"Loaded with helpful information and coding tutorials, this exploration around the extensive capabilities of BeagleBone Black has me excited to connect everything I encounter to the Internet."

—**Christine Long,** BeagleBoard.org Foundation

DEREK MOLLOY

EXPLORING BEAGLEBONE®

TOOLS AND TECHNIQUES FOR BUILDING WITH EMBEDDED LINUX®

SECOND EDITION

WILEY

Exploring BeagleBone®

Second Edition

Exploring BeagleBone®

Tools and Techniques for Building with Embedded Linux®

Second Edition

Derek Molloy

WILEY

Exploring BeagleBone®: Tools and Techniques for Building with Embedded Linux®, Second Edition

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

ISBN: 978-1-119-53316-0
ISBN: 978-1-119-53315-3 (ebk)
ISBN: 978-1-119-53317-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2018962584

To Sally, Daragh, Eoghan, Aidan, and Sarah

About the Author

Dr. Derek Molloy is an associate professor in the Faculty of Engineering and Computing's School of Electronic Engineering at Dublin City University, Ireland. He lectures at undergraduate and postgraduate levels in object-oriented programming with embedded systems, digital and analog electronics, and connected embedded systems. His research contributions have largely been in the fields of computer and machine vision, embedded systems, 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 that integrates his YouTube videos with support materials, source code, and user discussion. In 2015, he published the first edition of this book on the BeagleBone platform, *Exploring BeagleBone*, and followed up in June 2016 with *Exploring Raspberry Pi*. Both of these books have received strong acclaim for both their depth of coverage and accessibility.

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. This learning-by-doing approach is strongly reflected in his books.

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

About the Technical Editor

Marcia K. Wilbur is a technical communicator consulting in the semiconductor field, focusing on industrial IoT (IIoT). Marcia holds degrees in computer science, technical communication, and information technology. As the Copper Linux User Group interim president, she is heavily involved in the East Valley maker community, leading regular Raspberry Pi, BeagleBone, Banana Pi/Pro, and ESP8266 projects, including home automation, gaming consoles, surveillance, network, multimedia and other "pi fun."

In addition to tinkering, she volunteers to aid disaster-stricken areas in getting access to public domain content to enable students to continue learning. For fun, she serves the community as the lead Debian developer for Linux Respin, a backup and distro customization tool.

Credits

Acquisitions Assistant
Devon Lewis

Project Editor
Adaobi Obi Tulton

Technical Editor
Marcia K. Wilbur

Production Editor
Barath Kumar Rajasekaran

Copy Editor
Kim Wimpsett

Production Manager
Katie Wisor

Content Enablement and Operations Manager
Pete Gaughan

Marketing Manager
Christie Hilbrich

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Brent Savage

Proofreader
Debbye Butler

Indexer
Johnna VanHoose Dinse

Cover Designer
Wiley

Cover Image
Courtesy of Derek Molloy

Acknowledgments

Many thanks to everyone at John Wiley & Sons, Inc once again for their exceptional work on this project: to Jim Minatel for encouraging me to take on the revision of this book and for supporting the enhancement of a book that engages in deeper learning; to Devon Lewis for guiding the project forward and for his expert support and help throughout the development of this book; 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 for the third time; to Kim Wimpsett, the copy editor, for translating this book into readable U.S. English; to Barath Kumar Rajasekaran, the production editor, for bringing everything together to create a final, polished product. Thanks to the technical editor, Marcia Wilbur, for her careful review and constructive feedback on the technical content in this book. Continued thanks to the technical editors from my previous titles, Tom Betka, Robert Zhu (Microsoft), and Jason Kridner (BeagleBoard.org Foundation), on whose advice this work is based. Thanks also to Cathy Wicks (Texas Instruments) for her advice and support in the development of this book.

Continued thanks to the thousands of people who take the time to comment on my YouTube videos, blog, and website articles. I always appreciate the feedback, advice, and comments—it has really helped in the development of the topics in all of my books.

The School of Electronic Engineering at 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 once again absorbed in a book. Thanks again to (my brother) David Molloy for his expert software advice and support. Thanks to Jennifer Bruton, Martin Collier, Pascal Landais, Michele Pringle, Robert Sadleir, Ronan Scaife, and John Whelan for their ongoing expertise, support, and advice on the various titles I have written.

The biggest thank-you must of course go to my own family once again. This revision 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 (once again) while I was writing this book. Thank you, Mam, Dad, David, and Catriona for your endless lifelong inspiration, support, and encouragement. Finally, thank you to my extended family for your continued support, understanding, and constancy.

Contents at a Glance

Contents

Introduction

The Beagle platform continues to amaze! Given the proliferation of smartphones, the idea of holding in one hand a computer that is capable of performing two billion 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 purchase a board for as little as $25 in the form of a PocketBeagle.

