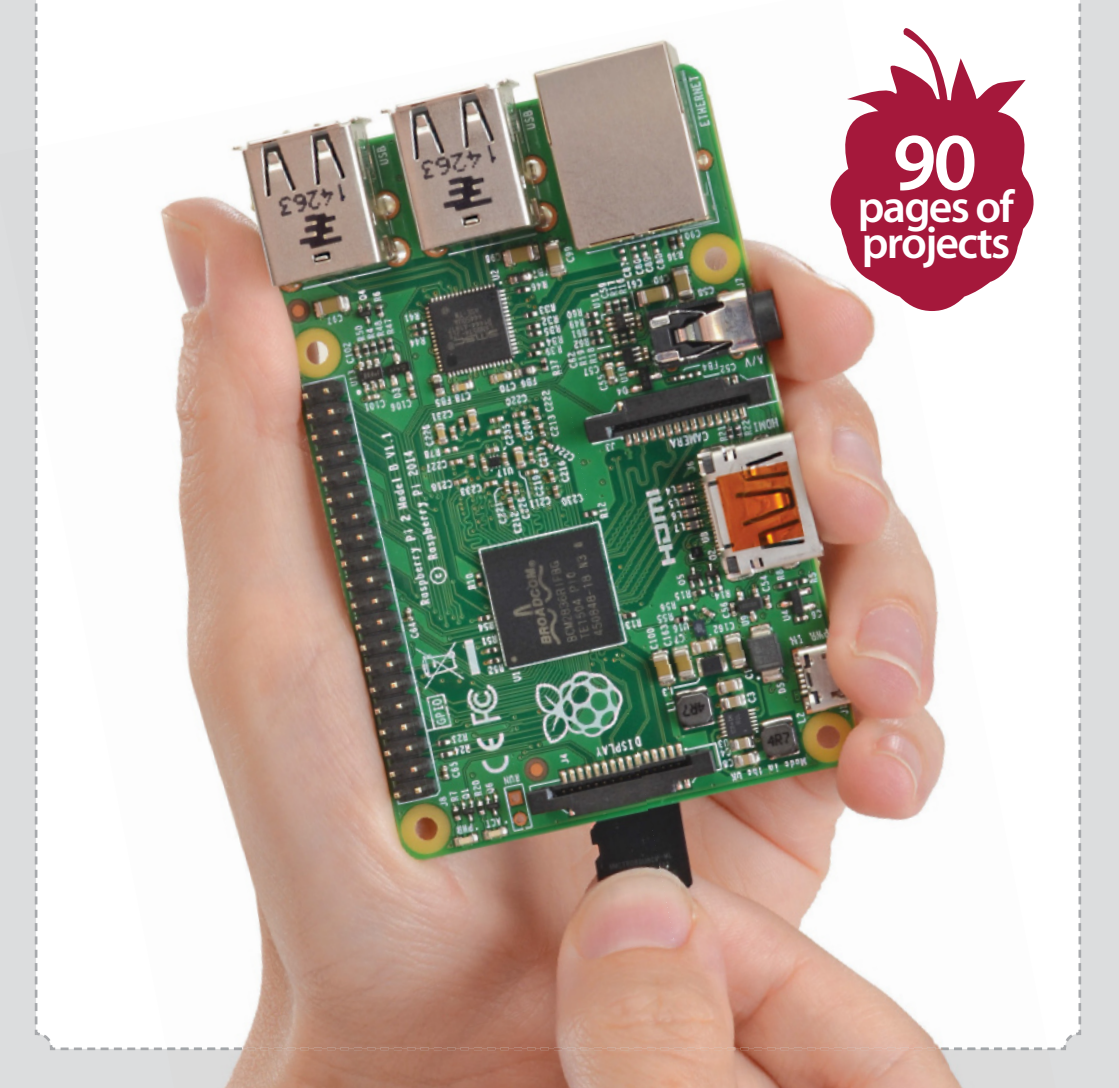
NEW

Raspberry Pi

The Complete Manual

The essential handbook for all Raspberry Pi users



90 pages of projects

Welcome to Raspberry Pi

The Complete Manual

The Raspberry Pi is one of the most exciting things to happen to computers since Steve Jobs revealed the iPad. As an educational tool, this credit card-sized PC has reignited interest in bare-metal computing in schools. As a platform for open-source software, it has also inspired millions of people to try Linux – many for the first time. Most exciting of all is the potential to incorporate the device into practical projects, as demonstrated by the tutorials in this newly revised edition of the Complete Manual to Raspberry Pi. So get creating!

Raspberry Pi
The Complete Manual

Imagine Publishing Ltd
Richmond House
33 Richmond Hill
Bournemouth
Dorset BH2 6EZ
☎ +44 (0) 1202 586200
Website: www.imagine-publishing.co.uk
Twitter: @Books_Imagine
Facebook: www.facebook.com/ImagineBookazines

Publishing Director
Aaron Asadi

Head of Design
Ross Andrews

Production Editor
Fiona Hudson

Senior Art Editor
Greg Whitaker

Designer
Perry Wardell-Wicks

Photographer
James Sheppard

Printed by
William Gibbons, 26 Planetary Road, Willenhall, West Midlands, WV13 3XT

Distributed in the UK, Eire & the Rest of the World by
Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
Tel 0203 787 9060 www.marketforce.co.uk

Distributed in Australia by
Network Services (a division of Bauer Media Group), Level 21 Civic Tower, 66-68 Goulburn Street, Sydney, New South Wales 2000, Australia, Tel +61 2 8667 5288

ISBN 9781785461651

Part of the
LinuxUser & Developer
bookazine series

IMAGINE PUBLISHING

Contents

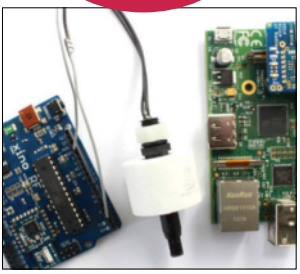
What you can find inside the bookazine

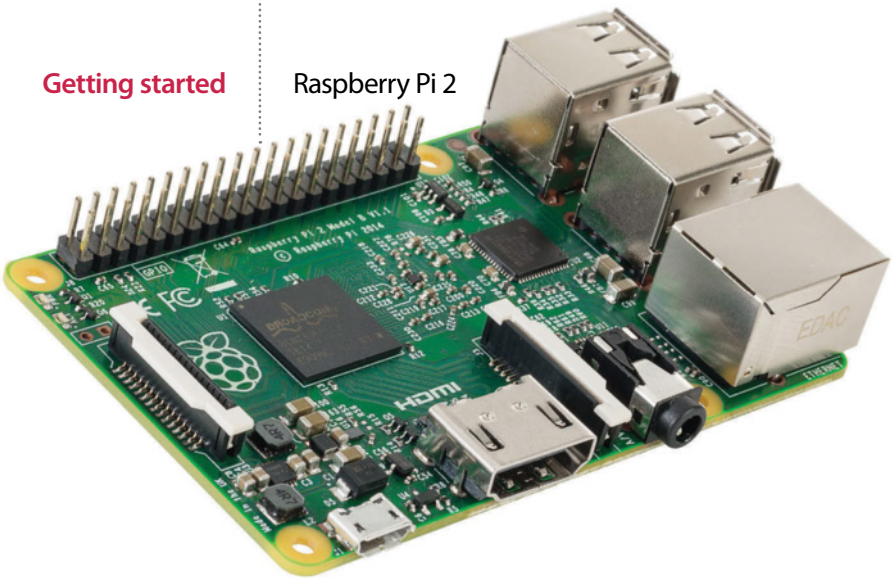
Getting started

Find out how you can make a Raspberry Pi-powered car starting on page 110

The projects

Code & create with your Pi!





Raspberry Pi 2

A super-charged Raspberry Pi that finally does everything you'd want it to, for the exact same price as the original

While the Raspberry Pi has enjoyed years of success, there's always been a couple of things a lot of users wanted. A slightly more powerful CPU that could handle day-to-day computing, more USB ports and maybe wireless to make connecting to the network easier.

The Raspberry Pi 2 solves most of these problems. As it uses the same board design as the Model B+, it has four USB ports, up from the two that were on the original Raspberry Pi Model B. More importantly, it has a much more powerful processor and more RAM, making it six times faster than the original Pi. Unfortunately, there is no Wi-Fi, but with the option to get a cheap dongle for the Pi 2 this is much less of a problem.

The new BCM2936 chip is the heart of the Raspberry Pi 2, a modified version of the BCM2835 chip from the old Raspberry Pi. Instead of having a single core, 700 MHz processor it's now a quad-core, 900 MHz beast that helps to make the Pi 2 a much more functional board. Whereas before you'd have had problems surfing the internet or writing a document, now the Pi 2 breezes through these tasks with ease and plenty of processor power to spare.

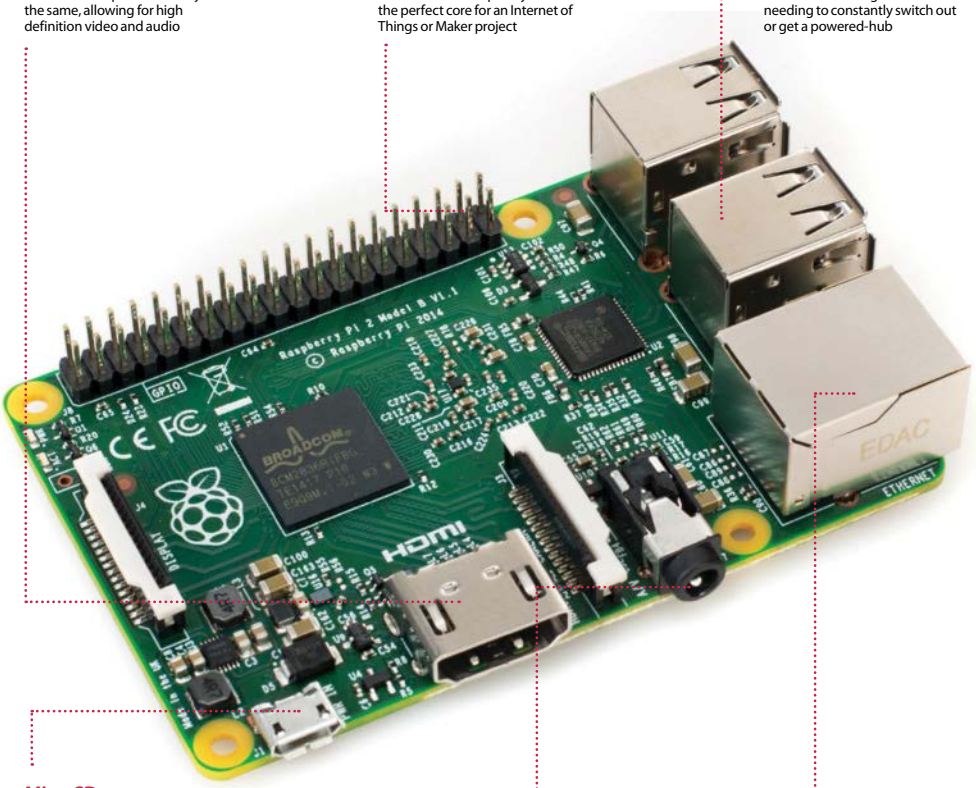
Otherwise it's still the same board as the Raspberry Pi B+. As well as the aforementioned four USB 2.0 ports, there's the Ethernet port for wired internet, a good-quality 3.5mm headphone jack for sound, a HDMI port for digital video and audio and a 40-pin GPIO port. This expanded GPIO port is fantastic for making your physical projects even more involved and complicated to do far cooler things.

For those worried about compatibility, all your old files and projects and such work just fine on the Raspberry Pi 2, and all you need to do is transfer them over like any normal files.

HDMI port
The Raspberry Pi's special ability is to decode 1080p video on the fly with very little problem, and the same tech still exists in the Pi 2. The HDMI port is basically the same, allowing for high definition video and audio

GPIO port
The 40 pins in the GPIO port give you a range of power and function slots to control a project or read more data from your surroundings. This makes the Raspberry Pi 2 the perfect core for an Internet of Things or Maker project

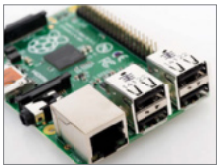
USB ports
The four USB ports give you much more flexibility with the Raspberry Pi 2, allowing you to easily add a keyboard, mouse, wireless dongle and external storage without needing to constantly switch out or get a powered-hub



MicroSD
Underneath the board is where the boot medium lives – the microSD card. Much smaller than the SD card of the original, it still holds the full operating system and allows the Pi 2 to be much smaller

Headphone jack
Need to listen to your Raspberry Pi privately? Connect it to a pair of portable speakers? The 3.5mm jack is still on the Pi 2, and is one of the higher-quality ones that was added to the B+

Ethernet port
The Pi 2 retains the wired network and internet connection that was on the Model B of the original Raspberry Pi. It still tops out at 100 MB, but that's plenty fast enough for the Raspberry Pi



"The Raspberry Pi 2 still won't be able to power a USB hub, so if you need to expand the compliment you'll need to get a powered-hub"

Raspberry Pi Model A+

Good things come in small packages: find out why the Raspberry Pi A+ is ideal for mobile projects.

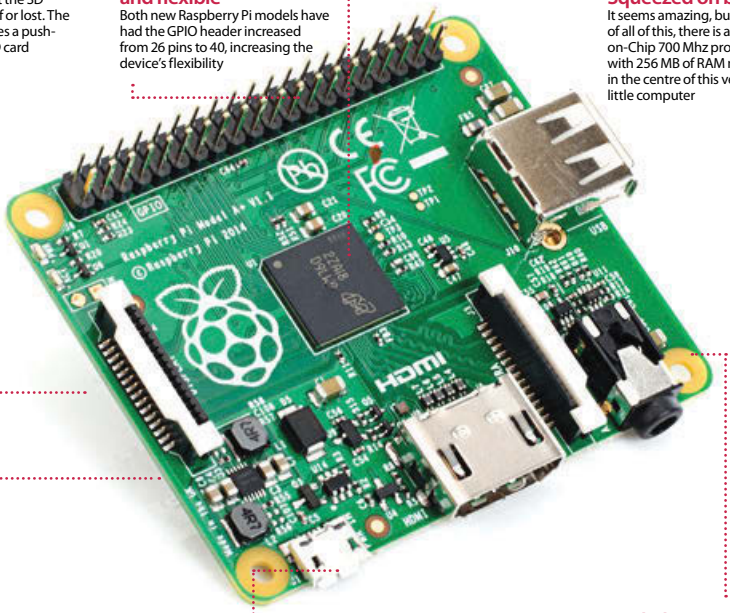
While the Raspberry Pi Model B+ is a step up from the Model B with its four USB ports, the Model A+ is smaller than its predecessor, weighing just 23g (down from 45g) and wielding one USB port. It's also limited to just 256MB of RAM on the SoC, compared to the 512MB enjoyed on the B+.

But don't think that all of this means that the A+ is inferior. Its 65mm length and lower weight is a clue as to how it can be used. The lack of an Ethernet port meanwhile, isn't a weakness, rather an illustration of the fact that this Raspberry Pi is designed not for media centres and print servers, but for projects where weight is a factor. Perhaps you'll mount it on an Arduino-powered robot, where its lower power requirement can be satisfied with a battery.

MicroSD Storage
No more worries about the SD card being snapped off or lost. The Raspberry Pi A+ features a push-push slot for a microSD card

Connectable and flexible
Both new Raspberry Pi models have had the GPIO header increased from 26 pins to 40, increasing the device's flexibility

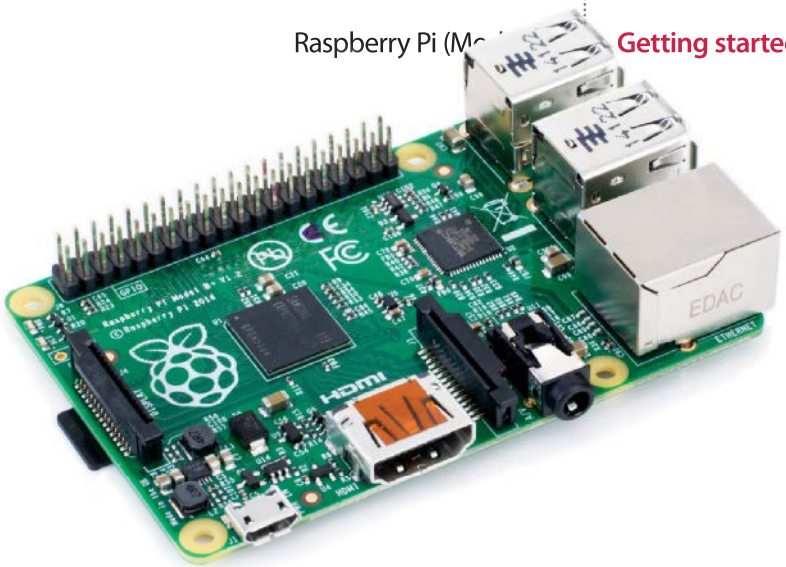
256 MB RAM Squeezed on board!
It seems amazing, but on top of all of this, there is a System-on-Chip 700 Mhz processor with 256 MB of RAM mounted in the centre of this versatile little computer



It's smaller!
The original Raspberry Pi Model A was a credit card sized 86mm in length. The A+ measures in at just 65mm long!

Repositioned power socket
With the micro USB power socket now beside the HDMI socket, it's easier to arrange and manage cables

It's lighter too!
Unbelievably, the Raspberry Pi Model A+ is just 23 g (0.81 oz), a reduction in weight of almost 50 per cent!



Raspberry Pi Model B+

The latest version of the original Raspberry Pi has some essential updates while still remaining largely the same. Is it worth the upgrade?

With the Model B+, an enhanced version of the Model B revision two, some of the hardware has been relocated into a more logical and tidier layout. The new layout lends itself to a much tidier work station or project innards. Everything is much more flush to the board without the USB ports or video connector sticking way out, and with all the ports along two sides rather than all four, the cabling can be a lot more tidy and the placement can be more flexible, and less awkward for specific setups.

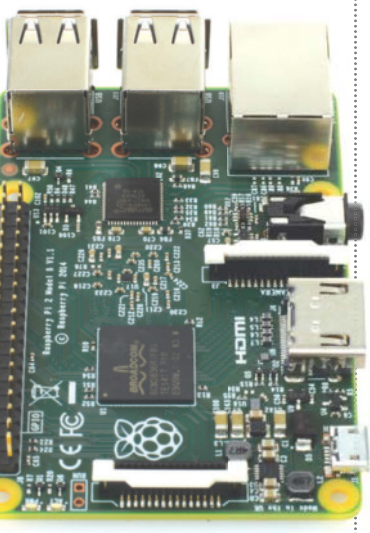
The inclusion of two more USB ports is a godsend to a lot of Raspberry Pi users who want to use a mouse, a keyboard, a Wi-Fi dongle and perhaps some external storage at the same time without needing a powered USB hub.

As well as two more USB ports there are 14 more pins added to the GPIO port, which still remains in roughly the same position on the side of the board. The extra pins are merely added on to the end of the original layout, meaning your old projects will still work just fine. You can now do a lot more with them, though.

The core of the B+ is fundamentally the same as the revision two of the Model B – it still has the same Broadcom ARM v6 system-on-a-chip along with the 512MB of RAM found in the revision two.

On the one hand, this is excellent. People getting into Raspberry Pi don't need to learn the hard way that all the online tutorials have stopped being relevant. Those who decide to upgrade don't need to learn anything specifically new and they can create a copy of their SD card that will work straight away in the B+.

Either way, the Model B+ is replacing the Model B as the flagship Raspberry Pi. For those still hanging onto the revision one Raspberry Pi, it's a great time to upgrade and get ahead of the curve.



The starter kit

There's more to your Pi than first meets the eye. Here are some vital peripherals to get you started

In order to get the very best experience from your Raspberry Pi, you're going to have to get hold of a few extras on top of the actual Raspberry Pi board itself. For example, you're going to need a keyboard and mouse with which to enter commands and navigate. While it's possible to do projects without a keyboard and mouse attached, you'll need them for the initial setup. An SD card is also an important purchase – it's where the operating system lives.

Perhaps you'll need a Wi-Fi adapter, or maybe just a length of network cable. Then there's the basic electronics side of the Raspberry Pi, what would you need to start some of the beginner electronics and control experiments? Clearly, there's more to the Raspberry Pi than some might think.

By 'peripherals', we mean other hardware that can be attached and utilised by the Raspberry Pi. They could be something as simple as a decent HDMI or they could be the latest, greatest bespoke gadgets that enhance your project capabilities.

There is an entire world of possibilities available for the Raspberry Pi; from robot arms to remote-controlled helicopters… The only limits are the hardware available and your imagination!

Did you know…

Most online retailers sell packages complete with all the accessories you might need – even pre-installed SD cards.

Keyboard and mouse

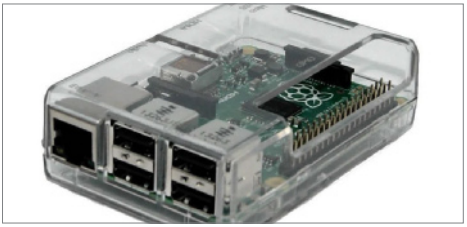
Let's start with the most basic of components, the keyboard and mouse. Generally speaking, virtually any USB keyboard and three-button scroll mouse will work with the Raspberry Pi, and although for some projects you won't even need a keyboard and mouse, you'll need them for initial setup.

SD card

Early Raspberry Pi computers required an SD card, whereas later models such as the Raspberry Pi 2 use microSD cards for storage. This is where your chosen operating system (such as Raspbian) is installed, and these can be bought in various sizes, pre-installed with the OS, or blank.

Power cable

The Raspberry Pi uses a standard micro USB connector for its power input, running at 5V. In most cases a micro USB to USB cable will suffice, of which one end can be plugged into your desktop computer's USB port. An Android phone charger should also work perfectly (5.25V 1500mA).



Case

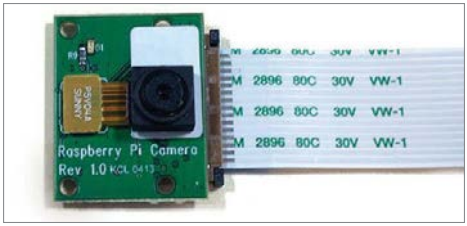
Securing your Raspberry Pi in a case will protect it and prevent the delicate GPIO pins from accidental damage. A case can also make your Raspberry Pi a more attractive or striking unit, perhaps as a media centre. Ensure you choose the right case for your Raspberry Pi model.

Video output

There are two video output ports, a HDMI port and an RCA Socket. HDMI is the primary video-output connector for most users, but the RCA video-out port can also be used to connect TVs, monitors or to a SCART cable. Remote access to your Raspberry Pi is also possible via SSH or VNC.

Powered USB hub

This is a custom designed add-on board that attaches to one of the Raspberry Pi's on-board sockets via a flexible cable. It's extremely small, but remarkably powerful, having a native resolution of five megapixels and supporting 1080p video. It's essentially a smartphone camera for the Pi!



Raspberry Pi camera board

This is a custom designed add-on board that attaches to one of the Raspberry Pi's on-board sockets via a flexible cable. It's extremely small, but remarkably powerful, having a native resolution of five megapixels and supporting 1080p video. It's essentially a smartphone camera for the Pi!

USB Wi-Fi adaptor

Using a USB Wi-Fi adaptor will bring flexibility to where you position your Raspberry Pi. Without a restrictive Ethernet cable, it could be used for more advanced projects where running a wired internet connection isn't a valid option. Just make sure you buy a Raspberry Pi-friendly Wi-Fi adaptor.

Set up your Raspberry Pi

Learn what goes where in your brand new Raspberry Pi with our easy-to-follow guide

While it looks daunting, setting up the Raspberry Pi for day-to-day use is actually very simple. Like a TV or a normal computer, only certain cables will fit into the specific slots, and the main job really is making sure you've got plugged in what you need at any one time. The Raspberry Pi itself doesn't label much of the board. However, most good cases will do that for you anyway – if you decide to invest in one.

Power adapter

The Raspberry Pi is powered using a microUSB cable, much like a lot of modern Android phones. It can be powered off a laptop or computer. But to make the most out of it, a proper mains adapter – like this one – is ideal

Monitor

The Raspberry Pi is capable of displaying a 1920 x 1080 output – otherwise known as 1080p. Some modern monitors allow you to plug HDMI straight into them, just like TVs do. However you may need an adapter in some cases

USB hub

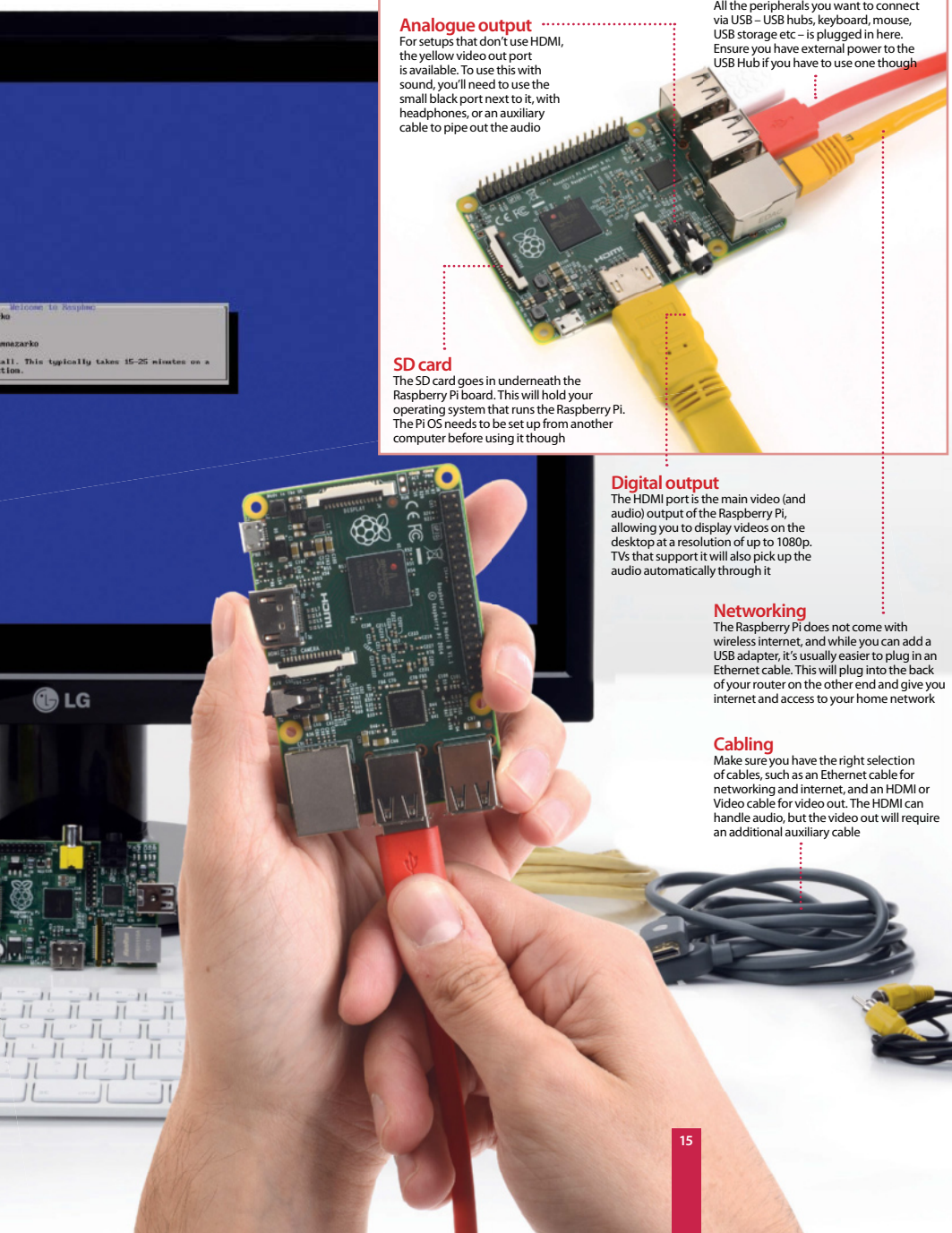
There are only a limited number of USB ports on a Raspberry Pi (just one, if you have Model A). To get around this you will need a USB hub. It's important to get a powered one, as the Pi cannot supply enough juice on its own

Case and accessories

A case is not necessary to use the Pi correctly, but a decent one can keep it well protected from dust, and make it easier to move while in operation. You will need an SD card, however, of at least 4GB

Keyboard and mouse

Like any computer, you'll need a keyboard and mouse for any standard PC-style operations you do with the Raspberry Pi. The more basic the keyboard, the better; same with the mouse, as some special ones need additional software



Analogue output
For setups that don't use HDMI, the yellow video out port is available. To use this with sound, you'll need to use the small black port next to it, with headphones, or an auxiliary cable to pipe out the audio

USB
All the peripherals you want to connect via USB – USB hubs, keyboard, mouse, USB storage etc – is plugged in here. Ensure you have external power to the USB Hub if you have to use one though

SD card
The SD card goes in underneath the Raspberry Pi board. This will hold your operating system that runs the Raspberry Pi. The Pi OS needs to be set up from another computer before using it though

Digital output
The HDMI port is the main video (and audio) output of the Raspberry Pi, allowing you to display videos on the desktop at a resolution of up to 1080p. TVs that support it will also pick up the audio automatically through it

Networking
The Raspberry Pi does not come with wireless internet, and while you can add a USB adapter, it's usually easier to plug in an Ethernet cable. This will plug into the back of your router on the other end and give you internet and access to your home network

Cabling
Make sure you have the right selection of cables, such as an Ethernet cable for networking and internet, and an HDMI or Video cable for video out. The HDMI can handle audio, but the video out will require an additional auxiliary cable

What you'll need…

Raspberry Pi downloads
www.raspberrypi.org/downloads

Did you know…

The official distro for the Pi is called Raspbian. That's what we recommend, but there are other options.

Install a distro

We take a look at some of the key aspects involved in installing a pre-built OS

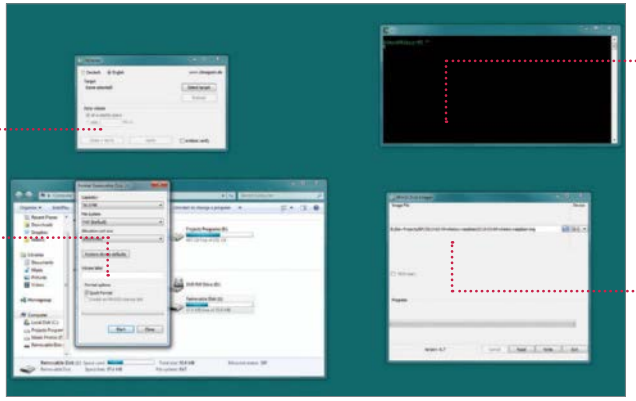
With its small size and cheap price, many people might be fooled into thinking that the Raspberry Pi is only usable for basic tasks, and learning to program on. While one of the primary goals of the Pi was to increase computer literacy at a lower level rather than just learning how to create Excel spreadsheets, the Pi has many other great uses.

As the Raspberry Pi is essentially a mini PC, with an HDMI and analog TV output rather than a traditional monitor connection, it can perform many common tasks that a laptop or desktop is often used for. While it doesn't really have the processing power or RAM to run the latest version of Windows, there are other options.

There are a wealth of fully fledged operating systems, many forked from their desktop big brothers that have been optimised specifically for the Pi. One of the most popular of these is Raspbian, which is a port of Debian. Debian is a key part of the Linux ecosystem, and many other popular open source distributions are forked from the Debian source code. The original Debian was released in 1993, and it's come a long way since. Raspbian needed work to get performance levels up to standard, as the Pi uses the older ARMv6 architecture. It's now a great everyday desktop.

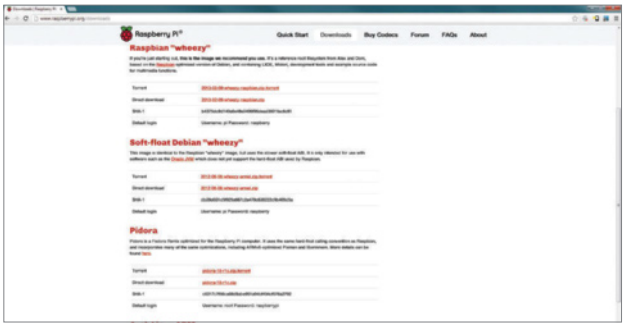
Card speed
It's a good idea to get a reasonably fast SD card to keep your system running smoothly. Class 4 or above is best

Card format
Before you copy your OS image, you'll need to make sure the SD card is formatted into the FAT32 file system



Command line
If you are using OS X or Linux, then its likely you will use the command line to install your prebuilt operating systems

Automated tools
There are a couple of graphical tools available which make installing an image onto an SD card easy



Obtaining OSs

01 One of your first questions may be "where can I find some operating systems to download?". Most of the common images can be found on the main Raspberry Pi site: **www.raspberrypi.org/downloads**. These are stable and well tested systems worth investigating.

Unzipping

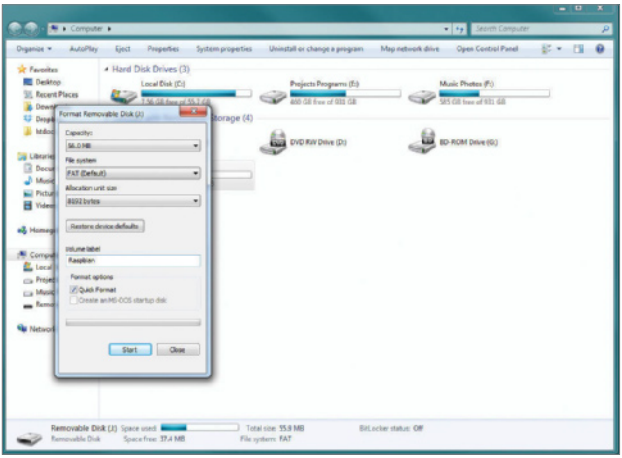
02 When you've downloaded your image, the first thing you'll most likely need to do is unzip it. This can be done in Windows by right clicking and choosing 'extract'. In OS X, just double click to extract the files.

OS Format

03 Within the zip you'll find a file with a .img or .iso extension. These are the equivalent of a 'snapshot' of an installation CD or DVD. Simply copying the file to the SD card won't do anything; you'll need to use a program to extract it.

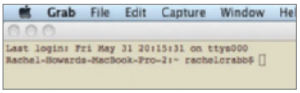
SD card format

04 The SD card that you'll boot from needs to be blank, so make sure there is nothing important on it first. You'll also need to format it to use the FAT32 file system. This is a common system, used by most USB sticks and cameras.



Formatting the card

05 In Windows, to format the card simply insert and wait for it to mount. Then click on 'My Computer' and then right click on the cards icon. After that choose format and then 'FAT32' from the drop-down menu.



Using the terminal

06 If you are using OS X or Linux, then you'll have to use the terminal to copy the image. In OS X, the Terminal app comes installed by default, and most Linux versions come with one in some form or other. It may be referred to as the 'console' or 'command line'.

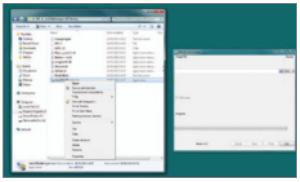
DD command

07 The command you need to use is called 'dd'. This is entered in the format of 'sudo dd bs=1m if=[img] of=/dev/[sdcard]'. Eg:

```
sudo dd bs=32m if=/Users/
rachelcrabb/Desktop/ArchLinux/
archlinux-hf-2013-02-11.img
of=/dev/disk1
```

Win 32 Disk Imager

08 Windows users can use Win32 Disk Imager. Once you've downloaded the tool, simply right click on the .exe, and choose 'run as administrator' and follow the prompts. When the installation is complete you can put the SD card in your Pi. Easy!



What you'll need…

Raspbian
www.raspberrypi.org

Command line basics

Learn an alternative way to control your Raspberry Pi by using the command line and your keyboard

Did you know…

The command line remembers your last commands. Simply use the Up and Down arrows to use them.

We've probably all been there with the Raspberry Pi. You've installed Raspbian or another Raspberry Pi OS to your SD card and you've rushed through the setup script or not quite done your research. You start the operating system and… you end up at a command line. The first step here is to not panic: this is perfectly normal. It may just be a bit of a foreign concept to you, only seen in films with streetwise hackers who want to bring down 'the system'.

The second step, at least in Raspbian's case, is simply to type:

```
$ start x
```

That's it. Raspbian will load up the desktop and you can start using the mouse again. Quick and painless in this case, and in that of many other operating systems as well. What you've done is use a command, specifically in this case to start the X server. The X server handles the graphical interface and can be turned off by default on some Pi systems.

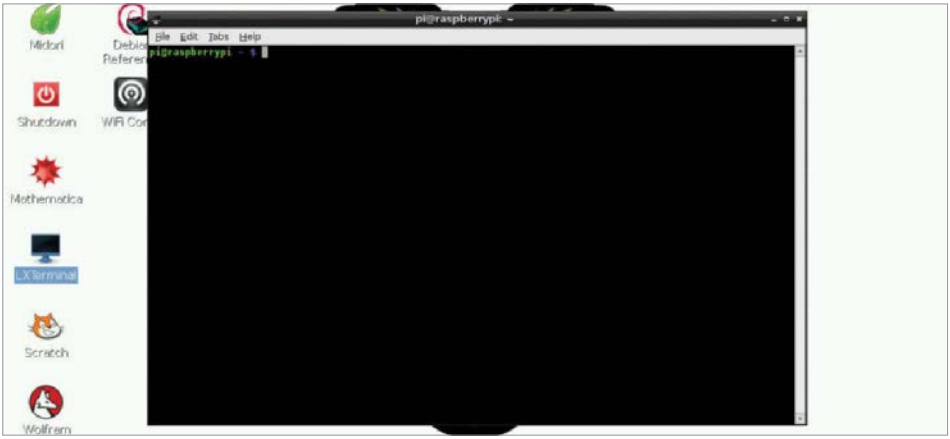


Fig 1: The terminal emulator allows you to access the command line while still being in the desktop environment

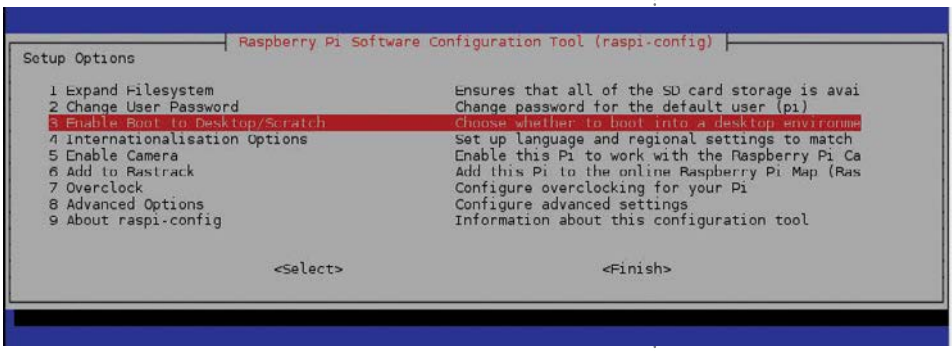


Fig 2: Access raspi-config to change settings such as boot to desktop or adding a camera module

A new world

Getting your Raspberry Pi into the desktop isn't the only thing you can do on the command line, though. There's a whole world of functionality built into the command line; in fact, most of the graphical programs you're using are just executing these commands in such a way. You don't have to leave the comfort of the desktop environment to perform these commands either, as all Raspberry Pi operating systems will come with an application known as a terminal emulator.

