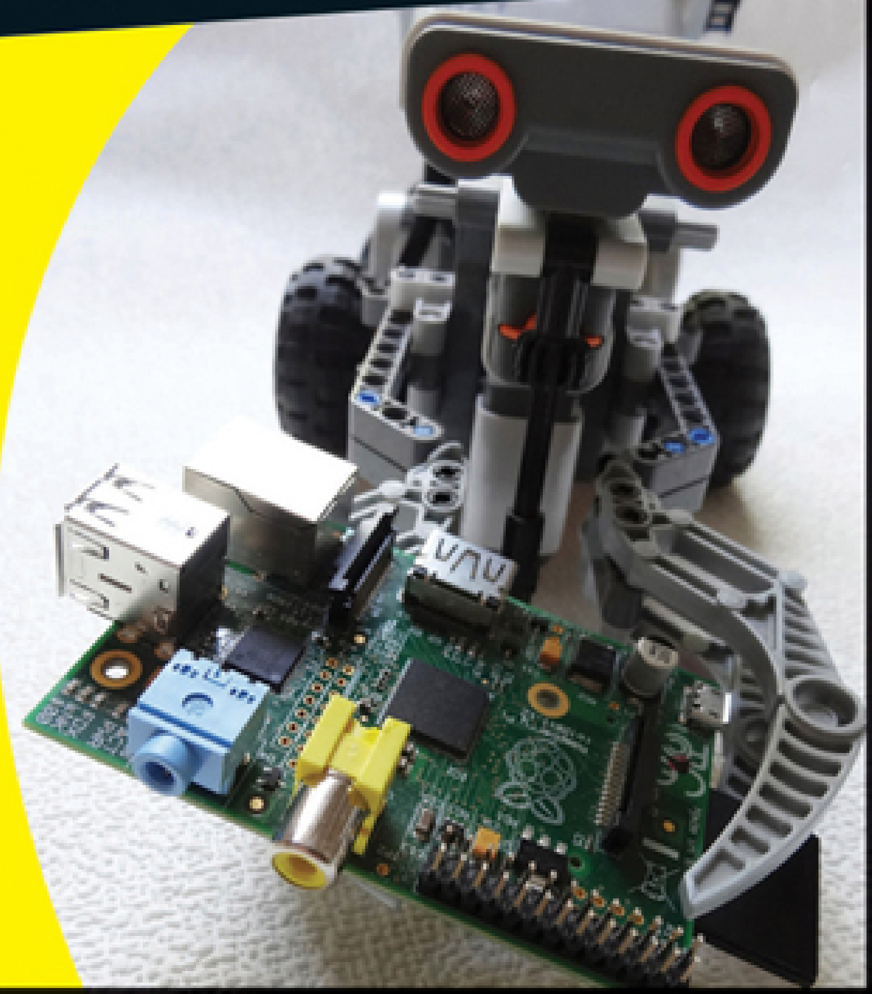
Making Everything Easier!™

Raspberry Pi Projects FOR DUMMIES®

A Wiley Brand



Learn to:

- Have fun with LEDs and create a magic light wand
- Enhance the Pi's capabilities with Lego™ and directly access Lego sensors
- Install and write games for the RISC OS
- Build a transistor tester and learn to talk with Arduino

Mike Cook

Introduction

Raspberry Pi Projects For Dummies is designed for people who are looking for something exciting to do with the Raspberry Pi. This book contains projects to amaze and inspire you! It takes you into a world of switches, lights, motors, home automation, and computer vision. It not only covers the theory behind what you're doing, but also gives you examples of putting that theory into practice, so you can learn to work on your own projects and not just blindly follow a list of instructions.

Sure, we could just give you a list of steps to follow. But we believe that you should try to understand what you're doing and why you're doing it, and that's what this book is all about. With this book as a resource, we encourage you to put your own stamp on projects, which is why many projects in this book aren't just cut-and-dried lists of things to do, but suggestions about how you can customize the projects and make them your own.

About This Book

The projects in this book all make use of the computer language Python 2. This book shows you how to use a wide variety of input and output devices, from a simple switch to a webcam. You can explore LEDs and multicolored LEDs, learn about a keypad matrix and see how they can be integrated to become part of your code so you can make these devices do what you want. Reach out with your Raspberry Pi and become part of the cloud or build your own web server. This book shows you how.

The Raspberry Pi can interface with other electronic devices, and in this book we show you how to interact with LEGO's latest robotic MINDSTORMS set, the EV3. You can send messages into the LEGO system or do your own thing and control the MINDSTORMS peripherals directly from the Raspberry Pi. Not only do we show you how these two systems interact, but we also show you some projects you can make using the Raspberry Pi and MINDSTORMS set together.

Linux is the staple operating system used in the Raspberry Pi world. However, there is a major alternative operating system you can run for just the price of another very small SD card, RISC OS. RISC OS is a mature, well-honed operating system, designed from the ground up to run on ARM chips, and as such, it's fast and compact. This book shows you how you can explore the RISC OS and gives you a glimpse of another world.

A few final notes about the book: Sidebars (text in gray boxes) and Technical Stuff paragraphs are skippable. Finally, within this book, you may note that some web addresses break across two lines of text. If you're reading this book in print and want to visit one of these web pages, simply key in the web address exactly as it's noted in the text, pretending as though the line break doesn't exist. If you're reading this as an e-book, you've got it easy — just click the web address to be taken directly to the web page.

Foolish Assumptions

In writing this book, we made a few assumptions about you:

- **You have a Raspberry Pi.** You could certainly read this book without a Raspberry Pi, but you won't get much out of it unless you have a Raspberry Pi to play with.

