"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

prototype reference design that you can use to build a complete product. It is an open-hardware platform, meaning you can download and use the BeagleBone or PocketBeagle hardware schematics and layouts directly within your own product design. In fact, despite the impressive capability of these boards, they do not fully expose all the features and interfaces of the Texas Instruments Sitara AM335x System on Chip (SoC).

Recent BeagleBone and PocketBeagle boards utilize an Octavo Systems System-in-Package (SiP), which incorporates the Sitara AM335x processor along with DDR memory, power management functionality, and all required passive components into a single ball-grid array (BGA) package, as displayed in Figure 1-1. This SiP design approach vastly simplifies the circuit layout of boards that are based on the AM335x processor platform and has allowed for the small form-factor of the PocketBeagle. You should keep this approach in mind should you decide to commercialize your designs, as it could accelerate the time-to-market of the final product by many months.

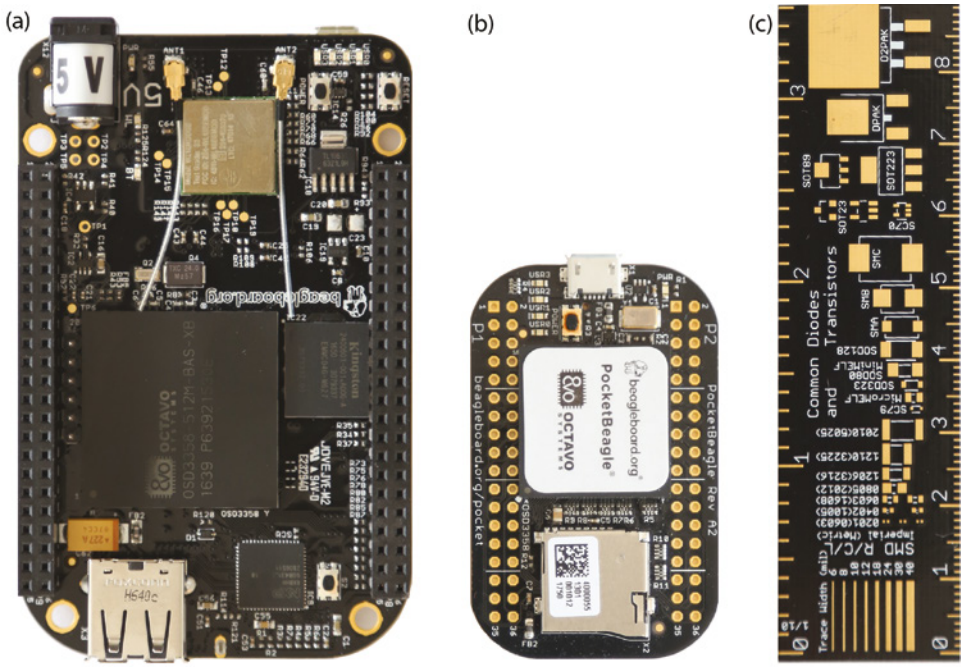

Figure 1-1: (a) BeagleBone Black Wireless, (b) PocketBeagle, and (c) an Adafruit PCB Ruler for relative scale

One impressive feature of the Beagle platform is that board functionality can be extended with daughter boards, called *capes*, which connect to the expansion headers (the two black 2 × 23 connector rows in Figure 1-1(a), or the unpopulated 2 × 18 rows in Figure 1-1(b)). You can design your own capes and attach them

securely to your board using these headers. In addition, many capes are available for purchase that can be used to expand the functionality of your board. Some examples of these are described later in this chapter.

The first five BeagleBone PCBs were designed by Gerald Coley, a co-founder of the BeagleBoard.org Foundation who is now the president of Embedded Product Design (`www.emprodesign.net`). Over the past few years, the boards and several of its capes have been manufactured by CircuitCo (`www.circuitco.com`), Element14 (`www.element14.com`), and its subsidiary Embest (`www.embest-tech.com`). Therefore, when you purchase a Beagle board, you are not purchasing it from the BeagleBoard.org Foundation; rather, the foundation is the focal point for a community of developers and users.

NOTE **CircuitCo has provided a short video of the BeagleBone Black manufacturing process at `tiny.cc/beagle101`— it highlights the complexity of the device and the work that goes into its manufacture.**

Who Should Use the Beagle Platform

Anybody who wants to transform an engineering concept into a real interactive electronics product, project, prototype, or work of art should consider using the Beagle platform. That said, integrating high-level software and low-level electronics is not an easy task. However, the difficulty involved in an implementation depends on the level of sophistication that the project demands.

The BeagleBoard.org community is working hard to ensure their platform is accessible by everyone who is interested in integrating it into their projects, whether they are students, makers, artists, or hobbyists. Tools and software development environments, such as Jason Kridner's BoneScript Node.js library (a co-founder of BeagleBoard.org) and the Cloud9 integrated development environment (IDE), enable users to write and build code directly in a web browser that is capable of controlling electronics hardware. The BoneScript library is introduced in Chapter 2.

For more advanced users, with electronics or computing knowledge, the Beagle platform enables additional development and customization to meet specific project needs. Again, such customization is not trivial: You may be an electronics expert, but high-level software programming and/or the Linux operating system might cause you difficulty. Or, you may be a programming guru but you have never wired an LED! This book aims to cater to all types of users, providing each type of reader with enough Linux, electronics, and software exposure to ensure that you can be productive, regardless of your previous experience level.

When to Use Beagle Boards

The Beagle boards are perfectly placed for the integration of high-level software and low-level electronics in any type of project. Whether you are planning to build an automated home management system, robot, smart display, sensor network, vending machine, or internet-connected work of interactive art, the boards have the processing power to do whatever you can imagine of an embedded device.

The major advantage over more traditional embedded systems, such as the Arduino, PIC, and AVR microcontrollers, is apparent when you leverage the Linux OS for your projects. For example, if you built a home automation system using the BeagleBone and you then decided that you wanted to make certain information available on the internet, you could simply install a web server. You could then use server-side scripting or your favorite programming language to interface with your home automation system to capture and share the information. Alternatively, your project might require secure remote access. In that case, you could install a secure shell (SSH) server simply by using the Linux command `sudo apt install sshd` (these commands are covered in Chapter 2). This could potentially save you weeks of development work. In addition, you have the comfort of knowing that the same software is running securely on millions of machines around the world. Linux also provides you with device driver support for many USB peripherals and adapters, making it possible for you to connect cameras, Wi-Fi adapters, and other low-cost consumer peripherals directly to your platform, without the need for complex and/or expensive software driver development. If you are connecting an embedded system to the internet or to a display (e.g., a touchscreen or monitor), you should consider a Linux SBC such as the Beagle boards before any other option.

When Should You Not Use the Beagle Boards

The Linux OS was not designed for real-time or predictable processing. As a result, there are significant challenges in using this OS for deterministic processing tasks such as sampling a sensor precisely every one-millionth of a second. Therefore, in its default state, the Beagle boards are not an ideal platform for real-time systems applications. Sophisticated real-time versions of Linux are available, but they are currently targeted at experienced Linux developers. However, unlike many other Linux SBCs, the BeagleBone does have an on-board solution that goes some way toward resolving this interfacing problem. Within the AM335x SoC, there are two on-board microcontrollers, called *programmable real-time units* (PRUs), which can be programmed for real-time interfacing applications. This is an advanced topic that is described in Chapter 15.

There are low-cost dedicated solutions available for real-time sampling and control tasks (such as the TI MSP430 or SimpleLink wired and wireless MCUs) that may be more appropriate for real-time interfacing. It is also important to remember that you can interconnect such real-time microcontrollers to the Beagle boards via electrical buses (e.g., I^2C, UART, CAN bus, and Ethernet) and have the Linux SBC act as the central processor for a distributed control system. This is an important concept as part of the Internet of Things (IoT) and is described in detail in Chapters 10, 11, and 12.

The second application type that the Beagle platform will find difficult is that of playing or processing high-definition video. The processing overhead of software decoding and playing encoded video streams is immense and is beyond the capability of the BeagleBone at high-definition video resolutions. The Raspberry Pi (`www.raspberrypi.org`) board has this capability because its Broadcom BCM2835/7 processors were designed for multimedia applications, and it has a hardware implementation of H.264/MPG-4 decoders and encoders. For applications such as running Kodi home media center (`kodi.tv`), you are better off purchasing a Raspberry Pi 3 (Model B+). In addition, you should of course purchase my book, *Exploring Raspberry Pi*, from the same Wiley mini-series!

If your intention is to develop an embedded Linux image processing or computer vision platform, then you should consider the Xilinx Zynq platform (`tiny.cc/beagle102`), as it integrates an ARM-based processor that can run Linux alongside the hardware programmability of an FPGA. This allows the computationally intensive but parallelizable image processing functionality to be offloaded from the Linux kernel to the programmable logic hardware. Boards such as the PYNQ, ZYBO, or Arty Z7 are available, but be aware that they are complex devices.

For interfacing Linux to electronic circuits, it is hard to beat the Beagle boards, as the range of input/outputs, openness of the platform, and quality of documentation available are second to none.

BeagleBone Documentation

This book integrates my experiences in developing for the Beagle platform with supporting background materials on embedded Linux, software development, and general electronics to create an in-depth guide to building with this platform. However, it is simply not possible to cover everything in just one book, so I have avoided restating information that is listed in the key documents and websites described in this section. The first starting point for supporting documentation is always the following:

- **The BeagleBoard.org website:** This provides the main support for this platform, with software guides, community links, and downloads to

support your development. An excellent "Getting Started" guide and blog are available at `www.beagleboard.org`.

A huge amount of documentation is available on the BeagleBone platform, but the most important documents are as follows:

- **Sitara AM335x ARM Cortex-A8 Technical Reference Manual (TRM):**[1] The key component of the Beagle boards are their Texas Instruments SoCs, and this document contains anything you could possibly want to know about the internal workings of the AM335x. It is a complex device, and that is reflected in the length of the AM3358 TRM—5,113 pages! If you need to understand something about the inner workings of the microprocessor or the device configuration on the BeagleBone or PocketBeagle, it is likely that the answer is contained in this document. I refer to tables in the TRM throughout this book so that ideally you will become familiar with the language contained therein. This document and the datasheet for the SoC are available free from `www.ti.com/product/am3358`.
- **The PocketBeagle System Reference Manual (SRM):** This is a live wiki document that describes the PocketBeagle hardware. It is maintained by the BeagleBoard community: `tiny.cc/beagle103`.
- **BeagleBone Black System Reference Manual (SRM):** This is the core document that describes the BeagleBone Black hardware. It is available at `tiny.cc/beagle104`.