This creates a window where a command can be written in the same way that we launched the desktop, and use the exact same commands (Fig 1). On Raspbian, look for the app LX Terminal in the Accessories section of the menu and click on it. If you've had to use **start x** to get into the desktop, then we can now fix that before continuing. In the terminal, enter:

```
$ raspi-config
```

Here's the initial setup screen (Fig 2). From here you can enable the desktop on boot (Fig 3), and even update the firmware and add support for the official Raspberry Pi camera module. This allows you to modify Raspbian without having to reinstall again.

You won't be using those two commands very often, though, so is there practical use for delving into the command line? Very much so. For starters, Raspbian doesn't have an official package manager. This is a program that allows you to browse the available software for the operating system, similar to the Pi Store.

However, there's other software available to Raspbian that you can't get through the store. You also can't specifically update the Pi software either, and all of this can be fixed using the command line.

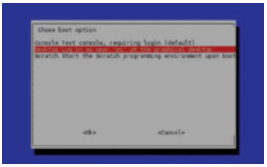


Fig 3: Change the default selection to desktop for the next time you use the Raspberry Pi

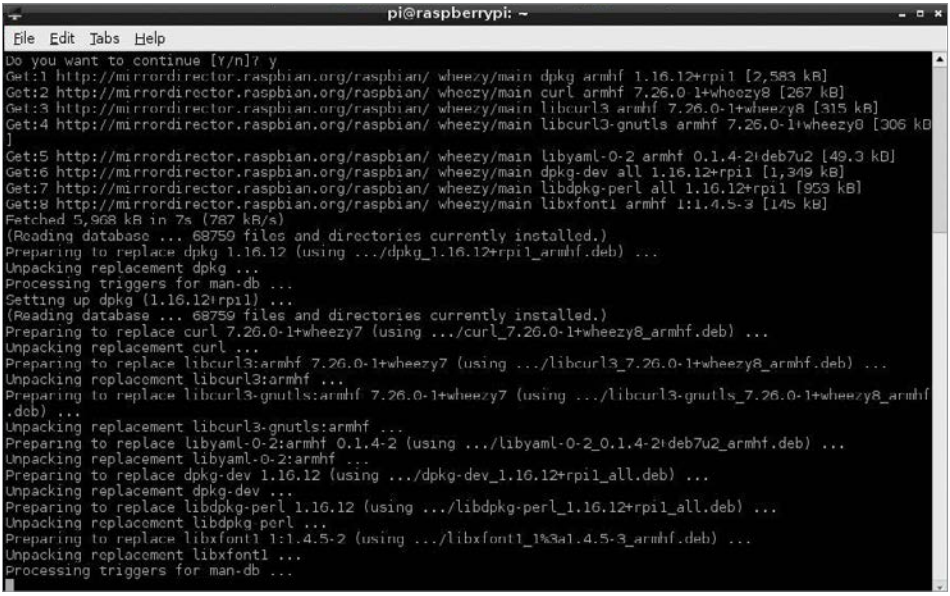


Fig 4: Upgrade your system and files, and have the latest updates and bug fixes in the process

Software for all

The first thing you'll want to do is let Raspbian know exactly what's available online. It's a very simple task: all you need to do is:

```
$ sudo apt-get update
```

This will run down a list of online repositories (or repos) that contain the software that Raspbian uses. Once it's finished, the command-line prompt will pop up again waiting for your next command. As this is the first time you've done it, you'll likely need to update the current software on your Raspberry Pi. You can do that with the command:

```
$ sudo apt-get upgrade
```

It may ask you to confirm the upgrade, in which case type 'y' and then press Enter. What we're doing both times is using the command-line package manager Aptitude (apt-get) to first check the repos, and then upgrade packages according to that (Fig 4). The first command, sudo, allows it to run the apt-get task as an administrator, and is used in a lot of other command-line operations. To install software you use **install** instead of update or upgrade, followed by the name of the package. For example, with the mathematical programming language, you can install it with:

```
$ sudo apt-get install wolfram-engine
```

Did you know…

You can use the Tab to complete commands. Just start typing and hit tab. If you like what you see hit Enter to finish!

Move and create

Installing and updating are just a couple of the many things you can do in the command line. You can also browse the entire file system, move files, create folders and delete items. All of these are very simple operations.

When you first open the terminal, it will open up in your home folder. While you can't specifically tell that it is, you can display exactly what kind of files are in the directory with (Fig 5):

```
$ ls
```

The tilde sign (~) is used to denote the home folder and can be used for navigating around the file system. To navigate, we'll be using the **cd** command, followed by the location you want to move to. This can be done like so:

```
$ cd /home/pi/Downloads
```

This will move you to the Downloads directory. As we were starting off in the home folder to begin with, we actually only needed to do this:

```
$ cd Downloads
```

It's context sensitive and knows to look in the directory it's already in. There's another trick you can use so you don't have to remember the exact name of the path – the command line or terminal will try to autocomplete the phrase if you press the Tab key while something is only partially typed. Try the cd command again, but try pressing Tab when you've only written 'Down'.

Finally, there are some quick commands you can use to manipulate files. Individual files can be copied using the command **cp**, followed by the filename and the new location like so:

```
$ cp file.txt ~/Documents/file.txt
```

You can also use this to rename files by doing:

```
$ cp file.txt otherfile.txt
```

The original file can then be deleted by using the **rm** command:

```
$ rm file.txt
```

Want to create a new folder? Use cd to move to the directory you need to add a folder to, and then use **mkdir** followed by the name you want to give the folder:

```
$ mkdir NewFolder
```

There's a lot more you can do with the command line, but these are the very basics. As you use Linux more and more, you'll be confronted with tasks that need the command line, and through this process you'll learn just how much can be accomplished when you work like a streetwise movie hacker.

Did you know…

You can always return to your Home folder in the command line by typing cd ~ and pressing Enter.

"The command line or terminal will try to autocomplete the phrase if you press the Tab key while something is partially typed"

Fig 5: There are many simple command-line tools that can help you browse and use your system

The Raspbian desktop

Although the Raspberry Pi's Raspbian operating system is closer to the Mac than Windows, it's the latter that the desktop most closely resembles

It might seem a little alien at first glance, but using Raspbian is hardly any different to using a Windows desktop. There's a menu bar, a web browser, a file manager and you'd expect, you simply open the Menu and click the applications you want to get started…

Menu button

The Windows-like Menu button in the top-left corner displays a list of programs and options. The main categories are Programming (where you'll find appropriate tools), Internet (browser and online resources), Games (Minecraft Pi is pre-installed), Accessories (an assortment of utilities), and Preferences (system tools). Programs downloaded from the Pi Store will appear in the appropriate category, while the Run launches a command-line interpreter, just like the one in Windows. Use Shutdown to switch off, logout or restart your Raspberry Pi.

Task Bar

Stripped across the top of the screen is the Task Bar, upon which the Menu is situated. To the right of this are shortcuts to the Epiphany browser, the File Manager (PCManFM), LXTerminal for inputting text-based commands, the Wolfram Mathematica computational software and the Wolfram Language programming application. Next to these shortcuts, you'll find that any open applications are docked, while in the right-hand corner you'll find the clock, current CPU load, volume status and control, and the network status.

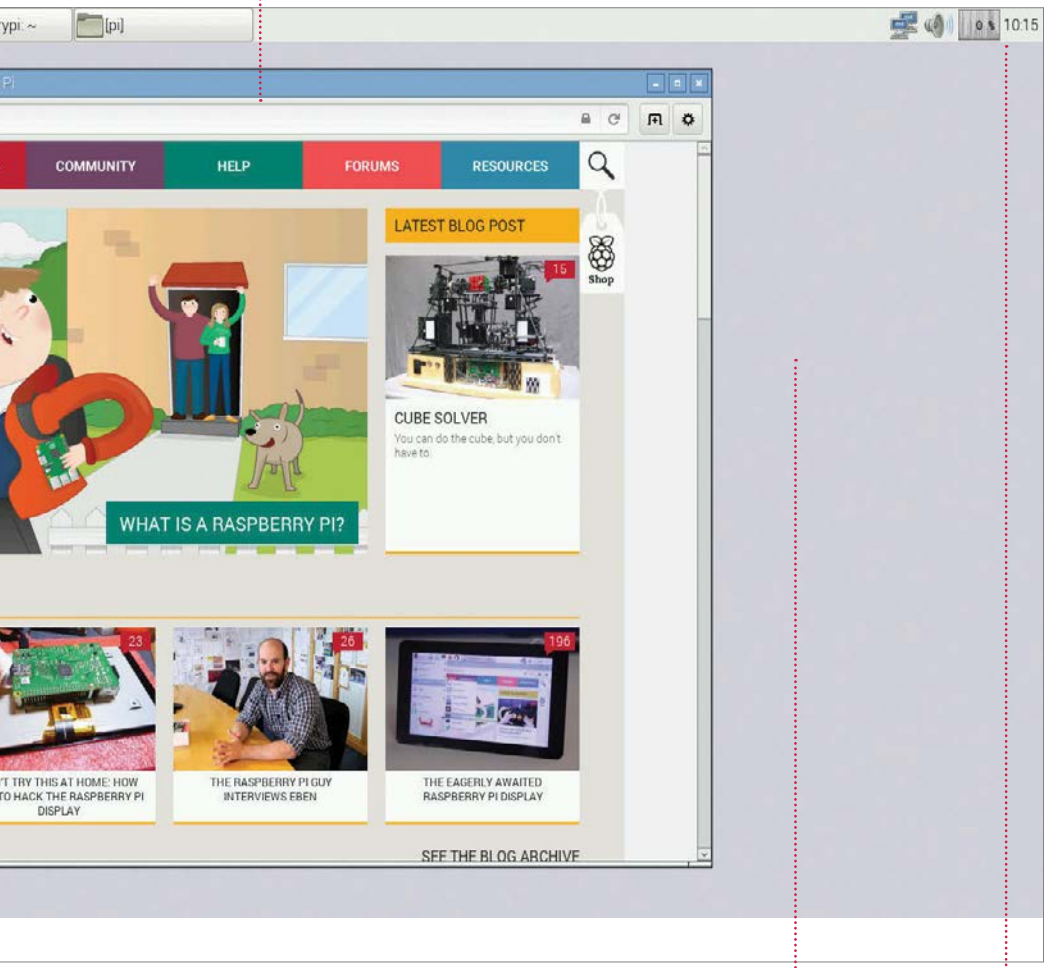
File manager

No computer would be complete without a file manager, which can be opened by clicking the Task Bar shortcut. Files can be copied, renamed and deleted by dragging to the trashcan on the desktop (or by simply highlighting and tapping Delete on the keyboard). You can also create tabbed windows in the file browser or open further ones. Folder locations can be bookmarked for easy access and files themselves can be viewed as icons or in a detailed list.

"It might seem a little alien at first glance, but using Raspbian is hardly any different to using Windows"

Web browser

Empathy is the default Raspberry Pi browser, although if you don't like it you can add a new one from the Pi Store. As with any browser, web addresses are typed into the address bar and standard navigation buttons are provided to navigate back and forth through webpages. Empathy supports multiple tabbed windows and features private browsing and the ability to clear browsing data. Click on the Options icon at the end of the tool bar to access these functions.



Pi Store

You can access the Pi Store from the Menu, or from a link on the **www.raspberrypi.org** website. Much like Steam, Google Play or the App Store, the Pi Store requires you sign up for an account (with IndieCity), and you'll need to sign in before you begin shopping for free and paid-for software. Any games and apps downloaded and installed from the Store will appear under the My Library tab, and some of those can only be launched from here.

Status and Time

On the far right side of the Task Bar is the Raspbian equivalent of the Windows system tray, where you'll find information concerning your Raspberry Pi's status. In the corner is the clock, which when clicked will also display the current date; if incorrect, you can adjust it here. To the left of this you will find a graph displaying current CPU load, the device volume control, and finally, confirmation that your Raspberry Pi is online.

What you'll need…

Raspbian:
www.raspbian.org

Master the Config tool

Tell your Raspberry Pi how to behave using the powerful built-in config tool

Did you know…

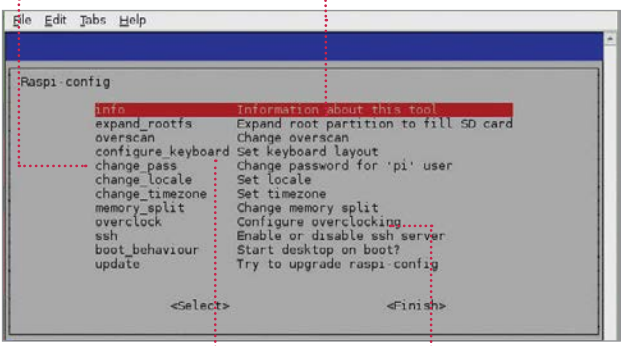
You can access the Config tool anytime by typing 'sudo raspi-config' from a terminal window or the command line.

The 'RasPi Config' tool allows configuration of your system that would otherwise be trickier in the Linux environment and it's the first thing you'll see when you install Raspbian. Why? Tasks such as setting the date and time or regional settings for your keyboard are often done in a command-line interface with no dialogs, no additional help – for a new user, this is a nightmare.

There are some further specifics for the Pi and Raspbian itself, such as: the ability to easily enable overscan for your TV; change the split of memory to the computer/graphics card or even overclock your system to make it a little faster; enable remote SSH access to the system; stop the system booting into the desktop environment among other things. The Raspi Config tool takes the pain out of the process and puts real power at your fingertips.

Change password
Change the password for your default 'pi' username to make it something more personal or easier to remember for you

Expand roots
Allows you to very quickly and easily change the partition of the roots to fill the SD card completely



Configure your keyboard
Set the correct keyboard up – there are many different layouts. Using the wrong one can be annoying

Overclocking
Allows you to quickly and easily overclock your Pi to give you some extra speed and power with little risk

Open the raspi-config tool

01 Start by double-clicking the LXTerminal icon on your desktop. This will start the command prompt, where you'll be able to run the config tool. To do this you'll need to run a command.

```
sudo raspi-config
```

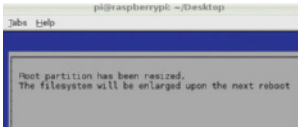
When asked for your password, you won't actually see it being typed.

When you've typed the password and pressed Enter to submit it, the config screen will be shown to you. There are a few settings of particular interest that we'll cover in this section, although they all have their uses in the running of your Pi. Some of the settings in this menu are important and some are irreversible, so use them with caution!

Expand the root file system

02 By default, the Raspbian root file system will be 2GB – this is done so that the image provided for it can fit on as many different SD cards as possible.

If your card is larger the 'expand_roots' option will make the OS use the entire space. Upon using this option, the command will be executed immediately. The operating can take some time. Reboot your system to see the changes.



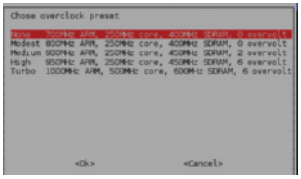
Configure your location

03 Locale is the language and regional settings that your Pi is using – while this generally has little impact on what you'll see, it is also responsible for any default currency settings, etc, so could prove to be an irritant at a later time if wrong.

Upon selecting the option, you'll be taken through a wizard. Use the arrow keys to check the built locale before building more (it takes a while). Timezone will take you to a tzdata screen where you can adjust it.

Overclock your Pi

04 You can set the clock speed and voltage of your Pi to several different presets. Setting the clock speed and voltage at higher rates than the specification may cause instability, so do so in small increments and ensure good airflow around your Pi.



If you see any noticeable instability, run this wizard again and set the clock speed back down to something slightly lower – repeat until your system is completely stable. For the scope of this tutorial, a Modest/Medium overclock is recommended – it seems to give a little extra performance with no noticeable side effects. It is also recommended to reboot your system after making this change. Hold the Shift key to temporarily disable overclocking.

Change the memory split

05 Changing the memory split of the Pi allows you to give either the system or the graphics processor a larger amount of memory.

The value you give to it must be either 16/32/64/128/256. Here are our recommendations:

32MB GPU memory for basic distro usage where video and 3D rendering aren't required.

64MB GPU memory for desktop use that requires video playback or have 3D effects enabled.

128MB GPU memory for graphical applications and games that do extensive multimedia or play 3D rendered games.

For most people, a 64MB split for graphics will suffice.

Change boot behaviour

06 By default, the Raspbian distro will boot into a command-line interface, whereby you have to first log in as 'pi'.

If you then want to run a window manager (in this case, it's called 'X'), you have to give the system a command to let it know that's what you want to do.

For a lot of people this isn't really ideal since command lines scare them. Because of this, there's an option to start X automatically, on boot. Set this option to 'Yes' to enable this behaviour by default.

You can obviously revert this at any time to return to a text-based login where you have to start manually:

```
startx
```

Turn overscan on and off

07 You may have noticed one of two behaviours if you're using your Pi with a modern HDTV.

There is a black border the whole way around the image output by the Pi – it just doesn't fit correctly. This is caused by underscan.

If you can't see the edges of your screen to get to them you're suffering from overscan.

If you have the former issue, you may need to either turn on overscan, or enable a 'zoom' mode or similar on your TV.

If you have the latter issue, you need to turn overscan off so that you can see the edges.

Update raspi-config

08 The raspi-config tool receives updates from time to time. This is generally to either add more features or fix small bugs, or both!

It's not a bad idea to run the updater when you use the tool – before you start changing any system settings. While it's much more likely that it'll be updated to look better or do more things, it's not impossible there could be miscellaneous bug fixes hidden within that would otherwise cause you some grief.

Remember, though, when you're trying to update your copy of the raspi-config tool, you'll need an active internet connection, either through an LAN cable or wireless dongle. Without them, it's never going to get any newer. Always try to make sure you're on the latest version.

What you'll need…

Any Raspberry Pi distro
www.raspberrypi.org/downloads

Wi-Fi dongle or Ethernet

Did you know…

If your Raspberry Pi is going to be placed near your Internet router, all you need to do is plug in an Ethernet cable.

Get online

To access a world of utilities, apps and resources you need to get online. This is how to do it…

The easiest way to get online is to buy a Raspberry Pi Model B+, as it comes with an Ethernet socket. The Model A not only lacks the Ethernet port, but is handicapped by only having one USB port. That means you will have to buy a power USB hub in order to get online. Back to the Model B+ though and to get online, simply plug an Ethernet cable into the socket on the Pi and connect it to a similar port on the back of your internet modem/router. Turn your Pi on and launch the desktop, then double-click on Empathy and you should see the internet appear (main image). To check that it's working, look at the lights on the Pi itself. The red power light should be on. Above this is the green light that flickers when accessing the SD card. Below the power light are the three Ethernet-related lights. Note that the Model A does not have these LEDs because it doesn't have the Ethernet socket. The middle light is green and comes on when it detects a Full Duplex LAN connection. This means it is able to send and receive data to the internet. The next light is green and flashes when actually accessing the internet by sending or receiving data. The last light is yellow and will come on and stay on when a 100Mb LAN connection is detected.

Fig 1: With a Wi-Fi dongle attached, run the utility and scan for available networks

Fig 2: Enter the pre-shared key in order to connect to your home router

The Wi-Fi option

If you aren't close enough to the modem/router to be able to plug in the Ethernet connection, or you simply have a Model A, then a powered USB hub is required. This plugs into a USB port on the Pi. You can then plug a Wi-Fi dongle into this. Boot up the Pi and launch the desktop. Then double-click on the Wi-Fi Config icon. You should see a name for the dongle in the Adapter section. Click on Scan to look for networks and a list of those found should appear (Fig 1). Double-click on the one you want to connect to and the details for it will be listed. Almost all home networks use a network key, which is usually written on the modem itself. Click on PSK, which stands for pre-shared key, and type it in (Fig 2). Then click on Add. It will process this, then associate the connection and

then finally, a new IP address for the Wi-Fi connection will appear. If you click on the Manage Networks tab, the network will now be listed and have an Enabled radio button active. To get on the internet, simply launch Empathy and you'll be connected. The Wi-Fi utility will remain running on the bottom right of the panel. If you right-click on the Wi-Fi icon you will see options to Disconnect or Reconnect, event history and the results of the most recent scan. Click on Status to see how it's performing.

Checking the connection

To check that the Pi has a valid internet connection, double-click on LXTerminal. Enter this command:

```
ip addr
```

You should see a list of numbers, with the bottom line starting 'inet' and then the IP address of the Pi connection (Fig 3). Typically this is something like 192.168.1.11 and this shows that the connection is working because the Pi has been assigned an IP address based on the one used by your internet modem/router. If this doesn't come up then there may be a problem at the router end. The modem/router should be running a DHCP server and when the Pi connects to it, it will be given the IP address. If it isn't running then nothing else connected to it will be able to access the internet either. Use the web interface with another device to log onto 192.168.1.0 or whatever is your modem's actual IP address in order to check that the DHCP server service is turned on. Finally, in the terminal, type:

```
ping google.com
```

Sharing a connection

If you don't have a Wi-Fi dongle, a powered USB hub or a long enough Ethernet cable, but do have another computer connected to the internet, there's another way of getting access. On a Mac, connect it to the Pi via a USB or Ethernet cable. Launch System Preferences; under Internet & Wireless, click on Sharing. Click on Internet Sharing, then select Wi-Fi (or AirPort) as the connection type to share, and select how the Pi is connected to your Mac (Fig 4).

On a Windows PC, go to Windows Explorer>Networking> Networking and Sharing Center>Change Adapter Settings.

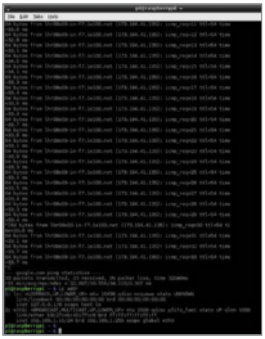
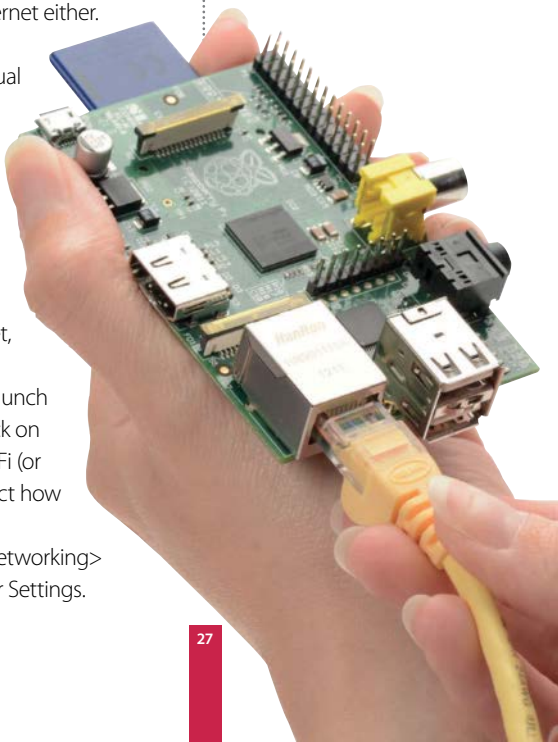


Fig 3: If it doesn't look like the connection is working, there are some easy ways of checking what's going on

Fig 4: Both Windows and Mac computers can share their internet connections with a directly-connected Pi



What you'll need…

Apt command help page:
http://linux.die.net/man/8/apt

Apt-get help page:
http://manpages.ubuntu.com/manpages/lucid/man8/apt-get.8.html

Did you know…

If you press the Tab key the command line will attempt to auto-complete your command for you – just press Enter to finish.

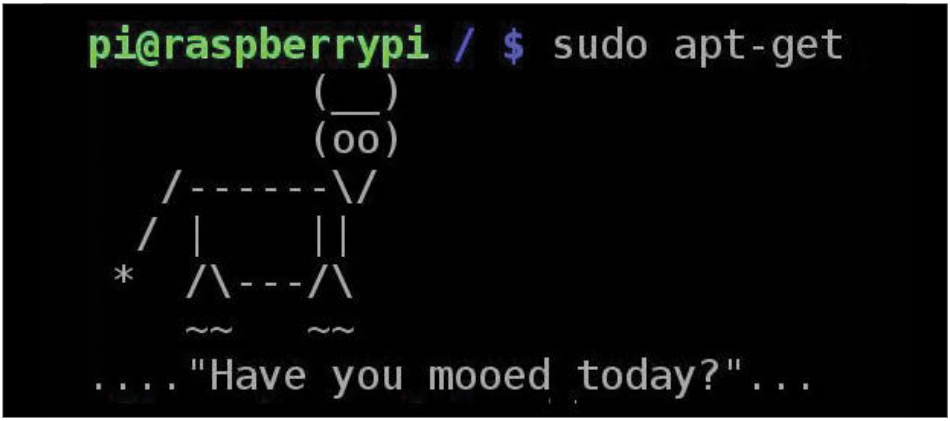
Install and use packages

The Raspberry Pi is great, but it's made better with the software you install onto it

On its own, the Raspberry Pi is a near-perfect mini computer. It already contains a wealth of educational software, a few games, some programming utilities and a number of system tools. But, as with most computers, this is only the tip of the proverbial iceberg. By installing more programs, you can do much more.

These programs, known as packages, are as wide and as varied as the developers who originally designed them. In Linux, if there's a need for a particular program, then someone develops one. They then put it out to the world and make the source code freely available, hence 'open source'. Once the program has been tested, it will eventually make its way onto one of the many remote servers for that particular Linux distro.

These remote servers, called repositories, or repos, contain all the elements of the package in order for it to be downloaded and installed onto your system. The process is very quick and easy once you know how it's done…



Update and upgrade

01 Getting hold of a package on the Raspberry Pi involves dropping into the command-line terminal, via the LXTerminal icon on the desktop, and entering a few commands. But before we do that, we need to make sure the system is up to date. Enter the following into the terminal:

```
sudo apt-get update
sudo apt-get upgrade
```

Or…

```
sudo apt-get update && sudo apt-get upgrade
```

Search for a package

02 The apt-get command (Advanced Package Tool) is the key to downloading and installing packages on the Raspberry Pi. In the previous instance, we updated the existing packages and system, upgraded any that needed it, and updated the current package list. Now, let's search the list of server packages for available games.

```
apt-cache search game | less
```

Apt searching

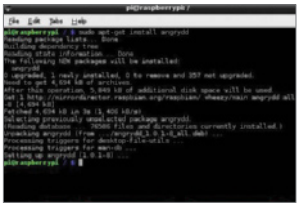
03 The current list you find yourself in is the name of all the packages labelled as 'games' from the available server. In the list, the part before the hyphen tells you the name of the package, which is what you will need to know to be able to install it. Use the arrow keys up/down to navigate; press 'Q' to exit.

Installing a package

04 Using the up and down arrow keys, navigate the list. If you find something you like the look of, say Angry Drunken Dwarves, remember the name of the package, in this case 'angrydd', and press 'Q' to exit the list.

To install the package, enter the following in the terminal:

```
sudo apt-get install angrydd
```



Executing the package

05 The result of the previous command should be the successful download and installation of the game, Angry Drunken Dwarves. To execute the newly installed package, you can either run it from the LXDE Menu under Games>Angry Drunken Dwarves, or by typing in the following into the terminal:

```
angrydd
```

Remove a package

06 This installing of packages is perfectly fine, and you can see just how powerful a command Apt really is. But, what if you want to remove a package?

Using the Apt command again, let's say we want to completely remove all trace of Angry Drunken Dwarves from the Raspberry Pi.

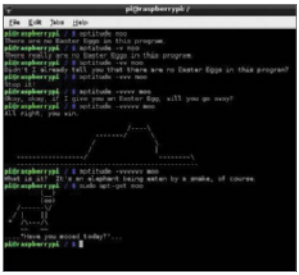
```
sudo apt-get --purge remove angrydd
```

Enter 'Y' to accept the removal.

Apt Easter eggs

07 The Apt command is a shorter, non-menu-driven variant of the Aptitude command. This command has a long history in Linux, and as a result has some rather special 'features', also known as Easter eggs. Purely for a little bit of fun, type in the following commands and see the results:

```
aptitude moo
aptitude -v moo
aptitude -vv moo
aptitude -vvv moo
aptitude -vvvv moo
aptitude -vvvvv moo
aptitude -vvvvvv moo
sudo apt-get moo
```



Man the Apt command

08 As you can see, there is more to the simple Apt command than what first meets the eye. There are many different sub-commands that you can run, and many different variations in which to run them.

If you want to see what else the Apt command can do, enter the following:

```
man apt
```

What you'll need…

Synaptic:
www.nongnyu.org/synaptic/

Use graphical installations

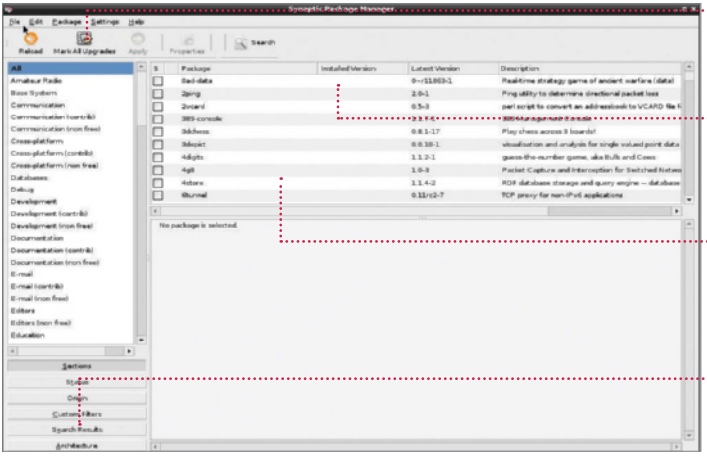
Would you prefer a graphical interface to install new programs? If so, then read on…

Did you know…

Synaptic has access to the same repo as via the command line as demonstrated on the previous two pages.

If you're new to Linux, you may find using its built-in Apt package management tool a bit intimidating and confusing. The apt-get command is used for installing applications through the internet, connecting to the remote servers – called repositories – which house the programs as packages. But it is used through the terminal command prompt, which can be daunting, so we need an alternative: a desktop environment interface method of getting hold of packages.

This is where Synaptic comes in. Synaptic is a friendly-looking graphical interface to the apt-get terminal command which allows you to manage your application installations, and removals, through the already familiar desktop environment. Think of it as a kind of online shop where you can pick and choose the programs you want and have them downloaded and installed onto your Raspberry Pi without you having to drop into the terminal.



Upgrade entire systems
Synaptic has the ability to update and upgrade every program or package, and it can upgrade your entire system to the latest version

Install and more
Synaptic is a very powerful tool. With it you can install, remove, upgrade and downgrade single or multiple packages and programs

Browse all documentation
From within Synaptic, you are able to browse and read all available online documentation related to a package or program

Easily find programs
Synaptic enables you to easily locate packages and programs by name, description, version and even by who developed the program

Update the system

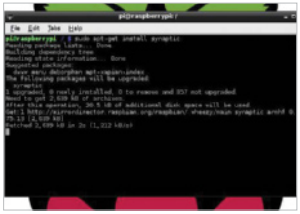
01 Unfortunately, if you have an aversion to dropping into the command-line terminal, then you're going to be stuck at the first step. Before we install anything, we need to make sure that the Raspberry Pi is fully updated and any existing packages are upgraded. Simply enter the following into the LXTerminal:

```
sudo apt-get update
sudo apt-get upgrade
```

Installing Synaptic

02 To install Synaptic, you'll first need to enter the LXTerminal and run the command below – don't forget to type Y to any prompts asking you to accept the installation:

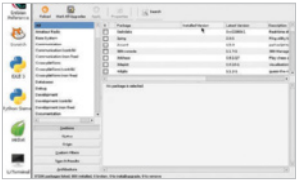
```
sudo apt-get install synaptic
```



Running Synaptic – Part 1

03 In essence, that's all you need to do. Synaptic is now installed and ready to use. However, due to its complexity, there may be some bugs that need ironing out first, so it's best to follow these steps. To test if Synaptic is working okay, first enter the following command into the terminal:

```
gksudo synaptic
```



Running Synaptic – Part 2

04 You should be now looking at the Synaptic program window, where you can scroll through the list of available programs and click on each to download and install. Now we need to test whether it will run from the LXDE menu. Click on the icon in the bottom left, then go to Preferences>Synaptic Package Manager.

Running Synaptic – Part 3

05 Running Synaptic from the LXDE menu results in an error. Don't panic, however: all it's doing is asking for a password. Enter the following password into the box:

```
raspberry
```

This is the default Pi password so we're assuming you haven't changed it!

Fixing Synaptic – Option 1

06 This will temporarily fix the issue, but to permanently resolve it, do one of the following. First, right-click the Synaptic icon in the menu and left-click Properties. In the Command text box, change the text to add the gksudo command. So instead of 'synaptic-pkexec', it will read:

```
gksudo synaptic-pkexec
```

Fixing Synaptic – Option 2

07 The second, and best, option is to drop back into the terminal and alter the way in which the program is executed from the menu. All that's needed is to change one line to another, so that the gksudo command is again used instead of the plain synaptic-pkexec. From the terminal, type:

```
sudo nano /usr/share/applications/
synaptic.desktop
```

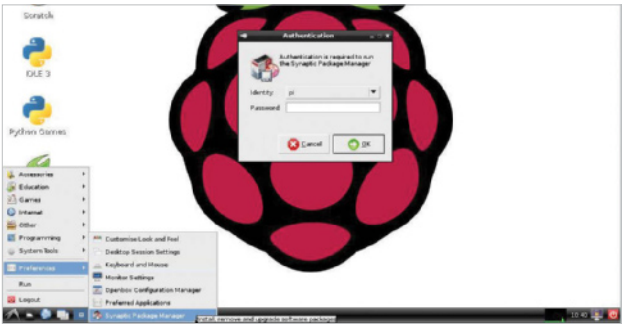
Change 'Exec=synaptic-pkexec' to…

```
Exec=gksudo synaptic-pkexec
```

Synaptic fully working

08 After you've entered those changes, exit nano via Ctrl+X, followed by Y to accept the changes, and then press Enter a couple of times to get back to the command prompt. You can now launch Synaptic from the menu, or by entering the following command when you're in the terminal:

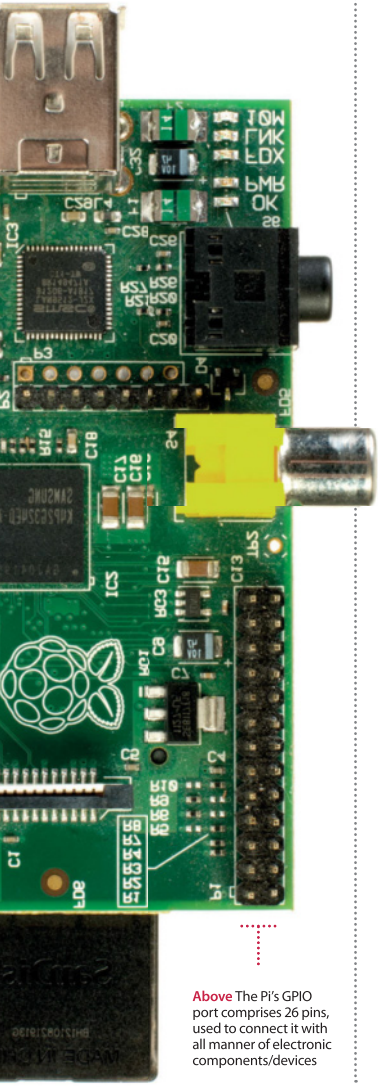
```
gksudo synaptic
```



What you'll need…

Prototyping equipment

RPi.GPIO
https://pypi.python.org/pypi/RPi.GPIO



Above The Pi's GPIO port comprises 26 pins, used to connect it with all manner of electronic components/devices

GPIO port explained

Learn how to harness the power of the GPIO port on your Raspberry Pi. It's easier than you think…

The general-purpose input/output (GPIO) pins on your Raspberry Pi are often central to the success of the projects you'll find in this book. Without them you have no way of interfacing with the real world, be it to trigger lights, buttons or buzzers or read sensors.

GPIO pins aren't special to the Pi; they're actually a standard designed to help control input and output behaviour with all kinds of integrated circuits. Usually you'll find that any one GPIO pin has no particular use pre-defined and they tend to be turned off by default.

Raspberry Pi GPIO

The GPIO pins on the Raspberry Pi can be controlled and triggered in many ways. You can use them from the terminal directly and through Bash scripts, or you can control them using specially designed modules for popular programming languages. Since Python is the official language of the Raspberry Pi, you'll find the GPIO module for Python gives you among the best control for inputs and outputs available. The library is called RPi.GPIO and is installed by default on all Raspberry Pi, but can otherwise be installed in exactly the same way you'd install any useful Python library. The project is hosted on SourceForge and can be found at **sourceforge.net/projects/raspberry-gpio-python**. You'll also find useful links, information and examples of how to use and control the GPIO pins from within simple Python scripts. It helps to have a basic understanding of Python if you plan to use RPi.GPIO, so we'd recommend a basic introductory course like the one found at **www.codecademy.com** or by reading the official Python documentation at **www.python.org/doc**.

There are 26 GPIO pins on the Raspberry Pi and you can use the vast majority of them in any way you want. There are a few pins that

have special purposes, though, so we recommend you familiarise yourself with their layout. For example, the very top row of pins are designed to offer power to external devices like buttons and lights. Since an earth line (often called 'ground') is needed to safely create a circuit, you'll also find several ground pins located in the GPIO port.

How to use GPIO pins

To exploit the power of the GPIO port you'll need a few essential components, the most important of which are jumper leads. Since the pins on the port are 'male', you'll need to purchase either 'female to male' or 'female to female' cables, depending on what hardware you intend to connect to your Pi. Assuming the device you're connecting to also has mail connectors, 'female to female' jumper leads will do nicely, but often you'll be using a breadboard to prototype your circuits, in which case 'female to male' connectors are preferred. Cables and breadboards can be bought very cheaply from just about any online store that sells Raspberry Pi accessories and can usually be found in the 'prototyping' section of the store.

Naming conventions

Once you're ready to connect your device, the next task is to find the right pin for the job. While it's true that all GPIO ports are multipurpose, some are more multipurpose than others! As we've already discovered, some pins are reserved for 5V, 3.3V and ground. Others also have special capabilities, but what's worse is that they can also be called different things. For example, GPIO 18 is also known as pin 12 and PCM_CLK. This particular pin (around halfway down the right side of the GPIO port) is capable of hardware pulse-width modulation (PWM), and is useful for controlling LED lights and motors among other things.

