

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

# Raspberry Pi Robotics Projects

## *Second Edition*

Get the most out of Raspberry Pi to build enthralling robotics projects

**Richard Grimmett**

**[PACKT]** open source\*  
PUBLISHING community experience distilled

# Raspberry Pi Robotics Projects

## *Second Edition*

Get the most out of Raspberry Pi to build enthralling robotics projects

**Richard Grimmett**



BIRMINGHAM - MUMBAI

# Raspberry Pi Robotics Projects

*Second Edition*

Copyright © 2015 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 authors, 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: February 2014

Second edition: April 2015

Production reference: 1270415

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

ISBN 978-1-78528-014-6

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

# Credits

**Author**

Richard Grimmett

**Project Coordinator**

Kranti Berde

**Reviewer**

Werner Ziegelwanger

**Proofreaders**

Safis Editing

Maria Gould

**Commissioning Editor**

Neil Alexander

**Indexer**

Monica Ajmera Mehta

**Acquisition Editor**

Tushar Gupta

**Production Coordinator**

Conidon Miranda

**Content Development Editor**

Vaibhav Pawar

**Cover Work**

Conidon Miranda

**Technical Editor**

Saurabh Malhotra

**Copy Editors**

Dipti Kapadia

Tani Kothari

Sonia Mathur

Karuna Narayanan

Kriti Sharma

Alpha Singh

# About the Author

**Richard Grimmert** has been fascinated by computers and electronics since his very first programming project that used Fortran on punched cards. He has a bachelor's degree and a master's degree in electrical engineering, and a PhD in leadership studies. He 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 his many robotics projects.

---

This book is the result of working with many wonderful students at Brigham Young University-Idaho. Also, it wouldn't have been possible without the help of my wonderful wife, Jeanne.

---

# About the Reviewer

**Werner Ziegelwanger**, MSc, has studied game engineering and simulation. He got his master's degree in 2011, and his master's thesis was titled *Terrain Rendering with Geometry Clipmaps*, Diplomica Verlag. His hobbies are programming, playing games, and all kinds of technical gadgets.

Werner has worked as a self-employed programmer for a few years, where he mainly did web projects. During this time, he started his own blog (<http://developer-blog.net>), which is about the Raspberry Pi, Linux, and open source.

Since 2013, Werner has been working as a Magento developer and is the head of programming at mStage GmbH, an e-commerce company that focuses on Magento.

# www.PacktPub.com

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

For support files and downloads related to your book, please visit [www.PacktPub.com](http://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](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.



<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](http://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

<b>Preface</b>	<b>v</b>
<b>Chapter 1: Getting Started with Raspberry Pi</b>	<b>1</b>
<b>Getting started</b>	<b>1</b>
The unboxing	3
<b>Powering your board</b>	<b>5</b>
<b>Hooking up a keyboard, mouse, and display</b>	<b>6</b>
<b>Installing the operating system</b>	<b>8</b>
<b>Accessing the board remotely</b>	<b>18</b>
Establishing Internet access on Raspberry Pi B+	18
Establishing Internet access on Raspberry Pi A+	18
Accessing your Raspberry Pi from your host PC	20
<b>Summary</b>	<b>34</b>
<b>Chapter 2: Programming Raspberry Pi</b>	<b>35</b>
<b>Basic Linux commands on Raspberry Pi</b>	<b>36</b>
<b>Creating, editing, and saving files on Raspberry Pi</b>	<b>41</b>
<b>Creating and running Python programs</b>	<b>43</b>
<b>Basic programming constructs on Raspberry Pi</b>	<b>46</b>
The if statement	46
The while statement	48
Working with functions	49
Libraries/modules in Python	51
Object-oriented code	52
<b>Introduction to the C/C++ programming language</b>	<b>55</b>
<b>Summary</b>	<b>59</b>