Key websites are also available to support your learning on this platform, with combinations of tutorials, discussion forums, sample code libraries, Linux distributions, and project ideas to stimulate your creative side. Here is a selection of important websites:

- **The website for this book:** `www.exploringbeaglebone.com`
- **My personal blog site:** `www.derekmolloy.ie`
- **The eLinux.org Wiki:** `www.elinux.org`
- **The Linux Foundation:** `www.linuxfoundation.org`

Getting started with the Beagle platform software is described in Chapter 2. The remainder of this chapter discusses the physical boards, explaining the functionality that is available, summarizing the SRM, and providing some examples of the types of peripherals and capes that you might like to connect to your board.

[1] At the time of writing, this is in revision P (March 2017) and has the TI document identification SPRUH73P.

The Beagle Hardware

At its heart, the Beagle boards use the Texas Instruments Sitara AM335x Cortex A8 ARM microprocessor. While the BeagleBone and PocketBeagle are the focus of this book, other boards have been developed by BeagleBoard.org, including BeagleBoard, BeagleBoard XM, and the Arduino Tre (BeagleBoard and Arduino combined on a single board). The BeagleBone and PocketBeagle are discussed in detail in the next section, but here are some summary details on the different boards (in historical order):

- **(2008) BeagleBoard ($125):** The original open-hardware ARM-based development board that had HD video support. It has a 720MHz ARM A8 processor but no on-board Ethernet.
- **(2010) BeagleBoard xM ($149):** Similar to BeagleBoard, except with a 1GHz ARM (AM37x) processor, 512MB memory, four USB ports, and Ethernet support. Despite the low cost of the new BeagleBone boards, the BeagleBoard xM is popular for its C64+TMDSP core for digital signal processing (DSP) applications.
- **(2011) BeagleBone ($89):** Smaller footprint than the BeagleBoard. It has a 720MHz processor and 256MB memory, Ethernet support, low-level input/output (e.g., analog to digital converters), but no on-board video support.
- **(2013) BeagleBone Black ($45–$55):** This board enhances the BeagleBone with a 1GHz processor, 512MB of DDR3 memory, Ethernet, eMMC storage, and HDMI video support.
- **(2014-2018) BeagleBone Green, BeagleBone Enhanced, BeagleBone Black Wireless, BeagleBone Blue Wireless, and PocketBeagle ($25–$90):** Variant boards that are substantially based on the BeagleBone Black platform.
- **(2017) BeagleBoard X15 ($270):** High-performance BeagleBoard based on the Sitara AM5728 that has dual 1.5GHz ARM Corex-A15 processors, with integrated C66x DSPs, ARM Cortex-M4 real-time processors, and PRUs (`tiny.cc/beagle105`).

The BeagleBone and PocketBeagle boards are the focus of this book, mainly because of their feature sets and price points in comparison to the other offerings; however, most of the discussion in this book applies generally to all platforms.

BeagleBone Versions

As previously mentioned, there are several versions of the BeagleBone available, as illustrated in Figure 1-2, in particular the older BeagleBone White, or just BeagleBone; the BeagleBone Black (BBB); and the wireless versions. All boards

have a small form factor, fitting neatly inside an Altoids mint tin; in fact, the PocketBeagle fits inside a tiny Altoids Smalls tin.

NOTE **Traditionally, Altoids tins have been upcycled by engineers as a low-cost housing for electronics projects. Given the complexity of the BeagleBone boards, it is impressive that the boards fit inside these tins—it also helps to explain the rounded corners on the BeagleBone boards! Holes can be formed in the case to provide access to the board connectors, but of course it is necessary to electrically insulate the aluminum tin before using it to house your board.**

To achieve such a small form factor, the components are densely placed on the BeagleBone, and a six-layer PCB is used to achieve interconnects. As an example, the AM335x (ZCZ) processors used on the BeagleBone Black platforms have a ball grid array of 324 pins, with a 0.80mm ball pitch.

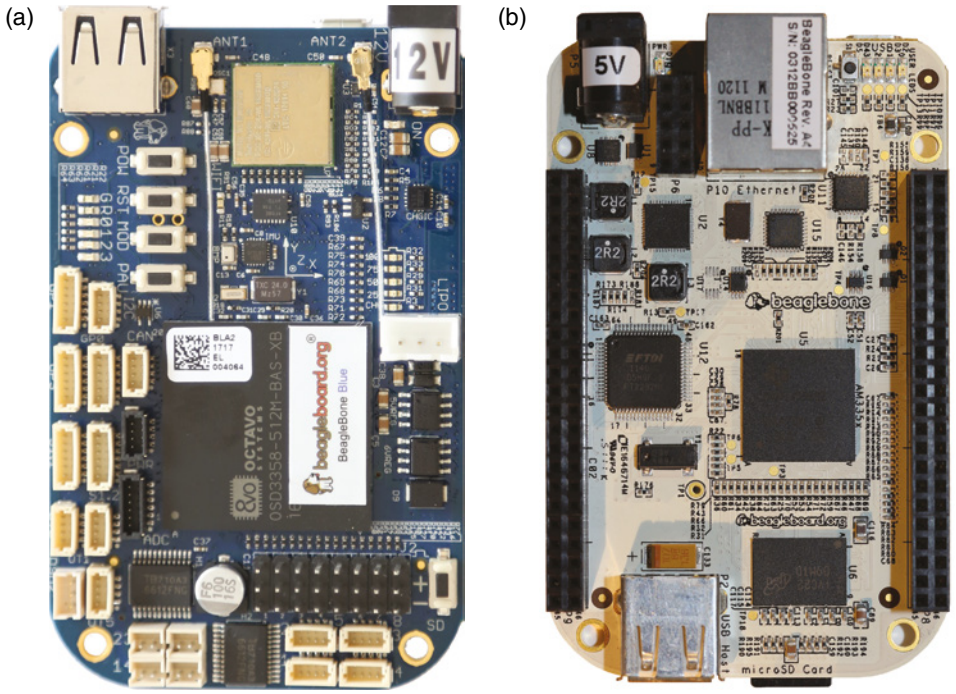

Figure 1-2: (a) The BeagleBone Blue with the Octavo OSD3358 SiP, (b) the original BeagleBone White with the AM335x SoC

Table 1-1 lists the main similarities and differences between the current Beagle boards. The obvious choice factors are the price and network connectivity options.

- The non-wireless BBB has Ethernet connectivity, which can be particularly useful for applications in which the board acts as a network Bridge Router (e.g., for 6LoWPAN applications).

- When wireless networking is required, the BBB Wireless works well in mobile connected embedded applications where video may be required, and the BeagleBone Blue is strong for mobile tasks that interface to motors, for applications such as robotics and automation.
- The PocketBeagle is particularly useful when cost, size, and weight are important considerations for a project. Despite having no on-board wireless connectivity, this can be added and customized for a particular project. For example, you might add one of Wi-Fi, Bluetooth, or 802.15.4-based communications to your project by interfacing modules to the boards USB or UART connections. Interestingly, the bottom side of the PocketBeagle has no components, which means that it can be mounted flush to a carrier printed-circuit board (PCB).

The Beagle Hardware

Figures 1-3, 1-4, and 1-5 detail the core systems of the BBB and PocketBeagle boards. The first set of callouts, 1 to 8, identify and describe the key systems on the BBB. The microprocessor on the BBB is a Texas Instruments Sitara AM335x Cortex A8 ARM Microprocessor.[2] It is a reduced instruction set computing (RISC) processor, so at 1,000MHz the processor executes 2,000 million instructions per second (MIPS). The processor runs at about 1W idle and 2.3W for heavy processing loads.

POCKETBEAGLE USB ON-THE-GO

The PocketBeagle can use USB On-the-Go (OTG) to connect to USB peripherals. USB OTG is often used for devices that switch between the roles of USB client and host. For example, USB OTG connectors are often used to allow cell phones or tablet computers to connect to external USB storage devices. The USB OTG connector allows the PocketBeagle host to connect to a slave device such as a Wi-Fi or Bluetooth adapter. One such adapter is illustrated later in the chapter in Figure 1-8(b).

The next set of callouts, 9 to 19, identifies the various connectors on the BBB, their physical characteristics, and their function. For connector 18, the JTAG connector, there are 20 pre-tinned pads. You need to purchase a connector (such as Samtec FTR-110-03-G-D-06) for this and carefully solder it to the board.

Table 1-2 details the various inputs and outputs that are available on the expansion headers. There are 92 pins on these headers (2 × 46) on the

[2]Early BBB boards used an XAM3359AZCZ100 processor, but more recent boards (from Rev C) use the AM3358BZCZ100 (even within the OSD3358 SiP). The feature set that is exposed to the BBB platform is the same, so the notation AM335x is used.

Table 1-1: A High-Level Comparison of Recent Beagle Boards

MODEL	BEAGLEBONE BLACK	BEAGLEBONE BLACK WIRELESS	POCKETBEAGLE	BEAGLEBONE BLUE	BEAGLEBOARD X15
Approximate price	$55	$70	$25	$90	$270
Processor	1GHz AM335x with two 32-bit PRUs	1GHz AM335x with two 32-bit PRUs	1GHz AM335x with two 32-bit PRUs	1GHz AM335x with two 32-bit PRUs	Two 1.5GHz ARM A15s, C66 DSP Cores, two ARM M4s and four PRUs
Memory	512MB DDR3	512MB DDR3	512MB DDR3	512MB DDR3	2GB DDR3
Storage	On-board 4GB eMMC and micro-SD card slot	On-board 4GB eMMC and micro-SD card slot	micro-SD card slot	On-board 4GB eMMC and micro-SD card slot	On-board 4GB eMMC and micro-SD card slot
Video	On-board HDMI	On-board HDMI	None	None	On-board HDMI (full)
Debugging	JTAG pads	JTAG pads	JTAG pads	JTAG pads	20-pin JTAG header
Interfacing	Two 46-pin female GPIO headers	Two 46-pin female GPIO headers	Two 36-pin unpopulated headers	JST interfaces and 24-pin male header bank	Four 60-pin headers
Wired Ethernet	10/100 Ethernet	None	None	None	Two Gigabit Ethernet
Wireless Network	None; available through USB Wi-Fi adapters	802.11bgn and Bluetooth 4.1 with BLE	None	802.11bgn and Bluetooth 4.1 with BLE	None

MODEL	BEAGLEBONE BLACK	BEAGLEBONE BLACK WIRELESS	POCKETBEAGLE	BEAGLEBONE BLUE	BEAGLEBOARD X15
Supply	5V USB or DC jack	5V USB or DC jack	5V USB and via header pins	12V DC jack	12V DC jack (5A)
Application	General-purpose prototyping with video and Ethernet	General-purpose prototyping with video and Wi-Fi/ Bluetooth	Self-contained Linux IoT or interfacing applications	Mobile robotics applications; includes an IMU, barometer, LiPo support, and H-bridges	High-end DSP and real-time interfacing applications, including eSATA and USB3