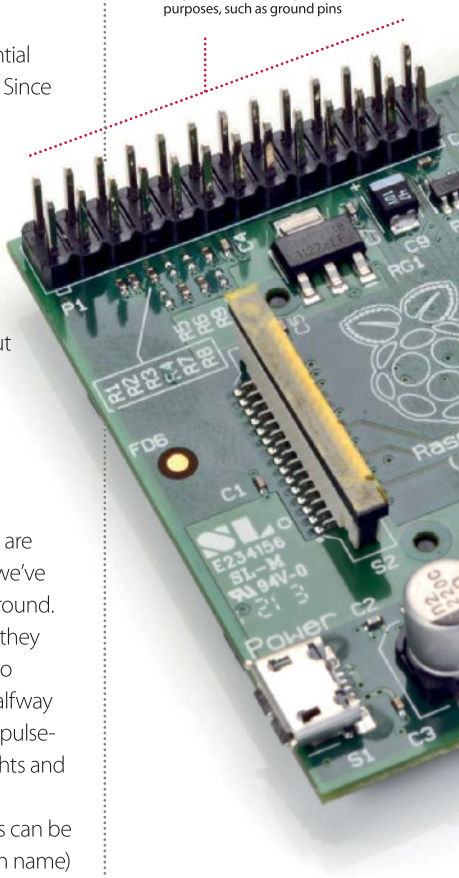
The pin-naming convention you use in your Python scripts can be set manually. This can either be set as BCM (the Broadcom pin name) or the physical pin locations (BOARD).

You'll see in any of the projects where we're using the GPIO port, the following line with either BCM or BOARD in the brackets:

```
GPIO.setmode(GPIO.BCM)
```

The easiest way to deal with the GPIO pin-naming issues is to pick a convention and stick with it!

Below Become familiar with the layout of the GPIO pins and what they do – some have special purposes, such as ground pins

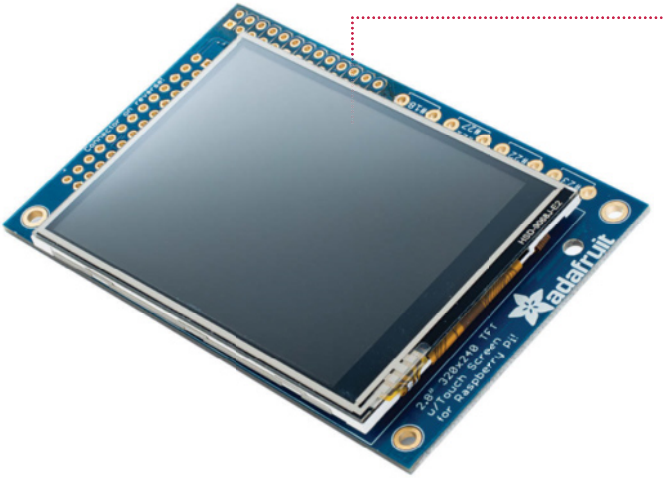


Did you know…

The RPi.GPIO library has some really useful documentation. All you need to do is click on the 'wiki' on Sourceforge.

Top four add-on boards

Get more out of your Raspberry Pi by using these add-on boards to extend its functionality



PiTFT Mini Kit £30

This is a 2.8-inch capacitive TFT LCD touchscreen that's been specifically designed with the Raspberry Pi in mind by the project gurus over at Adafruit. It's capable of slotting directly on top of the Raspberry Pi and is about as big as the Pi is itself.

There are numerous reasons why you'd want to add such a screen to a Raspberry Pi but they all generally come down to the fact that the Pi is very small and very portable and most monitors are not. While you could VPN in via a phone if you're on the go, the screen is connected directly to the Pi and doesn't involve awkward wireless networking. Also as a touchscreen you don't need to bring along other input devices, as it's powered off the Raspberry Pi as well.

This opens it up to a world of possibilities. Portable computer, touchscreen control pad, video camera… anything that could benefit from your Raspberry Pi having a screen and human input while away from your main monitor.

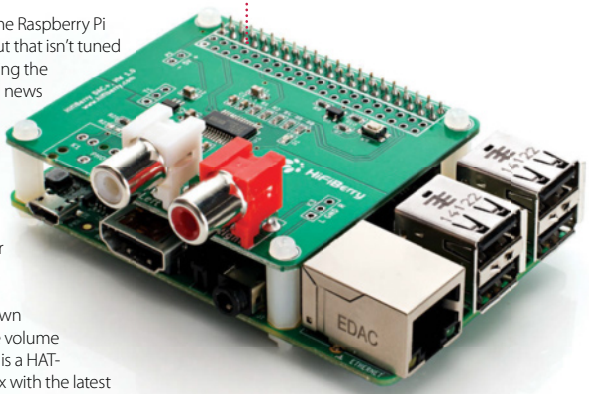
Pi Supply Switch £15

The Raspberry Pi has been so popular, in part, because of the extremely good value for money of the hardware. It packs a lot of punch for the price point and, because it is designed by a charity, they don't need to inflate the price with high profit margins as much as would be done with a more commercial product. Unfortunately, as with anything low-cost, some compromises had to be made in order to bring it in at such an affordable and small form factor.

When comparing it to your more standard desktop or laptop computer, one thing that it is obviously lacking is a power switch and power management functionality. It is surprising how something as simple as a power switch can be so very useful, and it is not until you do not have one that you realise this!

The Pi Supply Switch is a self-solder kit which provides an on, off and soft-off (file-safe shutdown) button to give you basic power management functionality for your Pi. With some provided sample scripts you can make sure your Pi is correctly shut down when you switch off – without the need to open any menus or issue any commands in the terminal – and the circuitry in the switch ensures that power is only removed after the Pi has been shut down. As well as making it more convenient for you, it also reduces the possibility of corruption to your SD card from prematurely pulling the power cable.

HiFiBerry DAC+ £30



As a high quality audio device, however, you may find the Raspberry Pi lacking due to the fact it only has a 3.5 mm stereo output that isn't tuned for high fidelity. You have probably dreamed of enhancing the audio and taking your setup to the next level. The good news is that the clever folk at the Raspberry Pi Foundation, from the second revision of the original Model B, have provided access to the I2S pins; initially on the separate P5 header, and now on the A+, B+ and 2B models it is available from the main 40-pin GPIO header.

I2S is a communications protocol designed specifically for audio devices and has enabled a number of companies like HiFiBerry and IQaudIO to create high quality audio add-ons. The HiFiBerry DAC+ is an add-on which brings a high resolution (192 kHz, 24-bit) Burr-Brown digital-to-analogue converter to your Pi. It has hardware volume control using Alsamixer, among other features, and as it is a HAT-compatible board. It works plug-and-play out of the box with the latest Raspberry Pi firmwares, and it works with all the popular operating systems for both standard use and media playback, such as Raspbian, Arch Linux, OSMC, OpenELEC, Volumio, Pi MusicBox and many more. If you are serious about your audio quality and want a high quality, low cost, Internet-connected solution, then you no longer have any excuse – you can build your own for under £100.

"The magic of the RasWIK is that it comes with a wireless radio chip that fits snugly onto the GPIO ports out of the box"

Energenie Pi-mote £50

Home automation is all the rage at the moment; automating tasks like fiddling with heating controls and turning off the lights before you go to bed can make our lives much easier.

One thing that we are always told is to turn off devices at the plug rather than leaving them on standby, as they use a lot of electricity when not properly turned off. This is where the Energenie Pi-mote control starter kit comes in. It contains two remote-controlled plug sockets which can be turned on and off with an RF remote.

What does this have to do with the Raspberry Pi? Well you also get an add-on board to enable you to control the sockets via software on the Raspberry Pi, which unleashes whole new possibilities – you could set your lamps to turn on and off automatically at specified times when you are away to avoid burglars, or create a basic web app to control your plug sockets remotely using your smartphone alone.

They only come in UK and EU plug types, so if you use a different plug then you may need to look for something else (and maybe send Energenie a request to make more versions).

What you'll need…

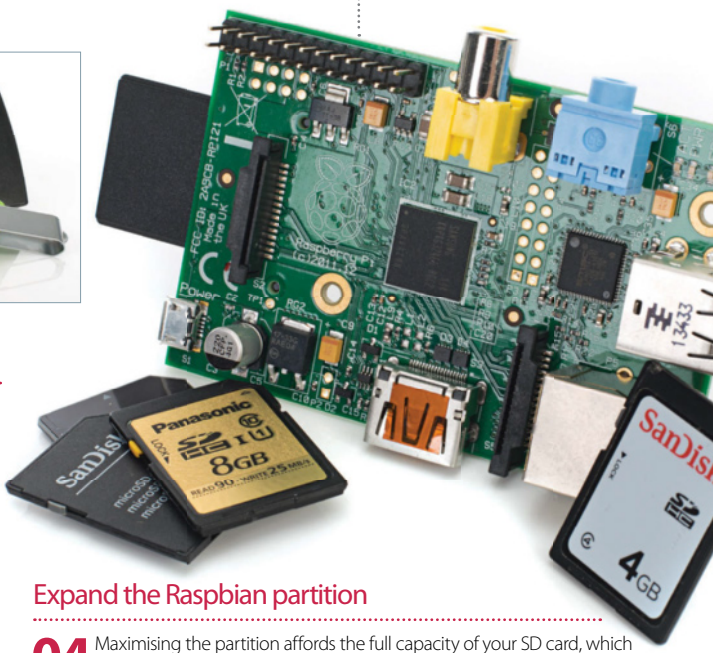
Raspbian
www.raspberrypi.org/downloads

SD card

Supercharge your Pi

Get the most out of your Raspberry Pi with these performance-enhancing tips and tricks

Your Raspberry Pi is plugged in. Raspbian is installed on the SD card and you are right in the middle of setting up a wireless print server or building a robot to collect your mail from your doormat. But are you truly getting the most from your little computer? Perhaps you haven't explored the full set of options in Raspbian, or you're running the entire OS from SD card, something that can reduce SD card lifespan. Various tools and techniques can be employed to improve performance, from choosing the right hardware to overclocking the CPU. You might even maximise storage space on the Pi's SD card or all but replace it with a secondary device. Use these tips and tricks to reconfigure your Pi setup and optimise software and hardware to ensure you get the best performance.



Use better storage hardware

01 Your choice of storage media can have an impact on your Raspberry Pi's performance, regardless of the operating system. A low capacity SD card with poor error correction, is going to be slower than a larger card with greater resilience, so you need to find the right balance for your project and shop wisely.

Choosing the best SD card

02 Various standards of SD card are available, with the more expensive designed for better error correction. For the best performance on your Raspberry Pi, choose an SDHC card with a high rating. The same advice applies to MicroSD cards, which you can use on your Raspberry Pi with an SD card adaptor or directly insert into a Raspberry Pi B+.

Make the most of your storage

03 You'll typically need 1-2GB of storage for your chosen Raspberry Pi distro, so any remaining storage on your SD card will be used for updates and data you create or save. In Raspbian you can open a command line and run the configuration utility to gain more space (only if your SD card's greater than 2GB):

```
sudo raspi-config
```

Expand the Raspbian partition

04 Maximising the partition affords the full capacity of your SD card, which will increase the media's lifespan (there is more space to write too, so the same sectors aren't being overwritten as often). With raspi-config running, use the arrow keys to select expand_rootfs in the menu. Then wait briefly while the partition is resized.

Write data to RAM

05 Rather than reading and writing data to your SD card – something that will eventually result in a deterioration of reliability and performance in the card – you can configure Raspbian to write to the system RAM, which will speed things up slightly and improve the SD card's overall performance. This is achieved using fstab (file systems table), a system configuration available in most Linux distros.

Enable fstab in Raspbian

06 This is much like creating a RAM disk in Windows. In the command line, enter:

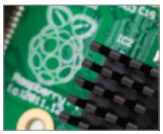
```
sudo nano /etc/fstab
```

Add the following line to mount a virtual file system:

```
tmpfs /var/log tmpfs
defaults,noatime,nosuid,mode=
0755,size=100m 0 0
```

Follow this by saving and exiting nano (Ctrl+X), then safely restarting the Pi:

```
sudo shutdown -r now
```



"You'll typically need 1-2GB of storage for your chosen Raspberry Pi distro"

Picking an external USB drive

Speeding up your Raspberry Pi by migrating the root filesystem to an external USB drive is a start, but what sort of device should you use for the best performance? With a USB thumb drive you can add flash storage up to 16GB without running into any significant problems (the larger the drive, the greater the current is required to read/write). Anything larger is expensive and unnecessary. If you're planning to use an external HDD, there are no power issues as it will have its own power supply. As ever, your choice should suit your project.

Configure fstab for fast performance

07 Upon restarting, the virtual filesystem will be mounted and /var/log on the RAM disk. Other directories that can be moved to RAM include:

```
tmpfs /tmp tmpfs defaults,noatime,nosuid,size=100m 0 0
tmpfs /var/tmp tmpfs defaults,noatime,nosuid,size=30m 0 0
tmpfs /var/log tmpfs defaults,noatime,nosuid,mode=0755,size=100m 0 0
tmpfs /var/run tmpfs defaults,noatime,nosuid,mode=0755,size=2m 0 0
tmpfs /var/spool/mqueue tmpfs defaults,noatime,nosuid,mode=0700,gid=12,size=30m 0 0
```

Add each to /etc/fstab in nano.

Move your OS to a HDD

08 If you're concerned about the lifespan of the SD card, why not reduce your Raspberry Pi's reliance on it? Instead of using the SD card as a sort of budget SSD, change its role and add a HDD or USB stick to run the operating system, leaving the SD card for bootstrapping. This can give a marked performance boost to the SD card.

Back up the SD card

09 Create a copy of your Pi's SD card. Shut down, remove the card and insert it into your desktop computer. In the command line, run:

```
sudo dd bs=4M if=/dev/sdb of=~/backup.img
```

The path /dev/sdb represents the SD card. Copying takes 5-10 minutes. When complete, remove the SD card and connect your USB device.

Copy Raspbian to USB

10 Using a blank Ext4-formatted USB thumb drive (or external HDD) as the destination drive, enter:

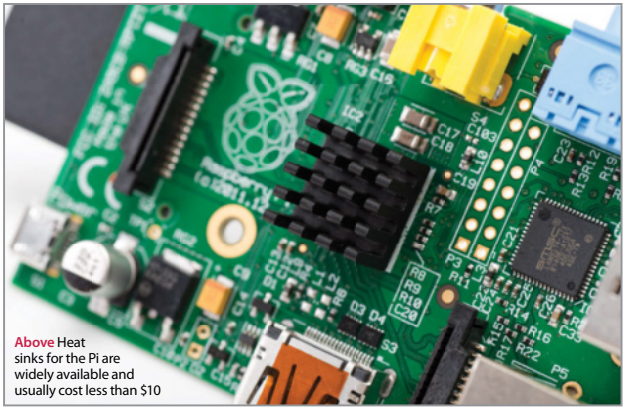
```
sudo dd bs=4M if=~/backup.img of=/dev/sdc
```

Leave the backup on your computer, just in case something goes wrong. With an SD card and USB storage device sharing an identical disk image, it's time to consider what you're going to do next – create a faster Raspberry Pi.

Below Having your filesystem on a USB stick is great for backups as well as performance boosts

Split the Raspbian partitions

11 Ideally, the boot partition should remain on the SD card while the root filesystem is run from the external HDD or USB thumb drive. Using your preferred partition manager (Disk Utility is in most distros), unmount and delete the root filesystem from the SD card, ensuring you have retained the boot partition. After removing the SD card, connect your USB device and delete the boot partition, taking care to leave the root filesystem intact. Then resize the root filesystem on the USB device, making sure that 10MB remains.



Above Heat sinks for the Pi are widely available and usually cost less than $10

Identify the root filesystem

12 You're going to have the SD card and the external USB storage connected, so you need to tell the Pi where the root filesystem is. On the desktop Linux computer with your SD card inserted, run:

```
sudo nano /boot/cmdline.txt
```

Find root=/dev/mmcblk0p2 (or similar) and change that to read root=/dev/sda2 which is your external USB storage. Save and exit.

Add other USB devices

13 You can now restart your Pi with the storage devices attached, but as soon as you connect further USB media you'll suffer problems. Avoid by installing gdisk:

```
sudo apt-get update
sudo apt-get install gdisk
```

Then run gdisk:

```
sudo gdisk /dev/sdb
```

Enter ? to display the options and select Recovery and Transformation options (experts only), followed by Load MBR and Build Fresh GPT. Tap ? one last time and select 'Write Table to Disk' and exit. Remove and replace the USB device and run gdisk again. This time enter I and then 1 to display the Partition Unique GUID.

Make your Pi fast and reliable

14 Make a note of the GUID and then switch to the SD card. Reopen cmdline.txt and change root=/dev/mmcblk0p2 to root=PARTUUID=XXXXXX, where the numerical string from the partition unique GUID should replace the XXXXXX. When you're done, save and exit. You can then start your Raspberry Pi. Congratulations, your Raspberry Pi is now faster and more reliable to use!

Boost performance with overclocking

15 Need more from your Pi? It is possible to overclock the computer, although you should be aware of the risks inherent with this activity. You should also ensure that your Raspberry Pi's processor is suitably cooled – heatsinks for the CPU, Ethernet controller and power regulator can be purchased online.

Overclock your Pi

16 Overclocking is available through raspi-config. Launch from the command line and arrow down to the overclock option. Four further options are available: Modest, Medium, High and Turbo. With your ideal clock speed selected, exit raspi-config and restart your Raspberry Pi to apply:

```
sudo shutdown -r now
```

Now you will need to perform tests to see how stable it is overclocked. Raspberry Pi founder, Eben Upton, suggests running Quake 3 as a good stress test. Should the Pi fail to boot, hold Shift to boot without overclocking, run raspi-config and select a more modest overclock.

Run Raspbian without the GUI

17 Despite these changes, you may find that the GUI remains slow. If you find yourself running a lot of commands in bash, the best thing to do is disable launching into X. In raspi-config, choose boot_behaviour and select the first (default) option to ensure your Pi boots to the command line. Should you need the GUI, enter 'startx' in Terminal.

What you'll need…

Hover
www.hoverlabs.co/#shop

Breadboard

Male to female jumper cables

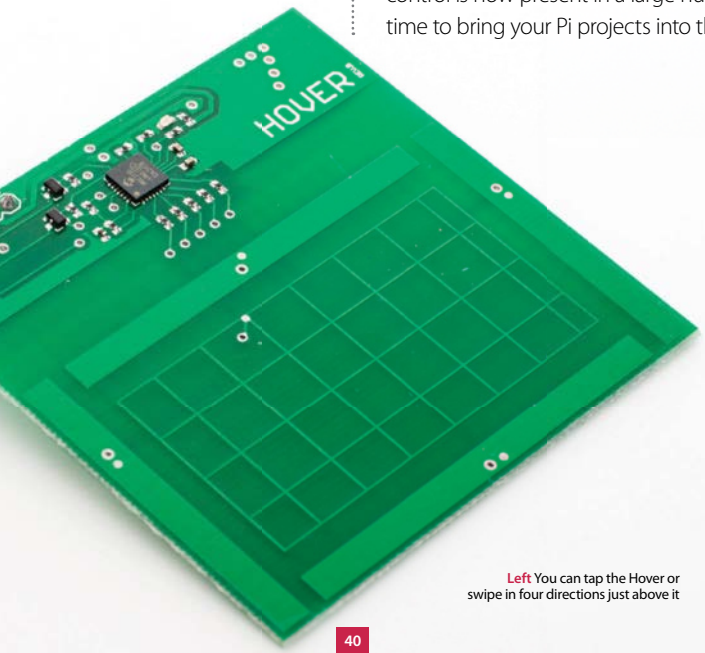
Speaker or headphones

Control with gestures

Hover is an impressive add-on board for your Raspberry Pi that allows you to easily add touch and gesture control to any project

People often ask what the best way is for them to get started with Raspberry Pi. In a more general sense it's often the hardware projects that win out. Pi projects teach a variety of skills and are also varied enough to both keep beginners interested and allow them to work out for themselves exactly what aspect they love best. Even a professional will get a kick out of physical computing!

One of the most important aspects of a hardware project is often the user input mechanism, and as technology is refined we see new and more intuitive ways to accomplish this task. Gesture and touch control is now present in a large number of consumer devices. It is time to bring your Pi projects into the 21st century with Hover!



Left You can tap the Hover or swipe in four directions just above it

Get the gear!

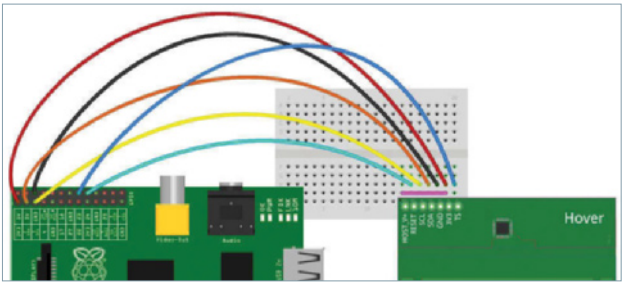
01 The Hover add on board is available to purchase direct from Hover (**http://www.hoverlabs.co/#shop**) for $39 (£25), however this will ship from Northern America and therefore if you are based in the UK or Europe it will likely be quicker and cheaper to order from one of the other retailers listed via the above link. The added benefit of ordering from a retailer is that if you need any of the other items you can likely get those at the same time!

Hover will work perfectly with any Raspberry Pi, including both the new plus versions and the older models – just make sure your system is fully up to date with:

```
sudo apt-get update
sudo apt-get upgrade
```

Update GPIO and I2C

02 When making use of GPIO and I2C (or any other interfacing technique on the Raspberry Pi) it is always good practice to update to the very latest software versions possible. Newer versions typically have bug fixes and additional futures which can come in very handy. GPIO and the RPi.GPIO Python library are installed by default on Raspbian, but you may need to enable I2C if you haven't already. This is a fairly standard process and has been covered many times so we won't go into it here. We would, however, highly recommend the brilliant I2C setup tutorial from Adafruit (**https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c**).



Set up the hardware

03 Make sure your Raspberry Pi is powered down and not connected to power before starting this step, to avoid any unnecessary damage to your Raspberry Pi. Pick up your Hover, breadboard and wires and connect the as shown in the Fritzing diagram. The physical pins you should be using on the Raspberry Pi are 1, 3, 5, 6, 16 and 18. Whilst a Model B Pi is shown, this will be the same connection on a Model A, B, A+ or B+ of any revision. Once completely set up like the image, reconnect the power cord and open an LXTerminal session.

Check the connection

04 Hover connects to the Raspberry Pi through the I2C interface located on the main 26 or 40 pin GPIO bank (depending on which version of the Raspberry Pi you are using). There is a very easy way to check if your Raspberry Pi is correctly connected to Hover using the simple command line I2C tools. Issue the following command:

```
sudo i2cdetect -y 1
```

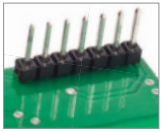
If you see 42 in the response then you are successfully connected to Hover!

Using a Rev 1 Pi?

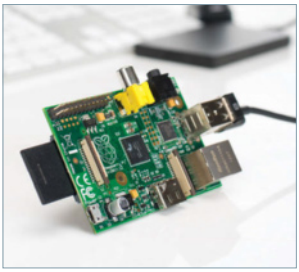
05 In the code, we have passed an option "-y 1" which tells the operating system which I2C bus to look at. The first revision Raspberry Pi) made use of I2C bus 0, whereas all other versions of the Raspberry Pi since have used I2C bus 1. So the above code would change to:

```
sudo i2cdetect -y 0
```

And you should expect the same output (42) as in Step 7. Additionally you will need to edit line 27 of the Hover_library.py file, changing bus = smbus.SMBus(1) to bus = smbus.SMBus(0). A patch that automatically detects the Raspberry Pi version and makes this change for you has been submitted, but not yet accepted into the master branch so this may not be necessary in future versions.



"The physical pins you should be using on the Raspberry Pi are 1, 3, 5, 6, 16 and 18"

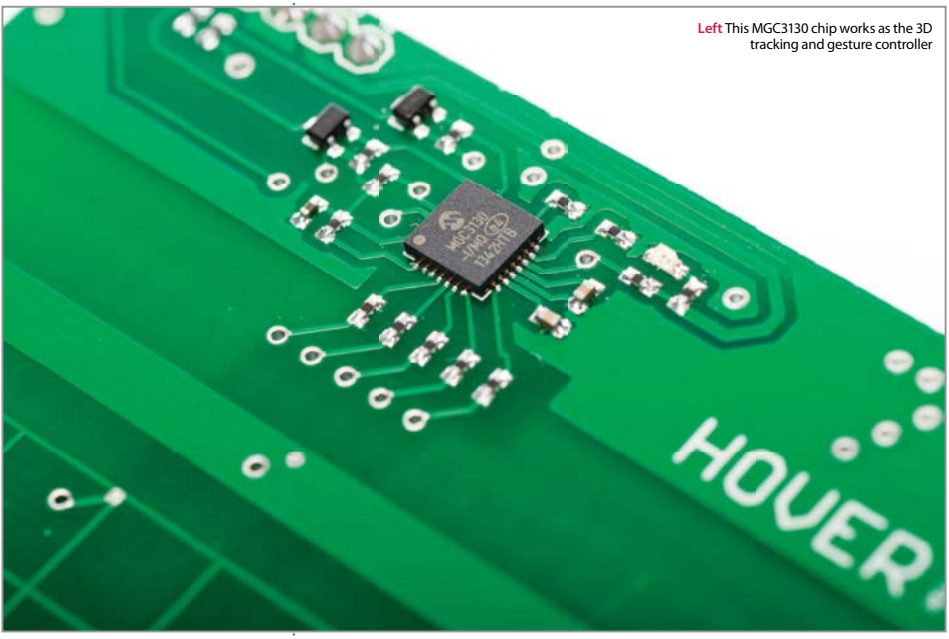


Download the sample code

06 Now you have everything hooked up correctly and your Raspberry Pi is fully up to date, it is time to get the Hover Python library, which makes using the board from Python scripts extremely easy. You can get this using the following command:

```
git clone https://github.com/jonco91/hover_raspberrypi.git
```

This should download a folder called hover_raspberrypi to your /home/pi directory containing all of the files needed for this article. Alternatively you can download the zip file from https://github.com/jonco91/hover_raspberrypi/archive/master.zip.



Left This MGC3130 chip works as the 3D tracking and gesture controller

Plenty of platforms

The Hover board has intelligent on-board level shifting, meaning that it can be used with either 3.3V or 5V logic levels which means it can be used with pretty much any microcontroller your heart desires. There are connection examples and code snippets available for Arduino, Sparkcore and PCduino on the Hover website (**hoverlabs.com**) and these can also be adapted to suit other devices fairly easily. If you decide to create your own example with another device then why not submit a pull request to the Hover GitHub (**github.com/jonco91**) if you are happy to share!

Run the example file

07 The current Hover library is simply a Python file with all of the necessary functions included within it, rather than an installable package (however, this may change in the future). In order to use the functions contained within the Hover_library.py script discussed above, it is therefore necessary to make sure that the Hover_library.py script is located in the same folder as any script you have written that makes use of any of the Hover functions. In a terminal session, navigate to the folder containing the Hover_example.py file and run it using:

```
sudo python Hover_example.py
```

The Hover board will initialise and you will then see the message 'Hover is ready', meaning you are good to go.

Investigate the output

08 Once you have completed Step 7, if you touch the Hover board or make gestures above it you will begin to see output in the terminal which is a bunch of 0s and 1s and then a description of what it has seen – right swipe, north tap, etc. The way the Hover works is that it can sense any one of nine different actions and these are sent to the Raspberry Pi over I2C as an 8-bit binary value. The first three bits describe whether it was a touch or gesture event and the remaining five bits describe the specific type or direction of the event. The exact breakdown can be seen in the code listing to the right.

Enable 3.5mm audio

09 Grab your speakers and plug them in to the 3.5mm jack plug on your Raspberry Pi. You will then need to route audio to the 3.5mm jack using the following command (you can skip this step if you are using an HDMI display, which has in-built audio):

```
sudo amixer cset numid=3 1
```

Make a drum machine

10 In the hover_raspberrypi folder is another folder called examples that contains code and sounds to turn Hover into a drum machine! Navigate to the hover_raspberrypi directory and then copy the Hover_library.py file into the examples folder by using:

```
cp Hover_library.py examples
```

You can then move into the examples folder and run the Hover_drum.py file using:

```
cd examples
sudo python Hover_drum.py
```

Make some gestures and taps on and around Hover and you will have your own basic drum machine!

Create your own responses

11 The great thing about having a Python library available is that it is easy to integrate this device into any of your existing or future projects. The code shown is all you need to get started with Hover. You will see that on line 15 and onwards there are comments saying 'code for … goes here'. Essentially all you need to do is insert the actions you want to occur on the particular event mentioned in the comment and you will be up and running... it really is that easy!

Full code listing

```
import time
from Hover_library import Hover

hover = Hover(address=0x42, ts=23, reset=24)

try:
  while True:

    # Check if hover is ready to send gesture
    # or touch events
    if (hover.getStatus() == 0):
      # Read i2c data and print the type of
      # gesture or touch event
      message = hover.getEvent()
      type(message)
      if (message == "01000010"):
        # code for west touch goes here
      elif (message == "01010000"):
        # code for centre touch goes here
      elif (message == "01001000"):
        # code for east touch goes here
      elif (message == "01000001"):
        # code for south touch goes here
      elif (message == "01000100"):
        # code for north touch goes here
      elif (message == "00100010"):
        # code for swipe right goes here
      elif (message == "00100100"):
        # code for swipe left goes here
      elif (message == "00110000"):
        # code for swipe down goes here
      elif (message == "00101000"):
        # code for swipe up goes here

      # Release the ts pin until Hover is
      # ready to send the next event
      hover.setRelease()
    time.sleep(0.0008)   #sleep for 1ms

except KeyboardInterrupt:
  print "Exiting..."
  hover.end()

except:
  print "Something has gone wrong..."
  hover.end()
```

Other project ideas

12 Most of you are probably now wracking your brains for projects you could use Hover in, but let's face it – pretty much any project that requires physical interaction would be made better with touch and gesture control. If you think it is cool but are lacking inspiration, we recommend looking at the projects section of the Hover website at **http://www.hoverlabs.co/projects**, where there are projects by the creators and community alike.

What you'll need…

A second computer
External storage

Did you know…

SD cards don't last forever so always make sure you've got a backup of your files and your entire OS.

Back up your Pi

Take the initiative and back up your Raspberry Pi to make sure you never lose files again

While the Raspberry Pi is a very solid piece of kit, failure can happen, so it's best to be well prepared and keep your files safe.

The good news is that the Pi's files are all kept externally on the SD card. If your Pi breaks, everything will still be available on the SD card and accessible from elsewhere. The SD card is still susceptible to problems, though. There are a number of ways to back up a Pi. The methods can be broken down into two main categories: saving the important files and creating an exact copy of the state of the SD card. The former involves having copies of files elsewhere, while the latter has you create the same kind of image that you'd normally write to the SD card when installing Raspbian or other Pi-operating systems.

Important files

To save important files, we need to create a copy on an external source, such as external hard drive or another PC. One of the best methods to do this doesn't even involve a Pi; all you need is a PC or laptop with a card reader and you're good to go.

Turn off your Pi, unplug it and remove the SD card from the slot. Find the SD card reader on your PC and slot it in. The main file system of the SD card can be read by Linux PCs by default, and a Windows or Mac computer once you've installed a program that lets it read the ext file system, such as Ext2Fsd. On Windows, the SD card will be listed with the rest of the drives under My Computer (Fig 1). On Linux and Mac, it will be listed wherever storage is shown on the menus and file managers.

Once you've found the SD card on here, open it up and navigate to **home** and then **pi**. This is where the Documents, Downloads, Desktop and other directories can be found. All you need to do is select the files you want to copy and move them to a secure directory on your PC or a connected external hard drive.

If you want to keep the files on another computer, that's fine, but it will be prone to the exact same problems as the Pi in the long run. Keeping them on an external hard drive is a good idea, and putting them on a cloud storage service is better yet, enabling you to access them from anywhere with an internet connection (Fig 2).

Cloning

Creating a clone depends on what operating system you're using on your main computer. For Macs and Linux, you can use a simple command-line tool called **dd** to create an exact copy of the SD card (Fig 3). This is done in the terminal emulator or command line, so bring that up first. Make sure the SD card is plugged in and enter:

```
$ fdisk -l
```

This lists all connected storage devices. The SD card will have 2, 4 or 8GB of space, depending on its size. It'll likely be listed as something like **/dev/sd[x]**, where x is the letter the computer attaches to the SD card. To copy it using dd, enter the following into the terminal:

```
$ dd if=/dev/sd[x] of=backup.img bs=1M
```

You can also add a path to the image you're creating to put it in a specific folder. The process will take some time, and will produce a multi-gigabyte file which you can then write onto the SD by reversing the previous command:

```
$ dd if= backup.img of=/dev/sd[x] bs=1M
```

While this is useful as a backup, you can also use the image to mass-produce SD cards to give to friends or keep in the various places where you use your Raspberry Pi.

Windows

To create a clone on Windows, we can use the Win32 Disk Imager (Fig 4). Download it from here to install it: **bit.ly/L8JdYG**.

Once installed, insert the SD card and launch the software. Choose a name for the backup file and select the SD card from the list of devices. Now press Read and it will create the backup file. Again, this might take a while; however, this time you are at least shown a progress bar.

Storing your cloned image is a little more difficult than your important files – the size of the image being in the gigabytes means it will fill up a lot of cloud-storage services. If you have the space, definitely keep it on there; however, you may need to put it on an external hard drive.

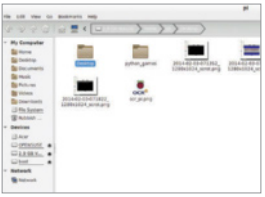


Fig 1: Accessing the SD card from another PC is an easy alternative to transferring files between machines via a USB stick

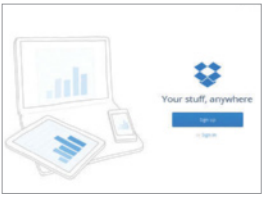


Fig 2: Cloud storage services make backing up files easy and secure, as they're a lot less prone to problems

Fig 3: The dd tool is what you'll use on a Linux and Mac computer, and it's available by default in the terminal emulators

Fig 4: Win32 Disk Imager makes backing up the entire OS easy, and you can even use it to write the image back to the SD card

What you'll need…

Full nano manual
www.nano-editor.org/dist/v1.2/nano.1.html

Beginner's guide to nano

Learn how to edit text on the command line and in a terminal with one of Linux's best tools

Did you know…

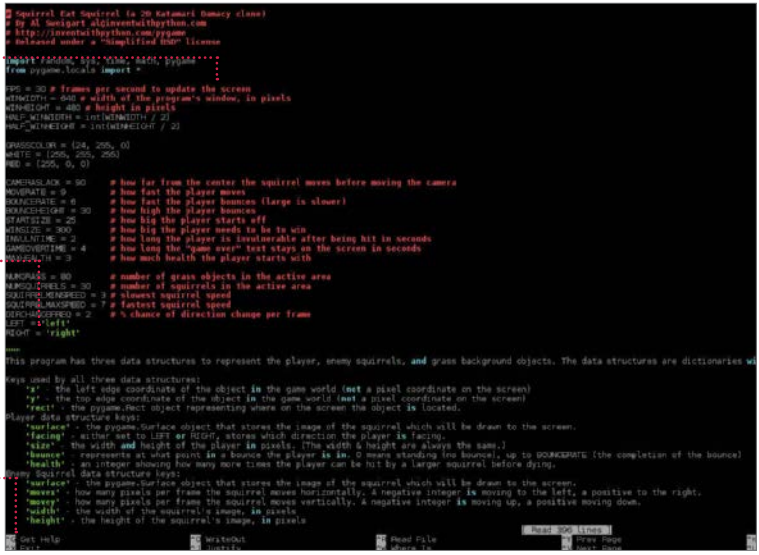
Nano is one of the best command line editors for beginners, but there are lots of options like Kate and Vim.

If you have the itch to do more with your Pi, one of the skills you'll need to learn to pull off many projects is the ability to edit system files. The command line text editor nano is definitely one of the best tools for the job. Text editors are very basic tools; the clue is in the name in this case. There's no formatting or colouring or anything of the sort you would get in a word processor, but that's the point. The kind of files you'll be creating or editing will generally contain code – code which doesn't require to be made bold or bulleted. nano removes all of these distractions, but still has a few of the more handy features you'd find in a graphical text editor. We'll teach you how to make the most out of nano to make your projects run quickly and efficiently.

Writing
Create plain text files in the command line, even write some code for a program

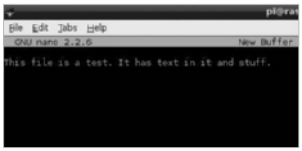
Editing
Edit system files to suit your needs and projects without digging through a file manager

Advanced functions
Search, copy, paste and insert text from another file using some of the built-in nano functions



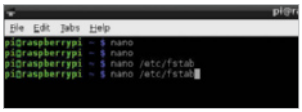
Open nano

01 Open the terminal, or enter the command line, and simply type **nano**. This will open a blank new file. From here you can create a simple text file such as a list, or create a system file, script or piece of code. How the system interprets your file depends on what you write and how you save it.



Save and continue

02 Once you've finished with nano, you can save/write out using Ctrl+O. All shortcuts and functions such as this are done using Ctrl and a letter key. It will ask you what name to save the file under. Whatever you name it, it will be saved in the directory you opened nano from unless you specify a path.



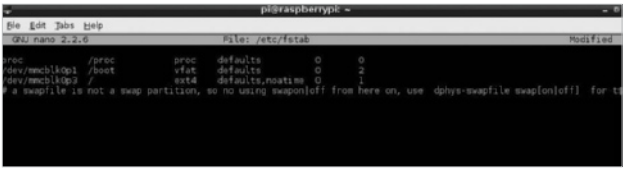
Opening files

03 To edit already existing files, you'll first need to know their location and name. To open them in nano, type the following:

```
$ nano /path/filename
```

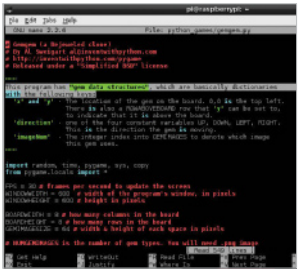
For example, to edit fstab you would type:

```
$ nano /etc/fstab
```



Save and exit

04 Once you've finished modifying the file and need to get on with the next task, you can press Ctrl+X to exit nano. It will ask if you want to save any modifications, which just requires a Y to confirm. If you want to exit without saving changes, you can just use N. To cancel before making a decision, use Ctrl+C.



Copy and paste

05 This is more of a terminal-specific command, but it can be used just as well in nano. While you cannot use your mouse to navigate around the file in nano, you can highlight text in the same way you would in a graphical text editor. Once highlighted, pressing Ctrl+Shift+C will copy any text. Ctrl+Shift+V will paste.