---

<b>Chapter 3: Providing Speech Input and Output</b>	<b>61</b>
Hooking up the hardware to make and input sound	63
Using Espeak to allow our projects to respond in a robotic voice	71
Using PocketSphinx to accept your voice commands	73
Interpreting commands and initiating actions	80
Summary	83
<b>Chapter 4: Adding Vision to Raspberry Pi</b>	<b>85</b>
Connecting the USB camera to Raspberry Pi and viewing the images	86
Connecting the Raspberry Pi camera board and viewing the images	89
Downloading and installing OpenCV – a fully featured vision library	93
Using the vision library to detect colored objects	98
Summary	104
<b>Chapter 5: Creating Mobile Robots on Wheels</b>	<b>105</b>
Gathering the required hardware	105
Using the Raspberry Pi GPIO to control a DC motor	108
Controlling your mobile platform programmatically using Raspberry Pi	111
Controlling the speed of your motors with PWM	114
Adding program arguments to control your platform	117
Making your platform truly mobile by issuing voice commands	119
Summary	122
<b>Chapter 6: Controlling the Movement of a Robot with Legs</b>	<b>123</b>
Gathering the hardware	124
Connecting Raspberry Pi to the mobile platform using a servo controller	127
Connecting the hardware	127
Configuring the software	129
Creating a program in Linux to control the mobile platform	135
Making your mobile platform truly mobile by issuing voice commands	138
Summary	140
<b>Chapter 7: Avoiding Obstacles Using Sensors</b>	<b>141</b>
Connecting Raspberry Pi to an infrared sensor using USB	146
Connecting a sensor using the USB interface	147
Connecting the IR sensor using the GPIO ADC	155

---

<b>Connecting Raspberry Pi to a USB sonar sensor</b>	<b>162</b>
Connecting the hardware	164
<b>Using a servo to move a single sensor</b>	<b>169</b>
<b>Summary</b>	<b>173</b>
<b>Chapter 8: Going Truly Mobile – The Remote Control of Your Robot</b>	<b>175</b>
<b>Gathering the hardware</b>	<b>176</b>
<b>Connecting Raspberry Pi to a wireless USB keyboard</b>	<b>183</b>
<b>Using the keyboard to control your project</b>	<b>183</b>
<b>Working remotely with your Raspberry Pi through a wireless LAN</b>	<b>189</b>
<b>Working remotely with your Raspberry Pi through ZigBee</b>	<b>194</b>
<b>Summary</b>	<b>203</b>
<b>Chapter 9: Using a GPS Receiver to Locate Your Robot</b>	<b>205</b>
<b>Connecting Raspberry Pi to a USB GPS device</b>	<b>207</b>
<b>Accessing the USB GPS programmatically</b>	<b>218</b>
<b>Connecting Raspberry Pi to an RX/TX (UART) GPS device</b>	<b>224</b>
<b>Communicating with the RX/TX GPS programmatically</b>	<b>225</b>
<b>Taking a look at the GPS data</b>	<b>229</b>
<b>Summary</b>	<b>238</b>
<b>Chapter 10: System Dynamics</b>	<b>239</b>
<b>Creating a general control structure</b>	<b>241</b>
<b>Using the structure of the Robot Operating System to enable complex functionalities</b>	<b>253</b>
<b>Summary</b>	<b>256</b>
<b>Chapter 11: By Land, Sea, and Air</b>	<b>257</b>
<b>Using Raspberry Pi to sail</b>	<b>258</b>
Getting started	258
<b>Using Raspberry Pi to fly robots</b>	<b>265</b>
<b>Using Raspberry Pi to make the robot swim underwater</b>	<b>275</b>
<b>Summary</b>	<b>276</b>
<b>Index</b>	<b>277</b>

---



# Preface

Robots seem to be everywhere these days. Not only in movies but also in our day-to-day lives. They vacuum floors, play with children, and build automobiles. These new machines are quickly migrating from university and government labs to our homes, offices, schools, and playgrounds. A similar migration has occurred in recent years with computers.

A good part of this migration, as in the case of computers, is that the untrained but interested general population has been able to take part in the development of these new machines. A big reason for this is the introduction of inexpensive hardware and free, open source hardware. Initially, it was Arduino, an inexpensive processor developed for the do-it-yourself crowd, which enabled normal people to create robotics projects with complex functionality. More recently, Raspberry Pi has extended this capability by providing an inexpensive, small, but powerful Linux computer for these projects.

This second edition is designed to allow you to take full advantage of the capabilities of Raspberry Pi in your robotics projects. It will not only teach you how to use the USB port to add additional hardware capabilities, but it will also show you how to connect the hardware to the GPIO as well. It will provide a step-by-step guide to get you well on your way to building your very own amazing robotics projects.

## What this book covers

*Chapter 1, Getting Started with Raspberry Pi*, helps you to power up your Raspberry Pi, connect it to a keyboard, mouse, display, and remote computer, and begin to access all that potential computing power.

*Chapter 2, Programming Raspberry Pi*, teaches you the basics of how to program the Raspberry Pi, both in Python and in the C programming languages.

*Chapter 3, Providing Speech Input and Output*, teaches your Raspberry Pi to both speak and listen.

*Chapter 4, Adding Vision to Raspberry Pi*, shows you how to use standard USB and Raspberry Pi cameras to allow your robotics projects to see.

