



C o m m u n i t y   E x p e r i e n c e   D i s t i l l e d

# BeagleBone Robotic Projects

Create complex and exciting robotic projects with  
the BeagleBone Black

Richard Grimmett

**[PACKT]**  
PUBLISHING

# BeagleBone Robotic Projects

Create complex and exciting robotic projects with  
the BeagleBone Black

**Richard Grimmett**

**[PACKT]**  
PUBLISHING  
BIRMINGHAM - MUMBAI

# BeagleBone Robotic Projects

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: December 2013

Production Reference: 1181213

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78355-932-9

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Disha Haria ([dishah@packtpub.com](mailto:dishah@packtpub.com))

# Credits

**Author**

Richard Grimmett

**Project Coordinator**

Leena Purkait

**Reviewers**

Álvaro García Gómez

Lihang Li

Derek Molloy

**Proofreader**

Chris Smith

**Indexer**

Tejal Soni

**Acquisition Editor**

Sam Birch

**Graphics**

Sheetal Aute

Abhinash Sahu

**Lead Technical Editor**

Chalini Snega Victor

**Production Coordinators**

Alwin Roy

Kirtee Shingan

**Technical Editors**

Jalasha D'costa

Monica John

Edwin Moses

Nikhil Potdukhe

Siddhi Rane

Sonali S. Vernekar

**Cover Work**

Kirtee Shingan

# About the Author

**Richard Grimmett** has always been fascinated by computers and electronics from his very first programming project that used Fortran on punch cards. He has a Bachelor's and Master's degree in Electrical Engineering and a PhD in Leadership Studies. He also has 26 years of experience in the Radar and Telecommunications industries, and even has one of the original brick phones. He now teaches Computer Science and Electrical Engineering at Brigham Young University - Idaho where his office is filled with many of his robotics projects.

---

I would certainly like to thank my wife and family for providing me the time and wonderful, supportive environment that encourages me to take on projects such as this one. I would also like to thank my students; they always amaze and inspire me with their creativity when released from the boredom of standard educational practices.

---

# About the Reviewers

**Álvaro García Gómez** is a computer engineer at the University of Valladolid (Spain) and a technical administrator of IT systems. He was focused on software development, but a short time later robotics and embedded devices aroused his curiosity. Now he is specialized in machine learning and autonomous robotics, which involve his two passions: computing and electronics. Now he is working in his own company that develops free software and hardware.

**Lihang Li** received his B.E. degree in Mechanical Engineering from Huazhong University of Science and Technology (HUST), China in 2012 and is now pursuing his M.S. degree in Computer Vision at National Laboratory of Pattern Recognition (NLPR) from the Institute of Automation, Chinese Academy of Sciences (IACAS).

He is a member of Dian Group from HUST and mainly concentrated on Embedded System Development when he was an undergraduate. He is familiar with Embedded Linux, ARM, DSP, and various communication interfaces (I2C, SPI, UART, CAN, and ZigBee, among others). He took part in a competition called The Asia-Pacific Robot Contest (ABU Robocon) with his team in 2012 and secured third place among 29 teams in China.

As a graduate student, he is focusing on Computer Vision and specially on SLAM algorithms. In his free time, he likes to take part in Open Source Activities and now is President of the Open Source Club, Chinese Academy of Sciences. Also, building a multicopter is his hobby and he is with a team called OpenDrone from Beijing Linux User Group (BLUG).

His interest includes: Linux, Open Source, Cloud Computing, Virtualization, Computer Vision algorithms, Machine Learning and Data Mining, and various programming languages.

You can find him at his personal website, <http://hustcalm.me>.

---

Many thanks to my girlfriend Jingjing Shao; it was her encouragement to push me to be a reviewer for this book. And I appreciate her kindness though sometimes I can't spare time for her. Also, I must thank all the team: Leena, who is a very good Project Coordinator, and the other reviewers, though we haven't met, I'm happy to work with you.

---

**Derek Molloy** is a senior lecturer in the School of Electronic Engineering, Faculty of Engineering & Computing at Dublin City University, Ireland. Since 1997, he has lectured in object-oriented programming, 3D Computer Graphics, and Digital Electronics at postgraduate and undergraduate levels. His research interests are in the fields of Computer & Machine Vision, 3D Graphics and Visualization, and e-learning. He is a key academic member of the Centre for Image Processing and Analysis (CIPA) at DCU. He has published his works widely in international journals and conferences, including an important textbook, *Machine Vision Algorithms in Java*, Springer (2001). In his spare time he runs the DerekMolloyDCU YouTube channel that contains many instructional videos on the use of the BeagleBone, and he integrates everything on his personal blog at [www.derekmolloy.ie](http://www.derekmolloy.ie).