Insert from file

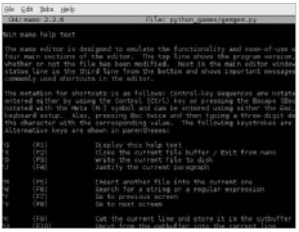
06 If you need to directly insert the contents of another file, there's a quick way to do it without needing to use copy and paste. Typing Ctrl+R and entering the path to the file will insert it into the spot your cursor is at.

Advanced navigation

07 You've probably been using the arrow keys to move one space at a time left, right, up or down. There are other ways to move around the file, though. Using Ctrl+A will act the same way as the Home key does in a graphical editor, moving the cursor to the start of the line. Same with Ctrl+E moving you to the end like the End key. Ctrl+V is page down, and Ctrl+Y is page up.

Searching a file

08 Sometimes you'll be looking for a specific line or phrase in a large text file. Instead of using the arrow keys and tirelessly reading every line, you can use the search function via Ctrl+W. Entering the search term will begin looking through the document, and once you're finished you can press Ctrl+C to exit the search.



Extra help

09 There are many more functions available in nano. For a full list of commands, you can use Ctrl+G, which lists all the shortcuts and what they do. The caret symbol (^) in this list denotes the use of Ctrl on your keyboard.

What you'll need…

TightVNC
www.tightvnc.com/release-2.7.php

Gain remote desktop access

Learn how to get access to your Raspberry Pi desktop without it being in front of you…

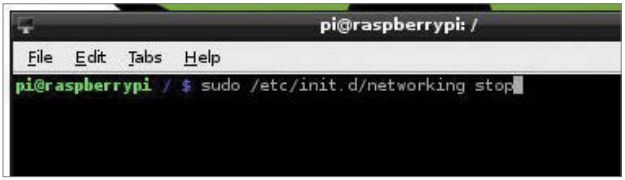
VNC (Virtual Network Computing) is a graphical desktop access and sharing system that allows a user to remotely control and use the desktop of another computer from their own system.

It's handy in many ways: to give you control over a remote system; to help out another user elsewhere; to allow a computer to remain powered on, but without the need for a keyboard, mouse or even a monitor.

In our case, we aim to allow access to the Raspberry Pi desktop without it being hooked up to the aforementioned peripherals. This way, you can do everything you would normally do, but with just the power supply and network attached, thereby freeing up space and saving extra expense. In real-world terms, this means you could potentially access the Pi desktop via an Android tablet or phone!

Did you know…

TightVNC Viewer is a free resource for accessing VNC servers, like the one we'll install in this tutorial.



Static IP address

01 The first step for us is to make sure that the Raspberry Pi has a static IP address. Basically, an IP address is a group of numbers that your network assigns to devices in order to tell them apart. To set up a static IP address, simply double-click LXTerminal and type the following and then press Enter:

```
sudo nano /etc/network/interfaces
```

Static IP part 2

02 This file controls the IP addressing for the Pi. You need to scroll down to the 'iface eth0' line and remove DHCP and replace it with static. Now, on the line directly below, enter the IP address that you want force your Pi to have, along with the subnet mask and the gateway:

```
address 192.168.1.93
netmask 255.255.255.0
gateway 192.168.1.254
```

Static IP part 3

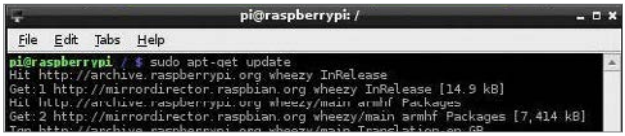
03 After you've entered those details, exit nano by pressing Ctrl+X, followed by Y to accept the changes, and then press Enter a couple of times to get back to the command prompt in the terminal. You can now either reboot your Pi, or type the following into the terminal:

```
sudo /etc/init.d/networking stop
sudo /etc/init.d/networking start
```

Installing VNC part 1

04 Now we'll install VNC and ensure that it starts automatically whenever the Pi is booted up. This used to be a bit annoying under older Raspbian versions, as configuring services often had a nasty habit of breaking the system. But no longer. Enter the following commands, pressing Enter after each one:

```
sudo apt-get update
sudo apt-get install tightvncserver
tightvncserver
```



Installing VNC part 2

05 When the packages have downloaded and installed, follow the instructions on screen (see below) to set up a password and confirm it, but answer 'n' to the view only option. This really is just a security feature and since you are only accessing the Pi at home, it's not absolutely necessary – it's still good practice, though.

```
You will require a password to access your
desktops
Password:
Verify:
Would you like to enter a view-only
password (y/n)? n
New 'X' desktop is raspberrypi:1
```

Configuring VNC server

06 That's the VNC server installed, up and running. Now we need to make sure it loads up as a service every time the Raspberry Pi reboots, so you can access it even if the Pi undergoes a power cycle. To configure the Pi to do this, type the following in the terminal and press Enter.

```
sudo nano /etc/init.d/tightvncserver
```

Configuring boot service

07 We're back in nano, and we'll need to enter some lines of commands in order to allow the Raspberry Pi to activate the VNC server when it boots. In the editor, type:

```
#!/bin/sh
# /etc/init.d/tightvncserver
# Set the VNCUSER variable to the name of  the user to start
tightvncserver under
VNCUSER='pi'
case "$1" in
start)
su $VNCUSER -c '/usr/bin/tightvncserver :1'
echo "Starting TightVNC server for $VNCUSER"
;;
stop)
pkill Xtightvnc
echo "Tightvncserver stopped"
;;
*)
echo "Usage: /etc/init.d/tightvncserver {start|stop}"
exit 1
;;

esac
exit 0
```

Reboot and ready to go

08 Now press Ctrl+X, then Y to save, followed by Enter a couple of times to get you back into the Terminal. What we need to do now is edit the permissions of the script we've just created so that it's executable and active. Do this by typing the following commands into the terminal, ensuring that you press Enter after each one otherwise it will not be registered:

```
sudo chmod 755 /etc/init.d/tightvncserver
update-rc.d tightvncserver defaults
sudo reboot
```

Once you have completed these steps, the final thing that you will need to do is to unplug the Raspberry Pi and locate it somewhere that has easy access to a network cable. If you install the likes of TightVNC Viewer, or any other remote access software (as long as it uses the Tight protocol) then you should be able to point the client to the IP address 192.168.1.93:1 (or whatever the static IP address is of the network that you wish to connect the device to) and have full access to the Raspberry Pi.

What you'll need…

Any Raspberry Pi distro
www.raspberrypi.org/downloads

Internet connection

Second computer

Access your files with SSH

Use the terminal of your home computer to gain quick and secure access to your Raspberry Pi

Did you know…

SSH is the most secure method of accessing any machine remotely. It's also quicker than VNC.

While remotely logging into the full X environment is – as we demonstrated on the previous pages – very useful, it has its disadvantages. Firstly, it's not particularly quick or convenient to do. The user experience can be slow and cumbersome too, but worst of all, it's not as secure as we'd really like.

If you want to take a more convenient and secure approach to accessing your Raspberry Pi from another computer, you'll find that all well-versed Pi enthusiasts will use SSH. SSH stands for Secure Shell and is a cryptographic network protocol which is designed to ensure secure data communication via the command line. While beginners might argue it's easier to remotely access their Raspberry Pi using the full graphical interface, as soon as you've learnt a handful of basic command-line techniques you'll quickly find 'dipping in' to your Raspberry Pi via SSH is by far the most convenient way to talk to it remotely.

You'll be pleased to hear that SSH is already well configured out of the box – there's very little you need to do to make it work, especially if your remote computer runs Linux or OS X, as we're going to demonstrate here.

Fig 1: Use the ifconfig command in the terminal to discover the IP address of your Raspberry Pi

SSH is easy with Linux and OS X

Assuming your Raspberry Pi is on and connected to a network either by Ethernet or a Wi-Fi dongle, the only piece of information you need is the IP address of your Pi. To discover this, all you have to do is open a terminal window on your Raspberry Pi and type the command `ifconfig` (Fig 1).

If you're connected by Ethernet cable you'll see 'eth0', then several lines of results. The IP address will be the number on the second line next to words 'inet addr'. If you're connected by Wi-Fi,

the result will be on the second line of 'wlan0'. The IP address itself is a group of four numbers separated by full stops. If you're on your home network, it will likely be similar to 192.168.0.15. We'll use this number for our guide, but replace it with the IP address you've seen via ifconfig.

With this information in hand, all you need to do is open a terminal window on your remote Linux PC and type:

```
ssh pi@192.168.0.15
```

This assumes your Raspberry Pi's username is still the default (which is **pi**) and you're replacing the IP address with the one you made note of just now. If you've added user accounts or changed the default, you'll need to replace the username before the '@' with whatever you've changed it to. When you press Enter, you'll be prompted to enter the password for your Raspberry Pi – again, if it's the default you can type **raspberry**, otherwise type in your Pi's password and press Return.

You'll now see that the username details on the command line have changed to reflect those of your Raspberry Pi – you're now connected remotely. Try looking through your files or using nano to open a file to edit it!

Never type another IP address

Of course, if you take your Raspberry Pi to another network, your Pi's IP address will be different. If you just want to connect to it remotely, it becomes incredibly tiresome to set up your Pi with a keyboard, mouse and monitor just to get the IP address. Wouldn't it be easier if you could just type the Raspberry Pi's name into the terminal to connect? It sounds too good to be true, but it's actually very easy to do.

All you need is a small piece of software which effectively lets you discover hosts and services on your local network by name instead of IP address (Fig 2). To set it up on your Pi all you need to do is open a terminal and type:

```
sudo apt-get install avahi-daemon
```

Once the installation is complete, all you have to do to access your Raspberry Pi via SSH is type the following into the terminal:

```
ssh pi@raspberrypi.local
```

What's more, you can use the name of your Raspberry Pi when you access it via any other form of networking, be it Samba, remote login with VNC or anything else!

SSH with Windows

As is the way with most things in the Windows world, accessing your Raspberry Pi with Microsoft's operating system isn't quite as straightforward as you'd think. Fortunately, there is a useful tool to help. Putty allows you to make your Secure Shell connection and it's easy to set up and use. You can download Putty from: **www.chiark.greenend.org.uk/~sgtatham/putty**

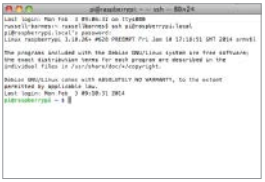


Fig 2: The Avahi tool enables you to log into your Pi via SSH by name, rather than needing to discover its IP address

What you'll need…

NagiosPi
piimagehub.com/project/nagiospi

Win32 Disk Imager
bit.ly/L8JdYG

Disk Utility
bit.ly/1Lec9r5

Internet connection

4 GB (or larger) SD card

Monitor your local network with NagiosPi

Embrace the power of Nagios to keep an eye on servers, switches and applications on your network

Is your PC offline? Has your Linux box stopped serving Minecraft or Counter-Strike? If you're out of the house, or even the country, there is no real way of knowing without trying to log in – something you probably won't be able to do without being on the premises (unless you're using remote desktop software).

A far better way would be to simply receive notifications when your network devices are knocked offline, and this is why we turn to NagiosPi, a Raspberry Pi-built version of the popular open source network monitoring tool.

NagiosPi is available as a full image ready to be written to SD card, with the real configuration taking place once it's up and running. Let's get started.

Download NagiosPi

01 Windows users should write the extracted contents of the NagiosPi_v2.0.zip file to a formatted SD card using Win32 Disk Imager. Linux desktop users can use Disk Utility or the command line (**bit.ly/1z36sp8**). With the image written to SD, safely eject the card and insert it into your Pi before booting.

Log in to NagiosPi

02 As with most Pi projects, you'll probably want to operate via SSH, so check your router's list of connected devices to find the IP address and connect. You can also use a keyboard and monitor connected to your Raspberry Pi. The default username and password for NagiosPi is pi/raspberry.

Expand the filesystem

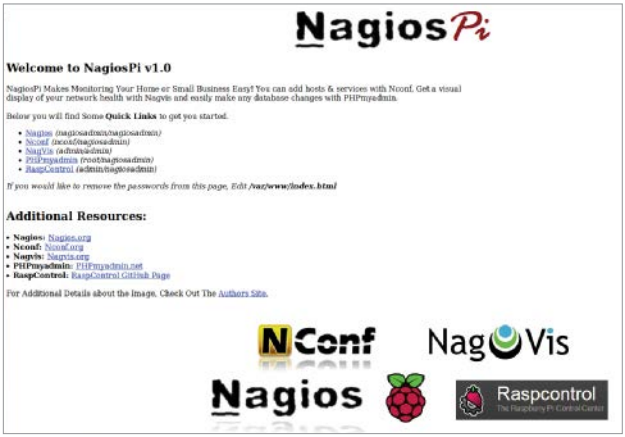
03 Before proceeding, run sudo raspi-config. You'll need to select the first option, Expand Filesystem, and wait a moment as the filesystem is expanded to the full size of the SD card.

Once done, select Change User Password to add some security to your NagiosPi, then select Finish and reboot.

Open in your browser

04 With the Pi rebooted, you'll be able to open the NagiosPi web console in your browser. Visit http://[your.IP.address.here] to see the available options.

Here you'll spot a menu of links in the top-left corner, each accompanied with the username and password to sign in. Start with RaspControl.



Monitor your NagiosPi box

05 In the RaspControl section you'll get a flavour of just what Nagios can do. On the home screen you'll see general hardware information such as connectivity and system status, and as you flick through Details, Services and Disks you'll see what level of monitoring is possible.

View host status

06 Next, go to Nagios and pick Hosts. Here you will see the current status for the configured hosts, which is a combination of items detected on your local network and preset entities. Look for Current Network Status in the upper-left area of the console, just below this you will find alternate views.

Add a host to monitor

07 Open NConf to add the server you wish to monitor, using the 'Hosts – Add' button to input the device hostname, IP address and alias. Click Submit when done, then switch to 'Services – Add', where you can assign a name and check command (such as check_ping) to monitor.

Create configuration file

08 Each check must be set up individually. Some require the installation of NRPE (Nagios Remote Plugin Executor) on remote devices to interrogate and present full system details, but this isn't necessary for basic things like ping.

When you're done, click Submit, then Generate Nagios Config. Following this, select Deploy.

Monitor your server

09 In the NagiosPi window, select Services for a view of currently monitored servers and devices. For each listed device, there will be additional information that you drill down into by clicking Actions. We've only shown you the basics of NagiosPi – investigation will demonstrate just how powerful it really is!

What you'll need…

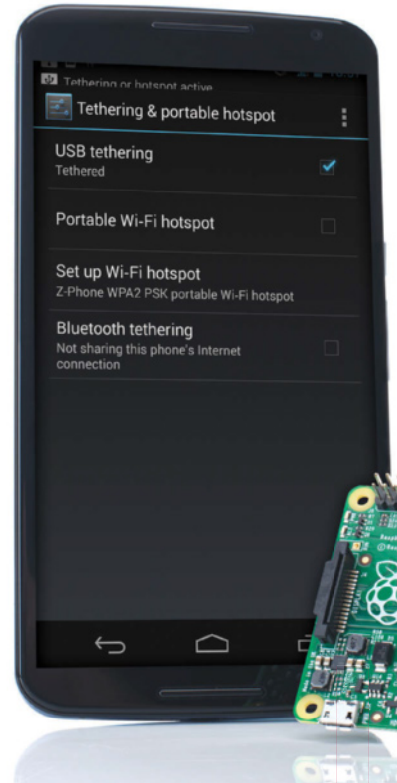
Android device

USB cable

Tether to an Android device

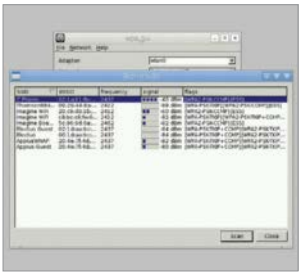
Need the internet on your Pi? Try out a physical tether to your Android device for online access

The portability of the Raspberry Pi is one of its most lauded features. Mini screens, mini wireless keyboard and mouse combos, portable batteries and more can get you out and about, but the internet is a stumbling block that you can't easily fix with an accessory. What you do also usually have with you is an Internet-connected magic pocket box called a smartphone that, with a bit of know-how, you can connect the Pi to and steal some internet from. Over the next two pages we will impart this know-how to get you using your Raspberry Pi on the Internet when you're on the go.



The easy way

01 Many smartphones have a Wi-Fi hotspot feature, which your Pi can easily attach to. First of all, turn the hotspot on and then boot into the Pi. Connect a wireless dongle and open up the wpa_gui in Preferences>Wi-Fi Configuration.

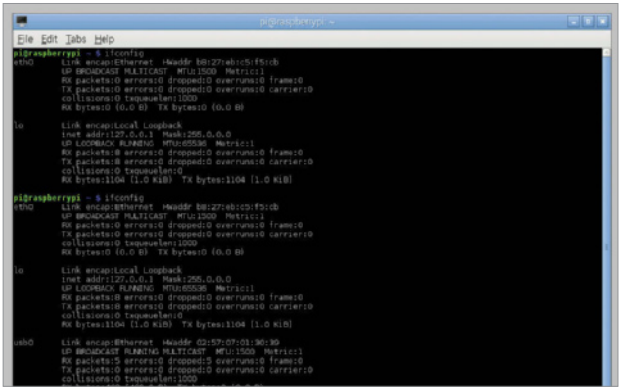


Scan for device

02 Click Scan to open up the scan window and then select Scan again from inside there. It should pick up your device – connect it as you would to any Wi-Fi network and the Pi will remember it for when it needs it next.

Set up tether

03 First connect your phone to your Raspberry Pi via a USB cable – depending on the amount of power your Pi has, it might have trouble charging your phone but it will still let you tether. In the tethering menu you can now activate USB tethering.



Check connection

04 Your Android device will create an interface known as eth0 on the Raspberry Pi. You can check to make sure this is happening, and that it will let you tether, by opening up a terminal and typing the following:

```
$ ifconfig
```

Test connection

06 There's a few ways to test your connection. We'd usually stay in the terminal and ping **www.google.com**, which you can do, or you can click on the browser and see if it loads the page.

Interface settings

08 Here you'll find all the current network settings – yours might look different from ours depending on if you have added any fixed wireless settings or passthroughs. Using the same syntax as the eth0 line, add:

```
iface usb0 inet dhcp
```

> "What you do also usually have with you is an internet-connected magic pocket box called a smartphone that you can connect the Pi"

Quick connect

05 You can connect from the terminal right now to access the Internet. You should be able to do this by typing the following into the terminal:

```
$ sudo dhclient usb0
```

This will automatically grab any available IP address that your phone will give to it.

Save the settings

07 Once you reboot your Raspberry Pi, it won't remember to automatically connect to the phone's tether. However, we can add an entry to its config so that it will try and do this in the future. From the terminal use:

```
$ sudo nano /etc/network/interfaces
```

Tether on the go

09 After a save and reboot, your Pi should now automatically connect to your phone, whether it's via Wi-Fi hotspot or a physical connection. It may draw a little more charge than usual while tethering, so be sure to keep an eye on your battery level.

What you'll need…

Raspbian
www.raspberrypi.org/downloads

Internet connection

Did you know…

You can network with other Linux, Windows and Mac OS X computers with Samba – it's totally cross-platform.

Share your files with Samba

Configure your Pi to share the contents of the home folder over your home network

Regardless of whether you plan to set your Raspberry Pi up as a media centre, gaming machine or full-on development platform, the ability to share and access the contents of your Pi from other machines is very useful.

While the Raspberry Pi is very well configured, by default it doesn't include the ability to share the contents of your home folder with your local network. As you'll see from this guide, though, it's not too difficult to accomplish thanks to Samba and it can help you access your scripts, media or documents from any other machine.

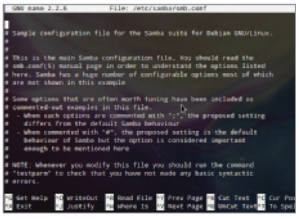
```
russb78@dell:~$ sudo apt-get update && sudo apt-get install
 samba samba-common-bin
```

Update and install Samba

01 Open a terminal window and type the following into the terminal to update the Raspbian repository and then install Samba:

```
sudo apt-get update && sudo apt-get install samba samba-common-bin
```

The operation itself shouldn't take too long. Just make sure that you're connected to the internet before you start.



Load smb.conf

02 With the core packages installed, you now need to tell Samba how you want it to work. You do this by putting your preferred settings in Samba's configuration file. This file is called **smb.conf** and it's located in the **/etc/samba** folder.

While it's easy to locate it using the file manager, it's easiest to open and edit it from the command line. Type the following into the terminal window to open it with nano:

```
sudo nano /etc/samba/smb.conf
```

Initial setup

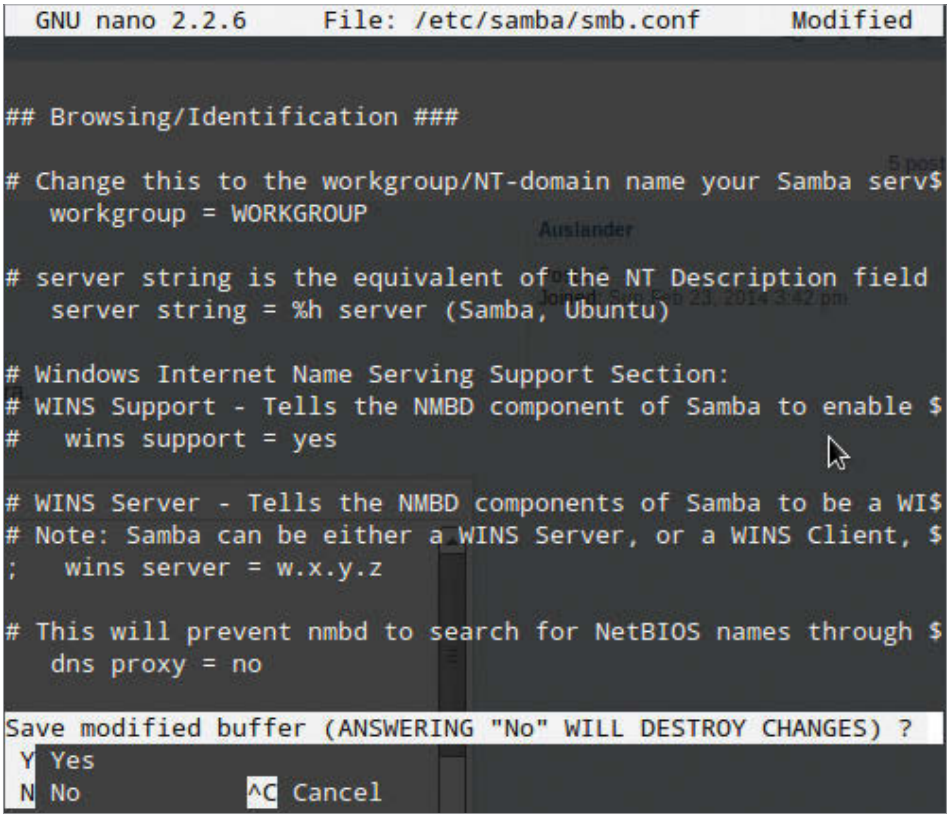
03 Take a look through the file. There's lots of information here and it seems a bit daunting, but there are only a handful of changes and additions to make. Firstly, you need to make sure you have assigned the correct workgroup. Unless your network has been configured differently, it will likely be called WORKGROUP.

To find the workgroup setting, press Ctrl+W and type 'workgroup' then press Enter – this will take you to the first occurrence of the word 'workgroup'. Use the arrow keys to move between lines – the mouse doesn't work in nano.

Windows support

04 If you have Windows computers in your network, you'll need to ensure WINS support is allowed. Assuming you're working from the default **samba.conf** configuration file, you'll find the flag for WINS support a few lines below the workgroup line. Edit it so it looks like this:

```
wins support = yes
```



Add your share

05 Next you need to find the section of the file called Share Definitions. Again, you can use the key combination Ctrl+W to open the search function, then type 'share definitions' and press Enter to be taken directly to it – it's located right towards the end of the document.

Go to the very end of the document and add the following:

```
[pi]
path=/home/pi
only guest = no
browseable = yes
writeable = yes
create mask = 0777
director mask = 0777
public = no
```

Set up passwords

06 You have finished editing the **samba.conf** file now. So, to save and exit, press Ctrl+X, then press Y to save, and Enter to exit.

With this configuration, a layer of password security is in place, so we now need to set up a Samba password to use with it. In your terminal window, type the following command:

```
smbpasswd -a pi
```

You'll be prompted to enter and confirm a new password. This is now your username and password for accessing your files.

Restart Samba

07 The final step is simply to restart Samba for your new settings to take effect.

At the terminal, type the following command to do so:

```
sudo /etc/init.d/samba restart
```

The next time you look at your home network from another machine, you'll see your Raspberry Pi's home folder listed as a share.

What you'll need…

LibreOffice
Installed through tutorial using apt-get

Turn your Pi into an office suite

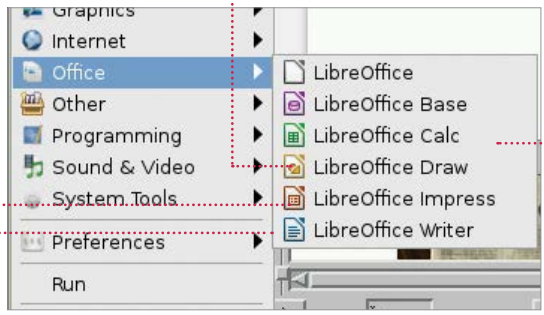
How to add a full, free office suite to your Raspberry Pi to turn it into the smallest home office ever

You can't get far with a computer if you don't have tools that cater for things like letter writing, spreadsheets and presentations. Thankfully this is well within the scope of the Raspberry Pi's capabilities and, although Raspbian comes bundled with a couple of text editors, sometimes you need the extra functionality a full office suite can offer. This is really easy on the Pi and the best option available is open source and thus completely free: LibreOffice.

LibreOffice is a fully featured Microsoft Office drop-in replacement and the Raspberry Pi will handle it well so long as you're not creating huge documents with hundreds of images. You'll need a working internet connection for this tutorial, so make sure your LAN cable or wireless dongle is plugged in and functional, because we'll be using the built-in package manager. Installation does take some time, though.

Presentations
A really impressive piece of kit, it allows creation of presentations in just a few simple clicks

Drawing
Not artistic drawing as you might think, but a great application for creating quick, good-looking flow-and-process diagrams



Word processing
LibreOffice comes with a great word-processing application that works brilliantly on the Pi with no hassle

Spreadsheets
Also bundled is a spreadsheet application – its ability to read any popular spreadsheet file format makes it very powerful

Install LibreOffice

01 You can install open-source software any time (so long as you have an internet connection or installation medium) at your total convenience – without having to worry about handing over payment details. So, let's get right on with doing exactly that. Open your terminal – you can do this by double-clicking LXTerminal from your desktop. Install the LibreOffice office suit by using the package manager:

```
sudo apt-get install libreoffice
```

Now type in your password. Press 'Y' and Enter when prompted to install other dependency packages. Now sit back, relax or make some tea!

Explore the suite

02 When you've installed LibreOffice, you can admire your new applications by hitting the system menu. You'll see a new 'Office' category, underneath which you'll see the following applications:

LibreOffice Base – A database management application.
LibreOffice Calc – Spreadsheets.
LibreOffice Draw – To make flowchart/visualisations.
LibreOffice Impress – For creating presentations.
LibreOffice Writer – The word processor, which is very similar to Word.

Most of these applications are able to open a majority of documents created in Microsoft's Office counterparts and are almost as feature rich.

Add some style

03 We're focusing on some of the word-processing features on the Pi specifically here, so go on ahead and open up LibreOffice Writer. When you're writing documents, it's usually the formatting where people stumble. When you start writing, to the left of the font drop-down menu there's another drop-down that currently reads 'Default'. Select some text, then use this drop-down to change the current applied style. If you don't like the defaults, select 'More' from the drop-down, or press F11. This will allow you to customise the styles and more.

Add an index

04 One of the great things about using formatting is that it allows you to create a contents page for your work easily. So long as you use consistent headings for the different levels of your document, it's a quick and easy way to give a guide to what's in your document. Another advantage is that if you export the file to PDF, you'll end up with each heading in the contents, hotlinking directly to the correct section within your document. To create your index, create some space at the top of your document, and then insert your index by using the menus to go to 'Insert Indexes and Tables>Indexes and Tables'. Review the selected options and click OK when you're happy. It'll insert to the section of the document that your cursor is on.

What about images?

05 There's no better way to brighten up a document than with carefully chosen imagery. This can be achieved with relative ease too.

There are a couple of solutions – the easiest of which is to drag an image in from the file manager directly into the document. Assuming the image is on your desktop, simply drag the file into LibreOffice Writer and you will see the icon change; upon letting go, your image will appear roughly where you dropped it.

Alternatively, you can use the Insert>Picture>From File option.

Export as a PDF

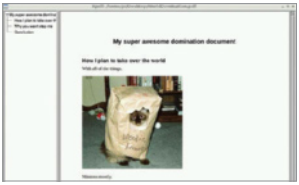
06 The PDF format is a very common way of exchanging documents – mainly because in its simplest form, it's an easy way of stopping accidental changes or modifications to a document. It's also easy to set passwords on PDFs and stop purposeful modifications to them.

The main advantage of PDFs, however, is that you don't really need to worry about the recipient having a specific application to view them with. Most modern operating systems have some sort of PDF viewer built in so you can be sure of a consistent experience.

With LibreOffice, this is easy. You can save your document as a PDF using the File>Export to PDF menu option. Simply choose your desired options, click Export and save it in the appropriate place.

Help yourself by using comments

07 When writing a long document you'll often think of things that you should really check up on or add to the document later. When you think of these things, you can make notes about them using 'Comments' – even more useful when there are multiple people moving through the document. Comments can be added at the current point in the document with Ctrl+Alt+C or by using the menu option Insert> Comment. If you want to get rid of the extra 'Comments' pane, you can use the View menu to toggle it.



Linking out to the world

08 If you've used outside resources in a document that you want to reference, or maybe you don't directly need to include all the information – it might be handy to have a way of getting back to that document later on. You could include an Appendix of information at the end of your document that links to these other resources. You can hyperlink to these easily using the Hyperlink toolbar button or the same from the Insert>Hyperlink menu. Select your text, then click it and you'll be presented with a dialog to link to one of the appropriate places. It's important to note that direct document links use full system paths. If it's being sent to someone else, avoid using these as they'll break for the reader. Ideally use it only a personal reference guide on your own machine.

What you'll need…

Scratch

Internet connection

Did you know…

The website for Scratch, **www.scratch.mit.edu**, contains lots of programs and games other users have made.

Program with Scratch

An interactive guide to coding with the Pi's graphical programming language

Would you like to delve into the world of animation and game creation? Do you want to bring your creative ideas to life without learning a software-development language? With Scratch you can do all this, and much more.

Scratch 1.4 is already installed on the official 'wheezy' Raspbian operating system image. If your Raspberry Pi doesn't already have Scratch installed, don't worry, just hop on over to the official MIT Scratch website to find the download and install instructions.

To begin, all we need to do is open the Scratch Studio. Click on Scratch's cat icon on the desktop, or find Scratch in the LXDE desktop menu.

The Scratch Studio is a complete development environment. It's divided up into a number of separate panels. Each panel has a specific role in the app-construction process and its own specific set of features and tools.

Scratch studio projects

01 Located at the top of the Studio are three quick-access icons and the main menu (Fig 2).

The first globe-style icon sets the language for the Studio. The other two buttons provide rapid access to the project save and share features.

Under the 'File' menu there's a typical set of file-management features to open, save and import Scratch projects. There is also a 'Project Notes' option where we are able to enter feature descriptions and comments.

The 'Edit' menu contains a mixed bag of animation, image and audio-editing tools. While the 'Help' section provides access to the browser-hosted help pages.

Rather interestingly, the 'Share' menu allows us to share our projects with the whole world. Any Scratch project can be posted onto the Scratch community website via the 'Share This Project Online…' option

and the 'Upload To Scratch Server' form (Fig 3).

Let's load the 'Aquarium' example project. Select the 'Open…' option in the 'File' menu to display the Open Project dialog. From the list of large buttons on the left, click on the one called 'Examples'. Next, on the right, select the 'Animation' folder with a double click. Then select the '6 Aquarium' item. The open dialog window contents should look like the one in Fig 4.

With the Aquarium project loaded our Scratch Studio should look similar to Fig 1. We'll begin our Studio tour with the staging area.

Fig 2: Scratch Studio Menu – Scratch Studio's main menu and shortcut icons in action

Fig 3: Project Share Dialog – We can share our projects using the Studio's Share menu

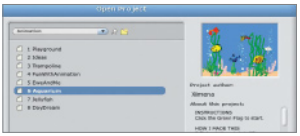


Fig 4: File Open Dialog – Use the Studio's File menu and 'Open…' to load the Aquarium project

Scratch studio stage

02 The stage is where all the action takes place and is located at the upper right of the Scratch Studio.

The stage is constructed from graphical elements called sprites. Here we have plants, bubbles, fish and other creatures. You can also add and create your own assets with scratch, but we'll come to that later.

At the top of the 'Staging Area' there's a green flag and a red circle. Click on the green flag to bring the aquarium to life. Now spend a little time studying the Aquarium animation. Note the creature's movements and rising bubbles. The red circle icon stops the action.

We can set the view mode with the three buttons located just above the green flag.

The two left-hand buttons increase or decrease the size of the Staging Area panel. A smaller Staging Area means the central area of the Studio increases in relative size compared.

The right-hand button is the Presentation Mode which displays the stage in full-screen mode (see Fig 5). Exit presentation mode with the curly arrow button at the top left, or press the 'esc' key.

Scratch studio sprites

03 Beneath the Staging Area is the collection of sprites for this project. The 'Stage' sprite is separated from the rest. It's a little different to the others and acts as the background image for the stage.

The three buttons across the top of this area offer various ways to create a new sprite. The first button opens up a blank canvas in the Paint Editor. The second button creates a new sprite based in an image file, as selected by the popup file section dialog window. The third will select a random image from the pre-installed image collection.

We can manage sprites directly from the stage using the four buttons to the right of the main menu. Here we click on a particular button and then a sprite on the stage.

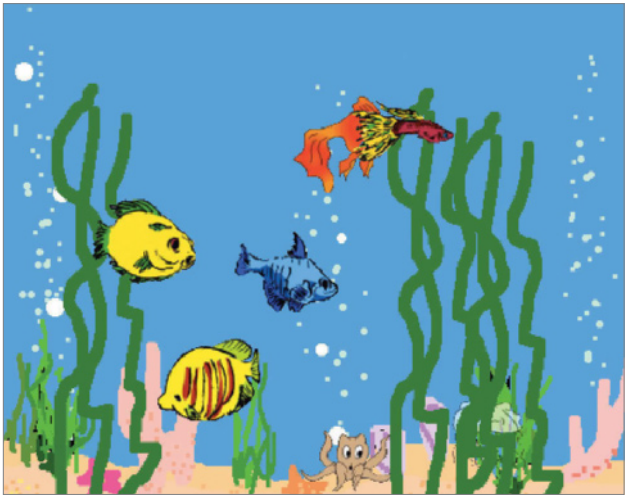


Fig 5: Stage Presentation Mode – The Studio's stage presentation mode is the best way to see the project

Use Scratch blocks and tools

Getting to grips with the Scratch Studio toolbox

Situated in the centre of the Studio is the Edit Panel. The panel contents relate to the currently selected sprite.

Let's start by selecting the jelly fish sprite, called Creature1, from the Sprite Collection area.

At the top we have the sprite's image and name, plus an indication of its current stage coordinates and direction. On the left are three animation control buttons. The top button will rotate the sprite, the second switches between left- and right-facing states, and the third turns animation off.

Below are three Edit Panel tabs. The script tab is where block scripts are created. Here's where we'll drag and drop our blocks, snapping them together in various combinations.

To change a sprite's visual appearance we'll use the 'Costumes' tab. Each sprite can have one or more costumes. For example, the jellyfish has two costumes (Fig 1). Each costume has buttons to edit, copy and delete. New costumes can be painted, imported or captured using the three 'New Costumes' buttons. The sound tab allows us to add audio to our project.

Fig 1: Jellyfish Sprite Costumes – The jellyfish sprite has two different costumes

Scratch Block Styles

01 The 'Blocks Palette' contains the complete collection of scripting blocks. Blocks come in three basic styles, namely hats, stacks and reporters (see Fig 3 on the opposite page).

A hat-style block will start block script execution based on a specific event. The classic hat block is the 'green flag' click event. However, there are numerous other hat blocks, including hat blocks that start script execution after a specific key press, a mouse click and even following sensor event from the some GPIO connected hardware. As you can probably tell, this offers a lot of options to Scratch programmers.

Reporter blocks allow us to specify textual, numeric and boolean values. They fit into specific shaped 'holes' in other blocks. A rectangular reporter will contain a text string. While the rounded end reporters are associated with numeric values, angle-ended reporters contain boolean true and false values.

Stack blocks are the core script building elements. They interconnect with other blocks via their top-edge notches and bottom-edge bumps. Many stack blocks contain 'holes' for reporter style blocks, which will modify their operation depending on the specified reporter block values.

The Scratch block collection is divided into groups. We select a block group using the eight buttons located at the top of the Block Palette panel, namely 'Motion', Control', 'Looks' and so on. These groups are colour coded. Apart from aiding block selection this colour coding provides a visual clue to a block's type when reading a block script in the Edit Panel.

Fig 2: Sound Recorder Tool – Scratch Studio includes a tool to record our own sounds

Scratch Block Help

02 As we've seen, there are many blocks, each with their own specific functionality and capabilities.

In one way this is great news. A large block collection means Scratch can be used in a vast range of software projects, such as games, animation, music, graphics, math, science, robotics, electronics and much more.

However, the wide selection of blocks can be quite a challenge for the novice Scratch coder. To help with this problem the Scratch Studio designers have included an informative set of block-centric help pages.

A simple right click on any block will display a pop-up help page option. The help page contains context-specific descriptions, graphical images and, where appropriate, a script example of how to use this particular block (Fig 4).

It's a terrific feature which greatly simplifies the process of deciding which blocks to use. More importantly, studying these help pages is a highly effective way to enhance our scripting skills and discover the potential contained within Scratch's feature-rich block collection.

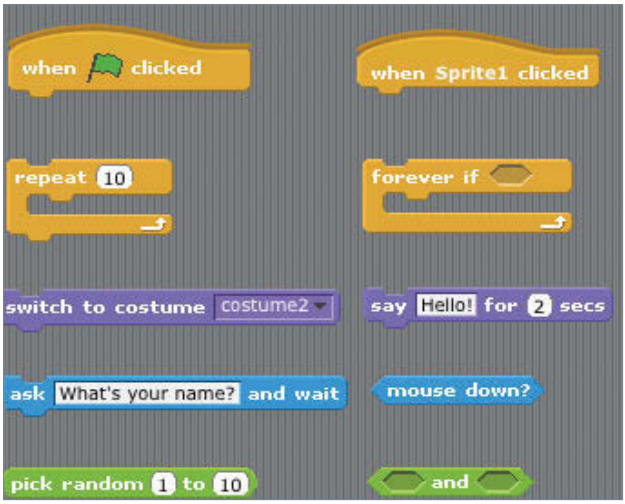


Fig 3: Block Style Examples – the Scratch blocks come in a number of different styles

Did you know…

There is more than one option available to start your scripts beyond the standard Green Flag block.