	Function	BeagleBone	PocketBeagle	Details
1	Processor	AM335x	OSD3358-SM	A powerful Texas Instruments Sitara ARM-A8 processor that is standalone or enclosed in an Octavo Systems System-In-Package (SiP) such as the OSD3358-SM.
		2 x PRUs	2 x PRUs	Programmable Real-time Units (PRUs). Microcontrollers that allow for real-time interfacing.
		Graphics Engine	Graphics Engine	Processor has a 3D graphics engine (Imagination Technologies PowerVR SGX530) that is capable of rendering 20 million polygons per second.
2	Graphics	HDMI Framer	None	The framer converts the LCD interface available on the AM335x processor into a HDMI signal (no HDCP).
3	Memory	512 MB DDR3	512 MB DDR3	The amount of system memory affects performance and the type of applications that can be run.
4	On-board Storage	eMMC (MMC1)	None	A 4GB on-board embedded multi-media card (eMMC), which is an SD card on a chip. The BeagleBone boards can boot without an SD card.
5	Power Management	TPS65217C	TPS65217C	Power management IC (PMIC). Sophisticated power management IC that has voltage regulators and is controlled by the main processor. Supports LiPo batteries.
6	Ethernet Processor	Ethernet PHY (10/100)	None	BBB can be connected to a network using a LAN8710A physical interface to an RJ45 connector. Not available on the wireless versions.
7	LEDs	6 x LEDs	4 x LEDs	Power LED and four user LEDs. The wired BeagleBone has LEDs on the RJ45 Ethernet socket (yellow = 100M link up, green = traffic).
8	Buttons	3 x Buttons	1 x Button	Power button. The BeagleBone boards have a reset button and a boot switch button for choosing to boot from the eMMC or the SD card.
	Connectors			
9	Video Out	micro-HDMI (HDMI-D)	None	For connecting to monitors and televisions. Supports resolutions up to 1280x1024 at 60Hz. It can run 1920x1080 but at 24Hz. Has HDMI CEC support.
		Audio out (HDMI-D)	None	HDMI can be broken out to a 3.5mm audio jack using accessories.
10	Network	Ethernet (RJ45)	None	10/100 Ethernet via a RJ45 connector. On board Wi-Fi and Bluetooth is available on the BeagleBone Black/Blue Wireless.
11	DC Power	5V DC supply (5.5mm) 12V DC supply on Blue	None	For connecting 5V DC mains PSUs to the board. PocketBeagle is usually powered via the USB connector but can be powered by battery or the expansion header.
12	SD Card	micro-SD card slot	micro-SD card slot	3.3V micro-SD card slot. The BeagleBone can be booted from this slot, flashed from this slot, or it can be used for additional storage when the board is booted from the eMMC.
13	Serial Debug	6 Pin Connector (0.1")	None	(UART0) Used with a serial TTL3V3 cable to connect to the serial console. This functionality is also available via USB on both boards.
14	USB	1 x USB 2.0 Client (mini-USB or micro-USB)	1 x USB OTG	(USB0) Connects to your desktop and can power the board.
15		1 x USB 2.0 Host (USB-A)		(USB1) You can connect USB peripherals (e.g., Wi-Fi) to the board with this connector.
16	Expansion Headers	Two 2x23 pin 0.1" female headers	Two 2x18 pin 0.1" unpopulated headers	These headers are multiplexed to provided access to a range of input/output features. Not all functionality is available at the same time. Used to connect capes.
17				
18	Other Debug	JTAG	JTAG	Unpopulated JTAG header that can be used to debug a board when used with additional hardware and software.
19	Other Power	Battery connectors	via headers	It is possible to solder pins to these points on the BeagleBone or to the headers on the PocketBeagle to power the board. Read the SRM carefully!

Figure 1-3: Table of BeagleBone and PocketBeagle subsystems and connectors

BeagleBone and 72 pins on the PocketBeagle (2 × 36); however, not all are available for general-purpose input/outputs (GPIOs). Several of the connections have a fixed configuration:

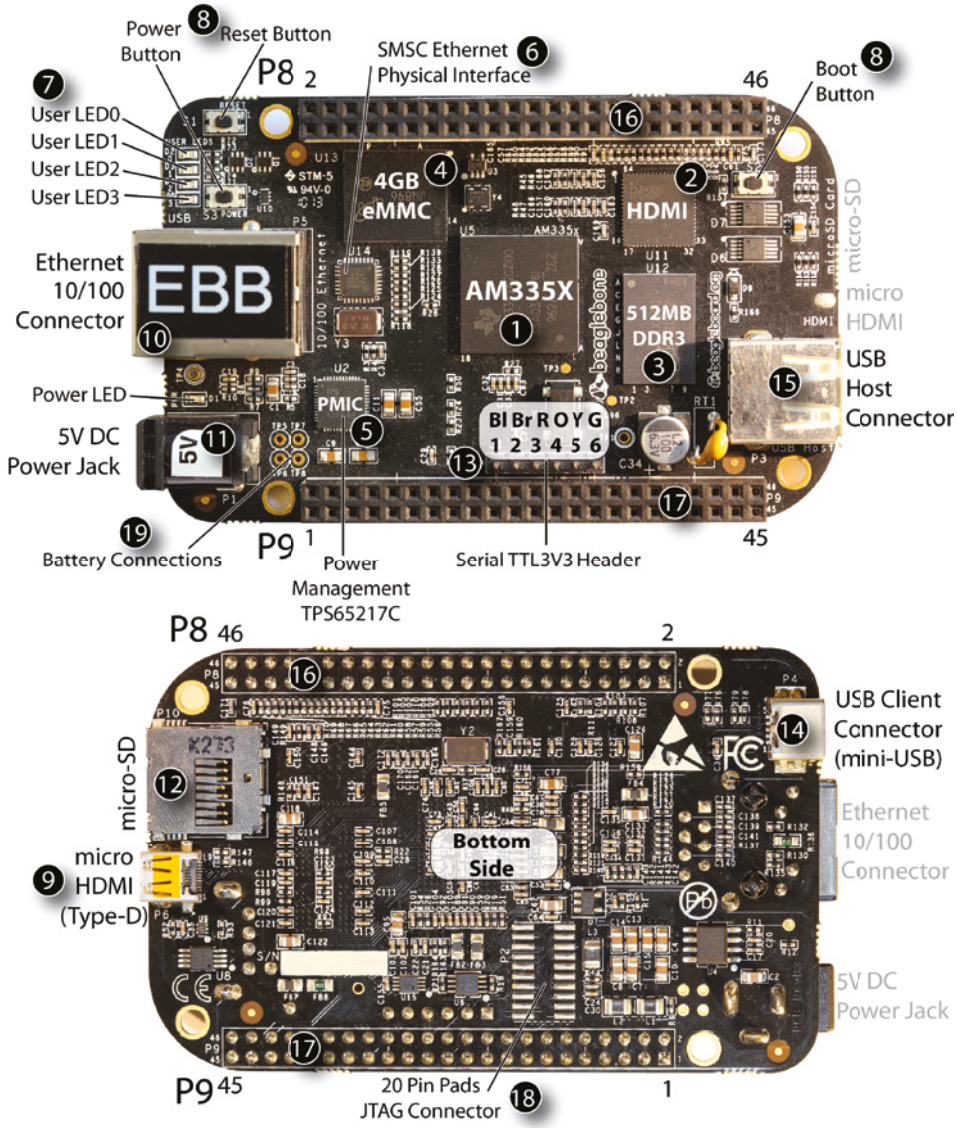

Figure 1-4: The BeagleBone Black (BBB) top and bottom views

- Several pins are connected to ground.
- Pins are required to support the analog inputs (e.g., a 1.8V reference voltage).
- Pins are allocated to 3.3V and 5V voltage supplies.

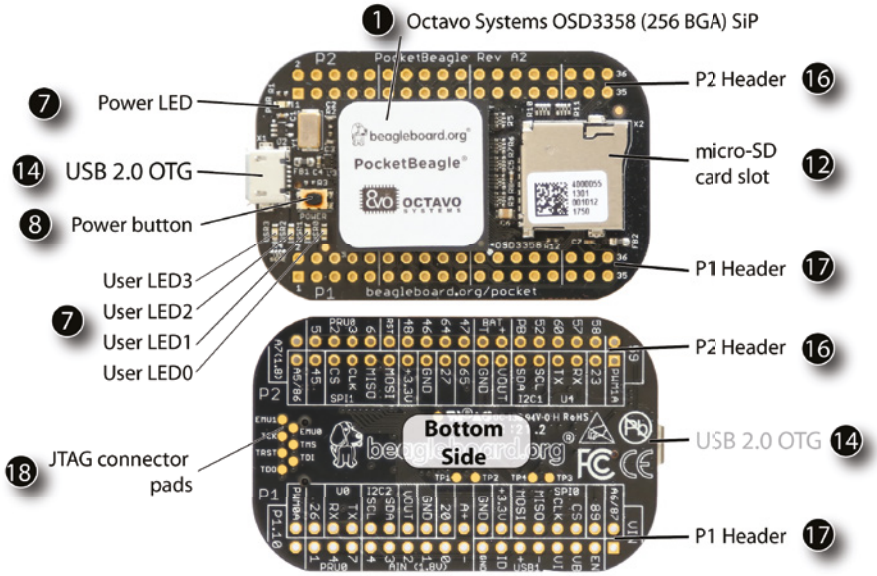

Figure 1-5: The PocketBeagle top and bottom views

The remaining connectors are available to be multiplexed to many different functions, several of which are listed in Table 1-2. The function of each of these input/output types is discussed in Chapter 6 and Chapter 8.

Table 1-2: Functionality Available on the Expansion Headers

EXPANSION HEADERS	BEAGLEBONE P8 AND P9	POCKETBEAGLE P1 AND P2	NOTE: NOT ALL FUNCTIONALITY LISTED HERE IS AVAILABLE SIMULTANEOUSLY. PLEASE SEE CHAPTER 6 FOR DETAILS.
GPIO	65	44	All general-purpose input/outputs are 3.3V tolerant and can only source or sink relatively small currents.
PWM	8	4	Pulse width modulated (PWM) outputs allow you to send a type of variable analog output (0V to 3.3V). PWM can be used to control servo motors or LEDs.

Continues

Table 1-2: (*continued***)**

EXPANSION HEADERS	BEAGLEBONE P8 AND P9	POCKETBEAGLE P1 AND P2	NOTE: NOT ALL FUNCTIONALITY LISTED HERE IS AVAILABLE SIMULTANEOUSLY. PLEASE SEE CHAPTER 6 FOR DETAILS.
Analog Input	7	8	12-bit 1.8V analog inputs that are always available on the headers (not multiplexed). These can be used for reading sensor values, but be careful as they are only 1.8V tolerant. Note that six are 1.8V tolerant on the PocketBeagle, but two are 3.3V tolerant.
Power Supply	5V, 3.3V	5V, 3.3V	5V and 3.3V supplies are available. The ADC circuitry also provides a 1.8V reference voltage, but this should not be used as a general supply.
Timers	4	4	Can be used to generate external clocks for interfacing to devices.
I2C	2	2	I2C is a digital bus that allows you to connect several modules to each of these two-wire buses at the same time. There are two public buses and one additional private bus.
UART	4	3	Used for serial communication between two devices. UART0 is the Serial Debug connector on the BeagleBone.
CAN	2	2	CAN Bus is used for Controller Area Networks (CAN), often on industrial processes or vehicles to communicate between various networked systems. There is a CAN cape available for the BeagleBone.
SPI	2	2	Serial Peripheral Interface (SPI) provides a synchronous serial data link over short distances. It uses a master/slave configuration and requires four wires for communication.

EXPANSION HEADERS	BEAGLEBONE P8 AND P9	POCKETBEAGLE P1 AND P2	NOTE: NOT ALL FUNCTIONALITY LISTED HERE IS AVAILABLE SIMULTANEOUSLY. PLEASE SEE CHAPTER 6 FOR DETAILS.
GPMC	1	1	General-purpose memory controller (GPMC) is used to connect to external memory devices like FPGAs or ASICs. This fast bus conflicts with the eMMC on the BeagleBone.
MMC	2	2	Interface buses that are used to connect the micro-SD card and the eMMC to the processor.
LCD	1	1	Useful for LCD screens (e.g., LCD capes). This interface conflicts with the HDMI Framer on the BeagleBone.
McASP	2	2	General-purpose audio serial port—multichannel audio serial port (McASP), connected to the HDMI framer on the BeagleBone.

Beagle Accessories

Most boards (except the PocketBeagle) are packaged with a USB 2.0 cable (either a mini-USB plug or micro-USB-to-USB-A plug), which is used to connect the BeagleBone (via the USB client connector) to a desktop computer. The boards do not come with a micro-SD card, and you will need one for the PocketBeagle in particular. The BeagleBone boots out of the box without the need for a card, as the Linux installation is already present on the board's eMMC.

The boards can be connected to a display using an HDMI cable (except on the BeagleBone Blue and the PocketBeagle), but most of the examples in this book assume the boards are used in headless mode—that is, not connected directly to a display; rather, the board is used as a networked device that interfaces to electronic circuits, USB modules, and wireless sensors.

Highly Recommended Accessories

The following accessories are recommended for purchase along with your board. If you are planning to carry out development work, then you should probably have most of them.

Headers for the PocketBeagle

If you plan on interfacing the PocketBeagle to electronic circuits, then the first thing you need to do is add male or female header pins to the P1/P2 expansion headers, as these are unpopulated. Depending on your application you can use two 2 × 18 female (2.54mm/0.1" spacing) headers or break-away male strip headers (2.54mm/0.1" spacing) that can be cut to 2 × 18 size, as illustrated in Figure 1-6(a). These can be mounted on the top side or bottom side of the PocketBeagle; however, the headers do not fit well against the OSD3358 module when mounted on the top side. The bottom side also has a useful pin identifier key, as illustrated in Figure 1-5. Surprisingly, the dimensions of the OSD3358 module also means that the component side of the board sits flat on a work surface. Therefore, my preference is to mount the pins on the bottom side, as illustrated in Figure 1-6(b).

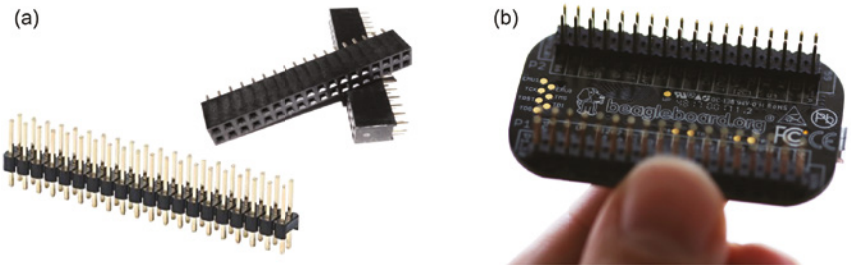

Figure 1-6: (a) Female and male header connectors; (b) the PocketBeagle with male headers soldered to the bottom of the board

In my experience, female headers are safer to use than male headers, as it is easy to accidentally short across two male header pins when these are stacked in a 2 × 18 array. And, you can connect to female headers simply with strands of wire.

NOTE **When soldering the male or female headers to the PocketBeagle, begin with the pins that are closest to the OSD3358 module. While slightly more expensive, four 1 × 18 headers make the soldering process more straightforward. Use a breadboard to help keep the header pins aligned and vertical.**

Micro-SD Card (for Booting or Flashing eMMCs)

A micro-SD card enables you to boot any board or write a new Linux image to a board that has an eMMC. If you have a BeagleBone, then the card can be important if you accidently damage the Linux file system during your experimentation, as the micro-SD card will enable you to restore your system using a "flasher" configuration. Ideally, you should have two dedicated SD cards, one for a boot image and one for a flasher configuration. Be careful not to mix them up!

Purchase a genuine, branded micro-SD card of at least 4GB capacity. Ideally you should use an 8–64GB micro-SD card with wear-levelling functionality. Larger micro-SD cards also work, but these may be cost prohibitive—alternative approaches to increasing the storage capacity include the use of USB storage devices.

You may also require a micro-SD-to-SD adapter so that it can be used in your computer's card reader. Many micro-SD cards are bundled with the adapter, which is a cheaper option than purchasing them separately. The micro-SD card should be of Class 10 or greater, as the faster read/write speed will save you time in writing images in particular. A blank micro-SD card can also be used for additional file system storage (discussed in Chapter 3), so the greater the card capacity, the better.