# www.PacktPub.com

## Support files, eBooks, discount offers, and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Chapter 1: Getting Started with the BeagleBone Black</b>	<b>9</b>
Mission briefing	9
The unveiling!	11
Hooking up a keyboard, mouse, and display	15
Changing the operating system	18
Adding a graphical user interface	22
Accessing the board remotely	26
Mission accomplished	34
A challenge	34
<b>Chapter 2: Programming the BeagleBone Black</b>	<b>35</b>
Mission briefing	35
Basic Linux commands and navigating the filesystem	36
Creating, editing, and saving files on the BeagleBone Black	42
Creating and running Python programs on the BeagleBone Black	43
Basic programming constructs on the BeagleBone Black	47
Introduction to the C++ programming language	54
Mission accomplished	59
A challenge	59
<b>Chapter 3: Providing Speech Input and Output</b>	<b>61</b>
Mission briefing	61
Hooking up the HW to make and input sound	64
Using eSpeak to allow your projects to respond in a robotic voice	70
Using PocketSphinx to interpret your commands	73
Providing the capability to interpret your commands and have your robot initiate an action	80

---

*Table of Contents*

---

Mission accomplished	83
A challenge	83
<b>Chapter 4: Allowing the BeagleBone Black to See</b>	<b>85</b>
Mission briefing	85
Connecting the USB camera to the BeagleBone Black and viewing the images	86
Downloading and installing OpenCV – a full-featured vision library	89
Using the vision library to detect colored objects	97
Mission accomplished	102
Challenges	102
<b>Chapter 5: Making the Unit Mobile – Controlling Wheeled Movement</b>	<b>103</b>
Mission briefing	103
Using a motor controller to control the speed of your platform	107
Controlling your mobile platform programmatically using the BeagleBone Black	117
Making your mobile platform truly mobile by issuing voice commands	122
Mission accomplished	124
A challenge	124
<b>Chapter 6: Making the Unit Very Mobile – Controlling Legged Movement</b>	<b>125</b>
Mission briefing	125
Connecting the BeagleBone Black to the mobile platform using a servo controller	130
Creating a program in Linux to control the mobile platform	138
Making your mobile platform truly mobile by issuing voice commands	142
Mission accomplished	143
A challenge	144
<b>Chapter 7: Avoiding Obstacles Using Sensors</b>	<b>145</b>
Mission briefing	145
Connecting the BeagleBone Black to a USB sonar sensor	148
Using a servo to move a single sensor	154
Mission accomplished	159
A challenge	159
<b>Chapter 8: Going Truly Mobile – Remote Control of Your Robot</b>	<b>161</b>
Mission briefing	161
Connecting the BeagleBone Black to a wireless USB keyboard	167
Using the keyboard to control your project	169
Mission accomplished	174
A challenge	174
<b>Chapter 9: Using a GPS Receiver to Locate Your Robot</b>	<b>175</b>
Mission briefing	175
Connecting the BeagleBone Black to a GPS device	176
Accessing the GPS programmatically and determining	

how to move to a location	188
Mission accomplished	193
A challenge	193
<b>Chapter 10: System Dynamics</b>	<b>195</b>
Mission briefing	195
Creating a general control structure so capabilities can communicate	197
Mission accomplished	205
A challenge	206
<b>Chapter 11: By Land, Sea, and Air</b>	<b>207</b>
Mission briefing	207
Using the BeagleBone Black in sailing robots	208
Using the BeagleBone Black in flying robots	216
Using the BeagleBone Black in submarine robots	222
Mission accomplished	224
A challenge	224
<b>Index</b>	<b>225</b>



# Preface

We live in an amazing age. We are mostly aware of how amazing it is as we live in an age where major changes to how we live occur well within a lifetime, sometimes within a few years. Nowhere is this more evident than in the general area of technology, and the specific area of computers. Not so many years ago, certainly within the lifespan of most of the baby-boomer generation, computers were distant machines kept in the backrooms of large corporations or universities. Access to them was tightly controlled. If you wanted to program them, you punched your computer cards, fed them into the card reader, and then, after an hour or so of wait, you went to receive your computer printout. This was, I regret to reveal, part of my first experience with a computer.