*Chapter 5, Creating Mobile Robots on Wheels*, shows you how to connect the Raspberry Pi to a mobile wheeled platform and control its motors, so your robots can be mobile.

*Chapter 6, Controlling the Movement of a Robot with Legs*, teaches you how to make your robot walk.

*Chapter 7, Avoiding Obstacles Using Sensors*, shows you how to sense the world around you. Now that your robot is mobile, you'll want to avoid or find objects.

*Chapter 8, Going Truly Mobile – The Remote Control of Your Robot*, shows you how to control your robot wirelessly – you'll want your robot to move around untethered by cables.

*Chapter 9, Using a GPS Receiver to Locate Your Robot*, shows you how to use a GPS receiver so that your robot knows where it is – if your robot is mobile, it might get lost.

*Chapter 10, System Dynamics*, focuses on how to bring it all together to make complex robots since you've got lots of capability.

*Chapter 11, By Land, Sea, and Air*, shows you how to add capabilities to robots that sail, fly, and even go under the water.

## What you need for this book

Here's a partial list of the software you will need for the book:

- 7-Zip: This is a utility to archive and unarchive software
- Image Writer for Windows: This is used to write images to an SD card
- WinSCP: This provides the ability to transfer files to/from a PC
- PuTTY: This allows the user remote access to the Raspberry Pi
- VNC Server/VNC Viewer: This allows the user remote access to the graphical interface of the Raspberry Pi

## Who this book is for

This book is for anyone who is keen on using the Raspberry Pi to create robotics projects that have previously been the domain of the research labs of major universities or defense departments. Some programming background is useful, but if you know how to use a personal computer you can, with the aid of the step-by-step instructions in this book, construct complex robotics projects that can move, talk, listen, see, swim, or fly.

## 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: "Go to the `/home/Raspbian/Desktop` directory."

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
a = input("Input value: ")
b = input("Input second value: ")
c = a + b
print c
```

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