Block Script Walkthrough

03 Let's dig deeper into how a block script works in practice. For this, we will use a simple Aquarium project block script. From the 'Sprite Collection' panel select the Stage sprite. Then go back to the central 'Edit Panel' and select the 'Scripts' tab.

There's just a single block script. Starting at the top there's a 'green flag' hat-style block to kick off the activity. Next there's a 'forever loop'. The blocks inside this loop are actioned until the stop button is pressed. This forever loop block contains two other blocks.

The first inner script block selects the next background image. Click on the 'Backgrounds' tab to view all the stage images. The second inner block simply pauses execution for a number of seconds. Setting the value to '1' means that this script will pause for a second before then performing the action specified by the next block.

It's important that you remember these two blocks are enclosed in the forever loop block. So, the stage background images will be displayed in sequence for one second each.

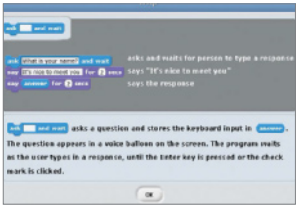


Fig 4: Block Help Window – Example of the help window associated with an 'ask and wait' block

What you'll need…

Scratch project archives
www.scratch.mit.edu/explore/?date=ever

Create a Snake clone in Scratch

Design your own version of Snake to test your new programming skills!

Here, we will create a version of the classic Snake game where you move the snake around the Scratch stage using the arrow keys. You control the head of the snake and must avoid a collision with either the body of the snake or the edge of the stage.

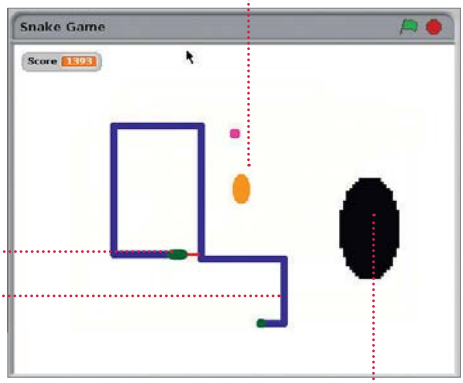
The snake body grows longer each time you eat an egg. You get points added to your score for eating good yellow eggs and lose points for eating bad black eggs. There are also bonus sprites to eat for extra points.

By following this tutorial you will learn to create your own simple sprite graphics, send and receive broadcast events, use a list variable to store data, play sound effects, generate random numbers and use sensing commands to detect when a sprite is touching something.

Did you know…

Snake is a very popular beginner game that started life in the arcades in the Seventies and is still popular today.

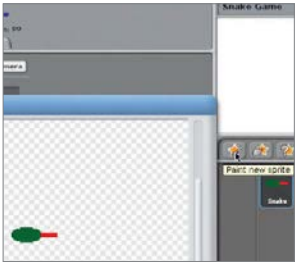
Egg sprite
The Egg sprite appears randomly on the screen and lets other sprites know when it has been eaten (touched by the snake tongue)



Snake sprite
The Snake sprite moves the head around the stage and draws the body behind it. It also detects collisions with the body or edge

Tail sprite
The Tail sprite follows the head, erasing the end of the body so the snake moves, and pausing when it needs to grow

Bad Egg sprite
The Bad Egg sprite also appears randomly but decreases the score when eaten. It also grows in size, getting harder to avoid



Paint the Snake sprite

01 Click the New Sprite: Paintbrush icon to paint the Snake sprite. In the Paint Editor, draw a small green ellipse for the snake head and add a red rectangle for the tongue. It's important that the tongue is a different colour to the head. Name your sprite Snake.

Add a Snake sound

02 When the Snake tongue touches the snake body or the edge of the stage, we are going to play a Game Over sound. We need to add this sound to the Snake sprite. With the Snake sprite selected, click the Sound tab and choose Import. Select the Electronic>Screech sound.

Respond to arrow keys

03 Drag four when key pressed commands from the Control palette, and four point in direction commands from the Motion palette. Configure them as shown so that the up arrow changes the direction to 0 degrees (up) and so on. Click the green flag above the stage to test this.

Make Snake variables

04 Click on the Variables palette. Make two variables, Score and Speed, that are visible to all sprites. Make a list called Next Direction which is visible to all sprites; it will store the sequence of directions that the head takes. Only have the Score variable checked so it appears on the stage.

Initialise Snake variables

05 Use a 'when green flag clicked' Control command and initialise the Snake variables as shown, using commands from the Variables palette. We want to start each game with an empty Next Direction list, so delete all of its entries. The Score must start at zero. The Speed sets difficulty.

Draw Snake body

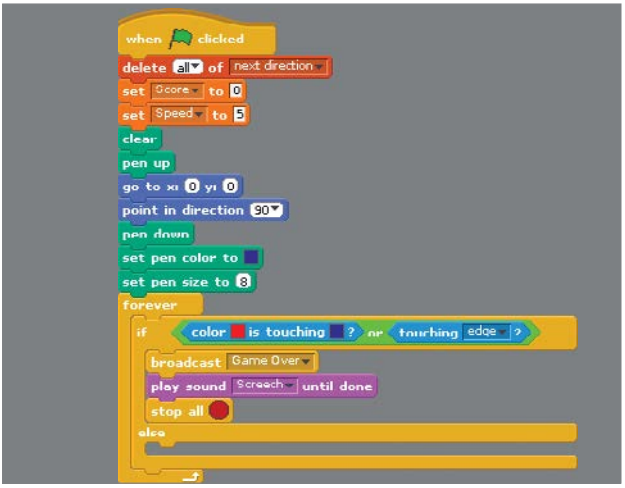
06 Use commands from the Pen palette to control the drawing of the snake body. You should also use commands from the Motion palette to move the snake to the centre of the stage and point left at the beginning of each game. The pen is up until the snake is in the starting position. In the next steps we'll use colours to check to see if the head is touching anything it shouldn't.

Add main action loop

07 Use a 'forever' command with an 'if-else' command nested inside. We have a collision if the red tongue is touching the blue body (the head is always touching the body) or the Snake sprite is touching the edge. Use the Eyedropper tool to select colours within Scratch.

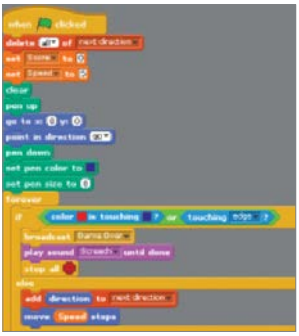
Handle Game Over

08 When a collision has been detected, broadcast a Game Over event (you'll need to create a new event) to the other sprites so they can also react. Also, play the Screech sound effect and stop all scripts so that the snake freezes in its current position at the end of the game.



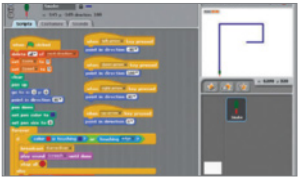
Handle movement

09 Now handle the typical case where there is no collision and the snake must move in its current direction. The pen is down so it will draw the body. The Speed variable determines how many steps to move. Add the current direction to the Next Direction list for the tail to read.



Try out the snake

10 You can now try out your Snake sprite. It will move around the screen in response to pressing the arrow keys. It will draw its body, which will just get longer and longer because we need the tail to erase it. And it will screech and end the game on detecting a collision.



Paint the tail

11 Click the New Sprite: Paintbrush icon to paint the Tail sprite. Draw a small green circle to represent the end of the tail. Name this sprite Tail. The Tail sprite will follow the Snake and erase the end of its tail so that the snake body doesn't grow indefinitely.

Make a Grow variable

12 Make a Grow variable which is for this sprite only – no other sprites need access to it. The Grow variable is used to determine when the snake body needs to grow and the tail therefore needs to pause before following to allow the head to get further ahead.

Handle events

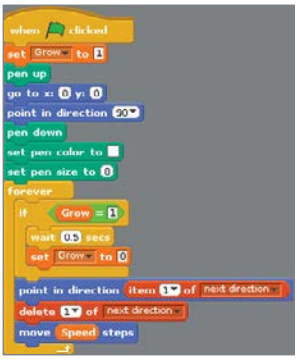
13 The Tail needs to listen for two new events which you create as you need them. When it receives an Egg Eaten event from one of the Egg sprites, it sets Grow=1 so that the tail can pause. And when it receives a Game Over event from the Snake, it must freeze.

Initialise the tail

14 When the green flag is clicked to start the game, Grow is set to 1 so the snake gets a short body. Move to the centre of the stage with the pen up and configure the pen to draw a trail the same colour and size as the stage background so that it erases the body.

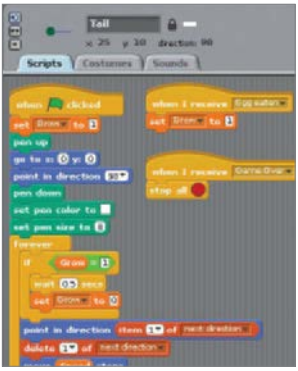
Grow and move

15 Use a 'forever' command to keep the tail moving. If Grow is 1 it should pause and reset Grow to 0 – this makes the body grow longer. Use the first value from Next Direction to set the direction and remove it so you get a new value next time. Move Speed steps.



Try out the tail

16 Now you can try out the Tail sprite. The snake won't keep growing yet because it won't receive any Egg Eaten events. But the tail will follow the snake head around the stage, erasing the snake body as it goes by drawing over it with a white pen (which is the same colour as the background).



Paint the Egg sprite

17 Click the New Sprite: Paintbrush icon to paint a new sprite. In the Paint Editor, draw a small yellow ellipse. Name the sprite Egg. The Egg will appear randomly on the stage and cause the snake to grow and increase its score.

Add Egg sound

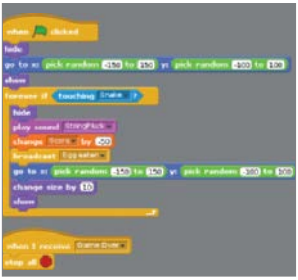
18 Go to the Sounds tab for the Egg sprite and import the Percussion>Cymbal Crash sound. Or you can choose a different sound if you like. This sound will play when the snake eats an egg.

Add Egg scripts

19 Copy the Egg script so that the Egg appears randomly at the start of the game. When the Egg senses that it has been eaten, it must hide, play the Cymbal sound (or whatever you chose in Step 18), update the score, broadcast the Egg Eaten event and then randomly appear again. When the Game Over event is received, it must stop.

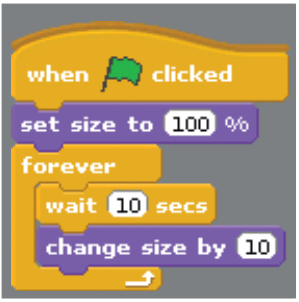
Make Bad Egg

20 Create the black Bad Egg in the same way as the Egg but using a different graphic and the Instruments>StringPluck sound. Drag the Egg's green flag scripts onto the Bad Egg to copy them – just change the sound that's played and reduce the score instead of increasing it.



Grow Bad Egg

21 Add another 'when green flag clicked' script to the Bad Egg so it sets its size to the default 100% when a new game is started and then increases its size by 10 every 10 seconds. The Bad Egg will get bigger and bigger and harder to avoid.



Create Bonus sprite

22 Create the Bonus sprite in a similar way. You can choose the shape and sound for the Bonus. Its scripts are similar to the Egg ones so you could drag one of those to the Bonus sprite and work from that. Make sure you change the sound and increase the score by a random bonus.

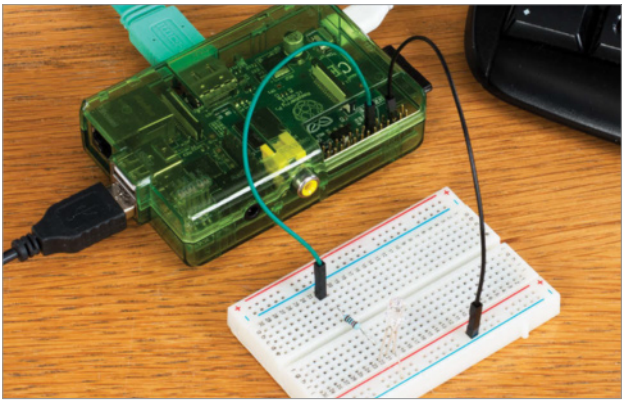
What you'll need…

Breadboard
LEDs
Buttons
Resistors
Jumper wires
ScratchGPIO3

Get interactive with Scratch

Experiment with physical computing by using Scratch to interact with buttons and lights on your Pi

Scratch is a very simple visual programming language, commonly used to teach basic programming concepts to learners of any age. In this project we'll learn how to light up an LED when a button is pressed in Scratch, and then change a character's colour when a physical button is pressed using the General Purpose Input/Output (GPIO) pins on your Raspberry Pi. With these techniques you can make all manner of fun and engaging projects, from musical keyboards to controllers for your Scratch games and animations. You'll need some components for this guide. Just search online for 'Raspberry Pi GPIO basic kit' to get you started.



Below Scratch can be used to do Internet Of Things projects with a few tweaks

"In this project we'll learn how to light up an LED when a button is pressed in Scratch using the GPIO pins"

Install the required software packages

01 Log in to the Raspbian system with the username Pi and the password raspberry. Start the LXDE desktop environment using the command startx. Next, open up LXTerminal and then type the following commands:

```
wget http://liamfraser.co.uk/lud/install_scratchgpio3.sh
chmod +x install_scratchgpio3.sh
sudo bash install_scratchgpio3.sh
```

This will create a special version of Scratch on your desktop called ScratchGPIO3. This is a normal version of Scratch with a Python script that handles communications between Scratch and the GPIO. ScratchGPIO was created by simplesi (**cymplecy.wordpress.com**).

Connect the breadboard to your Pi

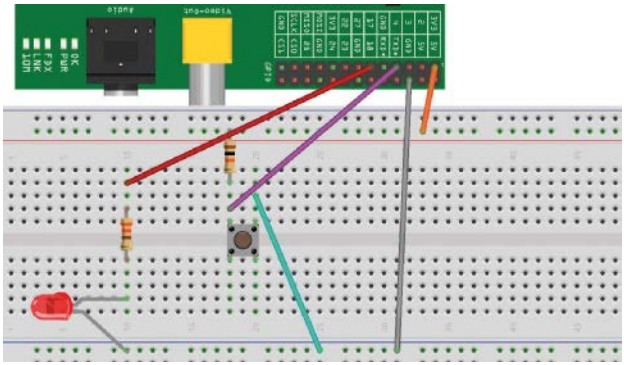
02 Power off your Pi and disconnect the power cable. Get your breadboard, an LED, a 330-ohm resistor and two GPIO cables ready. You'll want to connect the 3.3V pin (top-right pin, closest to the SD card) to one end of the 330-ohm resistor, and then connect the positive terminal of the LED (the longer leg is positive) to the other end. The resistor is used to limit the amount of current that can flow to the LED.

Then put the negative terminal of the LED into the negative rail of the breadboard. Connect one of the GROUND pins (for example, the third pin from the right on the bottom row of pins) to the negative lane. Now connect the power to your Pi. The LED should light up. If it doesn't, then it's likely that you've got it the wrong way round, so disconnect the power, swap the legs around and then try again.

Switch the LED on and off

03 At the moment, the LED is connected to a pin that constantly provides 3.3V. This isn't very useful if we want to be able to turn it on and off, so let's connect it to GPIO 17, which we can turn on and off. GPIO 17 is the sixth pin from the right, on the top row of pins. Power the Pi back on. We can turn the LED on by exporting the GPIO pin, setting it to an output pin and then setting its value to 1. Setting the value to 0 turns the LED back off:

```
echo 17 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo 1 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio17/value
```



Control the LED from Scratch

04 Start the LXDE desktop environment and open ScratchGPIO3. Go to the control section and create a simple script that broadcasts pin11on when Sprite1 is clicked. Then click the sprite. The LED should light up. Then add to the script to wait 1 second and then broadcast pin11off. If you click the sprite again, the LED will come on for a second and then go off. ScratchGPIO3

Wire up our push button

05 Power off the Pi again. This circuit is a little bit more complicated than the LED one we created previously. The first thing we need to do is connect 3.3V (the top-right pin we used to test our LED) to the positive rail of the breadboard. Then we need to connect a 10Kohm resistor to the positive rail, and the other end to an empty track on the breadboard. Then on the same track, add a wire that has one end connected to GPIO 4. This is two pins to the right of GPIO 17. Then, on the same track again, connect one pin of the push button. Finally, connect the other pin of the push button to ground by adding a wire that is connected to the same negative rails that ground is connected to.

When the button is not pressed, GPIO 4 will be receiving 3.3V. However, when the button is pressed, the circuit to ground will be completed and GPIO 4 will be receiving 0V (and have a value of 0), because there is much less resistance on the path to ground.

We can see this in action by watching the pin's value and then pressing the button to make it change:

```
echo 4 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio4/direction
watch -n 0.5 cat /sys/class/gpio/gpio4/value
```

Let there be light!

06 Boot up the Pi and start ScratchGPIO3. Go to the control section and add when green flag clicked, then attach a forever loop, and inside that an if else statement. From the operators section and add an if [] = [] operator to the if statement. Then go to the sensing section and add a value sensor to the left side of the equality statement, and set it to pin7. Enter 0 on the right. Broadcast pin11on if the sensor value is 0, and broadcast pin11off otherwise. Click the green flag. If you push the button, the LED will light up!

What you'll need…

Raspberry Pi
Raspbian:
www.raspberrypi.org/downloads

Pi Camera
Ashton's picam module:
https://github.com/ashtons/picam

Did you know…

The Camera board uses the same kind of sensor and hardware as the cameras in modern mobile phones.

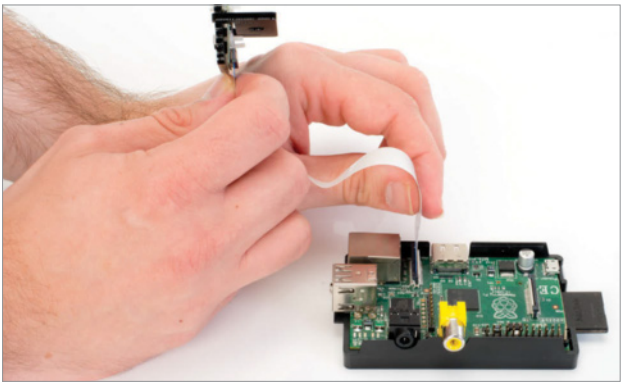
Use the Camera board

Here's how to get your new Pi Camera set up on your Raspberry Pi, and how to use it…

One of the most recent Raspberry Pi accessories is the tiny Pi Camera board – a small PCB with a camera sensor mounted to it that connects via a ribbon to the Raspberry Pi. It's not exactly plug-and-play, so you'll need to do some extra set up.

The Pi Camera has multiple functions, such as for time-lapse photography, using as a webcam, or even as an optical sensor for a Pi-powered robot, like the one we're making here. Because it doesn't take up any USB slots, and draws very low power, it can be a lot more versatile than a standard webcam.

The Pi Camera itself is not a low-quality piece of kit either – with a 5MP sensor, it's also able to create up to 1080p quality video, the same as the Raspberry Pi's HDMI output.



Attach Camera

01 To attach the Camera to the Raspberry Pi, locate the slot between the Ethernet and HDMI port and gently lift up the fastener. Insert the ribbon of the Camera board, making sure to align the ribbon's connectors with those on the Pi. The blue tab on the flex should be facing the Ethernet port.

Pi preparation

02 Before we try to enable the Raspberry Pi Camera, we need to make sure our firmware and software are all up to date with a quick software upgrade. In Raspbian, we do this by opening the terminal and then using:

```
$ sudo apt-get update
```

…followed by:

```
$ sudo apt-get upgrade
```

Pi config

03 Once that's finished, run in the terminal or command line:

```
$ sudo raspi-config
```

to start the standard configuration screen. Navigate down to Enable Camera, press Enter and then simply key over to enable and confirm with another press of Enter. Select Finish and then reboot.

Take pictures

04 To take pictures with the Raspberry Pi Camera, you'll simply need to enter:

```
$ raspistill -o image.png
```

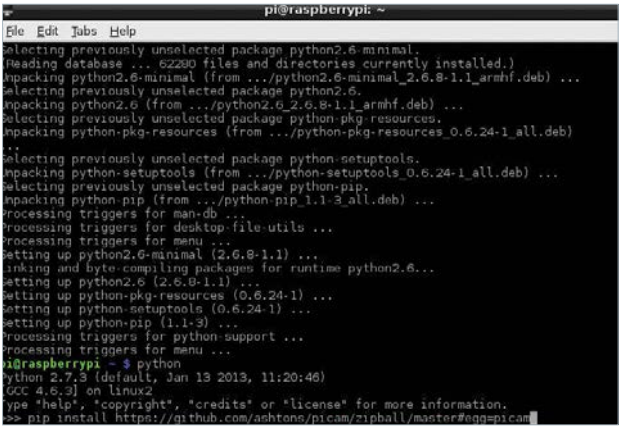
This will show a five-second preview of the input of the camera and then capture the last frame of the video.

Record video

05 To record a video, we use a similar command, raspivid, like so:

```
$ raspivid -o video.h264
```

It will also take five seconds of video by default.



Picam

06 If you want to do a little more with the Pi Camera, there's a simple Python wrapper currently available called picam. You'll need to install it first, though, and we'll use pip for that. Install pip with:

```
$ sudo apt-get install python-pip
```

and then enter:

```
pip install https://github.com/ashtons/picam/zipball/ master#egg=picam
master#egg=picam
```

Picam photos

07 With the module installed we can now use Python to construct a script to take photos with the picam module. It's not as difficult as you might think, so even if you've never used Python before it's worth a try. All you need to do is enter:

```
import picam
i = picam.takePhoto()
i.save('/home/pi/test.jpg')
```

And running it will take a photograph called test.jpg.

Advanced photos

08 You can have it take photos of specific size and quality with a time-based name by editing the code to look like this:

```
import picam
import time
ii = picam.takePhotoWithDetails(640,480, 85)
filename = "/tmp/picam-%s.jpg" % time.strftime("%Y%m%d-%H%M%S")
ii.save(filename)
```

Picam video and more

09 Picam also allows you to take video in a similar way to the above, with the main difference being that you'll use the recordVideo command. You can use the code to take photos or video at regular intervals for time-lapse, or have it trigger during a specified event.

> "With a 5MP sensor, it's able to create up to 1080p video, the same as the Raspberry Pi's HDMI output"

What you'll need…

Breadboard:
www.proto-pic.co.uk/half-size-breadboard

3mm LED light:
www.ultraleds.co.uk/led-product-catalogue/basic-leds-3-5-8-10mm.html

Wires:
www.picomake.com/product/breadboard-wires

270-ohm resistor:
http://goo.gl/ox4FTp5ntj0091

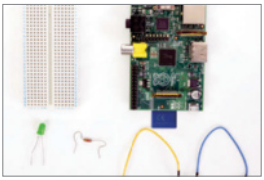


Fig 1: All the items you will need to get going adjusting an LED using PWM. The wires should have a male and female ends

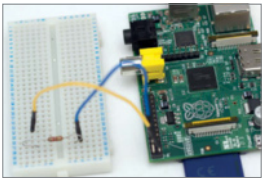


Fig 2: Place the female end onto the Pi, noting pin number 1 being identified by the small 'P1'. The blue wire is ground

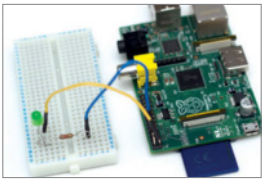


Fig 3: Once everything is connected up, plug in your USB power cable

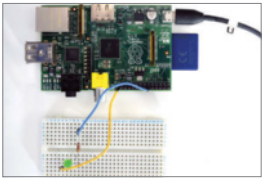


Fig 4: Switch the power on. The LED will light up. If it's dim, use a lower-rated resistor

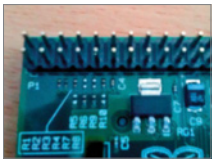
Control an LED using GPIO

An introduction into using an external output, such as an LED, on the Pi

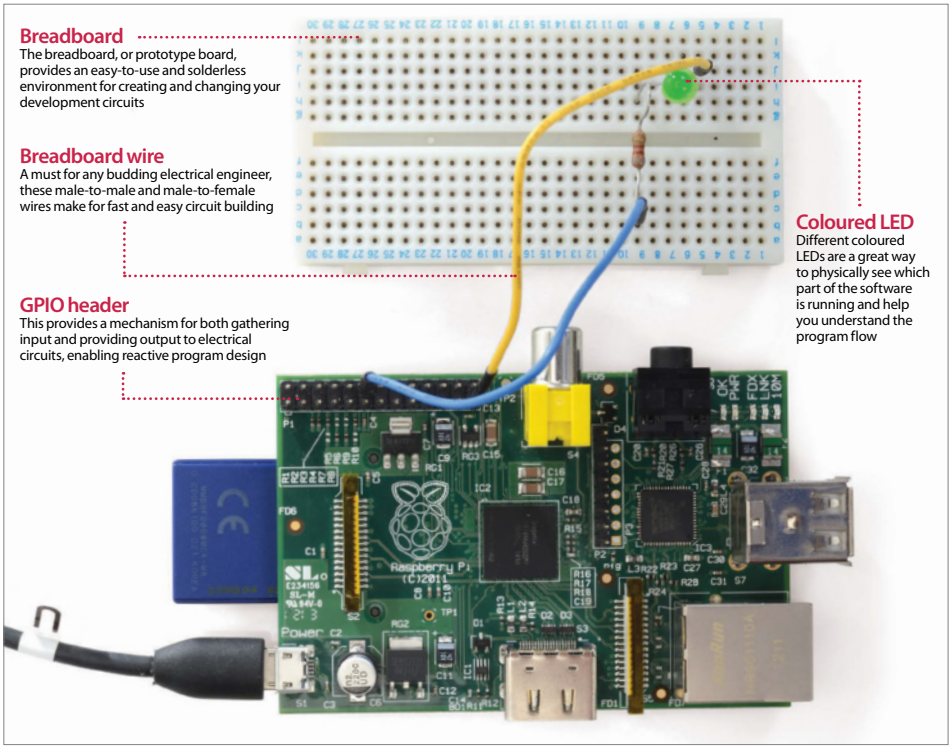
After you have fired up your Pi, maybe installed XBMC and had a play around with streaming, you might be ready for your next challenge. One route to go down is to find interesting uses for one of the many desktop OSs available for the little computer, perhaps using it as a web server, an NAS or retro arcade console. This is all great fun, but an often-overlooked feature of the Pi is its hardware pinouts. If you've never done any electronics before, then the Pi is a great place to start. Or maybe you have used a programmable microcontroller such as Arduino in the past; the Pi, with its increased CPU and RAM over the Arduino, opens up many more possibilities for fun projects.

The Raspberry Pi features a single PWM (pulse width modulation) output pin, along with a series of GPIO (General Purpose Input/Output) pins. These enable electronic hardware such as buzzers, lights and switches to be controlled via the Pi. For people who are used to either just 'using' a computer, or only programming software that only acts on the machine itself, controlling a real physical item such as a light can be a revelation.

This tutorial will assume no prior knowledge of electronics or programming, and will take you through the steps needed to control an LED using the Raspberry Pi, from setting it up to coding a simple application.



"We'll take you through the steps needed to control an LED using the Pi, from setting it up to coding"



The Pi's pins

01 Before we dive into writing code, let's take a look at the layout of the pins on the Pi. If you have your Pi in a case, take it out and place it in front of you with the USB ports on the right. Over the next few steps we'll look at some of the issues you'll encounter when using the GPIO port.

Pi revision 1 or 2?

02 Depending on when you purchased your Pi, you may have a 'revision 1' or 'revision 2' model. The GPIO layout is slightly different for each, although they do have the same functionality. Here we have a revision 1; revision 2s became available towards the end of 2012.

Pin numbers

03 If you take a look at the top left of the board you will see a small white label, 'P1'. This is pin 1 and above it is pin 2. To the right of pin 1 is pin 3, and above 3 is 4. This pattern continues until you get to pin 26 at the end. As you'll see in the next step, some pins have important uses.

Pin uses

04 Pin 1 is 3V3, or 3.3 volts. This is the main pin we will be using in this guide to provide power to our LED. Pins 2 and 4 are 5V. Pin 6 is the other pin we will use here, which is ground. Other ground pins are 9, 14, 20 and 25. You should always ensure your project is properly grounded.

GPIO pins

05 The other pins on the board are GPIO (General Purpose Input/Output). These are used for other tasks that you need to do as your projects become more complex and challenging. Be aware that the USB power supply doesn't offer much scope for powering large items.

Basic LED lighting

06 Okay, so let's get down to business and start making something. Firstly, get your breadboard, two wires, a 270Ω resistor and an LED. Note the slightly bent leg on one side of the LED; this is important for later on. Make sure your Pi is unplugged from the mains supply.

Wiring the board

07 Plug one wire into the number 1 pin, and the other end into the breadboard. Note that it doesn't matter where on the breadboard you plug it in, but make sure there are enough empty slots around it to add the LED and resistor to. Now get another wire ready.

Add another wire

08 Place the female end of the wire into pin number 6 (ground) and the other end into the breadboard, making sure to leave room for the resistor, depending on how large it is. Next, get your resistor ready. You can use a little higher or lower than 270 ohms, but not using a resistor at all will likely blow the LED.

Add the resistor

09 Next we need to add our resistor. Place one end next to the ground wire on the breadboard, and the other one slot below the 3V3 wire connection. This will sit next to the LED when we add it in a second. Note that there is no correct or incorrect way to add a resistor.

Add the LED

10 Grab your LED and place the 'bent' leg end next to the 3V3 wire in the breadboard. Place the other leg next to the resistor leg opposite the ground wire. This now completes the circuit and we are ready to test out our little task.

Power it up

11 Now get your micro-USB socket and either plug the mains end into the wall, or plug it into a computer or laptop port (and powered on!). You should see the LED light up. If not, then check your connections on the breadboard or Pi, or try a different LED.

Set up programming environment

12 Now, we need to be able to do a little bit more than just turn a light on – we want to be able to control it via code. Set up a new Raspbian installation (a guide to this is found on page 28). You don't need a GUI for this – it can all be done via the terminal if you so wish. Before starting, it's best to check everything is up to date with:

```
sudo apt-get dist-upgrade
```

Open up terminal

13 Assuming we want to use the GUI, rather than SSH into the Pi, open up a new terminal window by double-clicking on the LXTerminal icon. We need root access to control the LEDs, so either enter su now, or remember to prefix any commands with sudo.

```
su
```

followed by password or add

```
sudo
```

to the start of each command.

Download GPIO library

14 There is a handy GPIO Python library that makes manipulating the GPIO pins a breeze. From within your terminal window, use wget to download the tarball file, then extract it using tar. This will give us all the files in a new directory.

```
wget https://pypi.python.org/packages
source/R/RPi.GPIO/RPi.GPIO-0.5.2a.tar.gz
tar zxf Rpi.GPIO-0.5.2a.tar.gz
cd Rpi.GPIO-0.5.2a
```

Install the library

15 Now we need to install the library. This is simply a case of using Python's install method; so we need the dev version of Python. Make sure you are in the directory of the library before running this command.

```
sudo apt-get install python-dev
sudo python setup.py install
```

Import the library in a script

16 Create a new Python script. Next import the main GPIO library and we'll put it in a try-except block. Save the file using Ctrl+X and choosing 'yes'.

```
cd /
cd Desktop
sudo nano gpio.py
try:
        import RPi.GPIO as GPIO
except RuntimeError:
        print("Error importing GPIO lib")
```

Did you know…

The official Python docs are a great resource for beginners and professionals alike. **http://python.org/doc**.

Test the script

17 Now to make sure that the script imported okay, we just need to run the python command and then tell it the name of the script that we just created. If all goes well, you shouldn't see any error messages. Don't worry if you do, though. Just go back through the previous steps to check everything is as it should be.

```
sudo python gpio.py
```

Set GPIO mode

18 Reload the script in nano again. We will then set the GPIO mode to BOARD. This method is the safest for a beginner to adopt and will work whichever revision of the Pi you are using. It's best to pick a GPIO convention and stick to it because this will save confusion later on.

```
sudo nano gpio.py
GPIO.setmode(GPIO.BOARD)
```

Set pin mode

19 A pin has to be defined as either an input or an output before it can work. This is simplified in the GPIO library by calling the GPIO.setup method. You then pass in the pin number, and then GPIO.OUT or GPIO.IN. As we want to use an LED, it's an output. You'll be using these conventions frequently, so learn them as best you can so they soak in!

```
GPIO.setup(12, GPIO.OUT)
```

Using PWM

20 The next step is to tell the pin to output and then set a way of escaping our program. Here we call the GPIO class again and then the PWM method, passing in the pin number; the second value is the frequency in hertz – in this case, 0.5.

```
p = GPIO.PWM(12, 0.5)
p.start(1)
input('Press return to stop:')
p.stop()
GPIO.cleanup()
```

Adjust PWM

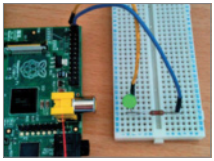
21 To add a timer to the LED so it fades out, we first need to import the time library and then set the 12 pin to have 50Hz frequency to start off with.

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
p = GPIO.PWM(12, 50)  # channel=12 frequency=50Hz
p.start(0)
```

Add the fade

22 Then we add in another try-except block, this time checking what power the LED is at – and once it reaches a certain level, we reverse the process. To run this code, simply save it from nano and then sudo python gpio.py.

```
while 1:
        for dc in range(0, 101, 5):
                p.ChangeDutyCycle(dc)
                time.sleep(0.1)
        for dc in range(100, -1, -5):
                p.ChangeDutyCycle(dc)
                time.sleep(0.1)
        except KeyboardInterrupt:
                pass
p.stop()
GPIO.cleanup()
```



"We'll set the GPIO mode to BOARD. This will work whichever revision of the PI you are using"

What you'll need…

Momentary switch
Suitable wire for PCB projects
Soldering iron and solder
Single pin pair header
HDD/motherboard jumper

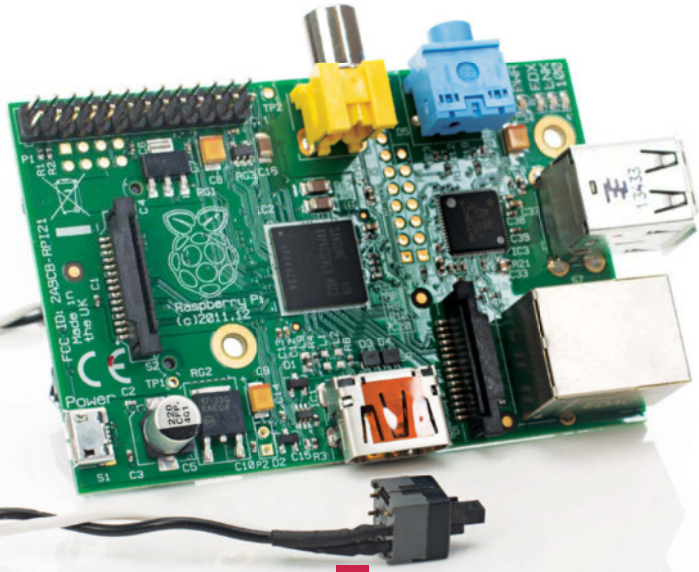
Add a reset switch

Need to restart your Pi after a system lock-up? Ease strain on the mains connector – install a reset switch

Shutting down a Raspberry Pi by removing the power cable is risky. Data may be writing to the SD card, leading to corruption. Repeated removal of the power cable can cause issues with the connector port.

Clearly this can cause problems when faults cause the Raspberry Pi to hang, so the simple fix here is to add a simple reset function to the device. There are three ways this can be done: with a USB reset button, a motherboard jumper on the GPIO bus or with a momentary button connected to newly-soldered pins on the P6 header on the Model B Rev 2 and B+ (a complicated option).

If you have an old PC, retrieving the reset button, cable and the connecting motherboard pins is achievable if you're handy with a soldering iron. Otherwise, we recommend purchasing parts online.



Check your Pi model

01 Only two models feature the P6 header: the Raspberry Pi Model B Rev 2 (which you can find next to the HDMI port) and the B+ (to the left of the '© Raspberry Pi 2014' label). You will need to install the pins manually, however, as they are not pre-installed for this function.

Find your components

02 Header pins can be purchased online, although this will invariably result in having to order more than you need. Alternatively, if you have an old motherboard, remove a pair of pins with a soldering iron. Similarly, you might buy a new reset button or use one from an old PC.

Solder pins to your Pi

03 In order to gain stability when soldering, place the Pi upside down on a layer of packaging foam, with the header slotted into the holes.

Using fine solder, secure the pins to the mainboard with your soldering iron. This will require a very steady hand, so get assistance if required.

Connect your reset switch

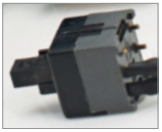
04 Leave the solder to cool for a few minutes before attaching the reset switch connector. Some cases don't have space for the pins and/or the connector, however, so take the time to plan ahead and make sure everything fits. If not, you may need to make some adjustments to your case.

Reset Pi following crashes

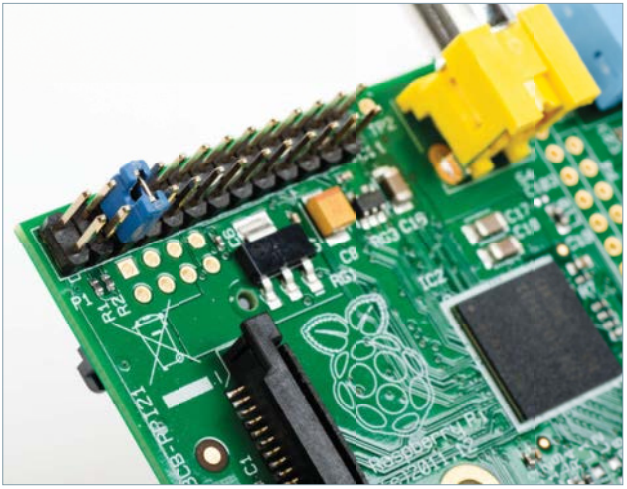
05 With the switch installed, you'll be able to reset the Raspberry Pi when required. Note, however, that this isn't an option to be used for whenever you feel like restarting your Raspberry Pi. Rather, it should be done only when the system fails to respond within a reasonable time frame.

Reset with a HDD jumper

06 Not keen on soldering new pins to your Raspberry Pi? That is perfectly understandable, but it doesn't mean you cannot reset the computer. We have another solution for you. Using a motherboard jumper, two GPIO pins and a script to initiate an ordered shutdown is a simple alternative that doesn't involve solder and potential PCB damage.



"Shutting down a Raspberry Pi by removing the power cable is risky"



Identify the GPIO pins

07 This method works on most models of the Raspberry Pi. Each has a GPIO array, 26 pins on the A and B (Rev 2) and 40 on the A+ and B+. The jumper should be placed on GPIO3, pins 5 and 6 counting from the left with the board the right way around.

Detect jumper with a script

08 Use the script at **bit.ly/1Ge5n0O** to detect the jumper, making it executable (sudo chmod 755) before running. Within a minute your Pi will shut down. Add this line to /etc/crontab to run the script whenever you boot up.