These large computers were the domain of companies such as IBM, with their model 360, Digital Equipment, with the Model PDP-7, and Hewlett-Packard, with the Model 1000. These computers cost many thousands of dollars, and were rarely seen except by a privileged few, who had access to climate-controlled computer rooms.

This model fit the world just wonderfully for many years, until the advent of the personal computer. I was lucky to know someone who purchased one of the very first IBM-PCs. It had two floppies, a monochrome monitor, and was an amazing piece of equipment. Suddenly the world changed and the technology that had seemed so remote was now available on the desktop. This same technological revolution in processing power also birthed a new breed of dedicated microprocessors. These could be used for specific tasks that had previously been the realm of analog circuitry or, in many cases, human interaction with mechanical systems.

These processing solutions to specific applications are named embedded systems. They take the digital calculating capability of personal computers and shrink them even further so they can be placed in common household and industrial objects. Embedded technology has also evolved with respect to price; fortunate, for few would be willing to pay several thousand dollars for a door lock or temperature sensor. The initial embedded devices were very limited in their technology, and developing applications with them became quite a challenge. It was very common to run out of either computing horsepower or memory. Many nights were spent by the talented few shoehorning the last features into the last few bytes of memory.

The computer age has spawned an amazing array of technical advances in both the hardware and software areas. Companies such as Intel and AMD have created processors with almost unfathomable computing power and more available memory than once thought possible, and both Microsoft and Apple have provided major advances in the area of software functionality and usability. The personal computer has become a standard tool in most households, schools, businesses, and factories.

As the personal computer has gone, so has the embedded systems world. From what were once four bit, special purpose processors with 2000 bytes of memory, now embedded processors have emerged that rival the performance and capability of standard personal computers. One has to look no further than the cell phone for an example of significant computing capability in very small packages, and at very inexpensive prices.

This has all reached a bit of a crescendo with the introduction of small, inexpensive systems that can not only run simple, focused applications, but have the capability of powering almost any type of computing need we can create. At the same time these small but powerful systems have outgrown the small, single purpose development environments as well. They are now paired with powerful operating systems, and provide personal computer-like functionality in very small packages. The overwhelming advance of tablets and smart phones has begun to take over the face of computing for many applications.

These advances have also affected the embedded area as well. Small, highly capable systems have married very inexpensive hardware with free, open source software to provide a platform for almost anyone to explore the embedded world. The Arduino, the Raspberry Pi, and now the BeagleBone Black are all platforms that offer not only an affordable price point, but also an open source software community that provides free capability and an easy way to interact with others to get answers to questions or exchange ideas. With these new capabilities, as we shall see later in the book, the sky is literally the limit.

This book will focus on just one of these processors, the BeagleBone Black. However, much of what is written here could be applied to other choices with some limited modifications. But this is not what you came here to learn. You came to learn how to build some very interesting, complex, amazing robotics projects. And processors such as the BeagleBone Black are impressive because they have the capability to not only make this possible, but to make it accessible to those outside of academic or research communities. In this book, we'll explore these capabilities, and build some very impressive projects.

Just a few comments on how the book is laid out. We'll start with a very basic introduction to the BeagleBone Black, and how to get the hardware and software up and working. Then, we'll build some basic functionality on top of the basic system, showing you how to add sound, vision, and control.

Then we'll tackle some fairly complex capability, including GPS, audio, and some advanced sensors. Finally, we'll wrap it up by showing you how to put an entire system together with some tools that can make that a bit less complicated.

In each chapter, I'll give you some very specific instructions for how to proceed. This is a bit dangerous, and the instructions are all going to be subject to change. Hopefully you'll understand the basics of what we are trying to accomplish, so if things don't go quite to plan, you'll be able to figure out how to proceed. There is a lot of help out there, between message boards and blogs, so don't be shy.

What is critical to remember is that this is not an academic exercise. Don't just read the book, but do something with the hardware. My hope is that by the end, you'll be building the kinds of machines that will lead us all into the 22<sup>nd</sup> century. I often tell my students that their children will grow up as comfortable with robots as they are with computers.

So, let's begin!

## What this book covers

*Chapter 1, Getting Started with the BeagleBone Black*, will provide instructions for initial power-up of your hardware.

