Community Experience Distilled

Transform your Raspberry Pi into a multi-purpose web server that supports your entire multimedia world with this practical and accessible tutorial!

Raspberry Pi 2 Server Essentials

With this book, you’ll learn how to use your Raspberry Pi 2 to host a website using a range of different languages, host a game server, store files, and run everything from a media center to a cloud network. If you want to take control of your technological world, start building your own server and find out what’s possible with the Raspberry Pi microcomputer.

Begin by getting your Pi set up – follow each step as the book shows you how to prepare a network and configure the additional features that you’ll need to build your projects. It doesn’t stop there, however – the book then dives into the exciting world of the Internet of Things (IoT).

You’ll learn how to install Windows IoT onto your Raspberry Pi. Once you’ve done this, you’ll be ready to explore IoT further, as the book shows you how to use your device to host a cloud network that can form the basis of a wider IoT project.

Who this book is written for

Seeking inspiration for some new tech projects? Want to get more from your Raspberry Pi? This book has been created especially for you!

What you will learn from this book

- Host your Raspberry Pi as a web server using the minimum power resources
- Connect your Pi to the Internet and perform network benchmarking
- Explore the cross-platform features of the Pi as you run Python, Node.js, ASP.NET, and PHP all in one place
- Share files over the Internet using your Pi as a file server
- Turn your Pi into a game server; host and engage into playing
- Enjoy live HD video streaming and exclusive real-time text overlays


Piotr J Kula

$ 29.99 US £ 19.99 UK

Prices do not include local sales tax or VAT where applicable
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 "Getting Started with the Raspberry Pi"
- A synopsis of the book’s content
- More information on *Raspberry Pi 2 Server Essentials*
Piotr J Kula started his passion for computer technologies when he took apart his father's IBM PC/XT machine at the age of 6. It became clear that Piotr had a natural ability with technology after he reconstructed the machine, and it booted.

After 25 years, Piotr has worked for various companies in different countries. He has always solved complex problems or engaged in new experiences. As long as it had a processor of some sort, Piotr was always the first to try, fix, or configure it.

Today, Piotr is the CEO of his own company, Kula Solution Ltd., in the United Kingdom. Piotr is certified in several Microsoft technologies, and he specializes in Microsoft and Linux technologies.

Piotr's dream is for a unified experience between Linux and Microsoft. He has never been deterred by negativity from either side, and instead, he has strived to find a way to build a bridge between these distant giants. After discovering Raspberry Pi, he completely replaced every other embedded device that he owned and was successful at releasing Raspberry Pi Server Essentials, with Packt Publishing.

Today, Piotr is one step closer to fulfilling his dream with the release of Raspberry Pi 2 Server Essentials. This book is mostly about Linux, but it also talks about integrating and running Windows on your beloved Raspberry Pi.
Preface

The purpose of this book is to get you started with Raspberry Pi, but this book has chapters focused on Raspberry Pi 2. The main goal here is to get your projects started with some solid hardware and programming tips, which are essential.

What this book covers

Chapter 1, Getting Started with the Raspberry Pi, will cover the basic equipment that we need to use with this book. We will have to look into other peripherals that we have bought to see how the Raspberry works. We will then see how to flash the newest Raspbian image to our SD card.

Chapter 2, Preparing a Network, will illustrate how to set up LAN and a wireless connection to our network to connect to the Internet. We will set some network benchmarks and understand some of its limitations. We will also look into Dynamic DNS hosting.

Chapter 3, Configuring Extra Features, will illustrate how to update the software and firmware of Raspberry Pi. You will learn about the watchdog and understand how to buy extra decoder licenses.

Chapter 4, Using Fast Web Servers and Databases, will illustrate how to set up a quick web server using nginx with PHP, and we can decide whether we want to use MySQL or SQLite.

Chapter 5, Setting Up the Raspberry Pi as a File Server, will illustrate how to attach and format a USB storage medium. We will not only look into the various ways of sending data to Raspberry Pi, but we will also understand how to share media on the network. As an extra task, we will look into creating the hardware RAID!
Chapter 6, *Setting Up Game Servers*, will explore open source game engines that are available on the repository. We will also have a sneak peek at the Jessie repository, which is in beta testing.

Chapter 7, *Streaming Live HD Video*, will explore the camera module and illustrate a simple technique to stream a video. This chapter contains exclusive streaming tutorials.