```
@reboot    root    /home/user/scripts/gpio_actions.sh
```

Remember to remove the jumper before booting up!

Try a USB reset button

09 Specialist online stores offer USB reset buttons that can be connected to your Pi for scenarios when the device needs to be rebooted. If the idea of using the HDD jumper or doing some minor soldering doesn't suit you, then a USB reset switch might be your best option.

What you'll need…

AA battery box
bit.ly/1FDiJGa

3-Amp UBEC
bit.ly/1HLKih7

3-Amp terminal strip

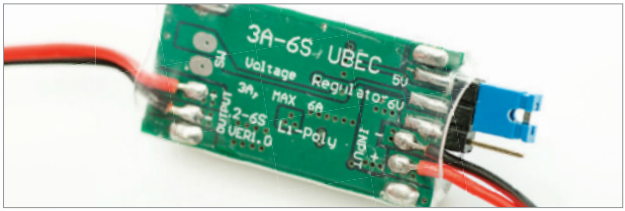
6x AA rechargeable batteries

Add a battery pack to your Raspberry Pi

Don't leave your Raspberry Pi behind – incorporate it into mobile projects by powering it up using humble AA batteries

Your Raspberry Pi's mobility is usually restricted by the length of the power lead. Rather than limiting it to your desk or living room, however, you can use it for mobile projects as diverse as launching it into near-Earth orbit or monitoring and automating your garden.

Of course, to do this you will need batteries, but adding battery power to your Raspberry Pi is simpler than you might have imagined. All that is required are six rechargeable AA batteries (or single-charge alkaline), a battery box with space for the batteries and a UBEC. The latter is a Universal Battery Elimination Circuit, a voltage regulator that will regulate the power supply and prevent damage to the Raspberry Pi, and can be bought for under £10.

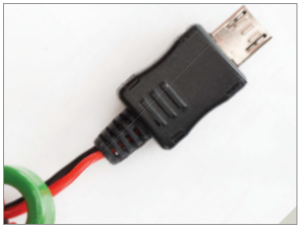


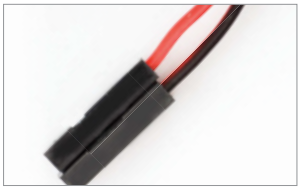
Make your order

01 If you're buying your components online, you should be able to get them all within five days. However, if you're ordering offline (specifically the UBEC), you should avoid traditional electronics stores and instead visit a model enthusiast store, as these circuits are regularly used in RC devices.

Check your UBEC

02 Two types of UBEC are available to choose from. If you used the store that we suggest in the resources box to the left, you'll receive one with a micro USB power connector for easy connection to your Raspberry Pi. However, if you bought one from eBay then there is a strong chance that you will receive one with a 3-pin connector.





Move connector pins

03 In order to use the UBEC with a 3-pin connector, you'll need to alter the position of the pins so that they occupy the two outer slots. Just use a small jeweller's screwdriver to lever up the small plastic catch and remove the red wire from the central slot, before sliding into the unoccupied outer slot.

Connect to battery box

04 With five batteries in the battery box, connect it to the UBEC (red-to-red, black-to-black) by twisting the wires, soldering or employing a 3-amp terminal strip, cut down to two pairs. It can be cut to size using a modelling knife.

Add a battery to boot

05 With your Pi ready to use and your Wi-Fi dongle plugged in, connect the UBEC to the micro USB port and insert the sixth battery into the battery box. The Pi's power and status lights should indicate that the computer is booting up, which gives you a fully portable computer.

Connect the 3-pin UBEC

06 If you purchased the UBEC with the now-modified 3-pin connector, you'll need to connect this to the Raspberry Pi's GPIO header. Connect the positive +5V (red) connector to Pin 2 and the negative 0V connector to Pin 6.

"You can use your device for mobile projects as diverse as launching it into near-Earth orbit"

Measure uptime

07 You should have already set up your Pi for SSH use, so connect to the device via Putty after giving it time to boot fully (at least 60 seconds). In the terminal, enter:

```
sudo dd bs=32m if=/Users/rachelcrabb/Desktop/ArchLinux/archlinux-hf-2013-02-11.img of=/dev/disk1
```

This command will display the system uptime and also keep the Wi-Fi connection active.

Judge your uptime results

08 Uptime results depend upon the type of battery you use and the Raspberry Pi model. Single-charge batteries will last a little bit longer, but this is a more expensive option. Meanwhile, newer models have greater power requirements but run for less time. For more power, add more batteries!

Power extreme!

09 More batteries added in parallel should result in almost double the uptime (at least 16 hours on a 256MB Raspberry Pi Model A), but instead of alkaline or rechargeable batteries you might consider a modern lithium-based AA cell, which will last considerably longer than alkaline batteries.

What you'll need…

Bare Conductive paint (pen or tub)

Male to female jumper wires

An assortment of LEDs, switches and resistors (optional)

Draw circuits with paint

Assembling circuits has never been so easy with the joys of conductive paint, enabling you to bring together art and electronics in a whole new way

Playing with electronics and physical computing is a very rewarding task. For a beginner though, the mess of wires and components can become very confusing quite quickly and things like soldering can be a safety concern when children are involved. Bare Conductive has taken the joy of electronics and made it far safer, easier and more versatile with their conductive paint. You can literally draw wires on paper with a paintbrush, use it for cold-soldering or a conductive adhesive and much, much more. There are not a great deal of boundaries to what you can do. Pair this paint with a microcontroller board and you could be creating interactive art, clothing and projects in no time.

Get your tools

01 Paint and a paintbrush aren't the first items that come to mind when you think about electronics, so you may be wondering where to get them from. Bare Conductive stock the paint and a selection of components in their shop (**bareconductive.com/shop**) but you will need to go somewhere else for art supplies. We would recommend trying a high street craft shop such as Hobbycraft (**hobbycraft.co.uk**) or a local independent.

Pick your platform

02 The great thing about Bare Conductive paint is that, when dry, it works just like normal wiring! That means you can use it with any of your favourite microcontrollers like the Bare Conductive Touch Board, a Raspberry Pi or Adafruit's wearable FLORA platform. Or you can just use some small pin batteries and flashing LEDs for a standalone system.

Start to paint

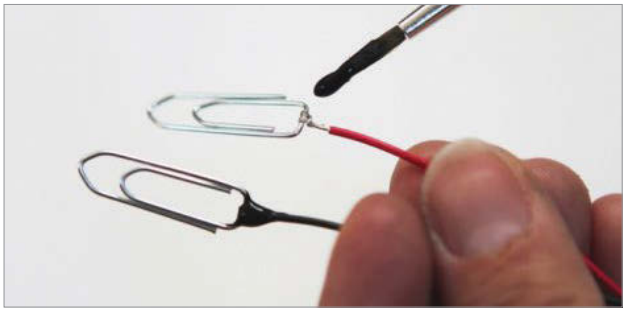
03 You can paint Bare Conductive paint onto pretty much any surface – paper, fabric, walls, clothing, wood, plastic and much more. For really accurate shapes and results, the best idea is to create or purchase a stencil (paper stencils are easiest to make at home but use vinyl for the best edge finish).

Connect it up

04 There are plenty of ways to connect to the conductive paint (from battery packs for example) no matter what surface it's on, because once it is dry it acts just like an uninsulated wire. Therefore you can use wires glued on with the paint, paper clips, bulldog clips, alligator clips or even sewn-in conductive snaps for wearable projects.

Make repairs

05 The conductive paint is thick and when it's dry it becomes quite strong. These means you can use it to cold solder things together and repair any breakages. In other words, you could glue components into a circuit board or glue wires together and they would still function electrically.



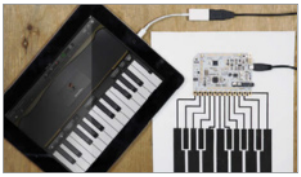
"The mess of wires and components can become very confusing"

Clean up

06 A lot of you are probably thinking that something as cool as conductive paint is going to be nasty stuff. Actually Bare Conductive paint is non-toxic, water-based and water-soluble, and can therefore be cleaned easily with soap and water.

Make it waterproof

07 This paint only comes in black and is not waterproof. However, the great thing is that you can use it underneath or alongside any regular paints, varnishes and waterproofing sprays in order to act as insulation – or just to add some colour into your designs!



Touch and sound

08 Bare Conductive paint can also be used as a capacitive surface, meaning you can use it for touch, gesture or proximity controls when it is paired with a suitable control board. Bare Conductive make their own called the Touch Board which has everything you need to start experimenting with touch and sound. It can even act as a MIDI controller, an interface or an instrument!

Send an SMS from your Raspberry Pi

What you'll need…

Raspberry Pi
Twilio account

Create a program that combines Twilio and simple Python code to enable you to send an SMS from your Pi to a mobile phone

Text messaging, or SMS (Short Message Service), has become a staple of everyday communication. What began life as a 40 pence message service is now offered by most tariff providers as an unlimited service used worldwide. Twilio, a cloud communications company, enables you to send SMS messages for free from your Raspberry Pi to a mobile phone using just six lines of code.

"Create other communication programs, such as making phone calls, recording a call, and retrieving data"

Set up your Twilio account

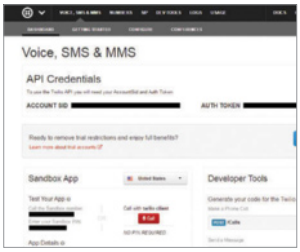
01 The first step of this project is to register for a Twilio account and Twilio number. This is free and will enable you to send an SMS to a registered, verified phone. Once signed up, you will receive a verification code via SMS to the registered phone. When prompted, enter this onto the Twilio site to authenticate your account and phone. Go to **twilio.com/try-twilio** and create your account now.

Register numbers

02 Your Twilio account is just a trial account unless you pay the upgrade fee, which means you can only send and receive communications from a validated phone number. Enter the phone number of the contact who you want to verify, ensuring that you select the correct country code. Twilio will text you a verification code and you will need to enter it into the website form and press submit.

The dashboard

03 Once registered and logged in on Twilio, visit the dashboard page, which will display your AccountSid and your Auth Token. These are both required to use the Twilio REST. Keep these secure and private, but be sure to make a note of them as you will need them for your Python program later.



Install the software

04 Now boot up your Raspberry Pi and connect it to the internet. Before you install the Twilio software, it is worth updating and upgrading your Pi. In the LX Terminal, type sudo apt-get update, then sudo apt-get upgrade. Once complete, type `sudo easy_install twilio` or `sudo pip install twilio` to install the software. (If you need to install pip, type `sudo apt-get install python-pip python-dev`, press Enter, then type `sudo pip install -U pip`.)

Twilio authentication

05 Now you are ready to create the SMS program that will send the text message to your mobile phone. Open your Python editor and import the Twilio REST libraries (line one, below). Next, add your AccountSid and Auth Token, replacing the X with yours, as you will find on your dashboard:

```
from twilio.rest import TwilioRestClient
  account_sid = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
                                # Enter Yours
  auth_token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
                                # Enter Yours
client = TwilioRestClient(account_sid, auth_token)
```

Create your message

06 You will probably want to be able to change your text messages rather than send the same one. Create a new variable in your program called message. This will prompt you to enter the phrase that you want to send to the mobile phone. When the program runs, this is the message that will be sent:

```
message = raw_input("Please enter your message")
```

Add your numbers

07 To send the message, you need to add the code line below and your two phone numbers. The first number is your mobile phone number, which is registered and validated with Twilio (Step 2). The second number is your Twilio account number, which can be retrieved from your dashboard page under 'Call the Sandbox number'. Change the Sandbox number to your country location and remember to add the international country code.

```
    message = client.messages.create(to="+44YOURMOBNUMBER", from_="+44YOURTWILIONUMBER", body=message)
```

Send the message

08 Now send your message. The code below is not required, but is useful to indicate your message has been sent. Add the lines and save your program. Ensure your Raspberry Pi is connected to the internet and that your mobile is on, then run your program. You have just texted from your Raspberry Pi!

```
print message.sid
  print "Your message is being sent"
  print "Check your phone!"
```

Other API and codes

09 You can create other communication programs, such as making phone calls, recording a call, and retrieving data including caller IDs and call duration. The API also complements a wide range of languages, including Ruby, PHP, Java and Node.js (**twilio.com/api**).

What you'll need…

OpenELEC openelec.tv
HDMI cable
USB IR receiver
IR remote
Case
Dedicated power supply
Optional USB storage

Make a Pi 2 HTPC

Finally create a more powerful and capable HTPC using the Raspberry Pi 2 and the excellent OpenELEC project

We know people who just have a Raspberry Pi for XBMC, now called Kodi. It's a great idea and a great use for the Pi – it works just well enough that you can easily play media locally or over the network. The biggest issue came with GUI response on the original Model Bs, and a lack of USB ports for connecting up everything that you want. While optimisation over the last few years has helped, the leap to Raspberry Pi 2 has basically solved all of these problems by giving you much more powerful hardware to play with. So if you're looking to upgrade or finally take the plunge, this handy guide will help you create the perfect Raspberry Pi 2 HTPC.

Choose the software

01 In the past, Pi HTPCs were just a choice between RaspBMC and OpenELEC. However, RaspBMC is on a bit of a hiatus and OpenELEC is your best bet for getting the most up-to-date software. There's not a massive difference between the two, as they both run XBMC.

Get the software

02 Head over to **openelec.tv** and look for the Download section. There's a specific Raspberry Pi section which is split up into original (ARMv6) Pi and the newer Raspberry Pi 2 (ARMv7). Grab the image file from this page for the Pi 2.

Install to card

03 Open up the terminal and use fdisk -l to determine where your SD card is located on your system. Something like /dev/sdb or /dev/mmcblk0 will be ideal. Navigate to the image using cd and install it with dd using:

```
$ dd bs=1M if=OpenELEC-RPi2.arm-5.0.5.img of=/dev/mmcblk0
```

First boot

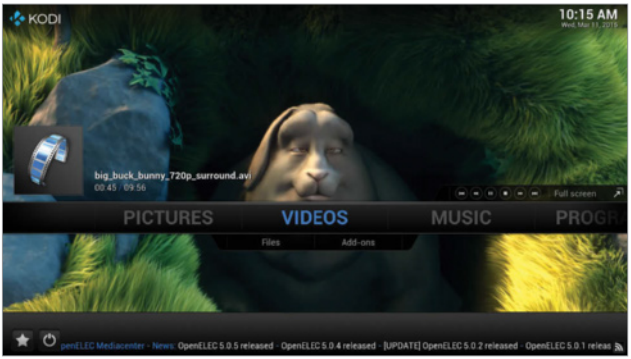
04 Plug in your Raspberry Pi, either to your TV or to another screen just to begin with, and turn it on. OpenELEC will resize the SD card partitions and write a few extra programs before finally booting into Kodi.

Configure Kodi

05 Go through the basic wizard to get through the interface – if you are connecting via wireless you will need to go to OpenELEC in the System menu and activate the wireless receiver before selecting your network and then entering your password.

Add network shares

06 You can attach a portable hard drive or USB stick to the Pi for storage, but it is really to stream over the network. Go to File manager under System and Add source. Go to Browse and choose your network protocol to browse the network or alternatively, add it manually.



"It's a great use for the Pi – it works just well enough that you can easily play media locally or over the network"

Add network shares

07 Placement of your Raspberry Pi is important. As it's going to be out all the time, we highly recommend getting a case for it – the Pibow cases from Pimoroni are quite well suited for this type of use as they are sturdy and can be attached to the rear of some TVs.

IR sensors and controllers

08 Kodi can be controlled with a number of different things – including USB game controllers and compatible IR sensors. We've used FLIRC in the past, but if you have your Pi behind the TV, you'll need a sensor on a wire that can stretch to a useful position.

Future updates

09 OpenELEC has the excellent ability to update itself without needing you to reinstall it every few months, meaning you won't need to do much maintenance on it at all. Now you can sit back and enjoy your media much easier than before.

What you'll need…

Latest Raspbian image
USB printer
USB wireless card

Print wirelessly with your Pi

Breathe new life into an old printer by using your Raspberry Pi as a wireless print server

Wireless printing has made it possible to print to devices stored in cupboards, sheds and remote rooms. You don't have to own a shiny new printer for this to work; old printers without native wireless support don't have to end up in the bin, thanks to the Raspberry Pi.

The setup is simple. With your Pi set up with a wireless USB dongle, you connect your printer to a spare USB port on the computer. With Samba and CUPS (Common Unix Printing System) installed on the Raspberry Pi, all that is left to do is connect to the wireless printer from your desktop computer, install the appropriate driver and start printing.

CUPS gives the Raspberry Pi a browser-based admin screen that can be viewed from any device on your network, enabling complete control over your wireless network printer.

Below Setting your Raspberry Pi to print wirelessly is a great way to get rid of annoying cables at your workstation

Check your printer works

01 Before starting, check that the printer you're planning to use for the project still works and has enough ink. The easiest way to do this is to check the documentation (online if you can't find the manual) and run a test print.

Detect your printer

02 With your Raspberry Pi set up as usual and the printer connected to a spare USB port, enter:

```
lsusb
```

This will confirm that the printer has been detected by your Raspberry Pi. In most cases you should see the manufacturer and model displayed.

Set up print admin

04 Set up the CUPS print admin tool. Boot into the GUI (startx) and launch the browser, entering 127.0.0.1:631. Switch to Administration, before ensuring that the 'Share printers' and 'Allow remote administration' boxes are selected. Select Add Printer and proceed to enter your Raspbian username and password.

Install Samba and CUPS

03 Install Samba on your Pi to enable file and print sharing across the entire network:

```
sudo apt-get install samba
```

Next, install CUPS:

```
sudo apt-get install cups
```

With a print server created, begin configuration by adding default user 'pi' to the printer admin group:

```
sudo usermod -a -G lpadmin pi
```

Add your printer

05 A list of printers will be displayed, so select yours to proceed to the next screen where you can confirm the details, add a name and check the Share This Printer box. Click Continue to load the list of printer drivers and select the appropriate one from the list.

Configure Samba for network printing

06 Using a Windows computer for printing? Samba will need some configuration. Open '/etc/samba/smb.conf' in nano, search (Ctrl+W) for '[printers]' and find 'guest ok' which you should change as follows:

```
guest ok = yes
```

Next, search for "[print$]." Then change the path as follows:

```
path = /usr/share/cups/drivers
```

Join a Windows workgroup

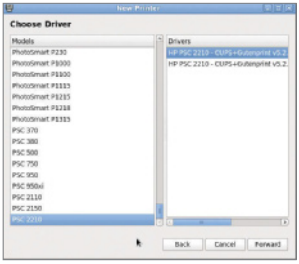
07 With these additions made, search for "workgroup" in the configuration file and then add your workgroup:

```
workgroup = your_workgroup_name
```

```
wins support = yes
```

Make sure you uncomment the second setting so that the print server can be seen from Windows. Next, save your changes and then restart Samba:

```
sudo /etc/init.d/samba restart
```



Accessing your printer

08 Meanwhile, it's a lot easier to access your wireless printer from a Linux, Mac OS X or other Unix-like system, thanks to CUPS. All you need to do is add a network printer in the usual way and the device will be displayed.

Add AirPrint compatibility

09 It's also possible to print wirelessly from your Apple iPad using Apple's AirPrint system. To do this, you need to add the Avahi Discover software:

```
sudo apt-get install avahi-discover
```

Your wireless printer will now be discoverable from your iPad or iPhone and will be ready to print.

What you'll need…

RetroPie
blog.petrockblock.com/download/retropie-project-image/

Did you know…

Emulation is something of a legal minefield. You must own the original games to emulate them on your Pi.

Make your own retro games console

Get your retro gaming fix with RetroPie, a distro for getting the games of yesteryear onto your Pi

There's a growing trend for people to create their own arcade cabinet or hack together their own retro console, and the Raspberry Pi's size and power makes it perfect for this. Follow our tutorial to turn your Pi into a fully functional emulating powerhouse.

Take control
Properly configure USB and PS3 controllers out of the box

Fast and flexible
Configure RetroPie for more power control

Xbox-friendly
Install drivers to use the Xbox 360 controller

Decades of fun
Turn your Raspberry Pi into the ultimate portable retro console

Install RetroPie

01 Download the latest RetroPie image from the website and unzip it. Like installing other Raspberry Pi distros, you simply need to write the image to the SD card with:

```
$ sudo dd bs=4M if=[path to image] of=/dev/[path to sd card]
```

Initial setup

02 On the first boot, you'll be asked to configure a controller, which can be done with a keyboard, a standard USB controller or a PS3 controller. The initial setup on RetroPie is for very limited controls, and you'll need to launch a separate tool for better config.

Calibrating controllers

03 Press the menu button you set up, and go to exit. You'll get to a command line. Connect a USB controller and enter the following:

```
$ cd RetroPie/emulators/RetroArch/tools
```

…and then:

```
$ ./retroarch-joyconfig >> ~/RetroPie/configs/all/retroarch.cfg
```

Follow on the on-screen instructions to properly configure your controller.

Using an Xbox 360 pad

04 To use a wired or PC-compatible Xbox 360 controller, you'll first need to install the correct drivers:

```
$ sudo apt-get install xboxdrv
```

Then edit `/etc/rc.local` by adding:

```
xboxdrv --trigger-as-button --wid 0 --led 2 --deadzone 4000 --silent &
sleep 1
```

…before exit 0. Change --wid to --id if it's a wired controller. Reboot.

Recognising the 360 controller

05 Add the 360 pad to your configuration file by first going to tools with:

```
$ cd ~/RetroPie/emulators/RetroArch/tools
```

Then type in the following:

```
$ ./retroarch-joyconfig -o p1.cfg -p 1 -j 0
```

And finally add the files to RetroArch with:

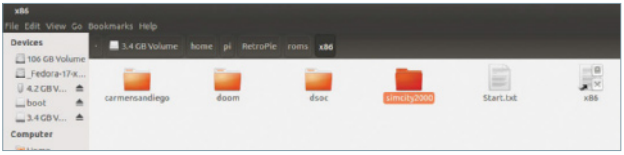
```
$ sudo cat p*.cfg >> ~/RetroPie/configs/all/retroarch.cfg
```

Save and reboot.

Add some games

06 Adding games to your Raspberry Pi is fairly simple. Grab the SD card from the Pi and put it in your PC. Move any ROMs or compatible PC files to the relevant folders in:

`[path to SD card]/home/pi/RetroPie/roms/`

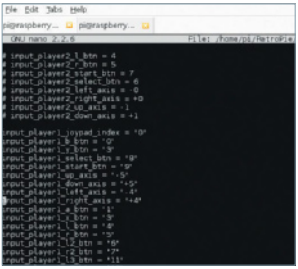


Navigate RetroPie

07 RetroPie will automatically know if there are games for a specific emulator and allow you to access it. Pressing left or right will move between the emulators, which then have options to load the games on the SD card.

Two players

08 For multiplayer gaming, it's best to use two controllers of the same type. Go to the bottom of `~/RetroPie/configs/all/retroarch.cfg` and copy and paste the code from `input_player1_joypad_index = "0"` to the bottom. Change each instance of player1 to player 2, and a second controller will now work.



Safe restart

09 Some of the emulators can't be quit out of, meaning you'll need to physically reboot your Pi by unplugging it each time. We can add a hotkey to exit the emulators by again going to `retroarch.cfg` and adding to the end:

```
input_enable_hotkey_btn = "X"
input_exit_emulator_btn = "Y"
```

…with X and Y being the corresponding numbers of buttons on your controller.

What you'll need…

Portable speakers or headphones

Sonic Pi
www.cl.cam.ac.uk/projects/raspberrypi/sonicpi/teaching.html

Did you know…

Sonic Pi is installed by default on the latest versions of the Raspbian operating system. If it's missing, follow Step 1!

Make music

Program your own melodies using Sonic Pi and create musical cues or robot beeps

One of the major features of Scratch is its ability to teach the fundamentals of coding to kids and people with no computing background. For kids, it's especially appealing due to the way it allows them to create videogames to interact with as part of their learning. In this kind of vein then, Sonic Pi teaches people to code using music. With a simple language that utilises basic logic steps but in a more advanced way than Scratch, it can either be used as a next step for avid coders, or as a way to create music for an Internet of Things or a robot.

"We can start making more complex melodies by using more of Sonic Pi's functions"

Below Sonic Pi is a great way to learn basic coding principles and have fun

Getting Sonic Pi

01 If you've installed the latest version of Raspbian, Sonic Pi will be included by default. If you're still using a slightly older version, then you'll need to install it via the repos. Do this with:

```
$ sudo apt-get install sonic-pi
```

Start with Sonic Pi

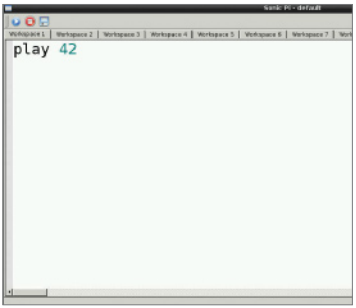
02 Sonic Pi is located in the Education category in the menus. Open it up and you'll be presented with something that looks like an IDE. The pane on the left allows you to enter the code for your project, with proper syntax highlighting for its own style of language. When running, an info pane details exactly what's being played via Sonic Pi – and any errors are listed in their own pane as well, for reference.

Your first note

03 Our first thing to try on Sonic Pi is simply being able to play a note. Sonic Pi has a few defaults preset, so we can get started with:

```
play 50
```

Press the Play button and the output window will show you what's being played. The pretty_bell sound is the default tone for Sonic Pi's output, and 50 determines the pitch and tone of the sound.



Full code listing

```
with_tempo 200
play_pattern [40,25,45,25,25,50,50]
2.times do
    with_synth "beep"
    play_pattern [40,25,45,25,25,50,50]
    play_pattern [40,25,45,25,25,50,50].reverse
end
play_pad "saws", 3
in_thread do
    with_synth "fm"
    6.times do
        if rand < 0.5
            play 30
        else
            play 50
        end
        sleep 2
    end
end

2.times do
    play_synth "pretty_bell"
    play_pattern [40,25,45,25,25,50,50]
    play_pattern [40,25,45,25,25,50,50].reverse
end
```

Set the beat

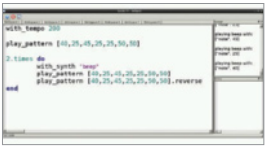
04 For any piece of music, you'll want to set the tempo. We can start by putting:

```
with_tempo 200
```

…at the start of our code. We can test it out by creating a string of midi notes using play_pattern:

```
play_pattern
[40,25,45,25,25,50,50]
```

This will play pretty_bell notes at these tones at the tempo we've set. You can create longer and shorter strings, and also change the way they play.



Advance your melody

05 We can start making more complex melodies by using more of Sonic Pi's functions. You can change the note type by using with_synth, reverse a pattern, and even create a finite loop with the x.times function; do and end signify the start and end of the loop. Everything is played in sequence before repeating, much like an if or while loop in normal code.

Playing a concert

06 Using the in_thread function, we are able to create another thread for the Sonic Pi instance, and have several lines of musical code play at once instead of in sequence. We have made it create a series of notes in a random sequence, and have them play alongside extra notes created by the position and velocity of the mouse using the play_pad function.

What you'll need…

Portable USB speakers
python-espeak module
eSpeak
Raspbian (latest image)

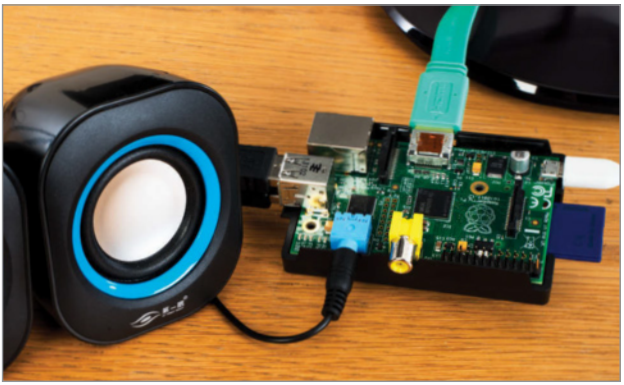
Voice synthesizer

Add the power of speech to your Raspberry Pi projects with the versatile eSpeak Python library

We've shown how the Raspberry Pi can be used to power all kinds of projects, but as a tiny computer it can also be the centre of an Internet of Things in your house too. For these reasons and more, using the Raspberry Pi for text-to-voice commands could be just what you're looking for. Due to the Debian base of Raspbian, the powerful eSpeak library is easily available for anyone looking to make use of it. There's also a module that allows you to use eSpeak in Python, going beyond the standard command-line prompts so you can perform automated tasks.

Did you know…

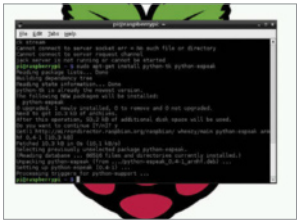
Using eSpeak you can control the way the words are spoken to add emphasis or make the voice sound different.



Everything you'll need

01 We'll install everything we plan to use in this tutorial at once. This includes the eSpeak library and the Python modules we need to show it off. Open the terminal and install with:

```
$ sudo apt-get install espeak python-espeak python-tk
```



Pi's first words

02 The eSpeak library is pretty simple to use – to get it to just say something, type in the terminal:

```
$ espeak "[message]"
```

This will use the library's defaults to read whatever is written in the message, with decent clarity. Though this simple command is fun, there's much more you can do…

Say some more

03 You can change the way eSpeak will read text with a number of different options, such as gender, read speed and even the way it pronounces syllables. For example, writing the command like so:

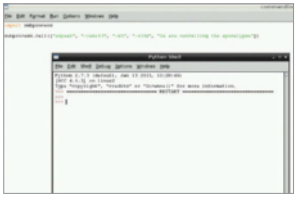
```
$ espeak -ven+f3 -k5 -s150 "[message]"
```

…will turn the voice female, emphasise capital letters and make the reading slower.

Take command with Python

04 The most basic way to use eSpeak in Python is to use **subprocess**. Import it, then use:

```
subprocess.call(["espeak", "[options 1]", "[option 2]",..."[option n]", "[your message here]"])
```



The native tongue

05 The Python eSpeak module is quite simple to use to just convert some text to speech. Try this sample code:

```
from espeak import espeak
espeak.synth("[message]")
```

You can then incorporate this into Python, like you would any other module, for automation.

A voice synthesiser

06 Using the code listing, we're creating a simple interface with Tkinter with some predetermined voice buttons and a custom entry method. We're showing how the eSpeak module can be manipulated to change its output. This can be used for reading tweets or automated messages. Have fun!

Full code listing

Import the necessary eSpeak and GUI modules, as well as the module to find out the time

Define the different functions that the interface will use, including a simple fixed message, telling the time and a custom message

Create the basic window with Tkinter for your interface, as well as creating the variable for text entry

The text entry appends to the variable we created, and each button calls a specific function that we defined above in the code

```
from espeak import espeak
from Tkinter import *
from datetime import datetime

def hello_world():
    espeak.synth("Hello World")

def time_now():
    t = datetime.now().strftime("%k %M")
    espeak.synth("The time is %s"%t)

def read_text():
    text_to_read = input_text.get()
    espeak.synth(text_to_read)

root = Tk()
root.title("Voice box")
input_text = StringVar()
box = Frame(root, height = 200, width = 500)
box.pack_propagate(0)
box.pack(padx = 5, pady = 5)

Label(box, text="Enter text").pack()
entry_text = Entry(box, exportselection = 0, textvariable = input_text)
entry_text.pack()
entry_ready = Button(box, text = "Read this", command = read_text)
entry_ready.pack()

hello_button = Button(box, text = "Hello World", command = hello_world)
hello_button.pack()
time_button = Button(box, text = "What's the time?", command = time_now)
time_button.pack()

root.mainloop()
```

"There's even a module that allows you to use eSpeak in Python, so you can perform automated tasks"

What you'll need…

Internet connectivity

Web browser

Google Coder
googlecreativelab.github.io/coder/raspberrypi/sonicpi/teaching.html

Build your first web server

Use Google Coder to turn your Raspberry Pi into a tiny, low-powered web server and web host!

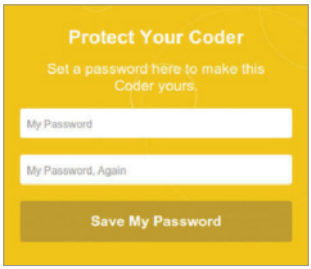
We're teaching you how to code in many different ways on the Raspberry Pi in this book, so it only seems fitting that we look at web development too.

There's a new way to use the web on the Raspberry Pi as well: internet giant Google has recently released Coder specifically for the tiny computer. It's a Raspbian-based image that turns your Pi into a web server and web development kit. Accessible easily over a local network and with support for jQuery out of the box, it's an easy and great way to further your web development skills.

Get Google Coder

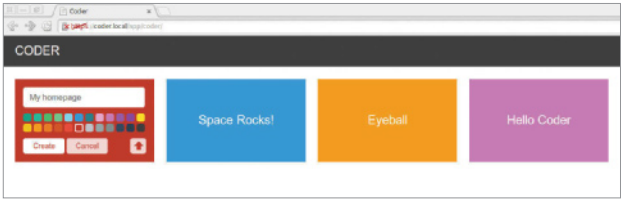
01 Head to the Google Coder website, and download the compressed version of the image. Unpack it wherever you wish, and install it using dd, like any other Raspberry Pi image:

```
$ dd if=[path to]/raspi.img
of=/dev/[path to SD card] bs=1M
```



Plug in your Pi

02 For this tutorial, you'll only need to connect a network cable into the Pi. Pop in your newly written SD card, plug in the power and wait a few moments. If you've got a display plugged in anyway, you'll notice a Raspbian startup sequence leading to the command-line login screen.



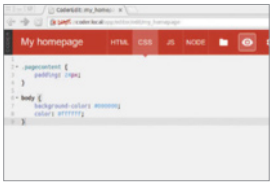
Connect to Coder

03 Open up the browser on your main system, and go to **http://coder.local**. You may have to manually accept the licence. It will ask you to set up your password, and then you'll be in and ready to code.

Language of the web

04 Now it's time to create your own app or website. Click on the '+' box next to the examples, give your app a name and then click Create. You'll be taken to the HTML section of the app. Change the Hello World lines to:

```
<h1>This is a HTML header</h1>
<p>This is a new block of default text</p>
```



Styled to impress

05 Click on the CSS tab. This changes the look and style of the webpage without having to make the changes each time in the main code. You can change the background colour and font with:

```
body {
        background-color: #000000;
        color: #ffffff;
}
```

Querying your Java

06 The third tab allows you to edit the jQuery, making the site more interactive. We can make it create a message on click with:

```
$(document).click(function()
{
        alert('You clicked the website!');
        }
);
```

Full code listing

HTML

Some simple HTML code that can point us to some important websites. The h2 tag is used to display the time thanks to Java

```
<h1>Welcome to the internet...</h1>
<h2></h2>
<p><a href="http://www.linuxuser.co.uk">Linux User & Developer</p>
<p><a href="http://www.reddit.com/">Reddit</p>
<p><a href="http://www.linuxfoundation.org/">The Linux Foundation</p>
<p><a href="http://www.fsf.org/">Free Software Foundation</p>
```

Java

We're calling the current time using jQuery in the JS tab so that we can ultimately display it on the webpage

We're going to display the time as a 12-hour clock in the first if statement, and use AM and PM to differentiate the time

We make the minutes readable by adding a 0 if it's below 10, then concatenate all the variables and assign to the tag h2

```
var d = new Date;
    var hours = d.getHours();
    var mins = d.getMinutes();
    if (hours > 12) {
        var hour = (hours - 12);
        var ampm = "PM";
    }
    else {
        var hour = hours;
        var ampm = "AM";
    }
    if (hours == 12) {
        var ampm = "PM";
    }
    if (mins > 9){
        var min = mins;
    }
    else {
        var min = "0" + mins;
    }
    var time = "The time is " + hour + ":" + min + " " + ampm;
    $("h2").html(time);
```

"Coder is a Raspbian-based image that turns your Raspberry Pi into a web server and web development kit. It's an easy and great way to further your skills"

Code your own Twitter bot

Create your very own Twitter bot that can retweet chunks of wisdom from others

Twitter is a useful way of sharing information with the world and it's our favourite method of giving our views quickly and conveniently. Many millions of people use the microblogging platform from their computers, mobile devices and possibly even have it on their televisions.

You don't need to keep pressing that retweet button, though. With a sprinkling of Python, you can have your Raspberry Pi do it for you. Here's how to create your own Twitter bot…

What you'll need…

Internet connectivity

Latest version of Raspbian
www.raspberrypi.org/downloads

Installing the required software

01 Log into the Raspbian system with the username Pi and the password raspberry. Get the latest package lists using the command sudo apt-get update. Then install the Python Package installer using sudo apt-get install python-pip. Once you've done that, run sudo pip install twython to install the Twitter library we'll be using.

Registering your 'app' with Twitter

02 We need to authenticate with Twitter using OAuth. Before this, you need to go to **https://dev.twitter.com/apps** and sign in with the account you'd like your Pi to tweet from. Click the 'Create a new application' button. We called our application 'LUD Pi Bot' and set the website to **www.linuxuser.co.uk**.

Full code listing

```
#!/usr/bin/env python2

# A Twitter Bot for the Raspberry Pi that retweets any
content from

import sys
import time
from datetime import datetime
from twython import Twython

class bot:
    def __init__(self, c_key, c_secret, a_token, a_token_
secret):
        # Create a Twython API instance
        self.api = Twython(c_key, c_secret, a_token,
a_token_secret)

        # Make sure we are authenticated correctly
        try:
            self.api.verify_credentials()
        except:
            sys.exit("Authentication Failed")

        self.last_ran = datetime.now()

    @staticmethod
```

Creating an access token

03 Go to the Settings tab and change the Access type from 'Read only' to 'Read and Write'. Then click the 'Update this Twitter application's settings' button. Next we create an access token. Click the 'Create my access token' button. If you refresh the details page, you should have a consumer key, a consumer secret and access token, plus an access token secret. This is everything we need to authenticate with Twitter.

Authenticating with Twitter

04 We're going to create our bot as a class, where we authenticate with Twitter in the constructor. We take the tokens from the previous steps as parameters and use them to create an instance of the Twython API. We also have a variable, last_ran, which is set to the current time. This is used to check if there are new tweets later on.

Retweeting a user

05 The first thing we need to do is get a list of the user's latest tweets. We then loop through each tweet and get its creation time as a string, which is then converted to a datetime object. We then check that the tweet's time is newer than the time the function was last called – and if so, retweet the tweet.

The main section

06 The main section is straightforward. We create an instance of the bot class using our tokens, and then go into an infinite loop. In this loop, we check for any new retweets from the users we are monitoring (we could run the retweet task with different users), then update the time everything was last run, and sleep for five minutes.