*Chapter 2, Programming the BeagleBone Black*, will give you a brief tutorial so that you can be successful implementing all the amazing functionality, as many of you are new to embedded systems, Linux, Python, or perhaps even programming in general.

*Chapter 3, Providing Speech Input and Output*, will show you how to add speech recognition as well as make your robot speak.

*Chapter 4, Allowing the BeagleBone Black to See*, will show you how to add the capability for your robot to see.

*Chapter 5, Making the Unit Mobile – Controlling Wheeled Movement*, will show you how to add wheeled movement to your robot.

*Chapter 6, Making the Unit Very Mobile – Controlling Legged Movement*, shows how to build robots that have the capability to walk.

*Chapter 7, Avoiding Obstacles Using Sensors*, shows how to use sensors to avoid barriers as it hardly makes sense to have mobility if your robot is going to run into obstacles.

*Chapter 8, Going Truly Mobile – Remote Control of Your Robot*, will show how to use a remote device to control your robot.

*Chapter 9, Using a GPS Receiver to Locate Your Robot*, shows how to add a GPS device to your robot.

*Chapter 10, System Dynamics*, introduces some methods for organizing all of the capabilities so that they are available at the same time.



*Chapter 11, By Land, Sea, and Air*, introduces some interesting possibilities for embedded projects that can fly, sail, or swim

## What you need for this book

Each chapter will lead you through not only the hardware, but also the software required for each project. However, for almost all of these projects you'll need a personal computer connected to the Internet, an additional Internet connection and LAN cable, the BeagleBone Black, and the power cable that comes with it.

## Who this book is for

This book is designed for the informed beginner. I would hope that before beginning the projects in this book you would be familiar with your personal computer and its basic use and functionality. You won't need prior programming experience, but it will be helpful. You'll be introduced to some of the most basic working of the Linux operating system, so any familiarity there will be helpful, but not essential. More than anything the book requires a curiosity about how robots or other embedded projects work, and the tenacity to work through the issues associated with building your own hardware and then adding software to get to a working system.

## Conventions

In this book, you will find several headings appearing frequently.

To give clear instructions of how to complete a procedure or task, we use:

### Mission briefing

This section explains what you will build, with a screenshot of the completed project.

### Why is it awesome?

This section explains why the project is cool, unique, exciting, and interesting. It describes what advantage the project will give you.

## Your objectives

This section explains the major tasks required to complete your project.

- ▶ Task 1
- ▶ Task 2
- ▶ Task 3
- ▶ Task 4, and so on

## Mission checklist

This section explains any pre-requisites for the project, such as resources or libraries that need to be downloaded, and so on.

## Task 1

This section explains the task that you will perform.

## Prepare for lift off

This section explains any preliminary work that you may need to do before beginning work on the task.

## Engage thrusters

This section lists the steps required in order to complete the task.

## Objective complete – mini debriefing

This section explains how the steps performed in the previous section allow us to complete the task. This section is mandatory.

## Classified intel

The extra information in this section is relevant to the task.

You will also find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "You can do this with the `ls -la /dev/sd*` command."

A block of code is set as follows:

```
#Smooth image, then convert the Hue
cv.Smooth(img,img,cv.CV_BLUR,3)
hue_img = cv.CreateImage(cv.GetSize(img), 8, 3)
cv.CvtColor(img,hue_img, cv.CV_BGR2HSV)
```