- **You have a computer other than the Raspberry Pi.** You need a computer to set up the Raspberry Pi. ***Note:*** We provide instructions on how to set up your Pi, but this information isn't the main thrust of the book. If you need more information on setting up your Raspberry Pi, a good companion book to this one is *Raspberry Pi For Dummies,* by Sean McManus and Mike Cook (Wiley), which covers in much more detail your first steps with this remarkable machine.
- **Your Raspberry Pi has some connection to the Internet.** It may not be connected all the time, but you're at least able to connect it for setting up the libraries you need to install.
- **You don't mind voyaging into less charted waters and you have an open mind on what constitutes computing and operating systems.**
- **You're eager to begin exploring the world of physical computing.** Physical computing takes a fresh look at inputs and outputs to a computer. The computer produces physical outputs — signals that make lights flash, sounds play, or robots move. Inputs are more than just typing — they include everything from simple push buttons to color sensors to webcams.
- **You have access to some basic hand tools, like a small saw and drill along with a soldering iron.** If you don't have these tools on hand, we assume you have the money to buy them — or you have a friend or family member whose toolkit you can raid!
- **You don't mind spending some money on the components you need to make your projects.** Most of these components aren't very expensive, but you'll need to buy them (and we recommend sources in this book).

Icons Used in This Book

In this book, we use a handful of *icons* (little pictures in the margins) to draw your attention to key pieces of information. Here’s what those icons mean:

TIP

When we give you an especially useful bit of information — something that can make your life with the Raspberry Pi easier or help you do something faster — we mark it with the Tip icon.

REMEMBER

You don’t need to commit this book to memory — it’s a resource for you to turn to whenever you need it. But every once in a while, we tell you something so important that you’ll want to remember it. When we do, we mark it with the Remember icon.

TECHNICAL STUFF

What can we say? We’re geeks. And as such, we sometimes get a little technical, telling you more than you really need to know to get the job done. When we veer into the technical, we mark that text with the Technical Stuff icon. If you’re short on time, you can skip anything marked with this icon without missing anything critical to the task at hand.

WARNING!

You’re bound to come across some pitfalls on your journey with the Raspberry Pi. We’ve walked this road before, so think of the Warning icon as orange cones in the road, helping you steer clear of those tire-destroying potholes or open manhole covers.

Beyond the Book

In addition to the material in the print or e-book you're reading right now, this product also comes with some access-anywhere goodies on the web. Check out the free Cheat Sheet at `www.dummies.com/cheatsheet/raspberrypiprojects` for information on connecting the Arduino and the Raspberry Pi, GPIO pin alternate functions, and powering other devices from the Raspberry Pi.

Also, at `www.dummies.com/extras/raspberrypiprojects`, you can find free bonus articles on topics like contact bounce and facial recognition.

Finally, throughout the book, we mention files that you can download from the book's companion website, `www.dummies.com/go/raspberrypiprojects`.

Where to Go from Here

If you're a beginner, you can't do better than starting at Chapter 1 and making sure you have your Raspberry Pi and your workspace set up. Even if you're experienced, it's worth reading the early chapters to pick up hints we've gathered from our extensive experience. If you're champing at the bit to start playing with your Pi, feel free to dive into the parts of the book that interest you most!

Part I

Getting Started with Raspberry Pi Projects

For Dummies can help you get started with lots of subjects. Visit www.dummies.com to learn more and do more with *For Dummies.*

In this part …

- ✓ Learn about your Raspberry Pi.
- ✓ Set up the hardware and operating system and your project-building workspace.
- ✓ Learn construction techniques.
- ✓ Understand the basics of programming.
- ✓ Install language extensions.
- ✓ Discover the Raspberry Pi family of computers.

Chapter 1

Getting to Know the Raspberry Pi

In This Chapter

- Getting a Raspberry Pi
- Finding out what's possible with your Raspberry Pi
- Connecting your Raspberry Pi
- Setting up your operating system
- Troubleshooting any problems

You probably wouldn't have picked up this book if you hadn't already heard about the amazing, low-cost computer for everyone, the Raspberry Pi. Besides being inexpensive, what's made the Raspberry Pi so appealing is that it's pretty easy to use. You can even change it to do things its designers never dreamed of. Unlike most consumer electronics, tablets, and desktop computers, the Raspberry Pi is designed to let you investigate how it works and change how it operates by writing your own software programs.