```
cd /home/Raspbian/Desktop
```

**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 **Raspbian** section and select the `.zip` file just to the right of the image identifier."



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 color images of this book

We also provide you with 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 [http://www.packtpub.com/sites/default/files/downloads/01460S\\_ColoredImages.pdf](http://www.packtpub.com/sites/default/files/downloads/01460S_ColoredImages.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](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 Raspberry Pi

Raspberry Pi, with its low cost and amazing package of functionality, has taken the robotic hobbyist community by Storm. Unfortunately, many, especially those new to embedded systems and programming, can end up so discouraged that the board can end up on the shelf, gathering dust next to the floppy disks and your old CRT.

### Getting started

There is nothing as exciting as ordering, and finally receiving a new piece of hardware. Yet things can go south quickly, even in the first few minutes. This chapter will, hopefully, help you avoid the pitfalls that normally accompany unpacking and configuring your Raspberry Pi. We'll walk through the process, answer many of the different questions you might have, and help you understand what is going on. If you don't go through this chapter, you'll not be successful at any of the others, and your hardware will go unused, which would be a real tragedy. So, let's get started.

One of the most challenging aspects of writing this guide is to decide the level at which I should describe each step. Some of you are beginners, some have limited experience, and others know significantly more in some of these areas. I'll try to be brief but thorough, trying to detail the steps to take in order to be successful. So, for this chapter, here are your objectives:

- Unbox and connect the board to power
- Connect a display, keyboard, and mouse
- Load and configure the operating system
- Access the board remotely

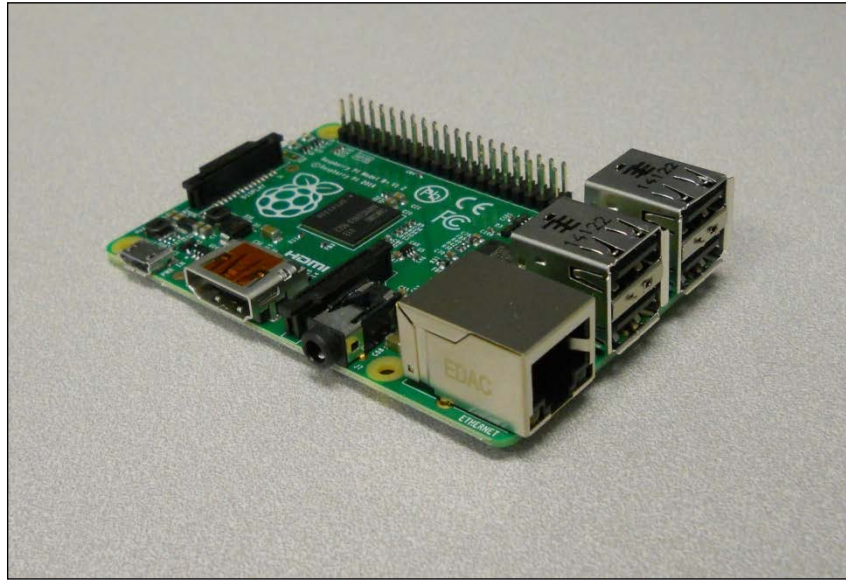
The Raspberry Pi comes in several flavors: the original A and B models, and the new and improved A+ and B+ models. The B+ flavor is the most popular. It comes with additional input/output capability, four USB connections, more memory, and will be the flavor you'll focus on in this book. That does not mean that many, if not most, of the projects here require the extra capability of the B+. As we go, I'll try and point out when you'll need the additional capability of the B+, and when the Raspberry Pi A+ might be enough. The Raspberry Pi now also comes in a B2 version. We'll not talk about that specifically here, but there is no reason why the projects shouldn't work with that version as well.

Here are the items you'll need for this chapter's projects:

- A Raspberry Pi, Model B+
- The USB cable to provide power to the board
- A display with a proper video input
- A keyboard, a mouse, and a powered USB hub
- A microSD card – at least 4 GB capacity
- A microSD card writer
- Another computer that is connected to the Internet
- An Internet connection for the board
- A LAN cable (if you are using the Raspberry Pi A+ you'll need a powered USB hub, a wireless LAN connection, and a Wireless LAN device; you'll learn how to configure this later in the chapter)