The Beagle boards on their own are too complex to be used by a general audience; it is the capability of the boards to run Linux 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, smart manufacturing, interactive art, 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 a Beagle board 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 his or her products, inventions, creations, or projects and truly understand the Beagle 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 here. Nor is there any grand design project at the end of this book for which you must purchase a prescribed set of components and peripherals to achieve a 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 examples using low-cost, widely available hardware 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 project or invention—that is left up to you and your imagination!

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 from mystery to mastery!
- 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), analog-to-digital converters (ADCs), buses, and UARTs, to high-level libraries such as OpenCV, Qt, and complex and powerful topics, such as real-time interfacing with the PRU-ICSS, and Linux kernel programming.
- To ensure that each circuit and segment of code is specifically designed to work with a Beagle board. Every circuit and code example in this book was built and tested on the BeagleBone Black wireless and PocketBeagle boards.
- To use the "Hello World" examples to build a library of code that you can use and adapt for your own Beagle projects.
- To make all of 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 a custom website, `www.exploringbeaglebone.com`.
- To ensure that by the end of this book you have everything you need to imagine, create, and build *advanced* Beagle board projects.

I wrote this second edition because of the popularity of the first edition of *Exploring BeagleBone*. The number of pages in this edition is more than 20 percent of the first edition, increased to include the following major additions:

- Full coverage of new Beagle boards, with a particular emphasis on the PocketBeagle and BeagleBone Black wireless boards
- Updated content to account for all recent changes to the Linux kernel and operating system
- Inclusion of electronics interfacing approaches, such as protection of I/O pins using optocouplers, the CAN bus, and many additional interfacing application examples using external I/O circuits
- New work on real-time interfacing using external slave processors, with a particular emphasis on building I^2C digital sensors
- A full account of new Internet of Things (IoT) full-stack frameworks, with an emphasis on MQTT and interfacing to Adafruit IO
- Full coverage of building wireless sensor networks using technologies such as Wi-Fi, Bluetooth, NFC, and ZigBee
- A complete rewrite of the PRU-ICSS chapter to account for Texas Instruments' decision to move away from UIO to Linux Remoteproc
- Inclusion of new work on writing Linux loadable kernel modules (LKMs)

Why the BeagleBone and PocketBeagle?

The Beagle boards are powerful single-board computers (SBCs), and while there are other SBCs available on the market, such as the Raspberry Pi and Intel NUC boards, the Beagle platform has one key differentiator—it was built to be interfaced to! For example, the Beagle board's microprocessor package even contains two additional on-chip microcontrollers that can be used for real-time interfacing—an area in which other Linux SBCs have significant difficulty.

Unlike most other SBCs, the Beagle boards are fully open-source hardware. The BeagleBoard.org Foundation provides source schematics, hardware layout, a full bill of materials, and comprehensive technical reference manuals, enabling you to modify the design of the Beagle platform and integrate it into your own product. In fact, you can even fork the hardware design onto Upverter (`www.upverter.com`) under a Creative Commons Attribution-ShareAlike license (see `tiny.cc/beagle001` for the full schematics). This is a useful feature should you decide to take your newest invention to market!

How This Book Is Structured

There is no doubt that some of the topics in this book are quite complex—the Beagle boards are complex devices! However, everything that you need to master the devices is present in the book within three major parts.

- Part I, "Beagle Board Basics"
- Part II, "Interfacing, Controlling, and Communicating"
- Part III, "Advanced Beagle Board Systems"

In the first part in the book, you learn about the hardware and software of the Beagle board platform in Chapters 1 and 2 and subsequently gain more knowledge through these three primer chapters:

- Chapter 3, "Exploring Embedded Linux Systems"
- Chapter 4, "Interfacing Electronics"
- Chapter 5, "Practical Beagle Board Programming"

If you are a Linux expert, electronics wizard, and/or software guru, then feel free to skip the primer chapters; however, for everyone else, you'll find a concise but detailed set of materials to ensure that you gain all the knowledge required to effectively and safely interface to your Beagle boards.

The second part of the book, Chapters 6 to 10, provides detailed information on interfacing to the Beagle board GPIOs, analog inputs, buses (I^2C, SPI, CAN bus), UART devices, USB peripherals, and real-time interfacing to slave processors. You'll learn how you can configure a cross-compilation environment so that you can build large-scale software applications. This part also describes how you can combine hardware and software to provide your board with the ability to interact effectively with its physical environment.

The final part of the book, Chapters 11 to 16, describes how the Beagle board can be used for advanced applications such as Internet of Things (IoT); rich user interfaces; images, video, and audio; real-time interfacing using the PRU-ICSS; and kernel programming. Along the way you will meet many technologies, including TCP/IP, ThingSpeak, Adafruit IO, PoE, Wi-Fi, Bluetooth, Zigbee, RFID, MQTT, cron, Apache, PHP, e-mail, IFTTT, VNC, GTK+, Qt, XML, JSON, multi-threading, client/server programming, V4L2, video streaming, OpenCV, Boost, USB audio, Bluetooth A2DP, text-to-speech, and Remoteproc.

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. Here's an example:

```
debian@ebb:~$ ping www.exploringbeaglebone.com
PING lb1.reg365.net (195.7.226.20) 56(84) bytes of data.
64 bytes from lb1.reg365.net (195.7.226.20): icmp_req=1 ttl=55 time=25.6 ms
64 bytes from lb1.reg365.net (195.7.226.20): icmp_req=2 ttl=55 time=25.6 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. I've edited the output like this to enable you to focus on only the most useful information. 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.

There are some additional styles in the text. Here are some examples:

- New terms and important words appear in *italics* when introduced.
- Keyboard strokes appear like this: Ctrl+C.
- All URLs in the book appear in this font: `www.exploringbeaglebone.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/beagle102` (e.g., link 2 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 Beagle 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.

What You'll Need

Ideally you should have a Beagle board before you begin reading this book so that you can follow along with the numerous examples in the text. If you do not yet have a board, it would be worth reading Chapter 1 before placing an order. Currently the board is manufactured by both CircuitCo and Embest—the boards from either manufacturer are compatible with the designs and operations in this book. You can purchase one of the boards in the United States from online stores such as Adafruit Industries, Digi-Key, Mouser, SparkFun, and Jameco Electronics. They are available internationally from stores such as Farnell, Radionics, Watterott, and Tigal.

A full list of recommended and optional accessories for the Beagle platform is provided in Chapter 1. In addition, each chapter contains a list of the electronics components and modules required if you want to follow along with the text. The book website provides details about where these components can be acquired.

Errata

I have worked really hard to ensure that this book is error free; however, it is always possible that something was overlooked. A full list of errata is available on each chapter's web page at the companion website. If you find any errors in the text or in the source code examples, I would be grateful if you could send the errors using the companion website 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.exploringbeaglebone.com`. It contains videos, source code examples, and links to further reading. Each chapter has its own individual web page. In the unlikely event that this website is unavailable, you can find the code at `www.wiley.com/go/exploringbeaglebone2e`.

All the source code is available through GitHub, which allows you to download the code to your Beagle board with one command. You can also easily view the code online at `tiny.cc/beagle002`. Downloading the source code to your board is as straightforward as typing the following at the Linux shell prompt:

```
debian@ebb:$ git clone https://github.com/derekmolloy/exploringbb.git
```

If you have never used Git before, don't worry—it is explained in detail in Chapter 3. Now, on with the adventures!

Part

I

Beagle Board Basics

In This Part

CHAPTER 1

The Beagle Hardware Platform

In this chapter, you are introduced to the BeagleBone platform hardware and its variant boards. The chapter focuses in particular on the BeagleBone and PocketBeagle boards and the various subsystems and physical inputs/outputs of these boards. In addition, the chapter lists accessories that can be helpful in developing your own Beagle-based projects. By the end of this chapter, you should have an appreciation of the power and complexity of this computing platform. You should also be aware of the first steps to take to protect your boards from physical damage.

Introduction to the Boards

Beagle boards are compact, low-cost, open-source Linux computing platforms that can be used to build complex applications that interface high-level software and low-level electronic circuits. These are ideal platforms for prototyping project and product designs that take advantage of the power and freedom of Linux, combined with direct access to input/output pins and buses, allowing

you to interface with electronics components, modules, and USB devices. The following are some characteristics of the single-board computing (SBC) boards:

- They are powerful, containing a processor that can perform up to 2 billion instructions per second.
- They are widely available at relatively low-cost, as little as $25–$90 depending on the board chosen.
- They support many standard interfaces for electronics devices.
- They use little power, running at between 1W (idle) and 2.3W (peak).
- They are expandable through the use of daughter boards and USB devices.
- They are strongly supported by a huge community of innovators and enthusiasts.
- They are open-hardware and support open-software tools and applications for commercial and noncommercial applications.

The BeagleBone and PocketBeagle boards run the Linux operating system, which means you can use many open-source software libraries and applications directly with them. Open-source software driver availability also enables you to interface devices such as USB cameras, keyboards and Wi-Fi adapters with your project, without having to source proprietary alternatives. Therefore, you have access to comprehensive libraries of code that have been built by a talented open-source community; however, it is important to remember that the code typically comes without any type of warranty or guarantee. If there are problems, then you have to rely on the good nature of the community to resolve them. Of course, you could also fix the problems yourself and make the solutions publicly available.

NOTE **The BeagleBone and PocketBeagle boards are quite different in physical appearance, as displayed in Figure 1-1, but they are similar devices under the hood. To illustrate this, both boards are typically booted with the same Linux image on a micro-SD card. The Linux image will automatically detect and configure the differing hardware during the boot sequence depending on the board it is booting.**

The BeagleBoard.org Foundation is a U.S. nonprofit corporation that aims to provide embedded systems education in open-source hardware and software. Over the last ten years, the Foundation has developed high-quality boards that are renowned in the open-source community for their detailed documentation, for their extensive support, and for providing a strong bridge between idea prototyping and commercial product design.

The platform boards are formed by the integration of a high-performance microprocessor on a printed circuit board (PCB) and an extensive software ecosystem. The physical PCB is not a complete product; rather, it is a