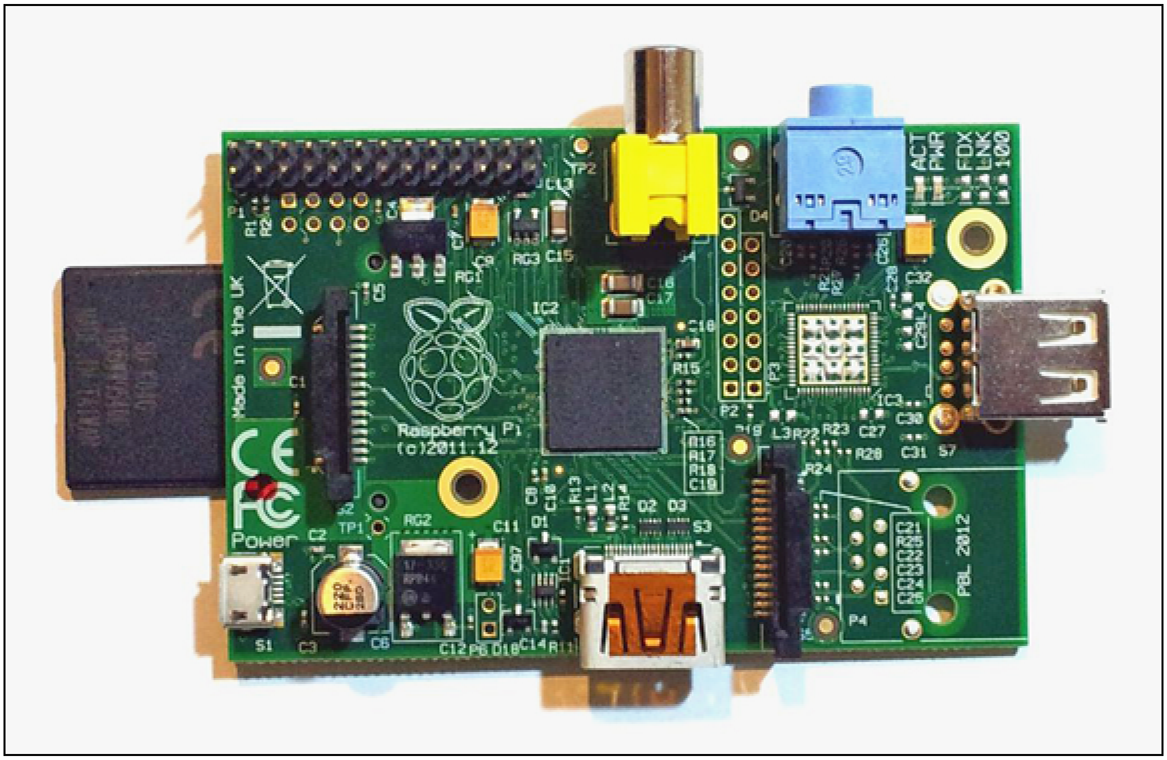
This is all possible because the Raspberry Pi uses an inexpensive but powerful processor and a free operating system, which is based upon the popular Linux platform. In this chapter, we take a look at what you need to get going and show you how to set it up.

We also tell you where to get a Raspberry Pi and the accessories you need to run it. We explain how to set up the operating system, how to connect the hardware, and what to do if you run into any problems along the way. Before long, you’ll be able to make your Raspberry Pi say, “Hello, world!”

Getting a Raspberry Pi

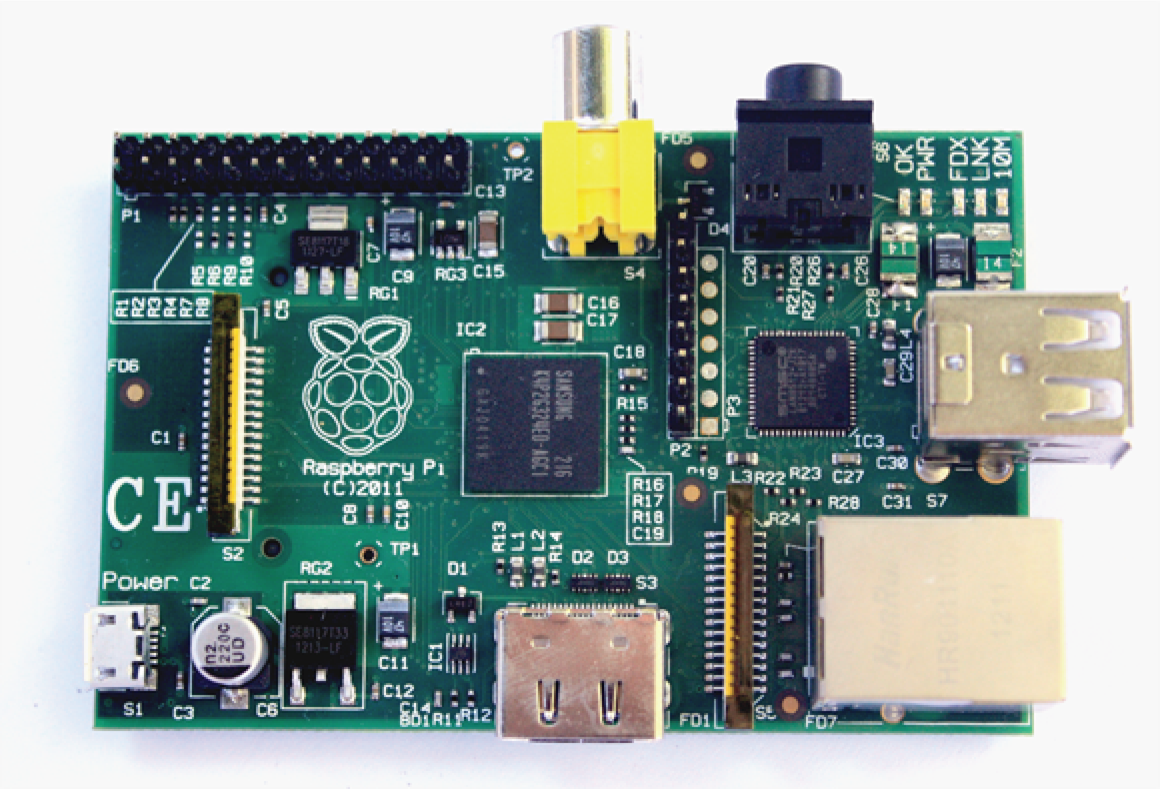
If you’re interested in building projects with a Raspberry Pi, you probably already have one. But if you don’t yet have your own Raspberry Pi, this is the section for you! You’ll be glad to know that there are a few places you can pick one up quickly and cheaply.

The Raspberry Pi comes in several versions: Model A is shown in Figure 1-1, and Model B is shown in Figure 1-2. There are other versions of the Raspberry Pi, though — Chapter 4 has a full rundown. The Model A and Model B use the same kind of processor, but the Model A is cheaper and uses less power; it has a single USB port and connections for your screen and audio. Model B has everything Model A has, plus an additional USB port and an Ethernet port for connecting to a network, so it costs a little more. For the projects in this book, you’ll want to get a Model B.



