even better, you can now purchase a Raspberry Pi Zero for as little as $5 (the price of a large cup of coffee)!

The Raspberry Pi boards on their own are too complex to be used by a general audience; it is the ability of the boards to run embedded Linux in particular that makes the resulting platform accessible, adaptable, and powerful. Together, Linux and embedded systems enable ease of development for devices that can meet future challenges in smart buildings, the Internet of Things (IoT), robotics, smart energy, smart cities, human-computer interaction (HCI), cyber-physical systems, 3D printing, advanced vehicular systems, and many, many more applications.

The integration of high-level Linux software and low-level electronics represents a paradigm shift in embedded systems development. It is revolutionary that you can build a low-level electronics circuit and then install a Linux web server, using only a few short commands, so that the circuit can be controlled over the Internet. You can easily use the Raspberry Pi as a general-purpose Linux computer, but it is vastly more challenging and interesting to get underneath the hood and fully interface it to electronic circuits of your own design—and that is where this book comes in!

This book should have widespread appeal for inventors, makers, students, entrepreneurs, hackers, artists, dreamers—in short, anybody who wants to bring the power of embedded Linux to their products, inventions, creations, or projects and truly understand the RPi platform in detail. This is not a recipe book; with few exceptions, everything demonstrated here is explained at a level that will enable you to design, build, and debug your own extensions of the concepts presented. Nor does this book include any grand design project for which you must purchase a prescribed set of components and peripherals to achieve a very specific outcome. Rather, this book is about providing you with enough background knowledge and "under-the-hood" technical details to enable and motivate your own explorations.

I strongly believe in learning by doing, so I present low-cost, widely available hardware examples so that you can follow along. Using these hands-on examples, I describe what each step means in detail, so that when you substitute your own hardware components, modules, and peripherals you will be able to adapt the content in this book to suit your needs. As for that grand design project, that is up to you and your imagination!

In late 2014, I released a well-received book on the BeagleBone platform titled *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Given the focus of this book on embedded Linux and the emphasis on introducing the core principles, there are some similarities between the introductory content in that book and this book. However, this book has been written from first principles purely for the RPi platform, focusing on its strengths and addressing several of its weaknesses. I also took the opportunity to extend the coverage of the material to cover topics such as Linux kernel development, the Arduino as a service processor, Wi-Fi sensor nodes, XBee communication, MQTT messaging, the Internet of Things (IoT), platform as a service (PaaS), and much more. If you have a copy of *Exploring BeagleBone*, you should visit this book's website (`www.exploringrpi.com`) to compare the content in both books before you make your purchasing decision.

When writing this book, I had the following aims and objectives:

- To explain embedded Linux and its interaction with electronic circuits—taking you through the topics and challenges on the popular RPi platform.
- To provide in-depth information and instruction on the Linux, electronics, and programming skills that are required to master a pretty wide and comprehensive variety of topics in this domain.
- To create a collection of practical Hello World hardware and software examples on each and every topic in the book, from low-level interfacing, general-purpose input/outputs (GPIOs), buses, bus-attached analog-to-digital converters (ADCs), and universal asynchronous receiver/transmitters (UARTs) to high-level libraries such as OpenCV and the Qt Framework.

The book also covers more advanced topics such as low-level register manipulation and Linux loadable kernel module (LKM) development.

- To enhance and extend the interfacing capability of the RPi platform by developing frameworks for connecting it to circuits (e.g., SPI-based ADCs), to service processors (e.g., Arduino and NodeMCU), and to cloud-based IoT platforms and services.
- To ensure that each circuit and segment of code has a broad pedagogical reach and is specifically designed to work on the Raspberry Pi. Every single circuit and code example in this book was built and tested on the RPi platform (most on multiple board versions).
- To use the Hello World examples to build a library of code that you can use and adapt for your own Raspberry Pi projects.
- To make all the code available on GitHub in an easy-to-use form.
- To support this book with strong digital content, such as the videos on the DerekMolloyDCU YouTube channel, and the `www.exploringrpi.com` custom website that was developed specifically to support this book.
- To ensure that by the end of this book you have everything you need to imagine, create, and build *advanced* Raspberry Pi projects.

How This Book Is Structured

There is no doubt that some of the topics in this book are quite complex. After all, Raspberry Pi boards are complex devices! However, everything that you need to master them is present in this book within three major parts:

- Part I: Raspberry Pi Basics
- Part II: Interfacing, Controlling, and Communicating
- Part III: Advanced Interfacing and Interaction

In the first part of the book, I introduce the hardware and software of the RPi platforms in Chapters 1 and 2, and subsequently provide three primer chapters:

- Chapter 3, “Exploring Embedded Linux Systems”
- Chapter 4, “Interfacing Electronics”
- Chapter 5, “Programming on the Raspberry Pi”

If you are a Linux expert, electronics wizard, and/or software guru, feel free to skip these primers. However, for everyone else, I have put in place a concise but detailed set of materials to ensure that you gain all the knowledge required to effectively and safely interface to the Raspberry Pi. The remaining chapters refer to these primers often.

The second part of the book, Chapters 6–11, provides detailed information on interfacing to the Raspberry Pi GPIOs, buses (I^2C, SPI), UART devices, and USB peripherals. You learn how to configure a cross-compilation environment so that you can build large-scale software applications for the Raspberry Pi. Part II also describes how to combine hardware and software to provide the Raspberry Pi with the capability to interact effectively with its physical environment. In addition, Chapter 11, "Real-Time Interfacing Using the Arduino," shows you how to use the Arduino as a slave processor with the Raspberry Pi, which helps you to overcome some of the real-time constraints of working with embedded Linux.

