Chapter 1

An introduction to Raspberry Pi

In this chapter, you'll learn why the Raspberry Pi was created and how it has been adopted for use in a variety of real-world applications. Most importantly, you'll learn the essentials you need to know to get started with Raspberry Pi. At the end of this chapter, you'll have your tiny computer up and running with a complete desktop environment.

An overview of Raspberry Pi

The Raspberry Pi is a single-board computer that sells for about $35. It is typically purchased as a “bare board,” as shown in Figure 1-1. This might not look like much, but that little board can do a whole lot of things. Its creators describe the Raspberry Pi as “a small and affordable computer you can use to learn programming.” That’s an accurate and concise description, but it doesn’t tell the whole story.

A computer that fits in the palm of your hand isn’t unusual nowadays, when you consider the capabilities of a typical smartphone. Mini PC products and “smart” devices with embedded computing systems are commonplace as well. Raspberry Pi shares characteristics with all of these types of devices. However, it is something quite unique altogether.

The first part of this chapter explains why the Raspberry Pi was created and the technology that made it possible. You’ll learn why it’s a great tool to use for learning to program and learning about computer technology in general. You’ll also see how it’s become popular for more than just education.

Today the Pi is used in hobbyist projects, corporate applications, and engineering development. It’s running web servers, robots, home automation systems, media and gaming consoles, 3D printers, and much more.

But it all began with a simple mission: to teach people about computing.

Recognizing a need for technology education

In 2006, a team of educators at the University of Cambridge Computer Laboratory noticed a troubling trend. The number of students applying to study computer science had been declining steadily over recent years. Those who did apply often lacked understanding of the most fundamental computing concepts. Basic programming and web design skills were common, but more advanced skills and knowledge were rare.

These Cambridge educators understood that the rapid evolution of computing technology had come at a cost. Consumer and commercial demand for advanced technologies continued to increase. Meanwhile, modern computers (including mobile devices) offered such convenience and ease-of-use that even “power users” and tech enthusiasts often had only a high-level understanding of how these machines worked. Few understood the inner workings of the hardware and software.

What was missing in education and technology

Previous generations of aspiring programmers and engineers were had been driven by necessity to build their own machines in order to hone their skills. That was no longer the case. And kids were more engaged with video games and social media than the electronics kits that had captivated earlier generations. But developing the groundbreaking technologies and killer apps of tomorrow requires a deeper understanding of how these machines work on a fundamental level.

Although mobile devices, laptops and PCs had become ubiquitous, there were further barriers to entry. The most common computing devices had been designed for the average consumer who neither needs nor wants to know what’s going on “under the hood.” The typical computer today separates the user from the essential functions of the system. That’s even truer of mobile devices, most of which are designed to prevent users from modifying or working with the underlying hardware and software.

Learning about and accessing those components requires knowledge and skills rarely taught in primary schools. Many educators saw computers as tools for word processing and web browsing, rather than tools to learn about computing itself. Students who were excited about technology didn’t have the chance to get excited about learning about it on a deeper level because, after all, you don’t know what you don’t know.

And even with the lowering costs of technology, providing students with devices suitable for learning was still cost-prohibitive for many schools and families.

In other words, a lack of understanding and resources in the educational communities, combined with a lack of affordable, accessible computing devices, had put critical learning opportunities out-of-reach for a large number of potential learners.

A new way to train and inspire students

The creators of the Raspberry Pi understood that the future of technology depends upon getting students excited about and engaged with technology in meaningful ways.

One of those Cambridge faculty members was Ebon Upton, then Director of Studies in Computer Science. His role as Director included responsibility for undergraduate admissions. Reviewing applications and interviewing prospective students showed him just how unprepared most college applicants were to study advanced technology and sciences.

Around that same time, he had also been recruited by Broadcom, a major developer of microprocessor chips and other integrated circuits widely used in telecommunications, networking and mobile devices. There he worked on the development of system-on-a-chip technology, while you'll read more about in a few pages. Basically, this is what makes it possible for your modern smartphone to perform tasks that rival a complete desktop PC from the previous decade, by putting all the components on a single microchip.

Upton was joined by fellow Cambridge Computer Laboratory lecturers Rob Mullins and Jack Lang, along with Professor of Computing Alan Mycroft. Together, these five addressed the problems of getting younger students interested in computer science and how to prepare them for college-level studies.

They decided that the solution was an affordable and accessible computer, something that would be easily available to anyone who wanted to learn. It also had to offer a degree of "openness," something that would let users interact with the hardware and software in ways such that they could really learn how things worked.

They spent several years developing their solution. Upton recognized that system-on-a-chip technology provided a way to produce low-cost computers with all the functionality needed to meet their goals. And a nonprofit organization could get those chips and the boards themselves produced at a low cost in order to put these affordable computers into the hands of aspiring learners.

The organization behind Raspberry Pi

In 2009, the five from Cambridge added two more tech professionals to their team – Pete Lomas of Norcott Technologies and David Braben of Frontier Developments – and formed the Raspberry Pi Foundation. The Foundation was registered in the UK as an educational charity with the goal of producing an affordable and accessible hardware platform to teach programming and computer science.

The Raspberry Pi Foundation is best known for the Raspberry Pi computers, but that's only part of its mission. In the next few pages, you'll read more about the Foundation's other initiatives and programs to promote education

The first generation Raspberry Pi was released in early 2012. It quickly became the best-selling PC in the U.K. In 2017, it became the third best-selling PC of all time worldwide.

The Foundation branched off into two separate organizations in 2013. They created a for-profit company to focus on commercial activities. The profits from this company, with Eben Upton as CEO, go towards funding the Foundation's non-profit work.

The technology that made Pi possible

The Raspberry Pi emerged as a solution to a problem. It became more popular than expected for numerous reasons we'll discuss later. Advances in technology drive demand for more advanced technology. And it was just such advances that made a device like Raspberry Pi possible.

In many ways, the technological developments that contributed to creating the Raspberry Pi reflect ongoing trends in the tech industry. Figure 1-4 outlines some of those. The rest of this and the next page discuss these further.

Miniaturization of computing

Over the past few decades, computers have become smaller in size and greater in power. The smartphone in your pocket today provides far more computing power than a mainframe system that might have occupied an entire floor in a university lab forty years ago. In other words, you can do a whole lot more and in a smaller package. This is part of the ongoing trend of miniaturization of computing.

The development of System-on-Chip (SoC) technology was a major breakthrough. Just as the microprocessor put mainframe computing power into a desktop machine, the SoC puts a complete PC in the palm of your hand. The multiple components that once comprised a sizable motherboard are now built-in as integrated circuits on a single chip. You'll learn more about those components, the roles they play, and why they're important, in later chapters.

You're likely already familiar with SoC technology. It's the reason that even an entry-level smartphone or modest tablet can do things to rival the capabilities of the high-end PCs of only a few years ago. The sudden and rapid adoption of mobile devices may be a defining characteristic of technological advances since the turn of the millennium.

In addition to the development of SoC, great advances have been made in core and multi-core processing, hardware and software graphics processing, networking capabilities, and software development. Together, these have created a present-day world that would have been unrecognizable only decades ago...and they are building the world of tomorrow that the children of today are only beginning to imagine.

Standardization of hardware platforms

Raspberry Pi is developed by a single entity that strictly defines the manufacture and production of the devices. Various third parties may sell their own Raspberry Pi kits, but any board legitimately bearing the Pi name can be expected reasonably to live up to the same standards. In other words, a standard platform provides consistency and reliability. You know what you're getting.

This is in contrast to another popular, similar device, the Arduino microcontroller. Although not a full-fledged PC, it is also a leader amongst small, low-cost computing devices for education and development. The Arduino is built on an open-source hardware platform. That means that the schematics are freely available and anyone with access to the right components can build and market their own version of Arduino.

That business model has benefits, but also drawbacks. Arduino branded boards can vary widely in design and quality, because there is no oversight. On the other hand, the Raspberry Pi and its components are manufactured by a select few companies which must adhere to quality standards.

This is not to say that the Arduino isn't a great tool in its own right. Both devices excel in different areas and are ideal for different kinds of applications. Both have extensive communities of professional and hobbyist developers creating hardware and software modules to work with the respective boards. In a sense, both are meant to offer a standard hardware platform. But by controlling the production of the device, the Raspberry Pi developers have ensured that what you purchase does indeed meet those standards.

A widely adopted and consistent standard platform allows for a broad ecosystem of hardware accessories and software development. That's exactly what's happened with the Raspberry Pi. From humble origins in secondary school classrooms, the little credit-card sized computer quickly became the center of global communities of developers and engineers. Hundreds of add-on accessories designed specifically for the Pi enable a modular approach that makes all kinds of things possible.

On the opposite page, you'll see just a few examples of hardware add-ons that are available. You'll also see what kind of exciting things people are doing with the Raspberry Pi, thanks to its unique hardware design and ecosystem.

Open software architecture

Meanwhile, a wide range of software developers have made their creations and code freely available for others to experiment with and build upon. In contrast to its closely controlled hardware specifications and production, the Raspberry Pi has an open software architecture. The only firmware included is a bootloader, which checks to see if an SD card is inserted and if it contains a bootable operating system.

Raspberry Pi initially was designed for embedded Linux computing. The official “Raspbian” operating system is a variant of Debian Linux optimized for Raspberry Pi. However, the Pi can support virtually any ARM-compatible Linux distribution. It can also run specialized versions of Windows and other operating systems.

There are dozens of free and open-source operating systems with versions optimized specifically to run on the Raspberry Pi. Many of these are built upon well-established distributions of Linux. That allows you to take advantage of extensive software repositories where you can download hundreds of freely available applications for nearly any purpose you might imagination.

Of course, the selection will depend upon the particular operating system and which applications support the device’s hardware. Throughout this book, you’ll become familiar with many of the leading apps that are used today on the Raspberry Pi. You’ll also learn how to evaluate compatibility and compile applications from source code when a pre-made installer package is not available.

In fact, some of the most interesting uses of Raspberry Pi come from those who write their own software scripts and application programs for the device. A complete course in software development is beyond the scope of this book, but you’ll get a feel for what you need to consider when developing software for the Raspberry Pi and other devices. If you have some programming skills, or go on to learn programming after reading this book, what you learn here will give you the foundation you need to start developing for single-board computers and embedded systems.

The success of Raspberry Pi

Over 100,000 units of the Raspberry Pi Model B were sold on the first day of its launch. Within the following two years, over 2 million units of the Model B were sold. A bestseller in the UK, the Raspberry Pi also became popular across the globe. Setting a new standard for computer education, it also became the platform of choice for uses ranging far beyond the classroom.

The Raspberry Pi is part of a larger phenomenon. Single-board computers and microcontrollers are bringing affordable and accessible computing to a new generation of technology professionals. Learners and hobbyists of all ages are discovering how such devices can be practical as well as fun. Meanwhile, established and emerging industries are making use of the same technologies. Raspberry Pi is one of the most well-known and widely adopted of these platforms.

How educators use Raspberry Pi

The Raspberry Pi is used in classroom in a variety of ways. At the most basic level, it can serve as a complete desktop computer with tools for word processing, mathematics, and of course programming. A school can outfit an entire computer lab at a fraction of what it would cost using traditional desktop PCs or even Chromebooks or tablets.

The Raspberry Pi isn’t a high-end performance computer. As you’ll see later, one of its limitations is in graphics processing. However, it can be an ideal *thin client* device for accessing applications in the cloud or on a private server. When the intensive processing and graphics rendering occur on a powerful server, schools can then deliver those desktops and apps to individual workstations powered by Raspberry Pi.

As you’ve learned, Raspberry Pi was made with programming in mind. That’s where its advantages really shine through. You can code on the Pi just as you can create office documents, but its greatest strengths are not as a workstation PC. What makes it most exciting is what you can do when you run that code on the Pi.

The ability to control hardware inputs and outputs is a main reason this is such a great tool for teaching. When programming the Raspberry Pi, students can interact with everything from low-level electronics circuits to high-level application APIs. They can write code to flash an LED on and off, then wire it up to the board themselves and see how it works. Or they can write a program to interact with Google’s voice assistant technology, connect speaker and microphone modules, and try it out in the classroom.

Moving beyond the classroom

Academics weren't the only ones concerned with the state of computer science studies. The past decade has seen an increased emphasis on Science, Technology, Engineering and Mathematics (STEM) education in the United States and worldwide. Educators, administrators, and parents are recognizing the importance of technical skills both practical and theoretical. Perhaps most encouraging is the fact that students of all ages are showing a renewed interest in these subjects.

Meanwhile, the technological developments noted in Figure 1-5 have created new devices, new opportunities, and new demands. Today's consumers are looking at home automation, streaming media and gaming devices, and all kinds of smart devices that still sound a little bit like science fiction - personal assistants that converse with you in natural language, cars that drive themselves, robots that intelligently map unexplored terrain to collect data about other planets or just vacuum your living room.

Smaller and more powerful computers are creating a world of smarter and more connected devices. This is the Internet-of-Things, where more and more everyday tools and appliances included embedded systems that enable communication, control and monitoring through other digital devices.

The size and cost of Raspberry Pi, plus its ability to run a wide range of software, make it ideal for developing and experimenting with these kinds of applications. It's unique and accessible input/output capabilities enable interacting with virtually anything one might imagine. As a result, it's being used as a platform for hobby projects and as a development board for professional applications. .

That flexibility is why the Raspberry Pi has caught the attention of many others, not just teachers and students. Demand for Raspberry Pi has led to a market of accessories developed specifically for it. Foremost amongst these are "Hardware Attached on Top" (HAT) add-ons, also known as Pi HATs or PHATs for short. These add-ons include things like sensors and motors – they're what enable interaction with the "real world," and you'll learn more about working with them throughout this book.

A brief overview of models

The Raspberry Pi has evolved over the years. Let’s take a quick look at the evolution of Raspberry Pi. Then we’ll show you the latest generation models currently available. You’ll also learn what differentiates the latest models from one another, and why each is best suited to particular types of applications.

The evolution of Raspberry Pi

The first Raspberry Pi was developed in 2011 and became available for purchase in 2012. These were the Raspberry Pi 1, Models A and B. The Model B+ followed shortly thereafter. These models were subsequently replaced by their version 2 and 3 variants.

These models are discontinued today: ….

Current Raspberry Pi models

The latest Raspberry Pi available today is the Raspberry Pi 3 Model B+.

Also available are the Pi 3 Model A and B.

The Raspberry Pi Zero and Zero W are smaller variants ideal for embedded systems. They cost about $5 and $10 respectively. Some features of the larger models were removed or scaled back to accommodate the smaller size and lower price. The original Zero did not include any networking capability. The Zero W added Wi-Fi and Bluetooth.

The Zero models are not as easy to work with for prototyping and experimenting. Typically, a developer will use a Raspberry Pi Model A or Model B while creating an application, then migrate it to a Zero or Zero W model if a smaller board is required for permanent installation.

Also available is the Raspberry Pi Compute Module. This board is about the same size as the Zero models, but priced comparably to the regular sized models. It features the same S-o-C technology as the other products, but is designed to be integrated into custom systems for industrial applications.

What you need for this book

We designed this book so that anyone with basic computer experience can complete the projects. All you need is a Raspberry Pi, a few accessories, and some freely available software.

Why we chose Raspberry Pi 3 Model B

Projects in this book have been designed to work with the Raspberry Pi 3 Model B. You'll learn concepts and skills that will apply to working with other versions and even other hardware platforms. However, for the best learning experience and most consistent results, we recommend this board. We have developed, tested and validated the projects in this book according to particular specifications. We recommend you do the same, and we'll show you just how to do that.

The Raspberry Pi 3 Model B is the most popular version of the current model (as of the time of this writing.) It offers the greatest flexibility and affordability for a variety of uses. Like all Pi models, it includes a Broadcom system-on-a-chip. The model B features a quad-core processor and a gigabyte of Random Access Memory (RAM,) also known as temporary storage.

Don't worry if you don't understand what all the technical details and specifications mean right now. You'll learn more about that in the next chapter. Then, you'll develop a deeper understanding of what these terms mean through your own hands-on experience.

Where to get it

You can purchase Raspberry Pi as a standalone board or as part of a complete kit. These are available at local electronics retailers and from many online stores. The Raspberry Pi Foundation website features a product page with links to authorized resellers.

Other hardware and accessory requirements

You'll need a microSD card with at least 16 GB of storage space. We also recommend an SD-to-microSD adapter if your desktop or laptop PC includes only the larger SD card slot. Most microSD cards sold today come with these adapterrs, including the ones packaged with most Rapsberry Pi kits.

You'll also need a microUSB power supply. A case is highly recommended to protect your board. Most kits include the microSD card, power supply and kits.

Whether your buy a kit or a standalone board, you'll still need a few other things. First, you'll need a monitor and video cable. Any reasonably moderrn desktop monitor will work, or even a television. If your display has VGA, DVI or DisplayPort inputs instead of HDMI, you'll need the appropriate adapters and cables to connect to the Pi's HDMI output.

Next, you'll need a keyboard and mouse. Bluetooth devices typically require configuration within the operating system. Since you need to interact with the device to install the operating system, we recommend a wireless keyboard and mouse combo that uses a USB dongle. Most of these will work right away without any setup. You can also use a keyboard with a wired USB connection.

A network connection is required for the projects in this book. We recommend a wired connection if available. In this case, you'll need an Ethernet cable to connect to the RJ-45 jack.

You'll also need access to a desktop or laptop PC. Individual projects may require additional hardware accessories or add-on modules. These will be indicated in the appropriate sections.

Software requirements

The software resources and skills required for this book are minimal. If you can follow our simple, step-by-step instructions, you will be successful regardless of your present skill level. We designed the book in a modular fashion, covering the most critical requirements in the first two chapters. After completing Section One, novices and advanced users alike should be able to complete any subsequent project throughout the book in any order.

All software required is freely available. We'll tell you what you need and where to get it as we go along. To get started, all you need is the official Raspberry Pi software installer loaded onto your microSD card. You'll learn all about the installer, as well as how to get it and how to prepare your microSD card, in the next few pages.

How to prepare for installing your OS

Raspbian is the name of the official operating system for all Raspberry Pi models. It is based on the well-established Debian version of Linux and optimized for Raspberry Pi hardware. The easiest way to install Raspbian is with the NOOBS installer. The name is an acronym for New Out-of-Box Software.

NOOBS will get you up and running almost as soon as you've taken your board out of the box it came in. Of course, you'll have to get everything connected and plugged in. But before you power on for the first time, you'll want to make sure that your microSD card is ready to boot up with the NOOBS installer. Next you'll see how to do that.

How to obtain the NOOBS installer

If you purchased a kit, it probably came with NOOBS pre-installed on an included microSD card. You can also purchase cars with the software installed. You can always download the latest NOOBS version for free from the Raspberry Pi website. Since it's so easy, we recommend you follow along and do just that even if you have a pre-installed card. That way, you can be sure you have the latest software available. You'll also know how to re-create the NOOBS installer card in case you want or need to in the future.

Simply navigate to https://www.raspberrypi.org/downloads/noobs/ in a web browser on your PC. Make sure to select the option labelled NOOBS: Offline and network install. The LITE version is a smaller download, but it just downloads the rest of the operating system once you run it on the Pi. The benefit of downloading the full version is that you can do an offline installation. That way, if you can't or don't connect your Pi to a network right away, you can still install the complete system.

Click the Download ZIP icon and make note of the download location on your machine. You can also download as a Torrent file if you prefer, but the ZIP option is fine for most users.

How to load NOOBS onto a microSD card

Installing NOOBS is as simple as copying the files to your microSD card. First, however, you need to ensure that the card is formatted properly. NOOBS requires a FAT (File Allocation Table) file system. In Windows, you can right-click on your card in the File Explorer and select Format. If the file system type is not already set to FAT, change it now and click Format. This will erase all data on the card.

Find the ZIP file you downloaded above and right click on it. Then select Extract Here. It should extract all files to a subfolder. Don't copy the folder itself. Open up the folder and select all items inside. Drag-and-drop or cut-and-paste these to the root directory of your microSD card.

A few helpful tips about microSD cards

Now that NOOBS is on your microSD card, you don't need to worry about keeping the files on your PC. When you install an operating system with NOOBS, it includes a recovery option that allows you to run the installer again at any time when you reboot. Also, you can always download the installer again.

You might want to make a backup of the microSD card once you have your operating system installed. This is especially important if you've configured the system to your needs, downloaded additional applications, and have important data stored on the machine. Since the card is comparable to the hard drive for your Pi, a mirror image of the system storage will allow you to restore things without repeating steps should something go wrong.

Also, as you work through the projects in this book, you'll be experimenting with different types of operating systems for the Raspberry Pi. Sometimes you can have multiple systems installed on one microSD card, but that is inconvenient for some applications and will depend on the storage space available. The projects in this book will mostly involve only a single installed operating system at a time. That means overwriting everything on the card when you start a new project.

If you've set up a system just the way you like it for a certain purpose, you may not want to lose that work. You can clone the disk image and save it on a hard drive so you can restore it again later. Alternately, you can save the microSD card itself and use a separate one for a new project.

It's a good idea to keep at least one or two cards on hand. That way, you can swap out operating systems for different projects without losing what you've already done. You might want to have several cards each with their own systems installed. That way, for example, you can change your Pi from a media player to a gaming console and again to a desktop PC just by switching out the memory cards.

How to get everything connected

Before you run the installer on your Raspberry Pi, you'll want to make sure you can interact with the device.

Actually, this is not always necessary or even desirable for some types of projects. For example, a headless server can be installed and configured remotely. We'll discuss those advanced cases later in the book. For now, you'll want to get everything connected before your first boot.

As mentioned earlier, you won't be able to pair with a Bluetooth device until you have a system to boot into. That's why we recommend a wireless keyboard and mouse combo with a USB dongle. You can also use a USB-wired keyboard and mouse. Either option requires simply plugging in to any of the four USB ports.

The other connections are equally straightforward. Plug in your display via HDMI cable to the HDMI output. If you want to use speakers or headphones, you can plug those in to the 3.5mm audio jack. (We'll talk about Bluetooth options once the system is up and running.)

If you have a wired LAN connection, plug your Ethernet cable into the networking port. If not, you'll be able to connect to a wireless network as part of the installation process.

Finally, connect your power supply via microUSB port. Remember, there's no on/off switch on the Raspberry Pi. Once your power supply is connected and plugged in to an outlet, the Pi is on. You'll only want to turn it off through the operating system's shutdown options. The only other alternative is to unplug the device. That can damage data on the card.

How to install an operating system

Installing an operating system on Raspberry Pi is similar to doing so on a desktop computer. The procedure varies depending on the operating system. Most of the operating systems designed for Raspberry Pi are Linux based. You may have heard that Linux is notoriously difficult to install. That was generally true in the past, and still holds true today for some versions. But many modern Linux distributions are just as easy to install as Windows. With NOOBS, the process is even easier.

Of course, you can manually install operating systems too. NOOBS includes only a handful of the available operating systems that work with Raspberry Pi. In a later section, you'll learn how to install other systems.

NOOBS offers a few other advantages in addition to being easy to use. First, it checks for new or updated operating systems to install when connected to a network. That means you'll have the latest options available each time you run NOOBS with a network connection. Next, it includes a recovery mode for removing, reinstalling, and updating operating systems. Finally, it provides an easy way to install multiple operating systems side-by-side on one machine.

We're going to start by installing Raspbian, the official operating system of the Raspberry Pi. Raspbian is based on Debian, one of the earliest Linux operating systems, with a worldwide community of active developers and users. Debian has a reputation for regular and stable software releases, and is known for its extensive repository of free software.

Rasbian is optimized for the Raspberry Pi hardware and offers a complete desktop environment with a graphical user interface. Many essential applications are included with the installation, and many more are available for download.

There is also a LITE version of Raspbian. This contains only the bare-bones system, with command line utilities and no graphical user interface. This is useful if you are familiar with the Linux terminal environment and want to customize a Raspbian system from the ground up. For now, we want the full version of Raspbian.

How to power on and boot up for the first time

Now you're ready to power on and boot up for the first time. Make sure the microSD card prepared with NOOBS is inserted into the slot. There's no on/off button on the Raspberry Pi. As soon as you connect the microUSB power supply to the device and plug it into an outlet, the computer will power on.

You'll be able to reboot and shut down the machine from within the operating system. You should always do this, rather than simply unplugging the device. However, when the machine is shut down, you have to disconnect then reconnect the power supply to turn it back on.

Once you power on the Raspberry Pi, you'll briefly see an intro screen as the installer loads. Then it's time to get started.

How to use NOOBS

When you boot up with NOOBS, you'll see a screen similar to that pictured in Figure 1-16. Installation is as easy as selecting and operating system and clicking Install. But let's take a closer look first.

The icons in the top row are self-explanatory. The Install icon will install the operating system(s) you select. The Edit config icon is grey because it is disabled. You can use a keyboard command during boot to log-in as a root user and activate that option, but that's not necessary now. You'll also see an option to connect to Wi-Fi. The Online help icon will be disabled if you are not connected to a network The Exit icon will exit the installer.

At the bottom of the screen, you can change localization options for language and keyboard layout.

The middle of the window shows you what operating systems are available. Your screen may show different options than what is pictured in the figure, depending on the version of NOOBS and whether or not you are connected to a network. With no network connection, you'll only see the Raspbian system that you downloaded in the ZIP file with the installer.

With an active network connection, NOOBS will show additional operating systems you can install. In that case, the system files will be downloaded as part of the installation process. The icons to the right of each operating system name indicate if the system files are present on the microSD card or must be downloaded.

You may notice that NOOBS allows you to check multiple boxes in the main window. This is because you can install multiple operating systems on one card, if space permits. Only one can run at a time, so switching between operating systems requires a reboot. Each time you power on or reboot, you'll be given the choice of which operating system to run.

NOOBS also includes a recovery mode which allows you to reach this screen again at any time. You can use this to remove or re-install any operating systems you've installed using NOOBS. You can also use it to add additional operating systems for a dual- or multi-boot system, such as described in the previous paragraph. You'll learn how to use recovery mode and multi-boot systems in a later chapter.

How to connect to a network

Since you've downloaded the latest NOOBS with the full Raspbian for offline install, a network connection isn't strictly necessary at this point. It's a good idea to connect right now anyway. NOOBS will use the same connection settings to configure your network in Raspbian, so you'll already be online when the installation completes.

If you plugged in to a wired connection prior to powering on your Pi, you should automatically be connected to the network. Otherwise, you'll need to connect to Wi-Fi. Click the Wi-Fi Networks icon located in the center of the top row of the window. Figure 1-16 shows you where to find it.

When you click the Wi-Fi Networks icon, you'll see a network selection box as pictured in Figure 1-17. Select your router or access point name from the list of networks. If there are many networks available, you may need to use the scrollbar to see the complete list.

Password authentication is checked by default. This is the preferred authentication method. Enter your password in the field labeled Password and click OK. For most typical connections, a username is not required.

How to complete the installation

Select the option labelled Raspbian [RECCOMMENDED.] It should be at the top of the list and will include "full desktop version" in the description. Scroll through the list to make sure you that only the latest Raspbian option is selected. Then click Install.

If you downloaded the latest NOOBS offline and network install package as described previously, installation should begin immediately. It will take about fifteen minutes to complete.

Older versions of NOOBS, such as those you might find on a pre-loaded card in a kit, may need to download the latest installation package. For the full Raspbian desktop, this is about 1.5 GB. Downloading and installing may take up to an hour or more, depending on your network speed.

Your Pi will reboot when installation is complete.

When Raspbian first loads, you'll be greeted with a log-in screen. The system is configured with a default user account. You'll see how to create and modify user accounts in chapter three. To get started, enter the default username pi and the password raspberry. Then click Log-In.

You should see a desktop similar to the one pictured in Figure 1-16. Take a look around and feel free to try out some of the included applications. There's a lot to explore. You'll learn all the ins and outs of the Raspbian desktop environment in chapter three.

But first, chapter two takes you in for a closer look at the hardware. You've got the software in place so you can see just what this little board can do. Now is a good opportunity to learn *how* it does these things, while learning about the various components that comprise the Raspberry Pi (and virtually all computers.)