Chapter 8, *Setting Up the Pi as a Media Center Server*, will illustrate how to connect an HD monitor and play some videos that were recorded or stored earlier. We will also look into OSMC, and we will explore the benefits of running it as a dedicated media player.

Chapter 9, *Running Your Pi from a Battery’s Power Source*, will explain the benefits of various types of battery technology and how to get the best performance without spending too much money.

Chapter 10, *Windows IoT Core*, is finally available on the embedded ARM, and it will explore the capabilities of running IoT as an Operating System with a basic programming tutorial with C#.

Chapter 11, *Running Your ownCloud*, will discuss how to install the ownCloud software on your Pi, and it will free you from privately owned services.

Chapter 12, *The Internet of Things – Sensors in the Cloud*, is more of a case study about how to complete your projects by storing accurate data accessible anywhere in the cloud.
Connecting to a network should be as easy as plugging in a cable. The question is, what can we do on the Raspberry Pi after we are connected to the Internet or a local network? This is why it is essential to learn about the hardware prerequisites and capabilities of the Raspberry Pi, so that your idea is theoretically possible to accomplish. Also, knowing your hardware inside out will make troubleshooting problems much easier to deal with later in the book.

The most common problems associated with the Raspberry Pi are related to power. These problems can cause it to restart, cause unexpected behavior, or may show a rainbow screen during the boot process if you have an external monitor connected.

This chapter is all about identifying your Raspberry Pi and the peripherals that you are using or may want to use along with it. There are two main pieces of information you should know about your Raspberry Pi: the model and version.

**Hardware requirements**

This book assumes that you are using a Raspberry Pi 2, Model B.

The Raspberry Pi 2 is mostly backwards compatible with all its predecessors. Whenever a compatibility issue occurs, it will be made clear to the user and an alternative solution will be provided, if possible.

The latest versions of Raspberry Pi at the most basic level only need a power supply and Micro SD card to run. To make initial configurations easier, it is recommended that you also have an HDMI cable, a wide screen monitor or television with HDMI input, an Ethernet cable, and a USB keyboard.
The following table describes the main differences between the various models:

<table>
<thead>
<tr>
<th>Model</th>
<th>Generation 2 Model B</th>
<th>Model B+</th>
<th>Model A+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soc</td>
<td>BCM2836</td>
<td>BCM2835</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>900 MHz Quad Core ARMv7</td>
<td>700 MHz single core ARMv6</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>VideoCore IV @ 250Mhz / OpenGL ES 2.0 / H.264/MPEG-4 AVC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDRAM</td>
<td>1 GB</td>
<td>512 MB</td>
<td>256 MB</td>
</tr>
<tr>
<td>USB</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Audio in</td>
<td>No direct input and two revision boards via I2S, all of which use a USB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>10/100 Mbit/s</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>GPIO</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>Header</td>
<td>40 pins</td>
<td>26 pins</td>
<td>26 pins</td>
</tr>
<tr>
<td>Power</td>
<td>800 mA (4.0 W)</td>
<td>800 mA (4.0 W)</td>
<td>200 mA (1 W)</td>
</tr>
<tr>
<td>SD card</td>
<td>Micro SD slot</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With the latest models of the Raspberry Pi, most peripherals should work out of the box, as these versions have been upgraded with a dedicated power circuit to handle peak loads but also reduce overall power consumption. Only the plus models have this new circuit, and all the older versions of Raspberry Pi might suffer from power problems caused by inefficient circuits. The most power-efficient Raspberry Pi is Model A+ and is commonly powered using batteries. We will discuss powering your Raspberry Pi from batteries in a later chapter of this book.

**Power supply**

The Raspberry Pi 2 should ideally be powered using a 2 amp USB power supply if you plan to connect peripherals to the USB ports. The most common power supplies found at home are 1 amp power supplies, which are typically supplied with smartphones, tablets, or mini computers. These chargers are usually made from good quality components and can easily handle the stress of additional power, power spikes, and produce a clean DC power source, which is very important for Wi-Fi or serial peripherals.

You should also pay attention to the USB cable that you are using as some cables are produced cheaply and the copper wire inside them is very thin, which struggles to deliver 1 amp or more of current when needed.
In an independent test, the best USB charger turned out to be the HP TouchPad charger (P/N 157-10157-00), and the next best ones are Apple A1265, Apple iPad A1357, and Samsung Cube ETA0U80JBE. They are considered to be the best because they produce very clean DC power and can handle peak loads for extended periods of time. Be careful, though, as a lot of counterfeit chargers are also being sold now, especially Apple A1265. As time passes, some of these power supplies may become obsolete. Try and search for newer tests.

To help visualize the DC noise problem, examine the following image. The waveform on the left-hand side is extremely noisy and will cause serious problems for communication peripherals such as Wi-Fi, bluetooth, or even serial. The waveform on the right-hand side is considered to be clean and stable:

![Image of waveforms showing clean and noisy DC power](image)

Search the Internet for tips on how to identify counterfeit chargers.

**USB hub**

When you purchase a powered USB hub, it is usually supplied with a 2 or 3 amp power supply. This is enough to power USB devices such as a Wi-Fi adapter, a USB hard drive, a few other peripherals, and even the Raspberry Pi itself.

A typical problem occurs with HD USB cameras where the picture goes black after a running for some time. Using a powered USB hub will solve this problem.
**SD cards**

SD cards all look alike, but, in fact, some use a multitude of different controllers and NAND Flash memory chips. All these combinations make up the reliability and speed of the card; the cheapest is the worst, usually. That said, spending a fortune on an SD card is not a good idea either. The speeds of SD cards are marked with a numeric symbol ranging from 1, the slowest, to 10, the fastest. A class 6 or 8 SD card is a good balance between performance and price.

However, not all Class 10 cards are the best choice either. The only way to be sure is to use a trusted speed test application to verify that the card is performing at the advertised read and write speeds; this is the best way to identify counterfeit cards:

![SD Card Comparison](image)

The SD card should only act as the primary partition for an operating system and applications installed on it. Using an SD card for frequently changing data, such as databases, is not a good idea as it degrades the life of the card faster than expected. Instead, we should use external storage devices, such as hard drives or **Network Attached Storage (NAS)**, to handle frequently changing data but also help with storing large amounts of media.

This book assumes that you are using at least an 8 GB SD card.

**Extra peripherals**

The Raspberry Pi is branded as a computer, and it is expected that we can connect various different devices to it. Raspbian is based on Debian, and an immense effort has been made to port the majority of drivers and software available for Raspbian. You might have some old USB peripherals lying around, for example, a joystick. If you can find a driver for any other Linux platform, it should be possible to make it work with Raspbian. Plug it in, use the `lsusb` command-line utility, and check whether it has been detected. If you manage to get it working with your knowledge, you should share this knowledge on a forum for the benefit of other users.
Essential peripherals
You should consider buying these peripherals and dedicating them for the use of your Raspberry Pi. They will really make it easier to set everything up and are even used for long-term purposes:

- **Wireless USB network adapters**: At the time of writing this book, the current Raspbian image supports a variety of wireless adapters without the need to install any extra drivers. Many of the mini, nano, or micro versions run directly from the Pi's USB ports and do not require a powered USB hub.

- **USB hubs**: Because some versions of the Raspberry Pi are limited to two USB ports, it might be wise to have a compatible, powered USB hub. Powered is the key word here, as this will allow you to plug in any USB device or several devices at the same time without affecting the Pi's power stability. At the time of writing this, Raspbian is not fully compatible with three USB hubs yet.

- **Keyboards and mice**: Most wired keyboards and mice will run directly off the Raspberry Pi USB port. Many Bluetooth keyboards and mice also work directly off the Pi's USB ports but require initial setup using a wired keyboard. Some wireless keyboards, such as the Microsoft 3000 series, do not need any configuration as the USB dongle emulates a PS/2 wired keyboard and can be used straight away at boot time without any extra configuration.

You can visit [http://elinux.org/RPi_VerifiedPeripherals](http://elinux.org/RPi_VerifiedPeripherals) for an up-to-date list of compatible peripherals.

Useful peripherals
As you grow more familiar with your Raspberry Pi, you will think of new bigger ideas for your projects. With such ideas, you might need a few more useful devices to help you out.

Internet 3G dongles
You can connect to the Internet using a 3G dongle. These require a lot of power and will need to run from a powered hub to operate at full speed. They are a really easy way to connect your Raspberry Pi to the Internet even in the most remote places of your country. As long as you have a basic voice signal, you should always be able to use GPRS (single channel 57.5 kbps or dual channel 115 kbps), which can be enough to send plenty of logging text data. Some countries offer free text messages, and this can also be used to send and receive the bare minimum of data. If you plan to run a server, it would be recommended that you use LAN or Wi-Fi connected to an ADSL/DSL connection instead.
Audio recording

The Raspberry Pi has its own sound output hardware, which is really good at giving you a high definition sound over HDMI or analog audio via the 3.5 mm jack.

You might find yourself in a situation where you would like to record audio from a line input or microphone; you could then use any USB 1.1 or USB 2.0 sound capture device to do this.

Other media add-ons

- **IR receivers**: Infrared (IR) receivers are a great way to control your Raspberry Pi using conventional remote controls. The FLIRC USB IR remote dongle is a great way for you to start doing this.

- **TV and radio receivers**: This is the ultimate way to turn your Raspberry Pi into a full DVR system. Record, playback, or pause live TV from HD satellite or digital TV. You can listen to your favorite radio channels too.

- **Webcams**: The Raspberry Pi has a port for its own dedicated HD camera module. Owning one of these cameras is a real treat, and the latest versions of Raspbian support UV4L (Video for Linux). A variety of USB webcams are also supported and support UV4L. Once a fairly complicated task, it has now become a lot easier to work with.

- **Multicard readers**: These come in handy if you work with various card types. Support is limited on generic types, but the USB 3.0 USRobotics all-in-one card works really well, and you can mount all six cards at the same time.

- **Alamode**: WyoLum is a start-up business that creates useful add-ons for various applications. Specifically, AlaMode is an Arduino-compatible board with a real-time clock and microSD slot that sits on top of the Raspberry Pi. You can communicate with Arduino using the Pi's dedicated UART (Universal Asynchronous Receiver/Transmitter), and it can run off the Pi's power source. If you like electronic projects and are already familiar with Arduino, this is worth looking at. You can even use it to flash other Arduino-compatible chips or upload firmware to run on its own!

- **HDMI to VGA**: If you use an older or spare monitor, television, or LCD screen that only works with VGA, you can purchase an inline HDMI to VGA converter from your favorite online auction shops or electronic stores. You must make sure to buy an active converter, which is slightly more expensive than a passive convertor. Active means that it contains a microcontroller that uses power from the HDMI port to convert the digital signal into the VGA standard. The Raspberry Pi is capable of powering this type of device.
Fun peripherals
You might have some of these lying around in your gadget box. Hopefully, reading about some of these less used devices might spark some creative ideas:

- **Joysticks**: Microsoft's Xbox 360 controller works like a mouse in X using `xboxdrv`, which can be installed from the package repository. Other joysticks might need a ported driver that can be found on Internet forums.

- **USB to SATA**: You can purchase simple USB to SATA controllers that allow you to attach SATA hard drives using dedicated power supplies. The real fun begins when you use hardware RAID-based USB to SATA controllers that can be chained in various configurations, which can give you massive storage, high redundancy, or maximum performance.

  Be careful, though, as the maximum throughput speed you can achieve is governed by the bandwidth of USB 2.0. In theory, this is a maximum speed of 60 MBps, but is shared by all the devices on the controller and not per port. There is more information about this later in the book.

- **CAN bus**: The CAN bus is the standard used in all modern motor vehicles. It is a standard port that gives mandatory data that can be interpreted by anybody, for example, throttle value, misfiring of cylinders, or air to fuel ratio. PEAK-System has a variety of peripherals and software that are compatible with the Raspberry Pi. If you have access to manufacturer-specific codes, you can even adjust engine mappings with these tools.

  Adjusting non-standard values may damage your ECU and will void any warranties.

- **Home automation**: A compatible device called TellStick runs well as a third-party, home automation device for the Raspberry Pi, but as an advanced Linux user, you should strive to make your own applications using real-time microcontrollers, such as Arduino on AlaMode, or the very cheap PICAXE microcontrollers.

- **USB missile launcher**: Available on a variety of websites and stores, this is the perfect gadget if you need to shoot plushy missiles at unidentified objects! This is purely an entertainment peripheral, but you could use it for DIY projects as well.
• **Fingerprint scanners**: Futronic's fingerprint scanners work well with Raspbian, and there are many examples that can be found online. They are standalone programmable devices that communicate with the Pi using simple messages over USB-UART and have extensive documentation available with the device.

• **Weather station**: Sparkfun sell a pretty complete weather station that includes a wind vane, anemometer, and tipping bucket rain gauge. The kit includes clamps and mounting masts. You can add a light, temperature, and humidity sensor easily and without breaking the bank. The challenge depends on using a real-time microcontroller and building your own database to log all the data. But don't worry, there are many tutorials online that will help you progress further.

---

**Installing Raspbian on the Raspberry Pi**

There are many distributions that can run on the Pi. Some are specific real-time operating systems, such as RISCOS, or mainstream operating systems, such as Raspbian or Archlinux. A few flavors of home media centers, such as OpenElec or OSMC (previously known as XBMC). As of June 2015, you will also be able to install Windows 10 Internet of Things. No Android support is foreseen for the near future.

The Raspberry Pi Foundation recommends that you use their image called NOOBS (New Out Of Box Software,) which contains frequently updated list of different operating systems. In this book, we will be using Raspbian. It is supported by the Foundation and has the best compatibility with ease of use. Raspbian is based on Debian and is similar to many other Linux operating systems. The steps for installing Raspbian are as follows:

1. For Windows and Macintosh users, it is recommended by the Raspberry Pi Foundation that you use the SD Formatter from [http://www.sdcard.com](http://www.sdcard.com).

For Windows, perform the following steps:

1. Install the SD card formatting tool.
2. Set the **Format size adjustment** option to **ON** in the **Option** menu.
3. Make sure you've selected the correct SD card.
4. Click on the **Format** button.

For Macintosh, perform the following steps:

1. Install the SD card formatting tool.
2. Select **Overwrite Format**.
3. Make sure you've selected the correct SD card.
4. Click on the **Format** button.

For Linux, perform the following steps:

1. It is recommended that you use the GParted or Parted tool in Linux.
2. Format the entire disk as FAT.
4. Unzip the archive.
5. Copy the extracted files onto the formatted SD card.
6. Insert the SD card into the Raspberry Pi. Plug in your HDMI or other video cable with a compatible keyboard and power it up.
7. The Pi will boot up and present the list of operating systems; select **Raspbian**.
8. If your display is blank, try to press the numeric keys, as listed here, while the Pi is booted up:
   - 1: HDMI mode
   - 2: HDMI safe mode
   - 3: Composite PAL
   - 4: Composite NTSC

### Understanding the design of the Raspberry Pi

The Raspberry Pi has two identifiable microchips on the PCB:

- In the center is one that's clearly marked with a Broadcom logo and text starting with BCM283x, which is the main processor
- The Raspberry Pi 2 has its RAM chip at the back of the PCB.
- Near the USB port, there is a smaller chip that is either a USB hub or a USB/LAN chip, depending on the model.

BCM283x is actually a high-performance GPU with an embedded ARM processor. It is a SoC (System on Chip), which means that there is small amount of space for code that executes when it gets turned on. This is known as Stage 1 in the boot process.
Boot process

Some network actions need to be performed during the boot process, and it is good to understand the various stages in case you need to troubleshoot something. The boot process is as follows:

1. Stage 1 begins on the GPU and executes the code SoC firmware, which starts to load Stage 2 code to the L2 cache.
2. Stage 2 reads bootcode.bin from the SD card. It initializes SDRAM (synchronous dynamic random access memory) and loads Stage 3.
3. Stage 3 is the loader.bin file. This loads start.elf, which starts the GPU.
4. During start.elf, it prepares to load kernel.img.
5. The kernel image then reads config.txt, cmdline.txt, and bcm283x.dtb.
6. If the .dtb file exists, it is loaded at 0 × 100, and the kernel is loaded at 0 × 8000 in memory.
7. The kernel image is the first binary that runs on the ARM CPU, and it can be compiled with custom support for specific hardware.
8. The operating system starts to load.

All the source code in stages 1 to 3 are closed source and protected by Broadcom. These closed source files are compiled and released by Broadcom only; you can update them on your SD card by running a firmware upgrade in Raspbian, which is covered later.

The kernel.img file connects the application to the hardware. Any computer with an operating system has a kernel of some sort. In Linux, it is possible to compile your own kernel, and it might be the first file that you might want to amend yourself. This allows you to change the boot screen, load custom drivers, or perform other tasks that you might need. This is an advanced task and is not covered in this book.

Other capabilities

BCM283x also has dedicated audio hardware together with video encoding/decoding. This allows the Raspberry Pi to playback HD (MPEG-4) content, such as videos, or render games using OpenGL ES. You can buy additional encoder/decoder licenses for extra functionality, such as MPEG-2, used in DVD video encoding and VC-1, which is used by Microsoft's WMV formats. This is also used for Silverlight live streaming.