```
    def timestr_to_datetime(timestr):
        # Convert a string like Sat Nov 09 09:29:55 +0000
        # 2013 to a datetime object. Get rid of the timezone
        # and make the year the current one
        timestr = "{0} {1}".format(timestr[:19], datetime.
now().year)

        # We now have Sat Nov 09 09:29:55 2013
        return datetime.strptime(timestr, '%a %b %d %H:%M:
%S %Y')

    def retweet_task(self, screen_name):
        # Retweets any tweets we've not seen
        # from a user
        print "Checking for new tweets from
@{0}".format(screen_name)

        # Get a list of the users latest tweets
        timeline = self.api.get_user_timeline
(screen_name = screen_name)

        # Loop through each tweet and check if it was
        # posted since we were last called
        for t in timeline:
            tweet_time = bot.timestr_to_datetime
(t['created_at'])
            if tweet_time > self.last_ran:
                print "Retweeting {0}".format(t['id'])
                self.api.retweet(id = t['id'])

if __name__ == "__main__":
    # The consumer keys can be found on your application's
    # Details page located at https://dev.twitter.com/
    # apps(under "OAuth settings")
    c_key=""
    c_secret=""

    # The access tokens can be found on your applications's
    # Details page located at https://dev.twitter.com/apps
    # (located under "Your access token")
    a_token=""
    a_token_secret=""

    # Create an instance of the bot class
    twitter = bot(c_key, c_secret, a_token, a_token_secret)

    # Retweet anything new by @LinuxUserMag every 5 minutes
    while True:
        # Update the time after each retweet_task so we're
        # only retweeting new stuff
        twitter.retweet_task("LinuxUserMag")
        twitter.last_ran = datetime.now()
        time.sleep(5 * 60)
```

What you'll need…

ADXL accelerometer
shop.pimoroni.com

Latest Raspbian Image
www.raspberrypi.org/downloads

Breadboard
shop.pimoroni.com

Male-to-female prototyping cables
shop.pimoroni.com

30W soldering iron
shop.pimoroni.com

Use an accelerometer

Adding tilt technology to your next Raspberry Pi project is easier than you might think

In this project we'll be using an ADXL345 accelerometer kit to create a prototype controller for a Space Invaders-style game we're working on. The kit is only £20, but does require about 15 minutes of basic soldering. Beyond that, the physical setup is really quite simple. We'll be placing the ADXL345 on our breadboard 'controller' and using just four wires to communicate with the Raspberry Pi's GPIO pins using I2C. You'll have a full set-up guide available from the retailer.

Let's start by making sure your Pi is up to date. Open a terminal window and type: `sudo apt-get update && sudo apt-get upgrade`

Install i2c-tools

01 In a terminal, type: `sudo apt-get install i2c-tools` followed by `sudo nano /etc/modprobe.d/raspi-blacklist.conf`. This will open a file in nano – locate the line 'blacklist i2c-bcm2708' and comment it out so it looks like this:

```
#blacklist i2c-bcm2708
```

Save with Ctrl+X, then press Y and Enter to exit. Finally, edit the /etc/modules file with nano, adding the line `i2c-bcm2708` `i2c-dev` to ensure it loads every time you boot. Reboot to finish the operation.

Connect the hardware

02 We'll also need to install the smbus Python library. Type `sudo apt-get install python-smbus` in the terminal. Then wire up the accelerometer to the Pi with male-to-female prototyping cables. The correct pins are marked on the ADXL345, but check our pictures for where to plug things into the Pi.

Full code listing

```
import pygame
from adxl345 import ADXL345

adxl345 = ADXL345() # Initialise the accelerometer
pygame.init() # Initialise Pygame

# Create a screen of 800x600 resolution
screen = pygame.display.set_mode([800, 600])

# Name the game window
# Set the mouse visibility and start an FPS clock
pygame.display.set_caption('ADXL345 Space Test - Press ESC to quit')
pygame.mouse.set_visible(False)
clock = pygame.time.Clock()

# Load the images we're using from:
# http://opengameart.org/users/rawdanitsu
background_image = pygame.image.load("Space-Background-4_0.jpg").convert()
player_image = pygame.image.load("ship0.png").convert()

# We can use colour key method to remove the
# background from the ship
player_image.set_colorkey([0, 0, 0])

player_position = [450, 350] # Initial ship tarting point
game_over = False # decide if the game should end
```

"We'll be placing the ADXL345 on our breadboard 'controller' using I2C"

```
def update_pos():
    """ Poll the adxl345 and update player pos based on readings"""
    move_data = adxl345.getAxes(True) # Returns a dict of axes results
    if move_data['x'] < -0.1 or move_data['x'] > 0.1:
        player_position[0] += move_data['x'] * 20
    if move_data['y'] < -0.1 or move_data['y'] > 0.1:
        player_position[1] += move_data['y'] * 20

def check_pos():
    """ Check player pos to make it
    wrap-around the game window"""
    if player_position[0] > 850:
        player_position[0] = -75
    elif player_position[0] < -75:
        player_position[0] = 850

    if player_position[1] > 650:
        player_position[1] = -75
    elif player_position[1] < -75:
        player_position[1] = 650

###### MAIN PROGRAM LOOP ######
while not game_over: # Handle control events while the game is in play
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True # Quit if close button is pressed
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                game_over = True
                # Quit if escape key is pressed

    update_pos() # Update the player's position
    check_pos() # Check the player's position

    # Update the background then the player's position on the screen
    screen.blit(background_image, [0, 0])
    screen.blit(player_image, [player_position[0],player_position[1]])

    pygame.display.flip() # Refresh the screen
    clock.tick(20) # Force frame-rate to desired number

pygame.quit() #Quit gracefully when 'game_over' is True
```

Final checks

03 With accelerometer attached to Pi, power it up, open a terminal window and type: `sudo i2cdetect -y 1`. The '1' at the end assumes you're using a Rev 2 Pi – replace it with '0' if yours is older. You should see some numbers denoting that the ADXL345 has been recognised. Finally, we'll use Git to grab the project – type the following:

```
sudo apt-get install git
```

Cloning and testing

04 From the terminal, navigate to your home folder (with cd ~) and create a project folder for the project to live in using: `mkdir adxl345_project`. Cd into the folder and from the terminal, type the following to clone the finished project from our **Github.com** account: `git clone https://github.com/russb78/adxl345_project.git`

How it works

05 To run the project from the **adxl345_project** folder type:

```
sudo python adxl_345_project.py
```

The Pygame window will open. Pick up your breadboard 'controller' and tilt it around. The ship should move around the screen in reaction to your movements. If it's backwards, you can adjust the orientation of the breadboard – it might just be 'upside down'!

Why it works

06 The code for this project is an excellent backbone for a Space Invaders-style game written in Python and Pygame. The code collects data from the ADXL345 accelerometer in the **update_pos()** function using the **move_data** variable. This data is piped to the **player_position** array – which is in the form of x and y co-ordinates. The move_data pushes the image of the spaceship around the screen.

What you'll need…

Latest Raspbian Image
www.raspberrypi.org/downloads

Breadboard

Connectors

Jumper wire

DSLR camera

Compatible shutter cable

Time-lapse camera trigger

Make shooting time-lapse video with your DSLR camera a cinch with our expert advice

You'd be forgiven for thinking that creating mesmerising time-lapse videos like those of Vincent Laforet (**www.laforetvisuals.com**) or John Eklund (**www.theartoftimelapse.com**) might be out of reach of the Average Joe. With the help of the Raspberry Pi and a sprinkling of Python code, though, that's no longer the case. In this guide we're going to trigger our DSLR camera to create pixel-perfect time-lapse imagery…

Set up the Raspberry Pi

01 For this tutorial we're assuming you're using a recent build of Raspbian. With the Raspberry Pi set up with a keyboard, mouse and monitor, open the terminal and type:

```
sudo apt-get update
```

Install the RPi.GPIO library

02 Next we want to make sure your development environment is set up. Follow these steps to make sure you're all set. In the terminal, type:

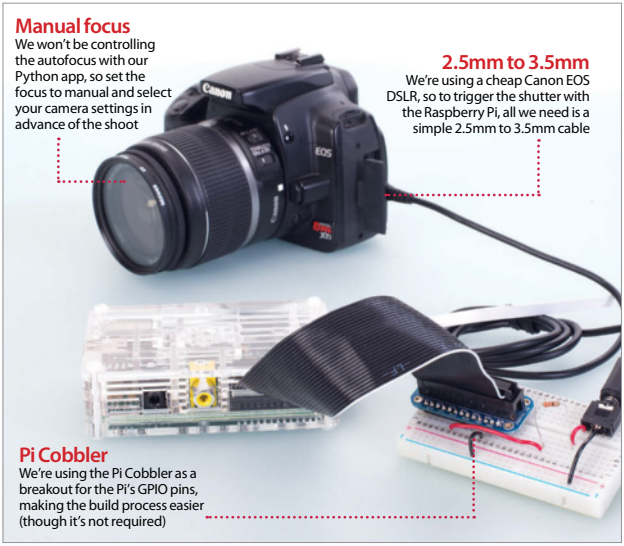
```
suda atp-get install python-dev
sudo apt-get install python-rpi.gpio
```

Set up the Pi Cobbler

03 For this tutorial we've used a cheap prototyping breadboard and an Adafruit Pi Cobbler to give us easy access to the Raspberry Pi's GPIO pins. As you can see from the picture, the Cobbler straddles the centre-point of the breadboard and a ribbon cable connects the two.

Full code listing

```
import RPi.GPIO as GPIO
import time

print '\nWelcome to the Complete Manual Time-lapse Tool.'
print "Just tell us how many shots you'd like to take and the interval between them.\n"
print "Try googling 'time-lapse interval calc' if you need help deciding.\n"


def main():
    shots = raw_input('How many shots would you like to take?\n ->')
    interval = raw_input('How frequently do you want to take them (in seconds)?\n ->')

    if shots.isdigit() and interval.isdigit():
        shots = int(shots)
        interval = int(interval)

        print "You'll be shooting for %d minutes.\n" % (shots * interval / 60)
        answer = raw_input('Are you ready to proceed?(yes/no):')
        confirm = answer.lower() in ['yes', 'y']

        if confirm:
            GPIO.setmode(GPIO.BOARD)
            GPIO.setup(16, GPIO.OUT)
            taken = 1
            print
            print 'Starting a run of %d shots' % (shots)
```




```
            for i in range(0, shots):
                print
                print 'Shot %d of %d' % (taken, shots)
                taken +=1
                GPIO.output(16, GPIO.HIGH)
                time.sleep(0.5)
                GPIO.output(16, GPIO.LOW)
                time.sleep(interval)
            GPIO.cleanup()
        else:
            print "Let's try again (or press Ctrl + C to quit):\n"
            main()
    else:
        print "Oops - You can only enter numbers. Let's try again:\n"
        main()

    print
    print 'Thanks for using the Complete Manual Time-lapse Tool!\n'
    again = raw_input('Would you like to do another time-lapse? (yes/no):\n -> ')
    proceed = again.lower() in ['yes', 'y']

    if proceed:
        main()
    else:
        print '\nSee you next time!\n'
        quit()

if __name__ == '__main__':
    main()
```

Configure the breadboard

04 For the Raspberry Pi's GPIO to control the camera, we need to create a circuit between a pin on the GPIO (in this case pin 23 on the Cobbler – but it's actually physical pin 16) and the pin that connects to the 'head' or 'tip' of the camera cable that activates the shutter when connected. The base of the connector cable is always ground, so make sure you ground the 'GND' pin on the Cobbler and the middle pin on the audio jack. With the circuit complete, we can focus on the code.

The Time-lapse Photography Tool

05 We've created a small 55-line Python utility called The Linux User Time-lapse Photography Tool, which asks the user to input how many shots they'd like to take and the frequency they'd like them taken. It then takes that information and uses it in a For loop to activate the shutter using GPIO pin 16 . If you'd like to use the project 'in the field' we'd recommend using the Android app ConnectBot to SSH into your RasPi for input and feedback. Don't forget to start your script with `sudo python time_lapse_camera.py`

Creating a video

06 With your camera packed with images, we need to collect and output them as a video file. While it's possible on the Pi, copy them to an easily accessible folder on a separate Linux PC to make it much faster. We're going to use FFmpeg. With the terminal open in the folder where your images are stored, type: `ffmpeg -f image2 -i image%04d.jpg -vcodec libx264 -b 800k video.avi`. This assumes you have libx264 installed on your machine and the '`image%04d.jpg`' assumes the file format and the number of digits it's dealing with (in this case: 'picture0001.jpg').

What you'll need…

A wireless internet connection

2 x momentary switches

4 x female-to-male leads (to connect your Pi to a breadboard)

2 x 220-ohm resistors

4 x male-to-male leads

Speakers connected to a 3.5mm headphone jack

Build a portable internet radio

Turn your Raspberry Pi into a portable Wi-Fi streaming radio

There are thousands of free radio stations on the internet, and with this project you can listen to all of them from one tiny little box. So let's build our streaming radio using a Raspberry Pi, a speaker and a few odds and ends…

Let's get set up

01 Firstly, we need to prepare our Pi. Using Raspbian, and a Pi connected to the internet, open a terminal and switch to the root user:

```
sudo su
```

And update your list of packages, then upgrade your Pi to the latest software:

```
apt-get update && apt-get upgrade -y
```

Install some extra packages

02 We need to install the Python packages to access the GPIO. In a terminal, logged in as root, enter the following.

```
apt-get install python-rpi.gpio
```

Now install MPlayer, which is what will be playing our audio.

```
apt-get install mplayer
```

Make the files executable and Install

03 To install the tools, we need to navigate to **PiAUISuite-master/Install**. We now need to make InstallAUISuite.sh executable for all users, so use:

```
chmod 777  InstallAUISuite.sh
```

Now that the file is executable, let's install:

```
sudo ./ InstallAUISuite.sh
```

Full code listing

We import the RPi GPIO library and set it to use BCM numbering system

Here we set up pins 23 and 24, which control the radio station selection

Here we say that if the button is pressed for 23, run the command below

Here we see the script kill any open MPlayer processes and then load the radio station

```
#!/usr/bin env python
import time import sleep
import os
import RPi.GPIO as GPIO
# I found loads of BBC Radio streams
# from http://bbcstreams.com/
GPIO.setmode(GPIO.BCM)
GPIO.setup(23 , GPIO.IN)
GPIO.setup(24 , GPIO.IN)
while True:
    if GPIO.input(23)==1:
        os.system('sudo killall mplayer')
        os.system('mplayer -playlist http://bbc.co.uk/ ↵
radio/listen/live/r1.asx &')
    if GPIO.input(24)==1:
        os.system('sudo killall mplayer')
        os.system('mplayer -playlist http://bbc.co.uk/ ↵
radio/listen/live/r6.asx &')
    sleep(0.1);
GPIO.cleanup()
```

Set up the software

04 Copy **radio.py** from **bit.ly/1JphBL0** and put that in your home directory. You can tweak it to suit your needs if you wish later on. Now open a terminal and switch to root, and edit your network interface config by doing the following:

```
nano /etc/network/interfaces
```

Wi-Fi configuration

05 We want the Pi to automatically connect to your router via Wi-Fi during boot. Edit your **/etc/network/interfaces** file to resemble this:

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
auto wlan0
iface wlan0 inet dhcp
wpa-ssid "ssid"
wpa-psk "password"
```

Replace the "ssid" and "password" with your own details, but keep the quotation marks.

Configure the radio to start at boot

06 In a terminal, as root, navigate to **/etc/init.d/** and then create a file called **radio** using nano.

```
nano radio
```

In that file, type in the following:

```
#! /bin/bash
modprobe snd_bcm2835
amixer cset numid=3 1
python /home/pi/radio.py
```

This loads the kernel module for the sound card Amixer sets the output to the 3.5mm headphone jack (that's what 1 means, HDMI is 2). Lastly it calls the Python script.

Make it executable

07 Save and exit radio in **/etc/init.d** by pressing Ctrl+X and then answer yes to the prompt. Now make radio executable by typing (as root):

```
chmod 755 radio
```

Then, as root, register radio to start on boot by typing in a terminal:

```
update-rc.d radio defaults
```

Raspi-config

08 In a terminal as root, use **raspi-config** to change the boot behaviour of your Pi. We don't want it to load the desktop, a terminal is all we need, as the project will not require a screen for future use. Once complete, reboot the Pi and watch as the output from boot whizzes across the screen.

First test

09 Once the Pi has finished loading, press one of the buttons on your breadboard. In a few seconds you should hear the audio come through the speakers that you attached to the headphone jack. That's it, you have a wireless internet radio. Why not add a mute function using amixer (**manpages.ubuntu.com/manpages/gutsy/man1/amixer.1.html**) and another momentary switch. Or even add an LCD screen (**www.rpiblog.com/2012/11/interfacing-16x2-lcd-with-raspberry-pi.html**) to show the station details.

What you'll need…

A portable hard drive

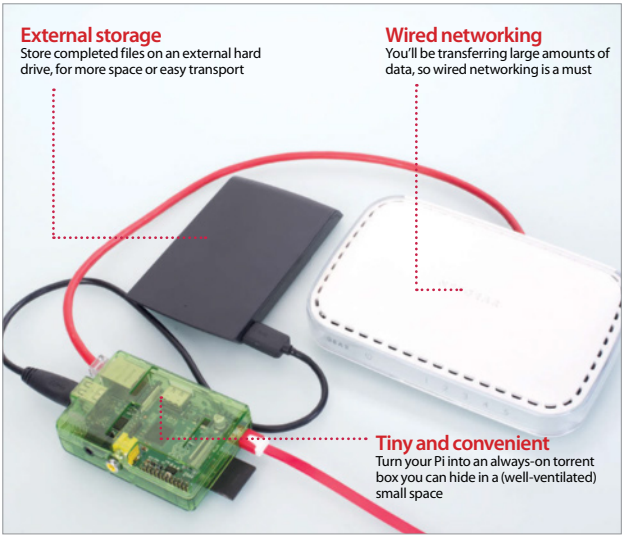
Raspbian
www.raspberrypi.org/downloads

PC with a desktop environment with Deluge
www.deluge-torrent.org

Build an always-on torrent box

Get the latest distros, packages and test builds faster with a low-power, mini torrent box

Torrenting your open source software has a number of advantages – it can be faster, alleviates bandwidth and allows you to share back with the community. Distros, packages and more are available via torrents, and the Raspberry Pi makes for a great tiny, low-wattage, always-on torrent box to better manage your files.



"The Raspberry Pi makes for a great tiny, low-wattage, always-on torrent box to better manage your files"

Install Raspbian

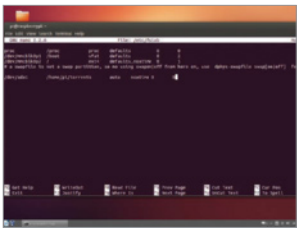
01 Raspbian works just fine for our torrent box. Install the image on an SD card and go through the basic setup process, making sure to enable SSH in the advanced options and to disable the desktop.

Remote access

02 Type ifconfig into your Pi's command line to find the IP address. At this point you can unplug the monitor and set it up remotely, but either way you can now access the Pi by typing:

```
$ ssh [user]@[IP address]
```

…and entering your password to log in.



Mount hard drive

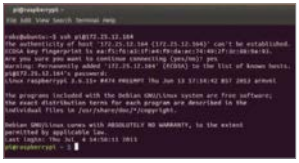
03 Unless you plan to reformat your portable drive, you'll need to install NTFS support onto your Pi. Type in:

```
$ sudo apt-get install ntfs-3g
```

Add the hard drive to **/etc/fstab** (open it with `sudo nano /etc/fstab`) by adding the line:

```
/dev/[hard drive address] [mount point] auto noatime 0 0
```

Use fdisk to find the name of the storage, and create a mount point such as /home/pi/torrents with mkdir. Reboot for it to mount.



Install Deluge

04 We'll use Deluge for our torrents. Install it with:

```
$ sudo apt-get install deluged deluge-console
```

Now start and then stop Deluge so it creates a config file we can edit with:

```
$ deluged
$ sudo pkill deluged
```

And finally, run the following to copy the config file in case we mess up:
$ cp ~/.config/deluge/auth ~/.config/deluge/auth.old

Basic configuration

05 Edit the file with:

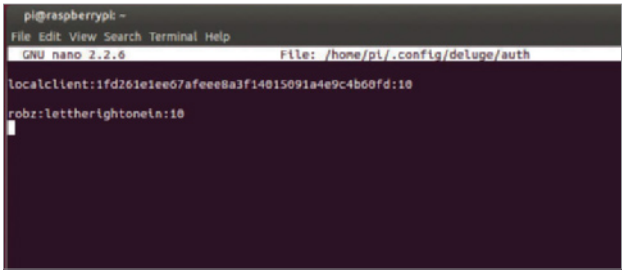
```
$ nano ~/.config/deluge/auth
```

And add to the bottom:

```
[user]:[password]:10
```

…to restrict access.
Now start it up with:

```
$ deluged
$ deluge-console
```



Remote connection

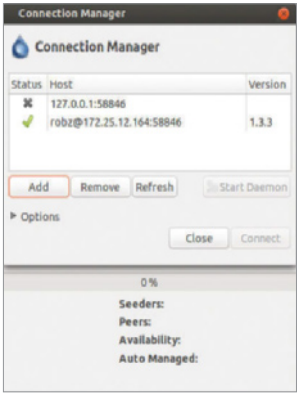
06 Now you're in the client, type the following three commands:

```
config -s allow_remote True
config allow_remote
exit
```

Restart the Deluge daemon with:

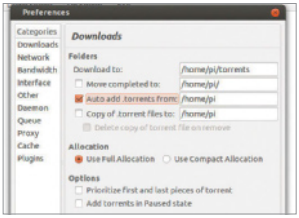
```
$ sudo pkill deluged && deluged
```

Now open the graphical client on your Linux PC.



Remote interface

07 Go to Edit>Preferences> Interface, then disable Classic Mode and restart Deluge. Click Add on the Connection Manager, and enter the IP in Hostname and the user we set up earlier. Click Connect to see any torrents you have downloading or uploading.



Download location

08 Go to Edit again and then Preferences, and change to the Downloads tab if it's not on there already. Set the download location to the directory we mounted the hard drive to, and enable 'Auto add .torrents', setting it to any destination if you plan to dump torrents to the Pi.

Start on boot

09 An init script from Ubuntu can be used to have Deluge start on boot. Download it with:

```
$ sudo wget -O /etc/default/deluge-daemon http://bit.ly/13nKOSj
```

Open **/etc/default/deluge-daemon** with nano and change the username to the one we set up earlier. Save it, then download the full init script and update with:

```
$ sudo wget -O /etc/init.d/deluge-daemon http://bit.ly/13nKKlz
$ sudo chmod 755 /etc/init.d/deluge-daemon
$ sudo update-rc.d deluge-daemon defaults
```

What you'll need…

2 USB Wi-Fi adaptors

Raspberry Pi case

Portable power pack

Portable Wi-Fi signal repeater

Boost your signal using a Raspberry Pi and two USB Wi-Fi dongles

What we like about the Raspberry Pi for this project is that it's very low maintenance and extremely easy to put in an appropriate location. All you need for this project are two USB wireless adaptors, a nice little case and a way to power it. Once you've set it up, you can basically leave it to its own devices, checking in over SSH every now and then to do some updates.

Set this up on a monitor with a keyboard, using a fresh image of Raspbian and with your Wi-Fi dongles plugged in. On first boot keep it as CLI and after doing the usual apt-get updates and upgrades, type startx to get to the desktop. Configure the wireless for wlan0 to connect to your home network and make sure it has a fixed IP address, then reboot to make sure it all works from the command line by using ping **www.google.com**. Now you need to install your first bit of software using:

Below One Wi-Fi adaptor picks up your main Wi-Fi network, while the other one broadcasts a new signal

Above As well as from the main Tor website, you can also download Tor from **bit.ly/1vh9LN5**

```
$ sudo apt-get install hostapd iw
```

After it's installed, you'll want to download and save the file we've created directly to config by entering the following two commands:

```
$ wget http://www.linuxuser.co.uk/wp- content/uploads/2014/08/repeater.zip
$ cp LUDRepeater.conf /etc/hostapd/hostapd.conf
```

Do a reboot and test the configuration file with:

```
$ hostapd -dd /etc/hostapd/hostapd.conf
```

If there are no errors, open the file using nano /etc/hostapd/hostapd.conf and then add the following to the file:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
RUN_DAEMON=yes
```

Also change the SSID to be the same one for your network in the ssid field. Once that's done, you'll need to install the bridge utilities with apt-get install bridge-utils. Then configure it with the following four commands:

```
brctl addbr bridge0
brctl addif bridge0 wlan0
brctl addif bridge0 wlan1
ifconfig bridge0 up
```

Test it out to make sure it works and then place it around the house to extend your wireless signal!

Secure Tor web station

Stay private online by routing all your web traffic through Tor on your Raspberry Pi

In a much more privacy-focused world, being able to browse securely online is an important freedom for many people. With the use of Tor and a few tweaks to the Raspberry Pi, you can make sure all your internet traffic is kept private.

First of all you'll need to make sure to install Tor from the repos. Open up the LXTerminal and simply type:

```
$ sudo apt-get install tor
```

Once that's done, edit the torrc file by using **sudo nano /etc/tor/torrc** and add this to the top of the file:

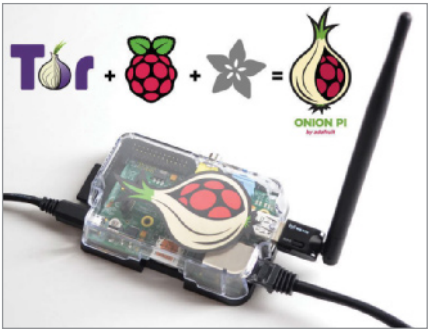
```
VirtualAddrNetworkIPv4 10.192.0.0/10
AutomapHostsOnResolve 1
TransPort 9040
DNSPort 53
```

Save this and then open the next file with **sudo nano /etc/resolv.conf file** and modify it:

```
nameserver 127.0.0.1
```

Finally, you need to change the iptables ruleset, but before you do this, use top to confirm the uid of Tor and make a note of it. Now open up a new file with **nano /etc/init.d/iptables** and enter the code found at: **http://www.linuxuser.co.uk/wp-content/uploads/2014/08/tor.zip**. Now save it and enter:

```
$ chmod 755 /etc/init.d/iptables
$ update-rc.d iptables defaults 12
```



Left Adafruit's Onion Pi Pack contains all the components you will need to set up a secure connection

What you'll need…

A router or switch

A Linux-compatible webcam

Raspbian
www.raspberrypi.org/downloads

Powered USB hub

Did you know…

Most modern webcams are compatible with the Raspberry Pi, but check it supports Linux before you buy.

Build a security camera

Want to keep an eye on something from the comfort of your web browser? All you need is a Pi and a webcam!

This article will teach you how to use the MJPG-streamer software to stream video straight from a webcam to your web browser. You could record the stream, and also display multiple streams from multiple Raspberry Pis on one page. The streams can also be viewed from mobile devices and tablets.

Network investigation

01 The first job we need to do is investigate the network your Raspberry Pi is on, so we can assign it a static IP address. That way, we'll always know where it is on the network so we can access with ease.

```
pi@raspberrypi ~ $ ip addr show dev eth0 | grep inet
    inet 172.17.173.94/24 brd 172.17.173.255 scope global eth0
pi@raspberrypi ~ $ ip route | grep default
default via 172.17.173.1 dev eth0
pi@raspberrypi ~ $ cat /etc/resolv.conf
nameserver 172.17.173.1
```

Assign a static IP address

02 Now that we have the network configuration, we can assign a static IP address. Open **/etc/network/interfaces** in an editor such as nano, and change the line:

```
iface eth0 inet dhcp
```

to a configuration similar to our expert's. Reboot to load the new configuration.

```
iface eth0 inet static
    address 172.17.173.94
    netmask 255.255.255.0
    network 172.17.173.0
    broadcast 172.17.173.255
```

Below It even works from mobile devices and tablets!

Installing the software

03 Log into the Raspbian system with the username 'pi' and the password 'raspberry'. The MJPG-streamer software isn't packaged for the Raspberry Pi, so we'll need to compile it ourselves. Update the package index with `sudo apt-get update`. We need to install Subversion, which we'll use to download source code. We'll also need libjpeg, and imagemagick, both of which are required by MJPG-streamer. Install these with:

```
sudo apt-get install subversion libjpeg8-dev imagemagick
```

Compile MJPG-streamer

04 Download and compile MJPG-streamer as shown below:

```
pi@raspberrypi ~ $ svn checkout svn://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer-code
pi@raspberrypi ~ $ cd mjpg-streamer-code/mjpg-streamer
pi@raspberrypi ~/mjpg-streamer-code/mjpg-streamer $ make clean all
```

Testing it out

05 To start MJPG-streamer:

```
export LD_LIBRARY_PATH=.
./mjpg_streamer -i "input_uvc.so" -o "output_http.so -w ./www"
```

You have to export the library path variable to the current directory (.) so that the various input and output plug-ins can be used. Type your Raspberry Pi's IP address followed by :8080 into a browser and click the Stream tab. If the Stream tab doesn't work, try the example that uses JavaScript, as that should work on most browsers, including Android.

Start MJPG-streamer at boot

06 Edit the **/etc/rc.local** file (you'll need to use sudo) to include the following lines, ensuring that it still ends with exit 0:

```
export STREAMER_PATH=/home/pi/mjpg-streamer-code/mjpg-streamer
export LD_LIBRARY_PATH=$STREAMER_PATH
$STREAMER_PATH/mjpg_streamer -i "input_uvc.so" -o "output_http.so -w $STREAMER_PATH/www" &s
```

Reboot the Pi to check that it comes back up happily and starts the stream.

Recording the stream

07 You can easily download a motion JPEG stream and convert it to a more useful format using VLC:

```
cvlc http://172.17.173.94:8080/?action=stream --sout file/mp4:stream.mp4
```

Alternatively, you can just download the stream URL with wget like so:

```
wget http://172.17.173.94:8080/?action=stream
```

Multiple streams at once

08 Here is an example of how to put multiple streams on a single webpage. These streams could come from a number of different Raspberry Pis all over the house.

```
<html>
<h1>Streaming Example</h2>
<table border="1">
<tr>
<td><h2>Stream 1</h2></td>
<td><h2>Stream 2</h2></td>
</tr>
<tr>
<td><img src="http://172.17.173.94:8080/?action=stream"/></td>
<td><img src="http://172.17.173.94:8080/?action=stream"/></td>
</tr>
</table>
</html>
```

Extensions and improvements

09 This example setup is very basic and has a few flaws. There is no authentication or SSL encryption, which means the streams are insecure and shouldn't be shared over the internet. The HTTP module that comes with MJPG-streamer is quite simple, so you could get SSL by using a reverse proxy, such as Pound. You'd be able to do SSL for each Pi you had with a single Pound instance and access the different Pis by having something like 'stream1, stream2' in the URL.

"Use the MJPG-streamer software to stream video straight from a webcam to your web browser"

What you'll need…

A toy RC car with two channels (steering and drive)

Adafruit PWM I2C servo driver

Female-to-female jumper cables

5V battery power bank

Estimated cost: £60 / $100

Components from
www.modmypi.com

Build a Raspberry Pi-powered car

Make use of cutting-edge web technologies to take control of a remote controlled car with a smartphone or tablet…

Did you know…

You can make this project work with just about any remote controlled car. Follow the guide for more details.

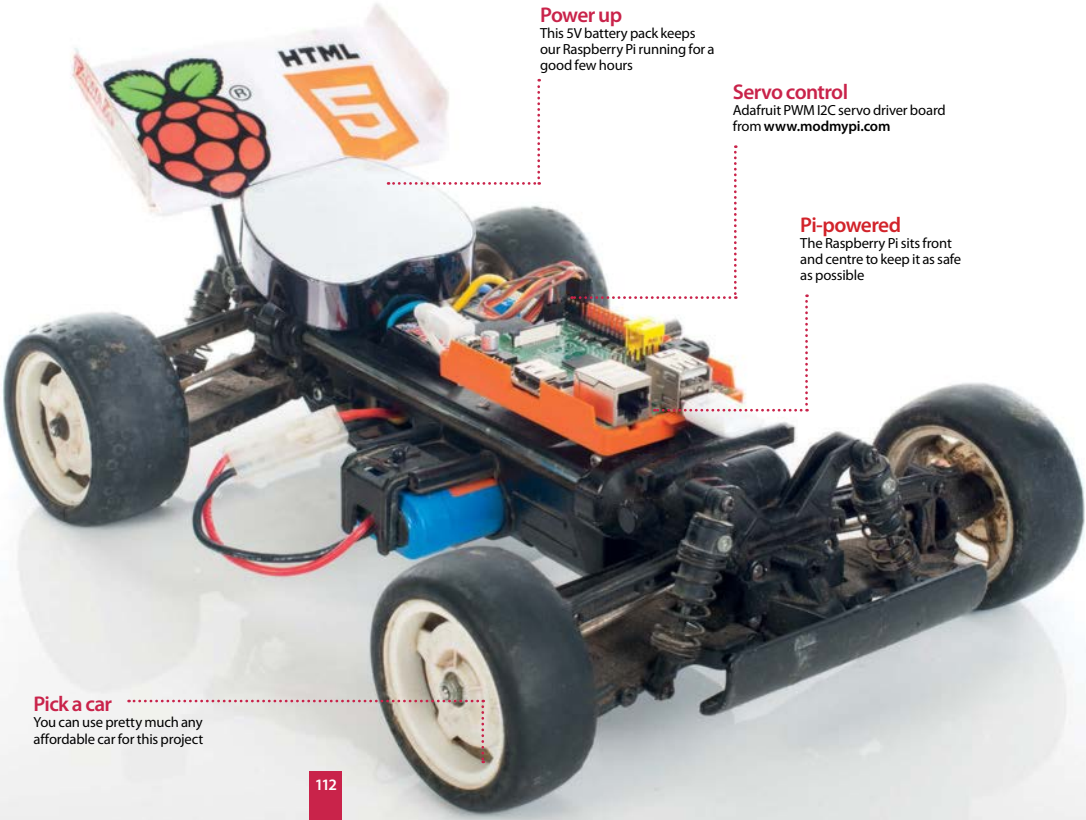
Web technologies are moving forward at a huge pace, cloud technologies are bringing mass computing to individuals, and hardware has reached a perfect moment in time where sensors, displays and wireless technology have all evolved into efficient and affordable devices. We truly are at a point where nearly anyone can take an idea from nothing to a working product in a week and at very little cost. Just like this project, which is fun, quick and easy to build on and a fantastic way to learn. We're going to grab an old remote-control car, rip off its radio receiver and replace it with the Raspberry Pi, hook it up on the network, fire up a bleeding-edge web server and then get your smartphone or tablet to control it by tilting the device. By the end of this, not only will you have a fun toy, you will have learnt about the basic technologies that are starting to power the world's newest and biggest economy for the foreseeable future.

Raspberry Pi-controlled car build process

To help our toy car come to life using the latest web technologies and our credit card-sized computer, we're going to need to make some pretty significant changes to its workings. Fortunately, the most complex aspects of the build can be accomplished with a couple of affordable purchases, namely a servo controller board to take care of the steering and throttle, and a 5V battery pack to keep the Raspberry Pi running smoothly.

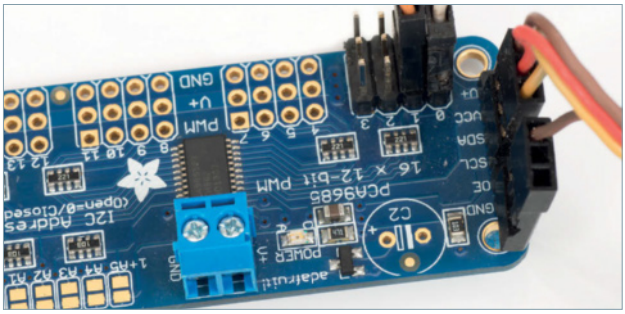
Identify and remove old radio

01 This project is effectively replacing the car's normal transmitter and receiver. Notice the three sockets on the original receiver: one goes to the motor controller and one to the steering servo. Some remote-control cars also have separate battery for the electronics, but those (especially with an electronic speed controller with BEC) get their 5V power supply directly from the speed controller, saving on components. If you don't have a speed controller with 5V BEC, you'll need to get a 5V supply elsewhere. Many shops sell 5V battery power supplies – often as mobile phone emergency top-ups. **www.modmypi.com** sells a suitable 5V battery power bank for under £20 and you should get a couple of hours of use from your Raspberry Pi.



Attach the servo cables to the new controller

02 We soldered our 16-channel I2C servo controller board from **www.modmypi.com** as per its instructions and simply plugged channel 0 (steering) and channel 1 (motor) headers onto it. There are six cables in total: the bottom two are ground, the middle two are the power and the top two are the PWM (pulse-width modulation) signals. This is a good time to think of places to mount the extra components and the best fixing method seems to be sticky-back Velcro.



Connect the I2C bus to the Raspberry Pi

03 We're using the Raspberry Pi's I2C bus to control the servo interface board, which only needs four cables – they all go between the Raspberry Pi and the servo controller board as pictured. Visit **http://bit.ly/N3nq4J** for a tutorial on how to set up I2C on the Raspberry Pi.

From top to bottom we need to use the 1. GND, 2. SCL, 3. SDA and 4. VCC, which map directly to the same ports on the Raspberry Pi. Essentially this is power, ground and two communication channels.

Hook it up to the Raspberry Pi

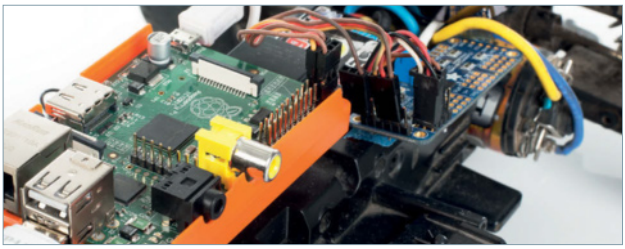
04 On a Rev 1 Raspberry Pi, the cables look the same. Though the Rev boards have different labelling, the physical pins are in the same place. Bottom left (closest to the RasPi power connector) is the 3.3V power; next to that is the SDA header, which is the data channel. Next to that in the bottom right is the SCL channel, which controls the clock of the I2C devices. And finally – on the top-right port – is the Ground. We recommend printing a labelled image of the GPIO pins.

Overview of the main components

05 You should now have the servo board in the middle with the steering servo and speed controller on one side and the Raspberry Pi on the other. The motor is connected to the other end of the speed controller (that end should have much thicker wires); the speed controller also has two thick wires going to the main car's battery – in this case a 7.2V NiCad. We now have two very separate power systems with the high current motors on one side and the low current electronics on the other. Let's make sure it stays that way!

Find everything a home

06 We can now put it together. Use plenty of sticky-back Velcro, tie wraps or elastic bands to keep everything secure and find spaces in the car's body to hide the wires where possible. While it is possible to stick or screw the Raspberry Pi directly to the car, we recommend to use at least the bottom half of a case for added protection and ease of access. Insert your SD card, network cable or Wi-Fi dongle and power supply. Sit back and admire your hacking skills!



What you'll need…

A RasPi car, ready to go

An internet connection

A reasonably modern smartphone/tablet

Pi car source code
github.com/shaunuk/picar

Control your Raspberry Pi-powered car

Control a toy car with a smartphone and the latest web technologies

Now we have our fantastic Raspberry Pi-powered car all wired and charged, it's time to make it come alive. We're using the best web technologies that the JavaScript programming language offers, to harness the natural movement of your hand and wirelessly drive the vehicle. Each little movement will trigger an event that calculates what the car should do and then sends it over a socket connection.

Did you know…

Our code will send instructions to our car over twenty times per second. It will be very responsive to drive!

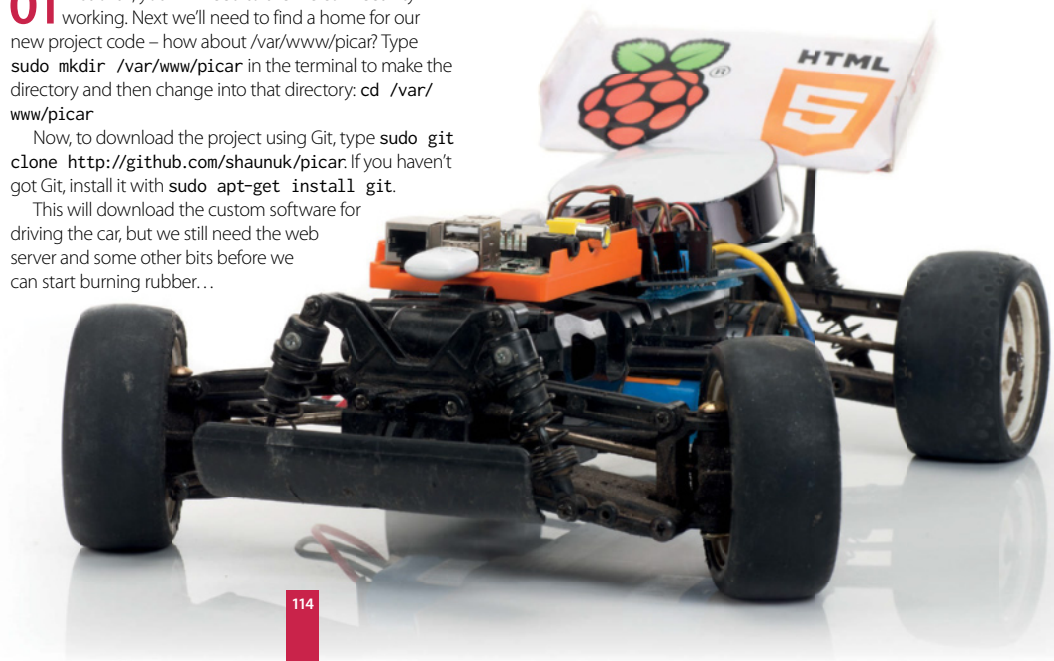
Download and install the software

01 First of all, you will need to the I2C connectivity working. Next we'll need to find a home for our new project code – how about /var/www/picar? Type `sudo mkdir /var/www/picar` in the terminal to make the directory and then change into that directory: `cd /var/www/picar`

Now, to download the project using Git, type `sudo git clone http://github.com/shaunuk/picar`. If you haven't got Git, install it with `sudo apt-get install git`.

This will download the custom software for driving the car, but we still need the web server and some other bits before we can start burning rubber…

Below All you need to finish off your project is access to a smartphone or tablet



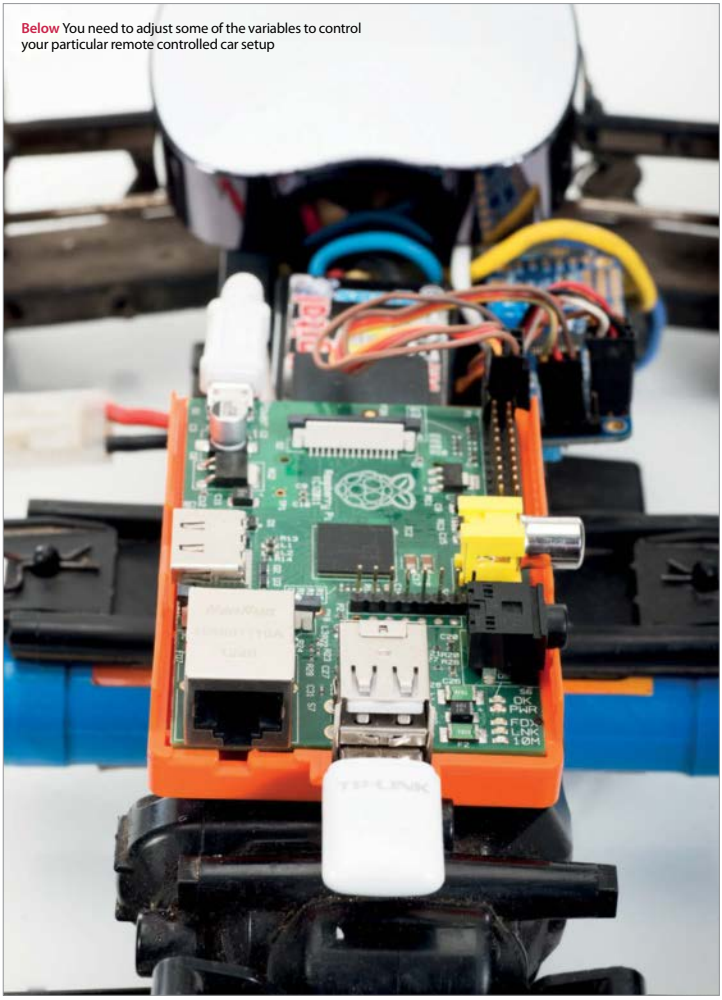
Download and install Node.js

02 Next we're using Node.js and its package tool, the Node package manager (npm). Type `sudo wget http://nodejs.org/dist/v0.10.21/node-v0.10.21-linux-arm-pi.tar.gz`. This will download a fairly recent version of Node.js – the version Raspbian has in its repositories is way too old and just doesn't work with the new technologies we're about to use. Extract the node package by typing:

```
sudo tar -xvzf node-v0.10.21-linux-arm-pi.tar.gz
```

Configure Node.js

03 To make it easy to run from everywhere, we will create symbolic links for Node and npm binaries. Type `sudo ln -s /var/www/node-v0.10.21-linux-arm-pi/bin/node /bin/node` and then `sudo ln -s /var/www/node-v0.10.21-linux-arm-pi/bin/npm /bin/npm`. Then, to get the extra modules, type `npm install socket.io node-static socket.io adafruit-i2c-pwm-driver sleep optimist`



Below You need to adjust some of the variables to control your particular remote controlled car setup

Get to know the project

04 Now we have everything, you should see three files: the server (app.js), the client (socket.html) and the jQuery JavaScript library for the client. The server not only drives the servos, but it is a web server and sends the socket.html file and jQuery to the browser when requested – it's a really neat and simple setup and just right for what we're trying to achieve.

Test the servos

05 Our handy little program (app.js) has a special mode just for testing. We use two keywords here: beta for servo 0 (steering) and gamma for servo 1 (motor control). Type node app.js beta=300. You should see the front wheels turn. Now the numbers need experimenting with. On our example, 340 was left, 400 was centre and 470 was right. Do the same for the motor by typing node app.js gamma=400 and take note of the various limits of your car.

"We're using the best web technologies that the JavaScript programming language has to offer"

Configure sensible defaults

06 Now you know what your car is capable of, we can set the defaults in app.js and socket.html. Edit app.js and find the section that says 'function emergencyStop'. Adjust the two numbers to your car's rest values. Then open socket.html and adjust the predefined values under 'Define your variables here'.

Going for a spin

07 We're almost ready to try it out, but you need to know the IP address of your Pi car, so type ifconfig at the terminal. Then fire up the app by typing node app.js. Now grab the nearest smartphone or tablet, making sure it's on the same network as your Pi. Open the web browser and go to **http://[your IP address]:8080/socket.html**. You should get an alert message saying 'ready' and as soon as you hit OK, the gyro data from your phone will be sent to the car and you're off!

Full code listing

socket.html

```
<html>
<head>
<script src="jquery-2.0.3.min.js"
language="javascript"></script>
<script src="/socket.io/socket.io.js"></
script>
<meta name="viewport" content="user-
scalable=no, initial-scale=1.0, maximum-
scale=1.0;" />
<script>
//------ Define your variables here
var socket = io.connect(window.location.
hostname+':8080');
var centerbeta = 400; //where's the middle?
var minbeta = '340'; //right limit
var maxbeta = '470'; //left limit
var multbeta = 3; //factor to multiply the
// raw gyro figure
var centergamma = 330;
var ajustmentgamma = 70; //what do we do to
the angle to get to 0?
var mingamma = 250; //backwards limit
var maxgamma = 400; //forward limit
var multgamma = 1; //factor to multiply the
//raw gyro figure by to get the desired
//rate of acceleration
window.lastbeta='0';
window.lastgamma='0';
$(function(){
  window.gyro = 'ready';
  alert('Ready -- Lets race !');
});
window.ondeviceorientation = function(event)
{
  beta = centerbeta+(Math.round(
event.beta*-1)*multbeta);
  if (beta >= maxbeta) {
    beta=maxbeta;
  }
  if (beta <= minbeta) {
    beta=minbeta;
```

```
  }
gamma = event.gamma;
  gamma = ((Math.round(event. ↵
gamma)+ajustmentgamma)* multgamma)+ ↵
centergamma;
//stop sending the same command more than
once
send = 'N';
if (window.lastbeta != beta) { send = 'Y' }
if (window.lastgamma != gamma) { send = 'Y'
}
window.lastbeta=beta;
window.lastgamma=gamma;
if (window.gyro == 'ready' && send=='Y')  {
//don't send another command until ready...
  window.gyro = 'notready';
    socket.emit('fromclient', { beta: beta
↵ gamma: gamma } );
    window.gyro = 'ready'; }}
```

app.js

```
//declare required modules
var app = require('http').
createServer(handler)
  , io = require('socket.io').listen(app)
  , fs = require('fs')
  , static = require('node-static')
  , sys = require('sys')
  , PwmDriver = require('adafruit-i2c-pwm-
driver')
  , sleep = require('sleep')
  , argv = require('optimist').argv;
  app.listen(8080);
//set the address and device name of the ↵
breakout board
pwm = new PwmDriver(0x40,'/dev/i2c-0');
//set pulse widths
setServoPulse = function(channel, pulse) {
  var pulseLength;
  pulseLength = 1000000;
  pulseLength /= 60;
  print("%d us per period" % pulseLength);
  pulseLength /= 4096;
  print("%d us per bit" % pulseLength);
  pulse *= 1000;
  pulse /= pulseLength;
  return pwm.setPWM(channel, 0, pulse);
};
//set pulse frequency
pwm.setPWMFreq(60);
//Make a web server on port 8080
var file = new(static.Server)();
function handler(request, response) {
  console.log('serving file',request.url)
  file.serve(request, response);
};

console.log('Pi Car we server listening ↵
on port 8080 visit http://ipaddress:8080/ ↵
socket.html');

lastAction = "";
function emergencyStop(){
 //center front wheels
  pwm.setPWM(0, 0, 400);
 //stop motor
  pwm.setPWM(1, 0, 330);
  console.log('###EMERGENCY STOP - signal ↵
lost or shutting down');
}

if (argv.beta) {
  console.log("\nPerforming one off servo ↵
position move to: "+argv.beta);
  pwm.setPWM(0, 0, argv.beta);
//using direct i2c pwm module
  pwm.stop();
  return process.exit();
}
if (argv.gamma) {
  console.log("\nPerforming one off servo ↵
position move to: "+argv.gamma);
  pwm.setPWM(1, 0, argv.gamma); //using
direct i2c pwm module
  pwm.stop();
  return process.exit();
}
//fire up a web socket server
io.sockets.on('connection', function (socket)
{
  socket.on('fromclient', function (data) {
  console.log("Beta: "+data.beta+" Gamma:
"+data.gamma);
  //exec("echo 'sa "+data+"' > /dev/
// ttyAMA0", puts);
//using http://electronics.chroma.se/rpisb.php
  //exec("picar.py 0 "+data.beta, puts);
 //using python adafruit module
  pwm.setPWM(0, 0, data.beta);
//using direct i2c pwm module
  pwm.setPWM(1, 0, data.gamma);
//using direct i2c pwm module
  clearInterval(lastAction);
//stop emergency stop timer
  lastAction = setInterval(emergencySt ↵
op,1000);
//set emergency stop timer
  });
});
process.on('SIGINT', function() {
  emergencyStop();
  console.log("\nGracefully shutting down
from SIGINT ↵ (Ctrl-C)");
  pwm.stop();
  return process.exit();
});
```

What you'll need…

Raspberry Pi 2

Latest Raspbian image
raspberrypi.org/downloads

Minecraft Pi Edition
pi.minecraft.net

Raspberry Pi case

USB game controller (PS3 preferable)

Build a Minecraft console for Pi

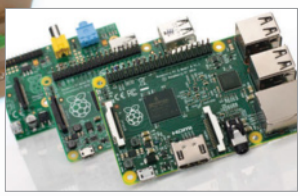
Create a fully functional, Pi-powered games console that you can play Minecraft on and learn how to program too

Minecraft means many things to many people, and to Raspberry Pi users it's supposed to mean education. Not everyone knows, though, that you can still have fun and play Minecraft as you normally would.

Using Raspberry Pi, it is also the cheapest way to get a fully functional version of Minecraft up onto your TV. However, in its normal state, just being on a TV isn't the end of it. Using all the features and functions of the Pi, we can take it to a state more fitting of a TV by making it into a hackable, moddable Minecraft console. In this tutorial, we will show you how to set it up in terms of both software and hardware, how to add a game controller to make it a bit better for TV use, and we'll even give you some example code on how to mod it. Now, it's time to get building, so head to Step 1.

Choose your Raspberry Pi

01 Before we start anything, everything we plan to do in this tutorial will work on all Raspberry Pi Model Bs with at least 512 MB of RAM. However, Minecraft: Pi Edition can chug a little on the original Model Bs, so we suggest getting a Raspberry Pi 2 to get the most out of this tutorial.



Prepare your Raspberry Pi

02 Minecraft: Pi Edition currently works on Raspbian. We recommend you install a fresh version of Raspbian, but if you already have an SD card with it on, the very least you should do is:

```
sudo apt-get update && sudo apt-get upgrade
```

Prepare Minecraft

03 If you've installed Raspbian from scratch, Minecraft is actually already installed – go to the Menu and look under Games to find it there ready. If you've just updated your version of Raspbian, you can install it from the repos with:

```
$ sudo apt-get install minecraft-pi
```

Test it out

04 If you've had to install Minecraft, it's best just to check that it works first. Launch the desktop, if you're not already in it, with startx and start Minecraft from the Menu. Minecraft: Pi Edition is quite limited in what it lets you do, but it does make room for modding.

X setup

05 If you have a fresh Raspbian install and/or you have your install launch into the command line, you need to set it to load into the desktop. If you're still in the desktop, open up the terminal and type in raspi-config. Go to Enable Boot to Desktop and choose Desktop.

> "If you've installed Raspbian from scratch, Minecraft is actually already installed – go to the Menu and look under Games to find it"

Set up Python

06 While we're doing set up bits, we might as well modify Mincecraft using Python for a later part of the tutorial. Open up the terminal and use:

```
$ cp /opt/minecraft-pi/api/python/mcpi ~/minecraft/
```

Minecraft at startup

07 For this to work as a console, we'll need it to launch into Minecraft when it turns on. We can make it autostart by going into the terminal and opening the autostart options by typing:

```
$ sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

Autostart language

08 In here, you just need to add @minecraft-pi on the bottom line, save it and reboot to make sure it works. This is a good thing to know if you also want other programs to launch as part of the boot-up process.

Turn off

09 For now, we can use the mouse and keyboard to shut down the Pi in the normal way, but in the future you'll have to start turning it off by physically removing power. As long as you've exited the Minecraft world and saved, that should be fine.

"Getting power to the Raspberry Pi 2 so that it runs properly can be tricky if you're using a USB port"

The correct case

10 In this scenario, we're hooking this Raspberry Pi up to a TV, which means it needs a case so that there's less chance of damage to the components from dust or static. There are many good cases you can get – we are using the Pimoroni Pibow here as you can mount it to the back of the TV.

Find a power supply

11 Getting power to the Raspberry Pi 2 so that it runs properly can be tricky if you're using a USB port or a mobile phone charger – the former will be underpowered and the latter is not always powerful enough. Make sure you get a 2A supply, like the official Raspberry Pi one.

Go wireless

12 We understand that not everyone has an ethernet cable near their TV, so it may be a good idea to invest in a Wi-Fi adapter instead. There is a great list of compatible Wi-Fi adapters on the eLinux wiki: **elinux.org/RPi_VerifiedPeripherals.**

Mouse and keyboard

13 Now that we have the Raspberry Pi ready to be hooked up, you should look at your controller situation – do you want to be limited by the wires or should you get a wireless solution instead? We will cover controller solutions over the page, but it's worth considering now as it will be easier to configure now than it will be at a later date.

Get ready for SSH

14 It will be easier to create and apply scripts to Minecraft by uploading them via the network rather than doing it straight on the Pi. In the terminal, find out what the IP address is by using ifconfig, and then you can access the Pi in the terminal of another networked computer using the following:

```
ssh pi@[IP address]
```

Have a play

15 We have built a fully functional Minecraft console. You could start playing if you so wish and you don't need to add a controller. You can flip over to page 122 to begin learning how to mod your Minecraft. However, if you do want to add controller support then carry on and take a look at Step 16.

Add controller support

16 Make sure the controller input functions are installed

Controls

Here's the full layout of the buttons used by the PS3 controller by default – you can change them in the script that you download in Step 18

on the Raspberry Pi. To do this, ssh into the Raspberry Pi like we did in Step 14 (where 'raspberry' is the password) and then install the following package:

```
$ sudo apt-get install
xserver-xorg-input-joystick
```

Controller mapping

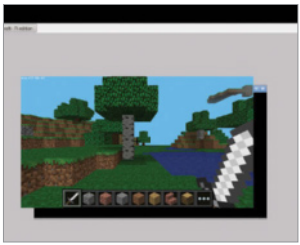
17 We have a controller map for the PS3 controller that you can download straight to your Pi, and with a bit of tweaking can fit most USB controllers as well. Go to the controller configuration folder with:

```
$ cd /usr/share/X11/xorg.
conf.d/
```

Replace the controller mapping

18 We'll remove the current joystick controls by using sudo rm 50-joystick.conf and then replace by downloading a custom configuration using:

```
$ sudo wget http://www.
linuxuser.co.uk/wp-content/
uploads/2015/04/50-joystick.conf
```



Reboot to use

19 After a reboot to make sure everything's working, you should be able to control the mouse input on the console. R2 and L2 are the normal mouse clicks and can be used to navigate the Minecraft menu to access the game.

Go full-screen

20 So far you may have noticed that Minecraft is running in a window – you can click the full-screen button to make it fill the screen, however you then heavily limit your mouse control. Thanks to the controller, you can get around that.

As soon as you load the game, make sure you use the sticks for movement and the d-pad for selecting items in the inventory.

Xbox controllers

Unfortunately, Xbox 360 controllers work slightly differently with Linux. As they use their own drivers that are separate to the normal joystick drivers we used for the PS3 pad and other USB controllers, a 360 controller doesn't work as a mouse and is harder to assign specific functions to. This makes it tricky to use in a situation such as this.

Mod your Minecraft

Here is some example code, and explanations for it, so that you can learn how to program in Python and mod Minecraft Pi

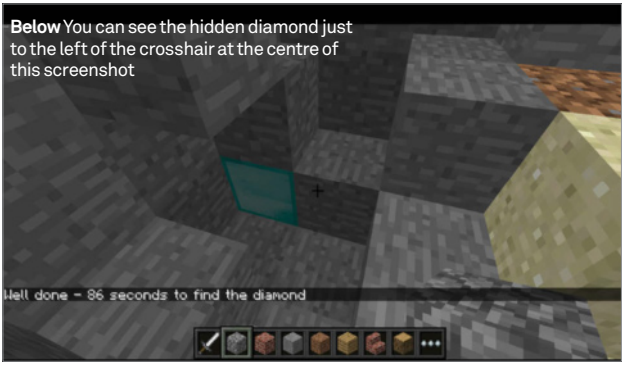
We program Minecraft to react in python using the API that comes with Minecraft: Pi Edition – it's what we moved to the home folder earlier on. Now's a good time to test it – we can do this remotely via SSH. Just cd into the Minecraft folder in the home directory we made, and use nano test.py to create our test file. Add the following:

```
from mcpi.minecraft import Minecraft
from mcpi import block
from mcpi.vec3 import Vec3
mc = Minecraft.create()
mc.postToChat("Hello, Minecraft!")
```

Save it, and then run it with:

```
$ python test.py
```

"Hello, Minecraft!" should pop up on-screen. The code imports the Minecraft function from the files we moved earlier, which allows us to actually use Python to interact with Minecraft, along with the various other functions and modules imported. We then create the mc instance that will allow us to actually post to Minecraft using the postToChat function. There are many ways you can interact with Minecraft in this way – placing blocks that follow the player, creating entire structures and giving them random properties as they're spawned as well. There are very few limits to what you can do with the Python code, and you can check out more projects here: **https://mcpipy.wordpress.com**.



Below You can see the hidden diamond just to the left of the crosshair at the centre of this screenshot

Over the page, we have a full listing for a hide and seek game that expands on the kind of code we're using here, where the player must find a diamond hidden in the level, with the game telling you whether you're hotter or colder. You can write it out from scratch or download it to your Pi using the following commands:

```
$ wget http://www.linuxuser.co.uk/tutorialfiles/Issue134/ProgramMinecraftPi.zip
$ unzip ProgramMinecraftPi.zip
$ cp Program\ MinecraftPi/hide_and_Seek.py ~/minecraft
```

Check out the annotations to the right to see how it works.

"We program Minecraft to react in Python using the API that comes with Minecraft Pi – it's what we moved to the home folder earlier"

Full code listing

Import
Here we're importing the necessary modules and APIs to program Minecraft

Locate
We find the player's position and round it up to an integer

Range finding
Calculate the distance from player to diamond. This compares the co-ordinates

Creation
Create a random position for the diamond within 50 blocks of the player position

Start
This starts the game. It asks to get the position of the player to start each loop

Notification
This pushes a message through to let the player know that the game has begun

Checking
We start timing the player with timeStarted, and set the last distance between the player and the block

Message writing
If you're two or more blocks away from the diamond, it will tell you whether you're nearer or farther away than your last position check

Success
If the loop has been broken, it tallies up your time and lets you know how long it was before you found the diamond. Finally, the last bit then tells Python to start the script at the main function


```
from mcpi.minecraft import Minecraft
from mcpi import block
from mcpi.vec3 import Vec3
from time import sleep, time
import random, math

mc = Minecraft.create()
playerPos = mc.player.getPos()

def roundVec3(vec3):
  return Vec3(int(vec3.x), int(vec3.y), int(vec3.z))

def distanceBetweenPoints(point1, point2):
  xd = point2.x - point1.x
  yd = point2.y - point1.y
  zd = point2.z - point1.z
  return math.sqrt((xd*xd) + (yd*yd) + (zd*zd))

def random_block():
  randomBlockPos = roundVec3(playerPos)
  randomBlockPos.x = random.randrange(randomBlockPos.x - 50, randomBlockPos.x + 50)
  randomBlockPos.y = random.randrange(randomBlockPos.y - 5, randomBlockPos.y + 5)
  randomBlockPos.z = random.randrange(randomBlockPos.z - 50, randomBlockPos.z + 50)
  return randomBlockPos

def main():
  global lastPlayerPos, playerPos
  seeking = True
  lastPlayerPos = playerPos

  randomBlockPos = random_block()
  mc.setBlock(randomBlockPos, block.DIAMOND_BLOCK)
  mc.postToChat("A diamond has been hidden - go find!")

  lastDistanceFromBlock = distanceBetweenPoints(randomBlockPos, lastPlayerPos)
  timeStarted = time()
  while seeking:

    playerPos = mc.player.getPos()

    if lastPlayerPos != playerPos:
      distanceFromBlock = distanceBetweenPoints(randomBlockPos, playerPos)
      if distanceFromBlock < 2:
        seeking = False

      else:
        if distanceFromBlock < lastDistanceFromBlock:
          mc.postToChat("Warmer " + str(int(distanceFromBlock)) + " blocks away")
        if distanceFromBlock > lastDistanceFromBlock:
          mc.postToChat("Colder " + str(int(distanceFromBlock)) + " blocks away")

      lastDistanceFromBlock = distanceFromBlock

    sleep(2)

  timeTaken = time() - timeStarted
  mc.postToChat("Well done - " + str(int(timeTaken)) + " seconds to find the diamond")

if __name__ == "__main__":
  main()
```

What you'll need…

Raspberry Pi 2
4 x 6mm tactile buttons
Female-to-female jerky
Terminal blocks
A glove
Router
2x CAT5 cables

Build a power move glove

Create a piece of wearable tech with power moves assigned to each button to enhance your Minecraft game

Many of you will be avid fans of Minecraft. In schools it is becoming a motivational teaching and learning tool, useful in areas such as programming, creating logic gates and setting up a network.

This project is framed around creating a simple networked Minecraft game where one player chases the other and tries to hit the block they are standing on. The real hack is programming a 'power glove' that enables you to assign power moves to each button. These powers can then be deployed and used to slow down the other player and get yourself out of sticky situations – check out the video at **bit.ly/1CQSmHS**! The real beauty of this hack is that you can then create and customise your own power moves. The possibilities are endless, limited only by your imagination. If you're confident with GPIO pins and setting up buttons, jump straight to Step 8.

Update the Raspberry Pi

01 This project is designed for the Raspberry Pi 2 which requires the updated operating system, although it is compatible with the Raspberry Pi B+ too. First ensure that the software is up to date – open the LX Terminal type:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install raspberrypi-ui-mods
```

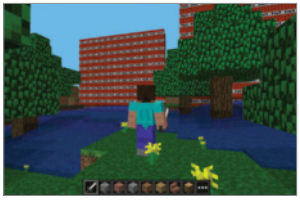
Connect a button to Pi

02 Take one of the buttons and connect a wire to each contact, then take the other two ends and connect to the Pi. You may find this easier using a female-to-female connector. To set up a test button, use GPIO pin 17 – this is physical pin number 11 on the board. The other wire connects to a ground pin, shown by a minus sign (the pin above GPIO pin 17 is a ground pin).

Test that the button works

03 Use this test code to ensure the button is functioning correctly. Once it is, the same setup method can be used throughout this project. To ensure the buttons are responsive, use the pull-up resistor with the code GPIO.PUD_UP – this will ensure that multiple touches aren't registered on each button. Using Python 2.8, type in the code below, then save and run. If working correctly, it will return the message 'Button works'.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setup(17, GPIO.IN, GPIO.PUD_UP)
while True:
 if GPIO.input(17) == 0:
     print "Button works"
```



Create a power move

04 Now to create your first power move. The glove and code are structured in such a way that once the basic code template is set up, you can create your own power moves and assign them to each button, keeping both your ideas and your gameplay fresh. The first power move you will program enables you to place a wall of TNT between you and the player trying to catch you. They have a choice to go around the wall or blow it up, but it will slow them down. In a new Python window, type the code below and save the program into the home folder:

```
import time
def Firewall():
    mc.postToChat("Firewall
```

PiGlovePowerMoves.py

```
import time
from mcpi import minecraft

mc = minecraft.Minecraft.
create("192.168.1.211")
#Replace with the other
players IP address

import RPi.GPIO as GPIO

#Set up the GPIO Pins
GPIO.setmode(GPIO.BCM)

#Sets the pin to high
GPIO.cleanup()
GPIO.setup(17, GPIO.IN, GPIO.
PUD_UP)
#11 on the BOARD GREEN
Firewall
GPIO.setup(18, GPIO.IN, GPIO.
PUD_UP)
#12 on the BOARD BROWN Lava
GPIO.setup(9, GPIO.IN, GPIO.
PUD_UP)
#21 on the BOARD BLUE Mega
Jump
GPIO.setup(4, GPIO.IN, GPIO.
PUD_UP)
#7 on the BOARD ORANGE Puddle
GPIO.setwarnings(False)
#switch off other ports

#Builds a wall of TNT which if
the player hits will explode
def Firewall():
    mc.postToChat("Firewall
Placed")
    TNT = 46,1
    x, y, z = mc.player.
getPos()
    mc.setBlocks(x-6, y, z+2,
x+6, y+10, z+3, TNT)
    time.sleep(1)

#Lays Lava to slow down the
other player
def Lay_Lava():
    Lava = 10
    check = 1
    mc.postToChat("Lava
Deployed")
    while check == 1:
        time.sleep(0.2  )
        x, y, z = mc.player.
getPos()
        mc.setBlock(x-1, y,
z, Lava)
        check = 0

#Peforms a Mega Jump to lose
players
def Mega_Jump():
    time.sleep(0.1)
    mc.postToChat("Mega-Jump")
    x, y, z = mc.player.
getPos()
    mc.player.setPos(x, y+15,
z)
    time.sleep(1)

#Creates a Puddle to slow down
your player
def Mega_Water_Puddle():
    mc.postToChat("Mega Water
Puddle")
    time.sleep(0.2)
    WATER = 9
    x, y, z = mc.player.
getPos()
    mc.setBlocks(x-5, y, z-4,
x-1, y, z+4, WATER)
    time.sleep(1)
```

```
Placed")
    TNT = 46,1
    x, y, z = mc.player.
getPos()
    mc.setBlocks(x-6, y, z+2,
x+6, y+10, z+3, TNT)
  time.sleep(10)

while True:
    Firewall()
```

Open Minecraft

05 The updated version of the Raspbian OS comes with Minecraft pre-installed, it can be found under Menu>Games – so load it up. If you have used the Minecraft: Pi Edition before you will be aware that it runs better in a smaller-sized window, so don't make it full screen. You may prefer to adjust and arrange each window side-by-side to enable you to view both the Python code and the Minecraft game at the same time.

Engage the firewall

06 Start a new Minecraft game, then go to the Python program and press F5 to run it. You may need to press the Tab key to release the mouse from the Minecraft window. The program will place a 12 x 10 wall of TNT behind the player every ten seconds – you can blow them up but the Pi might lag.

Assign the buttons

07 Now you have a working button and a power move, you can combine these two together. Basically, when you press the button the power move will deploy. Once this is working you can then set up the other three buttons

with the same method. Open a new Python window and type in the code below – save the file in the home folder. Start a Minecraft game as shown in Step 6 and then run the new program. Every time you press the button you will place a TNT firewall.

```
import time
from mcpi import minecraft
mc = minecraft.Minecraft.
create()
  import RPi.GPIO as GPIO
  GPIO.setmode(GPIO.BCM)
  GPIO.setup(17, GPIO.IN, GPIO.
PUD_UP)
#11 on the BOARD
def Firewall():
  mc.postToChat("Firewall
Placed")
    TNT = 46,1
    x, y, z = mc.player.
getPos()
  mc.setBlocks(x-6, y, z-2,
x+6, y+10, z-1, TNT)
while True:
        if GPIO.input(17) == 0:
            Firewall()
```

Set up further buttons

08 For the other buttons, use the same method from Step 2. Take the button and connect the two wires to either side. You can add all three buttons and test them for connectivity. Connect each set to the GPIO 17 and run the firewall program. If the firewall power move builds on each click of the button, the connections and buttons are working and you're ready to attach the buttons.

Create the glove

09 Attach the four buttons to the fingers. There are a number of ways to do this: glue the button on, sew them in, stick them with double-sided tape – the choice is up to you. Wires can be hidden or on display, depending on your preferences and how you want the glove to look.

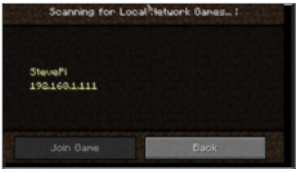
Connect the wires

10 Now you are ready to connect the wires to the Pi to enable all four power moves to be used. Take one end of each wire and connect them to the respective pins, as shown below. The other end of the wire is placed into a ground pin. Note that the RPi.GPIO pin numbering system is used here rather than the physical pin number on the board – for example, GPIO pin 17 is physical pin number 11. The following pins are used:

```
GPIO 17, pin 11 on the board
GPIO 18, pin 12 on the board
GPIO 9, pin 21 on the board
GPIO 4, pin 7 on the board
```

Set the network up

11 To interact with other players in the Minecraft world you will need to set up a networked multiplayer game. This is simple and can be achieved using an old router. Connect each Raspberry Pi via an Ethernet cable to the router and then back to the Ethernet port on each Raspberry Pi. Turn on the router and wait about 30 seconds for the IP addresses to be assigned. Now load up Minecraft and one of the players will start a new game.



Run the game!

12 After the game has loaded, the other connected player will see an option to 'connect to a multiplayer game', usually called Steve. The player selects this option and the networked game will begin. You will be able to see two 'Steves' and the Minecraft chat will inform you that 'Steve has

joined the game'. Open Python and the glove code, press F5 to run and you will be able to see the power moves being deployed.

Interact in other worlds

13 Under the current setup that if the Pi Glove is connected to the game and you use one of your power moves, it will only appear in your Minecraft world and not in the other players. This isn't great, as the power move is supposed to stop the other player! To resolve this, find the IP address of the other Pi, then enter this IP address into this line of code: mc = minecraft.Minecraft.create( ), filling the empty brackets with the IP address of the other Pi within your game. Remember that this is the IP address of the other player's Pi and not your IP address.

Find your IP addresses

14 To find the IP address of a Pi, load the LX terminal, type ipconfig and press Enter – the address will be displayed on the line that begins int addr:. This is the number that you enter into the brackets. Remember on the Glove Pi to enter the IP address of the other player's Pi, not yours.

Run both programs

15 No game would be complete without some healthy competition and strategy. A second program is deployed by the other player on the network which tracks and registers if they catch or hit you. The program checks the block that they have hit and compares it to the player's location.

Test for hits

16 To check if the other player has hit you, run the second program on the Pi of your opponent. The program finds the other 'glove' players current position and stores it. It then compares the position that you hit with your sword, storing this too. The program compares values, if they match then you have hit the other player. Note that in order to monitor where the other player is, you must set the code line mc1 = minecraft. Minecraft.create( ) to their IP address.

Game on

17 Check again that the IP addresses are set for the other Raspberry Pi and not your own. Build a new Minecraft world and start a new game on the Raspberry Pi with the player who is chasing. When loaded, the glove player joins the multiplayer game – this will be called Steve (see Step 11). When loaded, you should see both players in the world. Then run the 'Pi Glove power moves' program, and on the other Pi run the 'You hit me program'.

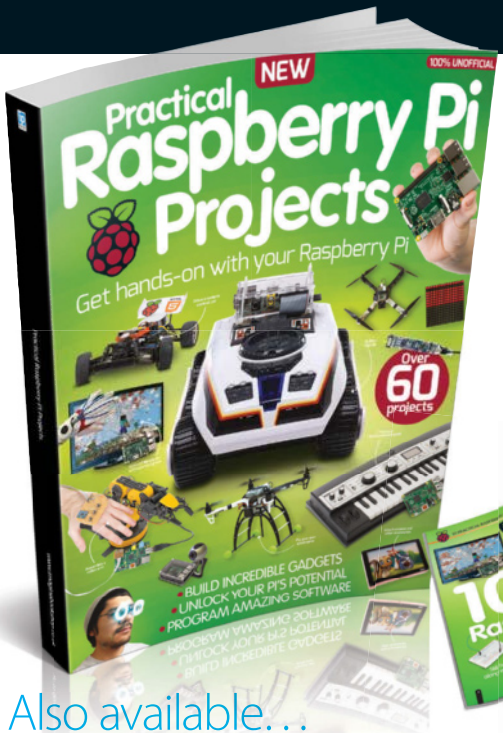
PiGlovePowerMoves.py (Cont.)

```
#Main code to run
try:
    lava_check = 0
    mc.postToChat("Minecraft
Power Glove Enabled")
    while True:
        if GPIO.input(17) ==
0:
            Firewall()
        if GPIO.input(18) ==
0: #needs to stop
            Lay_Lava()
```

YouWereHit.py

```
import time
from mcpi import minecraft

mc1 = minecraft.Minecraft.
create("192.168.1.245")
#The other players IP address
goes here

mc = minecraft.Minecraft.
create()
mc.postToChat("###Here I
come")
Hit = 1

while Hit == 1:

#Find the block stood on
  stood_x, stood_y, stood_z =
mc1.player.getTilePos()
            #GPIO.output(18,
GPIO.LOW)
        if GPIO.input(9) == 0:
            Mega_Jump()
        if GPIO.input(4) == 0:
            Mega_Water_
Puddle()

except:
    print "Error"

  time.sleep(3)

  blockHits = mc.events.
pollBlockHits()
  if blockHits:
    for blockHit in blockHits:
      if stood_z == blockHit.
pos.z and stood_y == blockHit.
pos.y+1:
        mc.postToChat("I got
you")
        time.sleep(2)
        mc.postToChat("###GAME
OVER###")
        time.sleep(1)
        Hit = 0

mc.postToChat("###Restart Hit
Code")
```

From the makers of Linux User & Developer



Practical Raspberry Pi Projects

Discover the potential of your Raspberry Pi and uncover what it's truly capable of. Unleash the power of your Pi by creating your own virtual assistant, drum machine, quadcopter and so much more in this comprehensive guide.

Also available...

A world of content at your fingertips

Whether you love gaming, history, animals, photography, Photoshop, sci-fi or anything in between, every magazine and bookazine from Imagine Publishing is packed with expert advice and fascinating facts.

BUY YOUR COPY TODAY

Print edition available at www.imagineshop.co.uk
Digital edition available at www.greatdigitalmags.com

Print Tablet Phone Desktop

Guides for Pi & Pi 2

Raspberry Pi

The Complete Manual

- ✔ **Introducing the Raspberry Pi**
  Find your way around the most recent versions of the innovative Raspberry Pi computer
- ✔ **Set up your Pi**
  Make sure you have the essential kit and the know-how to get started straight out of the box
- ✔ **Master Raspbian**
  Get comfortable using the official Raspberry Pi operating system by following simple tutorials
- ✔ **Understand applications**
  Everything you need to know about the best applications included with the Raspberry Pi
- ✔ **Get to grips with Linux**
  Learn how to use common Linux applications and master the command line basics
- ✔ **Get started with programming**
  Navigate the world of Scratch and Python programming through easy-to-follow tutorials
- ✔ **Take control with your Pi**
  Use the GPIO pins on your Raspberry Pi to control lights, switches and more
- ✔ **Create a wireless hotspot**
  Access the internet anywhere by tethering your Raspberry Pi to your Android device
- ✔ **Build & code with the web**
  Tether to an Android device, print wirelessly and stream music to your smartphone
- ✔ **Creative projects made easy**
  Add a battery pack, send SMS messages, build a Raspberry Pi car and much more!