## The unboxing

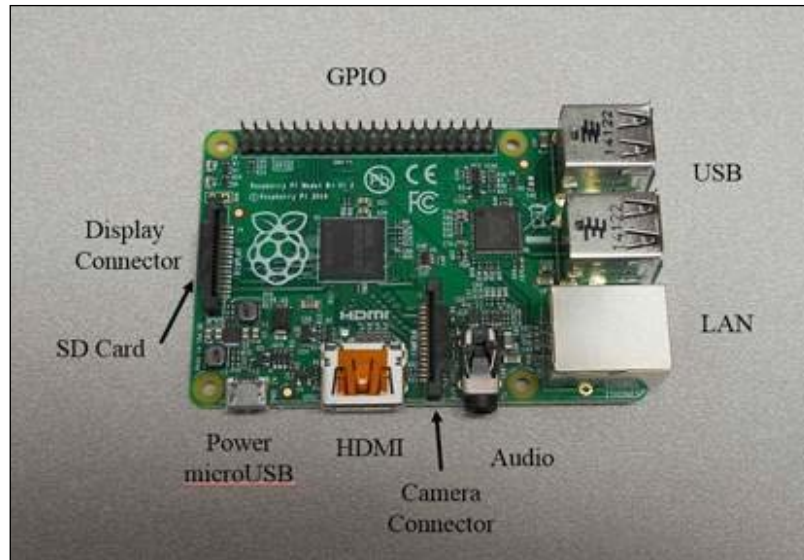
The first step to building any project with Raspberry Pi is to become familiar with Raspberry Pi itself. The Raspberry Pi comes in a box with a power cable. The following image shows what the Raspberry Pi B+ board looks like:



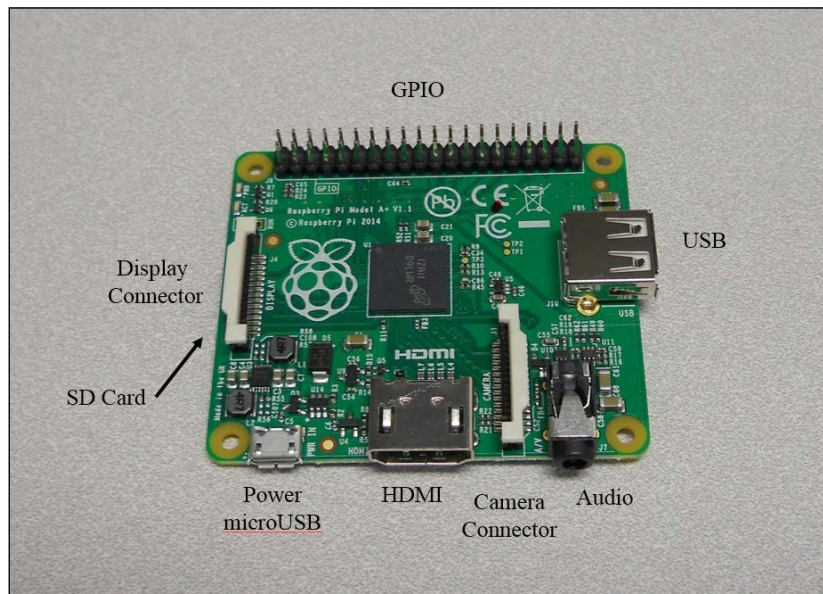
And the next image is that of the Raspberry Pi A+ board:



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 familiarize yourself with the different connections on the board. Here is the B+ board, labeled for your information:



The labels for the A+ board are shown in the following image:



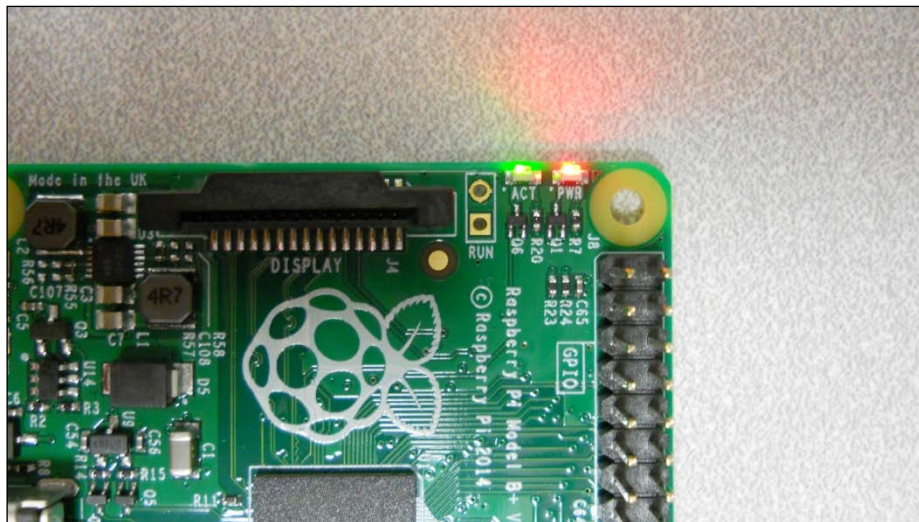
## Powering your board

Let's first power the board. To do this, you'll need to go through the USB client connection. This is done by performing the following steps:

1. Connecting the microUSB connector end of the cable to the board.
2. Connecting the standard sized USB connector to either a PC, or a compatible DC power source that has a USB connection. If you are connecting it to a PC, the USB 2.0 standard requires only 500 mA be available. The USB 3.0 standard requires only 900 mA be available. This means that some computers may not supply enough current for the Raspberry Pi to work correctly.

If you are going to use a DC power source at the standard USB connector end, make sure that the unit can supply enough current. You'll need a supply that can provide at least 1000 mA at 5 volts.

When you plug the board in, the **PWR** LED should be red. Here is a close up of the LED locations, just so you're certain which one to look for:



If you've reached this point, congratulations! You're ready for the next step.

## Hooking up a keyboard, mouse, and display

Now that you know your board works, you're going to add peripherals so that it can operate as a standalone computer system. This step is optional, as in the future, your projects will often be in systems where you won't connect directly to the board with a keyboard, mouse, and display. However, this can be a great learning step, and is especially useful if you need to do some debugging on the system.

You'll need the following peripherals:

- A USB mouse
- A USB keyboard (this can be wireless, and can contain a built-in mouse pad)
- A display that accepts HDMI or DVI-Video, although using a DVI input will require an adapter
- A powered USB hub (you will need this if you are going to use the Raspberry Pi A+ version for the projects)

You may have most of this stuff already, but if you don't, there are some things to consider before buying additional equipment. Let's start with the keyboard and mouse. Most mice and keyboards have separate USB connectors. This normally works fine for the Raspberry Pi model B+, as it has four USB ports. If you are going to use the Raspberry Pi A+ model, you may want to choose a keyboard that has a mouse pad built-in.