The SD card is also directly interfaced by the Broadcom chip using dedicated hardware inputs/outputs and interrupts.
All that dedicated hardware means that while those sections of the chip are fully utilized, the ARM CPU will be idle or hardly used. This allows you to compute other transactions synchronously, and this is what makes the Raspberry Pi a truly unique single board, credit card-sized computer!

**Hardware limitations**

All this hardware that is crammed into one tiny space has its drawbacks. Some are deliberate and others are not. You should consider that these are theoretical calculations; real-world performance may vary, but are usually slower than theoretically estimated.

**Network speeds**

It may be disappointing that the Raspberry Pi Foundation decided to use a 100 Mbps LAN chip instead of a gigabit one. We need to crunch some numbers to justify this decision, though. Let’s convert megabits to a more familiar megabytes. To get to megabytes per second from megabits per second, we divide 100 Mbps by 8 (there are eight bits in a byte). This equates to 12.5 megabytes per second at 100% LAN capacity. For a single user, this is only roughly 20% of what the USB hub can handle. This means that by design, this is an unchangeable bandwidth limitation for networking.

If you plan to share files with several users at the same time, each new user will bump down the other user's bandwidth to accommodate their own. As a workaround, you could add a USB gigabit LAN peripheral. But due to speed constraints of the USB hub, you will only use approximately 48% of the gigabit LAN. To make matters worse, any hard drives running on the USB port will start to fight for bandwidth. The USB controller has to share 480 Mbps across all ports! One port is used by the 100 Mbit network card, and the other connects the hub to the GPU. For one user, this means a maximum bandwidth of 240 Mbps. Why 240 Mbps? This is because 240 Mbps goes to the LAN and 240 Mbps goes to the hard drive, and theoretically, there is no USB bandwidth left for anything else.

This could be a problem for a multiuser environment, but for home use, you would not run into any major problems as the bandwidth can accommodate HD video streams while serving other clients. This is why the cheaper 100 Mbps version was used.

**USB bottlenecks**

As it was made clear by the bottlenecks found in the LAN, the worst thing about USB bottlenecks is that there is no way to work around this problem! This is because the USB controller connects to the Broadcom and LAN chips, respectively, on the PCB without any possibility of expanding or bypassing this chip.
Time

The Raspberry Pi also does not come with a real-time clock, so timekeeping is left to Internet-based time servers. For many people, this might not cause a problem, but if you wanted to create a remote, disconnected device that depends on recording events at various times of the day, you might be left a little bit disappointed.

One easy and reliable way to do this is to connect a USB or I²C RTC that runs off a small battery. There is an easier and free option, though, but it is not as accurate; you may want to install the `fake-hwclock` package. All you need to do is set the time once, and the software will keep the track of time using a file. If you have a power outage, the software will read the file and set the time back to the last known time. The drawback is that you lose that time as there is no way to determine how long the outage lasted for.

To get time without using the Internet, you can find a cheap GPS receiver. When the GPS gets a good lock, it will provide you with extremely accurate time. This same method is used to synchronize GSM voice calls on mobile phone technologies across the world.

Another method is to use the time broadcasted by long wave radio signals. These also broadcast extremely accurate time using atomic clocks. The availability depends on your location, though. These are currently available in Colorado for most of the US; Germany, Russia, the UK for Europe; and finally, Japan. The radio waves operate on different frequencies and more research is required for this method.

References

The following are the references for this chapter:

- Crystal Disk Mark is a great SD card benchmarking tool with a lot of test results available online for comparison.
Summary

One of the most popular questions found on the Internet was how to increase the Pi's performance! With the release of the Generation 2, ARM V7, quad core 900 MHz, this is no longer the case. At the time of writing this book, one of the most popular questions is how to run the Raspberry Pi using batteries, which we will cover in a later chapter!

The purpose of understanding the architecture is vital to a successful long term project. The Raspberry Pi works like any other computer, but it was designed purely for experimental and learning purposes. It should not be used in production environments, but it is an extremely attractive solution for production nevertheless.

It is an excellent platform to share media between friends at school; it is fantastic to stream HD media on your TV and is robust enough for many standalone applications.

In the next chapter, you learn how to set up networks on the Raspberry Pi.
Where to buy this book

You can buy Raspberry Pi 2 Server Essentials from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.