Photograph courtesy of the Raspberry Pi Foundation

Figure 1-1: Raspberry Pi Model A.



Photograph courtesy of the Raspberry Pi Foundation

Figure 1-2: Raspberry Pi Model B.

The newest model as of this writing is the Raspberry Pi 2 Model B, which replaced the Model B+, but we've kept all the projects in this book compatible with Model B and later.

The Raspberry Pi Foundation (which is technically a UK charity, not a business) created the Raspberry Pi. The Raspberry Pi Foundation licenses the manufacture of the Raspberry Pi to the biggest names in electronics in the UK, RS Components (www.rs-components.com) and Farnell, which supports Raspberry Pi under the brand name element14 (www.element14.com/community/community/raspberry-pi). If you're buying a Raspberry Pi for personal or home use, Farnell's outlet is CPC (order from http://cpc.farnell.com). In the United States, you can also buy from Newark (www.newark.com), which is a part of Farnell, and Adafruit (www.adafruit.com). These suppliers can provide you with everything you need to get your Raspberry Pi up and going, but you can only buy from them online.

If you simply can't wait to get your hands on a Raspberry Pi, and you live in the UK, you can also walk in to any Maplin electronics shop, where they're usually kept in stock. You'll pay a bit more for the convenience of shopping in a store, but you can get personal advice from the salespeople, which can be pretty useful if you have questions. At the time of this writing, you can walk into a Radio Shack in the United States and buy a Raspberry Pi starter kit, but this may change because the company is restructuring.

You can also find the Raspberry Pi for sale on eBay. There are usually plenty of listings for just a Raspberry Pi or for bundles that include all the accessories you need in order to hook it up.

If you decide to buy a Raspberry Pi on eBay, be sure to purchase from a reputable seller with plenty of good feedback. There are knock-offs out there, and they can't be guaranteed to be manufactured to the same standards as the real thing. We tend to think the cost savings isn't worth the risk of buying from eBay.

Discovering What You Can and Can't Do with a

Raspberry Pi

This book shows you how to get going with Raspberry Pi projects. After you’ve done some, you’ll have a pretty good idea of what’s possible. But when you want to go a bit further with your ideas, it’s good to know what you can realistically expect to achieve.

The first thing you see when you get up and running is a text-based prompt on the screen. You can do a lot of things just with text, but most people prefer to launch the familiar graphical user interface (GUI), the desktop environment you’re used to on any other computer. The operating system supports all the things you’d want to do in a desktop system, including playing games, browsing the web, word processing, using spreadsheets, editing photos, and playing audio and video.

But that’s not where the Raspberry Pi really shines. The great things you can do with the Pi come into play when you write your own programs and hook it up to electronics or other objects in the real world using the general-purpose input/output (GPIO) connector. Your Pi is well suited for this because these kinds of things don’t usually require the beefy processor in your desktop or laptop. Using your Pi for things you may not do with your usual computer is what makes it really fun — and that’s what this book is all about!

The Raspberry Pi uses a Broadcom BCM2835 central processing unit (CPU) and a VideoCore IV graphics processing unit (GPU) and shares the onboard memory between them. Either 256MB or 512MB of onboard memory is available. The CPU is an impressive piece of technology that enables fairly complex computing power at an extremely low price. The trade-off is that the Pi is not nearly as powerful as the full-fledged CPU in desktop and laptop computers — it’s a bit slower, roughly comparable to the capabilities of mainstream computers in the ’90s. You shouldn’t plan to do high-performance computing or run heavily graphics-intensive applications like gams or 3D modeling software — the Pi will run these, but they may be unusably slow.

You probably won’t be replacing your main computer with a Pi, but you can do a lot of experimentation with it that you may not try with your desktop or laptop, and you can easily connect your Raspberry Pi to sensors and motors in the real world, which we show you how to do in the projects in this book. And if you make any big mistakes that damage your Pi, it doesn’t cost a lot to get another one and start experimenting again!

Getting Familiar with Your Raspberry Pi

The Raspberry Pi is about the size of a credit card and has all the components that you need onboard so that you can connect it to a TV or display and start using it. These connections are shown in Figure 1-3.

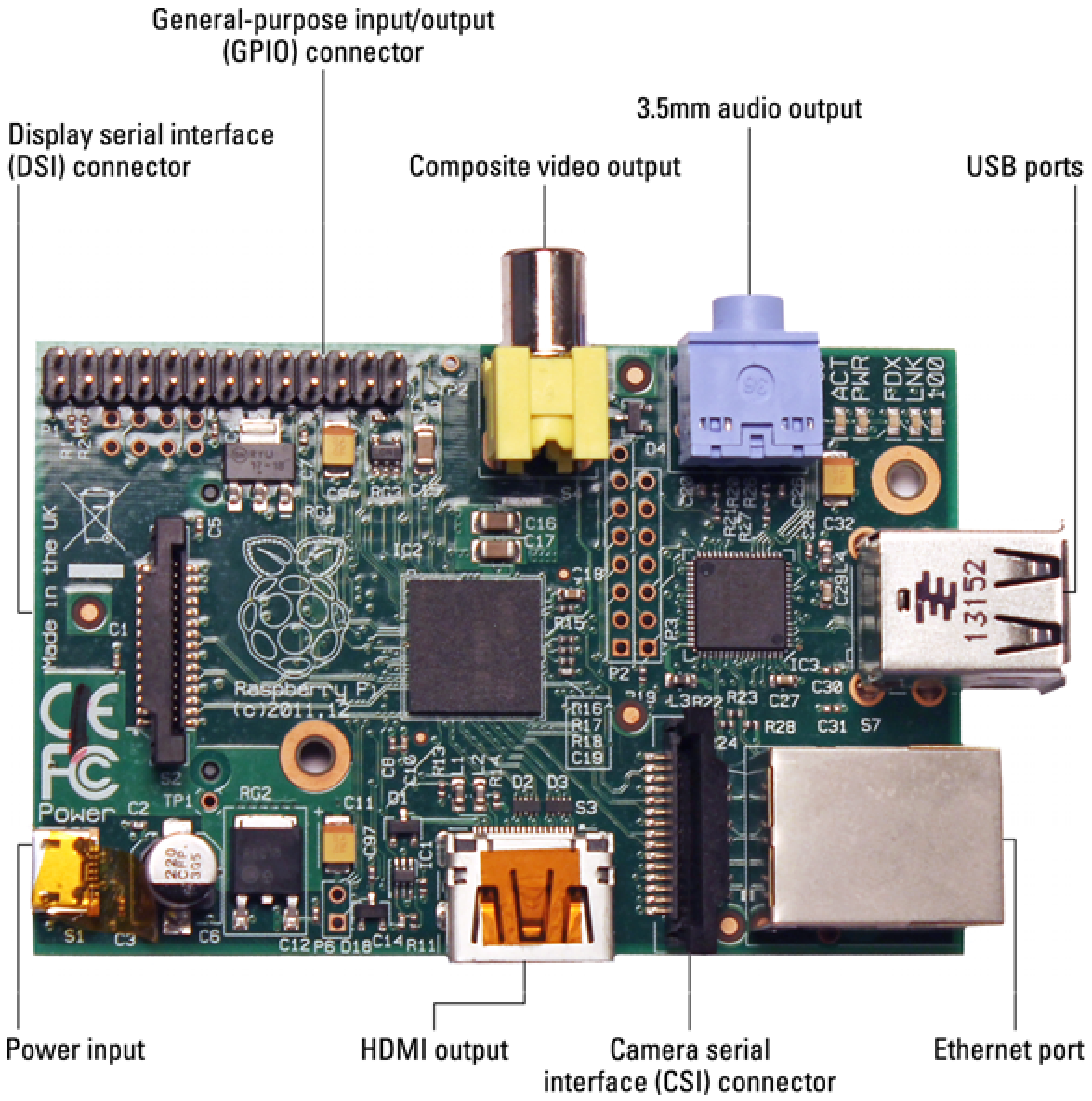

General-purpose input/output (GPIO) connector
3.5mm audio output
Display serial interface (DSI) connector
Composite video output
USB ports
Power input
HDMI output
Camera serial interface (CSI) connector
Ethernet port

Figure 1-3: Identifying the parts of the Raspberry Pi (in this case, the Model B).

Going clockwise around the board from the top left, you'll find the following connections:

- **General-purpose input/output (GPIO) connector:** This is a port for getting electrical signals into and out of your Raspberry Pi, such as for reading sensors and controlling motors. It's composed of two parallel rows of pins and is labeled P1 (for "Plug 1"). Different models of Raspberry Pi use these pins slightly differently due to the way the pins are routed on the board.

- **Composite video output:** This jack is used for connecting your Raspberry Pi to a composite video (standard TV) connection using an RCA cable.
- **Audio output:** This is a black 3.5mm jack on the upper right of the board.
- **USB port(s):** These ports allow you to connect USB accessories (such as a keyboard and mouse and external storage devices) to your board. The Model A has only one USB port to reduce costs. The Model B has two USB ports.
- **Ethernet port (Model B only):** This port is for connecting your Raspberry Pi to an Ethernet network and for accessing the Internet.
- **Camera serial interface (CSI) connector:** This slim black connector between the Ethernet jack and the HDMI output is for connecting a small camera such as a webcam. CSI connectors are available from the Raspberry Pi store.
- **HDMI output:** This port is used for sending digital video to a computer monitor. The HDMI output also can route your audio, so you may not need to use the audio output port.
- **Power input:** On the lower-left side is the micro USB power socket. The power is provided via a micro USB power supply that plugs into this port.
- **Display serial interface (DSI) connector:** In the middle of the left side of the board is a slim connector for connecting high-speed displays. It’s used for connecting a small LCD panel directly to your Raspberry Pi. You can use it for touch-based input as well!

Selecting Your Accessories

You probably have some of the important accessories lying around the house already, which was exactly what its creators had in mind. You can just use old stuff that’s gathering dust — you don’t have to buy anything, which keeps the cost down. You don’t have to get all the accessories shown in Figure 1-4 to complete the projects in this book. But at a bare minimum, you’ll need a display and a keyboard to get things going. Here’s what you see in Figure 1-4:

- **Monitor:** The Raspberry Pi’s onboard HDMI output allows you to connect a high-definition feed to just about any modern computer display. If your display has an HDMI input, all you need is a cable between the two.

  If your monitor doesn’t have an HDMI input, it probably supports DVI, which has a larger, wider connector. You can buy adapters that convert from HDMI to DVI that will allow you to use your DVI monitor. You can also use a VGA monitor (the VGA input contains three rows of holes), though these aren’t officially supported. If you want to use VGA, make sure to get an adapter that is specifically meant to work with a Raspberry Pi.

  If you don’t want to use a computer display and you have an old TV, you can use it as a display. Your Raspberry Pi has an RCA connection, which allows you to use a composite video signal to a TV with a composite video input. The picture won’t be as sharp as with a display, and text can be hard to read, so we recommend using a monitor if you can.

- **Ethernet cable:** Some of the projects in this book require connecting your Raspberry Pi to a network. For these projects, you'll need an Ethernet cable.
- **Case:** You can use your Raspberry Pi as is, right out of the box. It's pretty durable, but accidents happen, so lots of people have fun making cool cases to protect the circuit board from spills and dust. Some cases are even enhanced with glowing LEDs. The designer of the Raspberry Pi logo, Paul Beech, has designed some cool cases — check out `http://shop.pimoroni.com` to find them.
- **Mouse:** Any USB mouse will work fine. The Model A only has one USB port, so if you're using a Model A, you'll need to use a USB hub so that you can plug in both your mouse and your keyboard. You can also plug your mouse into your keyboard, if the keyboard has an extra USB port on it.
- **Keyboard:** There is a USB port on the Raspberry Pi circuit board, so you can plug in a USB keyboard. If you have an older keyboard with a round (PS/2) connector, you can use it, but you'll need a small adapter plug to convert between PS/2 and USB.
- **SD card:** Your Raspberry Pi doesn't have a hard drive, so you'll need to use some kind of external storage. An onboard SD card slot is provided for this purpose. When you plug in an SD card, your Raspberry Pi treats it just like a hard drive. SD cards are pretty cheap, so go for one with at least 8GB or 16GB of storage. SD cards have class numbers to indicate how fast they can read and write data. We recommend you get a Class 6 SD card or better.
- **USB hub:** If you're using a Model A Raspberry Pi, you may need a USB hub to connect your keyboard and mouse (see the preceding bullet). If you're using a

Model B, you don't need a USB hub, but you'll probably want extra USB inputs into your Pi, because your keyboard and mouse will take up the two USB ports. Make sure to get a USB hub that has its own power source — the Raspberry Pi can provide only limited power output via USB.

- ✓ **USB memory stick (not shown):** Memory sticks (also called *memory keys* or *flash drives*) can provide a great deal of extra storage that is fast and reliable. They're also handy for moving files from another computer or laptop to your Raspberry Pi.
- ✓ **SD card writer (not shown):** The Linux operating system for your Raspberry Pi is stored on an SD card. You can buy SD cards with the operating system already loaded, but you'll probably want to write your own at some point, so you need to make sure your computer has an SD card slot. Most desktops and laptops have one these days, but if yours doesn't, you should get an SD card writer. It plugs into your USB port and allows your computer to see what's on the card and write files to it.
- ✓ **Speakers (not shown):** Your Raspberry Pi has a 3.5mm audio jack so you can plug in headphones or external speakers. If you're using the HDMI connection and HDMI monitor as a display, the audio is sent over that cable to your screen.
- ✓ **Micro USB power supply (not shown):** Your Raspberry Pi gets its power via the micro USB connector on the side. You can use just about any power charger that fits this port, but it needs to supply 700 milliamperes (mA) of current (check the specifications printed on the side of the charger). Most good mobile phone chargers will work fine, as long as they supply 5V 700mA (3.5 watts). We recommend a

Raspberry Pi–compatible power adapter, which should be available from the supplier of your Raspberry Pi.

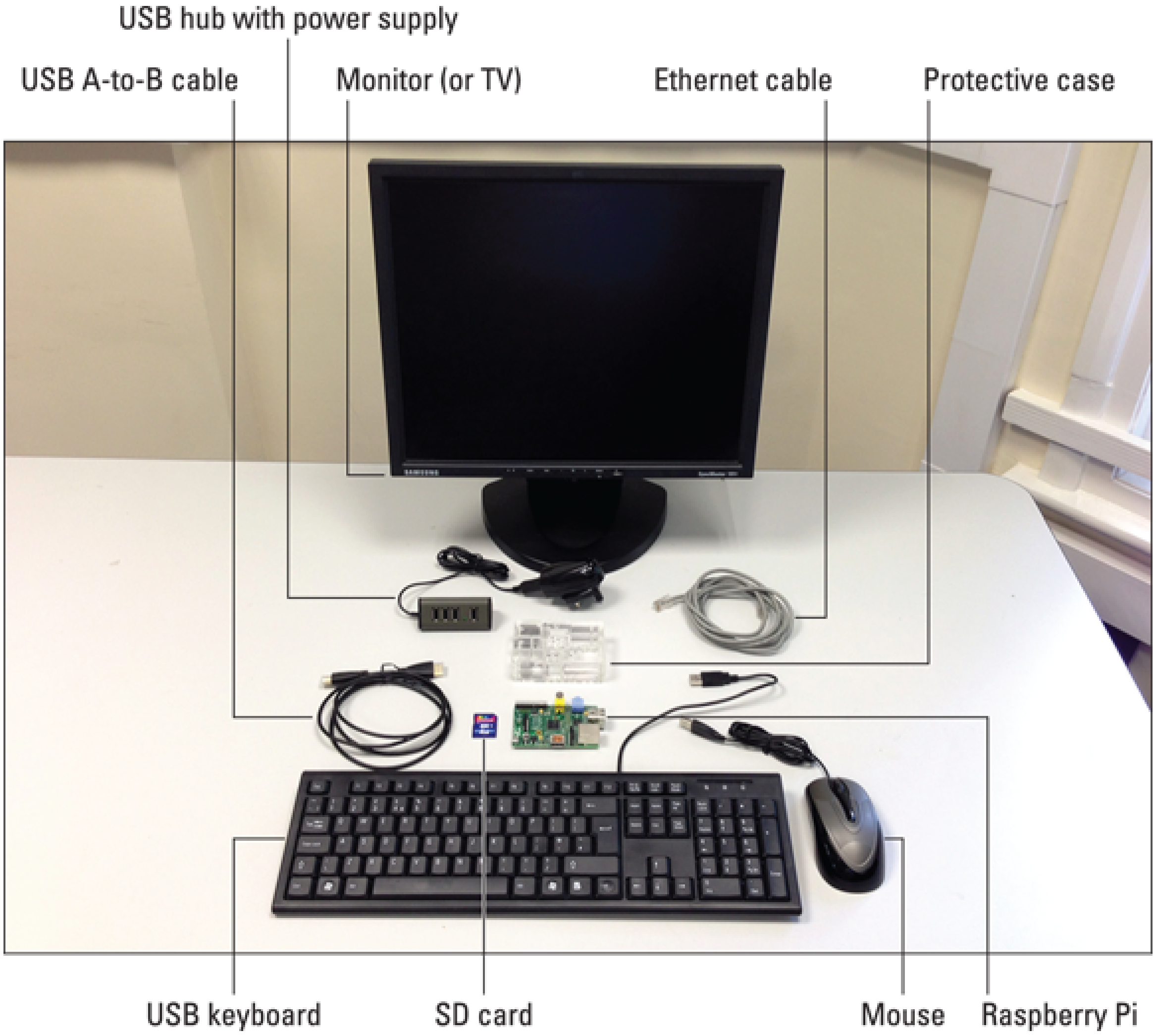


Figure 1-4: Key accessories.

Setting Up Your Operating System

To do anything useful with your Raspberry Pi, you need to have an operating system. The operating system provides the basic functions like the GUI, which most people know as the “desktop environment.” It also supports reading and writing files, runs general-purpose applications like your word processor and web browser, and runs the programs you write for your Raspberry Pi projects. Your Raspberry Pi uses the Linux operating system to do this.

Your operating system is stored on an SD card, not on a hard drive like most computers. When you turn on your Raspberry Pi, it reads the operating system that’s on the SD card. If your card isn’t inserted, the Raspberry Pi won’t be able to start up, so you need to get the operating system onto an SD card before you can do anything else.

Linux is a free operating system, unlike the ones used on Windows and Mac. It’s an open-source project, which means anyone can contribute to it — and thousands of people do. The Linux Foundation (`www.linuxfoundation.org`) coordinates these efforts and manages the standard Linux *kernel* (the core code that makes it work). All you have to do is download a copy of the operating system (see the nearby sidebar) and put it on your SD card.

You also can buy premade SD cards that already have the operating system written onto them. With one of these cards, all you have to do is insert your card and power up your Raspberry Pi. You can buy them from RS, element14, Amazon, eBay, or other online outlets. If you already have one, skip to the section on setting up your hardware, later in this chapter.

Selecting your Linux distribution

Because it’s an open project, many different versions of Linux are out there in the wild. These are referred to as Linux *distributions* (or *distros*), and you can download them for free. The different distributions are specialized for a variety of purposes. Some are made to be as bare bones as possible; others are optimized for performance.

The Raspberry Pi Foundation has endorsed a special distribution for beginners called Raspbian Wheezy, which is a version of the Debian Linux distribution. It includes a GUI called the Lightweight X11 Desktop Environment (LXDE). It also supports the programming languages that you use to write code for the projects in this book. Most of the projects use a scripting language called Python; some use a programming language called C. We assume you're using Raspbian Wheezy for the projects in this book. (***Tip:*** Your operating system is on your SD card, so if you ever want to use something other than Wheezy, you can load it onto an SD card and pop it into the SD card slot.)

You need to download Raspbian Wheezy so you can copy it to your card. The best place to get it is from the Raspberry Pi website at `www.raspberrypi.org/downloads`. Click the Download ZIP button next to Raspbian Debian Wheezy and save the file on your system in a place that you can easily find it. After you download Wheezy, you'll need to unzip the compressed file by clicking it. (***Note:*** If you have a Mac, you don't need to unzip the file before you create the SD card. Just follow the instructions for Mac later in this chapter.)

There is also a download file called NOOBS (which stands for "new out of the box software") on the Raspberry Pi website, which you can use to automate the process of creating an SD card. It's designed to be really easy to use. You just download NOOBS, unzip it, and put it on your SD card. NOOBS then manages the setup of your Raspberry Pi automatically. Even though NOOBS is supposed to be easy, problems sometimes occur, requiring you to get another program to format your SD card. We think it's simpler just to download the Raspbian Debian Wheezy and create your SD card yourself in a few easy steps.

Flashing Your SD Card

Your operating system is made up of a bunch of files that are run from the SD card itself. However, when you write the operating system's files to the SD card, they're written in a special format that Linux can read. You can't just copy them over as you would with other kinds of files. The Linux distribution you downloaded is in a special format called a *disk image.* And you *flash* the disk image to the SD card using a special little program. The program you need depends on whether you’re using Windows, Mac, or Linux.

Flashing an SD card in Windows

To create the image file in Windows, you use a special program called Image Writer for Windows. It’s free and pretty easy to use. Just follow these steps:

1. **Insert your SD card into your computer’s SD card slot or, if you don’t have one, into your SD card reader.**

   Take note of which drive letter is assigned to your SD card.

2. **Download the files at** www.sourceforge.net/projects/win32diskimager/files/latest/download **.**

   If you want more information about Image Writer for Windows, go to www.launchpad.net/win32-image-writer.

3. **Double-click the file to extract it, click Extract All Files to unzip the archive into a folder, and then open the folder.**

   Note: If the filename of the file you downloaded ends with `.exe`, when you double-click the file, an installation wizard may run.

   You should see the list of extracted files. Make sure that you aren’t looking at the zipped files.

4. **Click the file `Win32DiskImager.exe` to open it.**

5. **Click the folder icon to the right of the long white box and navigate to the Linux `.img` file you just unzipped; double-click to select it.**

This will put the file path into the long white box for you.

6. **From the Device menu, select the drive letter that your SD card has been assigned.**

   Be absolutely sure you’ve got the correct drive selected — the one that contains your SD card. Whatever drive you’ve chosen in the device menu will be completely erased!

7. **After you’ve double-checked that you’ve selected the right drive, click the Write button to create the image file on your SD card.**

Flashing an SD card on a Mac

On a Mac, you can use a simple script called RasPiWrite to do the work of flashing your image file to your SD card. First, you create a folder that RasPiWrite can use while it’s flashing your SD card. Then you use the script to create your image file. You do some of this by typing commands on the command line, using the Terminal program, which is found in your `Applications/Utilities` folder.

You need your system password to be able to flash the SD card. Just follow these steps:

1. **In your `Documents` folder, create a folder called `SD Card Prep`; in the `SD Card Prep` folder, create a folder called `RasPiWrite`.**
2. **Go to [`https://github.com/exaviorn/RasPiWrite`](https://github.com/exaviorn/RasPiWrite) to download the zip file of RasPiWrite.**

3. **Double-click the file you downloaded and open the resulting folder.**
4. **Drag the files in this folder to the `RasPiWrite` folder you created in Step 1.**
5. **Drag the zip file of your Linux distribution into your `RasPiWrite` folder.**
6. **Open the Terminal application, located in `Applications/Utilities`, and type** cd **and then a space.**
7. **Use the Finder to locate the `SD Card Prep` folder you created in Step 1; make sure you can see both the Finder window and the Terminal window, and then drag the RasPiWrite folder into the Terminal window.**

   This places that path name of that folder into the command line for you. (It's easier than typing it all out.)
8. **Press Return.**

   This switches you to the folder containing RasPiWrite.
9. **Type** ls **and press Return.**

   The list command produces a list of files in the `RasPiWrite` folder. You use it later to tell `RasPiWrite` where to get the source files for your disk image.
10. **Remove any external memory cards, USB sticks, or other pluggable storage device from your system so that you don't accidentally erase them.**
11. **Type** sudo python raspiwrite.py **to run RasPiWrite.**
12. **Enter your system password.**

    You see a progress report as your script creates the disk image. If all goes well, you should see a raspberry made of text characters.

13. **Insert your SD card into your Mac’s SD card slot or to an external SD card writer and press Return.**

14. **Follow the prompts to select the disk that corresponds to your SD card.**

    You can double-check to make sure you’ve selected the correct one by ensuring that the disk’s size (listed in the size column) corresponds to the size of your SD card. You don’t want to erase all the data on your main hard drive!

15. **You’ll be asked if you want to download a distribution; because you already did that, type** N**.**

    The program asks you to locate the disk image file.

16. **Scroll back up to where you used the `ls` command and copy the filename of the distribution; then scroll back down and paste this filename at the prompt and press Return.**

    The program extracts the image file and prepares it to upload onto your SD card. It then asks you to confirm that you’re about to erase your SD card. Be sure you’ve got the right SD card.

17. **Type** accept **to continue installing the image, and press Return.**

The flash process can take a long time. You’ll see some dots on your screen as the process continues. Depending on your system, it can take 30 minutes or even up to an hour. You can use your computer for other things during this process, but if you lose power or restart, you’ll have to start all over again.

If you’re presented with a message immediately after typing **accept**, there’s a problem. Even though the message may say `Transfer Complete`, the immediate response means that the transfer hasn’t been accomplished. This sometimes happens if the image file isn’t located where you indicated it was or if the distribution contains just an image file rather than an image file within a folder of the same name. If it happens, create a folder with the same name as the image file, drag the image file into it, and try again.

Flashing an SD card in Linux

If you’re using Linux, the process of flashing an SD card for your Raspberry Pi is pretty straightforward. We assume you’re using Ubuntu, one of the most popular Linux distributions. If you’re using another distribution, the following steps will be very similar.

When you download the Raspbian Wheezy distribution, make sure you save it where you can find it, such as in the `Documents` directory. Then follow these steps to flash your SD card:

1. **Remove any external drives, USB keys, or other SD cards from your system and insert the SD card you would like to flash for your Raspberry Pi.**
2. **Open a Terminal window.**

   This is located in the Applications menu under Accessories.
3. **Type** sudo fdisk –l **(the last character of this command is the letter l, not the number 1).**

   This starts the fixed disk program, a tool you can use to manage, erase, and separate disk drives into different logical partitions. It also shows you which drives are available on your system.

4. **Locate your SD card in the device list.**

   The list gives details about each of the drives on your system, including the size of each device in bytes and other details such as the number of heads, sectors, cylinders, and so on. Find the device that most closely matches the size of your SD card in bytes. For example, an 8GB SD card will be listed as about 8,068MB. Take note of the name of that disk’s directory. For example, on our system, the SD card is located in the directory: `/dev/sdg`.

5. **Use the `cd` command (“change directory”) to navigate to the directory where you saved your Raspbian Wheezy distribution.**

   For example, if it’s in the `Documents` directory, type **cd Documents** and press Enter.

6. **Display the name of your Raspbian Wheezy image file by typing** ls *.img**.**

7. **To write the Raspbian Wheezy image to the SD card, use the `dd` command.**

   Here’s what we would type on our system:

   ```
   sudo dd if=mydistribution.img of=/dev/sdc bs=2M
   ```

   You need to substitute the name of your distribution file where it says `mydistribution.img`. Substitute the directory where your SD card is located where it says `/dev/sdc`.

   The `sudo` command stands for “super user do” and tells Linux that you’re issuing the `dd` command as the administrator of the system. The operating system assumes you know exactly what you’re doing, and