Any command-line input or output is written as follows:

```
xz -cd ubuntu-precise-12.04.2-armhf-3.8.13-bone20.img.xz > /dev/sdX
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "The **Safe start violation** tab is set when you first enter the program; you need to clear this by clicking on the **Resume** button at the bottom-left corner of the screen."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code and colored images

You can download the example code files and colored images for this Packt book you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.



# 1

## Getting Started with the BeagleBone Black

Ordering the hardware (HW) is the exciting part of any project. You have wonderful dreams of all that you might accomplish once this amazing piece of technology is delivered. Unfortunately, the frustration of the first few attempts at accessing the capabilities of the unit can leave many developers, especially those with little experience with this type of dedicated system, so discouraged that the board can end up on the shelf, gathering dust next to the pet rock and cassette tape recorder.

### Mission briefing

There is rarely anything as exciting as ordering the latest new technology and anticipating its arrival. You daydream of the projects you'll build, the amazing things you can do, the accolades you'll receive from family, friends, and colleagues. However, reality rarely fulfills your fantasies. This project will hopefully help you avoid the pitfalls that normally accompany unboxing and configuring your BeagleBone Black. You'll step through the process, answer all kinds of clarifying questions, and help you understand what is going on. If you don't get through this project, then you'll not be successful at any of the others, so buckle up and get ready for an exciting ride.

The most challenging aspect of accomplishing this for me as your guide is trying to decide to what level I should describe each step. Some of you are beginners, some have limited experience, others will know significantly more than I in some of these areas. I'll try to keep it brief, but also try to be thorough, so that at least you'll know what steps to take in order to be successful. I'll also try to point out some of the different ways you can get help if you are encountering problems. So for this project, here are your objectives.

## Your objectives

Your objectives are as follows:

- ▶ Hooking up a keyboard, mouse, and display
- ▶ Changing the operating system
- ▶ Adding a graphical user interface
- ▶ Accessing the board remotely



### Downloading the example code and colored images

You can download the example code and colored images for this book you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Mission checklist

Here are the items you'll need for this project:

- ▶ A BeagleBoard Black
- ▶ The USB cable provided with the board
- ▶ A display with the proper video input
- ▶ A keyboard, mouse, and powered USB hub
- ▶ A micro SD card of at least 4 GB
- ▶ A micro SD card reader/writer that fits your computer
- ▶ Another computer that is connected to the Internet
- ▶ An Internet connection for the board

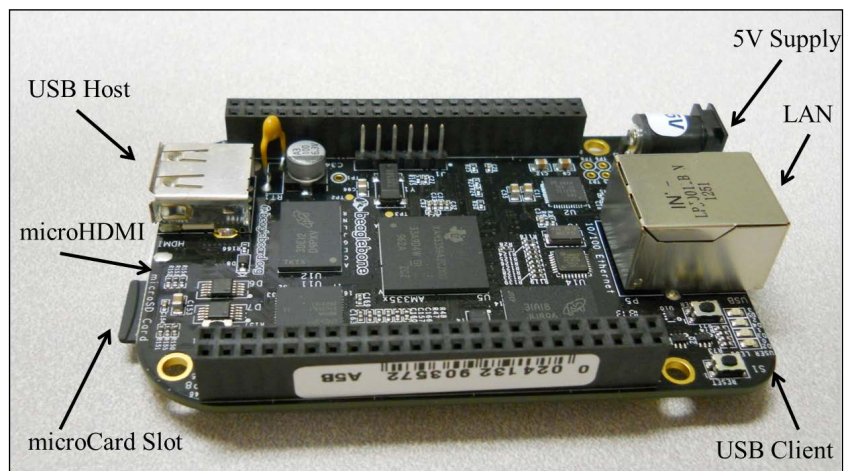
## The unveiling!

The board has finally arrived. Here is what should come with the standard package:



## Prepare for lift off

Before plugging anything in, inspect the board for any issues that might have occurred during shipping. This is normally not a problem, but it is always good to do a quick visual inspection. You should also acquaint yourself with the different connections on the board. Here they are, labeled for your information:





## Engage thrusters

So let's get started. You need to power the board, but you also need to hook up a way to interact with the board and see the results of your interaction. The first thing you'll notice is that there is no cable that fits the 5V DC connector. What's with that? Am I already hung up without ever powering on the board? Well, fortunately no, but you do need to talk about power for a moment. There are two ways to power the board. The first is through the USB client connection. This is done by:

- ▶ Connecting the micro-USB connector end of the cable to the board
- ▶ Connecting the standard sized USB connector to either a PC or a compatible DC power source that has such a connection

If you are going to use a DC power source, make sure the unit can supply at least one ampere. This is not optional. Although the board might not always draw this much current, if it senses that the unit cannot supply the required current, it will shut down.

There is another option to power the board: simply supply 5V DC to the connector. Make sure that the plug is 5.5 x 2.1 mm (centre positive) and that the unit can supply at least one ampere. As mentioned earlier, this is not optional.

Even if you are going to choose a DC power source for your board, initially let's connect the board via the provided USB cable. Almost all of the different projects you work on here will need to supply power from a battery pack anyway, and if you supply the board through the USB port and micro-USB connector, you can use your external computer to communicate with the board and ensure that it is up and working.

## Objective complete – mini debriefing

When you plug the board in, the **PWR** LED, located by the 5V input, should light blue on the board. Here is a close up of the LED locations, just so that you're certain which one to look for: