Raspberry Pi for Secret Agents
Second Edition

Turn your Raspberry Pi into your very own secret agent toolbox with this set of exciting projects

Stefan Sjogelid
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 “Getting Up to No Good”
- A synopsis of the book’s content

About the Author

**Stefan Sjogelid** grew up in the 1980s in Sweden, getting hooked on 8-bit consoles, Amigas, and BBSes. With a background in system and network administration, he packed his bags for Southeast Asia and continued to work in IT for many years, before love and a magic 8 ball told him to seek new opportunities in the North American continent.

The Raspberry Pi is the latest gadget to grab Stefan's attention, and after much tinkering and learning a great deal about the unique properties of the Pi, he launched the PiLFS website (http://www.intestinate.com/pilfs), which teaches readers how to build their own GNU/Linux distribution and applications that are particularly useful on Raspberry Pi.

I'd like to thank Anton for putting up with my late night writing sessions. A special thanks to my brother for showing me Southeast Asia and to my parents for buying me a PC instead of a moped.
Raspberry Pi for Secret Agents

Second Edition

The Raspberry Pi was developed with the intention of promoting basic computer science in schools, but the Pi also represents a welcome return to simple, fun, and open computing.

Using gadgets for purposes other than those intended, especially for mischief and pranks, has always been an important part of adopting a new technology and making it your own.

With a $25 Raspberry Pi computer and a few common USB gadgets, anyone can afford to become a secret agent.

What This Book Covers

Chapter 1, Getting Up to No Good, takes you through the initial setup of the Raspberry Pi and preparing it for sneaky headless operations over the network.

Chapter 2, Audio Antics, teaches you how to eavesdrop on conversations or play pranks on friends by broadcasting your own distorted voice from a distance.

Chapter 3, Webcam and Video Wizardry, shows you how to set up a webcam video feed that can be used to detect intruders, or to stage a playback scare.

Chapter 4, Wi-Fi Pranks – Exploring Your Network, teaches you how to capture, manipulate, and spy on the traffic that flows through your network.

Chapter 5, Taking Your Pi Off-road, shows you how to encrypt your Pi and send it away on missions while keeping in touch via smartphone, GPS, and Twitter updates.
Welcome, fellow pranksters and mischief-makers, to the beginning of your journey towards a stealthier lifestyle. Naturally, you’re all anxious to get started with the cool stuff, so we’ll only devote this first, short chapter to the basic steps you’ll need to get your Raspberry Pi up and running.

First we’ll get to know the hardware a little better, and then we’ll go through the installation and configuration of the Raspbian operating system.

At the end of this chapter, you should be able to connect to your Raspberry Pi through your local network and be up to date with the latest and greatest software for your Pi.

A brief history lesson on the Pi

The Raspberry Pi is a credit-card-sized computer created by the non-profit Raspberry Pi Foundation in the UK. It all started when a chap named Eben Upton (now an employee at Broadcom) got together with his colleagues at the University of Cambridge’s computer laboratory, to discuss how they could bring back the kind of simple programming and experimentation that was widespread among kids in the 1980s on home computers such as the BBC Micro, ZX Spectrum, and Commodore 64.

After several years of tinkering, the Foundation came up with two designs for the Raspberry Pi. The $35 Model B was released first, around February 2012, originally with 256 MB of RAM. A second revision, with 512 MB of RAM, was announced in October 2012 and the $25 Model A went on sale the following year, in February 2013.