If you are using the Raspberry Pi A+ model, you will want to consider purchasing a powered USB hub. Before deciding on the hub to connect to your board, you need to understand the difference between a powered USB hub and one that gets its power from the USB port itself. Almost all USB hubs are not powered, that is, you don't plug in the USB hub separately. The reason for this is that almost all of these hubs are hooked up to computers with very large power supplies and powering USB devices from the computer is not a problem. This is not the case for your board. The USB port on your board has very limited power capabilities, so if you are going to hook up devices that require significant power—a WAN adapter or webcam for instance—you're going to need a powered USB hub, one that provides power to the devices through a separate power source. Here is an image of such a device, available at [www.amazon.com](http://www.amazon.com) and other online retailers:





Notice that on this hub, there are two connections. The one to the far right is a power connection, and it will be plugged into a battery with a USB port. The connection to the left is the USB connection, which will be plugged into the Raspberry Pi.

Now, you'll also need a display. Fortunately, your Raspberry Pi offers lots of choices here. There are a number of different video standards; here is an image of some of the most prevalent ones, for reference:





There is an HDMI connector on the Raspberry Pi A+ and B+. To use this connector, simply connect your cable with regular HDMI connections to Raspberry Pi and your TV or monitor that has an HDMI input connector. HDMI monitors are relatively new, but if you have a monitor that has a DVI input, you can buy relatively inexpensive adapters that provide an interface between DVI and HDMI.

Don't be fooled by adapters that claim that they go from HDMI or DVI to VGA, or HDMI or DVI to S-video. These are two different kinds of signals: HDMI and DVI are digital standards, and VGA and S-video are analog standards. There are adapters that can do this, but they must contain circuitry and require power, and are significantly more expensive than any simple adapter.

You are almost ready to plug in the Raspberry Pi. Make sure you connect all your devices before you power on the unit. Most operating systems support the hot-swap of devices, which means you are able to connect a device after the system has been powered, but this is a bit shaky. You should always cycle power when you connect new hardware.

Even though your hardware configuration is complete, you'll still need to complete the next section to power on the device. So, let's figure out how to install an operating system.

## **Installing the operating system**

Now that your hardware is ready, you need to install an operating system. You are going to install Linux, an open-source version of Unix, on your Raspberry Pi. Now Linux, unlike Windows, Android, or iOS, is not tightly controlled by a single company. It is a group effort, mostly open-source, and while it is available for free, it grows and develops a bit more chaotically.

Thus, a number of distributions have emerged, each built on a core set of similar capabilities referred to as the Linux kernel. These core capabilities are all based on the Linux specification. However, they are packaged slightly differently, and developed, supported, and packaged by different organizations. Debian, Arch, and Fedora are the names of some of the versions. There are others as well, but these are the main choices for the distribution that you might put on your card.