The third and final part of the book, Chapters 12–16, describes how to use the Raspberry Pi for advanced interfacing and interaction applications such as IoT; wireless communication and control, rich user interfaces; images, video, and audio; and Linux kernel programming. Along the way, you encounter many technologies, including TCP/IP, ThingSpeak, IBM Bluemix, MQTT, Cgicc, Power over Ethernet (PoE), Wi-Fi, NodeMCUs, Bluetooth, NFC/RFID, ZigBee, XBee, cron, Nginx, PHP, e-mail, IFTTT, GPS, VNC, GTK+, Qt, XML, JSON, multithreading, client/server programming, V4L2, video streaming, OpenCV, Boost, USB audio, Bluetooth A2DP, text-to-speech, LKMs, kobjects, and kthreads!

Conventions Used in This Book

This book is filled with source code examples and snippets that you can use to build your own applications. Code and commands are shown as follows:

```
This is what source code looks like.
```

When presenting work performed in a Linux terminal, it is often necessary to display both input and output in a single example. A bold type is used to distinguish the user input from the output. For example:

```
pi@erpi ~ $ ping www.raspberrypi.org
PING lb.raspberrypi.org (93.93.128.211) 56(84) bytes of data.
64 bytes from 93.93.128.211: icmp_seq=1 ttl=53 time=23.1 ms
64 bytes from 93.93.128.211: icmp_seq=2 ttl=53 time=22.6 ms
...
```

The $ prompt indicates that a regular Linux user is executing a command, and a # prompt indicates that a Linux superuser is executing a command. The ellipsis symbol (. . .) is used whenever code or output not vital to understanding a topic has been cut. Editing the output like this enables you to focus on only the most useful information. In addition, an arrow symbol on a line entry indicates that the command spans multiple lines in the book but should be entered on a single line. For example:

```
pi@erpi /tmp $ echo "this is a long command that spans two lines in the ➔
 book but must be entered on a single line" >> test.txt
```

You are encouraged to repeat the steps in this book yourself, whereupon you will see the full output. In addition, the full source code for all examples is provided along with the book using a GitHub repository.

You'll also find some additional styles in the text. For example:

- New terms and important words appear in *italics* when introduced.
- Keyboard strokes appear like this: Ctrl+C.
- All URLs in the book refer to HTTP/S addresses and appear like this: `www.exploringrpi.com`.
- A URL shortening service is used to create aliases for long URLs that are presented in the book. These aliases have the form `tiny.cc/erpi102` (e.g., link two in Chapter 1). Should the link address change after this book is published, the alias will be updated.

There are several features used in this book to identify when content is of particular importance or when additional information is available:

WARNING **This type of feature contains important information that can help you avoid damaging your Raspberry Pi board.**

NOTE **This type of feature contains useful additional information, such as links to digital resources and useful tips, which can make it easier to understand the task at hand.**

FEATURE TITLE

This type of feature goes into detail about the current topic or a related topic.

EXAMPLE: EXAMPLE TITLE

This type of feature typically provides an example use case, or an important task that you may need to refer to in the future.

What You'll Need

Ideally, you should have a Raspberry Pi board before you begin reading this book so that you can follow along with the numerous examples. If you have not already purchased a Raspberry Pi board, I recommend the Raspberry Pi 3 Model B. Although it is presently the most expensive board ($35–$40), it is also the most powerful. This board has a 64-bit quad-core processor, a wired network adapter, wireless Ethernet, and onboard Bluetooth; therefore, it has all the features required to run any example in this book. You can purchase a Raspberry

Pi board in the United States from online stores such as Adafruit Industries, Digi-Key, SparkFun, and Jameco Electronics. They are available internationally from stores such as Farnell, Radionics, and Watterott.

A full list of recommended and optional accessories for the Raspberry Pi is provided in Chapter 1. If you do not yet have a Raspberry Pi, you should read that chapter before purchasing one. In addition, the first page of each chapter contains a list of the electronics components and modules required if you want to follow along. The book website (`www.exploringrpi.com`) provides details about how to acquire these components.

I purposefully focus the examples in this book on the lowest-cost and most widely available components, breakout boards, and modules that I could identify that meet the needs of the examples. This should help you follow along with many examples, rather than focusing your budget on a small few. Indicative prices are listed throughout the book to give you a feel for the price of the components before you embark on a project. They are the actual prices for which I purchased the items on websites such as `ebay.com`, `amazon.com`, and `aliexpress.com`.

NOTE **No products, vendors, or manufacturers listed in this book are the result of any type of placement deal. I have chosen and purchased all the products myself based on their price, functionality, and worldwide availability. Listed prices are indicative only and are subject to change. Please do your own research before purchasing any item that is listed in this book to ensure that it truly meets your needs.**

Errata

We have worked really hard to ensure that this book is error free; however, it is always possible that some were overlooked. A full list of errata is available on each chapter's web page at the companion website (`www.exploringrpi.com`). If you find any errors in the text or in the source code examples, I would be grateful if you could please use the companion website to send them to me so that I can update the web page errata list and the source code examples in the code repository.

Digital Content and Source Code

The primary companion site for this book is `www.exploringrpi.com`. It is maintained by the book's author and contains videos, source code examples, and links to further reading. Each chapter has its own web page. In the unlikely event that the website is unavailable, you can find the code at `www.wiley.com/go/exploringrpi`.

I have provided all the source code through GitHub, which allows you to download the code to your Raspberry Pi with one command. You can also easily view the code online at `tiny.cc/erpi001`. Downloading the source code to your Raspberry Pi is as straightforward as typing the following at the Linux shell prompt:

```
pi@erpi ~ $ git clone https://github.com/derekmolloy/exploringrpi.git
```

If you have never used Git before, don't worry; it is explained in detail in Chapter 3.

Now, on with even more adventures!

Part

I

Raspberry Pi Basics

In This Part