In July 2014, with over 3 million Pis sold worldwide, the Foundation unveiled the Raspberry Pi Model B+, a $35 final board revision incorporating numerous improvements requested by the ever-growing Pi community.
The following table shows the difference between the Raspberry Pi models:

<table>
<thead>
<tr>
<th>What's onboard?</th>
<th>Model A</th>
<th>Model B</th>
<th>Model B+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory (RAM)</td>
<td>256 MB</td>
<td>512 MB</td>
<td>512 MB</td>
</tr>
<tr>
<td>USB ports</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Storage card type</td>
<td>Standard SD</td>
<td>Standard SD</td>
<td>Micro SD</td>
</tr>
<tr>
<td>Power consumption</td>
<td>300 mA (1.5 W)</td>
<td>700 mA (3.5 W)</td>
<td>600 mA (3.0 W)</td>
</tr>
<tr>
<td>Ethernet networking</td>
<td>no</td>
<td>yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**The ins and outs of the Raspberry Pi**

At the heart of the Pi is the Broadcom BCM2835 System on a Chip (SOC) — imagine all the common hardware components of a PC baked into a small chip. The CPU is called ARM1176JZF-S, runs at 700 MHz, and belongs to the ARM11 family of the ARMv6 architecture. For graphics, the Pi sports a Broadcom VideoCore IV GPU, which is quite powerful for such a tiny device and capable of full HD video playback. The following figure shows the Raspberry Pi Model B:

![Raspberry Pi Model B board showing key components](image-url)
GPIO headers
At the edge of the board, we find the General Purpose Input/Output (GPIO) pins, which, as the name implies, can be used for any kind of general tinkering and to interface with other pieces of hardware.

The RCA video jack
The RCA video jack is for composite video output, which we can use to connect the Pi to one of those old television sets using an RCA connector cable. On the Model B+, this connector has been combined with the audio jack.

The Audio jack
We can get sound out of the Pi, either through the HDMI cable connected to a monitor, or from this 3.5 mm analog audio jack using headphones or desktop speakers.

Status LEDs
Status LEDs are used to tell us what the Pi is up to at the moment. They have the following meanings:

- The green light labelled ACT will blink whenever the Pi is accessing data from the SD card
- The red light labelled PWR should stay solid as long as the Pi has power
- On Model B, the three remaining LEDs will light up when a network cable is connected to the Ethernet port

USB
The USB 2.0 ports allow us to connect keyboards, mice, and most importantly for us, Wi-Fi dongles, microphones, video cameras, and GPS receivers. We can also expand the number of USB ports available with the help of a self-powered USB hub.

The Ethernet network
The Ethernet port allows us to connect the Pi to a network at a maximum speed of 100 Mbit/s. This will most commonly be a home router or a switch, but it can also be connected directly to a PC or a laptop. A Category 5 twisted-pair cable is used for wired network connections.
The CSI Camera connector

The Camera Serial Interface (CSI) is where the official Raspberry Pi camera module connects to using a flexible flat cable.

HDMI

The High-Definition Multimedia Interface (HDMI) connector is used to connect the Pi to a modern TV or monitor. The cable can carry high-resolution videos up to 1920 x 1200 pixels and digital sound. It also supports a feature called Consumer Electronics Control (CEC), which allows us to use the Pi as a remote control for many common television sets.

Power

The power input on the Raspberry Pi is a 5V (DC) Micro-USB Type B jack. A power supply with a standard USB to micro-USB cable, such as a common cell phone charger, is then connected to feed the Pi.

The most frequently reported issues from Raspberry Pi users are without a doubt those caused by insufficient power supplies and power-hungry USB devices. Should you experience random reboots, or that your Ethernet port or attached USB device suddenly stops working, it’s likely that your Pi is not getting enough stable power.

Take a look at the OUTPUT printed on your power adapter. The voltage should be between 5V to 5.25V and the amperage provided should be at least 700mA. The official 2A power supply sold by the Foundation is highly recommended (1A = 1000mA).
You can help your Pi by moving your devices to a self-powered USB hub (a hub that has its own power supply).

Also note that the Pi is very sensitive to devices being inserted or removed while it's running, and powering your Pi from another computer's USB port usually doesn't work well.

**SD card**

The SD card is where all our data lives, and the Pi will not start without one inserted into the slot. The Raspberry Pi Model A and B takes a standard-sized SD card while the Model B+ uses the tiny Micro SD.

SD cards come with a wide variety of data storage capabilities. A card with a minimum of 4 GB of storage space is recommended for the projects in this book. The SD cards also carry a class number that indicates the read/write speed of the card—the higher the better.

**Installing the Raspbian OS on the SD card**

Computers can't do anything useful without an operating system, and the Pi is no exception. To help us add one, we'll be using the easy operating system installation manager called **New Out Of the Box Software (NOOBS)**. NOOBS will let us choose from a growing list of operating systems available for the Pi, but we'll stick with the officially recommended OS—the Raspbian GNU/Linux distribution.

**Getting NOOBS**

There are two main ways to obtain NOOBS. You can either buy it preinstalled on an SD card from your Raspberry Pi dealer, or download NOOBS yourself and copy it to an empty SD card on a computer with an SD card slot.

If you do have access to a computer but it lacks an SD card slot, it's a wise choice to invest in an external SD card reader/writer. They don't cost much and chances are you'll want to re-install or try a different operating system on your SD card sooner or later.
To download NOOBS, visit the site http://www.raspberrypi.org/downloads, where you have the option of downloading the full installer that includes the Raspbian operating system image at about 740 MB in size, or the lite package that lets you pick and choose other operating systems to install over a wired Ethernet connection. Just click on the link for the full ZIP file and wait for your download to start or use the torrent link if you prefer, but we will not cover that in this book.

**Formatting the SD card**

Before we copy NOOBS to the SD card, it should be empty and formatted with the FAT file system. You can either use the native application of your computer's operating system to do so, or preferably the SD Formatter utility offered by the SD Association at http://www.sdcard.org/downloads. Perform the following steps in order to format the SD card:

1. Download and install the utility for either Windows or Mac.
2. Insert your SD card and start the application.
3. Ensure that SD Formatter has detected the correct volume of your SD card.
4. Click on the **Option** button and set **format size adjustment** to **ON**.
5. Click on **Format** to erase and format your SD card:

![SD Formatter running in Windows](image)

**Starting NOOBS**

All right, you've been patient long enough; it's time we take your Pi out for a spin! With your SD card formatted and the NOOBS download completed, extract the NOOBS ZIP file and copy all the contents to the SD card.
To avoid data loss, remember to always safely eject your SD card from your computer by right-clicking the SD card icon and selecting **Eject**.

For this first voyage, it is recommended that you go easy on the peripherals until we have properly configured the Pi and verified a basic stable operation. Connect a USB keyboard and mouse, a monitor or TV, and a Wi-Fi dongle or an Ethernet cable plugged into your home router. Finally, insert your SD card and attach the power cable.

Within seconds, you should see NOOBS starting up with a reminder saying that if you ever want to return to NOOBS in the future, for recovery purposes or to try out another operating system, simply hold down the *Shift* key when you see the message.

If your display remains blank after several minutes, try pressing the number keys 1 through 4 to switch between the different video output modes.

Finally, check the box next to **Raspbian** and click on the **Install** icon.

Installation will take around 20 minutes depending on the speed of your SD card.
Booting up and configuring Raspbian

Once NOOBS has completed the Raspbian installation and your Pi has been rebooted, you should see text scroll by on your display. These are status messages from the booting Linux kernel.

The output will come to a halt in a minute and you'll be presented with a menu-type application called raspi-config. Use your arrow keys to navigate and press the Enter key to select menu options. The menu options are as follows:

- **Expand Filesystem**: This option is only useful for when you write a Raspbian image to your SD card directly, without using NOOBS. For us, this step has already been taken care of.

- **Change User Password**: Select this option to change the password for the default user pi. This is strongly recommended. Just in case you forget, the default password is raspberry.

- **Enable Boot to Desktop/Scratch**: This option allows you to change whether the graphical desktop or the Scratch programming environment should be started automatically each time you boot the Pi. Since we will mostly work on the command line in this book, it's recommended that you leave this option as is.

- **Internationalisation Options**: This menu allows you to add non-English languages and keyboard layouts to the system. More importantly, it lets you set the correct time zone, because any scheduling we do in the later chapters depends on this. It's also nice to have the correct time in the log files.

- **Enable Camera**: Select this option if you have a camera module connected to the CSI connector on the Raspberry Pi board.
Chapter 1

- **Add to Rastrack**: This is a completely optional way of adding your Pi to an online map (http://rastrack.co.uk) that tracks where people are using Raspberry Pis around the world.

- **Overclock**: This option allows you to add some turbo boost to the Pi. Only experiment with overclocking once you have established that your system runs stable at default speed. Also note that while overclocking will not void the warranty of the Pi, it could reduce its lifetime.

The **Advanced Options** menu contains the following options:

- **Overscan**: If you see thick black borders around the blue background on your monitor, select this option and **disable** to make them go away the next time you boot the Pi.

- **Hostname**: This option allows you to change the name of your Pi as it appears to other computers on your local network. It is up to your home router to translate this name into the correct IP address of the Pi as we will see later in this chapter. The default hostname is **raspberrypi**.

- **Memory Split**: This option lets you change how much of your Pi’s memory the **Graphics Processing Unit (GPU)** is allowed to use. To use the camera module or play HD movies, the GPU needs 128 MB of the RAM.

- **SSH**: Select this option to enable or disable the **Secure Shell** service. SSH is a very important part of our setup and allows us to log in remotely to the Pi from another computer. It is active and enabled by default, so leave this option alone for now.

- **SPI**: This option enables support for a certain group of add-on boards that connects to the GPIO header of the Pi.

- **I2C**: This option enables support for a group of add-on chips that communicate via I2C such as real-time clock modules.

- **Serial**: This option allows or disallows communicating with the Pi via a serial cable and terminal application running on another computer.

- **Audio**: This option can be used to force the audio output through either HDMI or the analog audio jack.

- **Update**: This option will try to upgrade the raspi-config application itself to the latest version. You can leave this option alone for now as we will make sure all the software is up to date later in this chapter.

Once you’re happy with the configuration, select **Finish** and **Yes** to reboot the Pi.

At the **raspberrypi login** prompt, enter **pi** as the username and enter the password you chose.
Basic commands to explore your Pi

Now that you're logged in, let's have a look at a handful out of the several hundred possible commands that you can type in the command line. When a command is run prepended with `sudo`, it'll start with super user or root privileges. That's the equivalent of the Administrator user in the Windows world.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sudo raspi-config</code></td>
<td>This starts raspi-config, which lets you reconfigure your system.</td>
</tr>
<tr>
<td><code>sudo reboot</code></td>
<td>This reboots the Pi.</td>
</tr>
<tr>
<td><code>sudo poweroff</code></td>
<td>This prepares the Pi to be powered off. Always type this before pulling the plug!</td>
</tr>
<tr>
<td><code>sudo su</code></td>
<td>This becomes the root user. Just be careful not to delete anything by mistake!</td>
</tr>
<tr>
<td><code>df / -h</code></td>
<td>This displays the amount of disk space available on your SD card.</td>
</tr>
<tr>
<td><code>free -h</code></td>
<td>This displays memory usage information.</td>
</tr>
<tr>
<td><code>date</code></td>
<td>This displays the current time.</td>
</tr>
<tr>
<td><code>top</code></td>
<td>This starts a task manager that shows running processes with the most CPU hungry applications on top. Press Q to quit.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>This logs you out of your current shell or SSH session.</td>
</tr>
<tr>
<td><code>sudo touch /forcefsck</code></td>
<td>This will enable your Pi to check/repair the root file system at the next boot. It's a useful command if you suspect your SD card data might be damaged.</td>
</tr>
</tbody>
</table>

Getting help with commands

Here are a few tricks that will help you get the hang of the Linux command line:

- **Command tab completion**: If you can't quite remember the exact name of a command, but you think it stars with `raspi`, begin typing the first few letters and press the `Tab` key twice to get a list of all commands starting with those letters. Tab completion can also save you some typing when inputting directory paths and filenames.
• **Manual pages**: Most commands come with a manual that describes the usage of the command in more detail. For example, to read the manual for the `top` application, type `man top`. Use the arrow keys to scroll and press `Q` to quit.

• **Built-in help**: Most commands can be asked to print out a help text about their usage. The two most common arguments are `--help` and `-h`. For example, to see the help text for the `ls` command, type `ls --help`.

**Accessing the Pi over the network using SSH**

Pretty much all the pranks and projects in this book will be done at the command line while being remotely logged in to the Pi over the network through SSH. Before we can do that, we need to be sure our Pi is reachable and we need to know its IP address. First we'll look at wired networks, then at Wi-Fi.

**Wired network setup**

So you've plugged an Ethernet patch cable into the Pi and connected it to your home router, now what? Well, there should be all kinds of blinking lights both around the port of your router and on your Pi. The next thing that needs to happen is for the router to assign an IP address to the Pi using **Dynamic Host Configuration Protocol (DHCP)**. DHCP is a common service on network equipment that hands out unique IP addresses to all computers that want to join the network.

Let's have a look at the address assigned to the Ethernet port (`eth0`) on the Pi itself using the following command:

```bash
pi@raspberrypi ~ $ ip addr show eth0
```

If your DHCP service is working correctly, you should see a line similar to the following output:

```
et 192.168.1.20/24 brd 192.168.1.255 scope global eth0
```

The digits between `inet` and the `/` character is your Pi's IP address, `192.168.1.20` in this case.

If your output doesn't have a line beginning with `inet`, it's most likely that your router lacks a DHCP service, or that the service needs to be enabled or configured. Exactly how to do this is outside the scope of this book, but try the manual for your router and search for `dhcp`. 
For static address network setups without DHCP, see the *Setting up point-to-point networking* section in *Chapter 5, Taking Your Pi Off-road*.

**Wi-Fi network setup**

The easiest way to set up Wi-Fi networking is to use the included **WiFi Config** GUI application. Therefore, we will briefly enter the graphical desktop environment, configure the Wi-Fi, and save the information so that the Wi-Fi dongle will associate with your access point automatically on boot.

If you have a USB hub handy, you'll want to connect your keyboard, mouse, and Wi-Fi dongle now. While it's fully possible to perform the following actions using only the keyboard, a mouse will be very convenient:

1. Type `startx` to start the graphical desktop environment.
2. Click on the Menu button and then select WiFi Configuration located under Preferences.
3. From the **Network** drop-down menu, select **Add**.
4. Fill out the information for your access point and click on the **Add** button.
   
   If you're unsure about the Authentication type of your access point, pressing **Scan** might help you figure it out.

![Adding an access point in Wi-Fi Config](image-url)
5. Your Wi-Fi adapter will associate immediately with the access point and should receive an IP address, as listed under the **Current Status** tab.

6. From the **File** drop-down menu, select **Save Configuration**.

7. Exit the application and log out of the desktop environment.

To find out about the leased IP address of your Wi-Fi adapter (wlan0), without having to enter the graphical desktop, use the following command:

```
pi@raspberrypi ~ $ ip addr show wlan0
```

You should see a line similar to the following output:

```
inet 192.168.1.15/24 brd 192.168.1.255 scope global wlan0
```

The digits between *inet* and the */* character is your Pi's IP address, 192.168.1.15 in this case.

To obtain information about the associated access point and signal quality, use the `iwconfig` command.

### Connecting to the Pi from Windows

We will be using an application called **PuTTY** to connect to the SSH service on the Pi. The steps to be followed are:

1. To download the application, visit this address [http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html).

2. Download the all-inclusive windows installer called `putty-0.63-installer.exe`, since the file copy utilities will come in handy in later chapters.

3. Install the application by running the installer.

4. Start **PuTTY** from the shortcut in your Start menu.

5. In the **Host name (or IP address)** field, input the IP address of your Pi, which we found out previously. If your network provides a convenient local **DNS** service, you might be able to type `raspberrypi. (with the trailing dot)` instead of the IP address; try it and see whether it works.

6. Click on **Open** to initiate the connection to the Pi.

7. The first time you connect to the Pi or any foreign system over SSH, you'll be prompted with a warning and a chance to verify the remote system's **RSA key fingerprint** before continuing. This is a security feature designed to ensure the authenticity of the remote system. Since we know that our Pi is indeed our Pi, select **Yes** to trust this key and continue the connection.
8. Log in as **pi** and enter the password you chose earlier with `raspi-config`.
9. You're now logged in as the user **pi**. When you've had enough pranking for the day, type `exit` to quit your SSH session.

**Connecting to the Pi from Mac OS X or Linux**

Both Mac OS X and Linux come with command-line SSH clients. Follow these steps:

1. Open up a terminal (located in `/Applications/Utilities` on the Mac).
2. Type in the following command, but replace `[IP address]` with the particular IP address of your Pi that we found out previously:
   ```bash
   $ ssh pi@[IP address]
   ```
   If your network provides a convenient local DNS service, you might be able to type `raspberrypi` instead of the IP address try it and see whether it works.
3. The first time you connect to the Pi or any foreign system over SSH, you'll be prompted with a warning and a chance to verify the remote system's RSA key fingerprint before continuing. This is a security feature designed to ensure the authenticity of the remote system. Since we know that our Pi is indeed our Pi, select **Yes** to trust this key and continue the connection.
4. Type the password of the user **pi** that you chose earlier with `raspi-config`.
5. You're now logged in as the user **pi**. When you've had enough pranking for the day, type `exit` to quit your SSH session.

**The importance of a sneaky headless setup**

You might be wondering why we bother with SSH and typing stuff in the command line at all when Raspbian comes with a perfectly nice graphical desktop environment and a whole repository of GUI applications. Well, the first reason is that we need all the CPU power we can get out of the Pi for our projects. With the current graphics drivers for X (the graphics system), the desktop eats up too much of the Pi's resources and the CPU is more concerned with redrawing fancy windows than with running our mischievous applications.
The second reason is that of stealth and secrecy. Usually, we want to be able to hide our Pi with as few wires running to and fro as possible. Obviously, a Pi hidden in a room becomes a lot more visible if someone trips over a connected monitor or keyboard. This is why we make sure all our pranks can be controlled and triggered from a remote location.

**Keeping your system up-to-date**

A community effort such as Raspbian and the Debian distribution on which it is based is constantly being worked on and improved by hundreds of developers every day. All of them are trying hard to make the Pi run as smoothly as possible, support as many different peripherals as possible, and to squish any discovered software bugs.

All those improvements come to you in the form of package and firmware updates. To keep your Raspbian OS up to date, you need to know the following three commands:

- `sudo apt-get update`: This fetches information about what packages have been updated.
- `sudo apt-get dist-upgrade`: This proceeds to install the updated packages. Select **Yes** when prompted for installation.
- `sudo rpi-update`: This upgrades to the latest firmware from the Raspberry Pi Foundation’s GitHub repository (an online source code management service).

The firmware updates are more related to the Raspberry Pi hardware and may contain improvements to the Linux kernel, new drivers for USB gadgets, or system stability fixes.

**Backing up your SD card**

It happens to everyone at one point or another—you've put hours into perfecting your Raspbian installation, setting up applications, and hacking away at clever code when out of nowhere your cat/dog/next-of-kin swoops down on your keyboard and triggers the self-destruct mechanism from the Erasing the Pi should it fall into the wrong hands section in Chapter 5, Taking Your Pi Off-road.

Not to worry Agent, backing up an SD card is quite simple as long as you've got the required disk space to store it.
Complete SD card backup in Windows

We'll be making a complete mirror image of your SD card. The data will be stored in a single file that will be the same size as that of your SD card.

1. Power off your Pi safely and move the SD card to your computer's card reader.
2. Visit http://sourceforge.net/projects/win32diskimager/ and download the latest version of the Win32 Disk Imager application (Win32DiskImager-0.9.5-install.exe at the time of writing).
3. Install the application by running the installer.
4. Start Win32DiskImager from the shortcut in your Start menu.

![Win32 Disk Imager](image)

5. Ensure that the correct volume of your SD card is shown under Device.
6. Click on the folder icon and navigate to the folder where you'd like to store the image.
7. Enter a good file name for your image and click on Open. The standard file extension for image files is .img.
8. Finally, after verifying that the full Image File path looks good, click on Read.

Once your image backup has completed successfully, you can compress it to save quite a bit of disk space. Just right-click on the image file and select Send to, then click on Compressed (zipped) folder.

To restore your SD card from a backup image, simply point Win32 Disk Imager to your image file, and click on the Write button.
Win32 Disk Imager is also used to write the operating system images available for download at http://www.raspberrypi.org/downloads directly to the SD card without using NOOBS.

**Complete SD card backup in MAC OS X**

We'll be making a complete mirror image of your SD card. The data will be stored in a single compressed file, which should result in a smaller size than that of your SD card. The steps to be followed for a data backup are:

1. Power off your Pi safely and move the SD card to your computer's card reader.
2. Open up a terminal (located in /Applications/Utilities on the Mac).
3. Type `diskutil list` to obtain a readout of all connected storage devices.
4. To correctly identify your SD card, we're looking for a disk that has at least one Windows and one Linux entry under **TYPE** (there will be two of each type if we installed Raspbian through NOOBS).
5. Take note of that disk's first **IDENTIFIER** field (**disk1** in the screenshot).
6. As a security precaution, we will first unmount the SD card so that no applications running in the background can change data as we make our backup. Use the following command, but replace `[disk]` with the **IDENTIFIER** field of your SD card:

   ```bash
   $ diskutil unmountdisk [disk]
   ```

7. Now we'll do a complete copy of the SD card and store it in a file called `agent_sdcard.img.gz` on your desktop. Type the following command, but replace `[disk]` with the **IDENTIFIER** field of your SD card (note the letter `r` in front of `disk`):

   ```bash
   $ sudo dd if=/dev/r[disk] bs=4m | gzip > ~/Desktop/agent_sdcard.img.gz
   ```
8. You might be asked to input your user password so that `sudo` is allowed to start. The backup process doesn’t produce much output as it runs, but a status report can be produced by pressing `Ctrl + T` in the `Terminal` window.

```
agentbook:~ agent$ diskutil list
/dev/disk0
  #: TYPE NAME SIZE IDENTIFIER
  0: GUID_partition_scheme 121.3 GB disk0
  1: EFI EFI 209.7 MB disk0s1
  2: Apple_HFS MacHD 120.5 GB disk0s2
  3: Apple_Boot Recovery HD 650.0 MB disk0s3

/dev/disk1
  #: TYPE NAME SIZE IDENTIFIER
  0: FDisk_partition_scheme 15.9 GB disk1
  1: Windows_FAT_16 RECOVERY 1.5 GB disk1s1
  2: Linux 33.6 MB disk1s3
  3: Windows_FAT_32 BOOT 62.9 MB disk1s5
  4: Linux 14.3 GB disk1s6

agentbook:~ agent$ diskutil umountdisk disk1
Unmount of all volumes on disk1 was successful
agentbook:~ agent$ sudo dd if=/dev/disk1 bs=4m | gzip > ~/Desktop/agent_sdcard.img.gz
Password: 3798+1 records in
3798+1 records out
1593159456 bytes transferred in 894.658617 secs (17807395 bytes/sec)
```

Backing up an SD card in Mac OS X

To restore your SD card from a backup image, repeat the previous steps but use this command instead at step 7:

```
$ gzip -dc ~/Desktop/agent_sdcard.img.gz | sudo dd of=/dev/r[disk] bs=4m
```

If you type the wrong disk you could potentially overwrite your Mac’s internal hard drive without any warning. Do triple check!

The restore image method is also used to write the operating system images available for download at [http://www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads) directly to the SD card without using NOOBS.

**Complete SD card backup in Linux**

We’ll be making a complete mirror image of your SD card. The data will be stored in a single compressed file, which should result in a smaller size than that of your SD card.
1. Power off your Pi safely and move the SD card to your computer's card reader.

2. Open up a terminal.

3. Type `sudo lsblk -f` to obtain a readout of all connected storage devices.

4. To correctly identify your SD card, we're looking for a disk that has at least one vfat and one ext4 entry under FSTYPE (there will be two of each type if we installed Raspbian through NOOBS).

5. Take note of that disk's NAME (sdb in the screenshot).

6. If any of the partitions under your disk's NAME have a MOUNTPOINT listed, you should unmount it first. Use the following command, but replace [mountpoint] with the mountpoint of your partition:

   ```
   $ sudo umount [mountpoint]
   ```

7. Now we'll do a complete copy of the SD card and store it in a file called `agent_sdcard.img.gz` in your home directory. Type the following command, but replace [disk] with the NAME of your SD card:

   ```
   $ sudo dd if=/dev/[disk] bs=4M | gzip > ~/agent_sdcard.img.gz
   ```

8. The backup process doesn't produce much output as it runs, but a status report can be produced by typing `sudo pkill -USR1 dd` in another terminal console.

   ```
   $ sudo lsblk -f
   NAME   FSTYPE LABEL UUID       MOUNTPOINT
   sda     -sda1 btrfs  EeeHD   736b2bb6-906d-42b0-bba8-c4c37106aa95 /
   -sda2  swap      swap      f05d0121-2873-4073-bc59-8a7414bd91bd [SWAP]
   sdb     -sdb1 vfat   RECOVERY D083-9842
   -sdb2
   -sdb3 ext4   SETTING 4f7087b8-01d8-4a6d-bc09-ec298e6673d6
   -sdb5 vfat   BOOT     012f-16e2
   -sdb6 ext4   root     883f6b84-d74e-411d-9c55-1a0eaf6d28f
   $ sudo dd if=/dev/sdb bs=4M | gzip > ~/agent_sdcard.img.gz
   3798+1 records in
   3798+1 records out
   15931539456 bytes (16 GB) copied, 2782.84 s, 5.7 MB/s
   ```

Back up an SD card in Linux

To restore your SD card from a backup image, repeat the previous steps but use this command instead at step 7:

```
$ gzip -dc ~/agent_sdcard.img.gz | sudo dd of=/dev/[disk] bs=4M
```
If you type the wrong disk you could potentially overwrite your computer's internal hard drive without any warning. Do triple check!

The restore image method is also used to write the operating system images available for download at http://www.raspberrypi.org/downloads directly to the SD card without using NOOBS.

**Summary**

In this chapter, you took a look at the different parts of the Raspberry Pi board and learned a bit about how it came to be. You also learned about the importance of a good power supply and how a powered USB hub can help alleviate some of the power drain caused by hungry USB peripherals.

We then gave the Pi an operating system to run by downloading NOOBS to help us install Raspbian onto our SD card. Raspbian was booted and configured with the raspi-config utility. You also learned a few helpful Linux commands and how to set up remote connections from SSH clients over the network.

Finally, you learned how to keep Raspbian up to date and how to create a complete backup image of your precious SD card.

In the upcoming chapter, we'll be connecting sound gadgets to the Pi and getting our feet wet in the big pond of spy techniques.
Where to buy this book


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

Click here for ordering and shipping details.