I choose to use the Raspbian, a Debian distribution of Linux, on my Raspberry Pi projects for a couple of reasons. First, the Debian distribution is used as the basis for another distribution, Ubuntu, and Ubuntu is arguably the most popular distribution of Linux, which makes it a good choice because of the community support that it offers. Also, I personally use Ubuntu when I need to run Linux on my personal computer. It provides a complete set of features, is well organized, and generally supports the latest sets of hardware and software. Having roughly the same version on both my personal computer and my Raspberry Pi makes it easier for me to use both as they operate, at least to a certain degree, the same way. That lets me try some things on my computer before trying them on the Raspberry Pi. I've also found that Ubuntu/Debian has excellent support for new hardware, and this can be very important for your projects.

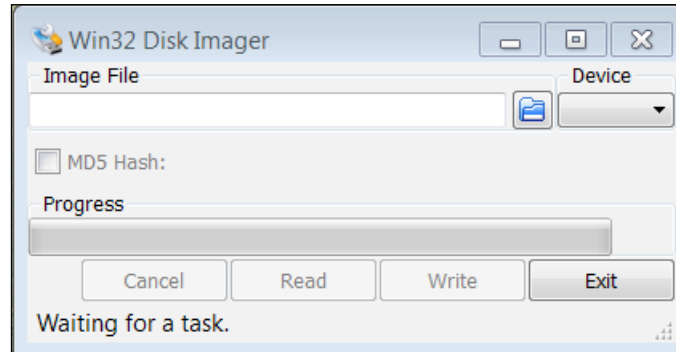
So, you are going to install and run Raspbian, on your Raspberry Pi.

There are two approaches to getting Raspbian on your board. The board is getting popular enough for you to buy an SD card that already has Raspbian installed, or you can download it onto your personal computer and then install it on the card. I'll assume you don't need any directions if you want to purchase a card – simply do an Internet search for companies selling such a product.

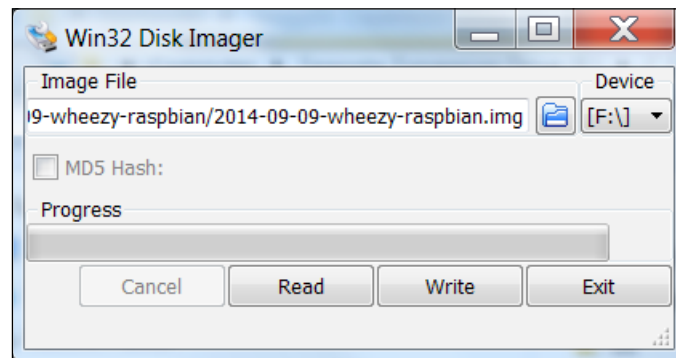
If you are going to download a distribution, you need to decide if you are going to use a Windows computer to download and create an SD card, a MAC OS X, or a Linux machine. I'll give brief directions for the Windows computer and Linux machines here. For directions on the MAC OS X, go to: <http://www.raspberrypi.org/documentation/installation/installing-images/mac.md>.

First, you'll need to download an image. This part of the process is similar for both Windows and Linux. Open a browser window. Go to Raspberry Pi organization's website, [www.raspberrypi.org](http://www.raspberrypi.org) and select the **Downloads** selection at the top of the page. This will give you a variety of download choices. Go to the **Raspbian** section, and select the .zip file just to the right of the image identifier. This will download an archived file that has the image for your Raspbian operating system. Note the default user name and password; you'll need that later.

If you're using Windows, you'll need to unzip the file using an archiving program like 7-Zip. This will leave you with a file that has the `.img` extension, a file that can be imaged on to your card. Next, you'll need a program that can write the image to the card. I use the Image Writer for Windows program. You can find a link to this program at the top of the download section on the [www.raspberrypi.org](http://www.raspberrypi.org) website. Plug your card into the PC, run this program, and you should see this:

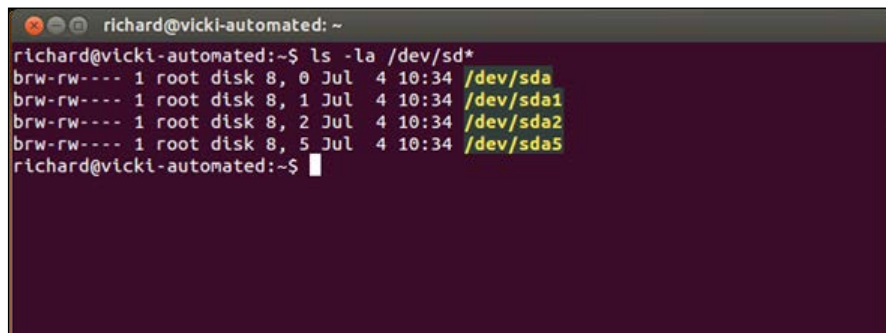


Select the correct card and image; it should look something like this:



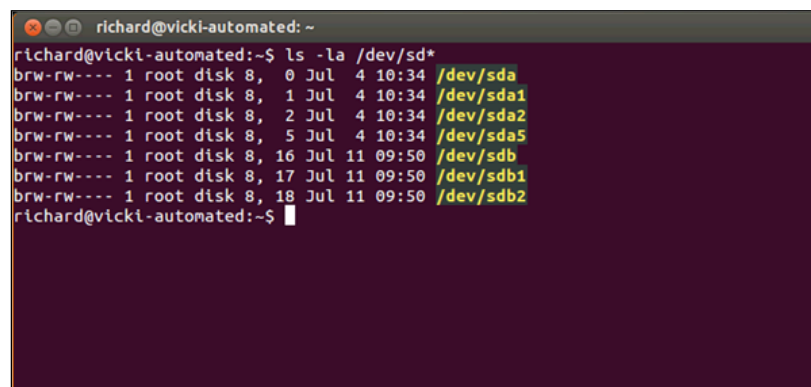
Then click on the **Write** button. This will take some time, but when it is complete, eject the card from the PC.

If you are using Linux, you'll need to un-archive the file and then write it to the card. You can do all of this with one command. However, you do need to find the `/dev` device label for your card. You can do this with the `ls -la /dev/sd*` command. If you run this before you plug in your card, you might see something like the following screenshot:



```
richard@vicki-automated: ~  
richard@vicki-automated:~$ ls -la /dev/sd*  
brw-rw---- 1 root disk 8, 0 Jul 4 10:34 /dev/sda  
brw-rw---- 1 root disk 8, 1 Jul 4 10:34 /dev/sda1  
brw-rw---- 1 root disk 8, 2 Jul 4 10:34 /dev/sda2  
brw-rw---- 1 root disk 8, 5 Jul 4 10:34 /dev/sda5  
richard@vicki-automated:~$
```

After plugging in your card, you might see something like the following screenshot:



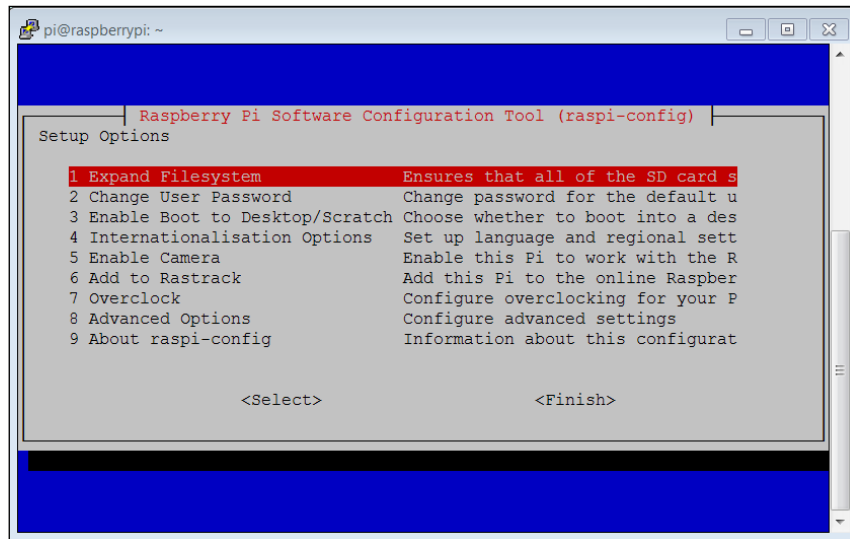
```
richard@vicki-automated: ~  
richard@vicki-automated:~$ ls -la /dev/sd*  
brw-rw---- 1 root disk 8, 0 Jul 4 10:34 /dev/sda  
brw-rw---- 1 root disk 8, 1 Jul 4 10:34 /dev/sda1  
brw-rw---- 1 root disk 8, 2 Jul 4 10:34 /dev/sda2  
brw-rw---- 1 root disk 8, 5 Jul 4 10:34 /dev/sda5  
brw-rw---- 1 root disk 8, 16 Jul 11 09:50 /dev/sdb  
brw-rw---- 1 root disk 8, 17 Jul 11 09:50 /dev/sdb1  
brw-rw---- 1 root disk 8, 18 Jul 11 09:50 /dev/sdb2  
richard@vicki-automated:~$
```

Note that your card is at `sdb`. Now, go to the directory where you downloaded the archived image file, and use the following command:

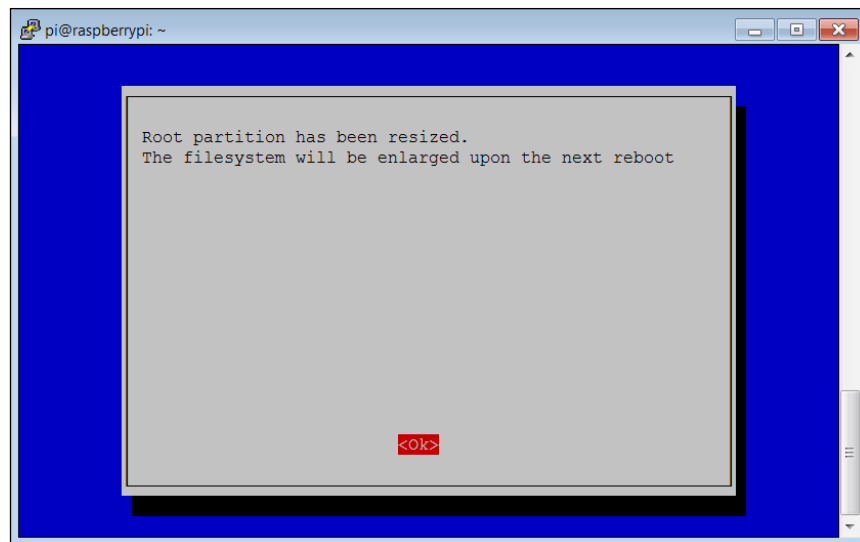
```
sudo dd if=2014-09-09-wheezy-raspbian.img of=/dev/sdX
```

The `2014-09-09-wheezy-raspbian.img` command will be replaced by the image file that you downloaded, and `/dev/sdX` will be replaced by your card ID in this example, `/dev/sdb`. Be careful to specify the correct device as this can overwrite the data on your hard drive. Also, this may take a few minutes. Once the file is written, eject the card and you are ready to plug it into the board and apply power.

Make sure your Raspberry Pi is unplugged and install the SD card into the slot. Then power the device. After the device boots, you should get the following screenshot:



You are going to do two things, and perhaps another, based on your personal preference. First, you'll want to expand the file system to take up the entire card. So, hit the *Enter* key, and you'll see the following screenshot:



Hit *Enter* once again and you'll go back to the main configuration screen. Now select the **Enable Boot to Desktop/Scratch** option, as shown in the following screenshot:

