



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

Mastering BeagleBone Robotics

Master the power of the BeagleBone Black to maximize your robot-building skills and create awesome projects

Richard Grimmett

[PACKT] open source*
PUBLISHING community experience distilled

Mastering BeagleBone Robotics

Master the power of the BeagleBone Black to maximize your robot-building skills and create awesome projects

Richard Grimmett



BIRMINGHAM - MUMBAI

Mastering BeagleBone Robotics

Copyright © 2014 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 2014

Production reference: 1161214

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-890-7

www.packtpub.com

Credits

Author

Richard Grimmett

Project Coordinator

Sageer Parkar

Reviewers

Shantanu Bhadoria

Marcelo Boá

Chris J Daly

Daniel Frenzel

Brian C. Tomlinson

Proofreaders

Maria Gould

Clyde Jenkins

Elinor Perry-Smith

Indexer

Rekha Nair

Commissioning Editor

Amarabha Banerjee

Graphics

Sheetal Aute

Valentina D'silva

Abhinash Sahu

Acquisition Editor

Sam Wood

Content Development Editor

Prachi Bisht

Production Coordinator

Manu Joseph

Technical Editor

Humera Shaikh

Cover Work

Manu Joseph

Copy Editor

Vikrant Phadkay

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 a Master's degree in Electrical Engineering and a PhD in Leadership Studies. He also has 26 years of experience in electronics and computers and has one of the original brick phones and a Google Glass. He currently teaches computer science and electrical engineering at Brigham Young University, Idaho, where his office is filled with many of his robotic projects. He recently completed a book on using the Arduino for robotic projects, and another on the Raspberry PI.

I would certainly like to thank my wife, Jeanne, and family for providing me with a wonderful, supportive environment that encourages me to take up projects like this one. I would also like to thank my students who have shown me that amazing things can be accomplished by those who are unaware of the barriers.

About the Reviewers

Shantanu Bhadoria is an avid traveler, Perl expert, and CPAN author based in Singapore. He has traveled to more than 30 countries around the world. When in Singapore, he works on paging and building control systems for skyscrapers and large campuses in Singapore, Hong Kong, and Macau. He has authored many libraries in Perl for control of gyroscopes, magnetometers, accelerometers, altimeters, PWM wave generators, and many other sensors and controllers.

He is also the author of `Device::SMBus`, a Perl library that is used to talk to devices over the I2C bus, and `Math::KalmanFilter`, a Perl library that implements the Kalman Filter sensor fusion algorithm.

His Perl code can be accessed at <https://metacpan.org/author/SHANTANU>, and his GitHub repositories are available at <https://github.com/shantanubhadoria>.

Marcelo Boá is an electronics technician who has a Bachelor's degree in Information Systems. He has worked for 10 years in the field of electronic maintenance. He has also worked on Java development, Oracle PL/SQL, PHP, ZK Framework, Shell scripts, HTML, JavaScript, Ajax, Linux, Arduino, and BeagleBone.

He started as a PL/SQL trainee at the Federal Technological University of Paraná, Brazil. When the electronics technical department was formed, he worked for some companies on many different kinds of electronic circuits, thereby gaining technical assistance from Sony, Aiwa, and Gradiente. Ten years later, he returned to Java development with the ZK Framework, developing software for call centers in Curitiba's Software Park. He is currently working as a systems analyst in the warehouse management systems and industrial automation department at SSI SCHAEFER. He is an Arduino and BeagleBone lover.

This was my first experience as a reviewer and the experience was very good. The book is amazing. There are a lot of skills to learn from it.

I would like to thank God for everything. I would also like to thank Sageer and Prachi for the invitation to join the book as a reviewer, and my family for allowing my training.

Chris J Daly lives near Portland, Oregon. During the day, he works as a software engineer, and in the evening, he spends time on hacking the Internet of Things projects, which can be found on his GitHub page at <https://github.com/cjdaly>.

Daniel Frenzel is a biochemist and biophysicist, and was working as a research associate at Forschungszentrum Jülich. Recently, he got a job offer from the EMBL in Heidelberg, where he will participate in the development of microfluidic systems. Daniel has experience in parallelization with OpenMP and CUDA under supercomputing conditions. He developed a C++ library for simulation of large-scale networks (code.google.com/p/annetgpgpu). Besides, Daniel maintains a quadcopter project based on the ArduPilot library, which allows him to control a quadcopter via WiFi (code.google.com/p/rpicopter).

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

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 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 for more details.

At 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.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read 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 a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Preparing the BeagleBone Black	7
Unpacking and powering up	8
Installing an operating system	9
Connecting to an external computer	10
Installing a Windows manager	11
Installing additional core software packages	12
Installing a vision library	16
Installing sound capability	19
Creating and recording sound	20
Making your BeagleBone Black speak	23
Installing speech recognition	23
Improving speech recognition accuracy	27
Responding to voice commands	29
Adding additional hardware and software for a fully functional core system	31
Summary	31
Chapter 2: Building a Basic Tracked Vehicle	33
Choosing the tracked platform	33
Connecting a motor controller to control the speed of your tracked platform	34
Choosing the battery	35
Connecting the motor controller	37
Connecting the motor controller system	39
Controlling your mobile platform programmatically using the BeagleBone Black and Python	42
Adding program arguments to control your platform	45

Accessing motor control via voice commands	47
Summary	50
Chapter 3: Adding Sensors to Your Tracked Vehicle	51
Basics of sensors	51
Adding distance sensors	51
Sonar sensors	52
Adding an array of inexpensive sonar sensors to the project	53
IR sensors	56
Dynamic path planning for your robot	60
Basic path planning	61
Avoiding obstacles	64
Summary	69
Chapter 4: Vision and Image Processing	71
Connecting a webcam to the BeagleBone Black	71
Using OpenCV	75
Finding colored objects in your vision system	77
Following colored objects with your vision system	81
Finding movement in your vision system	83
Following movement with your robot	89
Summary	91
Chapter 5: Building a Robot that Can Walk	93
Building robots that can walk	93
Working of servomotors	94
Building the quadruped platform	95
Using a servo controller to control the servos	99
Communicating with the servo controller via a PC	101
Connecting the servo controller to the BeagleBone Black	103
Creating a program on Linux to control your quadruped	106
Issuing voice commands to your quadruped	110
Summary	111
Chapter 6: A Robot that Can Sail	113
The BeagleBone Black and robots that can sail	113
Building the sailboat platform	114
Controlling servos with the BeagleBone Black	115
Controlling the servos on the sailboat from a program	118
Remote control of the sailboat	119
A ZigBee tutorial	120
Summary	128

Chapter 7: Using GPS for Navigation	129
Beginning with a GPS tutorial	129
Connecting GPS to the BeagleBone Black	132
Communicating with the GPS	134
Parsing the GPS information	139
Calculating distance and bearing	141
Summary	144
Chapter 8: Measuring Wind Speed – Integrating Analog Sensors	145
Connecting an analog wind speed sensor	146
Getting sensor data from the wind speed sensor	148
Some basics of sailing	151
Summary	156
Chapter 9: An Underwater Remotely Operated Vehicle	157
Building the hardware for the ROV	157
Controlling brushless DC motors using the BeagleBone Black	160
Program to control DC motors using the BeagleBone Black	162
Connecting to the BeagleBone Black via a long LAN	172
Accessing a camera for your project	175
Summary	176
Chapter 10: A Quadcopter	177
Basics of quadcopter flight	177
Building the quadcopter	180
Connecting the BeagleBone Black to the quadcopter	181
Controlling the quadcopter using the BeagleBone Black	183
Summary	194
Chapter 11: An Autonomous Quadcopter	195
Controlling quadcopter flight wirelessly	195
Adding a game controller to your system	197
Adding a webcam for autonomous flight	205
Adding GPS for autonomous flight	212
Summary	212
Index	213

Preface

Robotics, the art and science of building machines that can perform some of the same functions as humans or animals, has been a part of human creative ambition since the time of the Greeks, who conceived Talos—a warrior made entirely of bronze that protected their lands and people. Leonardo da Vinci designed a mechanical knight that could sit, stand, and raise its visor. The first *modern* robots emerged in the 1920s—robots that could perform the same sort of simple motions as da Vinci's mechanical man through the use of electrical motors and signals.

In recent times, advances in technology have pushed the development of robots to the point of reality. Most of these projects have been created in defense department laboratories and university research facilities. However, new developments in inexpensive hardware and open source software have created opportunities for almost anyone to construct and experiment with automated machines.

The purpose of this book is not only to facilitate but also inspire these kinds of efforts. Robotics is no longer reserved for the PhD or even the trained engineer. You—yes, you—can construct machines that can roll, walk, swim, and even fly with the kind of functionality that is normally associated with intelligent life. I often tell my students that while they laugh at me because I once had to wash dishes by hand, their children will laugh at them because they had to load and unload the dishwasher by hand.

So, off to building the next great robotic breakthrough!

What this book covers

Chapter 1, Preparing the BeagleBone Black, teaches you how to unbox, power up, and load the BeagleBone Black with all of the necessary software to build the projects described in this book.

Chapter 2, Building a Basic Tracked Vehicle, describes your first project, that is, a tracked vehicle. You'll learn how to construct the platform so that you can control the speed and direction of your robot.

Chapter 3, Adding Sensors to Your Tracked Vehicle, explains how, after creating a platform that is mobile, you'll add sensors to help your robot avoid, or find, objects.

Chapter 4, Vision and Image Processing, shows that your robot is maneuverable, but now you'll add the capability to not only sense but also see the world around it.

Chapter 5, Building a Robot that Can Walk, teaches you to build a robot that can walk—a quadruped robot controlled by the BeagleBone Black.

Chapter 6, A Robot that Can Sail, proves that while a robot that can roll or walk is impressive, one that can sail is even more amazing. You'll build a robot that can sail under the control of the BeagleBone Black.

Chapter 7, Using GPS for Navigation, shows that sailing your sailboat is easy if a human is controlling it. You can give your sailboat autonomy by giving it a sense of where it is and where it is going. The GPS system provides this feature and you'll learn how to use it to sail.

Chapter 8, Measuring Wind Speed – Integrating Analog Sensors, teaches you that to sail your sailboat, you'll need to know the wind direction. You'll learn how to use an anemometer (wind sensor) to sail your boat.

Chapter 9, An Underwater Remotely Operated Vehicle, demonstrates the construction of a vessel that can be controlled from the surface while it explores the environment under the water.

Chapter 10, A Quadcopter, shows that, now that you've explored the earth and the water, there is only one more place left to go—the air. You'll learn how to construct one of the most stable and maneuverable flying platforms—the quadcopter.

Chapter 11, An Autonomous Quadcopter, covers the final step in exploring the heavens, that is, providing your quadcopter with the capability of flying itself.

What you need for this book

Generally, each chapter will detail the hardware and software needed for that chapter. However, before you get started, you'll need a BeagleBone Black, a host computer (Windows, Linux, or Mac), and an Internet connection for your BeagleBone Black, probably through some sort of LAN cable.

Who this book is for

If you want a simple guide to building complex robots, then this book is for you. You'll need some programming knowledge, and experience of working with mechanical systems.

Conventions

In this book, you will 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, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Since the make system already knows how to build the `pocketsphinx_continuous` program, any time you make a change to the `continuous.c` file, it will rebuild the application."

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

```
storage = cv.CreateMemStorage(0)
```

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: "Go to the **WIFI Adapters** section to see some devices that others have successfully used."



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, 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.

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

You can download the example code files for all Packt books 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.

Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: https://www.packtpub.com/sites/default/files/downloads/89070S_Graphics.pdf.

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 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 if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Preparing the BeagleBone Black

The BeagleBone Black, with its low cost and amazing package of functionalities, provides an excellent set of core functionalities to build robotic projects. In this book, you'll build three robotics projects, each with a differing array of fascinating functionalities. Hopefully, these will inform and inspire you so that you'll feel comfortable creating your own dream projects in robotics.

But let's not get ahead of ourselves. In this book, I'm going to assume quite a bit of knowledge of not only the BeagleBone Black, but also programming in general and specifically Linux. If you're truly a beginner, you'll have to start with my other book on the BeagleBone Black, *BeagleBone Robotic Projects*, Packt Publishing. However, I'm not going to assume that you've read the other book, so this first chapter will lead you through, in an accelerated way, the steps you'll need to follow from unpacking to having a BeagleBone Black that is configured to be successful in building complex robotics projects. To understand this book, you'll also need a bit more knowledge of how to connect electronic devices. You'll use the **General-Purpose Input/Output (GPIO)** pins and even a soldering iron to build these robots.

In this chapter, you will learn about the following:

- Installing an operating system
- Installing a vision library and sound capabilities
- Creating and recording sounds
- Making your BeagleBone speak
- Installing speech recognition
- Improving accuracy of speech recognition
- Adding additional hardware and software for a fully functional core system

So let's start with unpacking our BeagleBone Black. For this chapter, you'll need the following hardware:

- A BeagleBone Black board
- A BeagleBone Black USB power cable
- A LAN cable
- A USB sound device that has a plug for a microphone and a speaker
- A microphone
- A speaker
- A USB Wi-Fi dongle (this is optional; needed if you want to communicate with the BeagleBone Black via a wireless LAN).

Unpacking and powering up

Let's unpack the single-board computer hardware device. Your BeagleBone Black will come in a box—in a static protection bag, with a USB cable that can power the unit from a USB connection. It should look somewhat like this:



The USB connection not only supplies power to the host computer, but also provides a simple, yet fairly limited, way to communicate with the board. You'll use this USB cable for power, but that's about it.



You can also choose to power the BeagleBone Black using the 5 V barrel jack. One that works well is from Adafruit, available at <https://www.adafruit.com/products/276>.

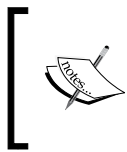
If you've never worked with the BeagleBone Black before, the `beagleboard.org` site can lead you through the first few steps, from powering on to making sure your board is up and working, and to some initial communications with the board.

You must know that there are several versions of the board now. The latest, at the time of the writing of this book, is version C. The most significant change from Version B to Version C is that Version C has a 4 GB eMMC disk. Even this might be a bit small for your projects, and you're going to use an 8 GB to 16 GB micro SD card anyway, so either version should be fine.

Installing an operating system

The first thing you need to do is install a useful operating system into an 8 GB, 16 GB, or 32 GB micro SD card—just a little background for this. Initially, the default operating system on the internal memory was a version of Linux called **Ångström**. Version C is now shipped with a version of **Debian**. You're going to move to a different operating system called **Ubuntu**, a close relative of Debian, but with a larger feature set and community of support. It'll make the projects you are about to tackle much easier.

Now the BeagleBone Black is getting popular enough, so you can buy a micro SD card that already has Ubuntu installed. Nevertheless, you can easily download it and then install it on the card. If you're going to download a distribution, you'll need to make sure you're getting the latest version of Ubuntu.




The easiest way to download is to go to one of the several sites that have an image you can put on your card. One is at `www.armhf.com/boards/beaglebone-black/#saucy` and another is at `elinux.org/Beagleboard:Ubuntu_On_BeagleBone_Black`.

Both of these sites come with instructions for building a card with the image on it. You'll need to use the username and password of the image you downloaded. Unfortunately, they are not the same for all images, but you should be able to easily find them in the same place where you found your image.

Connecting to an external computer

In this book, I'm going to assume that you'll be developing your code on an external computer while you create your projects. I'll primarily be using a Windows PC for development, but everything I'll be doing will also be available if you are using a Linux machine as the host computer. Whenever there are any significant differences between the two, I'll try to include details for both.

To do this development remotely, you'll need access to the BeagleBone Black via some type of LAN connection. To establish this connection, simply connect a LAN cable from a router or switch to the BeagleBone board, plug it into the LAN connector, and restart the BeagleBone.

 You can also use an FTDI console cable like the ones sold at <https://www.adafruit.com/products/954>, which are connected to the BeagleBone Black's J1 console UART pins. Refer the following link for details about setup: http://elinux.org/Beagleboard:BeagleBone_Black_Accessories

You'll need to know the IP address of your BeagleBone Black. You can certainly get this by adding a display and keyboard to the device, logging in, and typing `ifconfig` in the prompt. However, if you want to do this without rounding up the additional hardware, then you can use an IP scan tool to scan for the address of the BeagleBone Black. I used a tool called **Advanced IP Scanner**, but there are others available too. If you're using Linux as a host machine, you can try the `nmap` command in Linux. For example, you can type `sudo nmap -PR -sP 192.168.1.0/24`.

These will explore your network and then print the address where there are devices, the BeagleBone Black being one of those devices. Generally, there are two types of IP addresses that your board can be assigned: one is called **static** and the other is called **dynamic**.

In the static case, you will always be assigned the same address. In the dynamic case, the address might change each time the system boots, as it asks the system for an address, which it then uses for that session. Most systems are configured for the dynamic case. However, if your system isn't changing the address, you will most likely get the same address each time you power on and log in to the system.



To learn more about DHCP, visit www.teracomtraining.com/tutorials/teracom-tutorial-dynamic-IP-addresses-and-DHCP.htm.

Once you have the address, you should create an SSH connection to the BeagleBone Black. An **SSH** terminal is a **Secure Shell Hypertextual** connection, which simply means that you'll be able to access your board and type commands in the prompt, just like you have done without the Windows system. In order to do this, you need to have a terminal emulator on your remote computer. For Microsoft Windows, I would suggest you to use an application called **PuTTY**. It is free and does a very good job by allowing you to save your configuration, so you don't have to type it every time. You can go to www.putty.org for details on how to download and use PuTTY.

If you want to do this from a Linux machine, the process is even simpler. Open a terminal window and then type `ssh ubuntu@157.201.194.187 -p 22`. This will then bring you to the login screen of your BeagleBone Black.

SSH is an excellent way to communicate with your BeagleBone Black. However, you'll sometimes need a graphical look at your system. You can get this by installing a graphical interface on the BeagleBone Black and then using an application called **VNC server**.

Installing a Windows manager

First, let's install the Windows manager on your BeagleBone Black. Ubuntu generally comes with a very full-featured Windows system. However, it uses a good deal of memory and can interfere with the performance you might need later. So you should install a *light* Windows system at the top of your Ubuntu distribution. There are several choices, of which I like using **Xfce**. It is stable, seems to work well, and offers a fairly complete set of capabilities while not overwhelming your system resources. Before getting started, first type `sudo apt-get update`. This will update all the latest information about installation. To install the Windows manager, type `sudo apt-get install xfce4`. Again, the system will prompt you for your password and then start the installation. For large installations, the Linux system will ask you if you want to proceed. If you don't always want to type in `y` to this, you can use `apt-get -y` to automatically answer with a yes. This installation will take a significant amount of time as it has to install not only the Windows system, but also a number of packages the windowing system depends on.