ADDING A SECOND USB PORT TO THE POCKETBEAGLE

The PocketBeagle has a single USB port, which makes it difficult to power the board, connect to it over serial USB, and configure it to use a Wi-Fi adapter simultaneously. You can easily add a second USB port to the PocketBeagle for as little as $0.20 using the adapter boards that are illustrated in Figure 1-7(a), which can be wired to the PocketBeagle as illustrated in Figure 1-7(b) and Figure 1-7(c). The PocketBeagle pins used in these figures are configured to act as a USB port by default, but you may need to reboot the board for the port to be enabled. You can check that your USB device is detected using the `lsusb` command. For example, when a USB memory key is plugged into a USB A module, it will result in an output such as the following:

```
debian@ebb:~$ lsusb
Bus 002 Device 002: ID 13fe:4100 Kingston Tech Company Inc. Flash drive
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

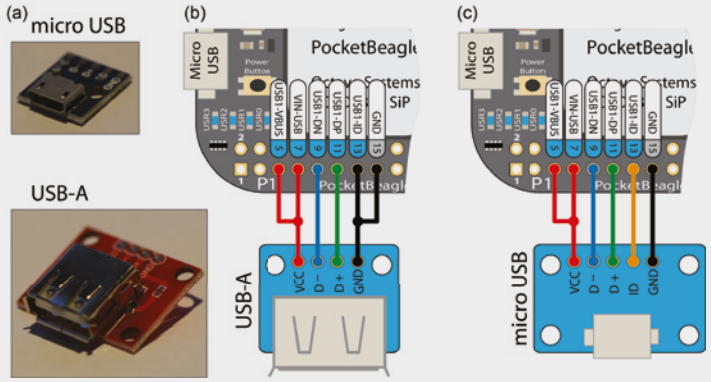

Figure 1-7: Adding low-cost USB socket adapters to the PocketBeagle to add a second USB device, (a) a micro USB and USB-A module; (b) a USB-A wiring configuration; (c) a micro USB wiring configuration

External 5V Power Supply (for Peripherals)

You can power the boards directly using the USB connection from your desktop/laptop computer to the USB client connector on the boards. For getting started, that is perfectly fine; however, once you begin to connect accessories such as Wi-Fi adapters, USB cameras, or on-board displays, it is possible that the power available over USB will not be sufficient for your configuration.

You can purchase a 5V DC regulated switching power supply that plugs directly into a mains socket. It should have a minimum DC output current of 1A. However, you should aim for a 2A current supply (2A × 5V=10W), if possible.

The BeagleBone Blue requires a 12V DC power supply with an output current of 3A. The 5V barrel connector (5.5mm diameter) from the supplies should be center positive in all cases.

Ethernet Cable (for Wired BBB Network Connection)

The Beagle boards can use a special networking mode, called internet-over-USB, to create a virtual network between the board and your desktop; however, if you are connecting the BBB to your home network, then you can use a Cat5 network patch cable to connect your BBB to the network using its RJ45 10/100 Ethernet connector. If you are planning to use more than one BBB simultaneously and network stability is important to your application, you could invest in a low-cost multi-port switch, which can be placed close to your desktop computer.

HDMI Cable (for Connection to Monitors/Televisions)

Several Beagle boards have a HDMI framer and connector that can be easily connected to a monitor or television that has an HDMI or DVI connector. The BBB has a micro-HDMI socket (HDMI-D), so be careful to match that to your monitor/television type (usually HDMI-A or DVI-D). The cable you are likely to need is a "HDMI-Micro-D Plug to HDMI-A Male Plug." A 1.8m (6ft.) cable should cost no more than $10. Be careful with your purchase—an HDMI-C (mini-HDMI) connector will *not* fit the BBB.

Alternatively, you can purchase a low-cost ($3) micro-HDMI (HDMI-D) plug to regular HDMI (HDMI-A) socket adapters or micro-HDMI (HDMI-D) plug to DVI-D socket adapter cables. These enable you to use regular-size HDMI-A or to connect to DVI-D devices, respectively (see Figure 1-8(a)).

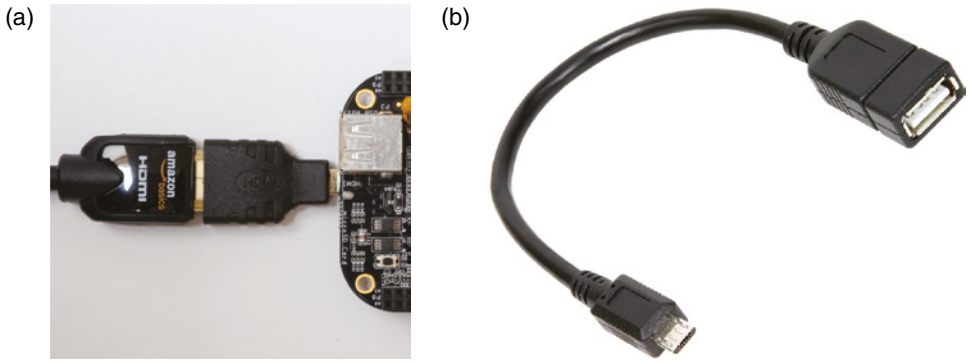

Figure 1-8: (a) BBB connected to micro-HDMI-to-HDMI adapter and then to a low-cost HDMI-A-to-DVI-D cable; (b) a USB OTG connector with the PocketBeagle

USB to Serial UART TTL 3.3 (for Finding Problems)

The USB-to-serial UART TTL serial cable is one accessory that is really useful when there are problems with the Linux installation on your board. It can provide you with a console interface to the board, without the need for connection to an external display and keyboard.

Please ensure that you purchase the *3.3V level* version and ideally purchase a version with six one-way 0.1" female headers pre-attached so that it can be used with the BeagleBone or the PocketBeagle. This cable contains a chipset and requires that you install drivers on your desktop computer, creating a new COM port. The FTDI TTL-232R-3V3 cable, as displayed in Figure 1-9(a), works well and provides a stable connection (about $20). See `tiny.cc/beagle106` for the datasheet and the "VCP" link to the software drivers for this adapter cable. Cheaper alternatives are available ($0.60), such as CH340G chipset devices as illustrated in Figure 1-9(c), but be careful that you set the voltage selector to be 3.3V.

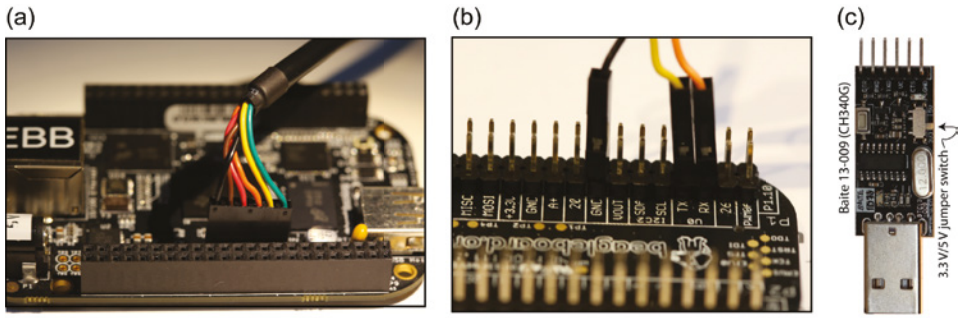

Figure 1-9: (a) The USB-to-TTL 3.3V serial cable; (b) its connection to the BBB (connection colors are black, brown, red, orange, yellow, and green); and (c) a low-cost USB-to-TTL connector

The cable connects to a serial UART on the BeagleBone or PocketBeagle boards. With your Beagle board powered using a regular USB 5V supply, connect the cable as described in Table 1-3, and as illustrated in Figure 1-9(b).

Table 1-3: Serial Debug Connections from the FTDI Cable to the BeagleBone or the PocketBeagle Board

FUNCTION	FTDI CABLE	BEAGLEBONE	POCKETBEAGLE
Ground	Black GND wire	Pin 1 J1 Header GND	GND P1 Header
TX→RX	Orange transmit wire	Pin 4 J1 Header RXD	U0 RX P1 Header
RX←TX	Yellow receive wire	Pin 5 J1 Header TXD	U0 TX P1 Header

Please note that the expansion headers are described in detail in Chapter 6. If you are planning to flash your own images to the BeagleBone or if you have a board that is not booting, I recommend you purchase one of these cables. The use of this cable is discussed in Chapter 2 and Chapter 3.

WARNING **The Beagle boards are 3.3V tolerant (and 1.8V in the case of some pins) but also have a 5V supply available on header pins. The easiest way to destroy your board is to accidentally connect these pins to a circuit that requires 3.3V logic levels or to accidentally short these pins with other pins on the GPIO header. Please be especially careful when working with the 5V pins.**

Optional Accessories

The following sections describe optional accessories that you may need, depending on the applications that you are developing (see Figure 1-10).

Figure 1-10: (a) USB Wi-Fi adapters; (b) the Logitech C920 camera; (c) a Velleman USB hub (bus powered)

USB Hub (to Connect Several USB Devices to a USB Host)

If you are planning to connect more than one USB device to your board at the same time, then you will need a USB hub. USB hubs are either bus powered or externally powered. Externally powered hubs are more expensive; however, if you are powering several power-hungry adapters (Wi-Fi in particular), then you may need a powered hub. Ensure that you plug the USB hub into the Beagle board host connector *before* powering on the Beagle board. I have tried different brands of USB hub and these have all worked without difficulty.

Micro-HDMI to VGA Adapters (for VGA Video and Sound)

Several low-cost micro-HDMI-to-VGA adapters are for sale (e.g., on Amazon or eBay) for converting the HDMI output to a VGA output. As well as providing for VGA video output, many of these connectors provide a separate 3.5mm audio line out, which can be used if you want to play audio using your BeagleBone, without requiring a television, high-end amplifier, or monitor. There are also USB audio adapters available that can provide high-quality playback and recording functionality. These adapters and their usage are described in Chapter 14.

Wi-Fi Adapters (for Wireless Networking)

The BeagleBone Wireless boards have on-board Wi-Fi, but for the BBB and PocketBeagle you can use a USB Wi-Fi adapter. Many different adapters are available, such as those in Figure 1-10(a); however, not all adapters will work under Linux. The Linux distribution and the chipset inside the adapter will determine the likelihood of success. You can find a list of adapters that are confirmed as working at `tiny.cc/beagle107`. However, please be aware that manufacturers can change chipsets within the same product and that buying an adapter from the list does not guarantee that it will work. You are more likely to succeed if you can confirm the chipset in the adapter you are planning to purchase, and evaluate that against the list. Wi-Fi configuration and applications are discussed in detail in Chapter 12, which tests a range of different low-cost adapters that are widely available.

USB Webcam (for Capturing Images and Streaming Video)

Attaching a USB webcam can be a low-cost method of integrating image and video capture into your projects. In addition, utilizing Linux libraries such as Video 4 Linux and Open Source Computer Vision (OpenCV) enables you to build "seeing" applications.

In Chapter 14, different webcams are examined, but the text focuses on the use of the Logitech C920 webcam in particular for video streaming

applications (see Figure 1-10(b)). This is a relatively pricey webcam (at about $70), but it is capable of streaming full HD video directly when using the Beagle boards, as it has H.264/MPG-4 hardware encoding built into the camera. This greatly reduces the workload for the board, allowing the processor to be available for other tasks. As with Wi-Fi adapters, it would be useful to confirm that a webcam works under Linux before you purchase it for that specific purpose. We'll look at several camera types in Chapter 14.

USB Keyboard and Mouse (for General-Purpose Computing)

It is possible to connect a USB keyboard and mouse separately to a USB hub or to use a 2.4GHz wireless keyboard and mouse combination. Small wireless handheld combinations are available, such as the iPazzPort Wireless Mini, Rii i8, and eSynic mini, all of which include a handheld keyboard with integrated touchpad. A USB Bluetooth adapter is also useful for connecting peripherals to the board.

Capes

Capes are daughter boards that can be attached to the P8/P9 expansion headers on the BeagleBone boards or the P1/P2 expansion headers on the PocketBeagle. These are called *capes* (as in Superman's cape) because of the shape of the boards as these wrap around the RJ45 Ethernet connector on the BBB. You can connect up to four capes at any one time when the capes are compatible with each other.

Some capes use a significant number of pins. For example, you will look at the LCD4 cape in Chapter 13. The LCD4 cape uses the P8 header pins 27 through 46 and some of the analog inputs for its buttons and resistive touch interface. If you are using the eMMC for booting the BBB, then few pins remain for GPIO use. In addition, the LCD cape does not carry forward the pin headers. Figure 1-11 shows two views of this cape when connected to the BBB, running the standard Debian Linux distribution. Similar issues arise with other capes.

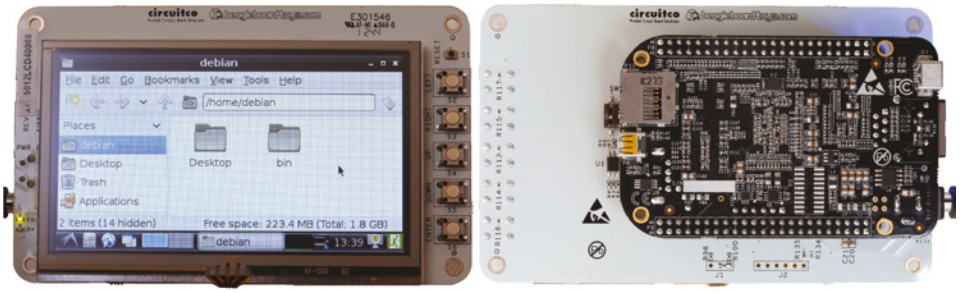

Figure 1-11: The LCD4 cape (top and bottom view)

More than 100 capes are currently available for the BeagleBone and PocketBeagle boards; you can find a full list at `www.beagleboard.org/cape`. Here is a selection of some example capes that you might find useful in your projects:

- The LCD capes are available in different sizes: 7" (800×480), 4" (480×272), and 3" (320×240), with the 4" version captured in Figure 1-11. These have resistive touch screens, meaning you use a stylus (or fingernail) to interact with the screens. This is different than the capacitive touch screens on recent phones/tablets. The Manga Screen 2 is a HDMI-compatible multi-touch LCD screen alternative that is available in a 4.8" (720p) or 5.9" (1080p) version. See `tiny.cc/beagle108`.
- The Adafruit Proto cape, as illustrated in Figure 1-12(a), is a low-cost (~$10) bare cape, which you can use to transfer your breadboard design to a more solid platform. Several other breadboard and prototyping capes are available for the BeagleBone and PocketBeagle boards. One particularly notable cape is the BaconBits cape (`tiny.cc/beagle110`), which adds seven-segment displays, an accelerometer, LEDs, POTs, buttons, and a USB-to-serial bridge to the PocketBeagle.
- The Replicape ($179) is an impressive open-source 3D printer cape that has five stepper motor drivers, including micro-stepping support. See `www.thing-printer.com` for more information.

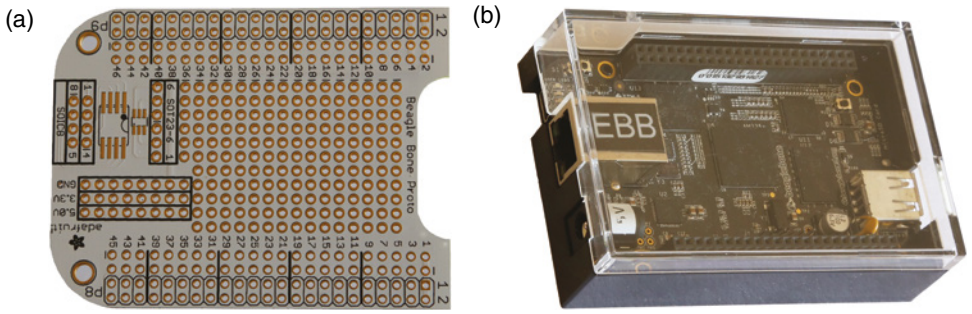

Figure 1-12: (a) The Proto cape; (b) a suitable enclosure case

You have to be careful about compatibility when interconnecting capes. There is a compatibility table covering the more common capes at `tiny.cc/beagle109`. The preceding list is just a small selection. Many more capes are available, and it is likely that additional capes will be developed over time.

How to Destroy Your Board!

The Beagle boards are complex and delicate devices that are easily damaged if you do not show due care. If you are moving up from boards like the Arduino to the Beagle platform, then you must be especially careful when connecting

circuits that you built for that platform. Unlike the Arduino Uno, the microprocessor on the boards cannot be replaced—if you damage the SoC or SiP, you will need to buy a new board!

Here are some things that you should *never* do:

- Do not shut the board down by pulling out the power jack/USB power. Correctly shut the board down by using a software shutdown (e.g., by pressing the power button once) or by holding the power button for about eight seconds for a "hard" power down. This enables the power management IC (PMIC) to shut down the board correctly. If you need to remove power by disconnecting the power supply, hold the reset button while doing so to lower system power usage.
- Do not place a powered board on metal surfaces (e.g., aluminum-finish computers) or on worktops with stray/cut-off wire segments, resistors, etc. If you short the pins (or the solder points) on the expansion headers, you can easily destroy your board. You can buy a case from suppliers such as Adafruit (see Figure 1-12(b)). Alternatively, you can attach small rubber feet to the board.
- Do not connect circuits that source/sink other than very low currents from/to the expansion headers. The maximum current that you can source from many of these header pins is 4-6mA and the maximum current you can sink is 8mA. The power rail and ground pins can source and sink larger currents. The Arduino allows currents of 40mA on each input/output. This issue is covered in Chapter 4 and Chapter 6.
- The GPIO pins are 3.3V tolerant (most of the ADCs are 1.8V tolerant). Do not connect a circuit that is powered at 5V or you will destroy the board. This is discussed in Chapter 4, Chapter 6, and Chapter 8.
- Do not connect circuits that apply power to the expansion header while the board is not powered on. Make sure that all self-powered interfacing circuits are gated by the 3.3V supply line or through the use of optocouplers. This is covered in Chapter 6.

Here are two steps that you should *always* follow:

- Carefully check the pin numbers you are using. There is a large number of pins in each header, and it is easy to plug into header connector 17 instead of 15. For connections in the middle of the headers, I always count twice—up from the left and down from the right.
- Read the SRM for your board in detail before connecting complex circuits of your own design.

If your board is dead and it *is* your fault, then I'm afraid that after you perform all the checks at `www.beagleboard.org/support`, you will have to purchase a new board. If it *is not* your fault, then see the BBB/PocketBeagle SRM manual and `www.beagleboard.org/support` website to return a defective board for repair by requesting a return merchandise authorization (RMA) number.

Summary

After completing this chapter, you should be able to do the following:

- Describe the capability of the Beagle boards and their suitability for different project types
- Source the important documents that will assist you in working with the Beagle platform
- Describe the major hardware systems and subsystems on the different boards
- Identify important peripherals and accessories that you can buy to enhance the capability of your board
- Have an appreciation of the power and complexity of the Beagle boards as physical computing devices
- Be aware of the first steps to take in protecting your boards from physical damage

Support

The key sources of additional support documentation were listed earlier in this chapter. If you are having difficulty with the Beagle platform and the issues are not described in the documentation, then you should use these two resources:

- The BeagleBoard Google Group, which is available at `groups.google.com/d/forum/beagleboard`. Please read the frequently asked questions (FAQs) and search the current questions before posting a new question.
- There is a live chat available at `www.beagleboard.org/chat` or directly on the Beagle IRC channel (by joining `#beagle` on `irc.freenode.net`) using a free IRC client such as X-Chat for Linux, HexChat for Windows, or Colloquy for macOS.

Please remember that the people in this group and IRC channel are community members who volunteer their time to respond to questions.

CHAPTER

2

Beagle Software

In this chapter, you are introduced to the Linux operating system and software tools that can be used with the Beagle boards. This chapter aims to ensure that you can connect to your board and control it. By the end of this chapter, you should be able to "blink" a system LED having followed a step-by-step guide that demonstrates how you can use Linux shell commands in a Linux terminal window. In this chapter, you are also introduced to a library of software functions, called BoneScript, which can be used with Node.js and the Cloud9 integrated development environment to build code that flashes the same system LED.

EQUIPMENT REQUIRED FOR THIS CHAPTER:

- Any Beagle board
- USB cable (typically USB A male to mini- or micro-USB A male)
- Micro-SD card (4GB or greater; Class 10+)
- Network infrastructure and cabling (optional)

Further details on this chapter are available here:
`www.exploringbeaglebone.com/chapter2/`.

Linux on the Beagle Boards

A *Linux distribution* is a publicly available version of Linux that is packaged with a set of software programs and tools. There are many different Linux distributions, which are typically focused on different applications. For example, high-end server owners might install Red Hat Enterprise, CentOS, Debian, or OpenSUSE; desktop users might install Ubuntu, Debian, Fedora, or Linux Mint. The list is endless, but at the core of all distributions is a common Linux kernel, which was conceived and created by Linus Torvalds in 1991.

In deciding on a Linux distribution to use for your embedded system platform, it is sensible to choose one with these attributes:

- The distribution is stable and well supported.
- There is a good package manager.
- The distribution is lean and suited to a low storage footprint for embedded devices.
- There is good community support for your particular device.
- There is device driver support for any peripherals you want to attach.

Linux Distributions for Beagle Boards

There are many different distributions of Linux for embedded system platforms, including expensive proprietary versions for real-time programming. At their heart, they all use the mainline Linux kernel, but each distribution contains different tools and configurations that result in quite different user experiences. The main open-source distributions used by the community on the Beagle boards include Debian, Ångström, Ubuntu, and Arch Linux.

Debian (a contraction of "Debbie and Ian") is a community-driven Linux distribution that has an emphasis on open-source development. There is no commercial organization involved in the development of Debian; in fact, there is a formal social contract (`tiny.cc/beagle201`) that states that Debian will remain entirely free (as in software freedom). The Debian distribution is used for many of the practical steps in this book; in fact, it is recommended as the distribution of choice for the Beagle boards, and it is currently distributed with new BeagleBone boards. In addition, Debian is used throughout this book as the distribution for the Linux desktop computer, as it provides excellent support for cross-platform development through Debian Cross-Toolchains (see `wiki.debian.org/CrossToolchains`). Currently, there are different versions of Debian available for download from the BeagleBoard.org website.

- *Debian Stretch LXQt*, which has the Lightweight Qt Desktop (LXQt) environment installed. This version should be used if you are attaching the board to a monitor or an LCD panel.

- *Debian Stretch IoT*, which is a headless image that has a much smaller footprint on the micro-SD card and (of greater consequence) the eMMC of a BeagleBone.

Ångström is a stable and lean Linux distribution that is widely used on embedded systems. The team of developers behind Ångström is experienced in customizing Linux distributions for embedded devices such as set-top boxes, mobile devices, and networking devices. Impressively, Ångström can scale down to devices with only megabytes of flash storage. Ångström makes extensive use of *BusyBox*, a multicall binary (a single executable that can do the job of many) used to create a compact version of command-line utilities that are found on Linux systems. Many of my YouTube videos use Ångström, as it was the primary distribution for the BeagleBone for quite some time.

Ubuntu is closely related to Debian; in fact, it is described on the Ubuntu website (`www.ubuntu.com`) as follows: "Debian is the rock upon which Ubuntu is built." Ubuntu is one of the most popular desktop Linux distributions, mainly because of its focus on making Linux more accessible to new users. It is easy to install and has excellent desktop driver support, and there are binary distributions available for the Beagle boards.

Arch Linux is a lightweight and flexible Linux distribution that aims to "keep it simple," targeting competent Linux users in particular by giving them complete control and responsibility over the system configuration. There are prebuilt versions of the Arch Linux distribution available for the Beagle boards; however, compared to the other distributions, it currently has less support for new Linux users (see `www.archlinux.org`).

NOTE **Don't be too worried that you might damage the Linux file system when you are practicing with the Beagle boards. In the worst case, you might have to write a new Linux image to the micro-SD card or to the eMMC. It takes about 20–45 minutes to write the image to the board. There is a guide to writing a new image to the BeagleBone eMMC on the chapter web page at** `www.exploringbeaglebone.com/chapter2/`**.**

Create a Linux Micro-SD Card Image

The easiest way to set up an SD card so that it can be used to boot the PocketBeagle or other Beagle boards is to download a Linux distribution image file (`.img` file in a compressed `.xz` wrapper) from `beagleboard.org/latest-images` and write it to an SD card using an image writer utility such as Etcher (`etcher.io`). Etcher is particularly useful because it can use the compressed image file directly. The application is available for Windows, Mac, and Linux host machines.

WARNING **All previous content on the micro-SD card is lost after performing this action. Please double-check that you are writing the downloaded image to the correct device on your desktop machine when using the Etcher tool.**

Communicating with the Boards

When you are ready to try your Beagle board, the first thing you should do is connect it to your desktop computer using a USB lead. After you apply power, the board will connect to the desktop in USB client mode. Once it's connected and discovered, your file manager, such as Windows Explorer, will display the contents of the board's FAT partition, as shown in Figure 2-1. The BeagleBoard.org team has put together a really excellent HTML guide on getting started with your board. You should double-click the `START.htm` file to display the guide, which is illustrated in Figure 2-1, within a web browser.

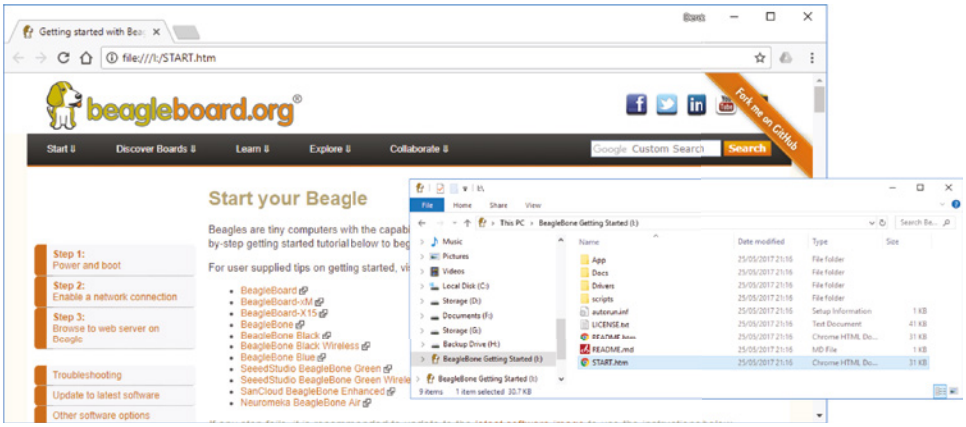

Figure 2-1: The `START.htm` guide to setting up your board (running under Windows 10)

NOTE **Earlier versions of Linux on the Beagle boards had important boot files, such as `MLO`, `u-boot.img`, and `uEnv.txt`, on the FAT partition shown in Figure 2-1. In recent images, these files have moved to the `/boot/` directory on the Linux partition and to a hidden partition, which means they cannot be edited directly from Windows.**

Installing Drivers

Follow the steps in the guide displayed in Figure 2-1. With the latest Beagle board Linux images you should no longer have to install drivers for your operating system. If you have an older image and need to install drivers, browse to the

`Drivers` folder and install the correct version. Once the board is fully booted, several new devices should be available on your desktop computer. For example, you will now have the following devices:

- Access to the FAT partition of the board (like a USB memory key).
- Serial access to the board using a new *Gadget Serial* driver COM port.
- A *Linux USB Ethernet/RNDIS Gadget* (for internet-over-USB). RNDIS stands for Remote Network Driver Interface Specification.

These new devices can be used to connect to the board. As you progress through the guide while the board is attached to your PC, the guide will highlight active connections, as illustrated in Figure 2-2. Similar steps for Linux and Mac desktop computers are available in the startup guide.

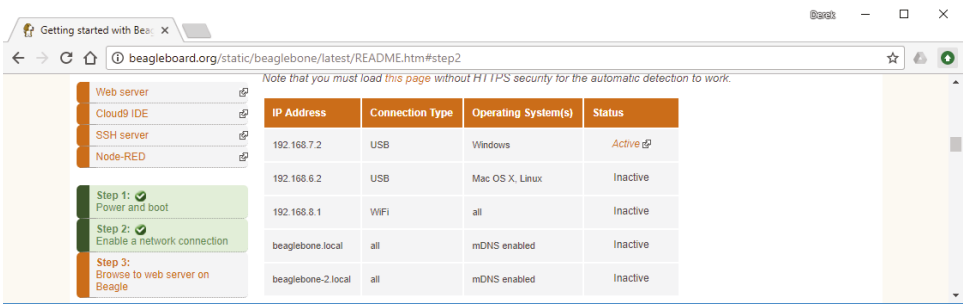

Figure 2-2: The `START.htm` guide will highlight active connections

Wired Network Connections

There are three main ways to connect to and communicate with the board over a wired network, each with its own advantages and disadvantages. The first way is to use *internet-over-USB*, which creates a "private" virtual LAN using a single USB cable. This approach works for all Beagle boards, but the following approaches require a board that has an Ethernet adapter. The second way is to use *regular Ethernet*, and the third is to use an *Ethernet crossover cable*. Connecting to the board over a network can be a stumbling block for beginners. It is usually straightforward if you are working at home with control of your own network; however, complex networks, such as those in universities, can have multiple subnets for wired and wireless communication. In such complex networks, routing restrictions may make it difficult, if not impossible, to connect to the board over regular Ethernet.

NOTE **Several boards have on-board Wi-Fi capabilities, but you may need to connect to the board over an internet-over-USB or serial connection to configure Wi-Fi for connection to your network. The default Wi-Fi address is 192.168.8.1, as listed in Figure 2-2.**

To modify the configuration files for a Wi-Fi adapter, you can use the USB-to-TTL cable that is described in the next section. Once you have a wired connection to the board, jump to Chapter 12 on configuring Wi-Fi, before continuing from this point.

Alternatively, you could mount the micro-SD card for the target board under a desktop Linux OS (or another booted BeagleBone) and modify the configuration files directly.

Internet-over-USB (All Boards)

The standard BeagleBoard.org Linux distributions provide support for internet-over-USB using the Linux USB Ethernet/RNDIS Gadget device. For new users and for users within complex network infrastructures, this is probably the best way to get started with your board. For this setup you need only the board, a USB cable, and access to a desktop computer, ideally with administrator access levels. Table 2-1 describes the advantages and disadvantages of internet-over-USB for connecting to a board.

Table 2-1: Advantages and Disadvantages of Internet-over-USB

ADVANTAGES	DISADVANTAGES
Provides a stable network setup for beginners.	Without significant effort, you are limited to a single board per desktop.
When you do not have access to, or control of, network infrastructure hardware, you can still connect the board to the internet.	Network sharing configuration can be difficult, especially on Macintosh desktop computers. Additional configuration must also be performed on the board's Linux OS.
Power is supplied by your desktop machine over USB.	Your desktop machine must be running to transfer data to/from the internet.

NOTE **By default, with internet-over-USB, the board has the fixed IP address 192.168.7.2 under Windows and 192.168.6.2 under macOS and Linux. From the perspective of the Beagle board, the desktop machine has the fixed address 192.168.7.1 under Windows and 192.168.6.1 under macOS and Linux.**

For example, if you fully installed the board drivers under Windows, you should now have a new network connection (click Start, type **view network status and tasks**, and select "Change adapter settings"). Figure 2-3 shows a typical Network Connections window under Windows. In this case, "Ethernet 3" is the Linux USB Ethernet/RNDIS Gadget device. The desktop computer remains connected

to your regular LAN (via Ethernet 2 in my case), which provides access to the internet and to a new private LAN that contains only the desktop computer (e.g., 192.168.7.1) and your board (e.g., 192.168.7.2). You can open a web browser and connect to the board's web server by typing **192.168.7.2** (or **192.168.6.2** on a Mac/Linux machine) in the address bar, as illustrated in Figure 2-3.

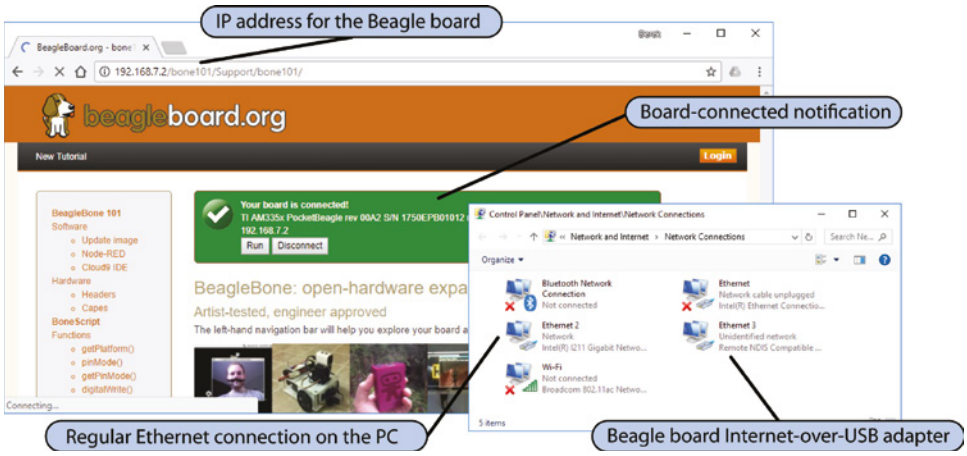

Figure 2-3: Windows network connections with Internet-over-USB connection and a web browser connection

At this point, you can connect to the board's web server using a web browser so you have a fully functional private network; however, you may also want the board to have full direct access to the internet so that you can download files and update Linux software directly on the board. To do this, you need to share your main network adapter so that traffic from the board can be routed through your desktop machine to the internet. For example, under Windows, use the following steps:

1. Choose your desktop/laptop network adapter that provides you with internet access. Right-click it and choose Properties.
2. In the dialog that appears, as shown on the left side of Figure 2-4, click the Sharing tab at the top and enable the option "Allow other network users...."
3. In the drop-down list, choose your private LAN (e.g., referring to Figure 2-4, this is "Ethernet 3" in my case). Click OK.
4. Right-click the private LAN (e.g., Ethernet 3) and select Properties.

5. Double-click Internet Protocol Version 4. In this dialog, select "Obtain an IP address automatically" and enable "Obtain DNS server address automatically" (see Figure 2-4 on the right side).
6. Click OK and then Close to save the configurations.

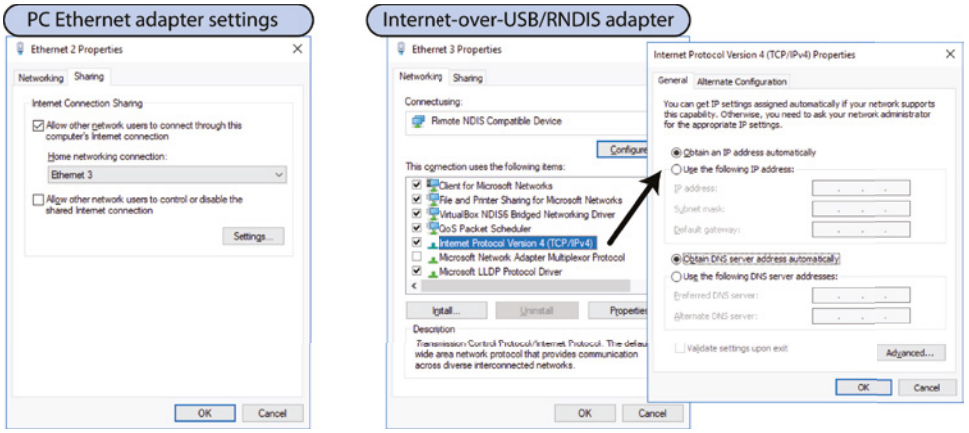

Figure 2-4: Configuring the network connection sharing properties under Windows

The steps are similar under macOS using Apple Menu ➪ System Preferences ➪ Sharing. If all goes well, you will not have noticed any difference at this point, and you should be able to reload the web page that is shown in Figure 2-3. It can take about one minute for the network configuration to finalize. Please note that the impact of the last two steps can be appreciated only when you open a terminal connection to the board.

WARNING **If you are planning to jump ahead, there is one more step to complete before your board will be able to "see" the internet. This change, which has to be made directly on the board, is covered later in this chapter in the section titled "What Time Is It?" Please note it may be necessary to unshare and reshare the internet connection if you are having internet connectivity problems, which may arise if the shared connection is used over a number of days.**

NETWORK SHARING FOR LINUX DESKTOP USERS

The settings for a Linux desktop to enable network sharing are as follows:

1. **With the internet-over-USB device attached, type `ifconfig` or `ip addr` in a terminal, which results in a display of the attached network interfaces.**
2. **Find your main adapter (e.g., `eth0`) and internet-over-USB adapter (e.g., `eth1`).**

3. **Use the `iptables` program to configure the Linux kernel firewall rules.**

```
molloyd@debian:~$ sudo iptables --table nat --append POSTROUTING
--out-interface
eth0 -j MASQUERADE
molloyd@debian:~$ sudo iptables --append FORWARD --in-interface eth1 -j
ACCEPT
```

4. **Then, use the following command to turn on IP forwarding:**

```
molloyd@debian:~$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Regular Ethernet (BBB and BeagleBoard Only)

By "regular" Ethernet, I mean connecting the board to a network in the same way you would connect your desktop computer using a wired connection. For the home user and power user of the BeagleBone Black (BBB), regular Ethernet is probably the best solution for networking and connecting to the board. Table 2-2 lists the advantages and disadvantages of using this type of connection. The main issue is the complexity of the network—if you understand your network configuration and have access to the router settings, then this is by far the best configuration. If your network router is distant from your desktop computer, you can use a small network switch, which can be purchased for as little as $10–$20. Alternatively, you could purchase a wireless access point with an integrated multiport router for $25–$35 so that you can integrate BeagleBone Black Wireless (BBBW) boards.

Table 2-2: Regular Ethernet Advantages and Disadvantages

ADVANTAGES	DISADVANTAGES
You have full control over IP address settings and dynamic/static IP settings.	You might need administrative control or knowledge of the network infrastructure.
You can connect and interconnect many BBBs to a single network (including wireless devices).	The BBB needs a source of power (which can be a mains-powered adapter).
The BBB can connect to the internet without a desktop computer being powered on.	The setup is more complex for beginners if the network structure is complex.

The first challenge with this configuration is finding your BBB on the network. By default, the board is configured to request a *Dynamic Host Configuration Protocol* (DHCP) IP address. In a home network environment, this service is usually provided by a DHCP server that is running on the integrated modem-firewall-router-LAN (or some similar configuration) that connects the home to an internet service provider (ISP).

DHCP servers issue IP addresses dynamically from a pool of addresses for a fixed time interval, called the *lease time*, which is specified in your DHCP configuration. When this lease expires, your board is allocated a different IP address the next time it connects to your network. This can be frustrating, as you may have to search for your board on the network again. It is possible to set the IP address of your board to be *static* so that it is fixed at the same address each time the board connects. Wireless connections and static IP connections are discussed in Chapter 12.

There are a few different ways to find your board's dynamic IP address.

- Use a web browser to access your home router (often address 192.168.1.1, 192.168.0.1, or 10.0.0.1). Log in and look under a menu such as Status for DHCP Table. You should see an entry that details the allocated IP address, the physical MAC address, and the lease time remaining for a device with hostname `beaglebone`. Here's an example:

```
Leased Table
IP Address       MAC Address Client   Host Name    Register Information
192.168.1.116    c8:a0:30:c0:6b:48    beaglebone   Remains 23:59:51
```

- Use a port-scanning tool like `nmap` under Linux or the *Zenmap* GUI version that is available for Windows (see `tiny.cc/beagle202`). The command `nmap -T4 -F 192.168.1.*` will scan for devices on a subnet. You are searching for an entry that typically has five open ports (e.g., 22 for SSH, 53 for DNS, 80 for the BeagleBoard.org web guide, 3000 for the Cloud 9 IDE, and 8080 for an Apache web server). It should also identify itself with Texas Instruments.
- You could use a serial-over-USB connection to connect to the board and type **`ifconfig`** to find the IP address. The address is the `inet addr` value associated with the `eth0` adapter. This is discussed shortly.

Once you have the IP address, you can test that it is valid by entering it in the address bar of your web browser (it's `192.168.1.116` in the previous example). Your browser should display the same page shown in Figure 2-3.

Ethernet Crossover Cable (BBB and BeagleBoard Only)

An Ethernet crossover cable is a cable that has been modified to enable two Ethernet devices to be connected directly together, without the need for an Ethernet switch. It can be purchased as a cable or as a plug-in adapter. Table 2-3 describes the advantages and disadvantages of this connection type.

Table 2-3: Crossover Cable Network Advantages and Disadvantages

ADVANTAGES	DISADVANTAGES
When you do not have access to network infrastructure hardware, you can still connect to the BBB.	If your desktop machine has only one network adapter, then you will lose access to the internet. It is best used with a device that has multiple adapters.
BBB may have internet access if the desktop has two network adapters.	BBB still needs a source of power (can be a mains-powered adapter).
Provides a reasonably stable network setup.	Replicates the functionality of internet-over-USB, so try that approach first.

Most modern desktop machines have an automatic crossover detection function (Auto-MDIX) that enables a regular Ethernet cable to be used. The network interface on the Beagle boards also supports Auto-MDIX; therefore, this connection type can be used when you do not have access to network infrastructure. If you have two network adapters on your desktop machine (e.g., a laptop with a wired and wireless network adapter), then you can easily share the connection to the internet with your BBB by bridging both adapters. For example, these are the steps that you must take under the Windows OS:

1. Plug one end of a regular (or crossover) Ethernet cable into the BBB and the other end into a laptop Ethernet socket.
2. Power on the BBB by attaching a USB power supply.
3. You can then bridge the two network connections—under Windows click Start, type **view network status** and tasks, select "Change adapter settings," select the two network adapters (wired and wireless) at the same time, right-click, and choose Bridge Connections. After some time, the two connections should appear with the status "Enabled, Bridged," and a network bridge should appear.
4. Reboot the BBB. Once the board has rebooted, it should obtain an IP address directly from the DHCP server of your network.

You can then communicate with the BBB directly from anywhere on your network (including the PC/laptop itself) using the steps described in the next section. This connection type is particularly useful inside complex network infrastructures such as those in universities, as the laptop can connect to the BBB using the address that it is assigned. The BBB can also continue to connect to the internet.

Communicating with Your Board

Once you have networked the board, the next thing you will want to do is communicate with it. You can connect to the Beagle board using either a serial connection over USB, USB-to-TTL, or a network connection, such as that just discussed. The network connection should be your main focus, as that type of connection provides your board with full internet access. The serial connection is generally used as a fallback connection when problems arise with the network connection. As such, you may skip the next section, but the information is here as a reference for when problems arise.

NOTE **The default login details for Debian are username** `debian` **with the password *temppwd*. Under Ubuntu use** `ubuntu/temppwd`**. Ångström and Arch Linux have username** `root` **and no password (just press Enter).**

Serial Connection over USB

If you installed the device drivers for the Beagle board in the previous section, the *Gadget Serial device* will allow you to connect to the board directly using a terminal emulator program. Serial connections are particularly useful when the board is close to your desktop computer and connected via the USB cable. It is often a fallback communications method when something goes wrong with the network configuration or software services on the board.

To connect to the board via the serial connection, you need a terminal program. Several third-party applications are available for Windows, such as RealTerm (`realterm.sourceforge.io`) and PuTTY (`www.putty.org`). PuTTY is also used in the next section. Most distributions of desktop Linux include a terminal program (try Ctrl+Alt+T or use Alt+F2 and type **`gnome-terminal`** under Debian). A terminal emulator is included by default under macOS (e.g., type **`screen/dev/tty.usbmodemfa133 115200`**).

To connect to the Beagle board over the USB serial connection, you need to know some information.

- **COM port number:** You can find this by opening the Windows Device Manager and searching under the Ports section. Figure 2-5 captures an example Device Manager, where the Gadget Serial device is listed as COM3. This will be different on different machines.
- **Speed of the connection:** By default you need to enter **115,200** baud to connect to the board.
- **Other information you may need for other terminal applications:** Data bits = 8; Stop bits = 1; Parity = none; and, Flow control = XON/XOFF.

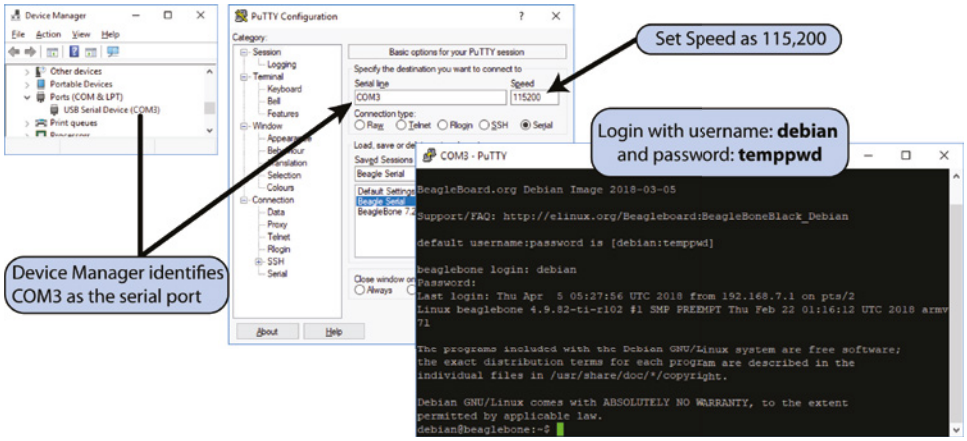

Figure 2-5: Windows Device Manager and opening a PuTTY serial connection to the board

Save the configuration with a session name so that it is available each time you want to connect. Click Open, and it is important that you *press Enter when the window appears*. When connecting to Debian, you should see the following output:

```
Debian GNU/Linux 9 beaglebone ttyGS0

BeagleBoard.org Debian Image 2018-03-05
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
default username:password is [debian:temppwd]
beaglebone login: debian
Password: temppwd
debian@beaglebone:~$
```

This allows you to log in with username `debian` and password `temppwd`.

On a Linux desktop computer, you can install the `screen` program and connect to the serial-over-USB device with these commands:

```
molloyd@debian:~$ sudo apt-get install screen
molloyd@debian:~$ screen /dev/ttyUSB0/ 115200
```

Serial Connection with the USB-to-TTL 3.3 V Cable

For this serial connection type, you need the specialized cable (or the cheaper alternative in Figure 1-9(c)) that is described in Chapter 1. Find the COM port from Windows Device Manager that is associated with a device called USB Serial Port. Plug in the cable to the six-pin connector beside the P9 header (black lead to the white dot/J1), as illustrated in Figure 1-9(a), or to the PocketBeagle expansion header as illustrated in Figure 1-9(b).

You can then open a serial connection using PuTTY (115,200 baud), and you will see the same login prompt as earlier. However, when you reboot the board, you will also see the full console output as the board boots, which begins with the following (in the case of the PocketBeagle):

```
U-Boot SPL 2018.01-00002-ge9ff418fb8 (Feb 20 2018 - 20:14:57)
Trying to boot from MMC1
U-Boot 2018.01-00002-ge9ff418fb8 (Feb 20 2018 - 20:14:57 -0600),
Build: jenkins-github_Bootloader-Builder-38
CPU  : AM335X-GP rev 2.1
I2C:   ready
DRAM:  512 MiB
...
Model: BeagleBoard.org PocketBeagle
...
```

This is the ultimate fallback connection, as it allows you to see what is happening during the boot process, which is described in the next chapter.

Connecting Through Secure Shell

Secure Shell (SSH) is a useful network protocol for secure encrypted communication between network devices. You can use an SSH terminal client to connect to the SSH server that is running on port 22 of your board, which allows you to do the following:

- Log in remotely to the board and execute commands
- Transfer files to and from the board using the *SSH File Transfer Protocol* (SFTP)
- Forward X11 connections, which allows you to perform virtual network computing (covered in Chapter 13)

By default, the BeagleBone Linux distributions run an SSH server (*sshd* on Debian and *Dropbear* on Ångström) that is bound to port 22. There are a few advantages in having an SSH server available as the default method by which you log in remotely to your board. In particular, you can open port 22 of the board to the internet using the port forwarding functionality of your router. Please ensure that you set a password on the root user account before doing this. You can then remotely log in to your board from anywhere in the world if you know its IP address. A service called *dynamic DNS* that is supported on most routers allows your router to register its latest address with an online service. The online service then maps a domain name of your choice to the latest IP address that your ISP has given you. The dynamic DNS service usually has an annual cost, for which it will provide you with an address of the form `MyBeagle.servicename.com`.