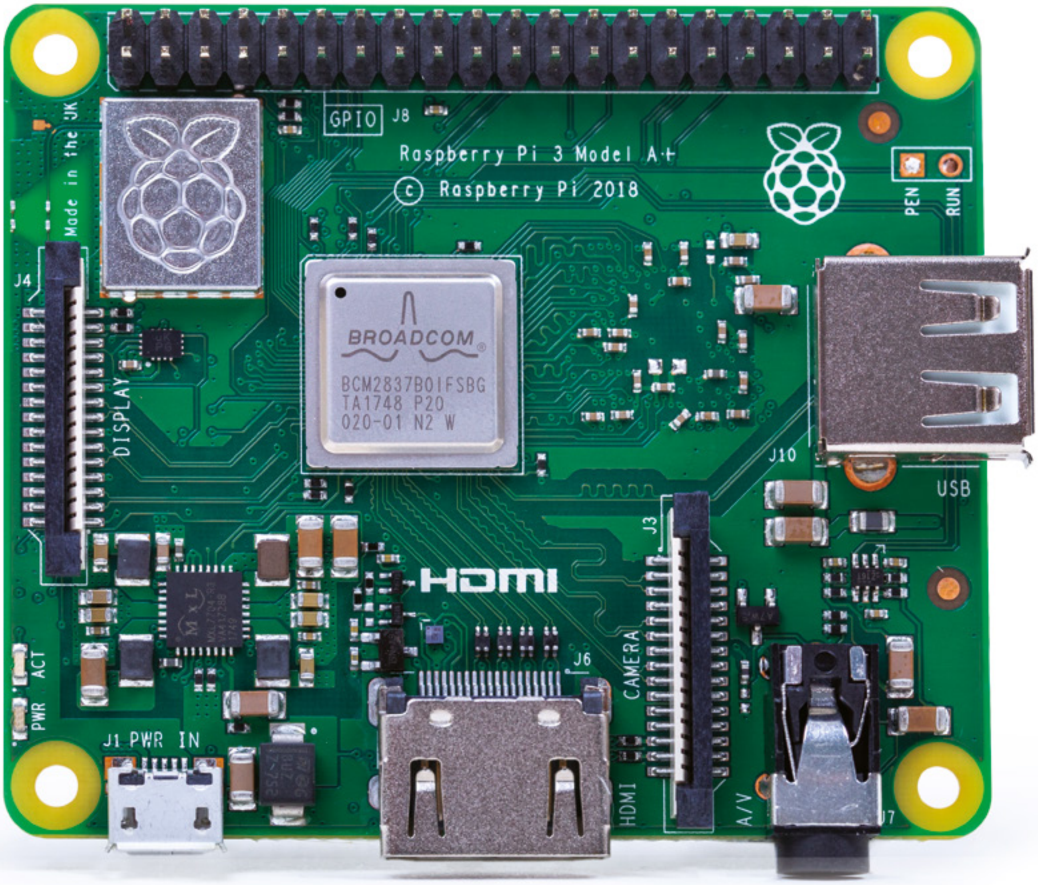
FREE KIT **NOT INCLUDED** with **digital edition**

# GET STARTED

## WITH RASPBERRY PI



FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

The MagPi ESSENTIALS
2ND EDITION
CONQUER THE COMMAND LINE
The Raspberry Pi TERMINAL GUIDE
UPDATED FOR 2019! + 4 NEW CHAPTERS

FREE KIT **NOT INCLUDED** with **digital edition**

# WELCOME TO GET STARTED WITH RASPBERRY PI!

**You are holding, in your hands, a piece of history.** This isn't just a book about a computer: It's a book with a computer.

Not just any computer either. Raspberry Pi 3A+ is an incredibly well-designed microcomputer. And it's part of the most creative computer family on Earth.

With a Raspberry Pi, you can hack, make and build all kinds of different things. It could be a digital camera, or a retro games console, or a home media centre. Or a sensor on board the International Space Station, or a programmable Minecraft machine.

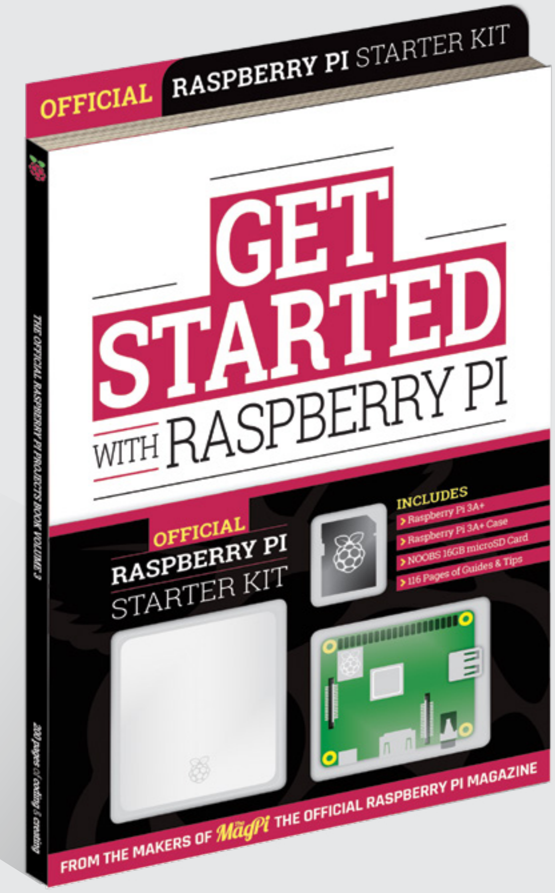
Almost everything you need to get started with a Raspberry Pi computer is inside this kit, including a memory card for the operating system and storage. All you need to add is a USB keyboard and mouse, HDMI cable, and 5V micro USB power supply (such as a phone charger).

Our tutorials will guide you, step-by-step, from setting up Raspberry Pi 3A+ hardware to learning how to use the Raspbian operating system, through to hacking electronics with the GPIO pins on your Raspberry Pi 3A+.

You'll also learn how to program a computer with Scratch and Python, two of the world's best programming languages.

We're so glad you've got Get Started with Raspberry Pi with the Official Starter Kit. Now let's begin...

**Lucy Hattersley**
Editor


OFFICIAL RASPBERRY PI STARTER KIT
GET STARTED WITH RASPBERRY PI
OFFICIAL RASPBERRY PI STARTER KIT
INCLUDES
Raspberry Pi 3A+
Raspberry Pi 3A+ Case
NOOBS 16GB microSD Card
116 Pages of Guides & Tips
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 5
200 pages of coding & creating

**FIND US ONLINE** magpi.cc **GET IN TOUCH** magpi@raspberrypi.org

The MagPi

## EDITORIAL
Editor: **Lucy Hattersley**
Features Editor: **Rob Zwetsloot**
Book Production Editor: **Phil King**
Contributors: **Wesley Archer, Gareth Halfacree, Dave Honess, Sean McManus, Marc Scott, Mark Vanstone**

## DISTRIBUTION
**Seymour Distribution Ltd**
2 East Poultry Ave, London,
EC1A 9PT | **+44 (0)207 429 4000**

## DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Lee Allen**
Designers: **Sam Ribbits. Mike Kay**
Illustrator: **Sam Alder**

## MAGAZINE SUBSCRIPTIONS
Unit 6, The Enterprise Centre, Kelvin Lane,
Manor Royal, Crawley, West Sussex,
RH10 9PE | **+44 (0)207 429 4000**
**magpi.cc/subscribe**
**magpi@subscriptionhelpline.co.uk**

## PUBLISHING
Publishing Director: **Russell Barnes**
**russell@raspberrypi.org**

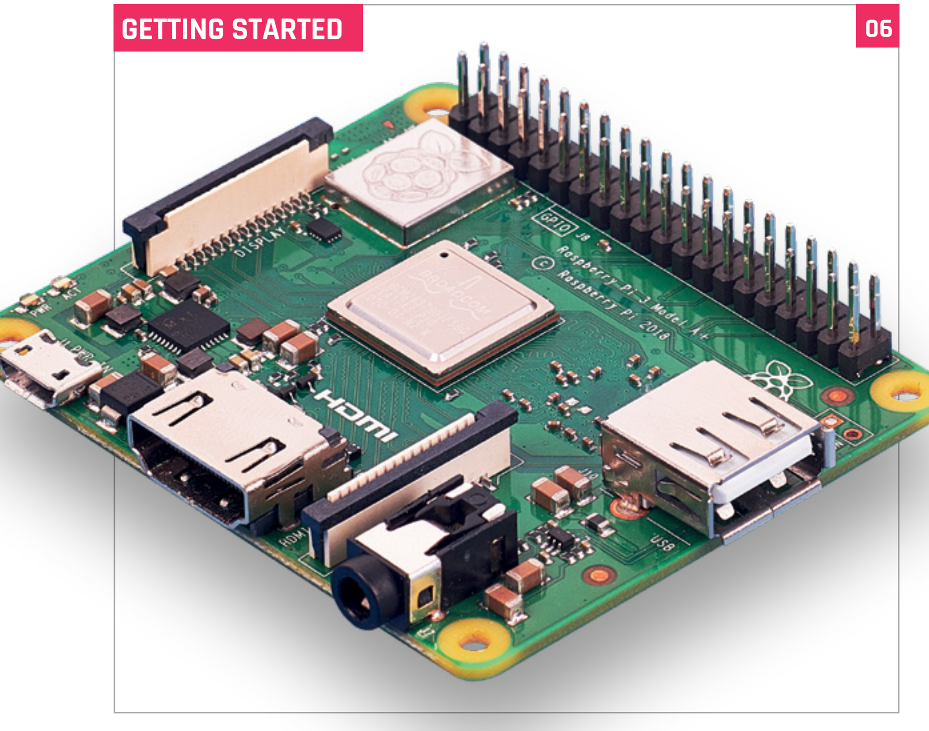
Advertising: **Charlotte Milligan**
**charlotte.milligan@raspberrypi.org**
Tel: +44 (0)7725 368887

Director of Communications: **Liz Upton**
CEO: **Eben Upton**

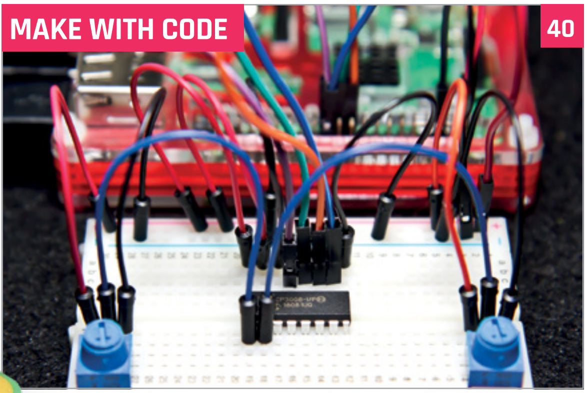
Available on the App Store
Get it on Google play

This official product is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISBN: 978-1-912047-70-3.

# Contents

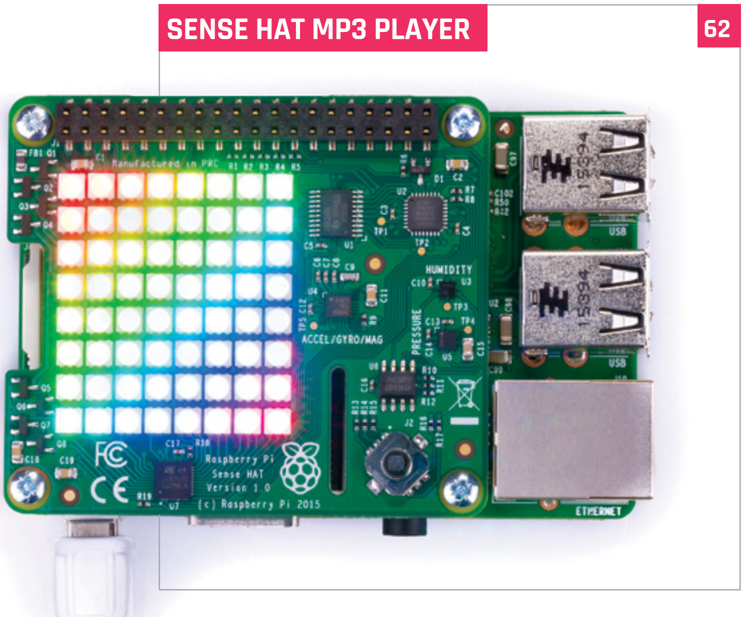


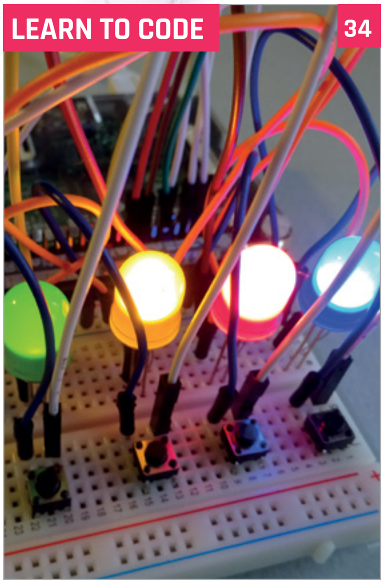
GETTING STARTED 06

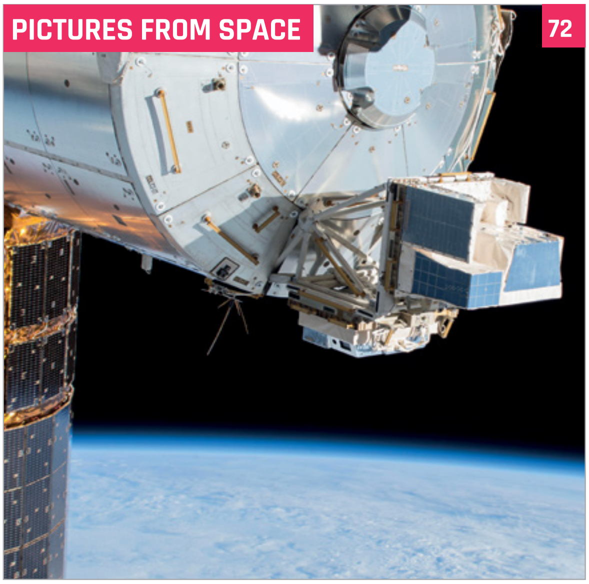


MAKE WITH CODE 40

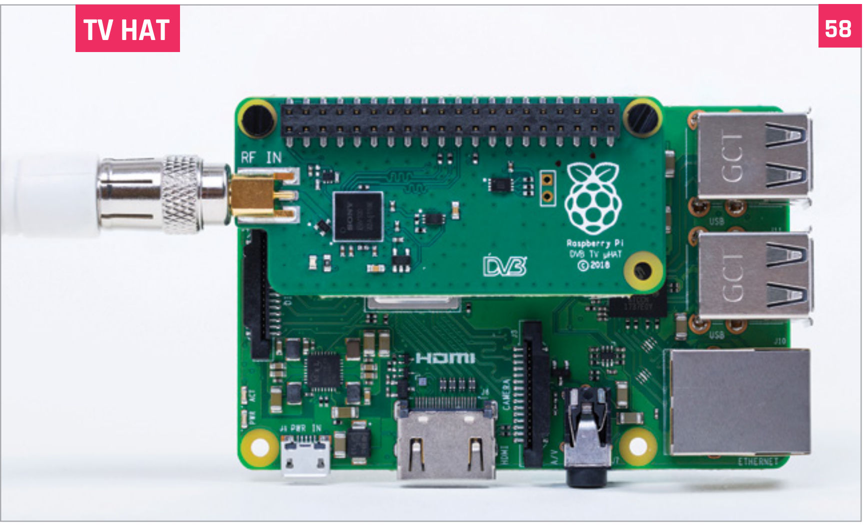
INSTALLING RASPBIAN 24

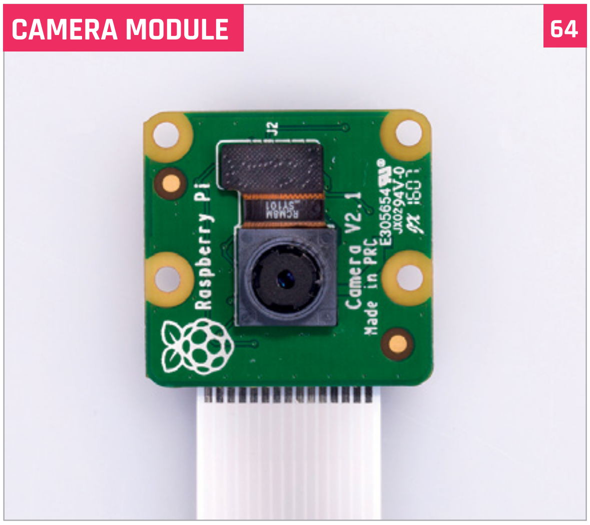






Camera: ○ Enabled ● Disabled
SSH: ● Enabled ○ Disabled
VNC: ○ Enabled ● Disabled
SPI: ○ Enabled ● Disabled
I2C: ○ Enabled ● Disabled
Serial Port: ○ Enabled ● Disabled
Serial Console: ● Enabled ○ Disabled
1-Wire: ○ Enabled ● Disabled
Remote GPIO: ○ Enabled ● Disabled





# GETTING STARTED WITH RASPBERRY PI 3A+

Packing an incredible amount of power into a small form factor, Raspberry Pi 3A+ is a superb single-board computer

**Raspberry Pi 3A+ is a low-cost microcomputer packed onto a single board.** While smaller than a Raspberry Pi 3B+, it packs all of the latter's processing power into a more compact form factor – not as tiny as a Raspberry Pi Zero, but still small enough to fit in the palm of your hand.

Despite its diminutive stature, Raspberry Pi 3A+ is no toy: it is a fully-fledged microcomputer with a 1.4GHz quad-core ARM CPU and 512MB RAM. It packs enough technology to run the full version of the Raspbian operating system with ease.

Raspberry Pi 3A+ is a versatile device that's ideal for creating Internet of Things and embedded projects requiring more power than a Raspberry Pi Zero. To keep the size down, it lacks an Ethernet port and features a single USB port – so you'll need a USB hub to connect more than one device.

A full-size HDMI socket enables you to use a standard HDMI cable to connect to a TV or monitor. It uses the same micro USB power input as most other Raspberry Pi devices, and you can use an official adapter or salvage a high-quality power supply from a mobile phone (2.5 A output is recommended).

Other ports include a 3.5 mm AV jack (which you can use to output audio or connect to a non-HDMI TV) and a connector to attach a Raspberry Pi Camera Module to turn Raspberry Pi 3A+ into a low-cost camera for taking photos and recording videos.

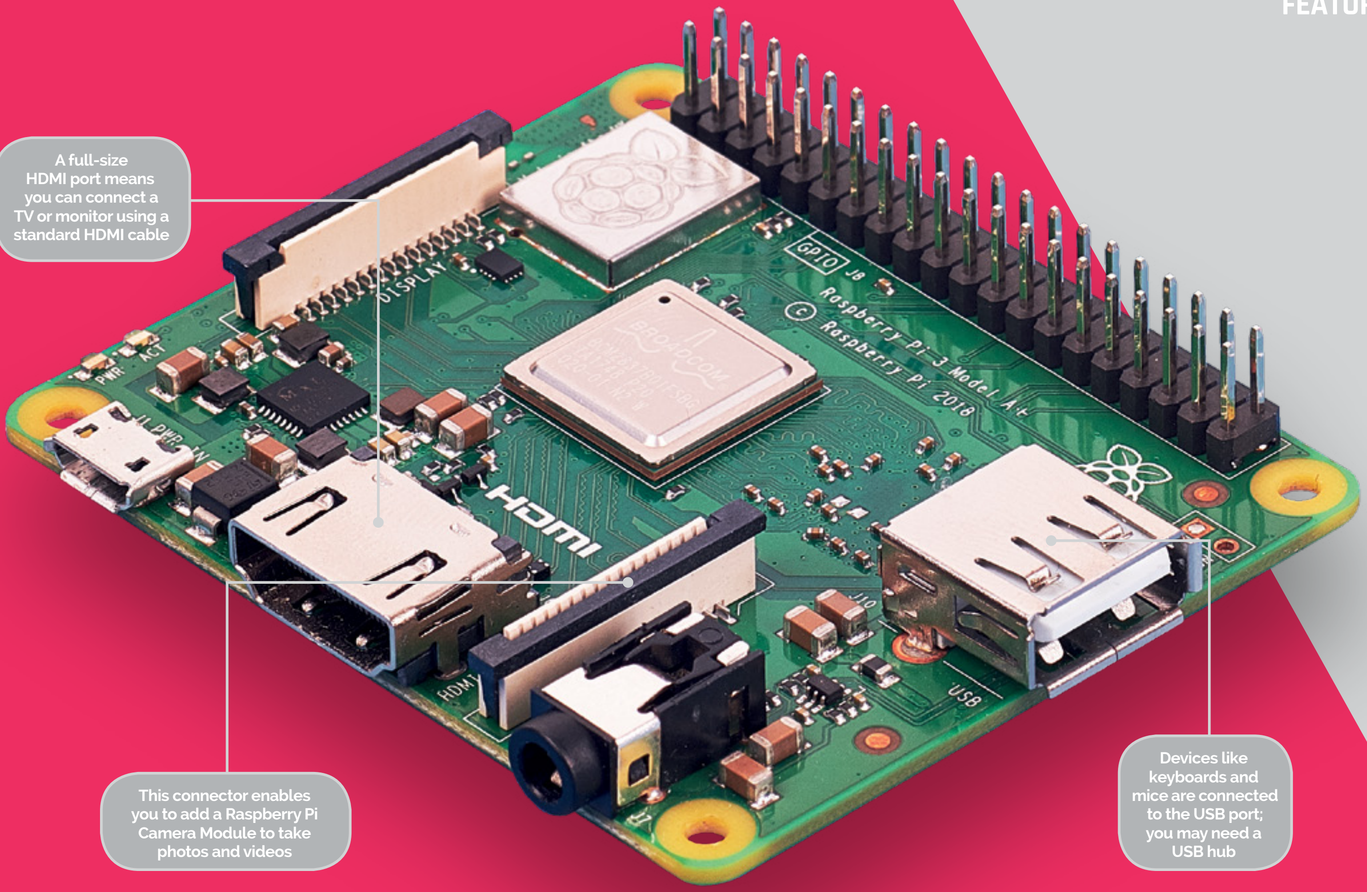
## RASPBERRY PI 3A+

### Powerful processor

Raspberry Pi 3A+ packs a powerful 1.4GHz quad-core ARM 54 CPU with 512MB RAM, enabling it to run a full desktop operating system with ease.

### Small form factor

It offers a full computer experience, complete with the Raspbian operating system, in a smaller form factor than a full-size Raspberry Pi 3B+ or 4 model.



A full-size HDMI port means you can connect a TV or monitor using a standard HDMI cable

This connector enables you to add a Raspberry Pi Camera Module to take photos and videos

Devices like keyboards and mice are connected to the USB port; you may need a USB hub

Hooking a Raspberry Pi 3A+ up to the internet is easy, thanks to built-in wireless networking. It also features Bluetooth 4.2 and LE so you can attach wireless devices.

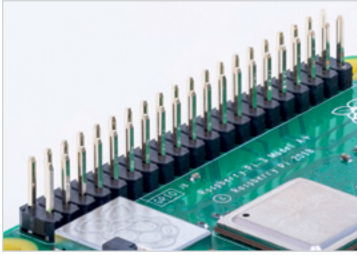
Raspberry Pi 3A+ has the full 40-pin GPIO header of the other Raspberry Pi models, enabling you to connect and control electronics or plug in various HAT (Hardware Attached on Top) devices.

Setting up a Raspberry Pi 3A+ is relatively simple and the end result is a pocket-sized, powerful, versatile computer that runs a full operating system and can be used for a whole range of projects. ■

The official case supplied in your Starter Kit protects Raspberry Pi 3A+ while offering access to all its ports and pins, thanks to cutouts and a removable lid

## GPIO to go

The 40-pin GPIO (general-purpose input/output) header enables Raspberry Pi to talk to other hardware, from LEDs and buttons to special add-on boards.



# EQUIPMENT YOU'LL NEED

All the kit you need to get a Raspberry Pi up and running for the first time

**You don't require much to get your Raspberry Pi started: a smartphone charger, a recycled HDMI cable, and a keyboard and mouse are all you need.**

Most items can be sourced from spare computer hardware around the house, or begged and borrowed from friends and family. If you're looking for the ultimate in low-cost computing, Raspberry Pi is it.

You should be able to source, salvage, and scavenge most equipment you need to get a Raspberry Pi up and running.

To get the most out of your Raspberry Pi in the long term, though, you should use high-quality components.

Not all USB power supplies are born equal. A reliable branded supply (such as the official Raspberry Pi one) will provide a steady stream of power, even when you attach multiple devices to Raspberry Pi.

Any equipment you can't recycle can be picked up from various online retailers, such as The Pi Hut and Pimoroni. For official add-ons, visit the Raspberry Pi products page (**rpf.io/products**).

## IN YOUR KIT: MICRO SD CARD

The microSD card in your kit acts as the hard drive for your Raspberry Pi. After removing it from the SD adaptor, you install the Raspbian operating system onto the card, then all your documents, files, and projects are saved to it as you work. The microSD card in your Starter Kit is a high-quality piece of equipment and comes with NOOBS pre-installed.

If you want to use a larger capacity card, take a look at the results from benchmark tests done by Raspberry Pi fan Jeff Geerling. Some cards run up to four times as fast as others. You can read more at **magpi.cc/xZLRky**.

The lid of the official Raspberry Pi case can be unclipped, offering fast access to the GPIO pins on the board inside

Raspberry Pi 3A+ can be used without a case, but it's better protected inside one

The official case features cutouts to provide easy access to all of the ports on Raspberry Pi 3A+

## HDMI cable

An HDMI cable is the easiest way to connect your Raspberry Pi to a computer monitor or television. You don't need an expensive one, and most people recycle one from an old games console or DVD player.

## USB power

A good 2.5A power supply provides you with enough power to run your Raspberry  Pi 3A+. Many people use an Android or iPhone adapter and micro USB cable. Or you can buy an official Universal Power Supply (**magpi.cc/2a14pye**).

## Keyboard and mouse

Any standard USB keyboard and mouse can be used with Raspberry Pi – or a Bluetooth keyboard and/or mouse, but wired devices are easier to set up. The official Raspberry Pi Keyboard (**magpi.cc/keyboard**) features a built-in USB hub to connect a mouse and other devices.

## USB hub

Raspberry Pi 3A+ has a single USB socket. You may be able to connect a wired mouse to a spare USB socket on your keyboard. Otherwise, you'll need a USB hub like this (**magpi.cc/2ykHg8r**). Connect the mouse and keyboard to the hub, and the hub to Raspberry Pi 3A+.

# SET UP YOUR RASPBERRY PI 3A+

Discover how to set up your Raspberry Pi 3A+ Starter Kit and start using it

### You'll Need

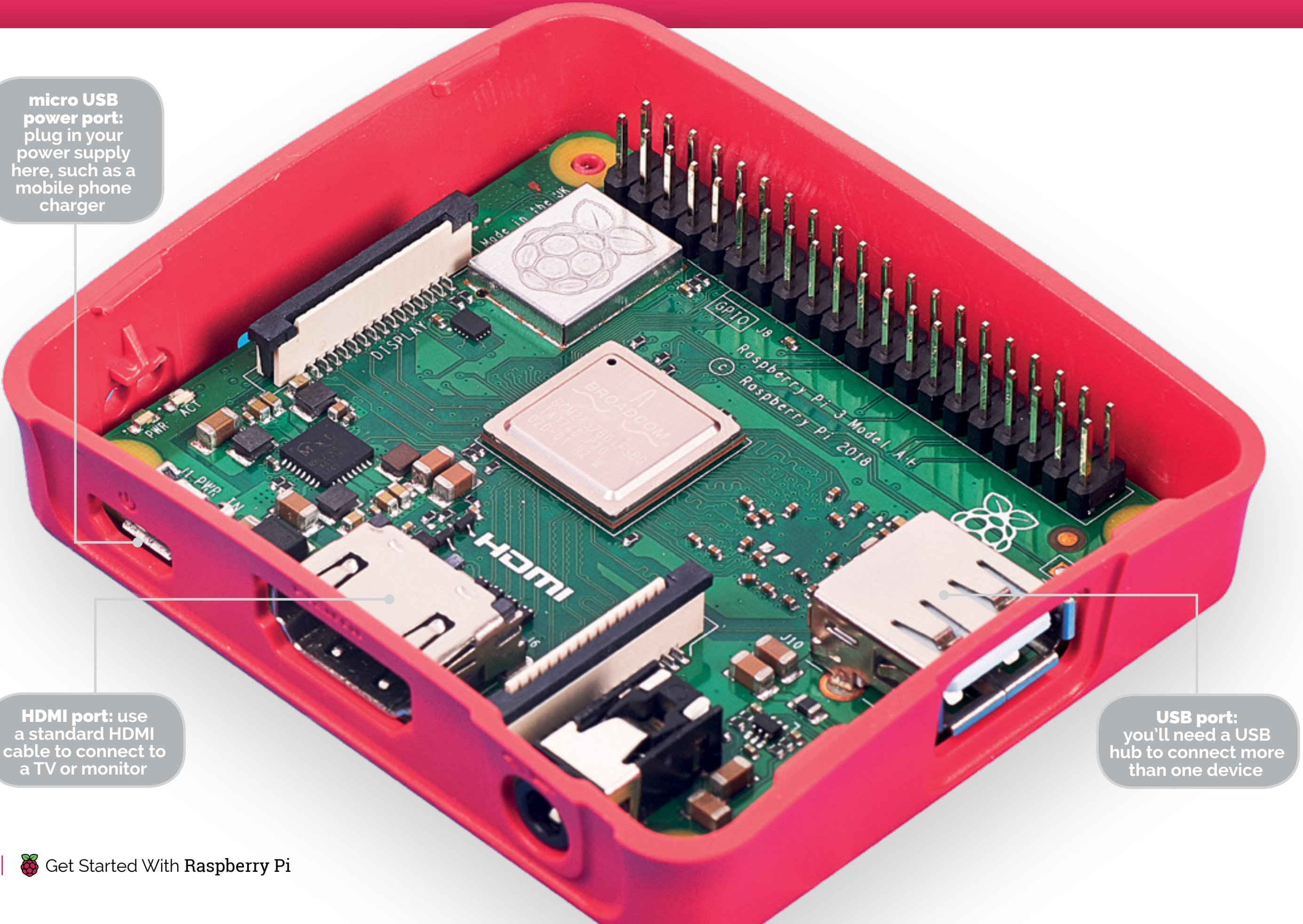
- Your Starter Kit
- USB mouse
- HDMI cable
- micro USB power supply

**So, you've unpacked your Starter Kit and are wondering what to do with all the bits?** Fear not, as here we'll guide you step by step through the process of setting up the hardware and installing the Raspbian operating system.

Raspberry Pi 3A+ may be small, but it's a fully fledged computer. To keep the size of the board down, however, it lacks the Ethernet port featured on larger models (such as Raspberry Pi 3B+ and 4) and there is only a single USB port. Therefore, to connect both a keyboard and mouse you'll either have to daisy-chain them using the USB port on the keyboard (if it has one) or use a USB hub.

On another edge of the board, you'll see a full-size HDMI port for connecting your Raspberry Pi 3A+ to a TV or monitor – all you need is a standard HDMI cable.

Turning your Raspberry Pi 3A+ board over, you'll see a small silver slot – this is where you'll insert your microSD card from the Starter Kit. Let's start putting it all together…



**micro USB power port:** plug in your power supply here, such as a mobile phone charger

**HDMI port:** use a standard HDMI cable to connect to a TV or monitor

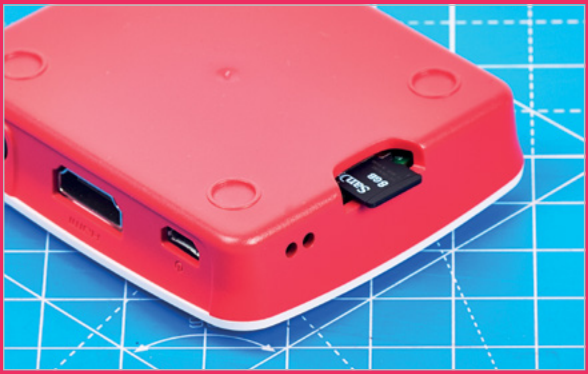
**USB port:** you'll need a USB hub to connect more than one device

# HOW TO: SET UP RASPBERRY PI 3A+

## >STEP-01

### Put it in the case

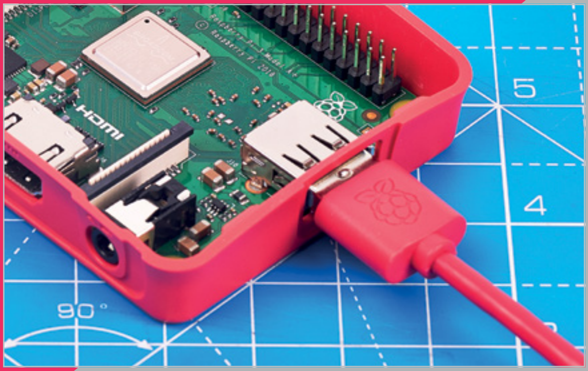
Place your Raspberry Pi 3A+ into the red base of the case, making sure that the USB, HDMI, 3.5 mm AV, and power ports line up with the cutouts in the case.



## >STEP-02

### Insert microSD card

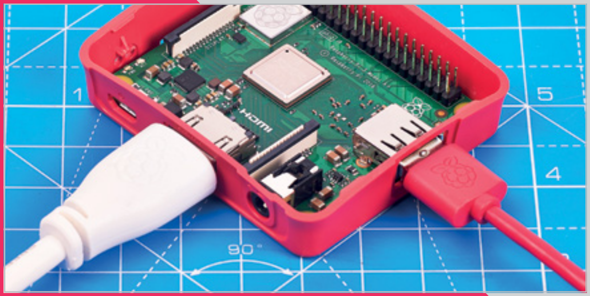
Turn the case over. That silver slot takes microSD cards like the NOOBS one in your Starter Kit. Insert the card into the slot with its metal contacts facing down – it'll only fit one way.



## >STEP-03

### Plug in a mouse

Connect a standard mouse to the USB port. If using a keyboard with its own USB port/hub, you could daisy-chain a keyboard and mouse, but only a mouse is needed for initial setup.



## >STEP-04

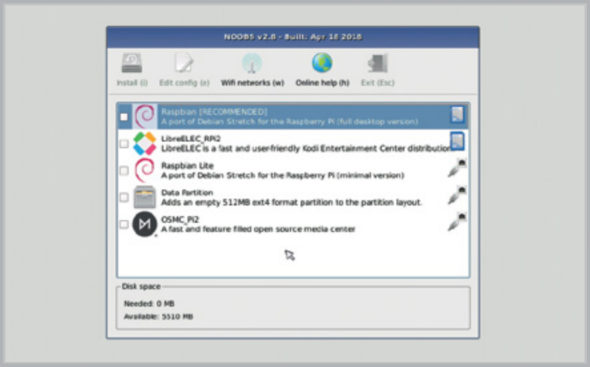
### Connect to a TV

Insert a standard HDMI cable into the large HDMI port and connect the other end of the cable to an HDMI socket on your TV or monitor. Turn the latter on and select the appropriate video input.

## >STEP-05

### Power it up

You're now ready to power up your Raspberry Pi. Connect a power supply (such as a phone charger) to its micro USB port. There is no on/off switch, so turn on the power at the mains.



NOOBS v2.8 - Built: Apr 16 2018

□ Raspbian [RECOMMENDED]
A port of Debian Stretch for the Raspberry Pi (full desktop version)

□ LibreELEC_RPi2
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution

□ Raspbian Lite
A port of Debian Stretch for the Raspberry Pi (minimal version)

□ Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

□ OSMC_Pi2
A fast and feature filled open source media center

Disk space
Needed: 0 MB
Available: 5510 MB

## >STEP-06

### Install Raspbian

A menu will soon appear on screen. Click the checkbox to the left of Raspbian, then click Install – it'll take a while. When prompted, click OK to start up Raspbian and after a few seconds you'll see its desktop interface.

# INSTALLING RASPBIAN

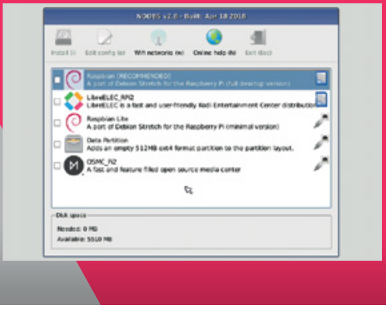
Discover how to use NOOBS to quickly set up the Raspbian operating system on your Raspberry Pi

**Before you start using your Raspberry Pi, it needs to have an operating system (OS).** This is the software used to start the hardware, and open and close programs.

Most Raspberry Pi owners use an open-source operating system called Raspbian, which is based on Linux. The current version is based on a version of Linux called Debian Buster, hence the name Raspbian (sometimes you'll hear it called 'Raspbian Buster').

You can install a range of different OSes on a Raspberry Pi, including LibreELEC, a version of the Kodi Entertainment Centre software. To view and install other OSes, click on the 'Wifi Networks' option in the NOOBS menu.

Raspbian is the official OS and the one most beginners should start with, however. It's the simplest to install, easiest to use, and most projects and tutorials use Raspbian as their base.



NOOBS v2.8 - Built: Apr 18 2018

□ Raspbian [RECOMMENDED]
A port of Debian Stretch for the Raspberry Pi (full desktop version)

□ LibreELEC_RPi2
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution

□ Raspbian Lite
A port of Debian Stretch for the Raspberry Pi (minimal version)

□ Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

□ OSMC_Pi2
A fast and feature filled open source media center

Disk space
Needed: 0 MB
Available: 5510 MB

NOOBS automates the process of installing Raspbian. Select the Raspbian option and click on Install to run it

## Start with NOOBS

There are two approaches to installing Raspbian and other operating systems. Beginners should start with NOOBS (New Out Of Box Software). More advanced users may copy an image file containing a whole operating system directly to the SD card.

In your Starter Kit is a microSD card (inside the larger SD adapter) preloaded with NOOBS.

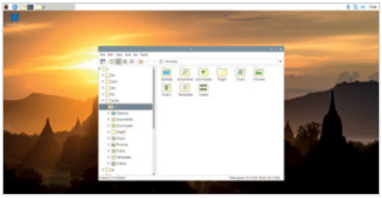
If you wish to use a different microSD card, you must first wipe it using the Windows FAT32 format The easiest way to do this on a Mac or Windows PC is to use SD Card Formatter (**magpi.cc/sdcard**).

Connect your microSD card to a Mac or Windows PC, typically using a microSD-to-SD card adapter or a USB card reader, and use SD Card Formatter to erase the card.

## OPERATING SYSTEMS

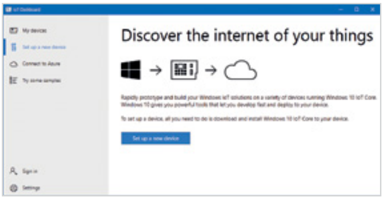
### Raspbian

The official operating system is the easiest to use, and the one beginners should start with. It works a lot like other popular operating systems.



### Windows 10 IoT Core

Not the full version of Windows, but Windows 10 IoT Core enables programmers to run Internet of Things and embedded projects.



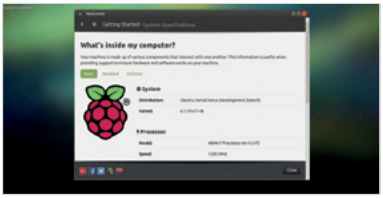
Discover the internet of your things

Rapidly prototype and build your Windows IoT solutions on a variety of devices running Windows 10 IoT Core. Windows 10 gives you powerful tools that let you develop fast and deploy to your device.

To set up a device, all you need to do is download and install Windows 10 IoT Core to your device.

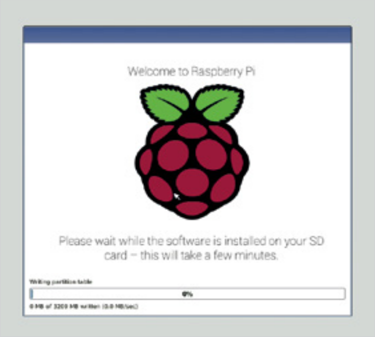
### Ubuntu MATE

Ubuntu is one of the world's most popular Linux operating systems, and Ubuntu MATE is a lightweight version that runs just fine on the Raspberry Pi.



What's inside my computer?

Your machine is made up of various components that interact with one another. This information is useful when providing support to ensure hardware and software works on your machine.



Welcome to Raspberry Pi

Please wait while the software is installed on your SD card – this will take a few minutes.

Writing partition table

0%

0 MB of 3203 MB written (0.0 MB/sec)

NOOBS automatically copies all the files needed to run Raspbian onto your microSD card

Next, download the NOOBS ZIP file from **magpi.cc/2bnf5XF**. Extract the contents of the file and open the NOOBS folder. Copy the contents across to the root of the SD card. See the 'Setting up NOOBS' steps for more information.

With the NOOBS files copied across, remove the microSD card from your computer and slot it into your Raspberry Pi. Now connect the keyboard, mouse, and HDMI cable. Finally, attach the USB power to boot up the Raspberry Pi.

The Raspberry Pi will boot, displaying the NOOBS installer menu. To install Raspbian, place a tick next to the Raspbian option and click Install. Click Yes in the Confirm alert to begin installing it.

Now you just need to wait while the Raspbian file system is extracted. When it's finished, you'll see the Raspbian desktop and the message 'OS(es) Installed Successfully'. Click OK to start using your Raspberry Pi.

## Installing image files

Installing an operating system directly from an image file (rather than via NOOBS) is a slightly more complex procedure, but one that more advanced users should learn.

You format the microSD card to FAT32 as usual, then download the operating system as an image file, a large file ending in '.img'. This file is then copied bit by bit as an exact replica to the microSD card.

There are various methods for doing this, but the easiest way is by using an application called balenaEtcher (**etcher.io**).

Etcher turns the whole process of flashing an OS image file into three simple steps: Select Image, Select Drive, and Flash Image. So it's difficult to go wrong.

More importantly, the same program, with the same interface, is available on all three types of computer – Windows, macOS, and Linux – which makes it easy for everybody to understand.

Learning how to copy image files is essential if you want to use operating systems other than Raspbian or the ones included in the NOOBS installer. Beginners should stick with NOOBS and Raspbian.

### LibreELEC

Based on Kodi, LibreELEC is ideal for setting up your Raspberry Pi as a home theatre PC to play back and stream video and audio.

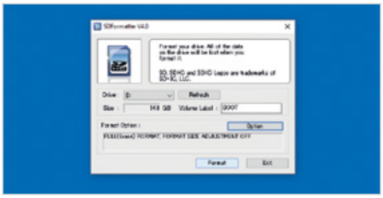
### OSMC

Also based on Kodi, OSMC (Open Source Media Centre) is an easy way to transform your Raspberry Pi into a video and audio player.

# SETTING UP NOOBS

### Format SD card

Open SD Card Formatter and you'll see your memory card in the Drive letter. Change the Volume Label to BOOT so you can identify it later. Now click Option and change Format Type to Full (Erase). Ensure Format Size Adjustment is set to Off and click OK. Click Format, then OK. Click Exit to close SD Card Formatter when it's finished.



SDFormatter V4.0

Format your drive. All of the data on the drive will be lost when you format it.

SD, SDHC and SDXC Logos are trademarks of SD-3C, LLC.

Drive : D: Refresh

Size : 14.8 GB Volume Label : BOOT

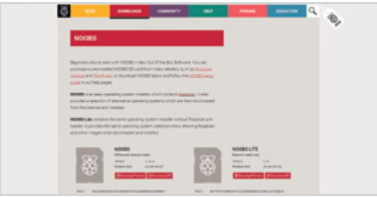
Format Option : Option

FULL(Erase) FORMAT, FORMAT SIZE ADJUSTMENT OFF

Format Exit

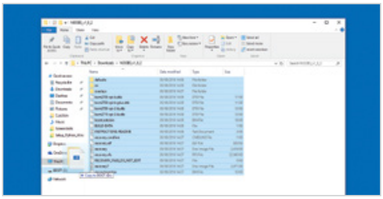
### Download NOOBS

The microSD card in your kit already has NOOBS installed on it. So you should only use these instructions to reinstall NOOBS. In a browser, visit **magpi.cc/2bnf5XF**. Click on Download ZIP. When the zip file download is complete, double-click it to uncompress the folder.



### Copy NOOBS files

Open the freshly extracted folder so you can view all the files. You'll need to copy all the files from the NOOBS folder to your SD card. Press **CTRL+A** (⌘**+A** on a Mac) to select all the files, then drag all the files to the SD card folder. Once they've copied across, eject your SD card. Be careful to copy the *files inside* the NOOBS folder to the microSD card (not the NOOBS folder itself).



# RASPBERRY PI STARTER GUIDE

## Get to know your brand new **Raspberry Pi**

By **Sean McManus**

**So, you've followed our guides and now have your Raspberry Pi 3A+ hardware set up and have installed Raspbian from the NOOBS microSD card. What next?**

We're here to help with a Starter Guide that should teach you some of the basics (and beyond!) of how to use it. First, we'll give you a guided tour of the Raspbian operating system, show you how to configure its settings, introduce you to some awesome applications, and reveal how to install extra software. We'll also show how to connect to and control your Raspberry Pi remotely from another computer using  SSH and VNC.

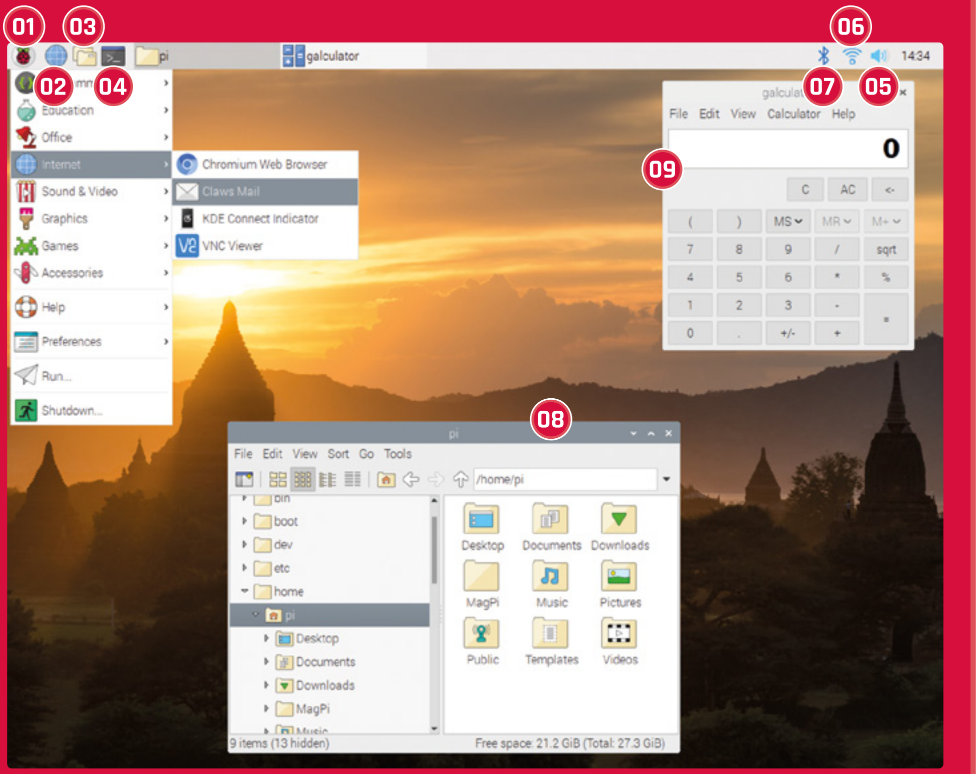
# Meet Raspbian with Desktop

**01** Click the Raspberry Pi logo to open the applications menu. This is where you'll find the software that's pre-installed in Raspbian, and anything else you add later.

**02** Click the globe for quick access to the Chromium web browser.

**03** Click the folders to open the File Manager. You can use it to find, move, copy, and delete files on your storage devices. Why not explore the Linux file system?

**04** Find the Terminal here. It gives you powerful tools for managing your files and devices, and the command line is often the quickest way to get things done.



**05** Volume control. Right-click to select audio output.

**06** WiFi options. Turn WiFi on and off, and switch networks here. If the icon is blue like this, you're connected. You can also hover the mouse pointer over the icon to see your Raspberry Pi's IP address.

**07** Manage Bluetooth connections. You can use Bluetooth devices such as keyboards and mice to wirelessly control your Raspberry Pi device.

**08** File Manager. Use the hierarchical browser on the left or the Go menu to find devices connected to Raspberry Pi. You should store your files in the **/home/pi** folder.

**09** With basic, scientific, and paper modes, this calculator is handy. Find it in the Accessories section of the applications menu.

# Raspberry Pi Configuration

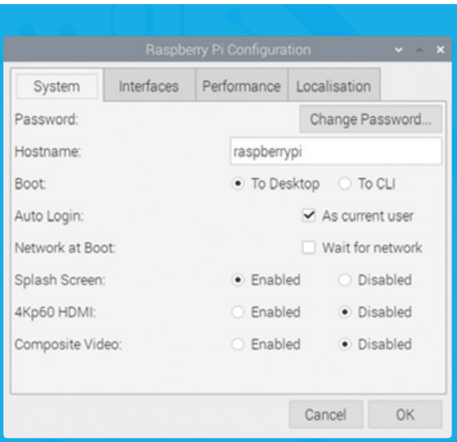
**Need to adjust some settings?** The Raspberry Pi Configuration tool is in the Preferences section of the applications menu. In its System tab, you can change your password, adjust display options, and set Raspberry Pi to boot to the command-line interface (CLI) instead of the desktop.

In the Interfaces tab, you can enable connections, including remote GPIO access and the camera.

To adjust the amount of memory for the GPU or to set older Raspberry Pi models to run faster (overclocking), visit the Performance tab.

You can change the time zone, keyboard, and other geographic options in the Localisation tab.

There are separate options in Preferences for configuring the appearance, audio, main menu, mouse and keyboard, and screen.



| | | |
|---|---|---|
| Password: | Change Password... | |
| Hostname: | raspberrypi | |
| Boot: | ● To Desktop | ○ To CLI |
| Auto Login: | ☑ As current user | |
| Network at Boot: | □ Wait for network | |
| Splash Screen: | ● Enabled | ○ Disabled |
| 4Kp60 HDMI: | ○ Enabled | ● Disabled |
| Composite Video: | ○ Enabled | ● Disabled |

# AWESOME APPLICATIONS

Discover some of the software that comes with Raspbian, and find out how to install more

**Raspbian comes with a selection of pre-installed software (which depends on which version you install), so you can start working, learning, and making things straight away.** Ranging from productivity suites to games, you'll find a well-curated collection of software is just a click away, in the applications menu. There are lots more packages to browse and install, too.

### LibreOffice Writer | OFFICE

No computer is complete without a word processor. LibreOffice Writer has all the font and formatting options you would expect, and has basic compatibility with Microsoft Word.

### LibreOffice Calc | OFFICE

Work out your budgets with this spreadsheet package. If you're familiar with Microsoft Excel, you'll feel at home here. LibreOffice Calc can open and use typical Excel files.

### LibreOffice Impress | OFFICE

If you're presenting to a room of people, don't panic: Impress has your back. Use it to craft and display your slide deck. It's largely compatible with Microsoft PowerPoint.

### Chromium | INTERNET

Chromium is the open-source version of Google Chrome. The default search engine in Raspbian is Duck Duck Go, which promises not to track you online.

### Claws Mail | INTERNET

Send and receive email. The setup wizard helps you add your email account, and the streamlined interface shows you your mailboxes, message list, and a preview of the selected message.

### VLC | SOUND & VIDEO

Listen to music while you're coding using VLC, a fully featured media player for music and video. It can play digital files, streams, and physical media such as CDs and DVDs.

### Minecraft Pi | GAMES

We call it a game, but it's more a way of life. With Minecraft Pi, you can build things in Creative mode, and write programs to change the (game) world.

### Python Games | GAMES

These tea-break games are fun, but they're also great Python demos. Find the code in the **/usr/share/python_games** folder, and open it in a Python editor to see how it works.

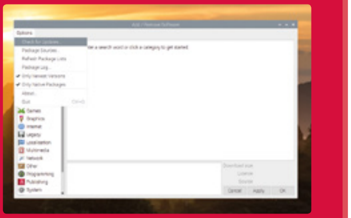
### SmartSim | PROGRAMMING

Experiment with designing and testing digital logic circuits. You can develop custom components and then incorporate them into other circuits. Download examples and read tutorials at **smartsim.org.uk**.

### Sense HAT Emulator PROGRAMMING

This emulator features on-screen controls to simulate temperature, pressure, humidity, and device position changes. A great way to try out the Sense HAT before you buy one!

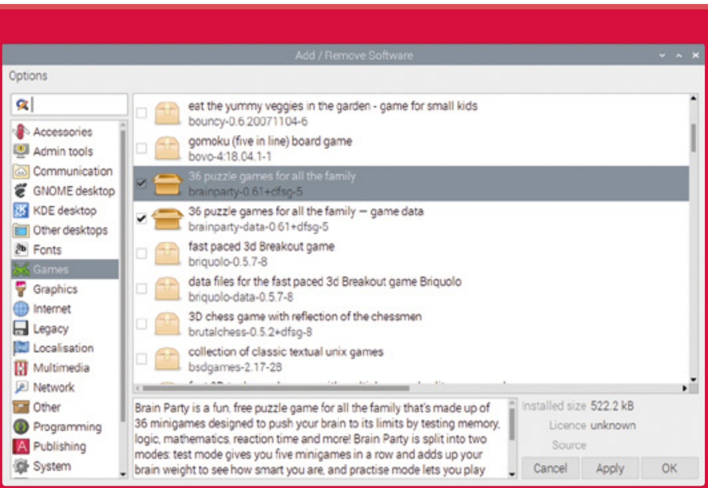
## **Updating** your software



**Linux software comes in packages, which are compressed archives that you can download.** A package manager is used to find and install them, including any other software they need to work.

Use Add/Remove Software to update your packages. From the menu, choose Preferences > Add / Remove Software. Click on Options and Check for Updates.

You can also update from Terminal. In Raspbian, the package manager is called APT. Updating all the software from the Terminal is a two-step process. First, enter `sudo apt-get update` in the Terminal to update the cache of available software. Then enter `sudo apt-get upgrade` to update the software installed on Raspberry Pi. You'll be told what changes will be made and asked to confirm by typing **Y** and pressing **ENTER**.



□ eat the yummy veggies in the garden - game for small kids
bouncy-0.6.20071104-6

□ gomoku (five in line) board game
bovo-4:18.04.1-1

☑ 36 puzzle games for all the family
brainparty-0.61+dfsg-5

☑ 36 puzzle games for all the family – game data
brainparty-data-0.61+dfsg-5

□ fast paced 3d Breakout game
briquolo-0.5.7-8

□ data files for the fast paced 3d Breakout game Briquolo
briquolo-data-0.5.7-8

□ 3D chess game with reflection of the chessmen
brutalchess-0.5.2+dfsg-8

□ collection of classic textual unix games
bsdgames-2.17-28

Brain Party is a fun, free puzzle game for all the family that's made up of 36 minigames designed to push your brain to its limits by testing memory, logic, mathematics, reaction time and more! Brain Party is split into two modes: test mode gives you five minigames in a row and adds up your brain weight to see how smart you are, and practise mode lets you play

Installed size 522.2 kB
Licence unknown
Source

## **Installing** software

**The easiest way to manage software is to use the Add / Remove Software tool in the Preferences part of the applications menu.** It provides a visual front-end for the package manager.

You can click a category on the left to browse applications, or enter a keyword in the search box in the top-left to look for a particular application. Choose the applications you want to install by ticking the box beside them. Some packages require other packages to work properly, but the tool will take care of that for you. To remove an application, untick its box. When you've made your choices, click OK to install or remove your software.

## **Why not** install these?

### Fritzing | PROGRAMMING

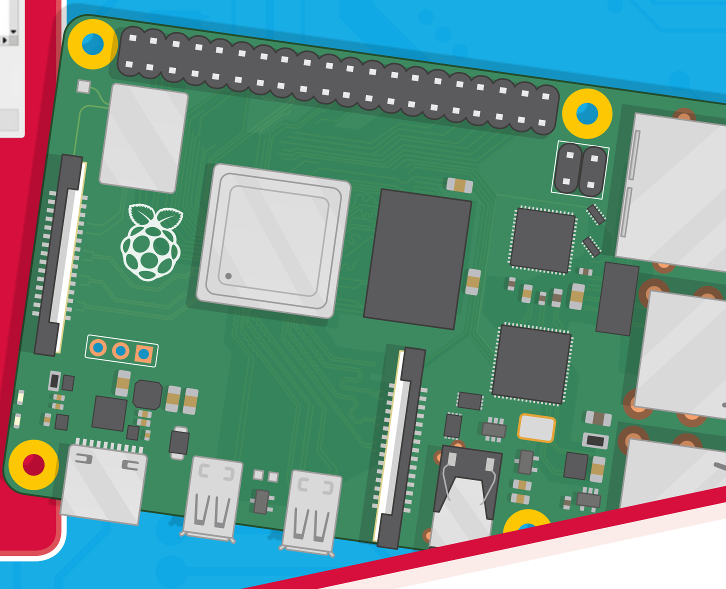
Design and document your electronics circuits based on Raspberry Pi with this design tool. We use it to make circuit diagrams in *The MagPi*.

### GNU Image Manipulation Program (GIMP) | GRAPHICS

GIMP is a powerful image editor. You can use it to create digital art, but its best feature is probably the clone tool, which enables you to retouch holiday photos.

### Mathematica PROGRAMMING

Good for more than just maths, Mathematica uses the Wolfram language, which has data and intelligence built in. Install it from Recommended Software, in the Preferences section of the applications menu.



# GET CONNECTED

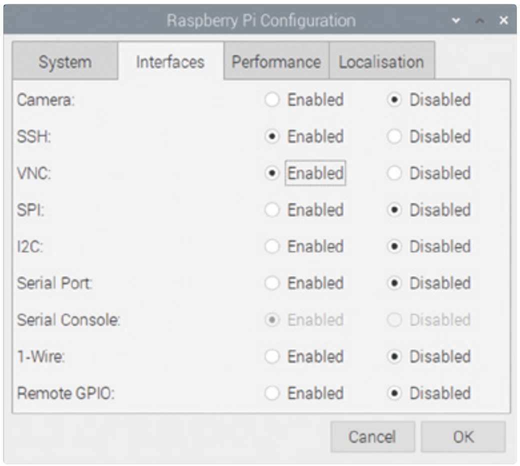
If your Raspberry Pi is up in a tree photographing nature, you don't want to climb up there just to update it. Log in remotely and take control

**There are two technologies you can use to connect to your Raspberry Pi from another device: Secure Shell (SSH) and Virtual Network Computing (VNC).** Before you can use them, they need to be switched on in the Raspberry Pi Configuration settings. To do so, click the Interfaces tab, then enable SSH and/or VNC.

Both approaches require you to know the IP address of your Raspberry Pi. The easiest way to find it is to hover the mouse pointer over the WiFi icon in the taskbar. Alternatively, click the Terminal icon and enter `ifconfig`. It will show you all your network connections. You're looking for an IP address, which will be four numbers with a dot between them, like 198.51.100.0. You'll find it beside 'inet', in the details for wlan0 if you're using WiFi or in the eth0 summary if you're plugged in to the network.

## Using SSH

SSH enables you to use the command line on your Raspberry Pi remotely, so it's very handy for installing software and fixing configuration files. You can't use SSH to run any applications that need the graphical desktop, though.



Raspberry Pi Configuration

Camera: ○ Enabled ● Disabled
SSH: ● Enabled ○ Disabled
VNC: ● Enabled ○ Disabled
SPI: ○ Enabled ● Disabled
I2C: ○ Enabled ● Disabled
Serial Port: ○ Enabled ● Disabled
Serial Console: ● Enabled ○ Disabled
1-Wire: ○ Enabled ● Disabled
Remote GPIO: ○ Enabled ● Disabled

▲ Enable SSH and VNC before you try to use them

The software for SSH is pre-installed on Linux, macOS, and Windows 10. Start by opening the command line. On a Mac, find it by typing 'Terminal' into the Spotlight search. On Windows 10, Use **⊞+R** to open the Run dialog box, then type in **cmd**.

Once you're in the command line, enter `ssh pi@198.51.100.0`, but replace the numbers with the IP address of your own Raspberry Pi device. The 'pi' bit is your username.

> " SSH enables you to use the command line on your Raspberry Pi remotely, so it's very handy "

The first time you connect to a device using SSH, you'll see a warning that shows the ECDSA key for the device you're trying to connect to. You can (if you wish) validate that this is correct by using `ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub` in the Raspberry Pi Terminal. But it's usually OK to just type in **yes** and then press **ENTER** to confirm you want to connect.

You'll be asked to enter the password for your Raspberry Pi device. You won't see the cursor move while you do this, so type on regardless. When you press **ENTER**, you're in! You'll see a Linux welcome message, the date and time of your last login, and then the Raspbian command prompt, waiting for instructions. When you've finished, enter `exit` to leave the SSH session.

If you're using an earlier version of Windows, download PuTTY from **putty.org**. Enter your Raspberry Pi's IP address in the Host Name box, and click Open. Again, you'll see a warning if this is your first connection, which you can safely dismiss. Log in as **pi**, and enter your password. Raspberry Pi is now at your command!

### Tip!

See our SSH tutorial on page 96 for more detailed information on using SSH to remotely control a Raspberry Pi.

▲ Using VNC Viewer to manage Raspberry Pi from an iPad

## Using VNC

Virtual Network Computing (VNC) enables you to remotely access the Raspberry Pi desktop, so you can manage files and run software using it. Some people use VNC to share their keyboard, mouse, and monitor between a PC and Raspberry Pi.

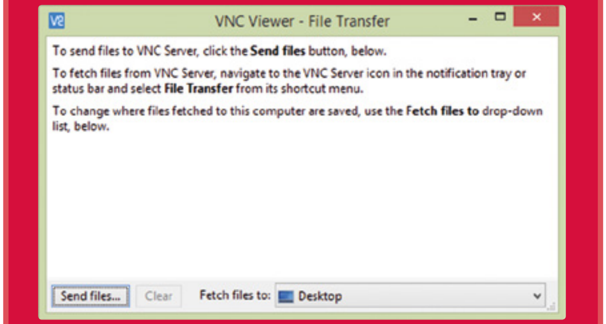
Raspbian includes VNC Server, which runs automatically in the background if you've enabled it in your settings. You'll need to download and install VNC Viewer (**magpi.cc/FuGnye**) on the device you want to use to control Raspberry Pi. VNC Viewer is available for Windows, macOS, and Linux. There are also Android and iOS apps, so you can use VNC to control Raspberry Pi from a mobile device, although it's rather fiddly without a real mouse and keyboard. Optionally, by creating a RealVNC account and registering your Raspberry Pi, you can then access it from anywhere in the world using VNC Viewer.

When you start VNC Viewer, it'll ask you to enter a VNC Server address. This is the IP address you noted when you ran `ifconfig` on your Raspberry Pi.

The first time you connect to a device, VNC Viewer warns you that it has no record of connecting to this device before, and shows you the device's signature and identification catchphrase. To verify you're connecting to the correct device, click the VNC icon on the right of the taskbar in Raspbian to see your device's details.

VNC Viewer prompts you to enter the username and password for the device you're connecting to. You'll then see your Raspbian desktop in the VNC Viewer window. You can now use your keyboard and mouse (or touchscreen on a mobile device) to control the Raspbian desktop, including using programs installed on Raspberry Pi.

## Top Tips for VNC



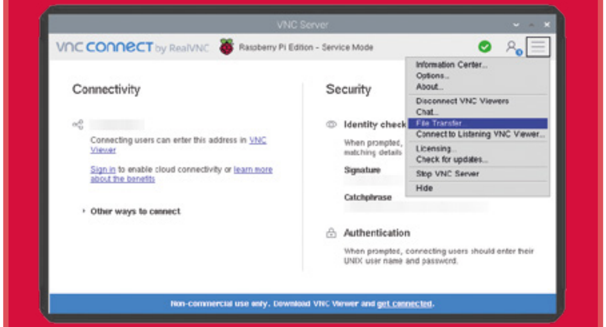
VNC Viewer - File Transfer

To send files to VNC Server, click the **Send files** button, below.

To fetch files from VNC Server, navigate to the VNC Server icon in the notification tray or status bar and select **File Transfer** from its shortcut menu.

To change where files fetched to this computer are saved, use the **Fetch files to** drop-down list, below.

▲ At the top of VNC Viewer in the middle is a tiny toolbar. The File Transfer option enables you to easily copy files to Raspberry Pi.



Connectivity

Connecting users can enter this address in VNC Viewer

Sign in to enable cloud connectivity or learn more about the benefits

Other ways to connect

Security

Identity check

When prompted, matching details

Signature

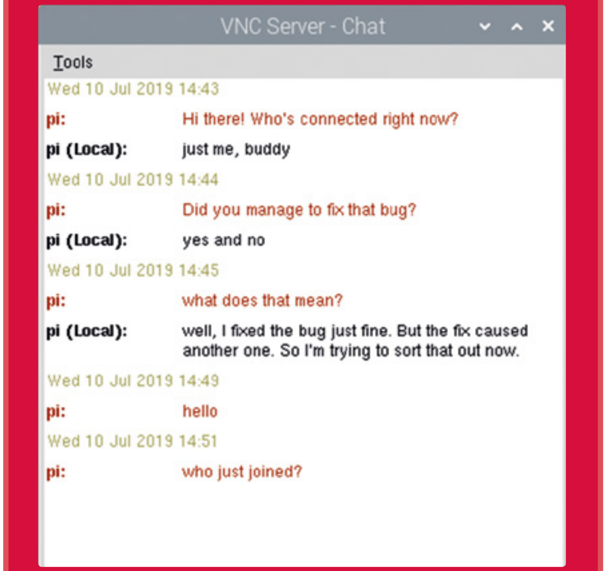
Catchphrase

Authentication

When prompted, connecting users should enter their UNIX user name and password.

Non-commercial use only. Download VNC Viewer and get connected.

▲ Send files from Raspberry Pi using the File Transfer option in VNC Server in Raspbian. Use the three-bars menu (top-right) to find it.



```
Wed 10 Jul 2019 14:43
pi:             Hi there! Who's connected right now?
pi (Local):     just me, buddy
Wed 10 Jul 2019 14:44
pi:             Did you manage to fix that bug?
pi (Local):     yes and no
Wed 10 Jul 2019 14:45
pi:             what does that mean?
pi (Local):     well, I fixed the bug just fine. But the fix caused
                another one. So I'm trying to sort that out now.
Wed 10 Jul 2019 14:49
pi:             hello
Wed 10 Jul 2019 14:51
pi:             who just joined?
```

▲ Use the toolbar in VNC Viewer to start a chat session with anyone using the server (Raspberry Pi device) and anyone else connected to it remotely.

# CODE TO JOY

## Raspberry Pi comes with everything you need to start programming

**If you've never programmed before, you're in for a treat.** Raspbian comes with several integrated development environments (IDEs) you can use to write your own programs, supporting languages that are friendly to use, and fun to tinker with. If you're a veteran coder, you'll find powerful IDEs are included to help you be productive, too.

### Scratch 2

**LANGUAGE:** SCRATCH

With its friendly drag-and-drop commands and a library of sprites and sound effects, Scratch makes it simple to get started with coding. By minimising typing, and guiding you to sensible block combinations, it helps you to avoid common mistakes. Don't be fooled by its accessibility, though: it's a fully-fledged programming language, with plenty of potential.

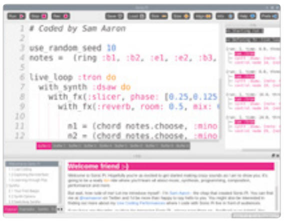
### Thonny

**LANGUAGE:** PYTHON

Python is one of the most popular languages on Raspberry Pi, and Thonny is our favourite way to edit it. In a single window it shows you your code, the shell, and your data (variables), so you can easily see what's going on and fix any bugs that creep in.

### Sonic Pi

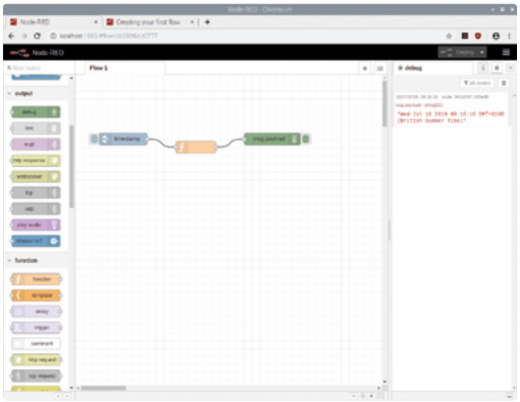
**LANGUAGE:** SONIC PI



```
# Coded by Sam Aaron

use_random_seed 10
notes =  (ring :b1, :b2, :e1, :e2, :b3,

live_loop :tron do
  with_synth :dsaw do
    with_fx(:slicer, phase: [0.25,0.125
      with_fx(:reverb, room: 0.5, mix:

        n1 = (chord notes.choose, :mino
        n2 = (chord notes.choose, :mino
```

Welcome friend :-)

Learn how to compose and perform music with code! Sonic Pi incorporates synths, samples, and effects that you can control by writing programs using a language based on Ruby. You can develop and modify programs while the music plays, to perform live concerts or improvise at home.



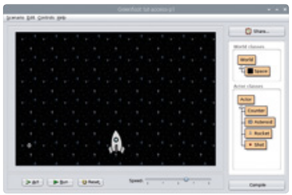
### Node-RED

**LANGUAGE:** JAVASCRIPT/NODE-RED

Node-RED enables you to use a flowchart to program data flows for Internet of Things applications on Raspberry Pi. It uses JavaScript, the language of the web. Run the Node-RED console in Raspbian and then visit **http://localhost:1880** in your browser to program it. For help, see **nodered.org**.

### Greenfoot

**LANGUAGE:** JAVA



Java is one of the world's most popular programming languages. Greenfoot makes it easier to learn, by providing a friendly environment for building simple games. The editor incorporates a game world, and you add Java code to image objects to control their interactions. Find out more at **greenfoot.org**.

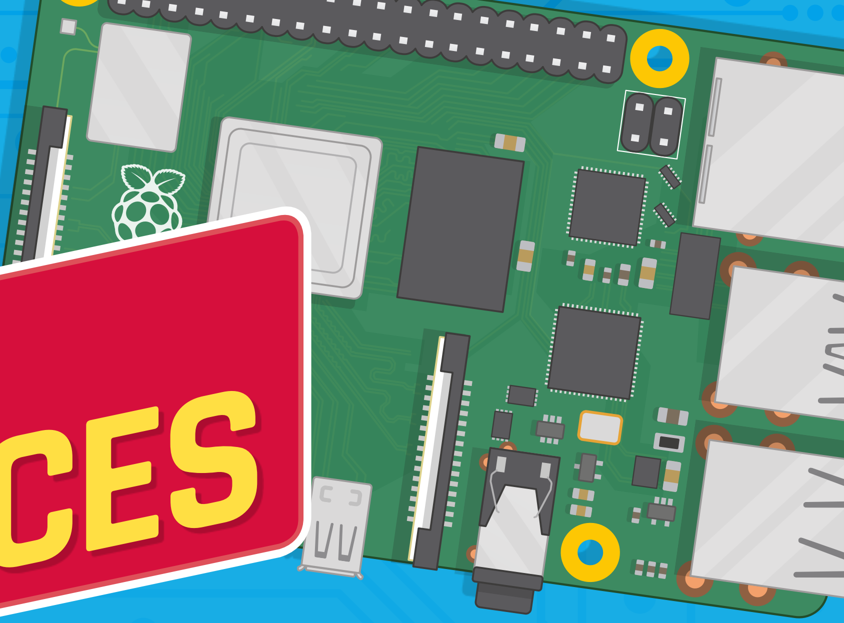
### Geany

**LANGUAGE:** LOTS!

If you're looking for a lightweight but powerful IDE, try Geany. It supports many languages, including HTML, C, Java, PHP, JavaScript, and Perl. Its code auto-completion and syntax highlighting can help you code faster, and reduce errors. Our favourite feature? Code folding, used to show or hide logical chunks of code in a long program.

**Tip!**

See our Learn to Code guide overleaf for information on how to get started programming with Scratch and Python.



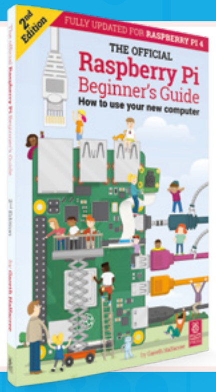
# FURTHER RESOURCES

There's a wealth of resources available to support you as you learn more about Raspberry Pi and Raspbian

## The Official Raspberry Pi Beginner's Guide

Available to buy in print and as a free PDF, this book shows you how to set up Raspberry Pi, and gets you started with programming it in Scratch and Python. It also covers the Sense HAT and Raspberry Pi Camera Module, with code examples you can build and tailor.

**magpi.cc/BGbook**


2nd Edition
FULLY UPDATED FOR RASPBERRY PI 4
THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer

## Official Raspberry Pi Documentation

The documentation provides concise user guides for Linux, Minecraft, Sonic Pi, Scratch, and Python. It's also the place to go for advice on configuring Raspberry Pi, hardware specifications, and remote access tips. Useful for both beginners and power users.

**rpf.io/docs**

## GPIO Zero Documentation

See how easy it is to start programming your own electronics projects for Raspberry Pi. The documentation for GPIO Zero shows you how to connect up sensors, LEDs, motors, and more. With a bag of cheap components, you can start building your first circuits.

**rpf.io/gpiozero**

## Conquer the Command Line

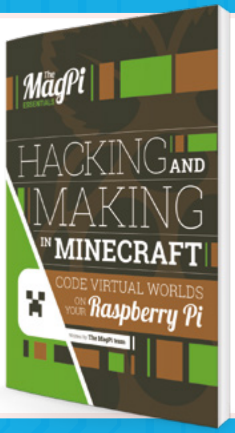
We publish a series of short books called *The MagPi Essentials*, and this one explains the Terminal, including using it for connecting disks, compiling software, and backing up. You can download all the books in the series, and past issues of *The MagPi*, at **magpi.cc/issues**.

**magpi.cc/CLIbook**

## Hacking and Making in Minecraft

Minecraft on Raspberry Pi is a great way into the world of coding. This book, another in *The MagPi Essentials* series, shows you how to shape the world with Python code, use a connected LED as a treasure sensor, and use Sonic Pi and Node-RED together with Minecraft.

**magpi.cc/MCbook**


The MagPi ESSENTIALS
HACKING AND MAKING IN MINECRAFT
CODE VIRTUAL WORLDS ON YOUR Raspberry Pi

## Python Games

Go behind the scenes of the Python games that come pre-installed in Raspbian with Al Sweigart's free online book. It explains the program code for seven of the games in detail, and lets you study another four listings yourself.

**magpi.cc/RTfkvD**

# LEARN TO CODE

## WITH SCRATCH & PYTHON

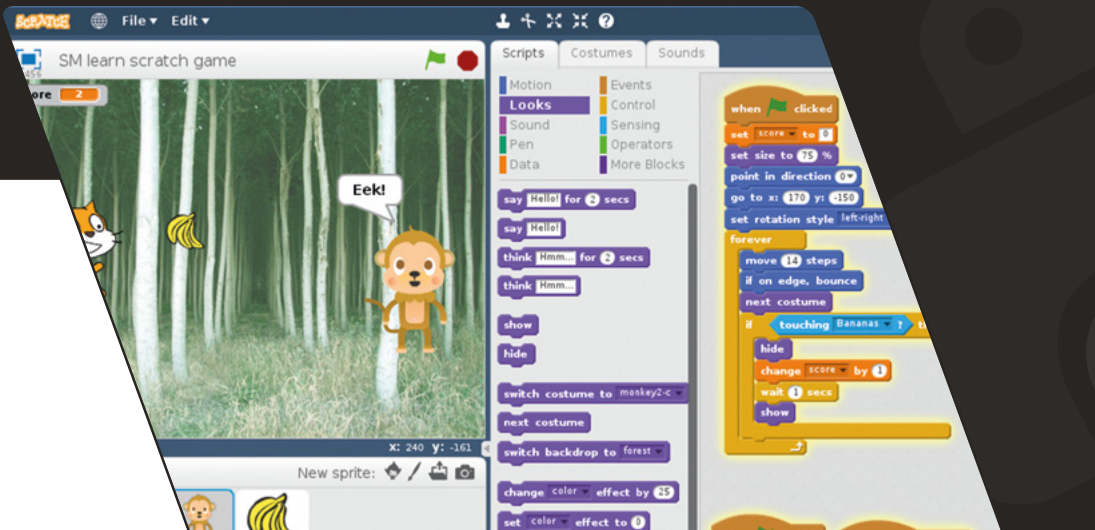
Anybody can learn to code! Programming a computer is much easier than you think. Come with us and we'll help you get started

AUTHOR

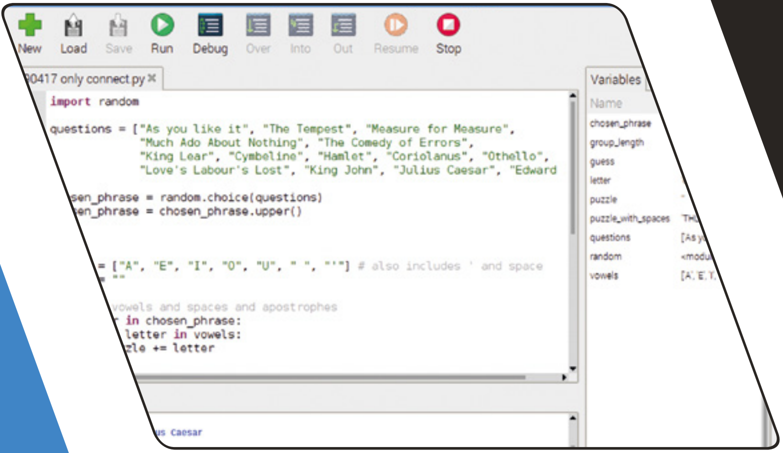
Sean McManus

Author/co-author of inspiring coding books including *Mission Python*, *Cool Scratch Projects in Easy Steps*, and *Raspberry Pi For Dummies*. Get free chapters at Sean's website.

**sean.co.uk**



```
when [flag] clicked
set score to 0
set size to 75 %
point in direction 0
go to x: 170 y: -150
set rotation style left-right
forever
    move 14 steps
    if on edge, bounce
    next costume
    if touching Bananas ? th
        hide
        change score by 1
        wait 1 secs
        show
```



```
import random

questions = ["As you like it", "The Tempest", "Measure for Measure",
             "Much Ado About Nothing", "The Comedy of Errors",
             "King Lear", "Cymbeline", "Hamlet", "Coriolanus", "Othello",
             "Love's Labour's Lost", "King John", "Julius Caesar", "Edward

sen_phrase = random.choice(questions)
en_phrase = chosen_phrase.upper()



= ["A", "E", "I", "O", "U", " ", "'"] # also includes ' and space
= ""

vowels and spaces and apostrophes
 in chosen_phrase:
 letter in vowels:
zle += letter
```

```
us Caesar
```

```
chosen_phrase
group_length
guess
letter
puzzle
puzzle_with_spaces   TH
questions            [As y
random               <modu
vowels               [A', E', 'I',
```

**Learning to code can be one of the most profound skills you will ever develop.** With code, you can control a computer. You can get it to do things for you, and also control gizmos and other computers. Kick back and let your computer do all the work.

Sure, that's cool. But coding is about more than that. It's about understanding how computers work, and getting a better understanding of how technology – and the modern world – works. It's about breaking down problems into little bits and solving them. It's an amazingly helpful life skill. That's why it's profound.

On a more practical level, knowing just a little code can lead to better job opportunities; a little more can open up well-paid and fun jobs. It's an impressive skill to put on your CV and anybody can do it. Anybody.

Coding is a lot easier than you think. And putting the power of computing and digital making into the hands of people is what Raspberry Pi is all about.

Raspberry Pi is 'the little computer that could', and you're 'the person who can'. Don't worry: you've got this. We can help you get started.

> Raspberry Pi is 'the little computer that could', **and you're 'the person who can'**

# Start coding with Scratch

Beginners, arise! It's time to take your first steps with coding, as we introduce you to Scratch and Python, with a sprinkling of twinkling LEDs! By **Sean McManus**

**Being able to write programs is like a superpower: it means you can get your computer to do whatever you want.** Join us as we show you how to make your first programs using Scratch and Python. You'll also see how easy it is to build simple electronics projects.

A program is just a set of instructions. In Scratch, the instructions are written with visual blocks that lock together to make a sequence called a script. The blocks are colour-coded to help you find them. To find the brown blocks, for example, click the brown Events button above the Blocks Palette.

Scratch makes coding easier, because you don't need to worry about the spelling of commands. And everything is laid out in front of you.

## Understanding coordinates

The Scratch screen is divided into units called steps. When the cat moves 10 steps, it only makes one movement, but that stride shifts it 10 positions across the Stage. The middle of the Stage is at x=0, y=0. The x-axis (left to right) runs from -240 to +240, and the y-axis (bottom to top) runs from -180 to +180. The directions the sprite can move in are numbered 0 (up), 90 (right), 180 or -180 (down), and -90 (left). You can use numbers in between those numbers too, so -45 would be a north-west direction. Why not try starting a new project and joining some Motion blocks together to experiment? You can run a script or a block by clicking it, or use the **when flag clicked** block as we did in our program here.

### You'll Need

- Scratch 2
- Raspbian with Desktop and Recommended Software
- A good sense of timing!

## 01 Start Scratch 2

Open Scratch by clicking on the Raspberry Pi Menu icon and choosing Programming > Scratch 2. You will see Scratch interface and a single character in the top-left, known as 'Scratch Cat'.

To control the Scratch Cat, we're going to drag blocks from the Blocks Palette into the Scripts Area and join them together.

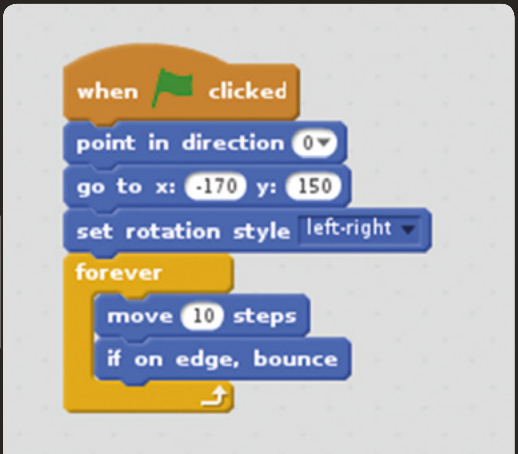
Start by clicking on Events and drag the when ⚑ clicked block to the Scripts Area.

Now click on Motion and drag a **point in direction 90** block and connect it the to the bottom of the when ⚑ clicked block.

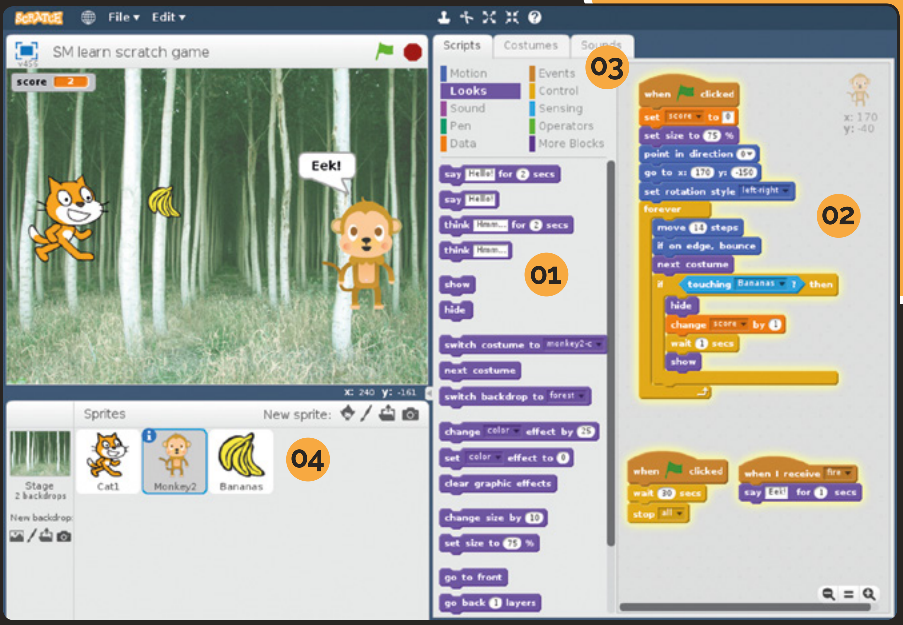
Click on the fields in the blocks to edit the numbers. Click on '90' and change it to '0'.

Now click and drag the blocks below, and edit their numbers, to build a script for Scratch Cat. This script runs when you click the green flag above the Stage. It sets the cat's movement direction to up, puts the cat in the top-left corner of the Stage, and sets it to always face left or right. Then, the movement blocks inside the forever bracket keep the cat moving all the time.

Click the green flag to run your script. Scratch Cat will move to the left side and bounce up and down.



```
when ⚑ clicked
point in direction 0▾
go to x: -170 y: 150
set rotation style left-right▾
forever
    move 10 steps
    if on edge, bounce
```



```
when green flag clicked
set score to 0
set size to 75 %
point in direction 0
go to x: 170 y: -150
set rotation style left-right
forever
    move 14 steps
    if on edge, bounce
    next costume
    if touching Bananas ? then
        hide
        change score by 1
        wait 1 secs
        show

when green flag clicked
wait 30 secs
stop all

when I receive fire
say Eek! for 1 secs
```

**01** **Blocks Palette:** Find instruction blocks here

**02** **Scripts Area:** Drag and drop blocks here to build your script (program)

**03** **Buttons:** Click to view different types of blocks in the Blocks Palette below

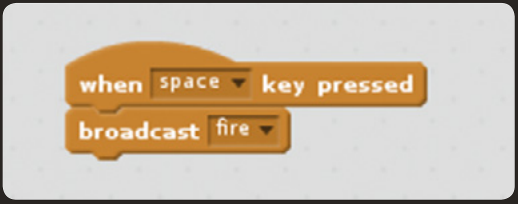
**04** **Sprite List:** Select and manage sprites here

## 02 Send a broadcast

The moving objects in Scratch, including the cat, are called sprites. One sprite can send a message to all the other sprites using a broadcast. You can't hear it or see it on screen, but sprites can listen for it, and then start a script when they receive it. We'll use a broadcast to make the cat throw some bananas. Click the brown Events button, and add the two blocks below to the Scripts Area. This new script doesn't join to the existing script (from Step 1) – it sits on its own in the same Scripts Area.

You need to change 'message1' to 'fire'. Click the down arrow next to 'message1' in the broadcast block and choose New Message. Enter the message name 'fire' and click OK.

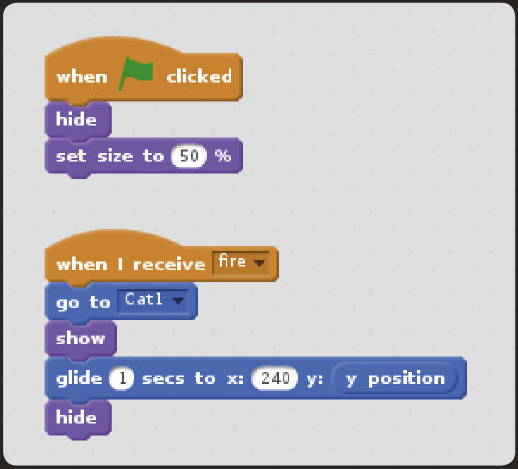
As the program is running, when you tap **SPACE**, the broadcast message is sent silently. Crafty!



```
when space key pressed
broadcast fire
```

## 03 Add aerodynamic bananas

Scratch enables you to get results fast because it includes its own images and sounds. Click the first New Sprite button above the Sprite List (it will display 'Choose sprite from library' as you hover over it). Click the Bananas sprite to add it to the Sprite List. Notice that its Scripts Area is blank. We'll give Bananas two scripts. The first one sets the sprite's size and makes it invisible when you click the green flag. The second runs when the cat broadcasts its fire message. Click and drag the blocks below to the Scripts Area.



```
when green flag clicked
hide
set size to 50 %

when I receive fire
go to Cat1
show
glide 1 secs to x: 240 y: y position
hide
```

When adding the glide block, drag the y position block inside the round 'y:' field to replace the '10' default value.

The 'fire' broadcast makes the bananas jump to the cat (use the go to mouse-pointer block, and choose Cat1 in its menu). Then it makes the bananas visible, glides them across the screen, and hides them again.

Click the green flag, and tap **SPACE** to test. Scratch Cat now throws bananas.

## Top Tip

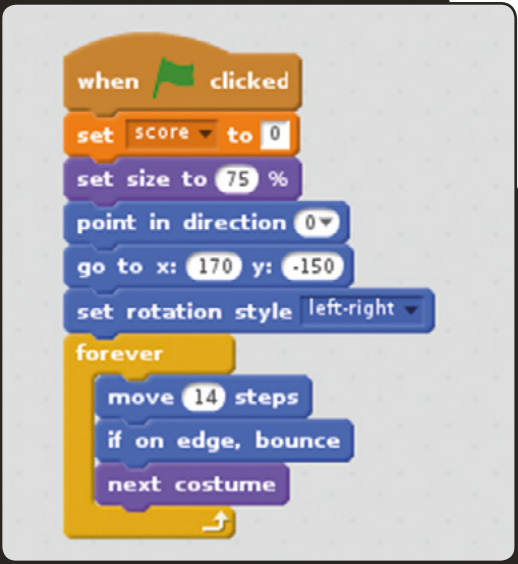
### Get the right sprite

Make sure you're adding scripts to the correct sprite. You can select a sprite by clicking it in the Sprite List.

## Top Tip

### Blocks inside blocks

Some blocks can go inside other blocks. In Step 3, the **y position** block keeps the sprite's y position the same while its x position changes.



```
when [flag] clicked
set score to 0
set size to 75 %
point in direction 0
go to x: 170 y: -150
set rotation style left-right
forever
    move 14 steps
    if on edge, bounce
    next costume
```

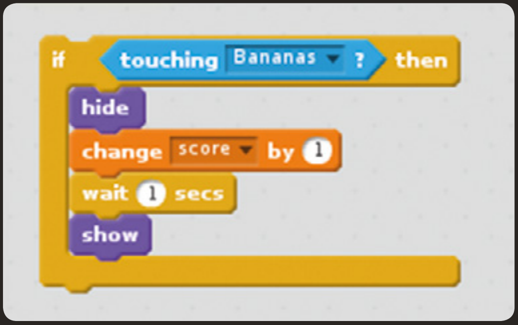
## 04 Add monkey magic

Variables are names used to remember information, such as a score, that might change. Click the Data button and click Make a Variable. Give it the name 'score', select For All Sprites, and click OK. Let's add a moving target for the cat to try to hit. Click the 'Choose sprite from library' icon and choose the Monkey2. Select Monkey2 in the Sprites area and give it the script above.

The monkey's script sets the new score variable to 0 when the game starts, then makes the monkey move up and down. Each sprite can have more than one image: the Next Costume block cycles through them, creating an animation.

## 05 Add collision detection

Our monkey will react when it's struck by a bunch of bananas. To do that, we use an if block; this checks whether something is true – in this case, whether the monkey is touching the bananas. If so, the blocks inside its bracket are run. Here, those blocks hide the monkey, add 1 to the score, and wait one second before showing the monkey again.



```
if touching Bananas ? then
    hide
    change score by 1
    wait 1 secs
    show
```

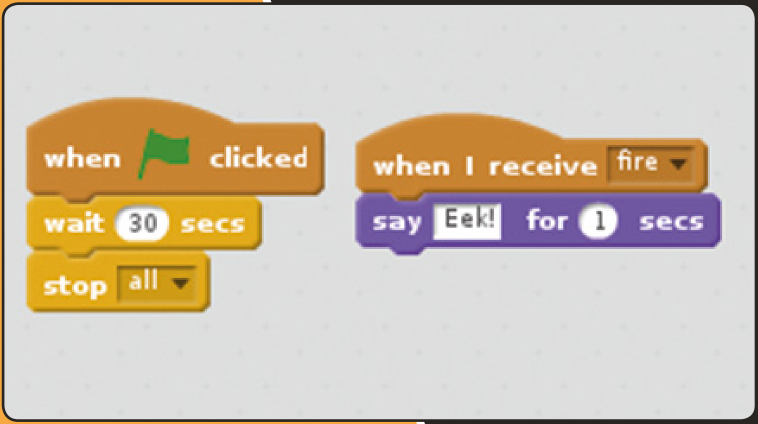
This whole code chunk goes inside the monkey's forever bracket – below the next costume block – so the program keeps checking whether the monkey has been hit.

## Where next?

We're huge fans of Scratch at *The MagPi*, so check out our past issues online for more Scratch tutorials. Issue 5 includes a memory game, like Simple Brian. Issue 34 has a multiple-choice quiz, and our 2018 Annual included an introduction to electronics and Scratch. See issue 76 for a roundup of resources to help you learn Scratch, and don't forget there's a Scratch book in *The MagPi*'s own Essentials series (**magpi.cc/learnscratch**) and the *Code Club Book of Scratch* (**magpi.cc/ccbook1**).

## 06 Finishing touches

Let's add a simple timer to stop the sprites moving after 30 seconds, and make the monkey react when it sees incoming bananas. Add these two scripts to the monkey sprite. Now the game is complete, why not try experimenting with it? Can you make the monkey move erratically instead of disappearing when it's hit? Can you change the sprites' positions and directions to turn the game sideways, making it more like Space Invaders? What about adding more targets to hit? One of the best ways to learn to code is by experimenting with existing programs.



```
when [flag] clicked
wait 30 secs
stop all
```

```
when I receive fire
say Eek! for 1 secs
```

▼ These last two scripts add a timer and make the monkey react when it sees bananas

# Code a quiz game with Python

Make your own text quiz game that mangles famous phrases using the Python language

Many people progress from Scratch to Python, a programming language that is powerful, easy to get started with, and much easier to read and write than other languages.

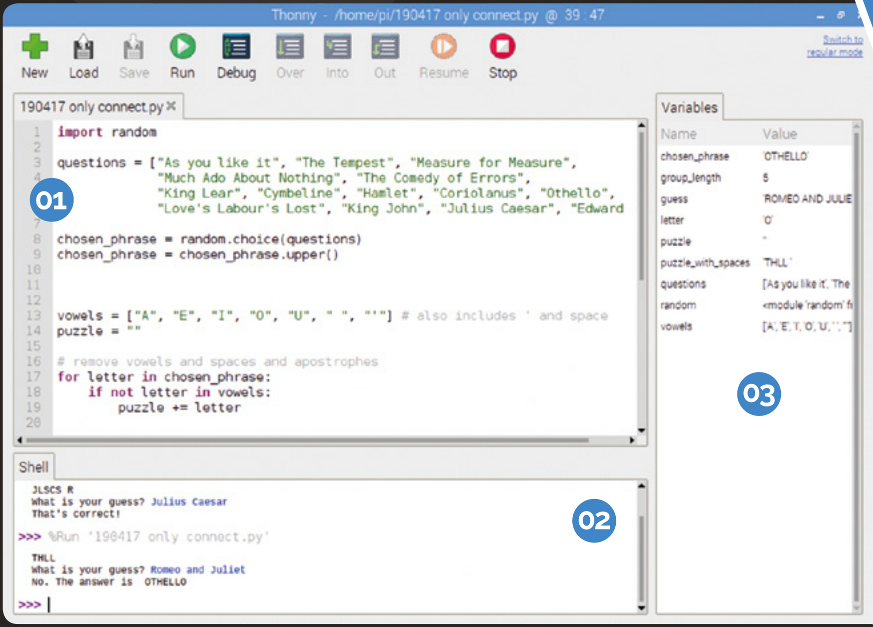
We're going to make a simple quiz question generator that strips the vowels and shuffles the spaces in a phrase. The player has to work out what that phrase is.

We'll be using Thonny, which provides a friendly single-screen environment for running and testing Python code. Like Scratch, the Thonny IDE (integrated development environment) comes pre-installed in the Raspbian with Desktop and Recommended Software operating system.

## 01 Create a list of questions

As well as variables, Python has lists, which can store multiple pieces of information. Our program creates a list called `questions`. Each item in the list is a piece of text, known as a string. In Python, strings are surrounded by double quotes to show where they start and end. The whole list is enclosed in square brackets, and there are commas between the list items. Type in the code below, save your program, and then click Run. If it worked, you should see no error messages in the Shell window.

```
import random

questions = ["As You Like It",
    "The Tempest", "Measure for Measure",
    "Much Ado About Nothing",
    "The Comedy of Errors",
    "King Lear", "Cymbeline",
    "Hamlet", "Coriolanus", "Othello",
    "Love's Labour's Lost",
    "King John", "Julius Caesar",
    "Edward III"]
```



```
import random

questions = ["As you like it", "The Tempest", "Measure for Measure",
             "Much Ado About Nothing", "The Comedy of Errors",
             "King Lear", "Cymbeline", "Hamlet", "Coriolanus", "Othello",
             "Love's Labour's Lost", "King John", "Julius Caesar", "Edward

chosen_phrase = random.choice(questions)
chosen_phrase = chosen_phrase.upper()



vowels = ["A", "E", "I", "O", "U", " ", "'"] # also includes ' and space
puzzle = ""

# remove vowels and spaces and apostrophes
for letter in chosen_phrase:
    if not letter in vowels:
        puzzle += letter
```

| Name | Value |
|---|---|
| chosen_phrase | 'OTHELLO' |
| group_length | 5 |
| guess | 'ROMEO AND JULIE |
| letter | 'O' |
| puzzle | '' |
| puzzle_with_spaces | 'THLL ' |
| questions | ['As you like it', 'The |
| random | <module 'random' fr |
| vowels | ['A', 'E', 'I', 'O', 'U', ' ', '''] |

```
  JLSCS R
  What is your guess? Julius Caesar
  That's correct!

>>> %Run '190417 only connect.py'

  THLL
  What is your guess? Romeo and Juliet
  No. The answer is  OTHELLO

>>> |
```

01 Type in and edit your program code here

02 Enter direct commands and see program input in the Shell here

03 Keep track of the data your program is processing here

### Getting indentation right

Python uses indentation to show which instructions belong to a function, an if statement, or a repeating section. As you can see in Step 3 (overleaf), you can have multiple levels of indentation. The last line belongs to the `if` instruction, and that is repeated inside the `for` loop. The best way to get the indentation right is to remember the colon at the end of the previous line. Then, Thonny will add the indentation for you automatically. If you forget, use four spaces at the start of the line to insert the indentation. You'll still need to fix that missing colon, though!

### You'll Need

- Raspbian with Desktop and Recommended Software
- Thonny

### Top Tip

**Pesky punctuation!**

Take care to add the colons at the end of the `if` and `else` instructions. The code won't work without them.

## 02 Pick a random question

Python includes modules of prewritten code you can use, such as the `random` module we imported in Step 1. The first new instruction creates a new variable called `chosen_phrase` and puts a randomly chosen question into it. The second line converts the `chosen_phrase` to upper case. Run the program a few times and look at the value of `chosen_phrase` in the Variables pane. You should see different names come up, although names can also repeat.

Add a line of space between the code in Step 1 and add the following code:

```
chosen_phrase = random.choice(questions)
chosen_phrase = chosen_phrase.upper()
```

## 03 Strip the vowels and spaces

Let's create a new list of forbidden characters, chiefly the vowels, but also the space and the apostrophe. That last list item in the `vowels` list is an apostrophe inside double quotes. We create an empty string variable, called `puzzle`. We're going to go through each letter in the phrase, check whether it's in the `vowels` list, and if not, add it to the end of the `puzzle` string. The `for` instruction sets up a repeating piece of code, called a loop. The instructions that should be repeated are indented from left. Each time around the loop, the variable `letter` is set to contain the next character from the `chosen_phrase` string. The `if` instruction checks whether the letter is in the `vowels` list. If it is not, the `letter` is added to the end of `puzzle`. The `+=` means 'add at the end'. Run the program, then test it's working by looking at the contents of `puzzle` in the Variables panel. It should contain no vowels, spaces, or apostrophes.

Add the following code to the program:

```
vowels = ["A", "E", "I", "O", "U", " ", "'"]
puzzle = ""

for letter in chosen_phrase:
    if not letter in vowels:
        puzzle += letter
```

## 04 Insert random spaces

Each character in the string can be referred to by its position number, starting at 0. The number is called an index, and you put it in square brackets after the string. Try this in the Shell (click on the line starting with '>>>'). Instructions in the Shell are carried out immediately. Enter the following:

```
print("Hello"[1])
```

You get 'e' back (because the first character is number 0). You can get a chunk too (called a slice) by giving a start and end index, like this:

```
print("Hello"[1:4])
```

It gives you 'ell' because the last index position (4) is left out. We'll create a new list, called `puzzle_with_spaces`, by adding chunks of the `puzzle` string and a space until there's no `puzzle` string left. The `while` loop repeats the indented instructions below as long as the length of `puzzle` is more than 0. The `group_length` variable is given a random whole number (integer) from 1 to 5. Then that many letters are added to `puzzle_with_spaces` from the front of `puzzle`, plus a space. Those characters are

## Where next?

You can find Python code to dissect in most issues of *The MagPi*. Issue 53 (**magpi.cc/53**) includes a more in-depth beginner's guide to Python, covering variables, looping with `while` and `for`, branching with `if`, and functions, which we'll cover shortly in this guide. Issue 54 (**magpi.cc/53**) introduces object-oriented programming in both Scratch and Python. Issue 73 (**magpi.cc/73**) includes a roundup of Python books and online resources. There is a book in our Essentials series too, called *Make Games with Python* (**magpi.cc/gameswithpython**).

## Debugging in Thonny

You can step through the program slowly to see what it's doing, which can help you to find errors. Click the Debug button in Thonny, then click the Over button to run through each instruction in turn. Watch the Variables pane on the right to see how the lists and strings change at each stage of the program. Thonny also helps you avoid errors by highlighting unclosed brackets and double quotes.

> “ Python is **easy to get started with** and much easier to read and write than other languages ”

then cut off the front of `puzzle`. The slicing here only uses one number, so the other one is assumed to be the start or end of the string.

Add this code to your program:

```
 puzzle_with_spaces = ""

 while len(puzzle) > 0:
     group_length = random.randint(1,5)
     puzzle_with_spaces +=
puzzle[:group_length] + " "
     puzzle = puzzle[group_length:]
```

## 05 Add collision detection

It prints the `puzzle_with_spaces`. It then uses the `input()` function to ask you what your guess is. Your answer goes into the `guess` variable, and is then converted to upper case to make sure it matches the correct answer if it's right. The `if` instruction checks whether `guess` is the same as `chosen_phrase`. If so, it prints one message. Otherwise, the instruction indented under `else` runs, to tell you the right answer. In Python, one `=` is used to put a value into a variable, but two (`==`) are used to compare items in an `if` instruction.

Add this code to the end of the program:

```
print(puzzle_with_spaces)
guess = input("What is your guess? ")
guess = guess.upper()

if guess == chosen_phrase:
    print("That's correct!")
else:
    print("No. The answer is ", chosen_phrase)
```

Click the Run button and hopefully you'll see some letters in the Shell and 'What is your guess?' Enter an answer and you'll see 'That's correct!' or 'No. The answer is' and the correct response.

If you've typed the code out by hand, it's likely that you'll see an error message. Go through your code line-by-line and compare it to the full code in **quiz_game.py**.

## quiz_game.py

▶ Language: **Python**

DOWNLOAD THE FULL CODE:

magpi.cc/github82

```
import random

questions = ["As You Like It", "The Tempest",
"Measure for Measure", "Much Ado About Nothing",
"The Comedy of Errors", "King Lear", "Cymbeline",
"Hamlet", "Coriolanus", "Othello","Love's Labour's Lost",
"King John", "Julius Caesar", "Edward III"]

chosen_phrase = random.choice(questions)
chosen_phrase = chosen_phrase.upper()

vowels = ["A", "E", "I", "O", "U", " ", "'"]
puzzle = ""


for letter in chosen_phrase:
    if not letter in vowels:
        puzzle += letter


puzzle_with_spaces = ""


while len(puzzle) > 0:
    group_length = random.randint(1,5)
    puzzle_with_spaces += puzzle[:group_length] + " "
    puzzle = puzzle[group_length:]


print(puzzle_with_spaces)
guess = input("What is your guess? ")
guess = guess.upper()


if guess == chosen_phrase:
    print("That's correct!")
else:
    print("No. The answer is ", chosen_phrase)
```

# Build an LED torch and electronic game

Discover how easy it can be to get lights blinking and buttons clicking using GPIO Zero and use your new-found skills to build an electronic game

**One of the best things about the Raspberry Pi is that you can easily hook up your own electronics projects.** Using some electronics components and the GPIO Zero library, you can program a puzzle game where you have to repeat a sequence of lights that gets longer each turn. You might remember a similar electronic game from your childhood, but we call ours Simple Brian. In *The MagPi* issue 77 (**magpi.cc/77**) we showed how to use Python code to play the game on screen. Here, we'll show you how to make the electronic game itself, building on your new-found Python skills from Missing Vowels.

First, we're going to show you how to build a torch by lighting up LEDs. Let's get going.

## 01 Connect your first button

The torch circuit diagram (**Figure 1**) shows an LED light connected to the GPIO pins of a Raspberry Pi using a breadboard (see **magpi.cc/breadboard** for a primer on using this piece of equipment). Press the button into the board, then use jumper wires to form a circuit with your Pi, as shown by the yellow wires in the diagram. The first button connects to the GPIO 2 pin on one side, and to the ground rail on the other side. We'll connect the latter to a ground pin on the Pi, so anything plugged in that row of holes connects to ground.

## 02 Connect your first LED

Always use a resistor when you connect an LED to your Pi, to prevent the LED drawing too much current and getting damaged. Both the LED and the resistor plug straight into your breadboard. The current flows from GPIO 18, through the resistor, through the LED (lighting it up), to the breadboard's ground rail. LEDs only work one way around: the short leg is the negative side, which you connect to ground. The LED won't light up yet.

## 03 Make an LED torch

You've made your first circuit! Let's test it by coding a torch. The **torch.py** code shows how to use an LED and a button. It imports the relevant parts of the GPIO Zero library, then sets up an LED called `light`, connected to GPIO pin 18. The button on pin 2 is set up with the name `button`. The `while True` loop checks whether the button is pressed forever. If so, the light is turned on. Otherwise, it's turned off. Pay attention to the capitalisation of `LED` and `Button` when setting them up.

## 04 Add the other buttons and LEDs

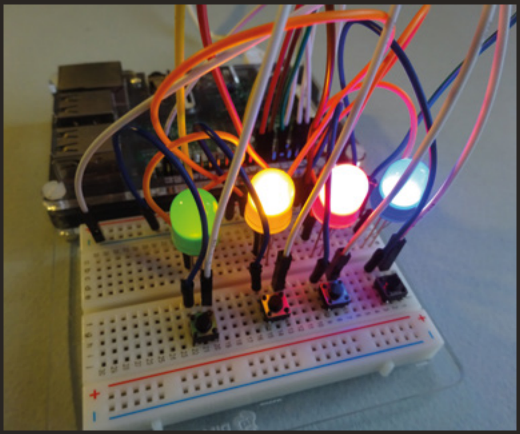
Take a look at **Figure 2** (overleaf). It looks complex at first, but the other three buttons and LEDs are connected in the same way as

### You'll Need

- 4 × LEDs (ideally different colours)
- 4 × 330 Ω resistors
- 4 × 6 mm Tactile momentary button switches
- 400-point breadboard
- 9 × Male-to-female jumper cables
- 8 × Male-to-male jumper cables
- PiBow case with Breadboard Base **pimoroni.com**

### Where next?

The online documentation for GPIO Zero (**magpi.cc/DPyuYc**) provides more code examples, including a button-controlled camera, an LED bar graph, and a motion sensor. We surveyed useful resources for basic electronics in issue 77, and there's a book in our Essentials series called *Simple Electronics with GPIO Zero* (**magpi.cc/gpio-zero**).



The finished game, with all the lights lit up for testing

the first ones, just using different pins on the Pi. All the buttons (yellow wires) connect to the Pi's inner row of pins, and the LEDs (blue wires) to the outermost row. We've separated the components in this diagram a bit so it's easier to see how to wire it up, but try to line up your LEDs and buttons on the breadboard so it's easier to play the game.

## 05 Test them all

Now you can test these LEDs and buttons too. Modify your torch code to use `LED(23)` and `Button(3)` and then run the program to test the next light switch works. Then check `LED(24)` and `Button(4)`, and finally `LED(25)` and `Button(17)`. The buttons should be next to the LED they illuminate.

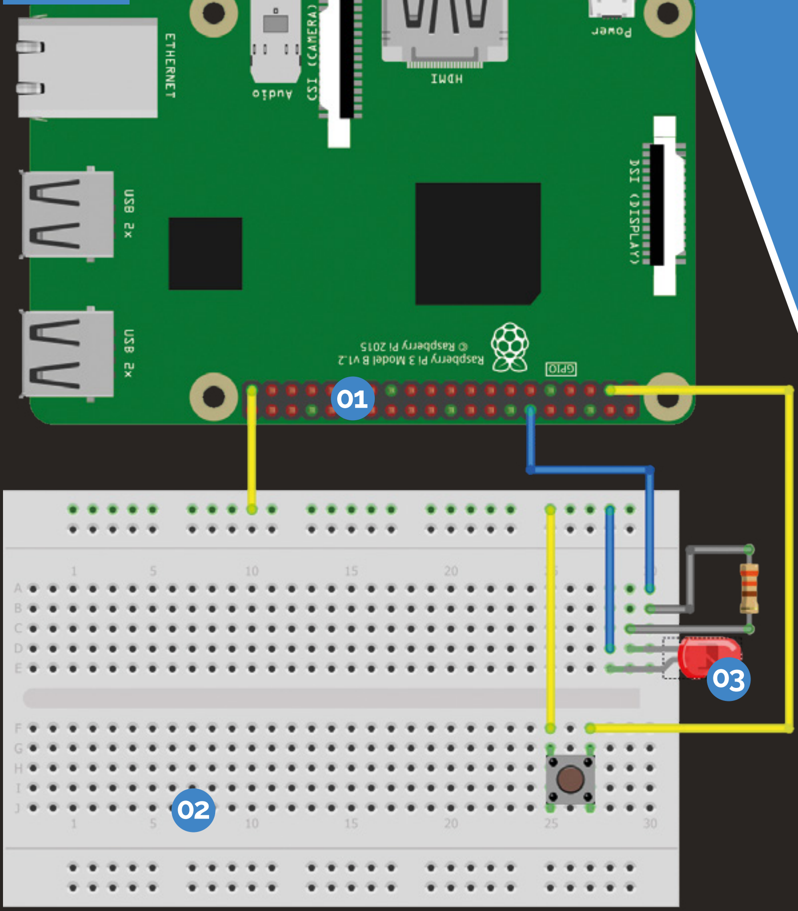
## 06 Build an electronic game

Now we're ready to start making the Simple Brian game (see **brian.py**). This starts by setting up a list of buttons, and a list of their associated LEDs. It also creates an empty list called `sequence`, which we'll use for the sequence of lights the player must repeat. With each turn, it'll get longer.

## 07 Add functions

Functions enable you to bundle up a set of instructions so you can reuse them. You have to define a function before you can use it. To define a function, you use `def`, followed by the function name, (), and a colon. The brackets are there to hold any info you're sending to the function, but we don't need to send any so they're empty. You can tell which instructions belong to a function, as they're indented. The `lights_on()` and `lights_off()` functions use a loop to go through all the items in

Figure 1



01 Count the pins to see where to connect your wires

02 The breadboard makes it easy to plug in components and quickly set up circuits

03 Using this simple circuit (with an LED and a button), you can make a push-button torch

### torch.py

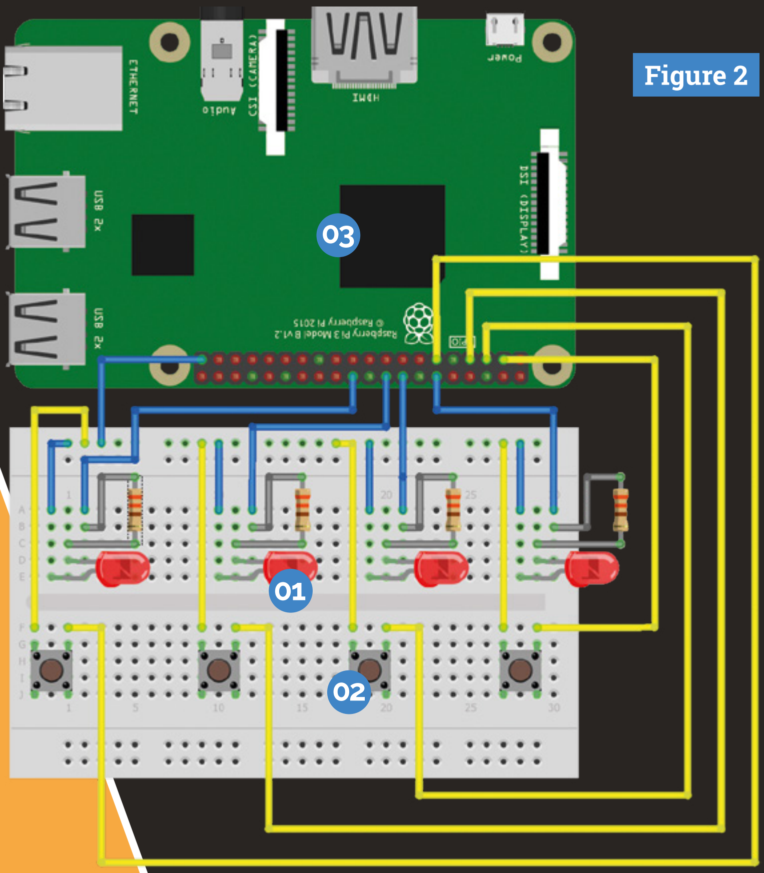
Language: **Python**

DOWNLOAD THE FULL CODE:

**magpi.cc/github82**

```
# Torch demo
from gpiozero import Button, LED

light = LED(18)
button = Button(2)

while True:
    if button.is_pressed:
        light.on()
    else:
        light.off()
```

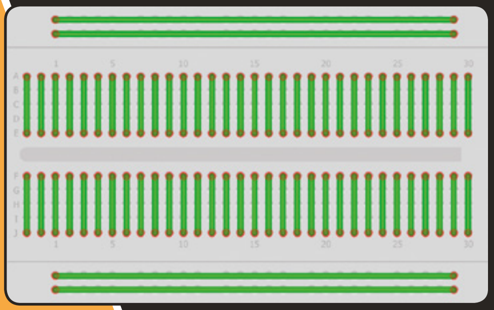
Figure 2



01 Just like its famous namesake, our Simple Brian game features four LED lights that flash up a sequence for you to memorise

02 The circuit also contains four buttons. We'll push these to repeat the sequence of lights that we've memorised

03 At the heart of our project is the Raspberry Pi. The buttons and lights are connected to our computer with wires. Code in the computer flashes the lights and keeps an eye on the buttons we're pushing



▲ On a breadboard, the rails along the edges are connected in a long line, and the short lines of dots in the middle are connected to each other

the `leds` list, putting them into `led`, then turning `led` on or off. The `flash_all()` function shows how to repeat a set number of times, in this case 3. The loop turns the lights on and off, with a 0.25 second pause after each change using `sleep(0.25)`.

## 08 Lights test flash all

After you've entered the functions (down to line 23), you can test the program by adding `flash_all()` as the last line and then running it. All the lights should flash together, three times. Delete that test line before you carry on. In line 25, the program runs the `lights_off()` function to ensure the lights are all off before the game begins.

## 09 Add to the sequence

Now we enter the main game loop, under `while True` (line 27). Everything from here on in is indented to show it belongs to that loop, repeating endlessly. The game sequence starts as an empty list, so the first thing we do is to add an LED. We pick a random LED using `random.choice()` and add it to the end of the `sequence` list using the `append()` list method. A list method is a built-in Python function that you can apply to a list. Other methods are available to insert and remove items, and sort the list, among other things.

## 10 Play the list sequence

The lights all flash three times using the `flash_all()` function before the sequence begins, to show this is the start of the sequence. Then a loop is used that takes each LED from the `sequence` list, and puts it into `light`, in turn. It's turned on, there's a short pause, then it's turned off. There's another short pause so it's obvious there are multiple flashes of the same light if it repeats in the sequence. In round one, there's only one light in the `sequence` list, but as the game progresses, this loop will get longer. You can run the program at this point to see the light sequence gradually extend, without the player getting a chance to guess.

## 11 Get the player's guess

Getting the player's guess uses a similar loop to the one that plays the lights sequence. It

## brian.py

Language: **Python**

**DOWNLOAD THE FULL CODE:**

magpi.cc/github82

```
001. from gpiozero import Button, LED
002. from time import sleep
003. import random
004.
005. buttons = [Button(2), Button(3), Button(4),
     Button(17)]
006. leds = [LED(18), LED(23), LED(24), LED(25)]
007.
008. sequence = []
009.
010. def lights_on():
011.     for led in leds:
012.         led.on()
013.
014. def lights_off():
015.     for led in leds:
016.         led.off()
017.
018. def flash_all():
019.     for _ in range(3):
020.         lights_on()
021.         sleep(0.25)
022.         lights_off()
023.         sleep(0.25)
024.
025. lights_off()
026.
027. while True:
028. # Add a new light to the end of the sequence
029.     new_light = random.choice(leds)
030.     sequence.append(new_light)
031.
032. # Flash all before playback
033.     flash_all()
034.
035. # play the sequence
036.     for light in sequence:
037.         light.on()
038.         sleep(0.5)
039.         light.off()
040.         sleep(0.25)
041.
042. # get the player's input
043.     for light in sequence:
044.         guess = None
045.         while guess == None:
046.             for button in buttons:
047.                 if button.is_pressed:
048.                     # convert button push to list
     index number
049.                     guess = buttons.index(button)
050.
051.         if leds[guess] == light:
052.             light.on()
053.             sleep(0.5)
054.             light.off()
055.             sleep(0.25)
056.         else:
057.             print("You failed at level ",
     len(sequence))
058.             for _ in range(10):
059.                 light.on()
060.                 sleep(0.15)
061.                 light.off()
062.                 sleep(0.15)
063.             sequence = []
064.             break
```

works its way through the `sequence` list, accepts a guess, and checks whether it matches the current item in the sequence. There are three loops inside each other here. The program sets the `guess` variable to `None`, a special value in Python. Then a `while` loop keeps repeating until the `guess` variable changes. Inside that, a loop goes through the `buttons` list, checking each one in turn to see whether it's pressed. If so, the `guess` variable is changed, ending the `while` loop. The program converts the button the player pressed into its index number in the list and puts that into the `guess` variable. That way, we can match the button to its LED, which will be at the same position in the `leds` list. You can find the position of an item in a list using `listname.index(item)`.

### 12 Check the player's guess

We're still inside the loop going through the sequence here, as the indentation of line 51 shows. Now we check whether the light the player guessed (`leds[guess]`) matches the current light in the sequence. If so, the light is turned on and then off again. If the two lights don't match, the player made a mistake. We can tell how long a list is using `len(listname)`. We use the length of the `sequence` list to tell the player which level they got to. The correct light is then flashed quickly ten times. The `sequence` list is emptied to start a new game, and the `break` instruction breaks out of the `for` loop that's getting player input. When the player has either guessed all the lights, or failed, the game repeats from line 28, adding a new light to the sequence.

### Top Tip

**Underscoring repetition**

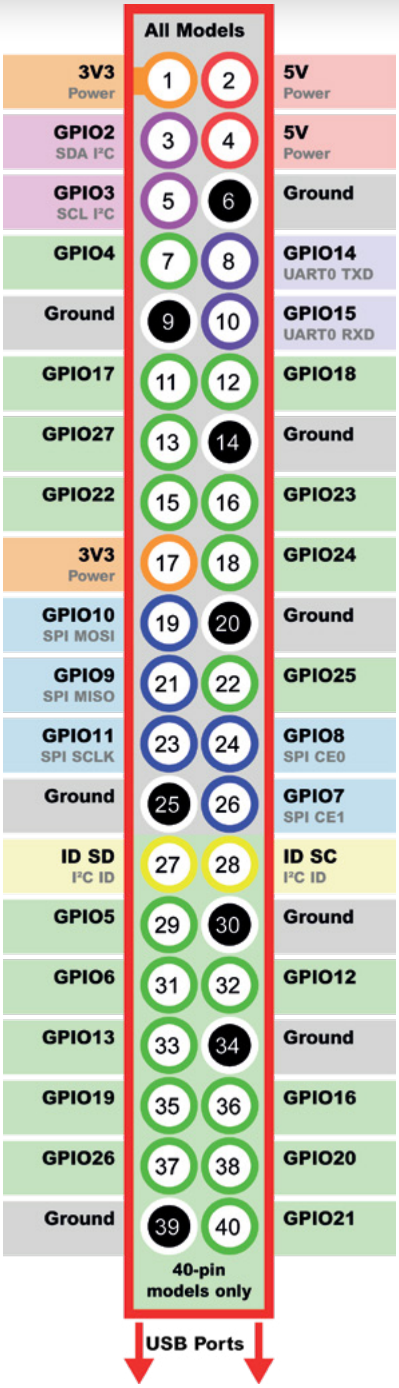
The line `for _ in range(3)` repeats the indented instructions three times. The `_` shows we don't need to use the loop number. Often, you'd use a variable name instead.

# BEGINNER'S GUIDE TO GPIO ZERO

Discover the easy way to control GPIO pins on your Raspberry Pi

## GPIO Zero Layout Guide

GPIO Zero uses the BCM number system to identify pins. These are the numbers referenced on the outside of the diagram, such as 18 for GPIO18. The board numbers (the ones on the inside) are just to help you count the pins. You can use any pin here marked GPIO, but the ones in blue double up with other functions. For this reason, we will stick to the ones marked in green in this tutorial.



All Models

| | | | |
|---|---|---|---|
| 3V3<br>Power | 1 | 2 | 5V<br>Power |
| GPIO2<br>SDA I²C | 3 | 4 | 5V<br>Power |
| GPIO3<br>SCL I²C | 5 | 6 | Ground |
| GPIO4 | 7 | 8 | GPIO14<br>UART0 TXD |
| Ground | 9 | 10 | GPIO15<br>UART0 RXD |
| GPIO17 | 11 | 12 | GPIO18 |
| GPIO27 | 13 | 14 | Ground |
| GPIO22 | 15 | 16 | GPIO23 |
| 3V3<br>Power | 17 | 18 | GPIO24 |
| GPIO10<br>SPI MOSI | 19 | 20 | Ground |
| GPIO9<br>SPI MISO | 21 | 22 | GPIO25 |
| GPIO11<br>SPI SCLK | 23 | 24 | GPIO8<br>SPI CE0 |
| Ground | 25 | 26 | GPIO7<br>SPI CE1 |
| ID SD<br>I²C ID | 27 | 28 | ID SC<br>I²C ID |
| GPIO5 | 29 | 30 | Ground |
| GPIO6 | 31 | 32 | GPIO12 |
| GPIO13 | 33 | 34 | Ground |
| GPIO19 | 35 | 36 | GPIO16 |
| GPIO26 | 37 | 38 | GPIO20 |
| Ground | 39 | 40 | GPIO21 |

40-pin models only

USB Ports

### You'll Need

- Rasbperry Pi
- Breadboard
- LEDs
- Button
- Jumper cables

**The 40 metal pins on your Raspberry Pi board are known as General Purpose Input/Output pins, or GPIO pins for short.** These pins are your connection between the virtual world of computer code and the real world. With GPIO, you can connect circuit components up to your Raspberry Pi.

It's easy to attach LED lights, buttons, buzzers, and all kinds of electronic components to your Raspberry Pi. Typically you connect these components to a breadboard, at least when you are starting out. A breadboard is a plastic prototyping board used to link circuit components together without having to connect them physically. The legs of the components, along with cables, are pushed into adjoining holes on the breadboard (see page 38 for more details).

You then hook the breadboard up to the GPIO pins on a Raspberry Pi (again, using jumper cables). You'll need male-to-male jumper cables to connect breadboard components together, and male-to-female jumper cables to connect the breadboard to the GPIO pins on your Raspberry Pi.

Different GPIO pins have different qualities. Some provide constant power, either at 5 volts or 3.3 volts. Others are ground pins, which should be used to complete a circuit.

Perhaps the most interesting GPIO pins are the ones that can be programmed. These can be turned on and off, powering up components (like LED lights) from code. Alternatively, they can be set to respond to voltage change inputs, such as a button push. Your code can then respond to these inputs.

Typically, you'll program GPIO pins using Python, although Scratch, Java, and a host of other languages can be used to control GPIO.

Programming GPIO in Python used to be quite a detailed task. But a Python library called GPIO Zero simplifies things massively. With GPIO Zero you can quickly connect components and start using them with just a few simple commands.

## 01 Set up a circuit

Programming GPIO pins is a relatively straightforward process, but remember that there are a few steps required to perform the most basic of tasks.

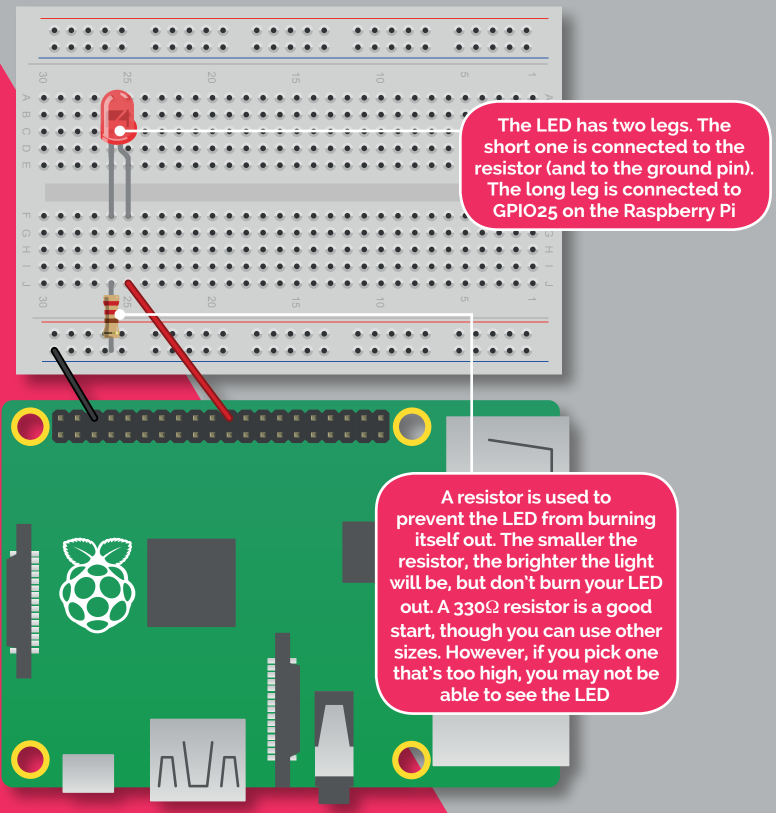
We're going to start by wiring up a single LED to GPIO25 as shown in this diagram. If you don't know how to use a breadboard, take a look back a couple of pages in this magazine. This circuit is extremely similar to the one we used there, except that instead of the long leg of the LED being connected to a live pin, it is connected to a programmable GPIO pin. This makes the LED light up when the GPIO pin is turned on in our program.

Use a female-to-male jumper lead to connect the GND pin to the ground rail on the breadboard.

Insert one end of a resistor into a hole on the ground rail, and the other end into a hole on the breadboard.

Connect the short leg of the LED into a hole on the same line and the longer pin on the line next to it.

Finally, take another male-to-female jumper wire. Place one end in a hole on the same row as the longer LED leg. Connect the other end to GPIO18.



The LED has two legs. The short one is connected to the resistor (and to the ground pin). The long leg is connected to GPIO25 on the Raspberry Pi

A resistor is used to prevent the LED from burning itself out. The smaller the resistor, the brighter the light will be, but don't burn your LED out. A 330Ω resistor is a good start, though you can use other sizes. However, if you pick one that's too high, you may not be able to see the LED

## 02 Regular GPIO

If we were going to light up this LED using regular code rather than GPIO Zero, this is what we would have to write (don't enter this code: it's just an example):

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)
GPIO.output(25,GPIO.HIGH)
```

It's not impossible to decipher. But this code is fussy, and concepts like 'OUT' and 'HIGH' get in the way of understanding the relationship between the code and the light. With GPIO Zero it's a lot easier.

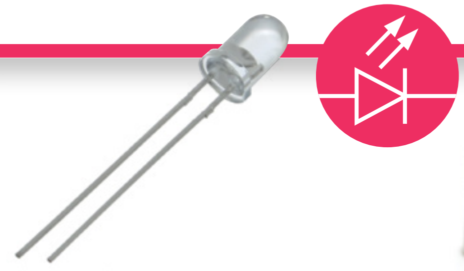
## 03 GPIO Zero

GPIO Zero simplifies things. The same code in GPIO Zero looks like this:

```
from gpiozero import LED
led = LED(25)
led.on()
```

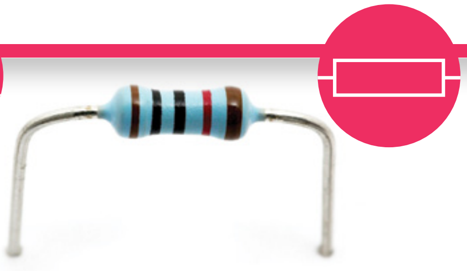
Enter each line of the code above into the Python shell one line at a time. Press **RETURN** after each line. When you enter **`led.on()`**, the LED will light up.

As well as consisting of fewer lines, this program is easier for young students to understand.



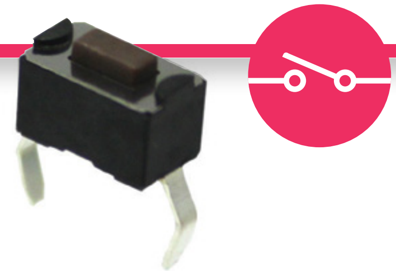
### LED

LEDs are easy to start with. LEDs have one leg longer than the other. They only work one way around.



### RESISTOR

Always use a resistor with LED lights. This prevents them from burning out from high voltage.



### BUTTON SWITCH

A button switch completes a circuit connection when it's pressed. Code can respond to button pushes.

The first line of the program is where we import the **gpiozero** library into Python.

The second line creates an **LED** object, with the pin number as the argument (inside the brackets).

The third line tells the pin to switch on.

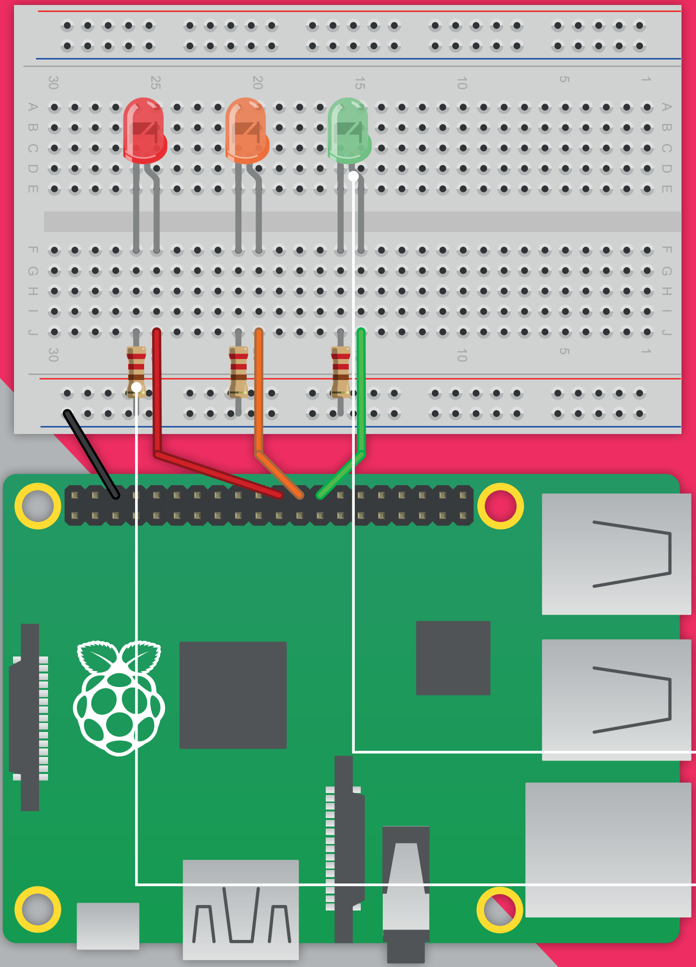
Now enter **led.off()**. The light will switch off.

## 04 Blink

The real joy of GPIO Zero is that it lets you perform rather complex tasks with simple instructions. Creating a blinking LED with more standard code requires you to import the **time** module, set an LED to on, pause for a period, then set the LED to off, and repeat in a loop. This process makes it difficult to perform other code tasks at the same time. In GPIO Zero, however, you simply enter this line:

```
led.blink()
```

The light will start blinking on and off at one-second intervals. Enter **led.off()** to stop it.



The short leg of each LED is connected to a resistor. Each longer LED is connected a separate GPIO pin. These pins are used to turn on, or off, the power to each light

All three resistors are connected to the ground rail. This rail is connected to a single GND pin on the Raspberry Pi

## 05 More control

One neat thing about GPIO Zero commands is that you can enter arguments inside the brackets. Enter **led.blink(** and stop at the open bracket. A yellow box appears showing the text '**on_time=1, off_time=1, n=none, background=True)**'.

These are the parameters available for the **blink** method: the number of seconds for which a light stays on and then stays off, how many times the light blinks, and whether you can add more code while the light is blinking.

After each argument is the default value: one second on, one second off, none (which means the light blinks until you say otherwise), and True (which lets you carry on adding code while the light is flashing).

To set default values, add a number for each value (from left to right):

```
led.blink(4,2)
```

The light comes on for four seconds, and off for two seconds. Enter **led.off()** to stop it. Alternatively, you can add the item and equals sign to pick a value to change (and keep the defaults).

```
led.blink(n=3)
```

The light will blink three times and stop.

## 06 Traffic lights

Let's take our LEDs and build something a little more complex. We're going to add another two LEDs to our circuit (three in total). We'll use one red LED, one amber, and one green.

Connect the new LEDs into the circuit using another two resistors to connect their shorter legs to the ground rail.

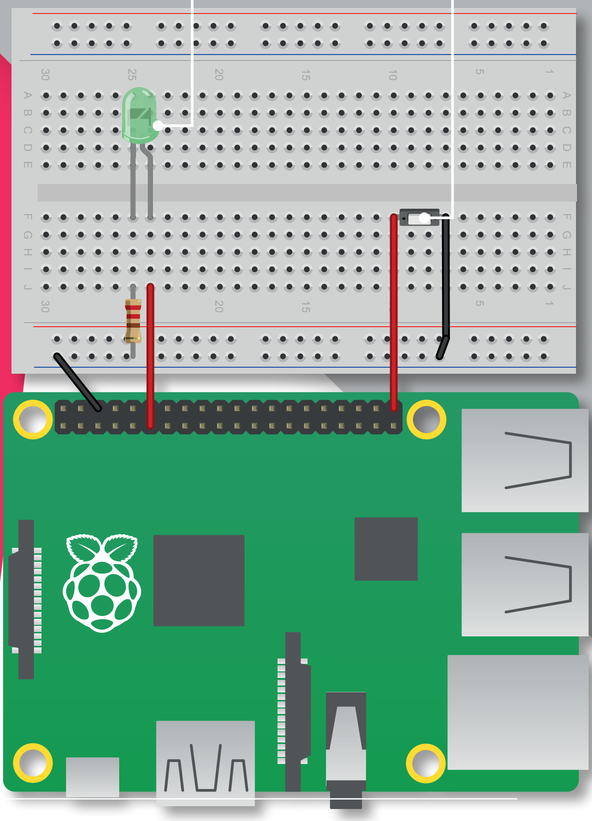
Connect the longer leg of the amber LED to GPIO8 and the longer leg of the green LED to GPIO7. These are the two pins next to GPIO25, so you have your LEDs all together.

Now we need to create a program to control our traffic lights. Open Thonny and choose **File > New**.

Type in the code from **traffic_lights.py**. Save and run it (hit the Play icon or press **F5**). The Python program will run, and you'll see your traffic lights in action.

The LED is wired up to GPIO 17 via a resistor. When GPIO Zero detects a button push, it lights up the LED

One leg of the button is connected to the ground rail. The other leg is connected to GPIO 21



## 07 Adding a button

Now we're going to add a button to our circuit and connect it to GPIO 21 with an LED wired up to GPIO 17. Buttons are wired up in a similar fashion to LEDs, using female-to-male jumper leads.

Like LEDs, buttons have legs. One leg is wired to a GND pin (via the same ground rail you've been using for LEDs); the other leg is connected to a GPIO pin. Some buttons have four legs, so you can connect them to the breadboard with their legs straddling the central groove (see the breadboard diagram).

## 08 Button responder

Unlike LEDs, you don't need to use a resistor with a button. The button's legs are the same length, and it typically doesn't matter which way around you hook it up to the breadboard.

When the button is pushed, it forms a connection between the ground rail and the pin, completing a circuit. The Raspberry Pi is set up, using GPIO Zero, to detect this connection and respond.

Open Thonny and create a new file. Enter the code from **button.py**, save the file, and run the code. When you push the button, the LED will light up.

### button.py

DOWNLOAD THE FULL CODE:

magpi.cc/2ehTqVq

Language: **Python**

```
from gpiozero import LED, Button
from signal import pause

led = LED(17)
button = Button(21)

button.when_pressed = led.on
button.when_released = led.off

pause()
```

### traffic_lights.py

Language: **Python**

```
from gpiozero import LED
from time import sleep

red = LED(25)
amber = LED(8)
green = LED(7)

green.on()
amber.off()
red.off()

while True:
    sleep(10)
    green.off()
    amber.on()
    sleep(1)
    amber.off()
    red.on()
    sleep(10)
    amber.on()
    sleep(1)
    green.on()
    amber.off()
    red.off()
```

## Going further

You can achieve a lot more with GPIO Zero than controlling LEDs and buttons.

For a more detailed look at how you can detect motion, control robots, read sensor information (such as movement sensors or thermometers), we have created a fantastic guide book: *The MagPi Essentials: Simple Electronics with GPIO Zero*. Learn more at **magpi.cc/gpio-zero**.

The MagPi Essentials
SIMPLE ELECTRONICS WITH GPIO ZERO
TAKE CONTROL OF THE REAL WORLD WITH YOUR Raspberry Pi

# HOW TO USE A BREADBOARD

This humble plastic block full of holes can be used to create just about anything

### You'll Need

- Breadboard
- LED light
- Resistor
- Male-to-female jumper leads
- Male-to-male jumper leads

**Most of our projects are tested using a small piece of plastic known as a breadboard.** Officially, it's known as a 'solderless breadboard' because it enables you to use circuit parts without soldering them together.

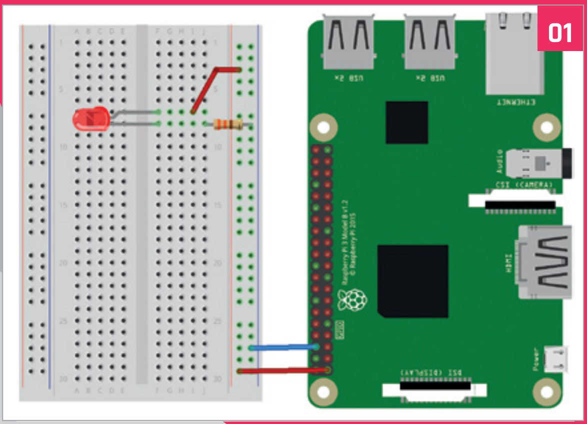
Electrical components are connected by pushing them into the holes in a breadboard. These holes are connected in strips, as shown in the main image. If you push a wire, or a different component, into one hole in a strip, and another wire into the hole next to it, it's as if you'd physically joined (or soldered) the two wires.

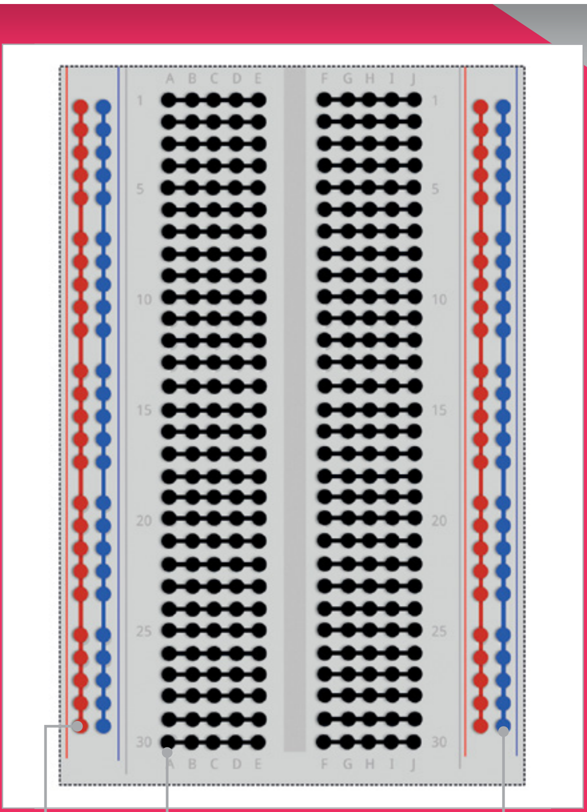
In the old days, people would either solder wire components together on an actual breadboard, or they'd wrap wires together around nails in a pinboard.

For a lot of Raspberry Pi fans, using a breadboard is part of life. But for many newcomers this quirky piece of kit is baffling: a smorgasbord of holes arranged in rows and columns that seem to make little sense.

So we think it's a great idea to read this beginner's guide to how a breadboard works. In this tutorial, we'll explain how these holes are arranged, and how to set up a circuit on your breadboard.

If you already know all this, feel free to move on. If not, stick around and learn about one of the most fun things you can do: building your own circuits and hooking hardware up to your Raspberry Pi.



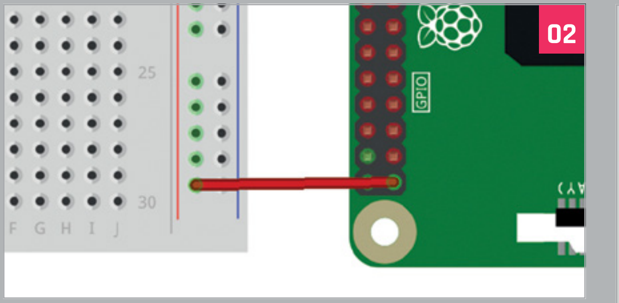


A hole in the blue (sometimes black) rail is connected to a GND pin. It then becomes the ground rail

These horizontal rows are where you place the components that make up your circuit. They are linked into groups of five (or six) holes

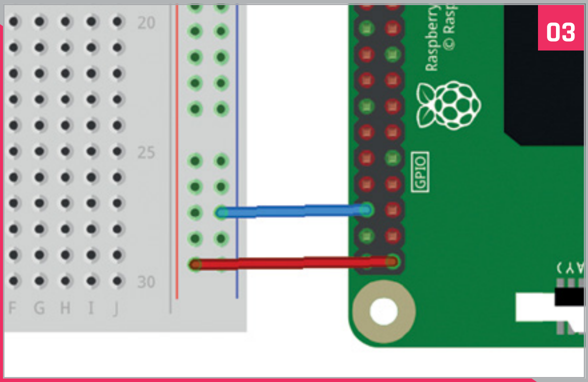
These long vertical strips are known as rails. They are used to provide regular constant power. The red rail is connected to a power pin and becomes the live rail

**01 Fritzing diagram**
Circuit diagrams can be a little hard to understand for the novice. So we use visual breadboard diagrams, like this. This complete diagram uses the power and ground pins from a Raspberry Pi to light up an LED.



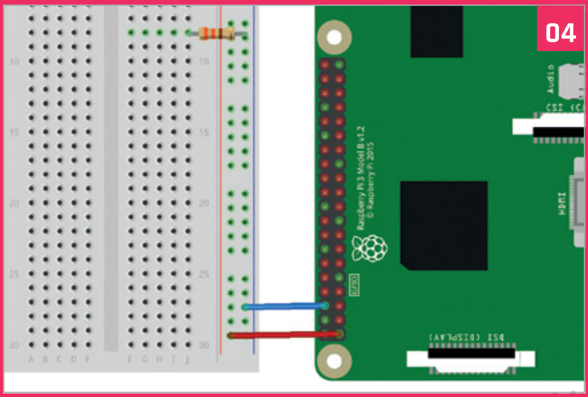
## 02 Wire up the live rail

Take a female-to-male jumper lead (the colour of the wire doesn't matter) and connect the female end to a 5V pin on the Raspberry Pi. Place the male end of the lead into a hole on the red rail on the breadboard.



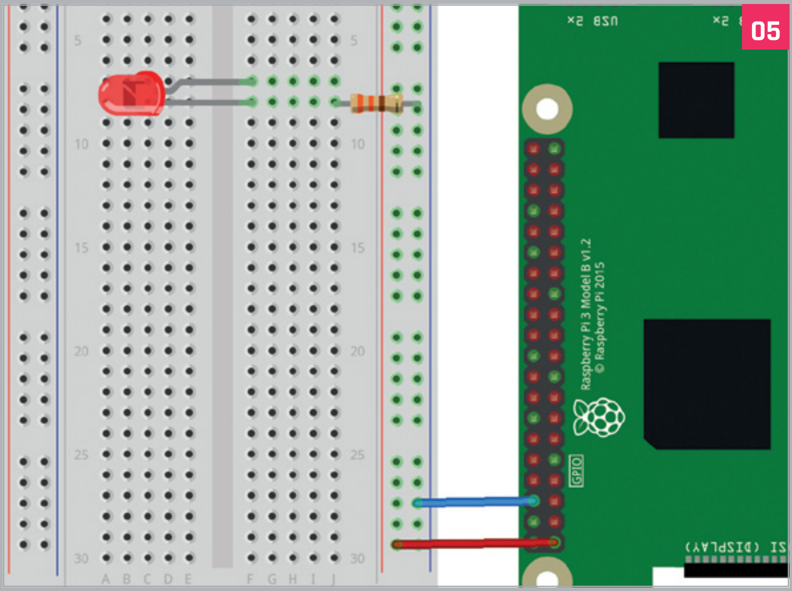
## 03 The ground rail

Take another female-to-male jumper and connect the female end to a ground (GND) pin on the Raspberry Pi. The male end goes into a hole on the blue (ground) rail. All blue holes now act as a ground pin.



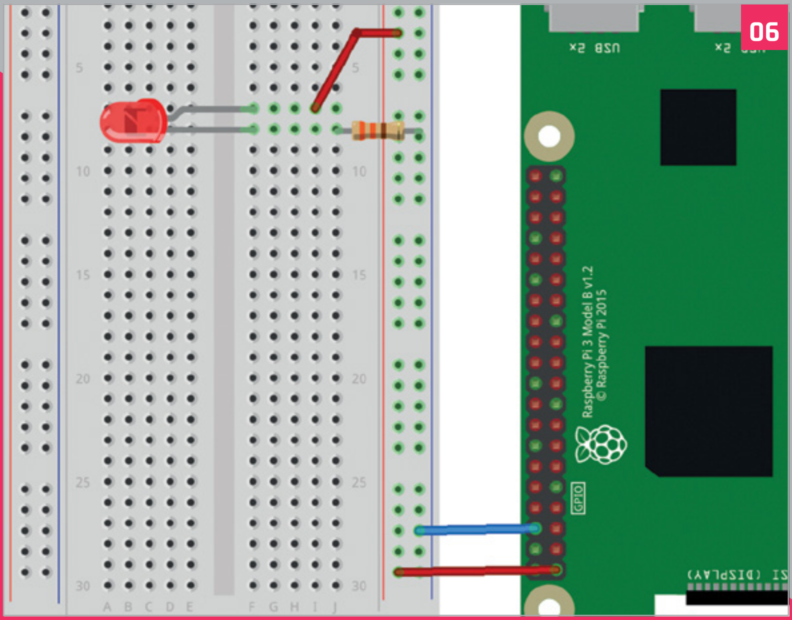
## 04 Add a resistor

Take a resistor and connect one leg of it to a hole on the ground rail of the breadboard. It's now linked to the ground pin of the Raspberry Pi (via the jumper lead we used in the previous step). Take the other leg and connect it to a hole on the main breadboard.



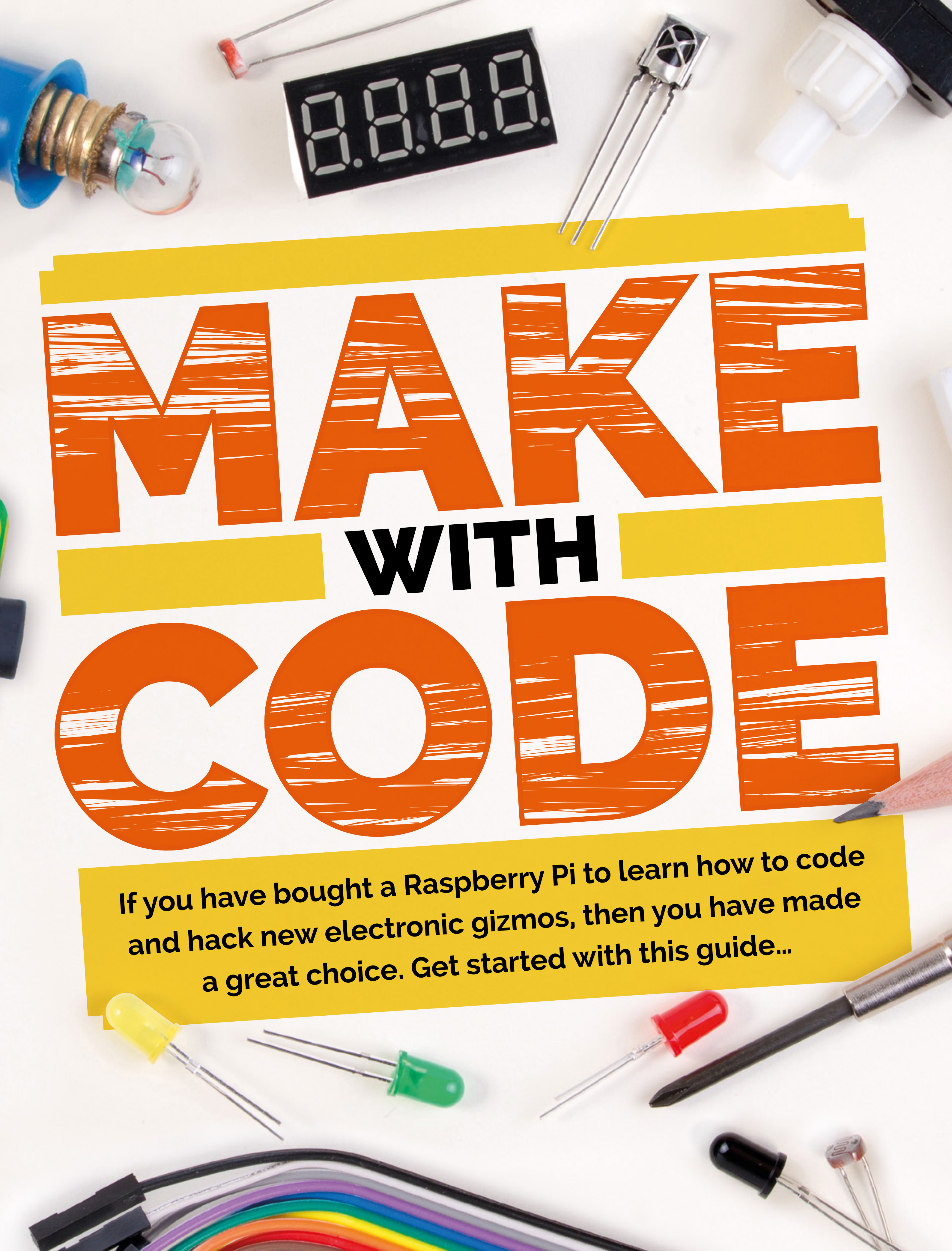
## 05 Add the LED

Take an LED component and look at the legs. Notice that one of the legs is shorter than the other. Place the shorter leg in a hole on the same row as the resistor. This leg is now connected to the resistor (which is linked to the ground rail, and therefore to the ground pin on the Raspberry Pi).



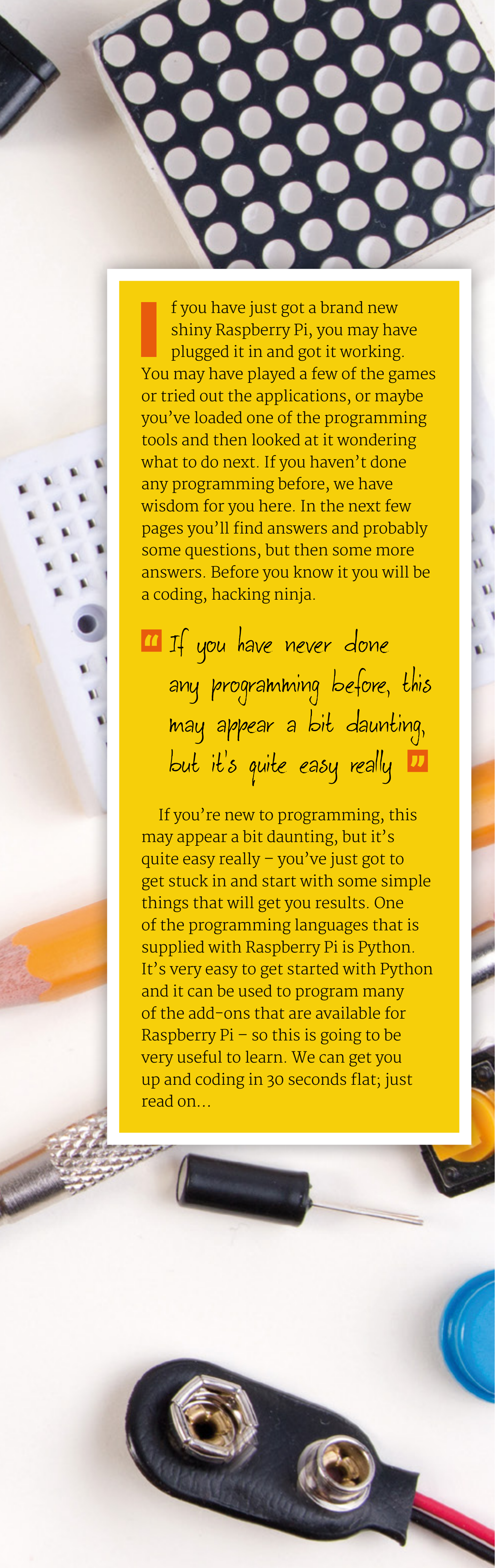
## 06 Wire it up

Place the longer leg in a hole on the next row along. Now take another male-to-male jumper lead and place one end in the hole next to the long leg of the LED. Place the other end in a hole on the red live rail to complete the circuit. The LED lights up.



# MAKE WITH CODE

If you have bought a Raspberry Pi to learn how to code and hack new electronic gizmos, then you have made a great choice. Get started with this guide…



If you have just got a brand new shiny Raspberry Pi, you may have plugged it in and got it working. You may have played a few of the games or tried out the applications, or maybe you've loaded one of the programming tools and then looked at it wondering what to do next. If you haven't done any programming before, we have wisdom for you here. In the next few pages you'll find answers and probably some questions, but then some more answers. Before you know it you will be a coding, hacking ninja.

> "If you have never done any programming before, this may appear a bit daunting, but it's quite easy really"

If you're new to programming, this may appear a bit daunting, but it's quite easy really – you've just got to get stuck in and start with some simple things that will get you results. One of the programming languages that is supplied with Raspberry Pi is Python. It's very easy to get started with Python and it can be used to program many of the add-ons that are available for Raspberry Pi – so this is going to be very useful to learn. We can get you up and coding in 30 seconds flat; just read on...

## MAKE AND RUN A PROGRAM

To get started with coding is really easy. Coding is just giving a Raspberry Pi an instruction to do something. The only thing you need to know is what language to talk to it in. In this case we are going to talk in Python 3. We will need to write our instructions somewhere so, to start, open a Terminal window – click the icon in the top left of the screen which is a grey box with a blue bar across the top. This opens a black window with a prompt: **pi@raspberrypi:~ $** in green. If you type `python3` and hit **ENTER**, Python starts and you will see the triple chevron prompt: **>>>**. Now type `print("Hello")` and hit **ENTER**. Bosh! Your first program. You have instructed your Raspberry Pi to print the word 'Hello' and, all being well, it has obeyed.

Entering programs like this is not going to be very useful most of the time, so now let's look at an app we can write and save a program with. Go to the desktop menu (click the Raspberry Pi logo in the top left of the screen) and in the Programming section, select Thonny (which should then open by default in its Simple mode). Try typing in the following program in the Thonny editor and save it, then run it by clicking the Run button. The output will be displayed in the Shell frame below the program editor.

Mark Vanstone

Educational software author from the nineties, author of the ArcVenture series, disappeared into the corporate software wasteland. Rescued by Raspberry Pi!

**magpi.cc/YiZnxL**
**@mindexplorers**

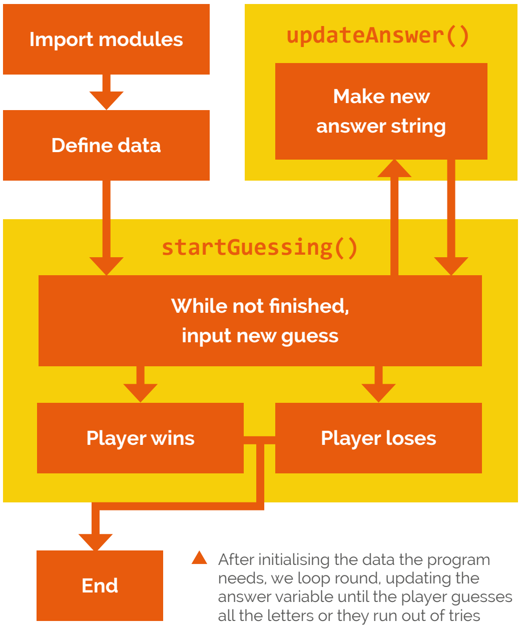
### mwc1.py

Language: **Python**
**magpi.cc/HNJhhd**

```
import random

correct = False
r = random.randint(1,10)
c = 0
while correct == False:
    n = input("Guess my number between 1 and 10: ")
    c = c + 1
    if int(n) == r:
        correct = True
    else:
        if int(n) > r:
            print("Sorry, my number is lower. Try again.")
        else:
            print("Sorry, my number is higher. Try again")

print("Well done. The correct answer was " + str(r) + ".
You got it in " + str(c) + " tries.")
```

# LEARN TO CODE

*Coding involves using several elements. Let's take a look at a range of them and how they work*



After initialising the data the program needs, we loop round, updating the answer variable until the player guesses all the letters or they run out of tries

In the last program, we get input from the keyboard and output text and numbers by using the `print()` function. We have also used a condition structure in the form of `if` and `else`. Python is very particular about how you indent the code with spaces (four per indent level); this shows that the indented code is inside another structure. In the last program, everything indented after the `while` statement will be part of that loop. Let's take a look at a few more coding techniques.

## 01 Using lists

We are going to write a Hangman-style game, where we start with a secret word and the player has to guess it, letter by letter. If correct, we show them where that letter appears in the word. They are allowed ten wrong attempts before the game ends. See the **mwc2.py** code to follow along. First, we make a list of words to choose from. A list is defined in Python using square brackets, like: `list = ["a","b","c"]`. We'll call our list `WORDLIST`. In this case we're writing the list name in upper case to show that it is a constant, i.e. It is not going to change throughout the program.

## 02 Pick a word

We can pick a word from our list using the `random` module. We import the module at the top of our code, then we can use the `random.choice()` function to get a word and store it in a variable: `theWord`. When we call a function that is inside a module, we use a full stop between the module name and the function name. Next, we want to get the player to start guessing what the word is. If we look at the bottom of the code, we can see that we call a function called `startGuessing()`. This is our own function that we need to define.

## 03 Defining functions

Each time we call a function, the code inside it runs. Sometimes functions have outputs, like our function `updateAnswer()` that returns the variable `result`. One of the rules of Python is that you must define a function before you call it, so we will need to define our `startGuessing()` function near the top of the code. To do so, we write `def` and then the name of the function, followed by brackets and a colon. If we want to pass variables as inputs into a function, we can add them inside the brackets.

## 04 Getting loopy

Now for the code in our `startGuessing()` function. We set the number of tries and dashes, one for each letter of the secret word, then we go into a loop. The code says: "While the player still has some tries left and the answer we have is not the secret word, run the following code." In our loop, we print the answer we have so far and how

## Coding guides

Want to learn more about coding? Check out our guides to Python coding and object-orientated programming in *The MagPi* #53 and #54 respectively.

**magpi.cc/53**
**magpi.cc/54**

## mwc2.py

Language: Python
magpi.cc/RqQdhR

```
import random

WORDLIST = ["orange", "table", "january", "balloon",
"mouse", "speaker", "lorry"]
theWord = random.choice(WORDLIST)

def startGuessing():
    triesLeft = 10
    answer = "-" * len(theWord)

    while triesLeft > -1 and not answer == theWord:
        print("\n" + answer)
        print(str(triesLeft) + " tries left")
        guess = input("Guess a letter:")
        if len(guess) != 1:
            print("Just guess one letter at a time.")
        elif guess in theWord:
            print("Yes that letter is in the word.")
            answer = updateAnswer(theWord, answer, guess)
        else:
            print("Sorry, that letter is not in the word.")
            triesLeft -= 1

    if triesLeft < 0:
        print("Sorry, you have run out of tries. The word was: " + theWord)
    else:
        print("Well done, You guessed right. The word was: " + theWord)

def updateAnswer(word, ans, guess):
    result = ""
    for i in range(len(word)):
        if word[i] == guess:
            result = result + guess
        else:
            result = result + ans[i]
    return result

print("I'm thinking of a word....")
startGuessing()
```

**Module**
A module is another code file. It can contain functions, variables, and data.

**List**
A list is a collection of variables, in this case string variables.

**Loop**
A loop is a part of the code that is repeated one or more times, usually depending on an equation being True.

**Input**
An input is data that a program receives, in this case from the keyboard.

**Variable**
A variable is a container for data, in this case a string which can change as the program runs.

**Function**
A function contains code that can be called from elsewhere in the program.

**Output**
Output is anything that the program produces, in this case words printed to the shell window.

**Condition**
A condition branches a program to execute one set of code or another. Sometimes there are several branches.

**Calling a function**
When a function is called, the code inside the function is run and then returns to the next line of code after the function call.

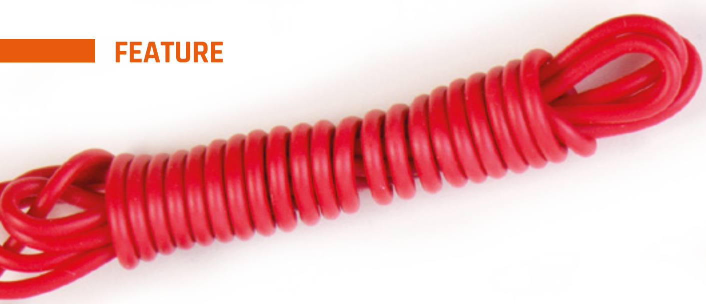
many tries the player has left. Then we use an **if**, **elif**, **else** condition structure to respond to the player, depending on what they typed.

## 05 Changing the answer

If the player guesses a letter correctly, we call another function: **updateAnswer()**. This uses a **for** loop to add the correct letters into our answer variable, then return that string (a variable containing letters/characters rather than a numeric value). This then becomes the answer variable that we print at the start of each loop in the **startGuessing()** function. When the player has guessed all the right letters in our word or they have got it wrong ten times, the program will drop out of the loop to reach the final part of the function.

## 06 Win or lose

We have an **if** and **else** structure to print congratulations, or let the player know they've run out of tries. A few of the functions are used with variables: **len()** finds the length of a string, and **str()** converts a number variable into a string so it can be added to the start or the end of another string. After the function is complete, it returns to where it was called, which is the end of the program.



# CONTROL THINGS WITH CODE

Now we've got the hang of the coding, let's put it to work with some electronic components

In the previous example, we imported a module into our code and used functions from it. If we want to control electronics from code, there is a very useful module available called **`gpiozero`**. GPIO stands for 'general-purpose input/output' and the line of double pins on one side of your Raspberry Pi are called GPIO pins. For details about the labels of all the pins, see **pinout.xyz**. If we connect electrical components to these GPIO pins, we can use the **`gpiozero`** module to make things happen. When we import a module, there are often many different functions inside. We can also make new coding 'objects' with them. Objects are like variables but have their own set of functions and properties inside them that we can call or change, and we do that using the same dot notation (full stop) that we did with the **`random`** module. For more details on using coding objects (called object-oriented programming or OOP), see issue 54 of *The MagPi*: **magpi.cc/54**.

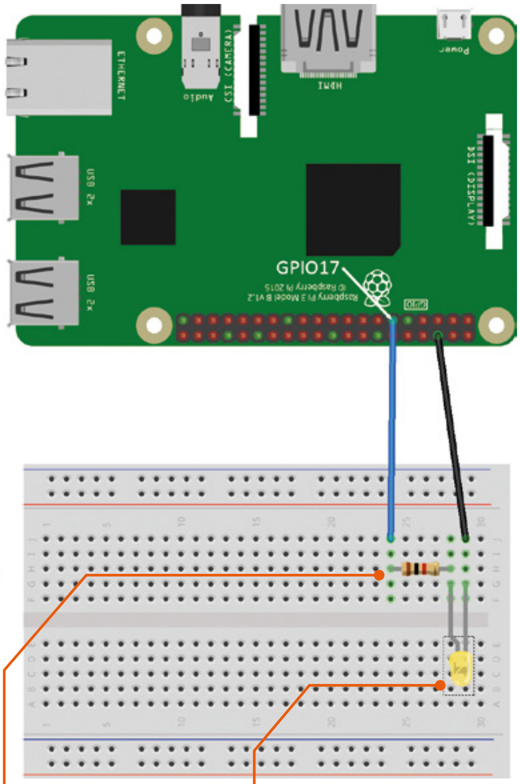
## You'll Need

A breadboard
**magpi.cc/NtjSiy**

An LED
**magpi.cc/WBVPxG**

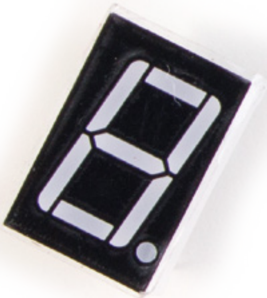
A resistor
**magpi.cc/iDTFag**

2 × male-to-female jumper wires
**magpi.cc/OkMyVX**



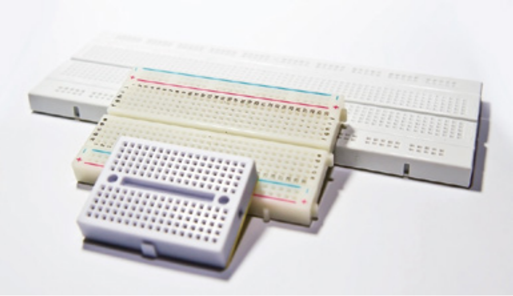
This resistor is 1kΩ (kilohm), but you can use a similar value

This is a yellow LED, but you can buy LEDs in many different colours



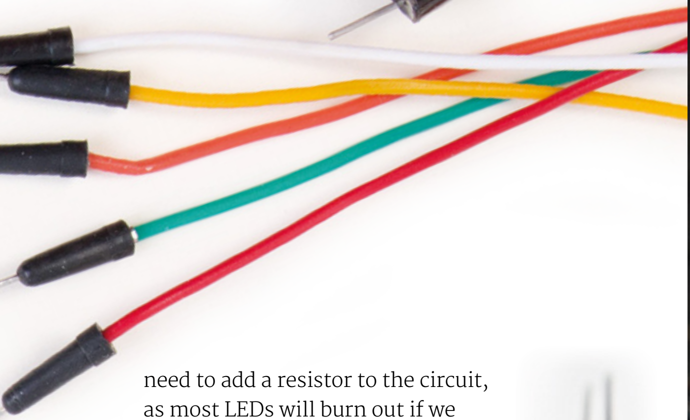
## 01 The breadboard

Breadboards come in various sizes, but they all work in the same way. If there are tracks marked red and black/blue (and/or + and −) on the long edges of the board, these are for power and are connected along the length – although sometimes divided into sections. The matrix of holes which make up the main part of the board are connected in lines the other way (vertically in the diagram). There is usually a break in the centre of the board so that the two sides are not connected.

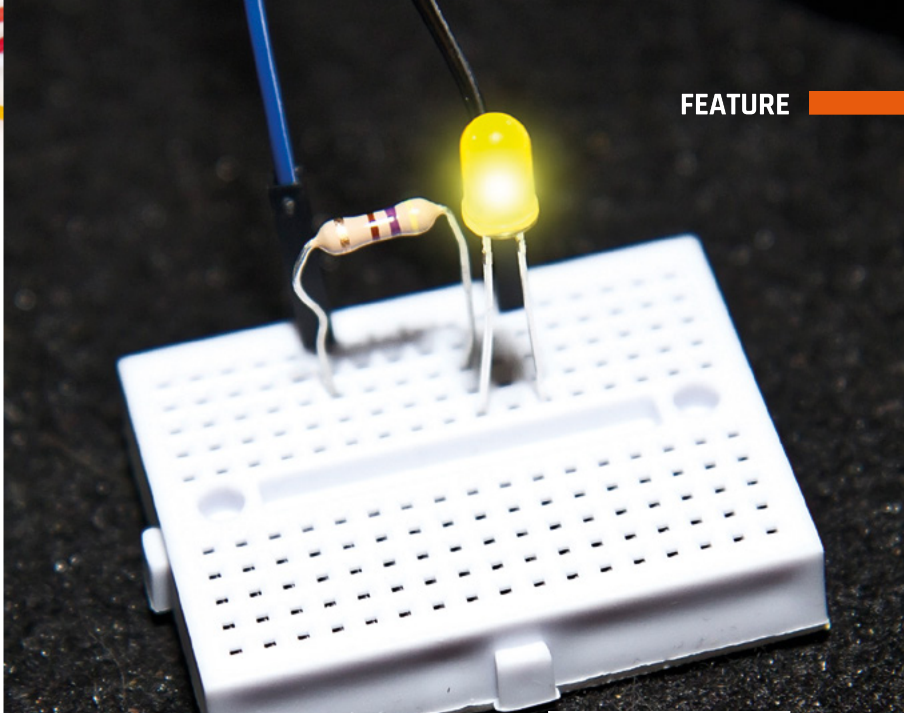


## 02 Light-emitting diode (LED)

An LED is a bit like a bulb in that if you apply electricity to it in the right way, it lights up. An LED is also a diode, which means that the electricity needs to flow in the correct direction or it will not light. When connecting an LED to Raspberry Pi, we



need to add a resistor to the circuit, as most LEDs will burn out if we connect them directly to the main power output. In the example, we are using a 1kΩ resistor, but it's fine to use another similar value. To light an LED using `gpiozero`, assemble the components as in the diagram, then write a Python program: start with `from gpiozero import LED`, then create an LED object with `led = LED(17)`, and finally type `led.on()` to light the LED.



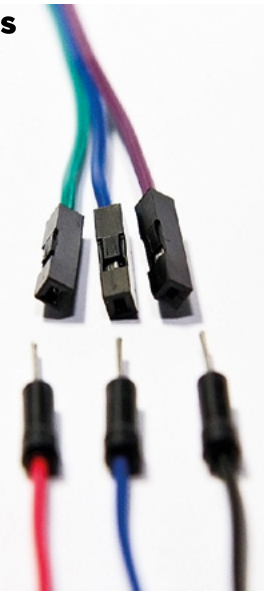
▲ You can control the state of the LED using the gpiozero module in Python

## 03 A resistor

Resistors do what their name suggests: they resist the flow of electricity (current). Some components need to have a certain amount of current in order to operate correctly. Resistors enable us to set the current or voltage to a suitable level for the other components we are using. There are many different types of resistors, with different resistance values. The resistance value can be read from the pattern of coloured stripes on the resistor. You can also get variable resistors, which are known as potentiometers.

## 04 Jumper wires

We need to connect our components to Raspberry Pi's GPIO pins. For this we use jumper wires. The ones we will be using have a female connector at one end, to go on the GPIO pins, and a male connector to go in the holes of the breadboard. You can also get jumper wires with both male connectors or both female connectors, for different situations. They can be bought in strips all joined together, sometimes known as 'jumper jerky'.

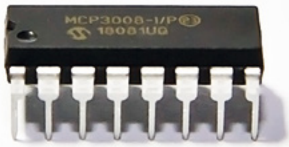


## 05 Potentiometer

A potentiometer is a variable resistor. It usually has a turning spindle that changes the resistance from one value to another, quite often from no resistance to full resistance (no electricity flowing). A potentiometer can provide us with a variable output voltage which we can measure with the GPIO, but there is a slight problem. The potentiometer provides an 'analogue' output (varies continuously between values) and the GPIO inputs are only digital, i.e. on or off. So we need another component: an analogue-to-digital converter (ADC).

## 06 Analogue-to-digital converter

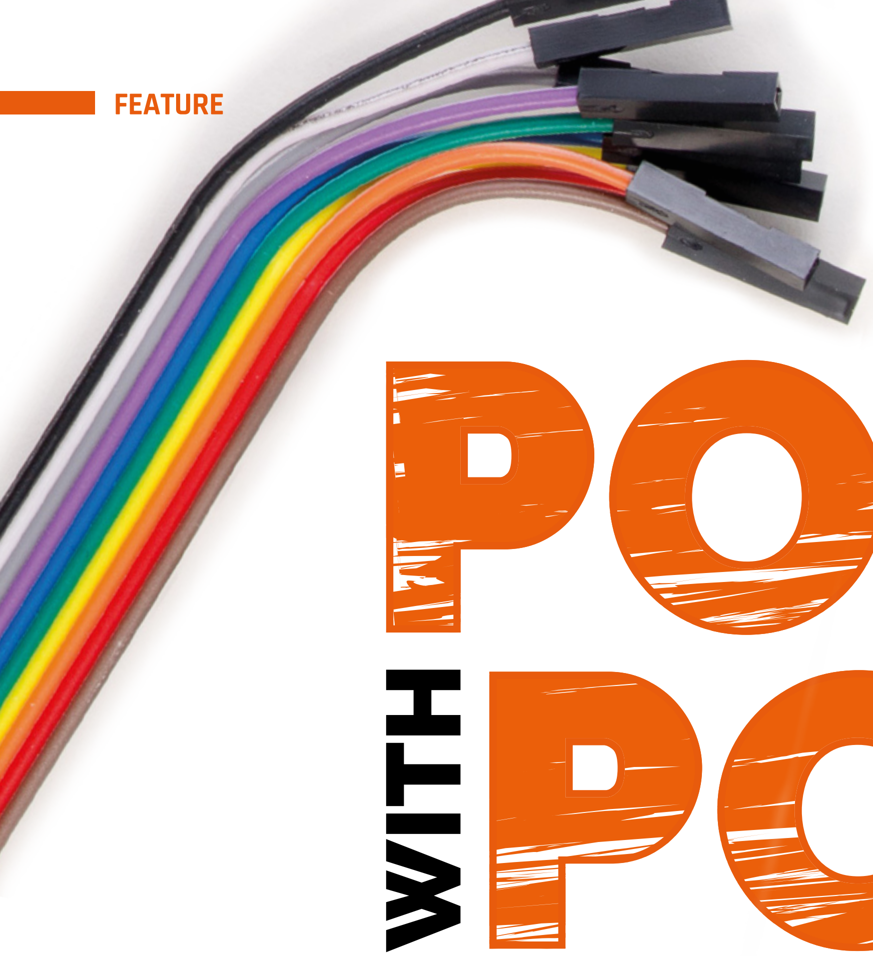
ADC components come in various forms, but the one we have in this example is called an MCP3008. It's an integrated circuit (IC), meaning that it's a box with some circuitry inside it. We don't need to know what is inside it – we just need to know what to connect to each of the legs of the IC. We will need to wire up several of the legs to GPIO pins and provide the IC with power; when we've done that, we can connect the potentiometer to the IC and then read values in showing the position of the potentiometer spindle using the `gpiozero` module. We'll cover the code in the next section.



### Electronics guide

For more details about using electronic components with Raspberry Pi, check out our Electronics Starter Guide in issue 64 of *The MagPi*.

**magpi.cc/64**



# PONG WITH POTS

Now to put what we have covered to the test: we will make a retro game and control it with our own homemade controllers

## You'll Need

- Breadboard **magpi.cc/vhZkGh**
- 6 × male-to-female jumper wires & 10 × male-to-male jumper wires **magpi.cc/ffZNwL**
- 2 × potentiometers **magpi.cc/oZRFEe**
- MCP3008 integrated circuit **magpi.cc/aCvXuo**

On the previous page we talked about potentiometers and analogue-to-digital converters, and this is where we get to use them. It's a bit more complicated than lighting up an LED, but only a little. First, we are going to write a program which has two rectangular bats at each side of the screen that can be moved up and down by two players. A ball bounces backwards and forwards between the bats until one player misses the ball and the other player scores a point. That's right, you guessed it, the game is Pong and we are going to create a controller for each player from a potentiometer and wire it into our Raspberry Pi.

### 01 Super-fast game coding

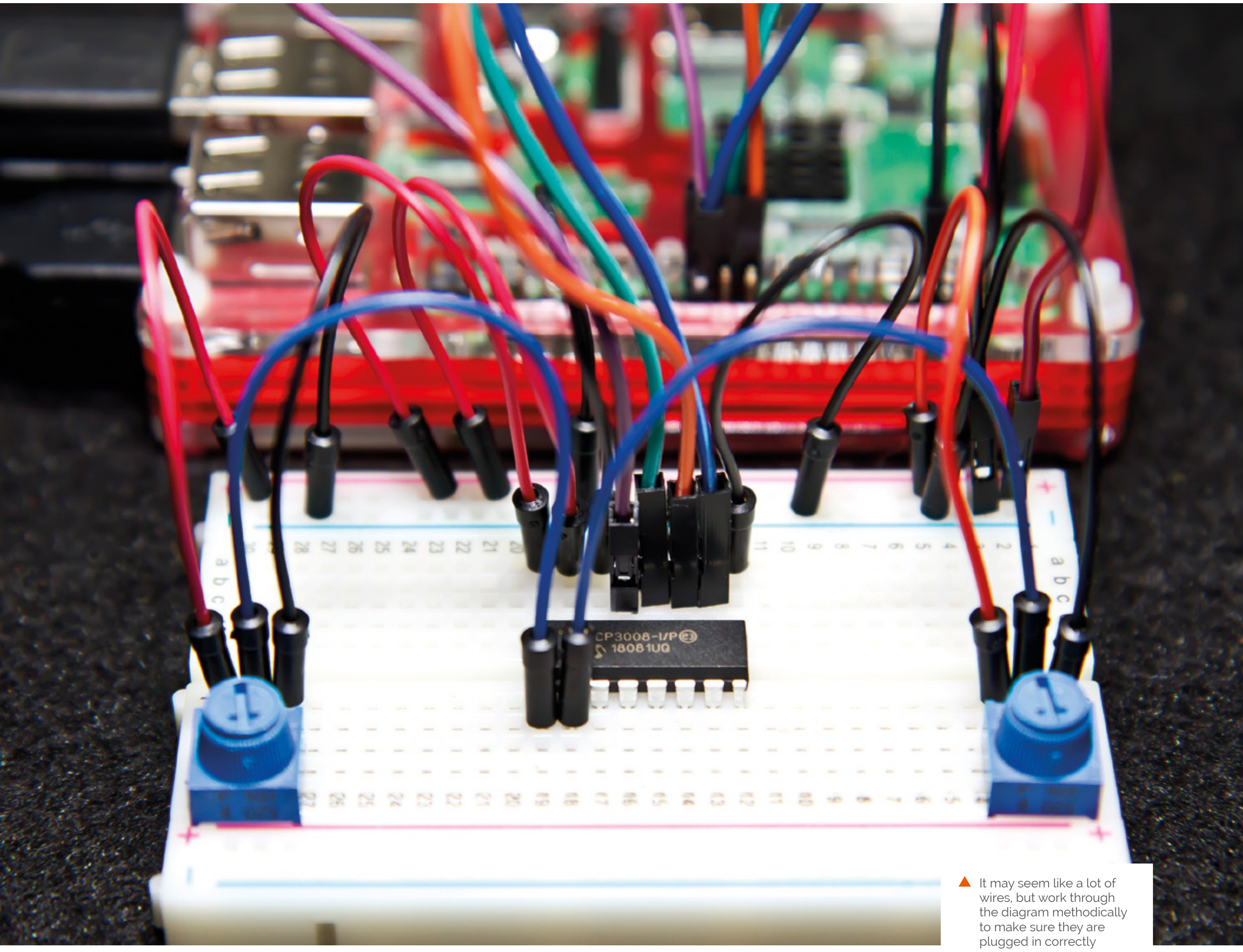
If you have been following other coding articles in recent issues of *The MagPi*, you will know that when writing a quick game on Raspberry Pi, Pygame Zero is your friend. We can make the basics of the game code very quickly by importing the `pgzrun` module, which holds all the Pygame Zero code. We need to call `pgzrun.go()` at the end of our code, and that's our game window sorted. As with all Pygame Zero programs, we have a `draw()` function to write the graphics to the window, and an `update()` function to update the game items between draw cycles.

### 02 Running the code

The listing **mwc3.py** provides all the code you need for the game to work. There is some code in the `updatePaddles()` function for the keyboard to control the paddles or bats, just in case you want to test it before making the proper controllers. We import several modules with this code. We have covered `pgzrun`, but we also need `random` so that the ball will move in a random direction each time it starts. In addition, we need `gpiozero` to deal with the input from the controllers, and we need the `math` module for calculating the direction of the ball.

### 03 Wiring it all up

One thing to bear in mind when connecting any electronics to a computer is that if the wires are connected in the wrong way, you may cause damage to the computer or the electronic



It may seem like a lot of wires, but work through the diagram methodically to make sure they are plugged in correctly

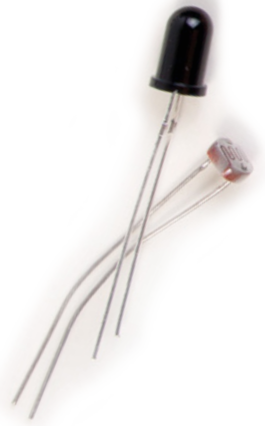
components, so it's always a good idea to power off your Raspberry Pi before connecting anything to the GPIO pins. Follow the wiring diagram (overleaf) carefully, making sure that the jumper wires are connected to the right GPIO pins and to the correct places on the breadboard. Once you have put everything in place, it's a good idea to have another check just to make sure.

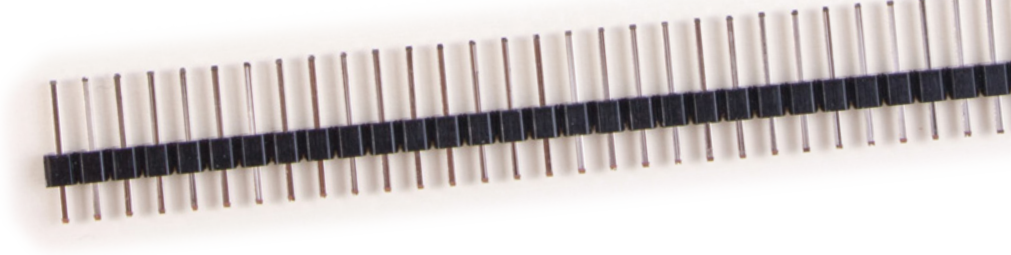
## 04 The MCP3008 IC

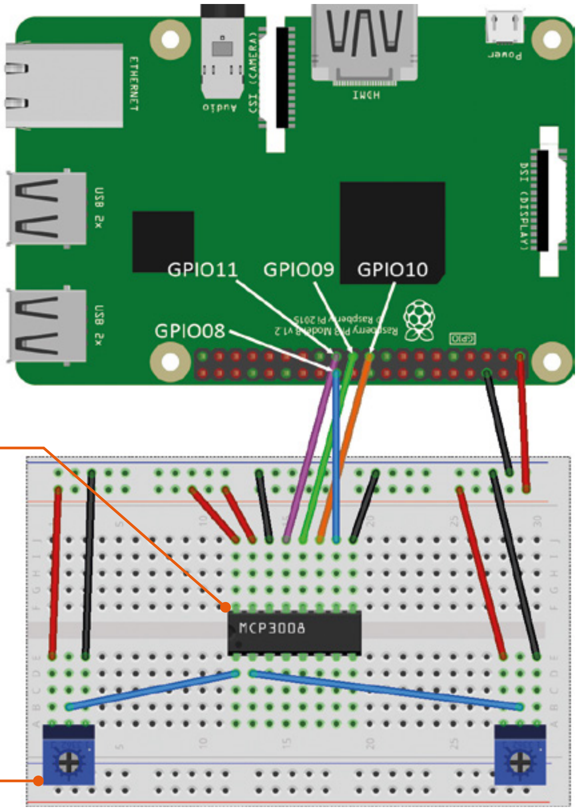
The MCP3008 converts the voltage from our potentiometers into a number with the help of the `gpiozero` module. It has eight channels for input, but we are going to just use two of them in this case. You will see from the diagram that all the legs on the top side of the IC are connected to either GPIO pins or to power lines. There are two red connections going to the positive power track, then a black lead to the negative or ground track. Then there are four coloured wires that go to: purple – GPIO 11; green – GPIO 09; orange – GPIO 10; and blue – GPIO 08. Then there is one last connector to the ground track.

## 05 The inputs

All the MCP3008 pins on the bottom side of the IC are input channels. We will use the first two pins, which are channel 0 and channel 1. We







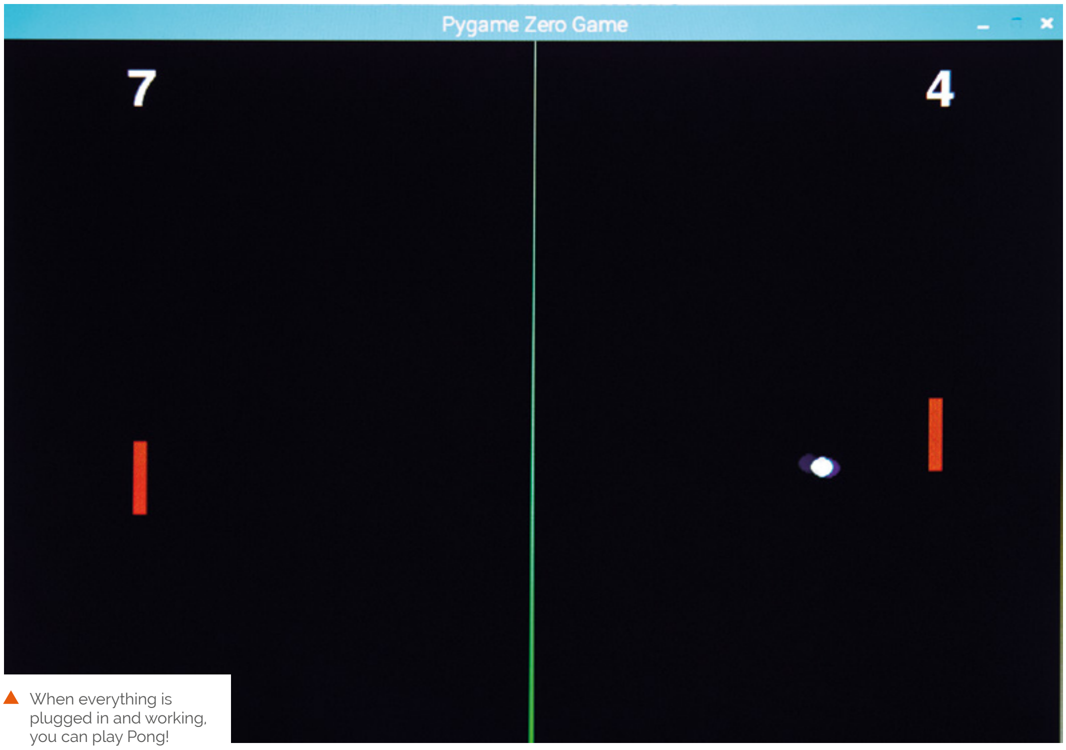
The MCP3008 straddles the centre break of the breadboard so that the pins on either side are not connected

The left-hand potentiometer's middle pin connects to pin 0, and the right one's to pin 1 of the MCP3008

connect the middle pins of our potentiometers to those channel pins, which will read the positions and convert that to a value in our program. If you want to know exactly what all the pins are for on this IC, you can do a web search for 'MCP3008 pinout' and that will give you descriptions of each. ICs are very useful in electronics as they mean that we can reduce how complex our circuits are and we don't need to know exactly how they work inside. They are a little bit like modules in Python.

## 06 Finishing the job

When you have checked that all the connections are correct, you can switch your Raspberry Pi back on and reload your program. You may want to initialise the SPI interface on your Raspberry Pi by going to the main desktop menu > Preferences > Raspberry Pi Configuration, and go to the Interfaces tab. It will work without this, but may cause a few warnings in the Thonny shell window. So, all being well, when you run your program you will have a game of Pong which can be controlled by two players with the potentiometers. If it doesn't work first time, check your code and then your wiring, and try again.



When everything is plugged in and working, you can play Pong!

## mwc3.py

Language: Python
magpi.cc/umUcfq

```python
import pgzrun
import random
from gpiozero import MCP3008
import math

pot1 = MCP3008(0)
pot2 = MCP3008(1)

# Set up the colours
BLACK = (0  ,0  ,0  )
WHITE = (255,255,255)
p1Score = p2Score = 0
BALLSPEED = 5
p1Y = 300
p2Y = 300

def draw():
    screen.fill(BLACK)
    screen.draw.line((400,0),(400,600),"green")
    drawPaddles()
    drawBall()
    screen.draw.text(str(p1Score) , center=(105, 40), color=WHITE, fontsize=60)
    screen.draw.text(str(p2Score) , center=(705, 40), color=WHITE, fontsize=60)

def update():
    updatePaddles()
    updateBall()

def init():
    global ballX, ballY, ballDirX, ballDirY
    ballX = 400
    ballY = 300
    a = random.randint(10, 350)
    while (a > 80 and a < 100) or (a > 260 and a < 280):
        a = random.randint(10, 350)
    ballDirX = math.cos(math.radians(a))
    ballDirY = math.sin(math.radians(a))

def drawPaddles():
    global p1Y, p2Y
    p1rect = Rect((100, p1Y-30), (10, 60))
    p2rect = Rect((700, p2Y-30), (10, 60))
    screen.draw.filled_rect(p1rect, "red")
    screen.draw.filled_rect(p2rect, "red")

def updatePaddles():
    global p1Y, p2Y

    p1Y = (pot1.value * 540) +30
    p2Y = (pot2.value * 540) +30

    if keyboard.up:
        if p2Y > 30:
            p2Y -= 2
    if keyboard.down:
        if p2Y < 570:
            p2Y += 2
    if keyboard.w:
        if p1Y > 30:
            p1Y -= 2
    if keyboard.s:
        if p1Y < 570:
            p1Y += 2

def updateBall():
    global ballX, ballY, ballDirX, ballDirY, p1Score, p2Score
    ballX += ballDirX*BALLSPEED
    ballY += ballDirY*BALLSPEED
    ballRect = Rect((ballX-4,ballY-4),(8,8))
    p1rect = Rect((100, p1Y-30), (10, 60))
    p2rect = Rect((700, p2Y-30), (10, 60))
    if checkCollide(ballRect, p1rect) or checkCollide(ballRect, p2rect):
        ballDirX *= -1
    if ballY < 4 or ballY > 596:
        ballDirY *= -1
    if ballX < 0:
        p2Score += 1
        init()
    if ballX > 800:
        p1Score += 1
        init()


def checkCollide(r1,r2):
    return (
        r1.x < r2.x + r2.w and
        r1.y < r2.y + r2.h and
        r1.x + r1.w > r2.x and
        r1.y + r1.h > r2.y
    )

def drawBall():
    screen.draw.filled_circle((ballX, ballY), 8, "white")
    pass

init()
pgzrun.go()
```

# THE BEST RASPBERRY PI ACCESSORIES

Our guide to the greatest add-ons, HATs, and expansions for your Raspberry Pi

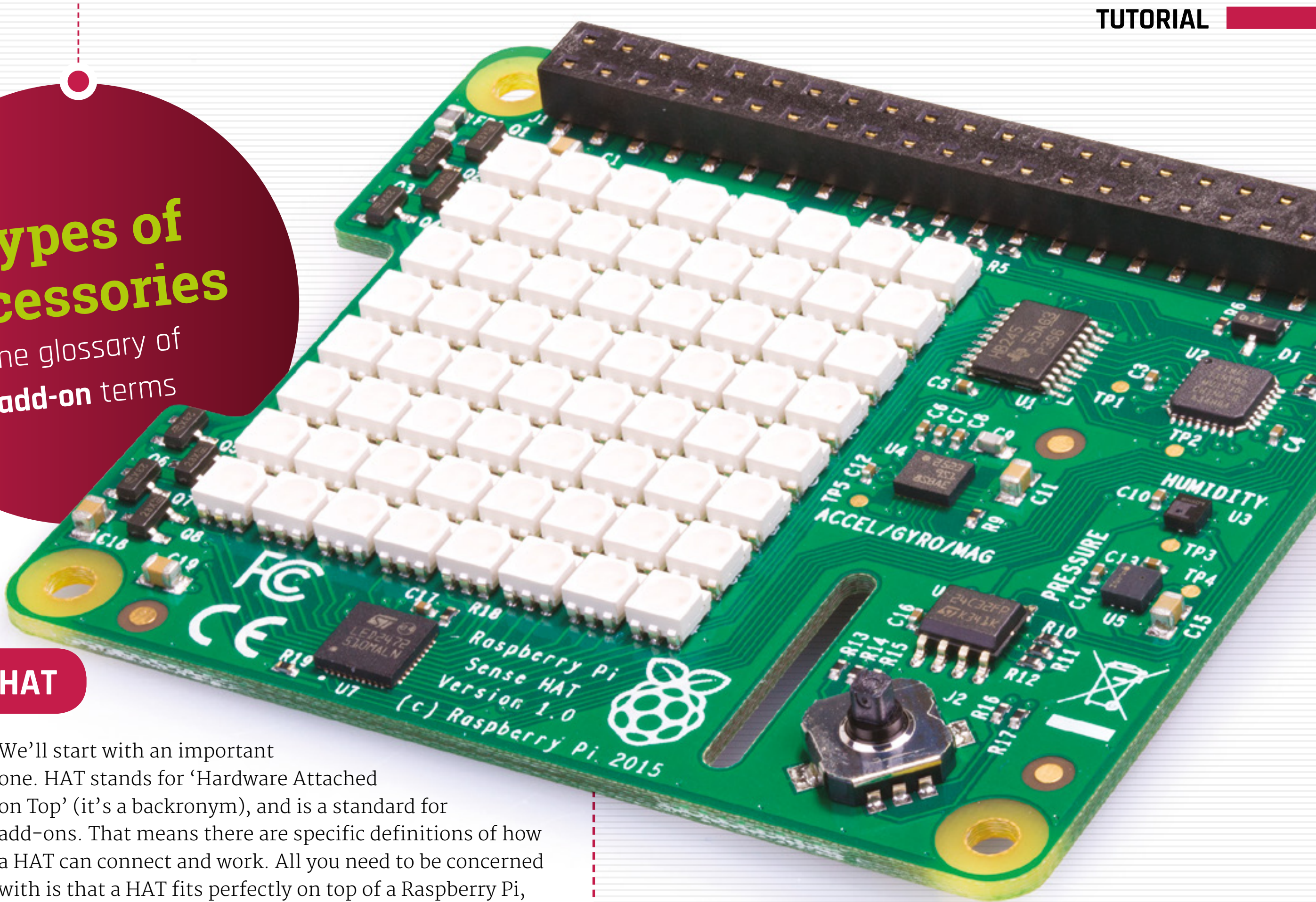
Raspberry Pi is fantastic on its own. With wireless LAN, Bluetooth, USB ports, the multitalented GPIO pins, and a full operating system, it really is a fully functional computer.

What if you could make it better? Do more? Make certain functions easier? This is where add-ons and accessories come in – there are hundreds of extras you can get for your Raspberry Pi that will improve its already impressive specs.

What add-ons should you get, though? We're here to show you the best the Raspberry Pi world has to offer.

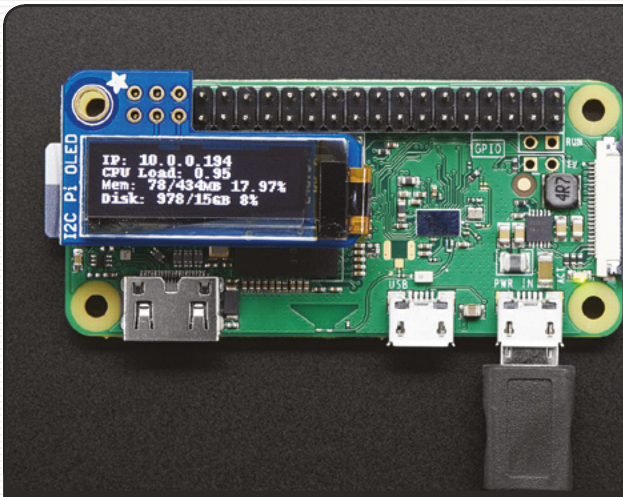
# Types of accessories

The glossary of **add-on** terms



## HAT

We'll start with an important one. HAT stands for 'Hardware Attached on Top' (it's a backronym), and is a standard for add-ons. That means there are specific definitions of how a HAT can connect and work. All you need to be concerned with is that a HAT fits perfectly on top of a Raspberry Pi, connecting to the GPIO pins. You will usually need to install extra software to get them working.

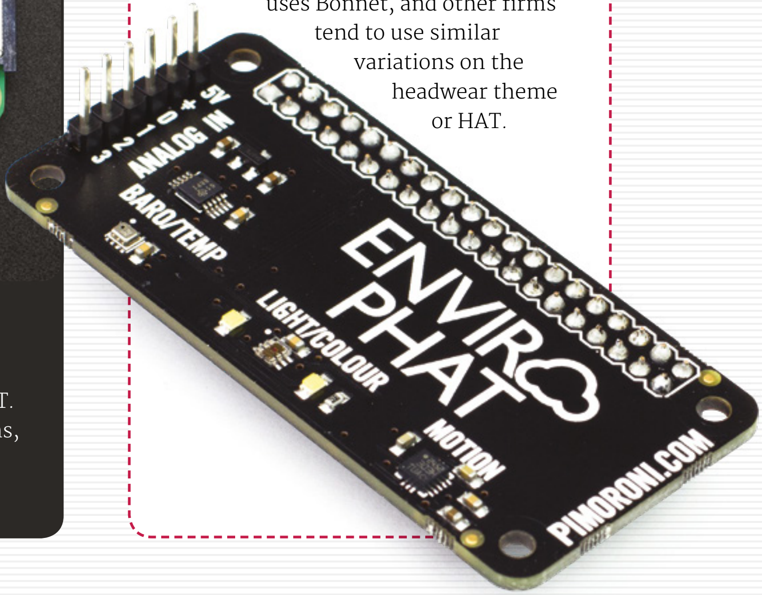


## Add-on

A HAT is an add-on, but an add-on isn't necessarily a HAT. Add-ons can be connected via USB, only specific GPIO pins, the camera port, etc. It's a catch-all term for something that you can add onto your Raspberry Pi in some way.

## pHAT/Bonnet/Cap/etc.

While the HAT standard has recently been extended to different sizes, many companies have created their own term for add-ons that act like a HAT but don't quite meet the technical criteria. Pimoroni coined pHAT, and Adafruit uses Bonnet, and other firms tend to use similar variations on the headwear theme or HAT.



# PROGRAMMING AND MAKER ACCESSORIES

Improve your **projects** with these add-ons

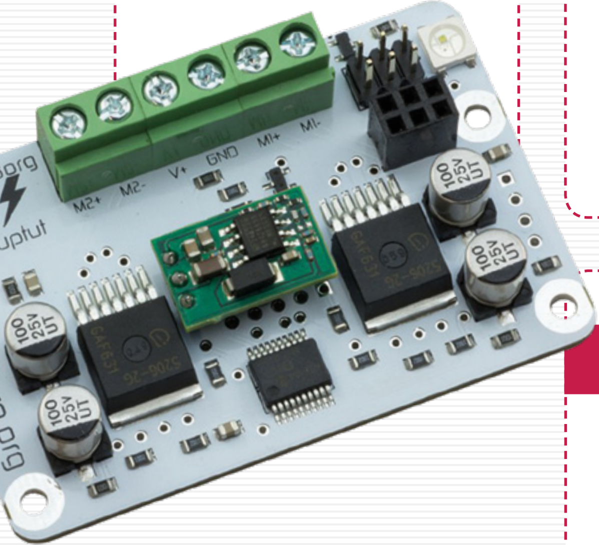
## ThunderBorg

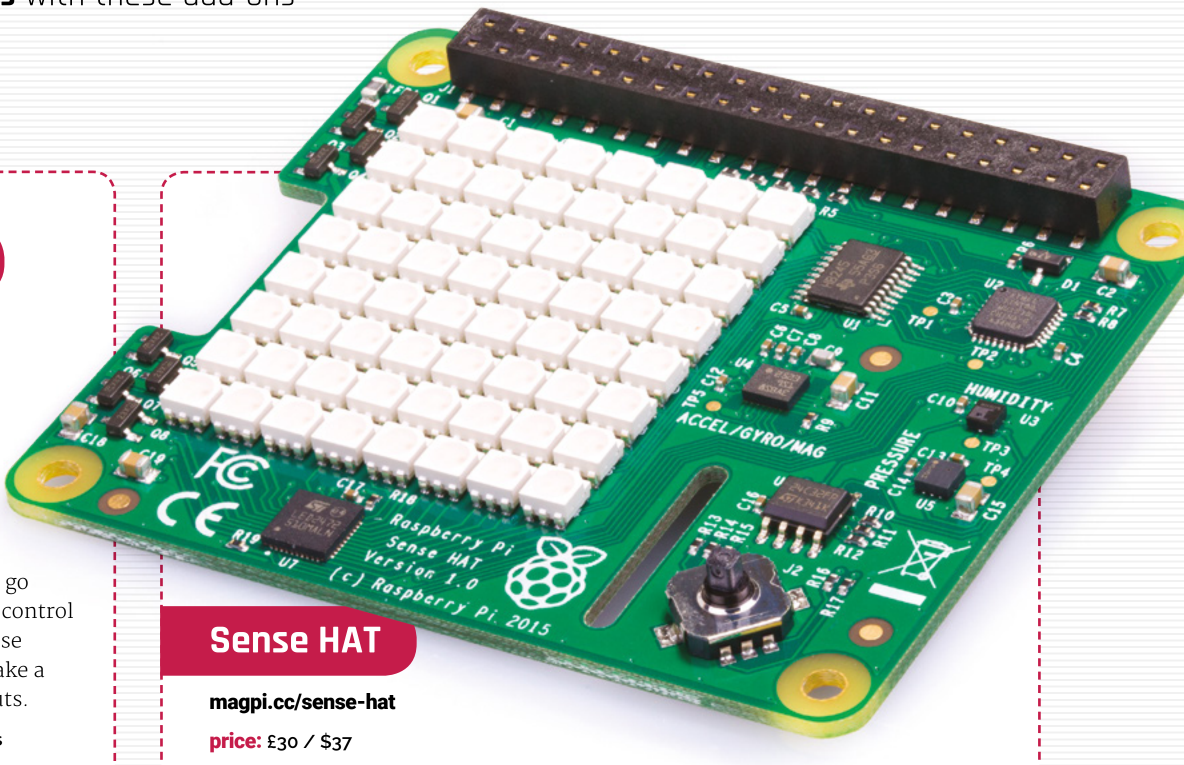
**magpi.cc/xcsSpm**

**price:** £37 / $50

This dual motor control board by the folks at PiBorg is a powerful and robust way to use a Raspberry Pi to control a robot. As well as being able to make the motors go forward and back, it can control the speed for more precise manoeuvring, and can take a wide array of power inputs.

**Use it with:** Robots, RC cars





## Sense HAT

**magpi.cc/sense-hat**

**price:** £30 / $37

Developed for the Astro Pi mission, the Sense HAT is primarily a set of environmental sensors. It can detect pressure, humidity, temperature, and orientation/movement. It also has an 8×8 LED matrix and a little joystick for very basic display and control. There are a couple of Sense HATs running on Raspberry Pi boards up on the ISS at the moment.

**Use it with:** Weather stations, space stations

## Hologram Nova

**hologram.io/nova**

**price:** From £53 / $64

Despite the world's best efforts, WiFi is not available absolutely everywhere yet. You can't even plug an Ethernet cable into a tree! The Nova solves this problem by letting you connect to cellular networks around the world on your Raspberry Pi via this USB device.

**Use it with:** Portable projects, outdoor cameras

## Edge TPU Accelerator

**magpi.cc/TFLQpM**

**price:** £61 / $75

Google was responsible for the AIY Voice Kit – a special HAT and extras that lets you turn a Raspberry Pi into an AI voice assistant. The Edge TPU accelerator aims to help Raspberry Pi out with AI tasks, and make your voice assistants better.

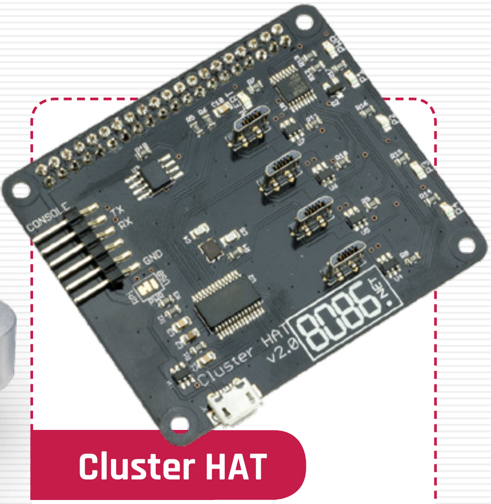
**Use it with:** Voice control, AI projects

## GFX HAT

**magpi.cc/ZWvcLG** **price:** £22 / $22

A very functional screen that draws a lot less power than most displays, it also includes six capacitive touch buttons for control. It's the successor to an excellent add-on known as Display-O-Tron, and makes interactive projects look great.

**Use it with:** Thermostat, practical projects



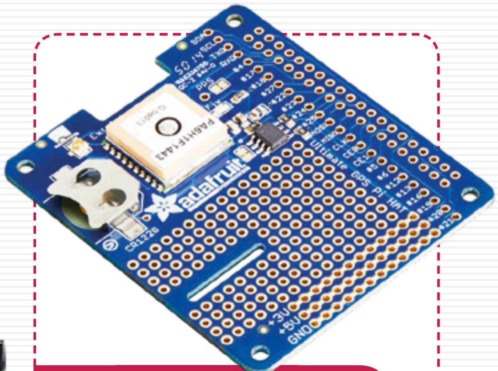
## Cluster HAT

**magpi.cc/Qrshbi**

**price:** £31 / $37

Putting Raspberry Pi boards into a cluster isn't that tricky, but the Cluster HAT makes it very easy. Using one Raspberry Pi, you can distribute computing to up to four Raspberry Pi Zeros. Cluster computing is a great way to perform different kinds of calculations.

**Use it with:** Mathematics, big algorithms



## Ultimate GPS HAT

**magpi.cc/vmPqLL**

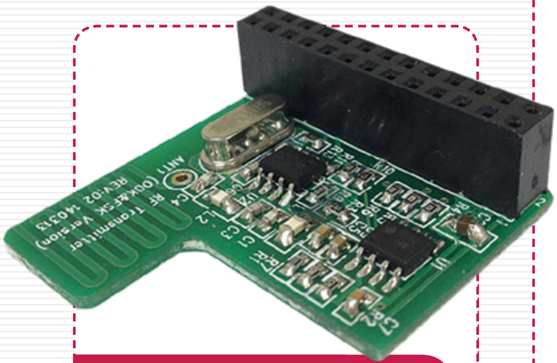
**price:** £42 / $45

This GPS add-on allows you to track the precise location of your Raspberry Pi – well, as precise as GPS can be. You'll have to combine the position data with map data to make much use out of it, but once you have it all working it's pretty fantastic.

**Use it with:** Sat-navs, route tracking

# INDUSTRIAL AND POWER

Using a Raspberry Pi for **enterprise?** You may want to check these out...

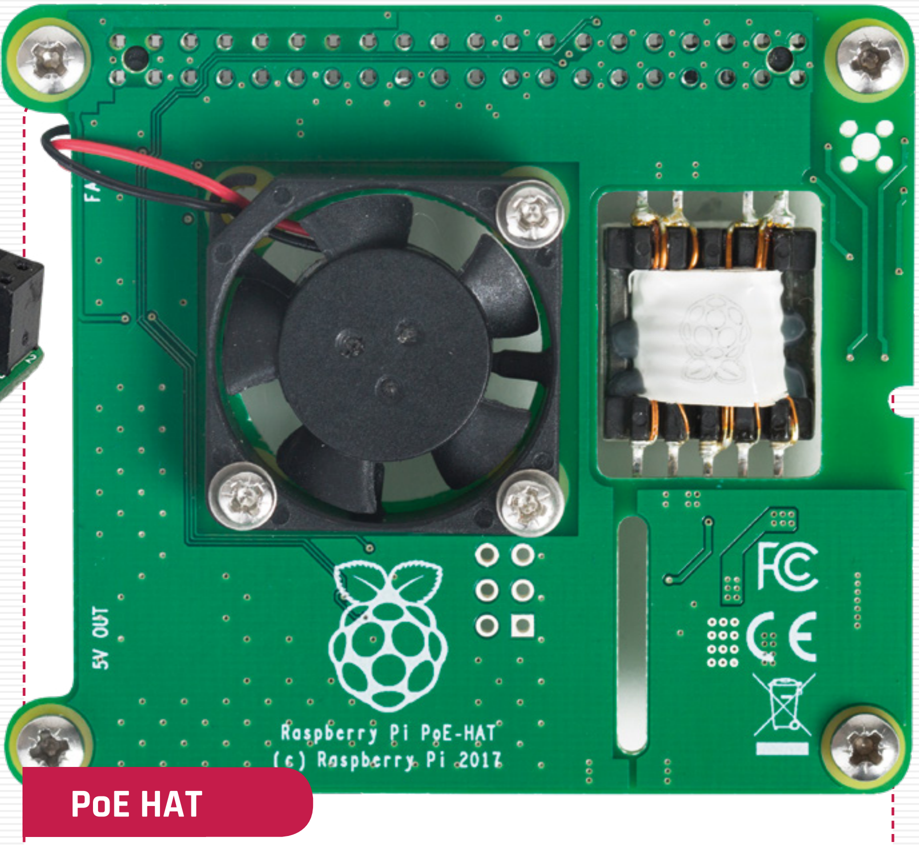


## Energenie Pi-mote

**magpi.cc/FnezCP**
**price:** £22 / $27

Another remote-control power solution, this one is designed for Raspberry Pi to control power elsewhere. There are various Energenie Pi-mote products, the cheapest starting at £10, but the starter kit version also includes compatible sockets. You can even get a Pi-mote that allows the plugs to talk to Raspberry Pi as well.

**Use it with:** Timed-lighting, some IoT

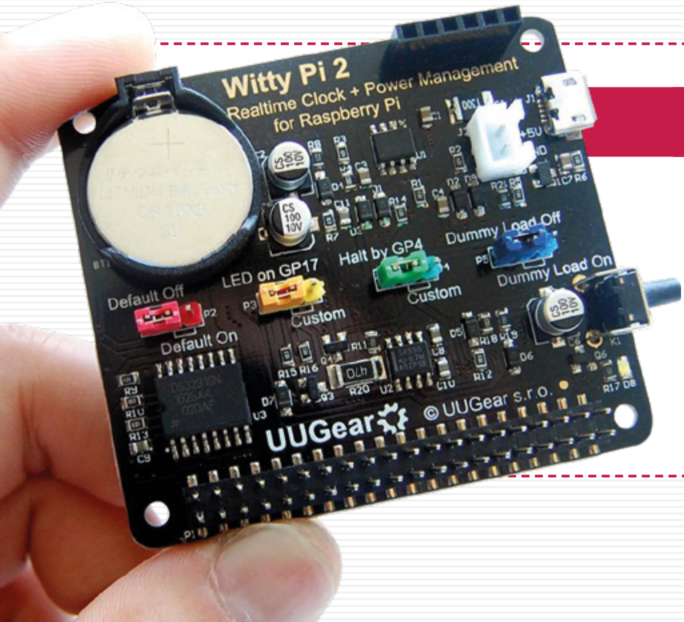


## PoE HAT

**magpi.cc/aqpwZc** **price:** £18 / $22

Raspberry Pi 3B+ and 4 models include proper Power-over-Ethernet support, although you need one of these HATs to get it working. Of course, the network needs to be set up to support (i.e. power) the PoE and Raspberry Pi combo. It also currently only works on Raspberry Pi 3B+ and 4.

**Use it with:** Server racks, headless solutions

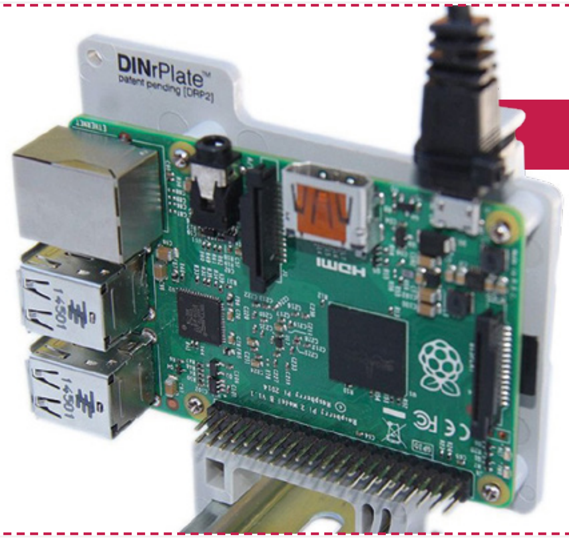


## Witty Pi 2

**magpi.cc/merPpf**
**price:** £20 / $25

Serious Raspberry Pi users often need power management and a real-time clock (RTC) for various reasons. The Witty Pi 2 offers these things, along with some extra on/off functions that can programmed. As Raspberry Pi normally requires the internet to check the time, an RTC is essential for scheduling tasks on a Raspberry Pi that's not online.

**Use it with:** Industrial settings, automated solutions



## DINrPlate DIN rail mount

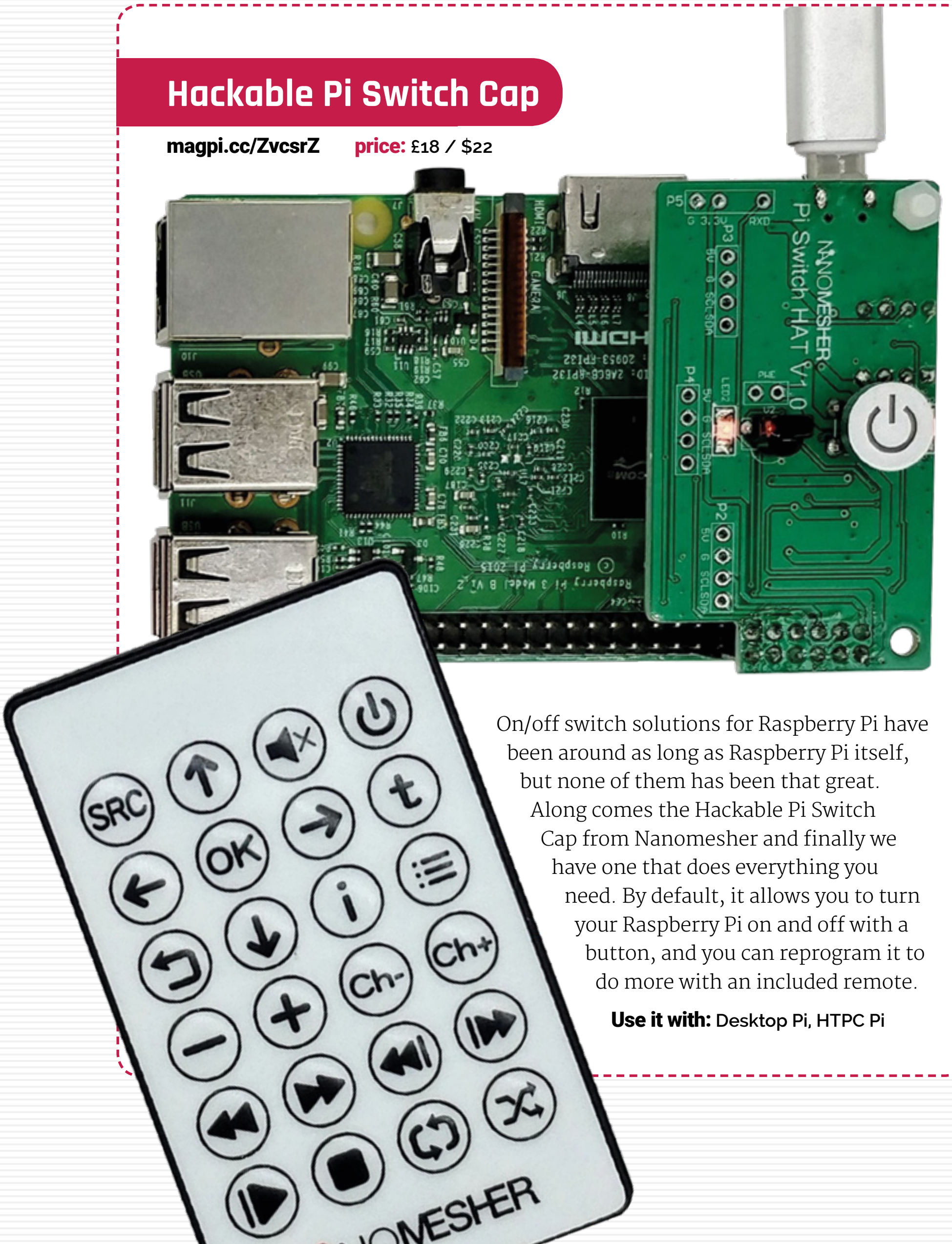
**dinrplate.com**
**price:** From £10 / $13

Want to use a Raspberry Pi in a server environment, but don't need a UPS or RTC or any of the advanced stuff like that? Just have some DIN rails handy anyway? Then the DINrPlate is an excellent add-on that lets you mount a Raspberry Pi to a DIN rail without any fancy extras. And the price reflects that.

**Use it with:** DIN rails, DIY servers

## Hackable Pi Switch Cap

**magpi.cc/ZvcsrZ** **price:** £18 / $22



On/off switch solutions for Raspberry Pi have been around as long as Raspberry Pi itself, but none of them has been that great. Along comes the Hackable Pi Switch Cap from Nanomesher and finally we have one that does everything you need. By default, it allows you to turn your Raspberry Pi on and off with a button, and you can reprogram it to do more with an included remote.

**Use it with:** Desktop Pi, HTPC Pi

## Strato Pi CM

**magpi.cc/rfBfvo**
**price:** From £206 / $266

Are you a sysadmin in a big company that would like to use Raspberry Pi boards? The Strato Pi turns a Raspberry Pi into a server good enough to pass the rigorous tests of enterprise IT. It has an impressive list of features: RTC, UPS, two-module DIN-rail case, RS-485, and more.

**Use it with:** Enterprise, servers

# MEDIA ADD-ONS

Improve your **listening, watching,** and **more**

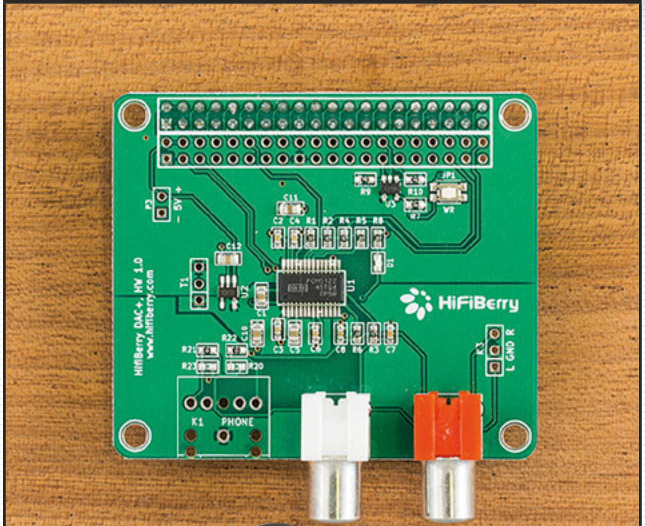
## HiFiBerry DAC+

**magpi.cc/aimAyy**

**price:** From £20 / $24

There are many digital-to-analogue converters (DACs) that turn Raspberry Pi into a powerful stereo, but we're partial to the HiFiBerry DAC+ range. There are cheaper versions, smaller versions, more powerful versions, and all of them improve your Raspberry Pi's sound.

**Use it with:** Stereo system, PA



## HyperPixel 4.0

**magpi.cc/WgSmAA**

**price:** From £37 / $37

The HyperPixel 4.0 is a massive, wonderful screen for Raspberry Pi that you can get with or without a capacitive touch function. It's dead simple to set up and, due to the high pixel density, it's great for watching media or looking at pictures.

**Use it with:** Portable media projects, wall screens

## USB game controller

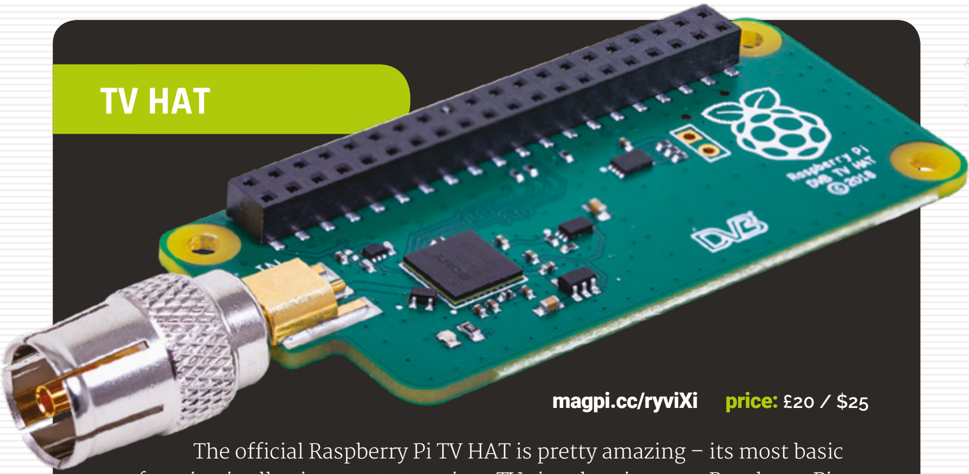
**magpi.cc/2ve6iUo**

**price:** £6 / $7

If you're playing retro games with your Raspberry Pi, you might want a retro controller to help it feel more authentic. We recommend one of the many available in this classic style, with enough buttons to play any game prior to 1996 and some afterwards as well.

**Use it with:** Retro games console, RC car projects

## TV HAT



**magpi.cc/ryviXi** **price:** £20 / $25

The official Raspberry Pi TV HAT is pretty amazing – its most basic function is allowing you to tune into TV signals using your Raspberry Pi. That's pretty standard, but the amazing part is its ability to stream said TV not only around your network, but also over the internet if you're abroad and missing the latest episode of *GBBO*.

**Use it with:** DIY DVR, TV anywhere

## FLIRC USB

**flirc.tv**

**price:** From £20 / $25

For applications such as Kodi, there are plenty of ways to control your Raspberry Pi-based home theatre PC via your smartphone or tablet. FLIRC makes the whole thing a bit simpler, though: take your favourite remote, let FLIRC know which button is which, and then just plug it into your HTPC. Simple.

**Use it with:** Kodi setups, remote-controlled IoT

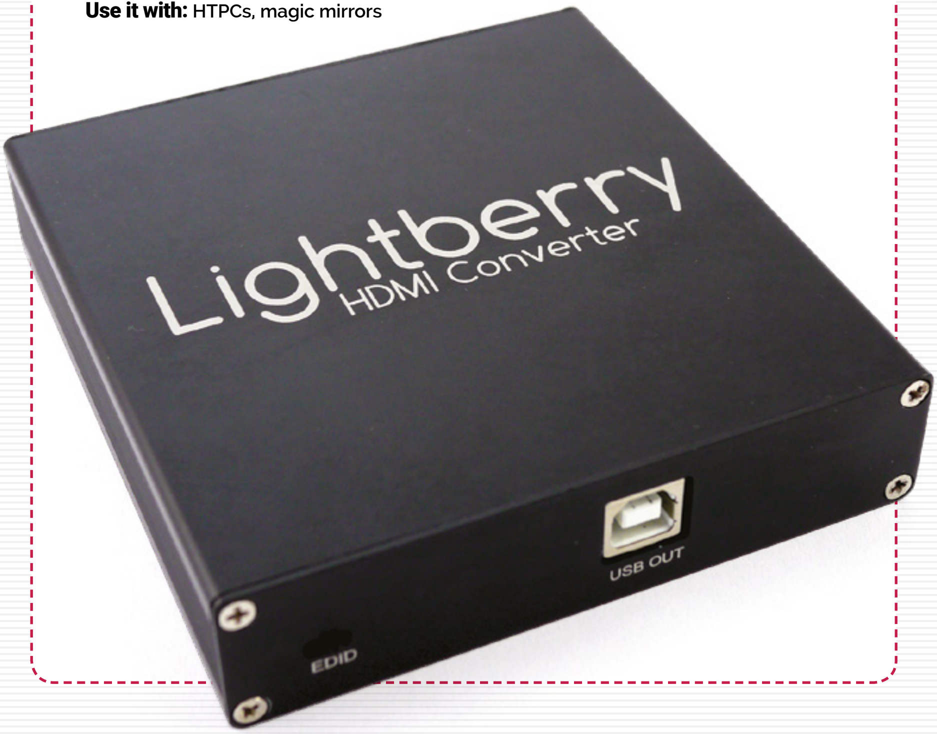
## Lightberry HD

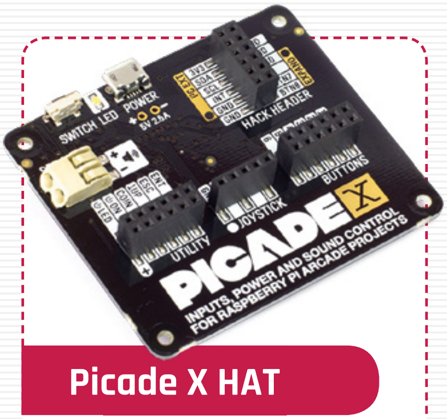
**lightberry.eu**

**price:** From £94 / $114

Ever wanted your films to expand out of your TV? With Lightberry, you can install DIY Ambilight-like effects to your TV so that special effects become just that little bit more special. There are various kits to choose from as well.

**Use it with:** HTPCs, magic mirrors




SWITCH LED
POWER
5V 2A
HACK HEADER
BUTTONS
JOYSTICK
UTILITY
PICADE X
INPUTS, POWER AND SOUND CONTROL FOR RASPBERRY PI ARCADE PROJECTS

## Picade X HAT

**magpi.cc/BupAFF**

**price:** £16 / $16

The Picade is Pimoroni's excellent tabletop arcade kit, and its newest iteration is controlled by the powerful Picade X HAT. It's so good, they made it so you could buy it separately to use in your own DIY Raspberry Pi arcade setup.

**Use it with:** Arcade builds, remote controls

# Watch live TV from Raspberry Pi with the TV HAT

Set up your new add-on to stream live TV from and to a Raspberry Pi

MAKER

Phil King

When not subbing *The MagPi* and editing Raspberry Pi books, Phil enjoys watching a mixture of quality (*Endeavour*, *The Crown*) and other (old *Dynasty* re-runs) TV.

magpi.cc

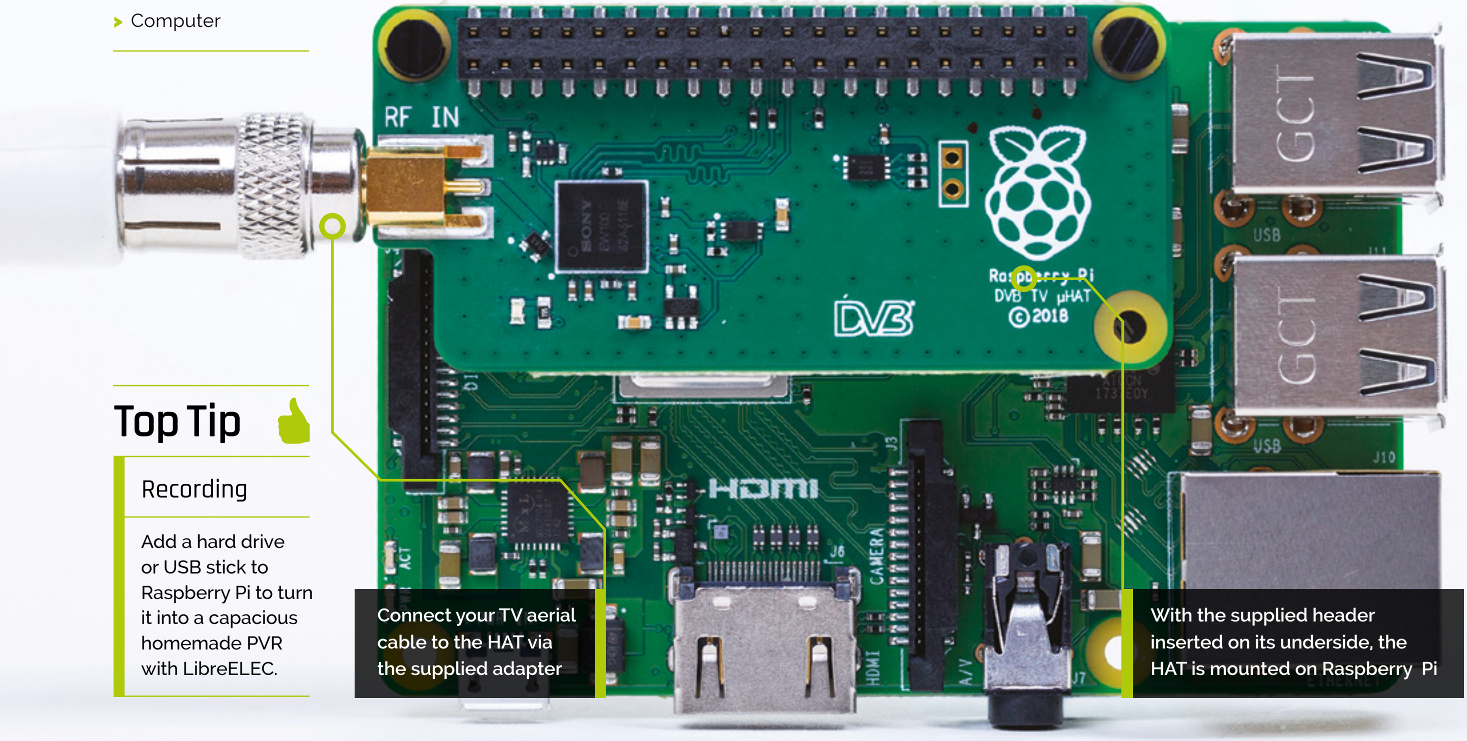
## You'll Need

- Raspberry Pi TV HAT
- Wireless network
- Computer

The latest official Raspberry Pi add-on is a must for telly addicts. The Raspberry Pi TV HAT enables any Raspberry Pi (with 40-pin GPIO) to decode live digital TV signals and stream them to remote devices – such as another Raspberry Pi, computer, or even a smartphone. The HAT's built-in Sony CXD2880 tuner supports DVB-T2 and DVB-T standards (Freeview in the UK), so you can watch all your favourite channels. Here we show you how to set it up, and even use it to turn a Raspberry Pi into an all-in-one TV/PVR.

## 01 Assemble and mount

Insert the thin end of the supplied silver RF cable adapter into the gold connector on the side of the HAT. Take the 40-pin header and insert its pins into the holes in the underside of the HAT – not into the black female header on the top! Mount the HAT on your Raspberry Pi, with the RF adapter at the microSD card slot end. To make it secure, you can screw it in using the supplied stand-offs.



Connect your TV aerial cable to the HAT via the supplied adapter

With the supplied header inserted on its underside, the HAT is mounted on Raspberry  Pi

## Top Tip

### Recording

Add a hard drive or USB stick to Raspberry Pi to turn it into a capacious homemade PVR with LibreELEC.

## 02 Install Tvheadend

Connect a coaxial RF cable from a TV aerial socket to the RF adapter on the HAT. You'll need decent digital terrestrial TV (Freeview in the UK) reception for the tuner to work. With Raspbian Stretch installed on its microSD card, turn on your Raspberry Pi – a green LED will light on the HAT.

We need to install a back-end TV service to stream video: we'll use Tvheadend. To install it, open a Terminal window and enter:

```
sudo apt-get update
sudo apt-get install tvheadend
```

## 03 Configure Tvheadend

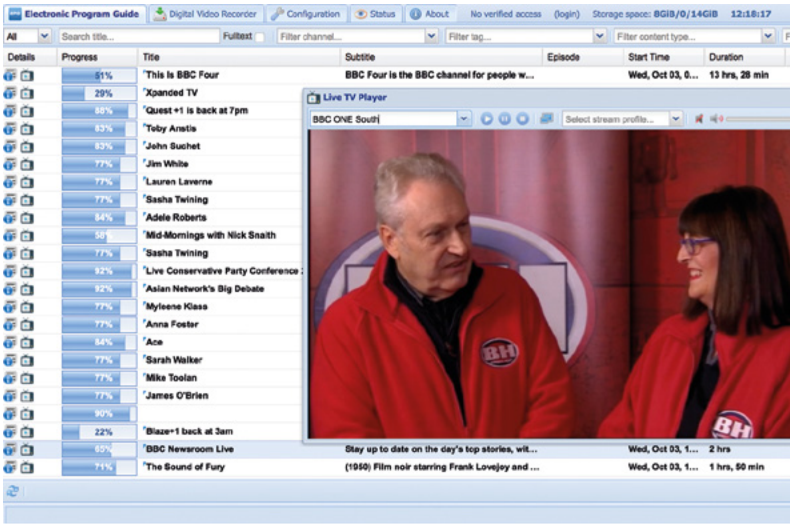
It takes a few minutes for Tvheadend to install, so you may want to go and make a cup of tea. Once installed, a package configuration screen will appear and you'll be prompted to enter a Tvheadend administrator user name. For simplicity's sake, enter **kodi** for this, then also enter **kodi** for the password – you can always change it later if you prefer. You're now ready to log in from another device.

## 04 Log in remotely

Now that Tvheadend is running, you can access it from another computer (possibly a second Raspberry Pi). In a web browser, enter this address: **http://raspberrypi.local:9981/extjs.html**. If that doesn't resolve, use your Raspberry Pi's IP address instead, followed by **:9981/extjs.html**. Log in using **kodi** for both the username and password. A configuration wizard will appear, enabling you to choose the TV network type and transmitter.

## 05 Configure TV network

In the configuration wizard's 'Network settings', it should show 'Tuner: Sony CDX2880 #0 DVB-T #0'. For the Network Type, choose 'DVB-T Network'. For predefined muxes, select your local transmitter: for example, ours was uk-Rowridge. Upon clicking Save & Next, it'll start scanning for muxes and services (channels). After it reaches 100%, tick all three boxes (Map all services, Create provider tags, Create network tags) and an electronic programme guide (EPG) will appear.



| Details | Progress | Title | Subtitle | Episode | Start Time | Duration |
|---|---|---|---|---|---|---|
| | 51% | This Is BBC Four | BBC Four is the BBC channel for people w... | | Wed, Oct 03, 0... | 13 hrs, 28 min |
| | 29% | Xpanded TV | | | | |
| | 88% | Quest +1 is back at 7pm | | | | |
| | 83% | Toby Anstis | | | | |
| | 83% | John Suchet | | | | |
| | 77% | Jim White | | | | |
| | 77% | Lauren Laverne | | | | |
| | 77% | Sasha Twining | | | | |
| | 84% | Adele Roberts | | | | |
| | 58% | Mid-Mornings with Nick Snaith | | | | |
| | 77% | Sasha Twining | | | | |
| | 92% | Live Conservative Party Conference : | | | | |
| | 92% | Asian Network's Big Debate | | | | |
| | 77% | Myleene Klass | | | | |
| | 77% | Anna Foster | | | | |
| | 84% | Ace | | | | |
| | 77% | Sarah Walker | | | | |
| | 77% | Mike Toolan | | | | |
| | 77% | James O'Brien | | | | |
| | 90% | | | | | |
| | 22% | Blaze+1 back at 3am | | | | |
| | 65% | BBC Newsroom Live | Stay up to date on the day's top stories, wit... | | Wed, Oct 03, 1... | 2 hrs |
| | 71% | The Sound of Fury | (1950) Film noir starring Frank Lovejoy and ... | | Wed, Oct 03, 1... | 1 hrs, 50 min |

You can access the EPG and watch live TV in a web browser on a remote computer

## 06 Watching TV

You can now watch live TV streamed from the TV HAT in your browser. In the EPG, click the TV icon to the left of any programme to show it in a window, which you can expand to full-screen. There are also options to schedule and record shows – use the Help option for more details.

As well as using a web browser, you can view the TV stream from a variety of clients, including Kodi, omxplayer, VLC, and smartphone apps – see **magpi.cc/ZpJtwd**.

## Using LibreELEC

The live TV stream from the TV HAT can be accessed in the Kodi media player, most easily by using the LibreELEC OS on a Raspberry Pi. Install the Tvheadend client by selecting Add-ons > Install from repository > PVR clients > Tvheadend HTSP Client.

In addition, using LibreELEC, it's possible to set up both the Tvheadend client and server on the same Raspberry Pi – a Pi 2, 3, 3A+, 3B+, or 4 is recommended – for an all-in-one TV/PVR device. To do so, select Add-ons > Install from repository > Services > Tvheadend Server 4.2, and install it. Access it from another computer using the IP address followed by **:9981** and set it up as in Step 5. You can then access it from within LibreELEC (with the Tvheadend client installed) by selecting the main 'TV' option.

To make TV recordings, you'll need to create a **recordings** folder on a connected USB stick or hard drive – navigate to it in File Manager, right-click, and select 'Make new folder'). Then, in the web interface on a remote computer, go to Configuration > Recordings and set the recording system path to **var/media/[your_drive_name]/recordings**.

## Top Tip

### Live viewing

The live TV stream can be accessed in many ways, including Android and iOS apps.

# Make a Sense HAT MP3 player

Turn a Raspberry Pi and Sense HAT accessory into an MP3 player

MAKER

**Marc Scott**

Marc is a former teacher of Computer Science, and is currently a learning manager at the Raspberry Pi Foundation.

**magpi.cc/ZHdSsd**

## You'll Need

- Raspberry Pi (not a Zero / Zero W)
- Sense HAT
- Speakers or headphones

In this project, you will be making your very own MP3 player using a Raspberry Pi and a Sense HAT. You'll be able to shift between tracks in your playlist, change the volume, and show a cool disco display on the LED matrix.

## 01 Get set up

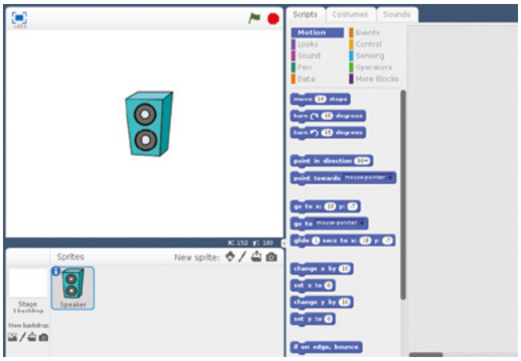
Open up Scratch 2 on your Raspberry Pi (choose Menu > Programming > Scratch 2). Now choose a sprite for your MP3 player. Right-click on the cat sprite and choose Delete. Click on the 'Choose sprite from library' icon; now choose Things in the sidebar and highlight Speaker. Click OK to add it to the Stage. It will add some blocks to the scripts area – right-click to remove them.

## 02 Add GPIO support

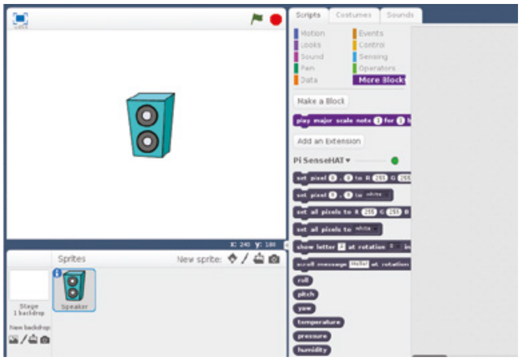
A special GPIO extension in Scratch 2 on Raspberry Pi is needed to control and read input and output components connected to the GPIO pins. With Scratch 2 open, click on More Blocks in the Scripts menu. Next, click on the 'Add an Extension' button. You can then double-click on the 'Pi SenseHAT' extension to add it. You should now see the Pi SenseHAT blocks in the More Blocks section.

## 03 Create a disco effect

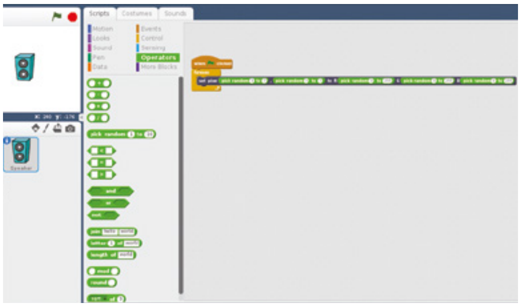
Click Events and drag a `when flag clicked` block to the scripts area. Choose the Control tab and add a `forever` block below `when flag clicked`. Within the More Blocks menu, find the `set pixel` block and drag it inside the `forever` block.



▲ **Step 1** Add the Speaker sprite



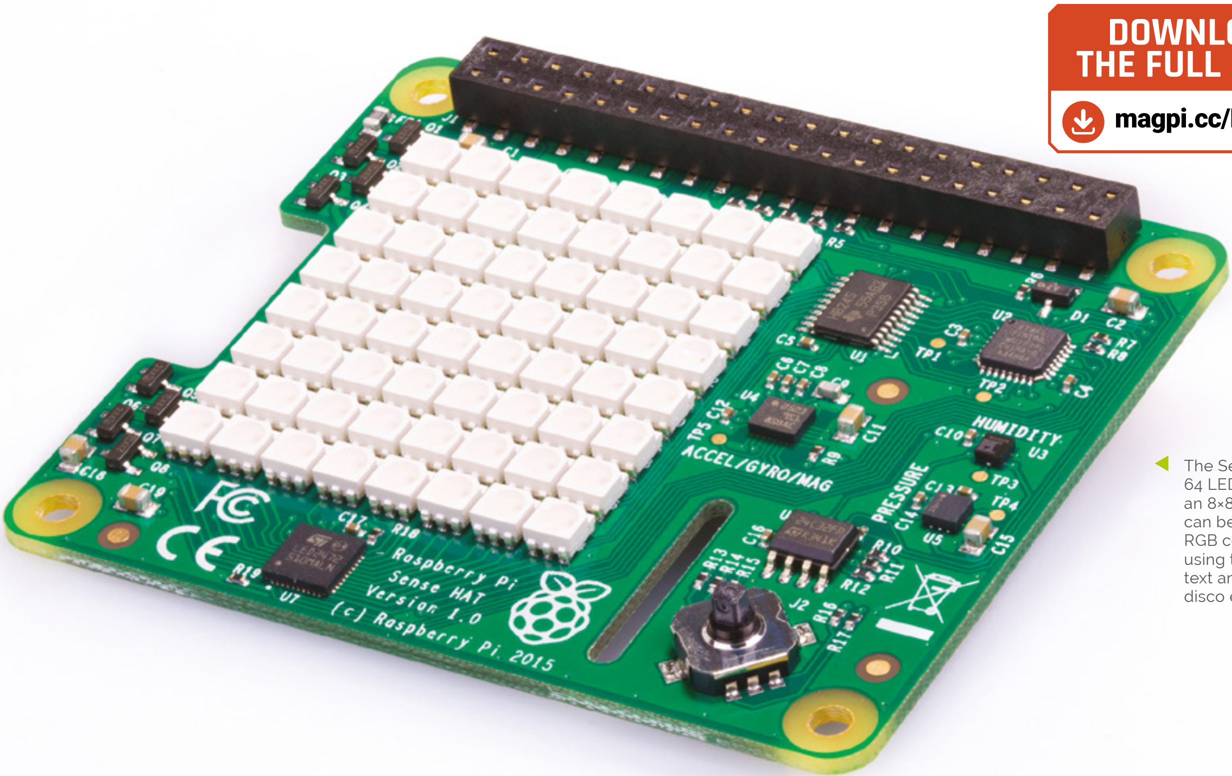
▲ **Step 2** Pi SenseHAT blocks appear under More Blocks



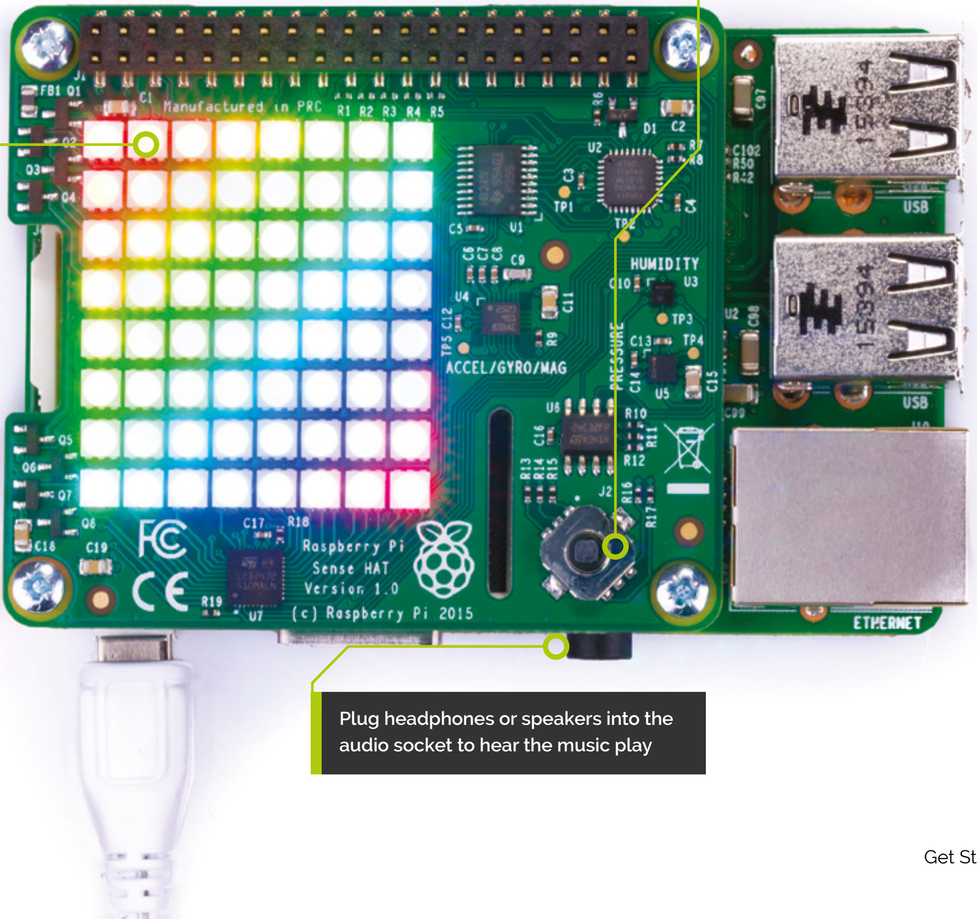
▲ **Step 3** Creating a random sparkle effect

**DOWNLOAD THE FULL CODE:**

magpi.cc/Nqwqmz



The Sense HAT has 64 LED lights (in an 8×8 matrix) that can be set to any RGB colour. We're using them to scroll text and create a disco effect



The LED matrix is used to scroll track information and produce a disco light effect

Use the joystick to switch tracks and adjust the volume

Plug headphones or speakers into the audio socket to hear the music play

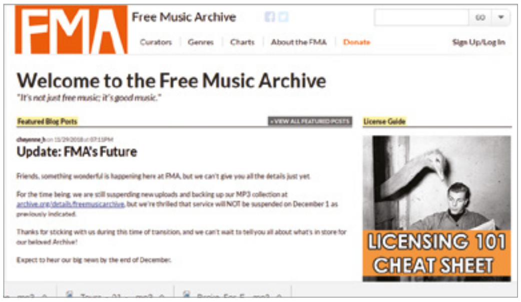
## Top Tip

### Analogue audio

If you're using headphones or speakers, don't forget to change the audio out on your Raspberry Pi from HDMI to Analog. Right-click on the volume icon on the desktop.

**Step 4** Find some music

**Step 5** Import the MP3 files



Free Music Archive

Welcome to the Free Music Archive

"It's not just free music; it's good music."

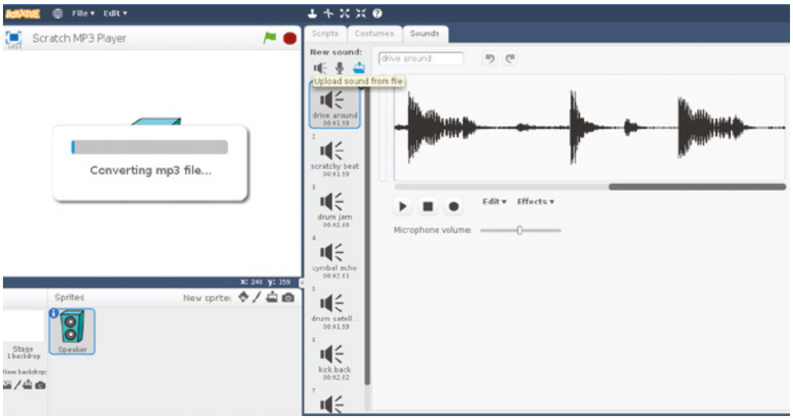
Update: FMA's Future

Friends, something wonderful is happening here at FMA, but we can't give you all the details just yet.

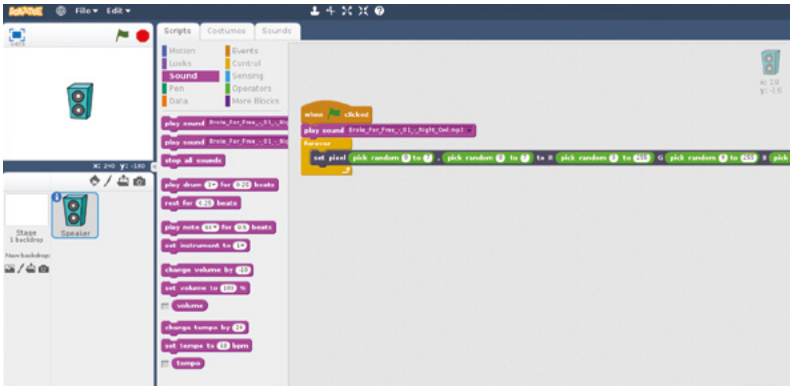
For the time being we are still suspending new uploads and backing up our MP3 collection at archive.org/details/freemusicarchive, but we're thrilled that service will NOT be suspended on December 1 as previously indicated.

Thanks for sticking with us during this time of transition, and we can't wait to tell you all about what's in store for our beloved Archive!

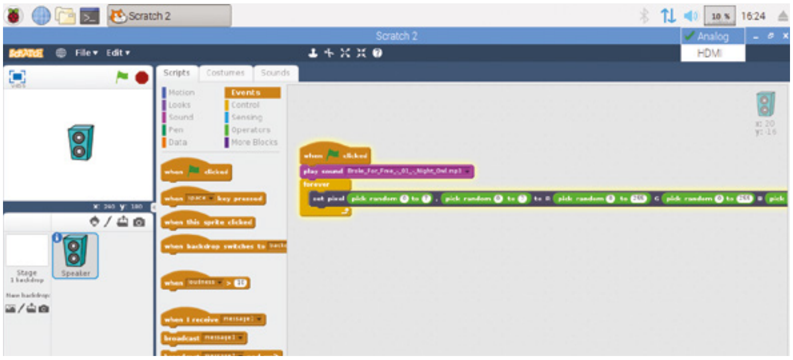
Expect to hear our big news by the end of December.



**Step 6** Add a play sound block



**Step 7** Change Raspberry Pi's audio output to Analog



With the `pick random` block (under Operators), you can set random pixel positions and random colours within the loop to make your Sense HAT sparkle, by choosing LEDs on the Sense HAT to light up (numbers between 0 and 7 on the 8×8 grid), and the amount of red, green, and blue to colour the pixels (numbers between 0 and 255).

Have a go at running your program by clicking on the green flag. You should see the LEDs on the Sense HAT begin to randomly sparkle.

## 04 Get some tunes

Now you have a disco effect, it's time to make your MP3 player play some tunes! First, save the project. Back in Scratch, click File > Save Project > Create Folder. Name the folder **Scratch MP3** and double-click it to navigate inside. Name the file **Scratch MP3 Player.sb2** and click Save.

> It's time to make your MP3 player play some tunes

Gather a few MP3 files together to begin with – you'll want at least three. (We've downloaded three tracks from **freemusicarchive.org**). Save them in the same **Scratch MP3** directory as your Scratch code.

## 05 Add the sounds

Click on the Sounds tab for your sprite. Get rid of all the stock sounds, such as 'scratchy beat' and 'drum jam' (right-click and select Delete).

Now use the 'Upload sound from file' button to upload all your MP3 files. Choose the MP3 files in your Scratch MP3 folder and click Open.

## 06 Face the music

Now it's time to play your tracks. Your code should currently look like that in Step 03. You can play a sound using the `play sound` block. Add this block above the `forever` loop.

Click on the green flag to play your track as the disco lights start. Make sure you have speakers or headphones plugged into your Raspberry Pi.

## 07 Analogue out

At this point you will need to change the sound output to analogue by right-clicking on the speaker icon in the header bar of the desktop and then clicking the Analog option.

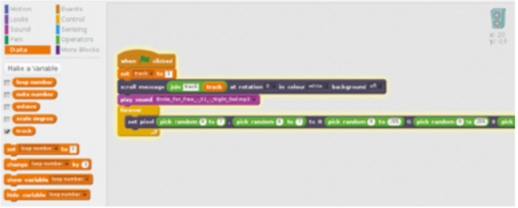
Next, you're going to display the number of the track that's currently playing. Click on Data and Make a Variable; call it 'track' (leave it set for all sprites and click OK).

## 08 Set track

Drag a `set` block above your `play sound` block and choose `track` and 1. Click More Blocks and drag a `scroll message` block under `set track`.

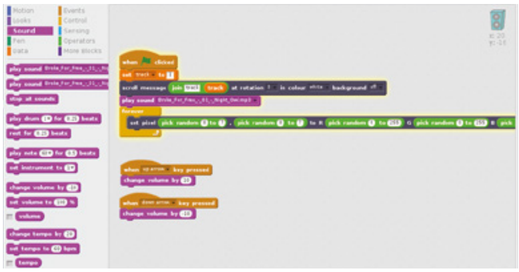
The `scroll message` block can use the `track` variable with a `join` block. Drag a `join` block and enter 'track' in the left field, and drag a `track` variable into the right field (as shown in the image). You can choose whichever colours you like.

**Step 8** Scroll the track number on the LED matrix



```
when [flag] clicked
set track to 1
scroll message (join track (track)) at rotation 0 in colour white background off
play sound [illegible]
forever
    set pixel (pick random 0 to 7) , (pick random 0 to 7) to R (pick random 0 to 255) G (pick random 0 to 255) B (pick
```

**Step 9** Controlling volume with the joystick



```
when [flag] clicked
set track to 1
scroll message (join track (track)) at rotation 0 in colour white background off
play sound [illegible]
forever
    set pixel (pick random 0 to 7) , (pick random 0 to 7) to R (pick random 0 to 255) G (pick random 0 to 255) B (pick

when up arrow key pressed
change volume by 10

when down arrow key pressed
change volume by -10
```

## 09 Volume control

You can now add some extra controls to your MP3 player. First of all, set up your script so that the Sense HAT joystick can control the volume.

Raspberry Pi treats the Sense HAT joystick in the same way as the arrow keys on your keyboard. This means that, for example, pushing the joystick up is the same as pushing the up arrow on your keyboard.

Test your scripts again: click the green flag, and then use the Sense HAT joystick to change the volume of your music.

## 10 Change tracks

Now that you know how to use the joystick, it's time to try to use it to change which track is playing. When the program starts, you will need to stop all sounds.

Use the `when left arrow is pressed` and the `when right arrow is pressed` blocks to `stop all sounds` and to increase or decrease the value of your `track` variable.

You can also use a `scroll message` block to show again which track is being played. You can use the `when arrow is pressed` blocks to change the value stored in your `track` variable.

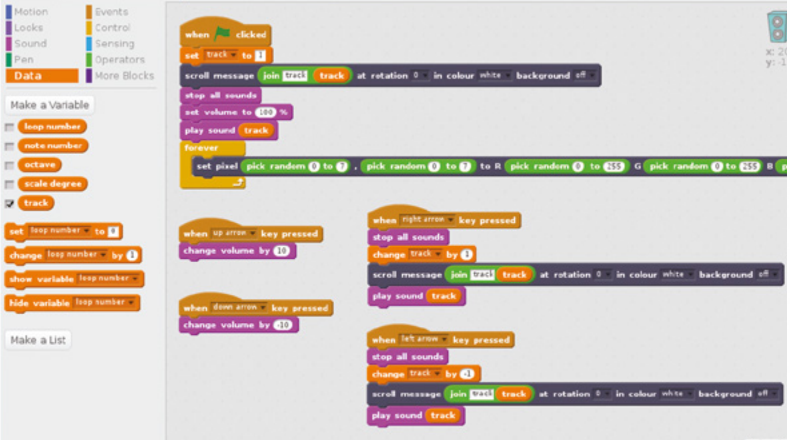
That's it – your MP3 player is complete! Click on the green flag and then have a go at changing your tracks using the Sense HAT joystick.

### Top Tip

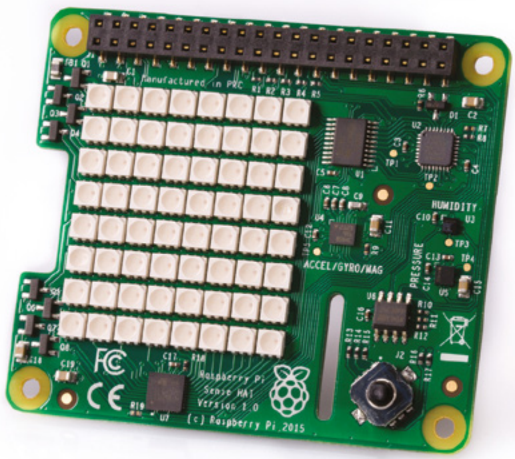
**Shuffle**

A nice feature to have on an MP3 player is the option to shuffle the tracks so that they play in a random order. Pushing the Sense HAT joystick inwards means the same thing to Scratch as pressing **ENTER** on the keyboard. Update the MP3 player code so that when the joystick is pushed, the code chooses and plays a random track.

**Step 10** The complete code

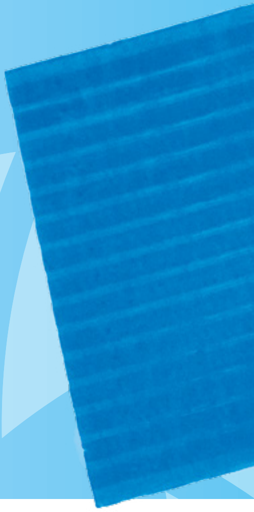


```
when [flag] clicked
set track to 1
scroll message (join track (track)) at rotation 0 in colour white background off
stop all sounds
set volume to 100 %
play sound (track)
forever
    set pixel (pick random 0 to 7) , (pick random 0 to 7) to R (pick random 0 to 255) G (pick random 0 to 255) B (p

when up arrow key pressed
change volume by 10

when down arrow key pressed
change volume by -10

when right arrow key pressed
stop all sounds
change track by 1
scroll message (join track (track)) at rotation 0 in colour white background off
play sound (track)

when left arrow key pressed
stop all sounds
change track by -1
scroll message (join track (track)) at rotation 0 in colour white background off
play sound (track)
```



The Sense HAT is a versatile piece of hardware made by Raspberry Pi. It's called a Sense HAT due to all the sensors on board (hardware and motion), but we're using its other features here

# Add vision with the CAMERA MODULE



A Raspberry Pi Camera Module should be an essential piece of any digital maker's kit, as it can be used in a wide variety of projects. **Wesley Archer** puts us in the picture

Give your Raspberry Pi the gift of sight with an official Camera Module (**magpi.cc/xSXqyL**). This handy add-on connects directly to the Raspberry Pi and is used in a wide range of vision-based projects.

Using a Raspberry Pi to monitor its surroundings is a fascinating subject, especially with the huge array of sensors, accessories, HATs, and pHATs available.

We've seen many exciting builds take advantage of these, but being able to see what is happening around your Raspberry Pi is very simple with the Camera Module. These are some of the best projects for beginners. They're easy to set up and produce visible results.

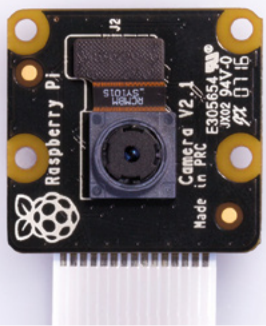
Using the Camera Module is a great way to add an extra dimension to most projects. Let's get started...

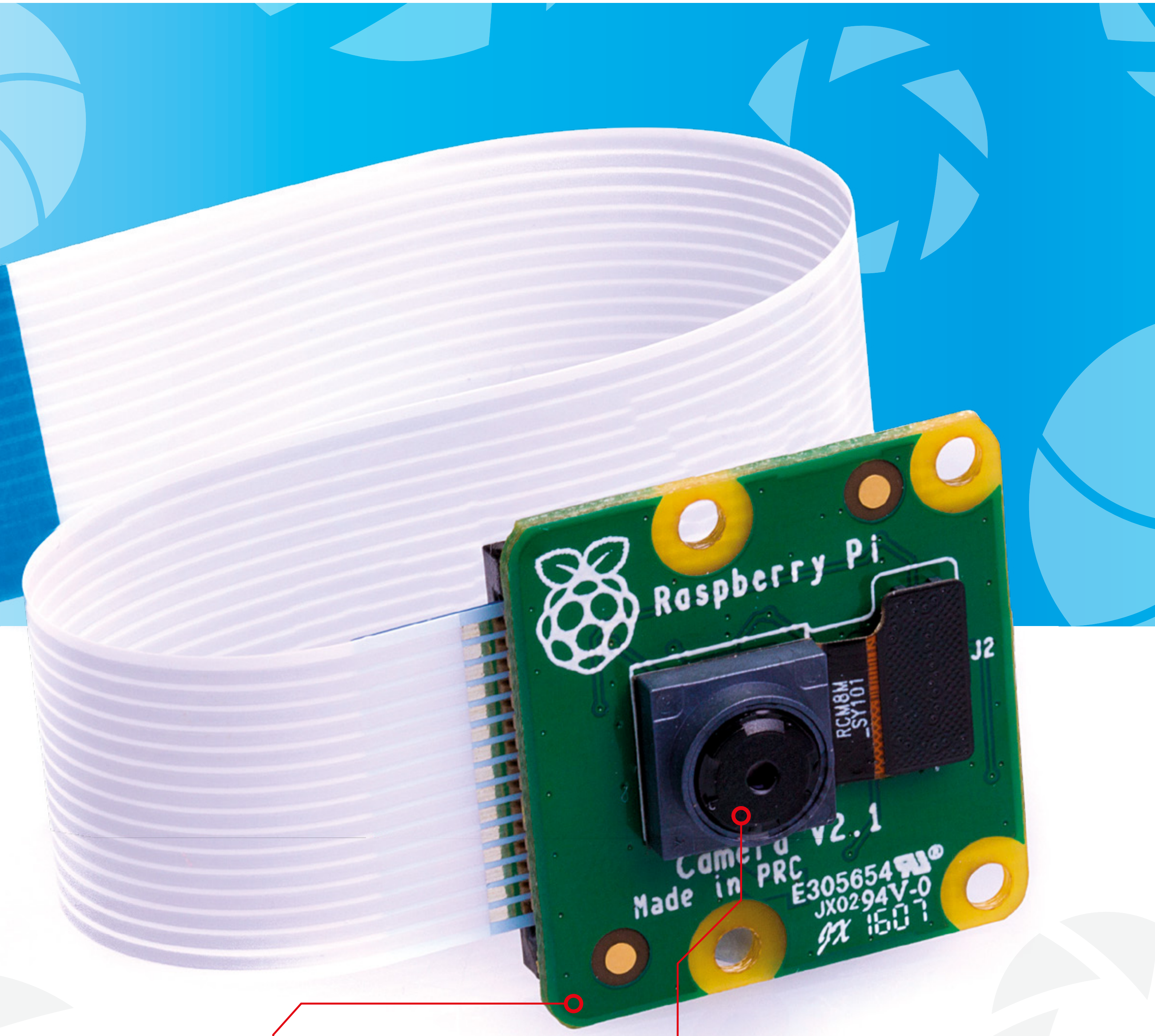
## Regular or night vision?

The Camera Module has two variants – the standard version and the NoIR version, which has the infrared filter removed to allow the camera to detect infrared light. It makes a great night-vision camera when used with an IR light source!

Camera Module V2.1

Camera Module NoIR V2.1





The Camera Module is small enough to fit inside bird-boxes or small project boxes

The small lens captures still images and video, which are saved to Raspberry Pi

# SET UP YOUR CAMERA MODULE

Before you can start snapping away with the Camera Module, you'll need to connect it to a Raspberry Pi and test it out

## You'll Need

- Raspberry Pi Camera Module **magpi.cc/kAtsHn**
- Pi Zero Camera Adapter (optional) **magpi.cc/uAUbdp**
- microSD card
- Raspbian

## Disable LED

The V1 Camera Module lights a red LED when activated (the V2 Camera doesn't have one). To disable this, add `disable_camera_led=1` to the **/boot/config.txt** file (and reboot).

Setting up a Camera Module couldn't be easier. You attach the Camera Module directly to the Camera Serial Interface (CSI) port on the Raspberry Pi board.

Connecting it to your Raspberry Pi is the same whether you own a Raspberry Pi 3A+/3B+, 4, Zero, or even an earlier Model B with 26 GPIO pins.

However, if you are going to connect a Camera Module to a Raspberry Pi Zero (1.3, W, or WH), you'll need to use a Pi Zero Camera Adapter. This is often sold separately (£4, **magpi.cc/uAUbdp**), and sometimes included with the Official Raspberry Pi Zero case (**magpi.cc/LECcmF**).

If you received a Raspberry Pi Zero W free with a 12-month subscription to *The MagPi* (**magpi.cc/subscribe**), you will have got a Zero Camera Adapter cable free with your Starter Kit.

Once you've got your Camera Module and Raspberry Pi, you use the cable to attach the two together. We'll show you how...

## 1 Connect the cable to a Raspberry Pi 2/3/3+/4

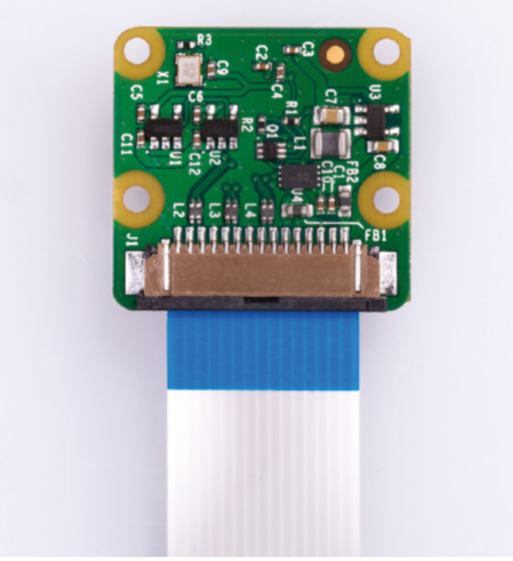
On a full-size Raspberry Pi, you will need to connect the camera cable to the Camera port nearest the Ethernet port. Gently lift the small plastic clasp on the Camera port and connect the camera cable with the blue part facing towards the Ethernet port, then gently push the clasp back down so it holds the cable in place.

## 2 Connect the cable to a Raspberry Pi Zero W

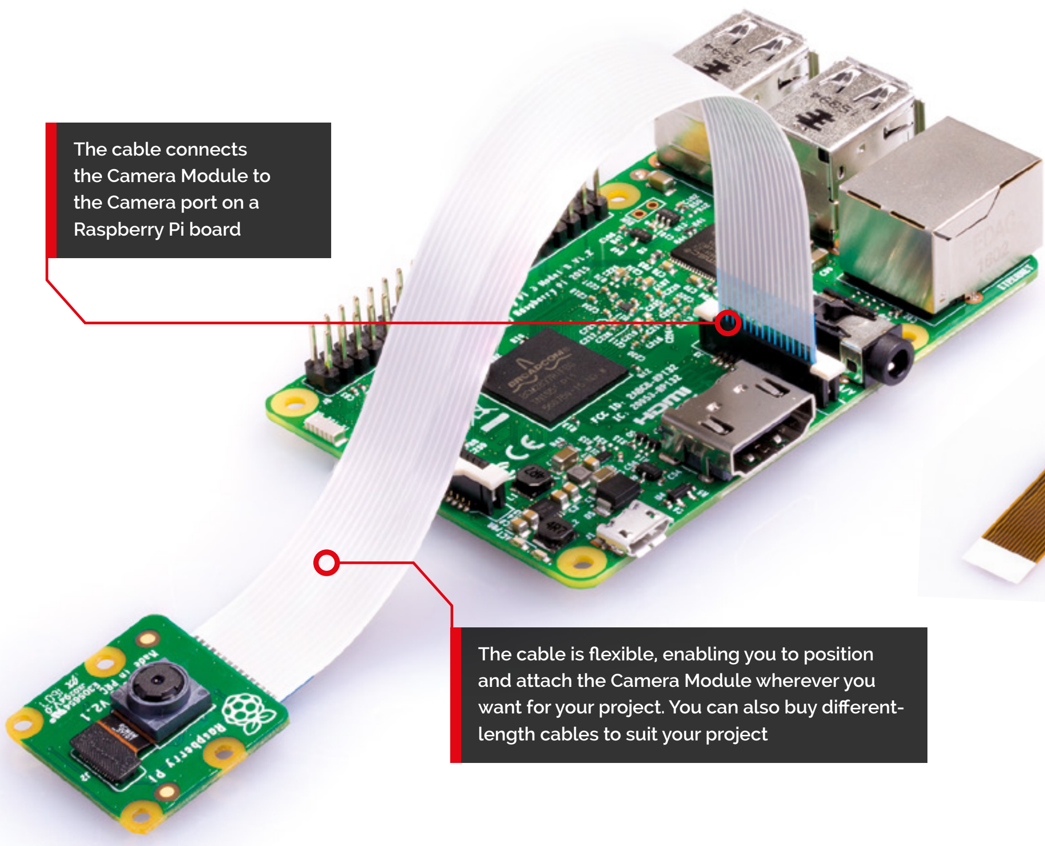
The Raspberry Pi Zero W has a smaller Camera port, but the official Raspberry Pi Zero case usually includes the cable. As with a full-size Raspberry Pi, gently pull the plastic clasp and then slot the camera cable in – you should see the white part facing upwards. You'll know if the cable is in correctly if the camera works, but it is best to check before booting your Raspberry Pi.

## 3 Connect the Camera Module

Similar to the way the camera cable connects to Raspberry Pi, gently pull the small plastic clasp on the Camera Module and then slot the other end of the camera cable into the port – this time the blue part should be facing you, with the camera lens pointing the opposite direction. Gently close the clasp, locking the cable in place – the clasps can be fragile, so handle with care.

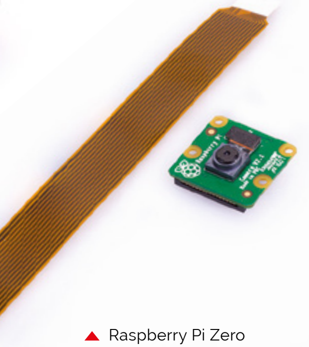


While small, the Camera Module takes high-resolution photos (up to 3280×2464 pixels) and captures HD video (up to 1080p) without breaking a sweat



The cable connects the Camera Module to the Camera port on a Raspberry Pi board

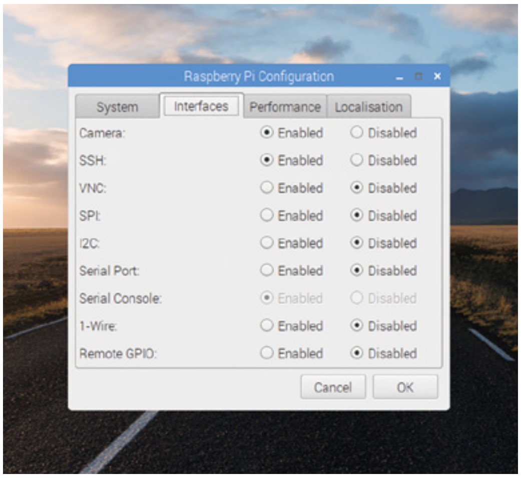
The cable is flexible, enabling you to position and attach the Camera Module wherever you want for your project. You can also buy different-length cables to suit your project



▲ Raspberry Pi Zero owners will need a Pi Zero Camera Adapter to connect the Camera Module to the smaller Camera socket on the board

## 4 Enable the Camera Module

By default, the Camera Module is not enabled in Raspbian, but it is very easy to do this. If you have your Raspberry Pi connected to a mouse, keyboard, and monitor, you can use the Raspberry Pi Desktop to enable the Camera function. Click on the raspberry icon (top left) and choose Preferences > Raspberry Pi Configuration. Click the Interfaces tab and set Camera to Enabled. Click OK; when prompted to reboot, click Yes.



Raspberry Pi Configuration

System | Interfaces | Performance | Localisation

| | | |
|---|---|---|
| Camera: | ● Enabled | ○ Disabled |
| SSH: | ● Enabled | ○ Disabled |
| VNC: | ○ Enabled | ● Disabled |
| SPI: | ○ Enabled | ● Disabled |
| I2C: | ○ Enabled | ● Disabled |
| Serial Port: | ○ Enabled | ● Disabled |
| Serial Console: | ● Enabled | ○ Disabled |
| 1-Wire: | ○ Enabled | ● Disabled |
| Remote GPIO: | ○ Enabled | ● Disabled |

◀ You must activate the Camera interface in Raspberry Pi Configuration before using the camera

## 5 Enabling via SSH

If you are accessing your Raspberry Pi over SSH, you'll need to enable the camera via the command-line Raspberry Pi configuration tool instead. Enter the following in a Terminal window:

```
sudo raspi-config
```

Now select '5 – Interfacing Options > P1 – Camera' and choose <Yes> to enable the camera. Press **ENTER** on <Ok> and choose <Finish>. When asked to reboot, choose <Yes>. We're now set to snap away.

## 6 Test the Camera Module

Before diving into a project, it's worth testing your Camera Module works as expected first. Open a Terminal and enter this command:

```
raspistill -o /home/pi/hello.jpg
```

Smile for a few seconds – you can see yourself on the live camera preview – while the photo is taken and saved to the **/home/pi** directory. Why not open it in File Manager to see the results?

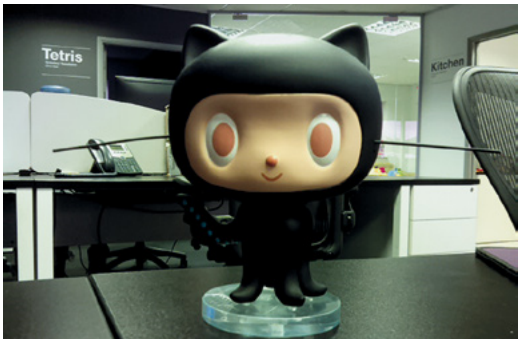
### Raspberry Pi Camera Case

Using a full-sized Raspberry Pi? Why not buy a Raspberry Pi case designed to accommodate the Camera Module, such as the PiShell?

# CAMERA MODULE SETTINGS

Get the most out of your Camera Module by tweaking the various settings and creating different effects

The Camera Module doesn't just snap photos – it also records video – but did you know that there are lots of different settings that can be applied to customise how the photo or video appears? In this guide, we're going to explore some of the settings and commands that can be used to tailor how you use the Camera Module in your project, and we'll explore some of the more quirky features too!

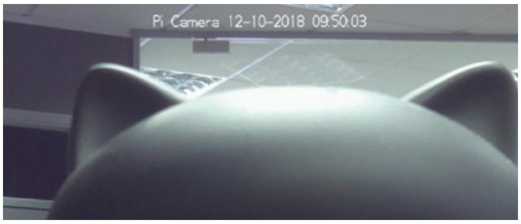


## 1 Adjusting resolution and brightness

The v2.1 Camera Module has a maximum resolution for a photo of 3280×2464 pixels, but images this large can take up a lot of storage space over time and take longer to process by Raspberry Pi. Fortunately, raspistill, the program that captures photos on our Raspberry Pi, allows us to specify the photo resolution very easily. For example, you can set the width (640 pixels) and height (480 pixels) using the `-w` and `-h` parameters accordingly. You can also specify the brightness with the `-br` parameter and a value between 0 and 100. The image above was taken with the NoIR Camera Module (hence the odd colours) using:

```
raspistill -w 640 -h 480 -br 40 -o /home/
pi/hello.jpg
```

This demonstrates how you can take a smaller photo (640×480 pixels) with a brightness of 40, but feel free to experiment to see what works best.



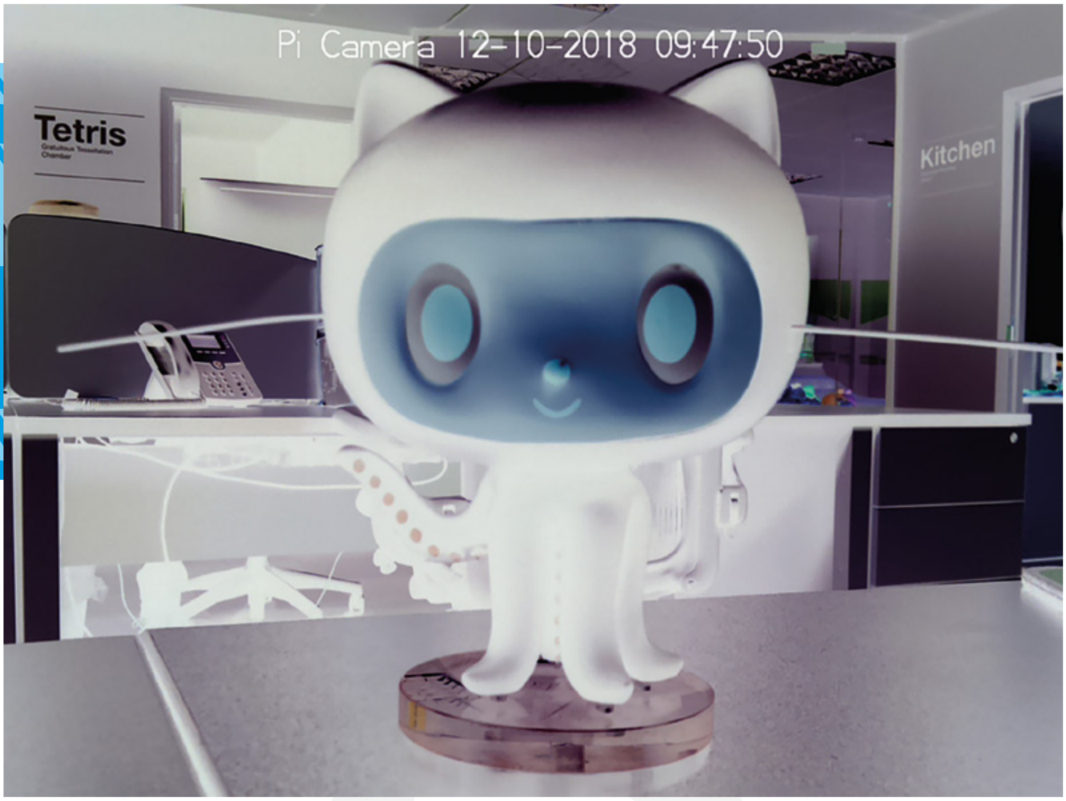
## 2 Adding text

If you're old enough to remember actual printed photographs, then you may have noticed that some cameras added the date and time to record when the photo was taken. If you're too young to remember, don't worry as you've just learned something new! With raspistill you can add your own text to your photos too, using the `-a` or `--annotate` parameter. Take the following example:

```
raspistill -a 4 -a "Pi Camera %d-%m-%Y
%X" -o /home/pi/text.jpg
```

The first `-a 4` part sets the bitmask notation in the metadata to record the time (more info on that here: **magpi.cc/2egcYup**).

The second `-a` part includes our custom text (Pi Camera) and formats the date into dd-mm-yy (`%d-%m-%Y`) before adding the time (`%X`) and then saving the photo in our Raspberry Pi's home directory (**/home/pi**) as **text.jpg**.



"You can add image effects to photos taken by the Camera Module, such as inverting the colours to a negative"

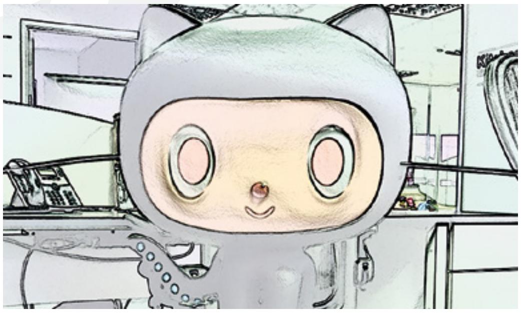
## 3 Add image effects

Another quirky feature of raspistill is that you can add image effects to photos taken by the Camera Module, such as inverting the colours to a negative, making the photo look like a sketch, or even adding a cartoon effect. Now, before you all get carried away, some of the effects can be a little hit-and-miss, but they can add a cool look to your photos nonetheless! This is done by adding the `-ifx` parameter along with the desired effect (full breakdown available here: **magpi.cc/SuuECJ**).

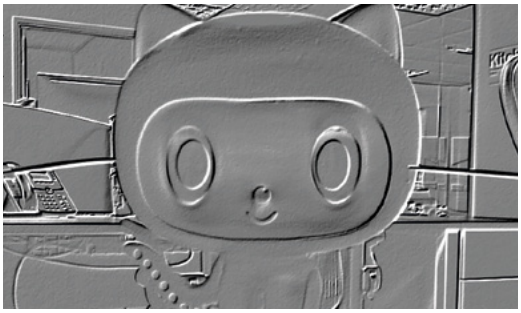
The following example applies a negative effect to the photo, as well as adding our custom text, date, and time:

```
 raspistill -ifx negative -a 4 -a "Pi
Camera %d-%m-%Y %X" -o /home/pi/negative.
jpg
```

As you can see from the image above, this is an interesting effect! Be sure to try out the sketch and emboss commands too.



Make the photo look like a watercolour painting instead of a negative with the `-ifx watercolour` parameter

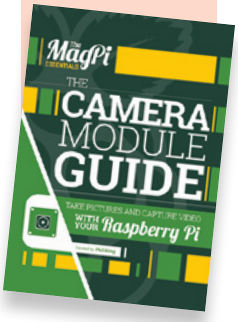


Embossing is the process of creating a raised image, usually in paper or card. You can recreate the effect with the `-ifx emboss` parameter

## Camera Essentials

We've only just scratched the surface of what the Camera Module can do in our guide, so why not take a look at *The MagPi Essentials Camera Module Guide* for a more comprehensive overview of all the commands, hints, and tips to ensure you get the most out of it?

**magpi.cc/camera-essentials**



# CAMERA MODULE PROJECTS

We've selected six Camera Module-related projects to try out. If you need some inspiration, then take a look at the pick of the bunch

## Raspberry Pi Zero LED case hack

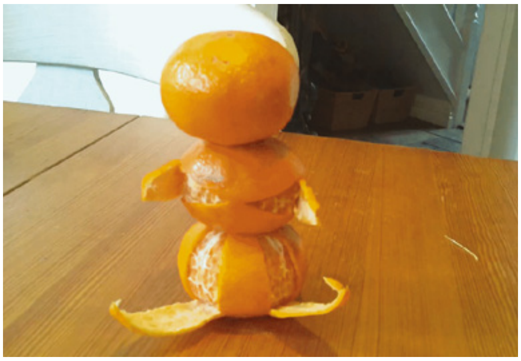
As you now know, the NoIR version of the Camera Module has the infrared filter removed from the lens so that it can detect infrared light. This makes it an ideal choice for projects in the dark. In this neat project, you can easily add IR LEDs to provide additional infrared light, so that your pictures capture more detail in the dark – and all housed within the official Raspberry Pi Zero case, too!

More info: magpi.cc/xCozST

## Stop-motion animation

Stop-motion animation – think Wallace & Gromit – is where a series of photographs are taken, but with minor changes made to the 'scene' being filmed. The scene is changed slightly between shots – known as frames – and when they are all stitched back together afterwards, the illusion of animation is complete! In this fantastic tutorial, learn how to add a button to your Camera Module setup and create your very own stop-motion animation!

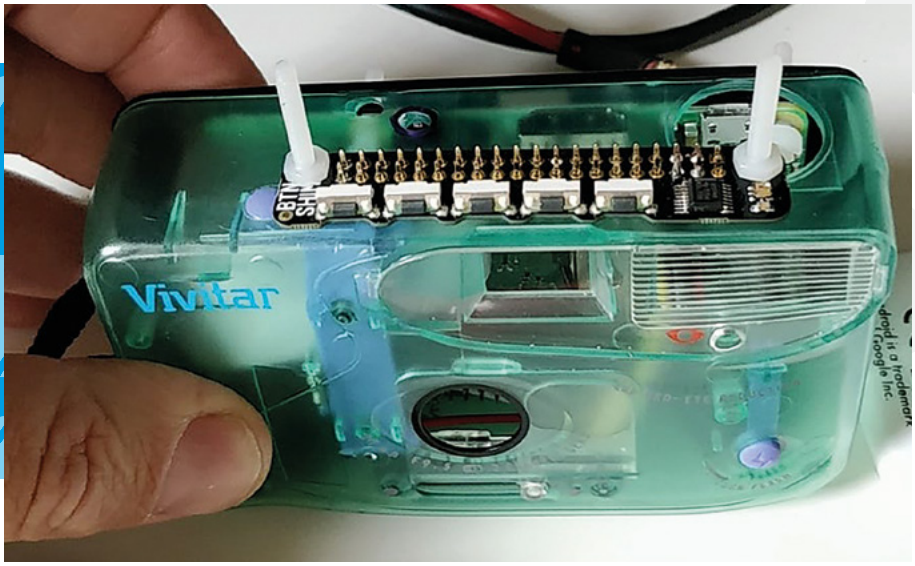
More info: magpi.cc/KiYbdb



## Rodak: a Raspberry Pi in a Kodak

Taking photos is what the Camera Module is designed for, so what better way to do that than to upcycle an old Kodak Brownie Flash III camera to house a Raspberry Pi Zero, like Alex Ellis did? It also contains some additional components, such as a LiPo battery for power-on-the-go, a push-button to trigger the camera, and a cool NeoPixel status LED to show you when the camera is ready to fire again!

More info: magpi.cc/sAxjyc

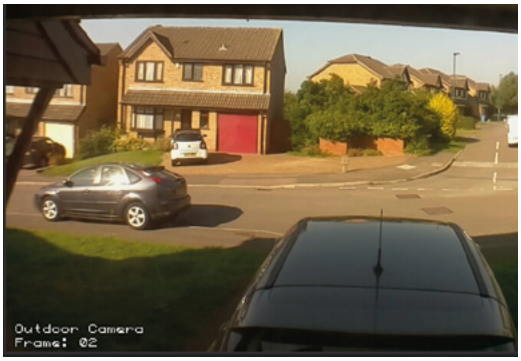


"Easily add IR LEDs to provide additional infrared light, so that your pictures capture more detail in the dark!"

## Google Vision: Camera Tell

This cool project, by Dan Aldred (aka TeCoEd), makes use of the Google Vision cloud service. If you haven't heard of Google Vision, it uses artificial intelligence to analyse an image and tells you what is in it! Dan takes this project even further by creating a portable camera – housed inside an old disposable camera – that uses a Camera Module, as well as Google Vision to announce what you have just taken a photo of.

More info: magpi.cc/VadAHy



## Home CCTV with motionEyeOS

If you missed the Home Security feature in *The MagPi* 73 (**magpi.cc/73**), you should definitely check out the handy motionEyeOS, which is a standalone operating system that runs on Raspberry Pi and allows you to easily create your very own home CCTV setup with a few clicks. Once the OS image is flashed to your microSD card, you will be up and running in no time and be able to keep an eye on your home when you are away.

More info: magpi.cc/mpLqjp

## Make a parent detector

Do you suspect your parents sneak into your room when you're out? Do they play on your Xbox or PS4 without your permission? Then follow this tutorial and build a parent detector, using the Camera Module and a PIR (passive infrared) sensor, and catch them red-handed! Learn how the PIR sensor can trigger the Camera Module to take a photo whenever it detects movement, then see how your parents react when confronted with the evidence.

More info: magpi.cc/nlaODg

# Pictures from space via ham radio

Have you ever wanted to receive a radio signal from space? It's fun and a lot easier than you might think!

MAKER

Dave Honess

Dave lives in the Netherlands and works at the European Space Research and Technology Centre, where he helps coordinate the educational operations for European crew on the ISS.

@dave_spice

### You'll Need

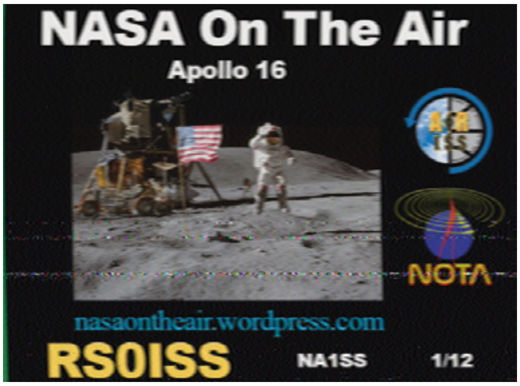
- Raspberry Pi 2B or later
- 3.5 mm speakers or headphones
- RTL-SDR USB dongle (search for 'RTL2832U') with a good whip or dipole antenna (or just get this starter kit: **magpi.cc/uBMKQg**)
- Tape measure, at least 1 metre

**What is Slow Scan TV (SSTV), then? You convert a picture into sound (similar to fax machine or dial-up modem noise) transmit the sound via radio, and convert the received sound back into a picture at the other end.** The International Space Station (ISS) has a long history of transmitting SSTV pictures for global events, each lasting several days, where people the world over tune their radios to the ISS and try to receive the pictures every time it flies over their home. It's a hobby of mine!

One such event began on **12 April 2019** to celebrate the anniversary – to the day – of the launches of Yuri Gagarin in 1961 and the first Space Shuttle, 20 years later, in 1981.

## 01 Install prerequisites and set the time

We presume you are running a clean install of Raspbian and are online.



This is one of the clearest pictures your author has been able to receive with this setup. It was from an 83 degree elevation pass near Amsterdam, in the Netherlands, on 15 February 2019

Firstly, switch the audio output to the 3.5 mm headphone jack. This is needed for the way PulseAudio loops back the system audio as an input device. On the Raspbian desktop, simply right-click the speaker icon in the menu bar and choose Analog.

Then, open a Terminal and install the required software for the project:

```
  sudo apt-get update
  sudo apt-get install rtl-sdr sox
 pulseaudio qsstv ntpdate -y
```

Next, set your time zone and pull the correct time from the internet. In the Raspbian main menu, go to Preferences > Raspberry Pi Configuration > Localisation tab > Set Timezone, change Area and Location accordingly, then click OK.

Finally, go back to the Terminal window to pull the time:

```
  sudo ntpdate pool.ntp.org
```

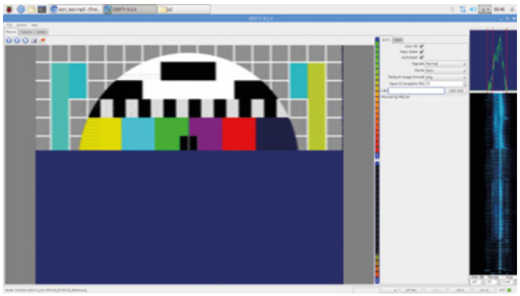
Note that the desktop clock in the top right only updates every minute.

## 02 Install and test the RTL-SDR dongle

Insert the RTL-SDR dongle into your Pi and screw in the antenna (it can sit on your desk at this stage). Then verify the RTL-SDR is working with the rtl_test program. In a Terminal window, enter:

```
  rtl_test
```

If you see a 'failed to open rtlsdr device' error, remove the dongle, reinsert it, and try `rtl_test`



▲ It takes about two whole minutes for a picture to be decoded. When they do this from the ISS, a picture is transmitted continuously for about two minutes, with two minutes of off-time before the next picture starts

again. Leave for about 30 seconds and look out for any messages about loss of samples. Press **CTRL+C** when 30 seconds has elapsed. The loss of 100 bytes or so is acceptable.

## 03 Listen to a commercial FM radio station

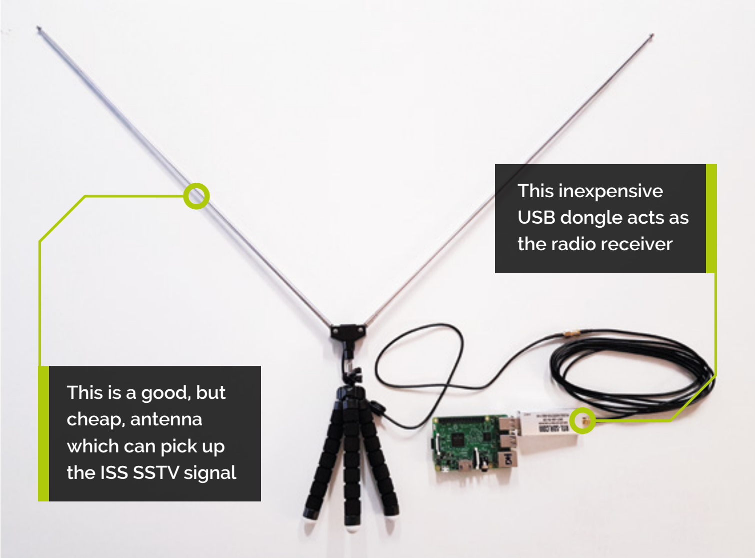
Let's tune into a commercial FM radio station to check that both the dongle and the antenna are working correctly. The command below pipes raw data from rtl_fm into 'play', which will then produce the audio output. Modify the value after the `-f` to specify your own FM station/frequency and don't forget to put the capital M after the number so it knows you mean megahertz and not hertz (98.8MHz is BBC Radio 1 in the UK). In a Terminal, enter:

```
  rtl_fm -M wbfm -f 98.8M | play -r 32k -t
raw -e s -b 16 -c 1 -V1 -
```

Note that the 'Tuned to' value in the output doesn't match. This is due to a property of all SDR hardware, called 'DC spike', and can be ignored. Plug in your speakers/headphones and you should now be able to hear audio from the commercial FM station. Move the antenna around or select a different FM radio station with a closer transmitter if the audio is noisy. Press **CTRL+C** to quit from rtl_fm.

## 04 Set up QSSTV

QSSTV is the program that will listen to the audio output on the Raspberry Pi and decode the pictures from the fax/modem noise when the ISS is in range. Let's just check the settings are all correct. In the main menu, select Internet > QSSTV, then select Options > Configuration and choose the following options in the Sound tab:



This inexpensive USB dongle acts as the radio receiver

This is a good, but cheap, antenna which can pick up the ISS SSTV signal

**Audio Interface** – PulseAudio
**Input and Output Audio Device** – default -- Playback/recording through the PulseAudio sound server
**Sound Input** – From sound card
**Sound Output** – To sound card

Click OK. Next, run the Calibrate function and go make a cup of tea/coffee: Options > Calibrate. Click OK when both progress bars reach 99%.

## 05 Decode a picture from test audio

Now that QSSTV is set up, we can download a test audio file, play it, and see a picture slowly appear. In a Terminal, enter:

```
  wget https://raw.githubusercontent.com/
davidhoness/sstv_decoder/master/sstv_test.mp3
```

Select the Receive tab in QSSTV and choose the following settings:

**Use VIS** – on
**Auto Slant** – on
**Autosave** – on
**Signals** – Normal
**Mode** – Auto

### Top Tip

**Play with SSTV on mobile**

For a quick test, you can try these free mobile apps: Robot36 on Android; CQ SSTV on iOS. Play this MP3 (**magpi.cc/ruANWC**) and put the phone near the speaker.

## Top Tip

### RTL-SDR starter kit

Your author personally recommends this starter kit that comes with the USB dongle and a good v-dipole antenna: **magpi.cc/uBMKQg**

Click the play ▶ button. Nothing will happen, but just click it to start the receiver.

Open **sstv_test.mp3** in the Chromium browser. You can either drag and drop the file from File Manager, or enter **file:///home/pi/sstv_test.mp3** into the address bar.

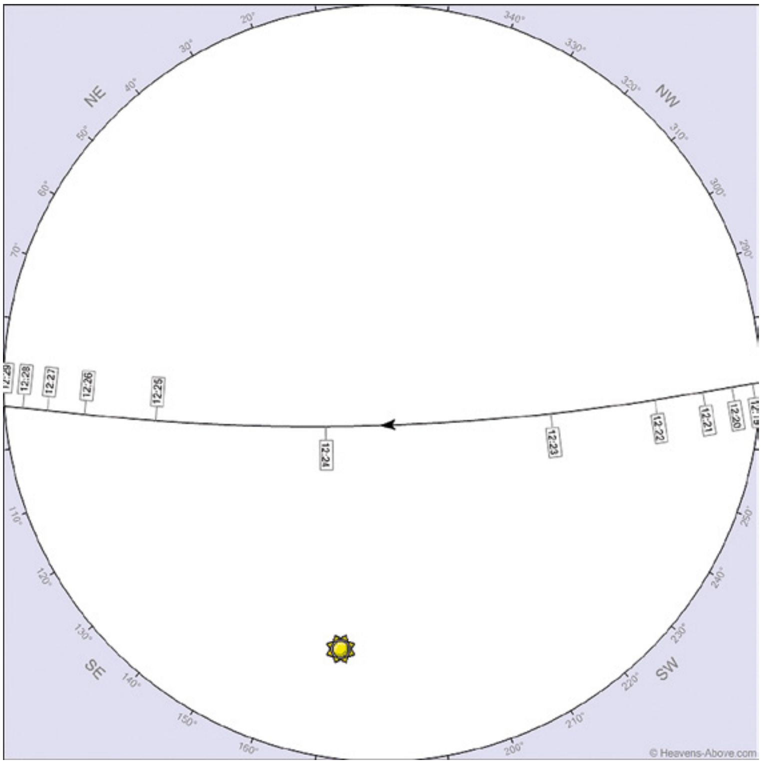
Let it play and watch the picture appear in QSSTV. Notice the waveform and waterfall display activity on the right of the screen. Other SSTV test recordings can be found online, such as: **magpi.cc/dXpjRr**.

## 06 Prepare for the next SSTV event

To get pictures from the ISS you need to know three things:

- When the ISS will be transmitting the SSTV signal. It's not always-on.
- What frequency the SSTV signal will be transmitted on (usually 145.8MHz).
- When the ISS will fly over your location.

The ISS flies over most places a few times every day (unless you live in the Arctic/Antarctic). You can find out when the ISS is transmitting on this website: **ariss-sstv.blogspot.com**.



▼ Imagine holding this picture above your head and aligning it with the compass directions. The line shows the flight path of the ISS across the sky above you

At the time of writing, for instance, an SSTV event was scheduled to start on **4 August 2019** to honour NASA astronaut Owen Garriott, the first person to operate amateur radio from space. These events usually last for a few days at a time, meaning you will get multiple ISS passes during which you can receive the SSTV signal.

## 07 When is the ISS flying over me?

There are many websites you can use for this, but your author's personal favourite is **heavens-above.com**.

Click 'Unspecified' in the top right to set your location. Type your postal code and country into 'Enter place to search for' and click Search. Scroll down and click Update. You'll now be back on the homepage; under Satellites, click ISS. For 'Passes to include', select 'all'.

The table now showing lists the times of the upcoming ISS passes for your location. The Alt value under 'Highest point' is the maximum elevation of the pass above the horizon; 90 would mean it goes directly overhead. Passes with higher maximum elevation will give better results than lower ones. Clicking on any row in the table shows the sky chart for that pass. Imagine holding that picture above your head and aligning it with the compass directions. The line shows the flight path of the ISS and this is useful for roughly positioning your antenna. It can also be useful to look at the 'Ground track' (see link in top right). Plan ahead and set up your ground station with time to spare before the ISS flies over.

## 08 Adjust your antenna

If you are using the RTL-SDR kit antenna, please read the getting started guide for it: **magpi.cc/qzUKtZ**.

It's a good idea to adjust the length of the telescopic arms based on this dipole calculator (**magpi.cc/uGNNxP**) to match the antenna's resonant frequency to 145.8MHz. This will help pick up weaker signals from low elevation passes.

Under Desired Frequency, enter 145.8. Under Select Antenna Calculation, select One Side. Click Calculate. The answer should be 48.9 cm.

Remember, there is about 2 cm of metal in each side of the antenna base. So take 48.9 and, before screwing in the arms, use a tape measure to make them both this length minus 2 cm (excluding the screw threads). It doesn't have to be perfect, but make it as close as possible.

## 09 Deploy your antenna

As a general rule, try to get your antenna up as high as possible with a good view of the sky in all directions. Outside is always better, but you can still get the signal from inside a house – you'll just see more noise in the pictures. Then, with the dipole arms flat, roughly align them with the flight path of the ISS that you find on **heavens-above.com** and bend the two arms upwards into a right-angle V-shape. You don't need to realign it for every pass, but it will yield better results if you do.

Antennas and their performance are sometimes a bit of an art form as opposed to an exact science. Experiment with different configurations on different ISS passes. Try one arm horizontal and one arm vertical, for instance.

## 10 Set the correct time again

QSSTV saves the decoded images with a UTC time file name and this is useful later when working out which ISS passes they are from. In a Terminal, enter:

```
sudo ntpdate pool.ntp.org
```

## 11 Tune into the ISS frequency

We'll use rtl_fm again for this, but note that the command is different to the commercial radio station one. A commercial radio station uses *wide band* FM whereas the ISS SSTV transmission uses *narrow band* FM and so we have to set rtl_fm up differently. In a Terminal, enter:

```
rtl_fm -M fm -f 145.8M -s 48k | play -r 48k -t raw -e s -b 16 -c 1 -V1 -
```

You should be able to hear white noise now (don't forget the capital M for megahertz), but this will change to the fax/modem noise when the ISS flies over with the transmission active. Keep this window open. A picture is transmitted from the ISS continuously for about two minutes, with another two minutes of off-time before the next picture starts.

▲ The ARISS VHF antenna installed on the bottom of the Columbus module *Credit: NASA/ESA*

## 12 Start QSSTV

In the Raspbian main menu, select Internet > QSSTV. Select the Receive tab in QSSTV and choose the following settings:

**Use VIS** – on
**Auto Slant** – on
**Autosave** – on
**Mode** – Auto
**Save if Complete (%)** – 10

Click the play ► button. The waveform and waterfall display on the right should show noise coming from rtl_fm. Leave QSSTV running.

> " Antennas and their performance are sometimes a bit of an art form as opposed to an exact science "

## 13 What is Doppler shift?

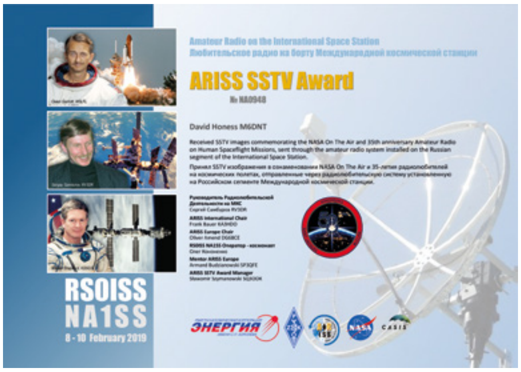
A common example of Doppler shift is the change of pitch heard when a police car or ambulance speeds past you. Compared to the emitted frequency of the siren, the frequency you hear is higher during the approach, identical at the instant of passing by, and lower during departure. The same thing happens with radio waves as with sound waves. The ISS is moving at 7 km per second!

To compensate for this, ground stations must continually retune their receiver as the ISS approaches (starting ~3.5kHz above 145.8MHz), passes overhead (reaching 145.8MHz exactly), and flies away (ending ~3.5kHz below 145.8MHz). The good news is that QSSTV is clever and can compensate by tracking the centre frequency of the audio signal as the ISS flies over. So you don't need to do anything!

### Top Tip

**Antennas to avoid**

Avoid the little 30 cm mag-mount antennas that often come with these RTL-SDR dongles. You will have little or no hope of picking up the ISS with them.



Amateur Radio on the International Space Station
Любительское радио на борту Международной космической станции

ARISS SSTV Award

№ NA0948

David Honess M6DNT

Received SSTV images commemorating the NASA On The Air and 35th anniversary Amateur Radio on Human Spaceflight Missions, sent through the amateur radio system installed on the Russian segment of the International Space Station.

Принял SSTV изображения в ознаменовании NASA On The Air и 35-летия радиолюбителей на космических полетах, отправленные через радиолюбительскую систему установленную на Российском сегменте Международной космической станции.

Руководитель Радиолюбительской Деятельности на МКС
Сергей Самбуров RV3DR

ARISS International Chair
Frank Bauer KA3HDO

ARISS Europe Chair
Oliver Amend DG6BCE

RS0ISS NA1SS Оператор - космонавт
Олег Кононенко

Mentor ARISS Europe
Armand Budzianowski SP3QFE

ARISS SSTV Award Manager
Slawomir Szymanowski SQ3OOK

RS0ISS
NA1SS
8 - 10 February 2019

▲ You can potentially get a certificate like this, even if you only decode a few lines of one picture

## 14 Compensate for Doppler shift (optional)

It doesn't make much difference, but if you want to have some fun with this, your author has created a Python script to compensate for Doppler shift. It tracks the ISS using ephem, computes the corrected frequency for when the ISS is passing over, and retunes rtl_fm using a UDP socket: **magpi.cc/zNtRDB**.

To use it you'll first need to run `sudo pip3 install ephem` and then just start the script in another Terminal window. But first you will need to modify the code to set your location. See the `LATITUDE`, `LONGITUDE`, and `ALTITUDE` variables at the top. Get the values from the top right of **heavens-above.com** and just estimate your altitude.

## 15 Wait for the ISS to arrive

You may want to leave your ground station overnight or over the weekend to capture several passes. When you return, select the Gallery tab in QSSTV to see what images were received. Alternatively, it's always exciting to watch the pictures coming in live as the ISS is above you!

## 16 Get an SSTV award!

On your Raspberry Pi, the decoded pictures can be found in **/home/pi/qsstv/rx_sstv**. You can upload them on this website for verification and recognition: **magpi.cc/Kdjnmy**.

Check the information for the event on **ariss-sstv.blogspot.com** to find out how to apply for the award certificates. This will be a PNG file that is emailed to you and will show a unique number. Good luck!

## doppler.py

▶ Language: **Python**

```python
#!/usr/bin/python3
import urllib.request
import math
import socket
import ephem
import time
import sys
import ssl


C = 300000000.0
F0 = 145800000.0

LATITUDE = "52.219308"
LONGITUDE = "4.419926"
ALTITUDE = 20


class tle_reader(object):
    """
    For keeping ephem two line element sets up to date
    """
    def __init__(self,
                 tle_name="ISS (ZARYA)",
                 tle_file=
"https://celestrak.com/NORAD/elements/stations.txt",
                 tle_max_age=3600):
        self._tle_name = tle_name
        self._tle_file = tle_file
        self._tle_max_age = tle_max_age
        self._tle = None
        self.reload()

    def build_index(self, tle_lines):
        index = {}
        for i in range(0, len(tle_lines), 3):
            index[tle_lines[i].strip()] = (tle_lines[i +
1], tle_lines[i + 2])
        return index

    def reload(self):
        print("Loading: %s" % self._tle_file)

        try:
            ctx = ssl.create_default_context()
            ctx.check_hostname = False
            ctx.verify_mode = ssl.CERT_NONE
            with urllib.request.urlopen(self._tle_file,
context=ctx) as response:
```

DOWNLOAD THE FULL CODE:

magpi.cc/zNtRDB

```
                tle_lines = response.read().decode("utf-8").splitlines()
            index = self.build_index(tle_lines)
            tle_data = index[self._tle_name]
            self._tle = ephem.readtle(self._tle_name, tle_data[0], tle_data[1])
        except Exception as e:
            print(e)

        self._tle_age = time.time()

    @property
    def tle(self):
        return self._tle

    @property
    def tle_expired(self):
        return time.time() - self._tle_age > self._tle_max_age


class rtl_fm_remote(object):
    """
    For remote control of rtl_fm command line program
    """
    def __init__(self,
                 host="localhost",
                 port=6020):
        self._host = host
        self._port = port
        self._s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self._s.connect((self._host, self._port))

    def set_freq(self, freq):
        self.send_cmd(0, freq)

    def set_mode(self, mode):
        self.send_cmd(1, mode)

    def set_squelch(self, squelch):
        self.send_cmd(2, squelch)

    def set_gain(self, gain):
        self.send_cmd(3, gain)

    def send_cmd(self, cmd, param):
        cmd_bytes = (cmd).to_bytes(1, "little")
        param_bytes = (param).to_bytes(32, "little")
        self._s.send(cmd_bytes + param_bytes)

    def __del__(self):
        self._s.close()


rtl = rtl_fm_remote()
iss = tle_reader(tle_name="ISS (ZARYA)", tle_max_age=5520)

if iss.tle is None:
    sys.exit(0)

myloc = ephem.Observer()
myloc.lon = LONGITUDE
myloc.lat = LATITUDE
myloc.elevation = ALTITUDE

freq = F0
running = True

try:
    while running:
        myloc.date = time.strftime('%Y/%m/%d %H:%M:%S', time.gmtime())
        iss.tle.compute(myloc)
        alt = math.degrees(iss.tle.alt)

        if alt > 0:  # iss is flying over location
            new_freq = int(F0 - iss.tle.range_velocity * F0 / C)  # doppler
            if new_freq != freq:
                print(new_freq, round(alt, 2), myloc.date)
                rtl.set_freq(new_freq)  # set new frequency in rtl_fm
            freq = new_freq
        elif iss.tle_expired:
            iss.reload()  # we could be running for days / weeks
        else:
            time.sleep(10)  # do nothing, wait for iss to arrive
            freq = F0
except KeyboardInterrupt:
    running = False

print("Bye")
```

# THE 50 BEST RASPBERRY PI TIPS

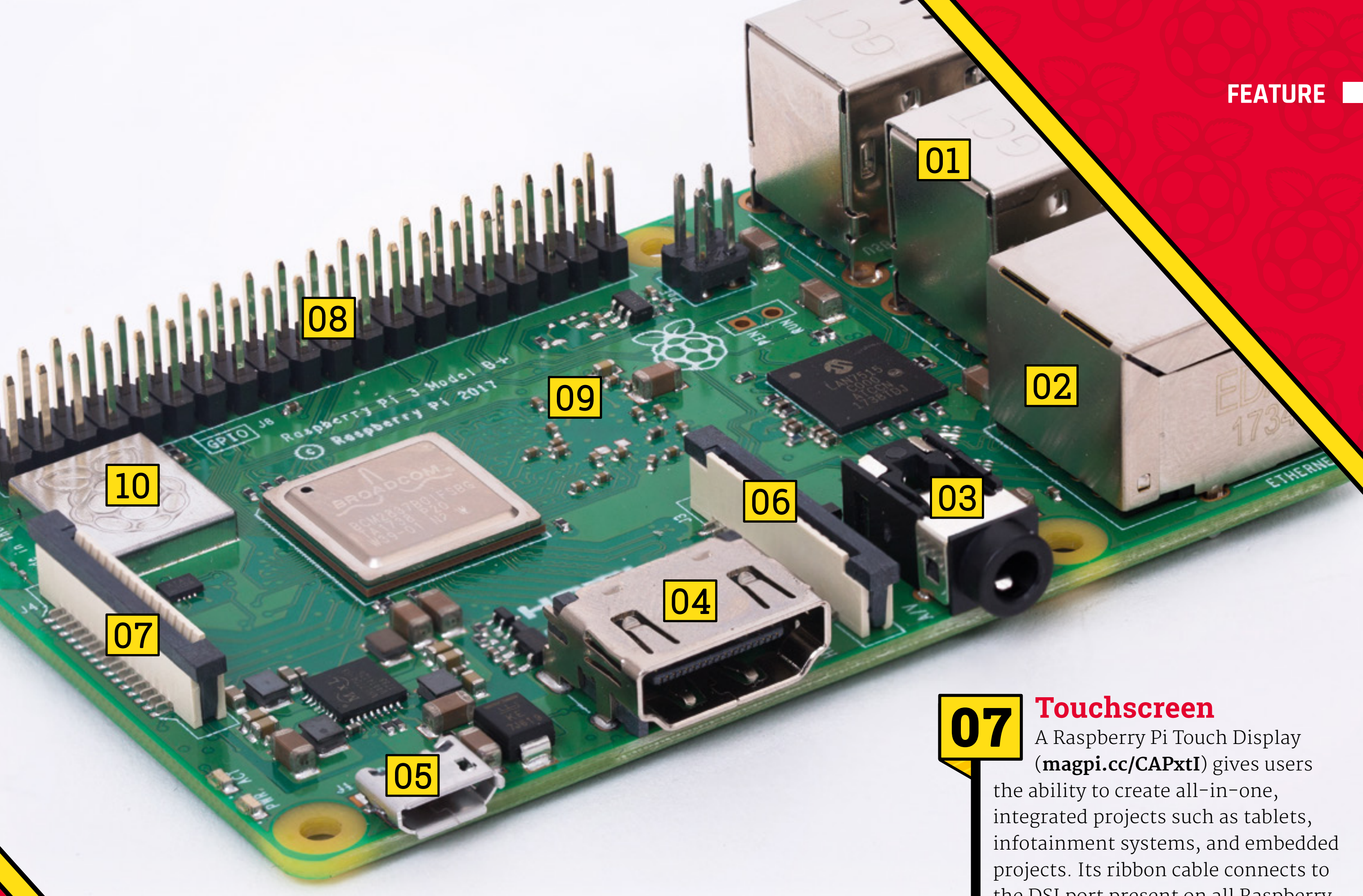
MASTER RASPBIAN AND RASPBERRY PI WITH THESE HACKS, HINTS & TRICKS

## 01 Boot from USB

You can boot some Raspberry Pi models (3B, 3B+, 3A+, 2B v1.2) from a USB port instead of a microSD card. It's faster and more reliable for long-term use, plus you get lots of extra storage space. See the official documentation at **magpi.cc/eXHfjq**.

## 02 Power over Ethernet

The official Raspberry Pi PoE HAT enables you to power a Raspberry Pi 3B+ or 4 via an Ethernet cable. Great for those projects that require a constant wired connection.



### 03 Audio output

All Raspberry Pi models have a composite out port to connect analogue devices. Model B+ and later combine the audio out and composite out on to the same 3.5 mm jack plug. This requires a particular type of lead, with audio left on the tip, audio right on ring 1, ground on ring 2, and video on the sleeve. See **magpi.cc/tDfjLV**.

### 04 Switch sound options

The HDMI port outputs audio as well as video. If your display lacks a built-in speaker, you'll want to use the speaker jack. If so, right-click on the volume icon in the menu bar and switch from HDMI to Analog. Alternatively, use the command line:

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

Set the last number to '1' for Analog, or '2' for HDMI. Use raspi-config to change the audio setting at boot:

```
sudo raspi-config
```

Choose Advanced Options > Audio and 'Force 3.5 mm ('headphone') jack' or 'Force HDMI' to set this as the default.

### 05 Power it up

The power requirements of your Raspberry Pi increase as you make use of the various interfaces. The GPIO pins can draw 50 mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16 mA. The HDMI port uses 50 mA, the Camera Module requires 250 mA, and keyboards and mice can take as little as 100 mA or quite a bit more. Buy the Official Power Supply and check the power rating of your devices: **magpi.cc/xfszUR**.

### 06 Use a camera

You can use a webcam, but Raspberry Pi has a dedicated option: the Camera Module (**magpi.cc/jbKzbf**). This is attached to the CSI (Camera Serial Interface). The Camera Module is used to take high-definition video and still images. It's easy to use for beginners, but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries Raspberry Pi has developed to create visual effects.

### 07 Touchscreen

A Raspberry Pi Touch Display (**magpi.cc/CAPxtI**) gives users the ability to create all-in-one, integrated projects such as tablets, infotainment systems, and embedded projects. Its ribbon cable connects to the DSI port present on all Raspberry Pi boards (except Raspberry Pi Zero and Zero W).

### 08 GPIO pins guide

Open a Terminal and enter:

```
pinout
```

...for a quick visual guide to the GPIO pins. Or enter:

```
pinout -x
```

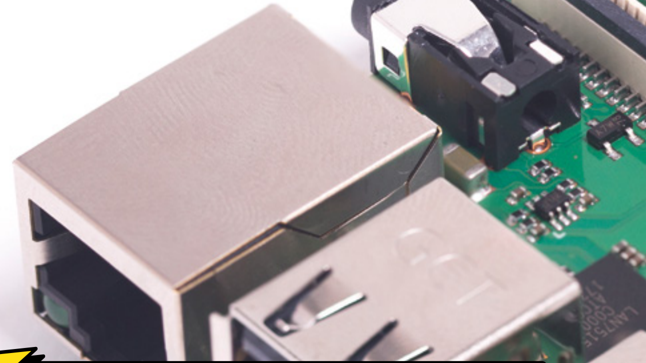
...to open the web browser and point it to the interactive model at **pinout.xyz**.

### 09 Hardware diagrams

You can see the mechanical drawings for Raspberry Pi 3B+ (and all other models) on the Raspberry Pi website: **magpi.cc/WWFUji**.

### 10 Wireless hotspot

Many Raspberry Pi models feature a wireless LAN adapter to connect to local network. But you can also use the access point to create a wireless hotspot. The process is a little long, but it's great for learning more about how wireless LAN works. Read more at **magpi.cc/nnriRT**.



# RASPBIAN TIPS

Customise Raspberry Pi's operating system with these tips and tricks.

**By Lucy Hattersley**

## 11 Demo programs

Raspbian comes with a bunch of demo programs that demonstrate various features. Find them here:

```
cd /opt/vc/src/hello_pi
```

They're written in C. Compile them with:

```
./rebuild.sh
```

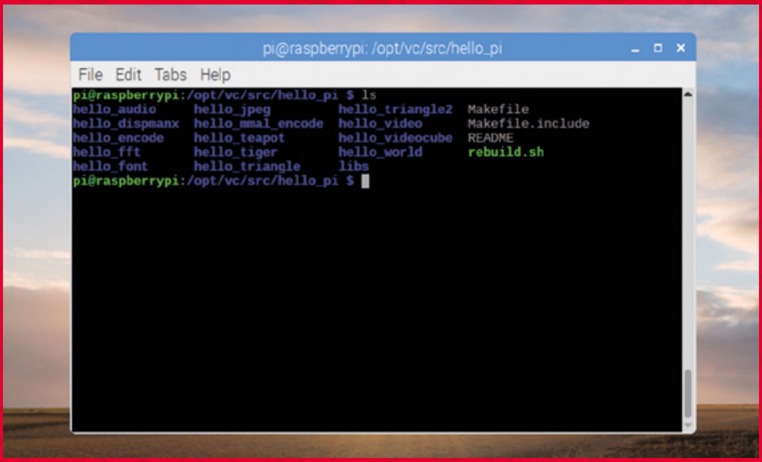
You will now be able to run the .bin files in each example folder. For instance, run:

```
cd hello_tiger
./hello_tiger.bin
```

...to see a large rotating image of a tiger on the screen. For more info, see **magpi.cc/tpNaGV**.

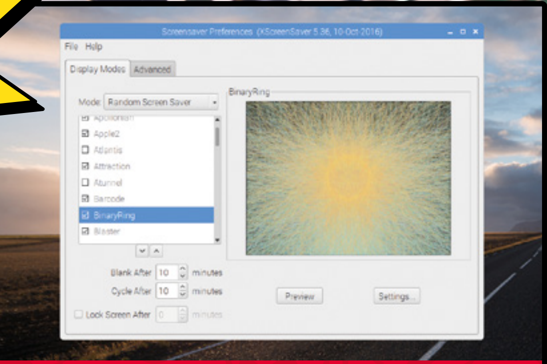
## 12 Screensaver

By default, the Raspberry Pi Desktop doesn't have easy-to-use screensaver software installed. Firstly, you should install the X Windows screensaver application.



```
pi@raspberrypi:/opt/vc/src/hello_pi $ ls
hello_audio     hello_jpeg          hello_triangle2  Makefile
hello_dispmanx  hello_mmal_encode   hello_video      Makefile.include
hello_encode    hello_teapot        hello_videocube  README
hello_fft       hello_tiger         hello_world      rebuild.sh
hello_font      hello_triangle      libs
pi@raspberrypi:/opt/vc/src/hello_pi $
```

▼ Raspbian has a selection of demo programs hidden in the **opt** directory



Mode: Random Screen Saver

☒ Apollonian
☒ Apple2
☐ Atlantis
☒ Attraction
☐ Atunnel
☒ Barcode
☒ BinaryRing
☒ Blaster

Blank After 10 minutes
Cycle After 10 minutes
☐ Lock Screen After 0 minutes

▶ Add some visual flair to the Raspbian interface using the Screensaver program

```
sudo apt-get update
sudo apt-get install xscreensaver
```

This may take a few minutes. The screensaver application will then appear under Preferences in the main desktop menu. This provides many options for setting up the screensaver, or disabling it.

## 13 Understand config.txt

Note: this is for advanced users only! Raspberry Pi uses a configuration file instead of the BIOS found on most PCs. This file is normally accessible as **/boot/config.txt** from Linux, and must be edited as root.

```
sudo nano /boot/config.txt
```

Any changes will only take effect after you have rebooted your Raspberry Pi. After Linux has booted, you can view the current active settings using the following commands:

```
vcgencmd get_config <config>
```

...displays a specific config value; for example, `vcgencmd get_config arm_freq`.

```
vcgencmd get_config int
```

...lists all the integer config options that are set (non-zero).

```
vcgencmd get_config str
```

...lists all the string config options that are set (non-null).

A range of options can be edited in the **config.txt** file – see **rpf.io/configtxt**.

# SECURITY TIPS

You'll need to think about security when connecting Raspberry Pi to the outside world

## 14 Free up space

You can save space on your microSD card for projects, documents, and code by removing some of the programs provided in Raspbian by default. You can remove software from the command line:

```
sudo apt purge mathematica
sudo apt clean
sudo apt autoremove
```

But it's much better to choose Menu > Preferences > Recommended Software. Clear the tick next to the unwanted program and click OK.

## 15 Kernel update (& rollback)

If you use the standard Raspbian update/upgrade process, this will automatically update the kernel to the latest stable version. This is the recommended procedure. However, sometimes you may wish to update to the latest 'bleeding edge' or test kernel.

```
sudo rpi-update
```

The rpi-update utility will download the latest (unstable, testing) kernel version and copy all required files onto your system. Note that the latest kernel from rpi-update is not guaranteed to work correctly! Make sure that it doesn't conflict with your distribution packages. It doesn't provide a way of automatically uninstalling the files.

To revert to the current stock Raspbian kernel after trying rpi-update or a custom kernel, you need to reinstall both these packages, by running:

```
sudo apt-get install --reinstall
raspberrypi-bootloader raspberrypi-
kernel
```

## 16 Change your password

With the standard install of Raspbian, the default username is **pi** and the password is **raspberry**. If you haven't at least changed this password, anyone can log into your Raspberry Pi! Go to Menu > Preferences > Raspberry Pi Configuration. In the System tab, click Change Password, enter one, confirm it, and click OK.

## 17 Keep Raspbian up to date

From time to time, security vulnerabilities are found in software, so it's always best to get the latest versions on a regular basis. In a Terminal, type `sudo apt-get update` to update your package lists, then `sudo apt-get dist-upgrade` to get the latest version of the packages on your system.

## 18 Install Fail2ban

If someone wants to hack into a Raspberry Pi, they may try to guess your username and password. It will probably take a lot of attempts, but this is called 'brute-forcing'. To prevent this, you can install a program called Fail2ban. Install using `sudo apt install fail2ban` and a user will be banned for ten minutes if they fail login five times.

## 19 Change default SSH port

If you change the default SSH port, anyone trying to connect would need to know which port to use. To alter it, `sudo nano /etc/ssh/sshd_config` and change the line #Port 22 to read Port 2222 (without the #) or another memorable number. Save and exit (**CTRL+O**, **ENTER**, **CTRL+X**), then restart SSH with `sudo service ssh restart`.

## 20 Turn off interfaces you don't need

The safest way to stop hackers is to not have any network connection, but another way of shutting down the system a bit more is to go into Main Menu > Preferences > Raspberry Pi Configuration and select the Interfaces tab. Make sure everything you don't need is disabled.

# CODING AND MAKING TIPS

Get more from your makes with Raspberry Pi.

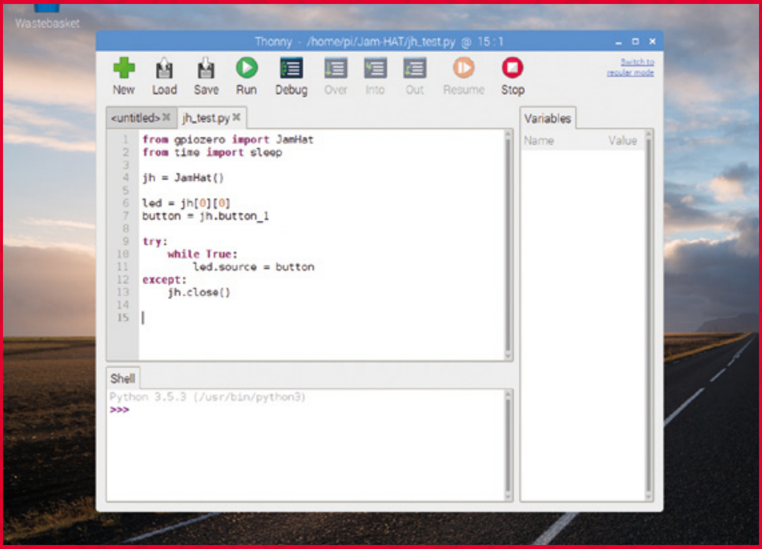
**By Lucy Hattersley**

## 21 Switch to Thonny

If you're still programming in Python IDLE or using Text Editor, then switch to Thonny. It is just as easy as the other programs, but has a simple debugger that makes it much easier to fix flaws in your code. Press **CTRL+F5** instead of **F5** to run a program step-by-step. Unlike more advanced IDEs, you don't need to set breakpoints. Press **F6** for a big step and **F7** for a small step. It's ideal for makers because it enables you to fix problems, but isn't so involved that you get completely absorbed in code and so you can carry on with your build.

## 22 GPIO Zero

Make sure you learn to code using the GPIO Zero Python library (now in version 1.5). This boilerplate library is designed to take the hassle out of using electronics components and kits with Raspberry Pi. With GPIO Zero you can spend more time putting your project together, and less time figuring out how to get it to work. Bookmark the documentation: **gpiozero.readthedocs.io**.



```
from gpiozero import JamHat
from time import sleep

jh = JamHat()

led = jh[0][0]
button = jh.button_1

try:
    while True:
        led.source = button
except:
    jh.close()

```

```
Python 3.5.3 (/usr/bin/python3)
>>>
```

▼ The Thonny IDE (integrated development environment) makes it easier to debug your programs without getting too distracted

## 23 Make circuit diagrams

Fancy making a circuit board diagram like the ones you see in *The MagPi* magazine? The software you're looking for is Fritzing. It's currently in beta, and we mostly use the Windows version to create the images you see in *The MagPi*. However, a version is available for Raspberry Pi.

```
sudo apt-get update
sudo apt-get install fritzing
```

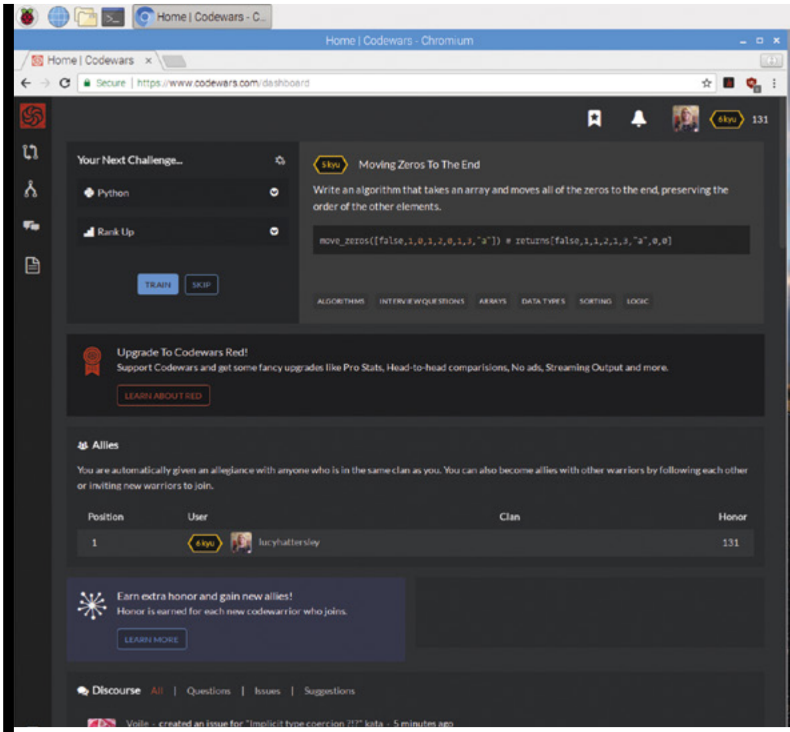
Restart your Raspberry Pi and you'll find the program under Menu > Programming. We are currently having trouble opening the parts bin on Raspberry Pi; if any readers have a fix, then please let us know.

## 24 Raspberry Pi and Arduino

Raspberry Pi works fantastically well with an Arduino attached. Despite looking visually similar, the two boards are very different. Raspberry Pi is a complete single-board computer, while the Arduino range are microcontrollers. They work fantastically well together because the Arduino is handy for controlling parts, with Raspberry Pi better for the higher-level interface and master control. You can install the software to control an Arduino from Raspberry Pi:

```
sudo apt-get upgrade
sudo apt-get install arduino
```

Arduino programs are called 'sketches', and are based on the C programming language. For more info, see our book on learning C (**magpi.cc/learn-c-book**) and our guide to Arduino and Raspberry Pi (**magpi.cc/67**).



Your Next Challenge...

5kyu Moving Zeros To The End

Write an algorithm that takes an array and moves all of the zeros to the end, preserving the order of the other elements.

```
move_zeros([false,1,0,1,2,0,1,3,"a"]) # returns[false,1,1,2,1,3,"a",0,0]
```

Upgrade To Codewars Red!
Support Codewars and get some fancy upgrades like Pro Stats, Head-to-head comparisions, No ads, Streaming Output and more.

Allies

You are automatically given an allegiance with anyone who is in the same clan as you. You can also become allies with other warriors by following each other or inviting new warriors to join.

| Position | User | Clan | Honor |
|---|---|---|---|
| 1 | 6kyu lucyhattersley | | 131 |

Earn extra honor and gain new allies!
Honor is earned for each new codewarrior who joins.

Voile - created an issue for "Implicit type coercion ?!?" kata - 5 minutes ago

## 25 Codewars

One of the best ways to boost your programming skills is to complete challenges. If you are looking for some programming challenges, then sign up for Codewars on your Raspberry Pi. The website (**codewars.com**) has a video game style, and you get points and advance through levels by completing challenges (kata). We've even seen the latter being used in job interviews and to gain entry to coding boot camps.

## 26 Learn to solder

There's a technique to soldering wires together called the 'Western Union' or 'Lineman' splice, and you should learn it if you're going to solder (**magpi.cc/mckKVW**). It was developed during the introduction of the telegraph (hence the name), and the wrapping pattern is designed to tighten as the two ends pull against each other.

## 27 Make a maker kit

We can't recommend highly enough putting together a maker kit. This should contain a wide range of electronics parts, bits, and tools. Find a good components case and put in it a selection of resistors, capacitors, LEDs, buttons, potentiometers, a piezo buzzer, and a breadboard with lots of jumper cables. You can pick up kits and parts from most Raspberry Pi resellers, or visit the Raspberry Pi store in Cambridge. We use a half-size breadboard in our diagrams, so if you get that type you'll find it much easier to follow our projects: **magpi.cc/dXHByV**.

## 28 Second-hand

Get to know your local second-hand ('thrift') store. Used objects can be a great source of inspiration, and upcycling old tech with new Raspberry Pi parts is a popular pastime. High-profile maker Martin Mander has rescued old radios, televisions, and toys. Follow him on Twitter (**@MartinWMander**) for inspirational ideas.

## 29 Share early and often

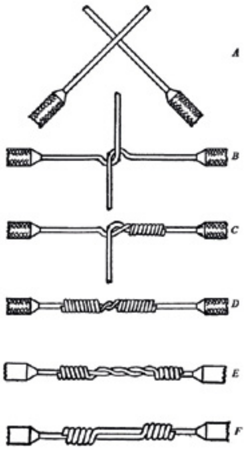
The best way to get inspiration and help for your project is to share it with the Raspberry Pi community. In March 2019, *The MagPi* ran #MonthOfMaking on Twitter, to encourage readers to share projects with one another. But don't wait till next year – share your project on our forum (**magpi.cc/forum**), Facebook (**magpi.cc/facebook**), or Twitter (**@TheMagPi**). The more other readers see your project, the more likely you'll be to finish it.

## 30 Tips for makers

Element14 has two collections of Tips for Makers packed with interesting ideas. Sign up with the Element14 website to access them: **magpi.cc/XQaCyC**.

▲ Codewars is a good web resource for practising and flexing your programming muscles

▼ Learning the 'Western Union' splice technique will make your soldering joints stronger



# NETWORKING TIPS

Need to know more about Raspberry Pi networking? Here are some of the basics and some interesting tips if you're already networked up

You can connect your Raspberry Pi to a local area network and the internet using an RJ45 cable or by configuring WiFi

### You'll Need

- Network cable
- Internet router
- Internet connection

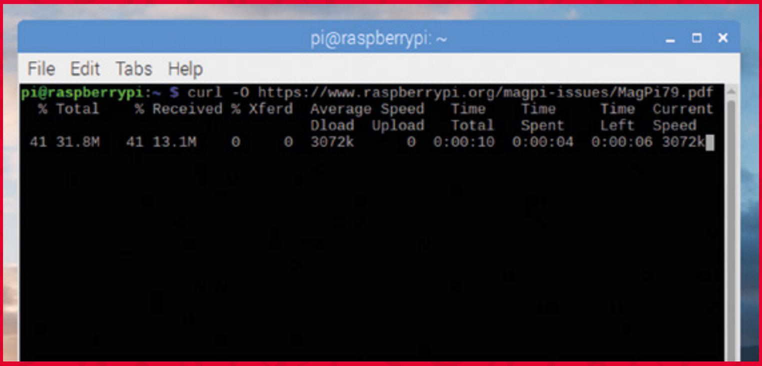
## 31 Wired or wireless

Raspberry Pi has two standard ways of connecting to a network: with an Ethernet cable or on-board WiFi (on newer models). The Ethernet cable is usually either a Cat5e or Cat6 (for fast networks) type, with an RJ45 connector on each end. One end plugs into a router or hub and the other end into Raspberry Pi. Wired connections are generally faster than WiFi and will connect as soon as the cable is plugged in. Home wireless network connections can usually be connected to by just selecting the network and typing in a password – see **magpi.cc/HiDkyk**.

## 32 Know your IP

An IP (Internet Protocol) address is a number allocated to your Raspberry Pi so that other computers know where to find it on the network. There are several ways to find your Raspberry Pi's current IP address. In a Terminal window, there are a couple of commands you can use. The old command (which still works) is `ifconfig`. There is a new shorter command now, `ip`, which will provide similar information. Your local IP address will be the number that looks something like 192.168.0.34. You can also find your IP address by hovering over the network icon (top right) of your desktop.

You can use the curl tool to download files, such as PDFs of *The MagPi* issues



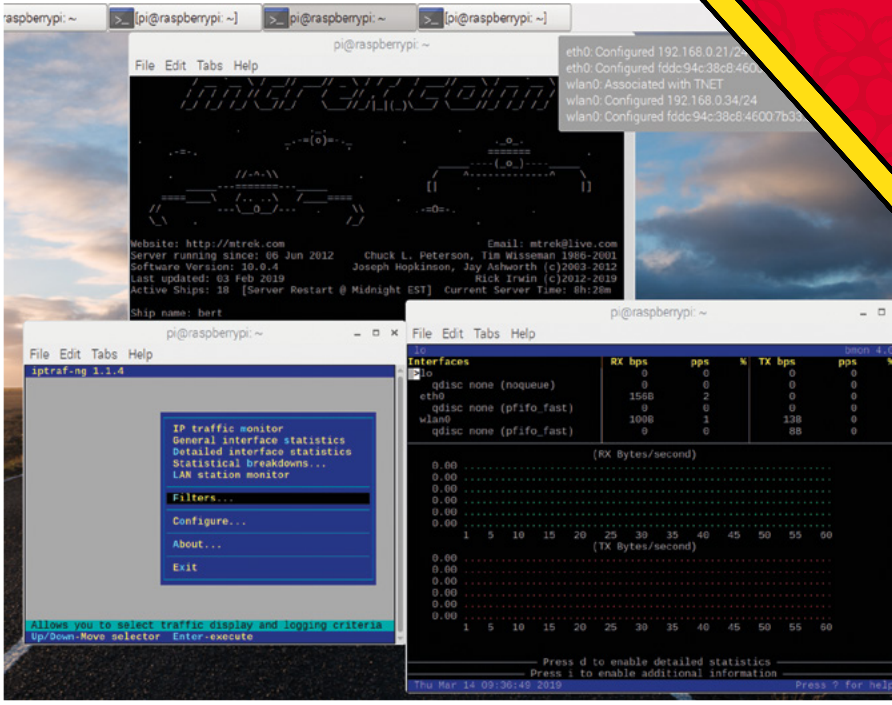
```
pi@raspberrypi:~ $ curl -O https://www.raspberrypi.org/magpi-issues/MagPi79.pdf
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 41 31.8M   41 13.1M    0     0  3072k      0  0:00:10  0:00:04  0:00:06 3072k
```

## 33 IPv4 and IPv6

The terms IPv4 and IPv6 are descriptions of the internet protocol and address. IPv4 uses 32-bit numbers for the address and looks something like: 192.168.1.23. As the internet has now grown so large, the limited number of available IPv4 addresses has become a problem. A new version of IP (IPv6), using 128 bits for the IP address, has been defined. In July 2017, the final definition of the protocol was published, although IPv6 has been in use for over 20 years. An IPv6 address looks something like: fe80:94c:38c8:4600:7b33:64bb:baf7:e866. On a home network you will normally be using IPv4 format addresses, but look out for that to change in the future.

## 34 Useful commands

You can do quite a lot with network commands in the Terminal. Some commands need to be installed first. The ping command is there by default. If you type `ping` and then either an IP address or a web domain, it will tell you how long it takes to send a message to that computer/server and then get a reply. Cat is also pre-installed. Type `cat /sys/class/net/wlan0/operstate` to see the status of your WiFi connection. Install whois with `sudo apt-get install whois` and then type `whois` with a domain name and you will find the registration details of that domain.



```
eth0: Configured 192.168.0.21/24
eth0: Configured fddc:94c:38c8:460
wlan0: Associated with TNET
wlan0: Configured 192.168.0.34/24
wlan0: Configured fddc:94c:38c8:4600:7b33
```

```
Website: http://mtrek.com                                   Email: mtrek@live.com
Server running since: 06 Jun 2012     Chuck L. Peterson, Tim Wisseman 1986-2001
Software Version: 10.0.4            Joseph Hopkinson, Jay Ashworth (c)2003-2012
Last updated: 03 Feb 2019                             Rick Irwin (c)2012-2019
Active Ships: 18  [Server Restart @ Midnight EST]  Current Server Time: 8h:28m

Ship name: bert
```

```
iptraf-ng 1.1.4

    IP traffic monitor
    General interface statistics
    Detailed interface statistics
    Statistical breakdowns...
    LAN station monitor

    Filters...

    Configure...

    About...

    Exit

Allows you to select traffic display and logging criteria
Up/Down-Move selector  Enter-execute
```

```
 lo                                                                bmon 4.0
Interfaces                   RX bps       pps     %  TX bps       pps
 >lo                              0         0             0         0
    qdisc none (noqueue)          0         0             0         0
  eth0                         156B         2             0         0
    qdisc none (pfifo_fast)       0         0             0         0
  wlan0                        100B         1           13B         0
    qdisc none (pfifo_fast)       0         0            8B         0

                        (RX Bytes/second)
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
        1   5    10   15   20   25   30   35   40   45   50   55   60
                        (TX Bytes/second)
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
   0.00 ...........................................................
        1   5    10   15   20   25   30   35   40   45   50   55   60

            Press d to enable detailed statistics
           Press i to enable additional information
 Thu Mar 14 09:36:49 2019                                  Press ? for help
```

▲ You can use lots of different tools to find out about your Raspberry Pi network connections; you can even play online games in the Terminal window!

## 35 Networking files

Raspberry Pi has several files where it holds details about networking. It's quite useful to know where some of these are, as you may need to change some of the information for custom configurations. The file **/etc/hosts** has a list of names that map to IP addresses. You could have an entry that says '192.168.0.3 myserver' and that would mean that if you typed 'myserver' into a web browser address bar, the network would try to connect to 192.168.0.3. Other useful files are **/etc/networks**, which has network names mapped to IP addresses; **/etc/protocols**, which has a list of all the network protocols that can be used; and **/etc/services**, which has a list of all the network services available.

## 36 Network traffic

If you want to know what's happening between your Raspberry Pi and the rest of the network, there are a few commands you can install. A simple stats command is available with `sudo apt-get install vnstat`. When you run this command, it will give you some basic data transmission totals. If you want some extra tools, you could install **iptraf-ng**, which has a menu of different tools and filters to analyse the network traffic. To see some real-time data transfer stats, install **bmon**, which shows data quantities per second.

## 37 Telnet

Although Telnet is a very old networking system (first designed in 1969), it is still available and can be installed on Raspberry Pi using `sudo apt-get install telnet`. Telnet is a bit like a very early version of websites and when this is installed, your Raspberry Pi can connect to any Telnet server. For example, enter `telnet towel.blinkenlights.nl 23` to see an ASCII version of *Star Wars*, or try `telnet mtrek.com 1701` for a *Star Trek* game. There are lots of Telnet sites still out there, some of which can be found at: **telnet.org/htm/places.htm**.

## 38 PiServer

PiServer is a useful tool for networking many Raspberry Pi boards together without needing SD cards in them. For info on how to set it up, go to **magpi.cc/RQDQXx**.

## 39 Downloading files with curl

You can download files directly in the Terminal with the curl tool. Use `curl -O <URL>`. See **magpi.cc/JrKoTA** for more info.

## 40 Headless wireless

If you use the Raspberry Pi in a 'headless' mode (without the Raspbian interface), it can be tricky to connect to a wireless network. You will need to define a **wpa_supplicant.conf** file for your particular wireless network and put this file in the **boot** folder. See here for more information: **magpi.cc/RHviuV**.

# COMMAND LINE TRICKS

Looking to up your command-line game? **PJ Evans** introduces the most popular shortcuts and tricks

## 41 Piping

Applications in Linux systems, such as Raspbian, have three ways of communicating with you and other applications: standard in (stdin), standard out (stdout), and standard error (stderr). Output from a command-line application arrives on stdout and you can use the pipe character '|' to send that output directly into another application, like this:

```
ls -l | wc -l
```

Here we get a long-form listing of the current directory and pipe that output into 'wc', a counting application. The `-l` argument means 'count the lines', so we now have a 'new' app that tells us how many items are in a directory. You can chain as many commands as you wish to achieve complex tasks.

## 42 Output redirection

Try this:

```
curl https://cataas.com/cat
```

The command downloads a cat image – but you're on the Terminal, which can't show graphics. The image arrives as a lot of nonsense on stdout. We can use redirection to save it as a file like this:

```
curl https://cataas.com/cat > cat.jpg
```

The '>' operator tells Raspbian to put everything arriving on stdout into the file instead of your screen. You can also use '>>' which appends to an existing file, or '<' to feed the contents of a file into a command.

## 43 Run multiple commands

You can put multiple commands together on a single line so that they execute one after the other:

```
command1; command2; command3
```

To only proceed if the previous command has succeeded:

```
command1 && command2 && command3
```

For example:

```
mkdir ~/myDir && cd ~/myDir && nano
myFile.txt
```

...will create a directory, move into it, and open up the nano text editor with a new file, **myFile.txt**. You can save this line somewhere for future use or create an alias (see Tip 47).

## 44 Run background tasks

Running a command 'in the background' enables you to issue other commands while it works. If you're using the Desktop, the simplest solution is to open another Terminal window. Alternatively, to have a process run in the background, add an ampersand (&) at the end of the command:

```
very-long-command &
```

This works well, but any output to stdout and stderr will appear on your console screen. To avoid this, use 'nohup':

```
nohup very-long-command &
```

Now all output is written to **nohup.log** in your current directory.

## 45 Watching log files

Sometimes when a process is generating output, you might need to follow what's happening. For example, your web server isn't working quite right, so you can troubleshoot the problem by looking at the logs. To show the contents of any file on-screen:

```
cat text.txt
```

# TROUBLESHOOTING TIPS

However, if the log file is huge, you'll struggle to cope. To show the file in 'pages':

```
less text.txt
```

To just see the last few lines:

```
tail text.txt
```

Best of all, to 'follow' the log file as it grows:

```
tail -f text.txt
```

## 46 A miscellany of shortcuts

**Tilde** – The tilde character '~' always refers to your home directory in paths. So `cd ~` always takes you home (as does `cd` on its own).

**TAB autocompletion** – Whenever possible, Raspbian will attempt to complete the command name you are typing. Try typing `whoa` followed by **TAB**. It will expand to `whoami`. This also works for paths.

**History** – Forgotten that command? Your Raspberry Pi is keeping track. Pressing the up arrow key repeatedly goes through previous commands. Or run `history` to get a full list.

**!!** – Two exclamation marks means 'the previous command'. Forgot to use sudo? Don't type the command again, just enter `sudo !!`.

## 47 Using aliases

If there's a long command you're going to be using a lot, create an alias for it. Aliases are kept in a hidden file (non-existent by default) called **.bash_aliases**, in the home directory. To create or view contents:

```
nano ~/.bash_aliases
```

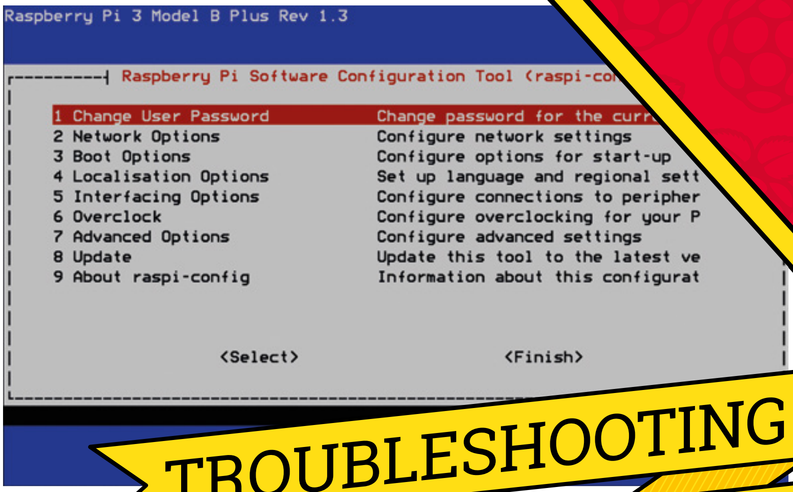
For each alias you want to create, enter a line following this pattern:

```
alias howmany="ls -l | wc -l"
```

Now you can run `howmany` for that command.

## 48 It's got stuck!

Stuck with a crashed program on your screen? To gain access back to the prompt, try **CTRL+C** to stop the process, or **CTRL+Z** to put it in the background. Now enter `killall <commandname>`. You can use `ps ax` to show a list of running processes. Still there? Try `sudo killall <commandname>`. As a last resort, use `sudo kill -9 <processid>` (get the process ID from `ps ax`).



```
Raspberry Pi 3 Model B Plus Rev 1.3

Raspberry Pi Software Configuration Tool (raspi-co

 1 Change User Password          Change password for the curr
 2 Network Options               Configure network settings
 3 Boot Options                  Configure options for start-up
 4 Localisation Options          Set up language and regional sett
 5 Interfacing Options           Configure connections to peripher
 6 Overclock                     Configure overclocking for your P
 7 Advanced Options              Configure advanced settings
 8 Update                        Update this tool to the latest ve
 9 About raspi-config            Information about this configurat

            <Select>                      <Finish>
```

▲ Some command-line applications take on a basic graphical appearance to make them easier to use. The most likely you are to see is raspi-config

## 49 Find out more

Many commands come with a huge array of 'arguments' that control behaviour. These are in the form of 'switches' (simple flags) or 'parameters' that provide additional info such as an input file. Most commands have an accompanying 'man' page (short for manual). To see if a 'manpage' exists for your command, just enter:

```
man command
```

You can even learn about 'man' itself:

```
man man
```

Alternatively, many commands respond to the switch `--help` (or `-h` for short) to provide a summary of options.

## 50 Get in remotely

We've all been there: happily working away in Raspberry Pi Desktop and something goes awry. Now the screen has locked up, your Doom high score is lost, and the only option is to unplug your Raspberry Pi, which risks data corruption. Well, you may be able to get in using SSH, which allows you to log into your Raspberry Pi (so long as you've already enabled SSH on it) from another computer and get to the command line:

```
ssh raspberrypi.local
```

Then, if you get to a prompt:

```
sudo shutdown -h now
```

...and your Raspberry Pi is rescued.

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi* magazine. She enjoys hacking and fixing technology, and building quirky projects. She's done the Samba all weekend long!

**magpi.cc**

# Build a Samba file server

Use Samba/CIFS to share a Raspberry Pi directory with other computers on a network

It's easy to use a Raspberry Pi as a Samba file server where you can store backups and share files from all the other computers on your network.

Samba is the Linux implementation of the SMB/CIFS file-sharing standard used by Windows PCs and Apple computers, and widely supported by media streamers, games consoles, and mobile apps.

With Samba activated, you can quickly copy files from a computer on your network to a Raspberry Pi using wireless LAN (or a direct Ethernet connection).

This tutorial assumes that you'll use a keyboard, mouse, and monitor to set up your file server; alternatively, you can enable SSH (**magpi.cc/ssh**) and connect to it remotely from another computer on your local network.

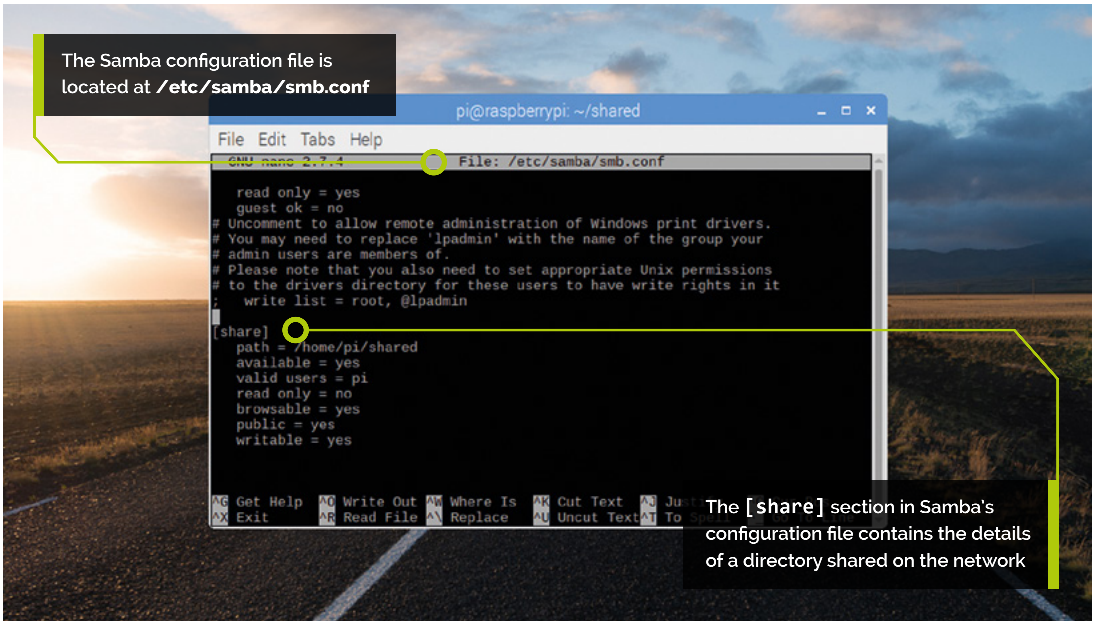
We also assume you're using a 32GB (or smaller) microSD card, which provides a reasonable amount of storage space without requiring any extra steps to make it accessible. However, if you need extra storage, it's possible to mount a large external USB drive and create a Samba entry for it.

Alternatively, if you want to keep things compact, you can install Raspbian on microSD cards of up to 256GB, although we think it's wiser to install Samba on an external drive and use a smaller microSD to boot Raspbian and connect to it.

We suggest checking non-working microSD cards online (**magpi.cc/2q97aGO**) before you buy, to make sure you get one that's fully compatible with the Raspberry Pi.

## You'll Need

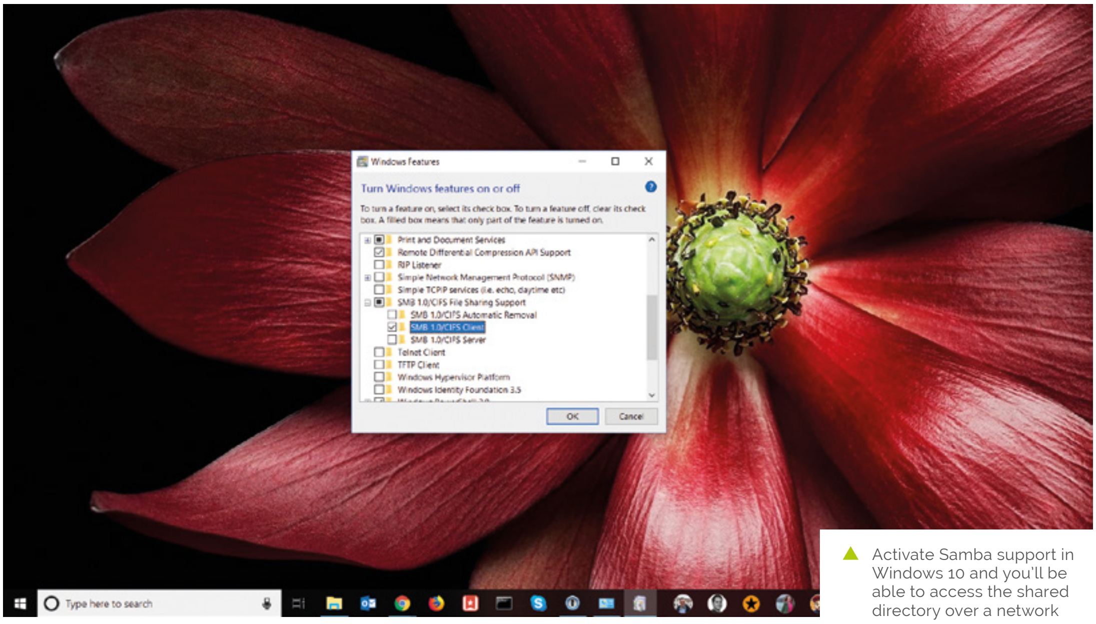
- Raspberry Pi
- microSD card
- Keyboard and mouse for setup
- Network and router (Ethernet recommended)



```
  GNU nano 2.7.4              File: /etc/samba/smb.conf

   read only = yes
   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;    write list = root, @lpadmin

[share]
   path = /home/pi/shared
   available = yes
   valid users = pi
   read only = no
   browsable = yes
   public = yes
   writable = yes

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Ju
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To
```

The Samba configuration file is located at **/etc/samba/smb.conf**

The **[share]** section in Samba's configuration file contains the details of a directory shared on the network


Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ Print and Document Services
- ☑ Remote Differential Compression API Support
- □ RIP Listener
- □ Simple Network Management Protocol (SNMP)
- □ Simple TCPIP services (i.e. echo, daytime etc)
- ■ SMB 1.0/CIFS File Sharing Support
  - □ SMB 1.0/CIFS Automatic Removal
  - ☑ SMB 1.0/CIFS Client
  - □ SMB 1.0/CIFS Server
- □ Telnet Client
- □ TFTP Client
- □ Windows Hypervisor Platform
- □ Windows Identity Foundation 3.5

Activate Samba support in Windows 10 and you'll be able to access the shared directory over a network

Once set up, you can mount your home file server on all the other computers on your network, and use it as a convenient place to store everything from music files you want to share with your housemates, to backups of important documents and save-game files you'd like to share between computers.

We recommend using a wired Ethernet connection for stability and fast transfer speeds. The project will still work if you connect your Raspberry Pi via WiFi, although performance will be affected, particularly when it comes to copying over large files.

## 01 Install Raspbian

Download the latest version of NOOBS (or the latest Raspbian Stretch with Desktop image file) and install it to a microSD card. See **magpi.cc/quickstart** if you're unsure on how to set up a Raspberry Pi for the first time. Insert the microSD card into your Raspberry Pi and connect the keyboard, mouse, and HDMI cable. We suggest you connect an Ethernet cable between your Raspberry Pi and modem/router (otherwise use the wireless LAN icon to connect to a wireless network when booted up). Attach the power cable to boot up your Raspberry Pi.

## 02 Install Samba

Samba is available in Raspbian's standard software repositories. We're going to update our repository index, make sure our operating system is fully updated, and install Samba using APT. Open a Terminal and type:

```
sudo apt update
sudo apt upgrade
sudo apt install samba samba-common-bin
```

Answer **Y** when prompted.

## 03 Create shared directory

Now we're going to create a dedicated shared directory in our home directory.

```
sudo mkdir /home/pi/shared
```

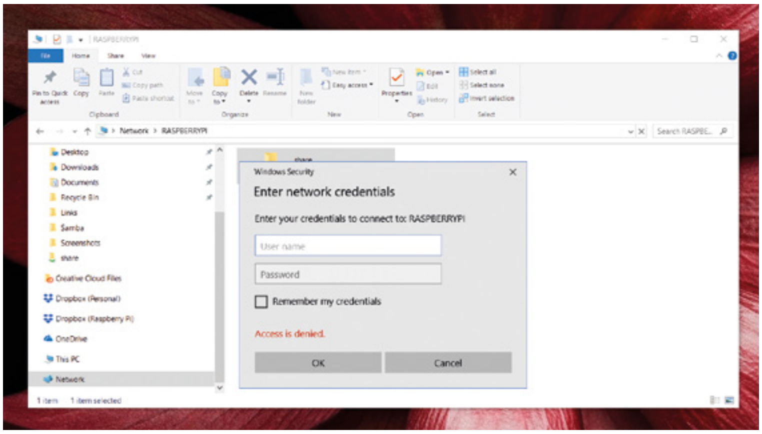
You can put it anywhere, but ours is in the home directory (it makes it easier to access without adjusting permissions). You'll be able to access this

## Top Tip

### Status report

You can see the status of the Samba drive using `sudo service smdb status`. And you can use `smbd restart` to turn the service off and on again.

Enter the credentials (username and password) created during the setup process to access the Samba share


Windows Security

Enter network credentials

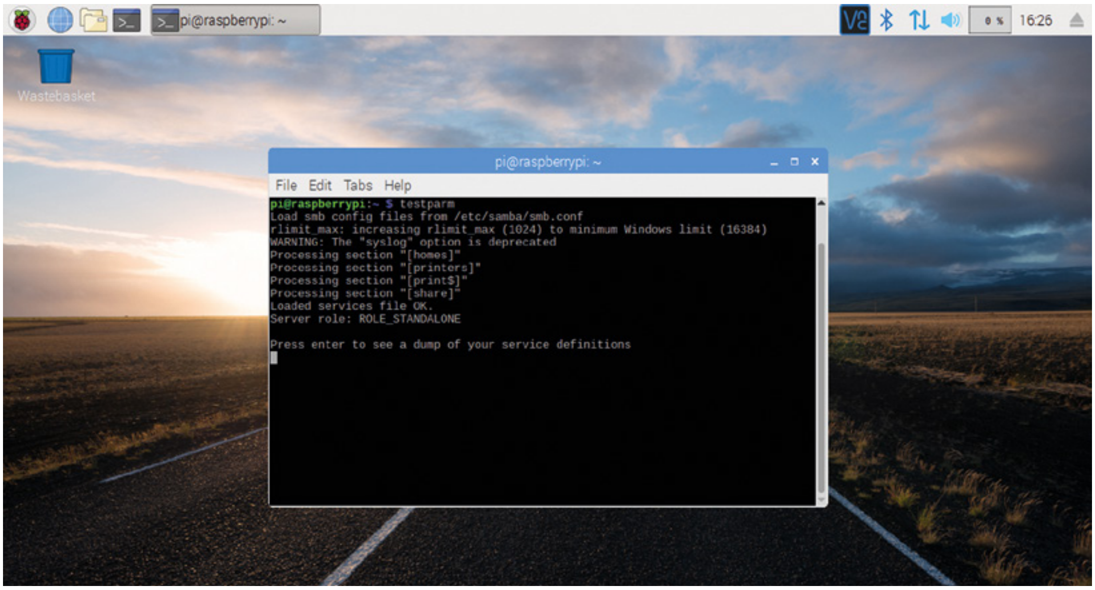
Enter your credentials to connect to: RASPBERRYPI

User name

Password

□ Remember my credentials

Access is denied.



```
pi@raspberrypi:~ $ testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
WARNING: The "syslog" option is deprecated
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[share]"
Loaded services file OK.
Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions
```

Use testparm to test the edits you make to **smb.conf**

directory from other computers on your network. Files you drop in here are accessible on other computers, and files you drop into the directory from other computers will appear on Raspberry Pi.

## 04 Configure Samba

Now we need to edit Samba's configuration file to make the shared directory visible to other computers on the network. First, let's make a backup just in case you want to look at the original.

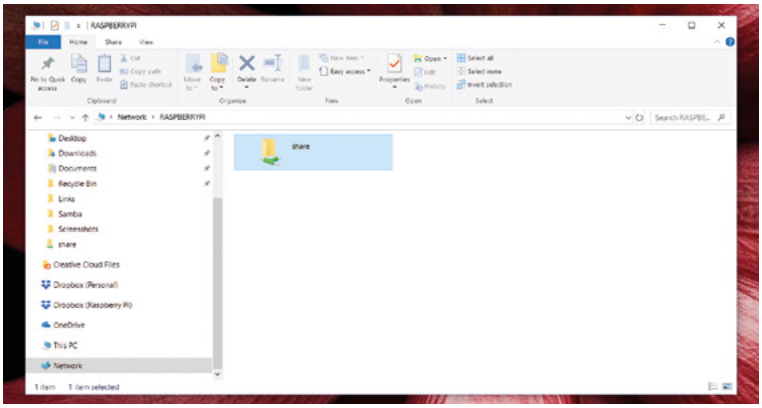
```
sudo cp /etc/samba/smb.conf /etc/samba/
smb.conf_backup
```

Now let's edit the Samba configuration file. Enter:

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

The **smb.conf** file is packed with detailed information on the Samba configuration (it's a good idea to read through it all). Scroll to the end of the file and add the configuration code from the **smb-pi.conf** listing. Press **CTRL+O** to save the file, and **CTRL+X** to exit nano.

File Explorer displaying the Samba share in its Network



## 05 Looking at the configuration file

The Samba configuration file uses the same format as Windows .ini files, with sections surrounded with square brackets. The commands are case-insensitive and ignore white space. Let's take a closer look at some of the lines in the file...

The location of the shared folder:

```
path = /home/pi/shared
```

Set this to 'no' to turn off the service:

```
available = yes
```

This is the list of users allowed to log in to the service:

```
valid users = pi
```

Whether this share is seen in the list of available shares in a net view and in the browse list:

```
browsable = yes
```

Ensures that users can write to files (the default is no):

```
writable = yes
```

Take a look at the Samba.org documentation for more information on all the configuration options available in **smb.conf** (**magpi.cc/alEJZY**). Test out the Samba configuration file with testparm. Enter the following:

```
testparm
```

...and you should see a series of 'Processing section' tests. At the end it should say 'Loaded services file OK'. If not, head back to the start of this step and check you have entered the additional setup text correctly.

## 06 Create a password

Before we start the server, you'll want to set a Samba password. Enter:

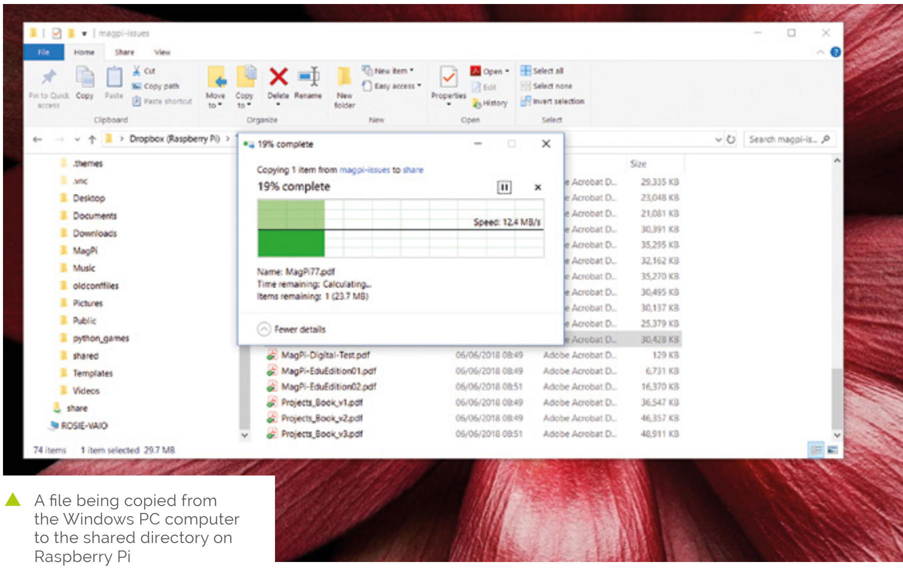
```
sudo smbpasswd -a pi
```

Enter a simple and memorable password for testing (you can create a more secure one later by rerunning the command at any time).

Now restart your Samba server:

```
sudo service smbd restart
```

The Samba service will restart and the shared drive will now be available on the network.



19% complete

Copying 1 item from magpi-issues to share

19% complete

Speed: 12.4 MB/s

Name: MagPi77.pdf
Time remaining: Calculating...
Items remaining: 1 (23.7 MB)

Fewer details

| | | | |
|---|---|---|---|
| MagPi-Digital-Test.pdf | 06/06/2018 08:49 | Adobe Acrobat D... | 129 KB |
| MagPi-EduEdition01.pdf | 06/06/2018 08:49 | Adobe Acrobat D... | 6,731 KB |
| MagPi-EduEdition02.pdf | 06/06/2018 08:51 | Adobe Acrobat D... | 16,370 KB |
| Projects_Book_v1.pdf | 06/06/2018 08:49 | Adobe Acrobat D... | 36,547 KB |
| Projects_Book_v2.pdf | 06/06/2018 08:49 | Adobe Acrobat D... | 46,357 KB |
| Projects_Book_v3.pdf | 06/06/2018 08:51 | Adobe Acrobat D... | 48,911 KB |

74 items 1 item selected 29.7 MB

A file being copied from the Windows PC computer to the shared directory on Raspberry Pi

### smb-pi.conf

Language: **ASCII**

**DOWNLOAD THE FULL CODE:**

**magpi.cc/ryrLvF**

```
[share]
    path = /home/pi/shared
    available = yes
    valid users = pi
    read only = no
    browsable = yes
    public = yes
```

## 07 Turn on SMB in Windows

Samba used to be installed by default in Windows, but it has recently become an optional installation.

In Windows 10, click on the Search bar and type 'Control Panel'. Click on Control Panel in the search results. Now click 'Programs' and 'Turn Windows features on or off'. Scroll down to find 'SMB 1.0/CIFS File Sharing Support' and click the '+' expand icon to reveal its options. Place a check in the box marked 'SMB 1.0/CIFS Client'. Click OK. This will enable Samba client support on your Windows 10 PC so it can access your Raspberry Pi.

## 08 Find your Pi on the network

You'll now be able to find your Raspberry Pi file server (named RASPBERRYPI by default) from any device on your local network.

Click on the File Explorer icon in Windows and choose the Network option in the sidebar. You will see RASPBERRYPI in the list of devices. Click it and the 'Enter network credentials' window will appear. Enter your username **pi** and the password you created in Step 6. Click 'Remember my credentials' if you want to save the password, and then click OK.

The File Explorer will now display the shared directory. You can drag files and folders from Windows to the shared folder and they will be copied to your Raspberry Pi.

## 09 Find the Pi on a Mac

Samba works across many different devices. And it's perfectly possible to access the shared folder from a Mac computer, or even an iOS or Android device (with the right app). Open a new Finder window on a Mac and choose Network in the sidebar. Click on 'raspberrypi' and 'shared' to access the directory. As with Windows, you can drag files directly to and from Raspberry Pi.

### Top Tip

**Reboot everything**

If you've set it all up and have checked everything, try restarting your Raspberry Pi and Windows computer. Sometimes this helps Windows find the drive on the network (once the shared drive appears, we find it stable).

# Use SSH (Secure Shell) with Raspberry Pi

Remotely control your Raspberry Pi from a PC, Linux, or Mac computer and transfer files using SSH. By **Lucy Hattersley**

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi*. She likes retro gaming, building robots, coding, and making magazines and websites.

**magpi.cc**

**SSH (also known as 'Secure Shell') is an encrypted networking technology that enables you to manage computers from the command line over a network.**

SSH is handy if you want to quickly connect to a Raspberry Pi from a terminal window on another computer. It's also ideal for lightweight distro installations that don't have graphical interfaces, and projects that don't have a screen (such as robots). It's especially useful when creating Internet of Things (IoT) projects, as these may be embedded inside other hardware.

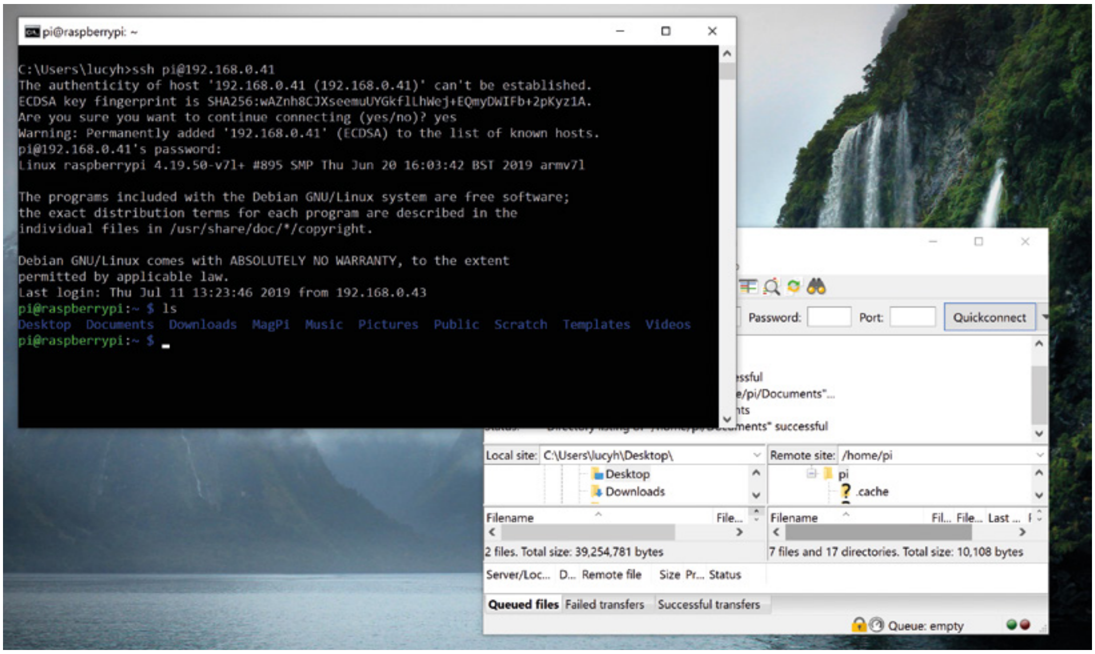
We covered VNC (Virtual Network Computing) on page 19 – and in more detail in issue 71 of *The MagPi*, **magpi.cc/71** – and Secure Shell offers a similar service. But while VNC shares the entire desktop, SSH works from the command line.

SSH also enables you to share files to, and from, a Raspberry Pi using SCP (secure copy) on the command line, or SFTP (SSH File Transfer Protocol).

You don't need to install any software to start using Secure Shell. Linux, macOS, and Windows 10 have the SSH command-line application installed by default.

You'll need to use the password for your Raspberry Pi to log in. For security reasons, we strongly recommend changing the default password (**magpi.cc/tZdxha**).

SSH uses an encrypted network, so it doesn't send your password as plain text. More advanced users can control the encryption keys, using ssh-keygen. For now, we'll look at setting up and using Secure Shell.

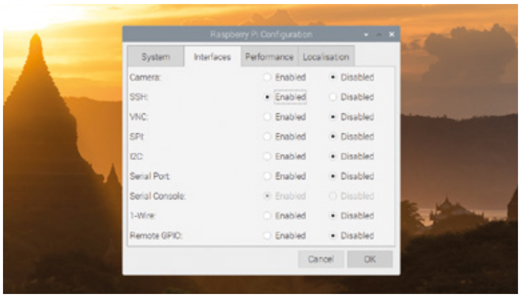


```
C:\Users\lucyh>ssh pi@192.168.0.41
The authenticity of host '192.168.0.41 (192.168.0.41)' can't be established.
ECDSA key fingerprint is SHA256:wAZnh8CJXseemuUYGkflLhWej+EQmyDWIFb+2pKyz1A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.41' (ECDSA) to the list of known hosts.
pi@192.168.0.41's password:
Linux raspberrypi 4.19.50-v7l+ #895 SMP Thu Jun 20 16:03:42 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 11 13:23:46 2019 from 192.168.0.43
pi@raspberrypi:~ $ ls
Desktop  Documents  Downloads  MagPi  Music  Pictures  Public  Scratch  Templates  Videos
pi@raspberrypi:~ $ _
```

pi@raspberrypi: ~
Password:
Port:
Quickconnect
Local site: C:\Users\lucyh\Desktop\
Desktop
Downloads
Remote site: /home/pi
pi
.cache
Filename
2 files. Total size: 39,254,781 bytes
7 files and 17 directories. Total size: 10,108 bytes
Server/Loc... D... Remote file Size Pr... Status
Queued files Failed transfers Successful transfers
Queue: empty

## You'll Need

- Raspberry Pi
- Network router
- Raspbian OS
- Second computer (Windows, macOS, or Linux)



| | | |
|---|---|---|
| Camera: | ○ Enabled | ● Disabled |
| SSH: | ● Enabled | ○ Disabled |
| VNC: | ○ Enabled | ● Disabled |
| SPI: | ○ Enabled | ● Disabled |
| I2C: | ○ Enabled | ● Disabled |
| Serial Port: | ○ Enabled | ● Disabled |
| Serial Console: | ● Enabled | ○ Disabled |
| 1-Wire: | ○ Enabled | ● Disabled |
| Remote GPIO: | ○ Enabled | ● Disabled |

▲ SSH is turned off by default in Raspbian and has to be enabled. The easiest way is to use Raspberry Pi Configuration Tool

## 01 Activate SSH in Raspbian

For security reasons, Secure Shell is not turned on by default in Raspbian. On your Raspberry Pi, choose Menu > Preferences > Raspberry Pi Configuration. Click on Interfaces and set SSH to Enabled. Click OK. You don't need to restart your Raspberry Pi, and SSH will be enabled whenever you use that installation of Raspbian from that point on (be sure to have updated your password from the default, which is 'raspberry').

## 02 Activate SSH Client in Windows

Linux and macOS both support SSH out-of-the-box; skip ahead to Step 3 if you are using one of those operating systems.

Windows 10 supports SSH, but you need to activate it. Click on Search and look for 'Manage Optional Features'. Click it in Search to open the Settings window.

Click 'Add a feature' and wait for the list of Optional Features to load. Scroll down the list to Open SSH Client (Beta). Click Install.

## 03 Get your IP address

Connect your Raspberry Pi to a local network. Use wireless LAN, or connect Raspberry Pi directly to a router with an Ethernet cable. Open a Terminal window and enter the following command:

```
hostname
```

You should see:

```
raspberrypi
```

In some circumstances you can use this host name, which is handy if you don't have the IP address, but it's more reliable to use the IP (internet protocol) address. Enter this to get your IP address:

```
hostname -I
```

This will return four numbers separated by dots. For instance, ours is:

```
192.168.0.41
```

Write this number down. You'll need it shortly.

## 04 Connect via SSH

Open Command Prompt on a Windows PC, or a Terminal window in Linux or macOS. Enter this command:

```
ssh pi@<IP>
```

Replace `<IP>` with the IP address of your Raspberry Pi. In our case, we enter `ssh pi@192.168.0.41`.

The first time you do this, you'll get a message saying that the authenticity can't be established, followed by a long cryptographic hash of letters and numbers. It will say, 'Are you sure you want to continue connecting?'

Enter **yes** and press **RETURN**. You'll be asked to enter the password for your Raspberry Pi.

## 05 On Raspberry Pi

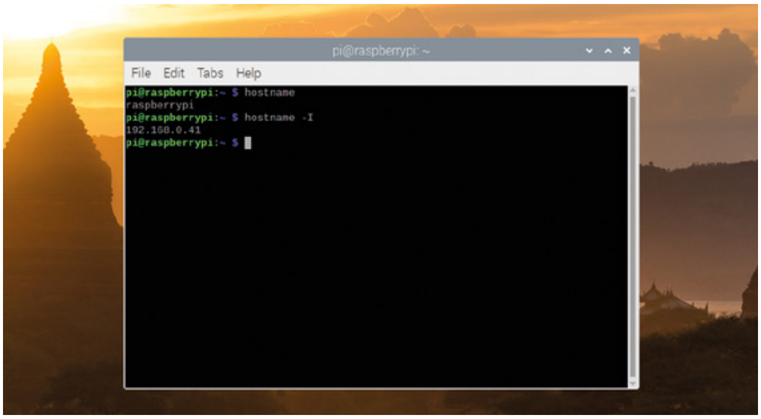
You will now see your usual command line replaced with **pi@raspberrypi: ~$**. You are now logged in and working on the command line from your Raspberry Pi. Enter `ls` and you'll see **Desktop**, **Downloads**, **Documents**, and the other unique Raspberry Pi folders and files. You can create, edit, move, and work with files as if you were using Terminal on your Raspberry Pi.

## Top Tip

### SSH from boot

You can activate SSH when setting up Raspbian by placing a file named **ssh**, without any extension, onto the boot partition of the microSD card from another computer. When Raspberry Pi boots, it looks for the **ssh** file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.

▼ Use the `hostname` command in Terminal to confirm the host name 'raspberrypi', and add the `-I` flag to get your assigned IP (internet protocol) address



```
pi@raspberrypi:~ $ hostname
raspberrypi
pi@raspberrypi:~ $ hostname -I
192.168.0.41
pi@raspberrypi:~ $ 
```

## Top Tip

### Turn SSH on and off

You can activate and start SSH from the command line on your Raspberry Pi:

```
sudo systemctl
enable ssh
sudo systemctl
start ssh
```

And turn it off using:

```
sudo systemctl
stop ssh
sudo systemctl
disable ssh
```

## 06 Visual interfaces

There are limitations to SSH. A plain SSH connection can't open programs that use a graphical interface, so you should learn to use command-line alternatives (such as nano or vim instead of Leafpad for text editing). See How to Edit Text files on Raspberry Pi (**magpi.cc/BNFKeP**).

On Linux computers, you can forward your X session over SSH, to allow the use of graphical applications, by using the `-Y` flag:

```
ssh -Y pi@<IP>
```

Now you are on the command line as before, but you have the ability to open up graphical windows. For example, typing:

```
scratch &
```

...will open up Scratch in a graphical window.

Note that X11 is no longer present on Macs with macOS (**magpi.cc/AqEvnA)**, so you have to download and install it (**xquartz.org**) separately to access graphical windows over SSH.

## 07 Copy files

It's not easy to share files using Secure Shell from the command line, but it's possible and a handy skill to learn. If you're still accessing Raspberry Pi, enter `exit` at the command line to return to your local shell:

```
exit
```

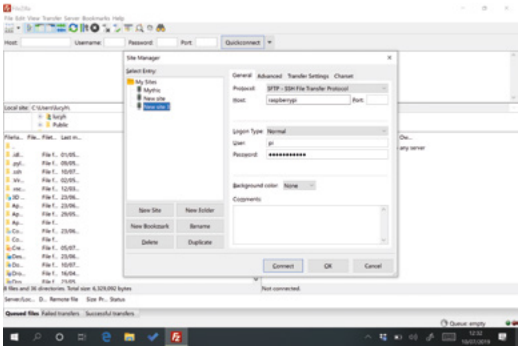
With SSH turned on, you can copy files from your source computer to Raspberry Pi using SCP. We've downloaded *The MagPi* issue 85 (**magpi.cc/85**) PDF file to our **Downloads** folder. We're going to copy it to the **MagPi** directory on our Raspberry Pi.

Use the `scp` command followed by the path to the source file, and then the destination location. The destination is `pi@<IP>` followed by a colon ':' and the path to the destination location.

```
scp <source> pi@<IP>:<destination>
```

Replace the `<source>` and `<destination>` parts with the paths to the file and desired destination folder, and add your Raspberry Pi's IP address.

For instance, copying the PDF file on our machine looks like this:



Site Manager

Protocol: SFTP - SSH File Transfer Protocol

Host: raspberrypi

Logon Type: Normal

User: pi

Password: ••••••••••••

Background color: None

Not connected.

▲ Use SFTP (Secure File Transfer Protocol) in a program such as FileZilla to transfer files to and from your Raspberry Pi via a graphical interface

```
scp Downloads\MagPi85.pdf pi@192.168.0.41:/
home/pi/MagPi
```

This copies *The MagPi* issue 85 PDF from the **Downloads** folder on our Windows PC to the **MagPi** directory on Raspberry Pi.

Notice that the directory path uses backslash ('\') for Windows source and slash ('/') for the Linux destination (if you're using a Mac or Linux machine, it'll be a slash in both source and destination.)

## 08 Copying files

If you copy a lot of files back and forth from a computer to Raspberry Pi, then it's a good idea to set up SFTP using a dedicated FTP program, such as FileZilla (**filezilla-project.org**).

Install FileZilla on your computer (it is available for Linux, Windows, and macOS).

Open FileZilla and choose File > Site Manager.

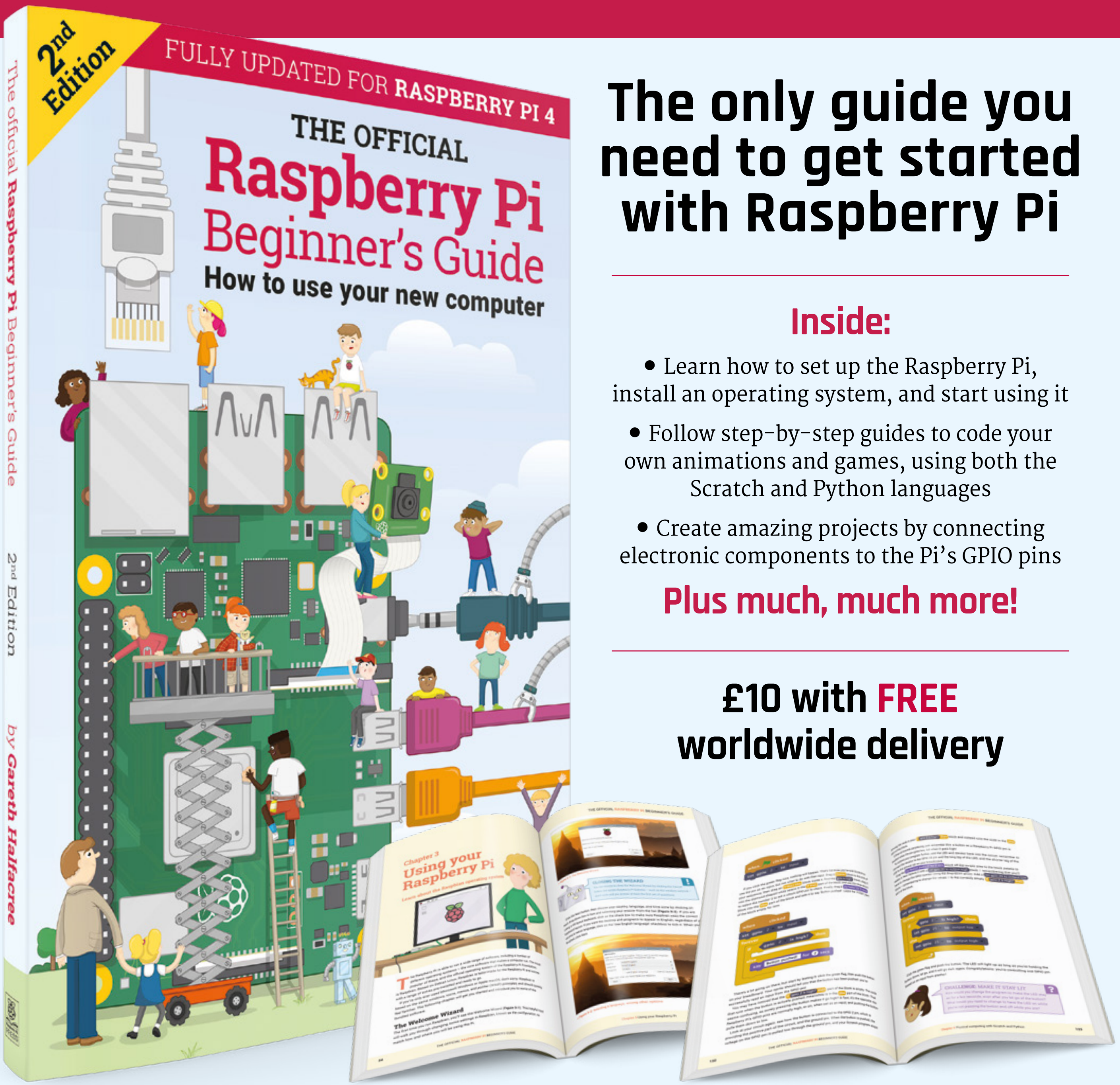
Now click New Site and you can give the site a custom name, such as 'Raspberry Pi'.

Change the Protocol to 'SFTP – SSH File Transfer Protocol'.

Enter 'raspberrypi' in the Host field, 'pi' in the User field, and your Raspberry Pi password in the Password field.

Click Connect and you will connect remotely to the Raspberry Pi. You will see all your Raspberry Pi files in the right-hand 'Remote' side of the display. On the left, in the 'Local Site' section will be all the files on your computer. Double-click directory folders in either side to navigate into them, and use the '..' folders at the top of each list to navigate back up to the parent folder.

Double-click a file in Local Site to send it to the Remote section, and vice versa. Follow the FileZilla tutorial if you need more guidance (**magpi.cc/WovBgn**).


2nd Edition
FULLY UPDATED FOR RASPBERRY PI 4
THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer
The official Raspberry Pi Beginner's Guide
2nd Edition
by Gareth Halfacree
Chapter 3
Using your Raspberry Pi

# SWITCH TO THE COMMAND LINE

By using the command line, you are able to work faster and smarter. Discover how to get started today…

### You'll Need

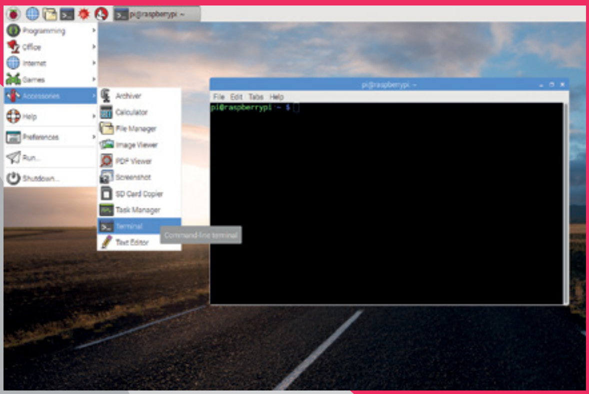
- Raspberry Pi
- Raspbian

**There's nothing wrong with GUIs (graphical user interfaces), and Raspbian comes with a rather fine desktop interface.** But beneath the icons sits a whole other world: the command line. This is where your real computer is. With the command line, you're not locked into doing just what desktop applications enable you to do. You can do just about anything to your computer, and you can do it much faster.

Think of it like driving a car. If you've only ever used a GUI then you're driving an automatic. The command line is like switching to manual. It can be daunting for newcomers, but it really needn't be. With just a few commands, you can master the command line.

## Typing commands

When you boot a Raspberry Pi, you start by default inside the desktop interface. The fastest way to get access to the command line is through the Terminal app. Click on the Terminal icon in the top menu bar (or choose Menu > Accessories > Terminal). A window opens with a black background and some green and blue text. You will see the command prompt:



```
pi@raspberrypi:~ $
```

Most people access the command line through the Terminal app in the desktop interface

```
pi@raspberrypi:~ $
```

You are now at the command line. You enter commands using the text interface. Enter `echo Hello World` and press **RETURN**, you'll see 'Hello World' printed on the line. Below this is another **$** prompt, ready to accept another command.

Most users get to the command line via the Terminal app, but there is another way known as 'virtual console'. Press **CTRL+ALT+F1** and the desktop will vanish. A black screen appears, displaying 'Raspbian (or Debian) GNU/Linux 8 raspberry pi tty' and below it, 'raspberrypi login'. If you are not automatically logged in , enter **pi** and press **RETURN**, then enter your password (**raspberry** by default, but you should always change it).

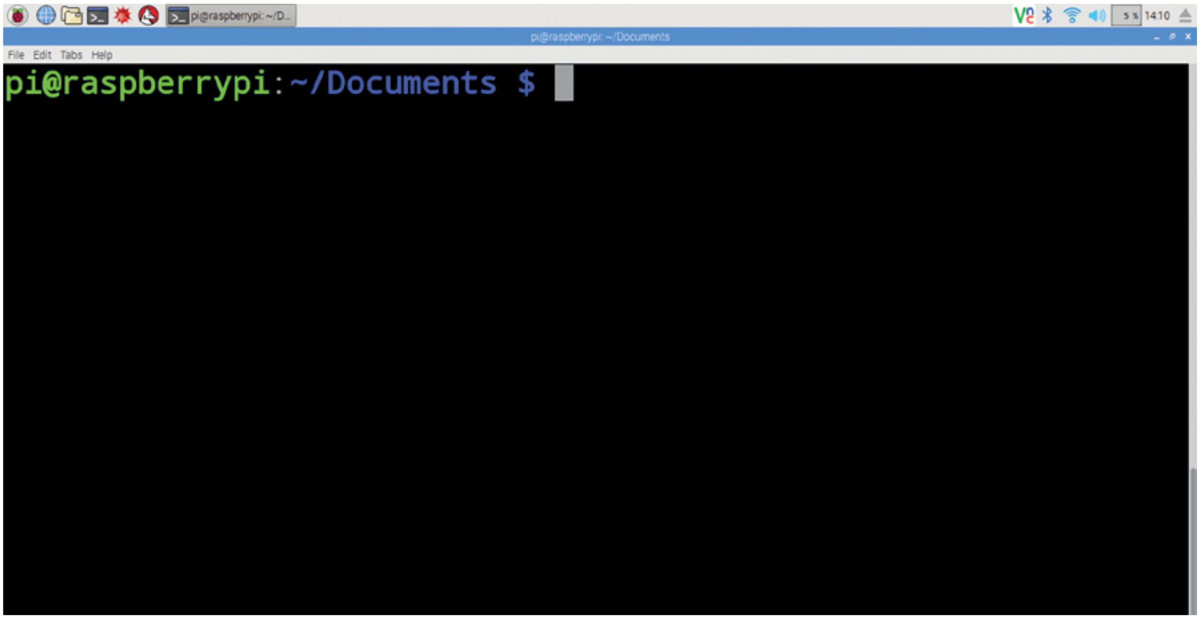
You can now use the command line in full-screen mode. You can get back to the desktop using **CTRL+ALT+F7** and switch back to the virtual console using **CTRL+ALT+F1**. Additional virtual consoles can be accessed using **CTRL+ALT+F2** to **F6**. Each has its own login and operates independently.

If you prefer the command line, you can boot Raspbian directly to the command line instead of the desktop interface. Open Raspberry Pi Configuration (Menu > Preferences > Raspberry Pi Configuration). Change the Boot setting to 'To CLI' and click OK. Now when you reboot, you'll start in the command line (enter `startx` to start up the desktop).

## Locate yourself

The first thing you need to learn is how to find out where you are. You are in your home folder by default. Enter the following command and press **RETURN**:

```
pwd
```



```
pi@raspberrypi:~/Documents $
```

**pi@**
The first part of the command line is your user name followed by an @ symbol. You can see this on the command line by entering `whoami`.

**raspberrypi**
After the @ comes your host name. It is the name of your computer: 'raspberrypi' by default.

**~/Documents**
After the host name is your current working directory. This displays just '~' when you are in your home folder.

**$**
The dollar sign shows you're operating as a normal user.

This command is short for 'print working directory' and it tells you where you are. The command line will return **`/home/pi`**.

The home folder is the same one that appears by default when you open the File Manager app. You view the files and directories inside the working directory using the list (**`ls`**) command:

```
ls
```

Here you'll see the same directories (or folders) that are in the File Manager app: **Desktop**, **Downloads**, **Documents**, and so on.

### The file path

Before going any further with directories, you need to understand the file path and the difference between a 'relative' and 'absolute' path.

Files are placed inside folders (which are called 'directories' in the command line). In the visual GUI, you can see these as folders that you open, revealing files and more folders. Your home folder contains a **Documents** folder, and inside that are three more folders: **Blue J Projects**, **Greenfoot Projects**, and **Scratch Projects**.

```
/home/pi/Documents/Scratch\ Projects
```

In the file path above, the first slash is the root of your hard drive. Here you have a directory called **home** that contains all users. In here is another directory called **pi** (that's you), and inside that is another directory called **Documents**, and inside that is one called **Scratch Projects**.

## UNDERSTAND THE LANGUAGE

There's a lot of confusing jargon thrown around related to the command line. Terms like command line, shell, and terminal are often used interchangeably.
Each has a precise meaning…

- **TERMINAL:** This is the program you use to access the command line from the PIXEL desktop in Raspbian (its full name is LXTerminal).
- **CONSOLE:** This is a physical terminal display with a keyboard. Consoles used to be empty computers that connected to a large mainframe computer.
- **VIRTUAL CONSOLE:** These are virtual versions of a physical console. In Linux, you have multiple virtual consoles accessed using **CTRL+ALT** and the function keys.
- **TTY:** Teletypewriter. In Linux, tty is used to display which virtual console you are using: tty1, tty2, and so on.
- **COMMAND LINE:** This is the text-based environment in general or the specific line you are working on. The command line starts with a dollar sign ($), known as the 'prompt'.
- **SHELL:** This is a command-line interpreter. It surrounds the computer's kernel (hence the name). To get to the kernel, you go through the shell. The shell interprets your text commands and turns them into code the kernel understands.
- **BASH:** This stands for 'Bourne Again Shell' and is the type of shell used in Debian (the version of Linux upon which Raspbian is based).

`ls`

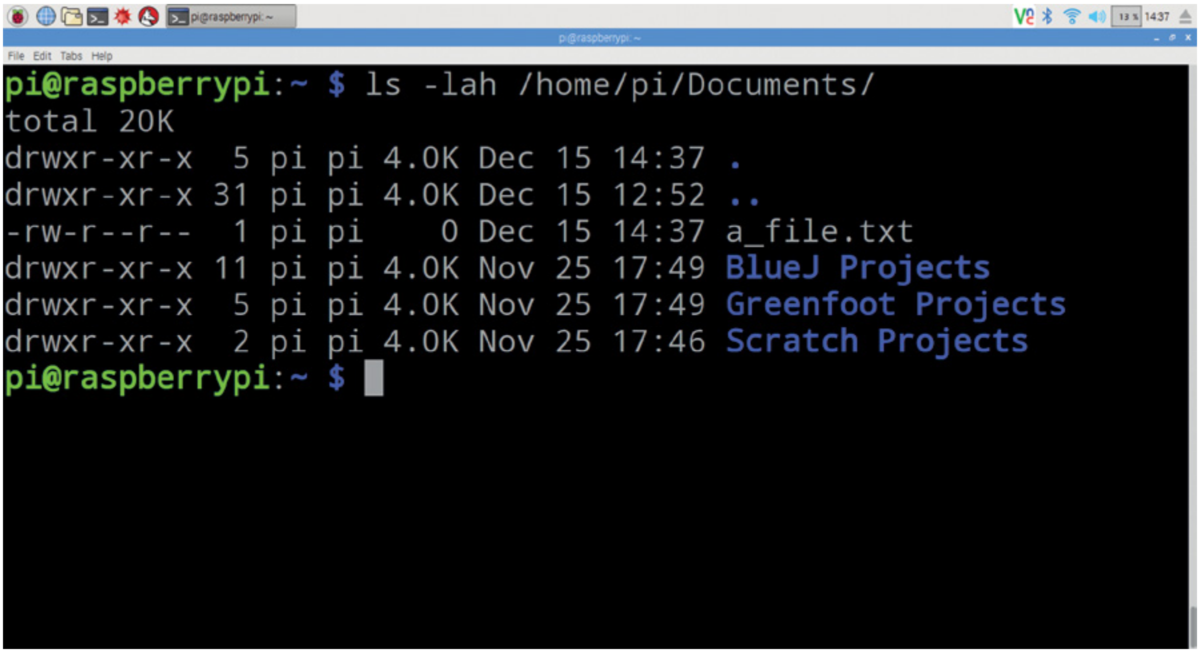
The first part of a command is the command itself. Here we have `ls`, which lists the contents of a directory.

`-lah`

After the command come options. These start with a hyphen and are typically single letters. Each modifies the command. Here we have 'l', 'a', and 'h'. These stand for long listing mode, all files, and human-readable.

`/home/pi/Documents`

The final part of the command is the arguments. These are often file names or file paths. Here we are listing an absolute (direct) path to the Documents directory. If you omit the argument, it'll display the contents of the current directory.



```
pi@raspberrypi:~ $ ls -lah /home/pi/Documents/
total 20K
drwxr-xr-x  5 pi pi 4.0K Dec 15 14:37 .
drwxr-xr-x 31 pi pi 4.0K Dec 15 12:52 ..
-rw-r--r--  1 pi pi    0 Dec 15 14:37 a_file.txt
drwxr-xr-x 11 pi pi 4.0K Nov 25 17:49 BlueJ Projects
drwxr-xr-x  5 pi pi 4.0K Nov 25 17:49 Greenfoot Projects
drwxr-xr-x  2 pi pi 4.0K Nov 25 17:46 Scratch Projects
pi@raspberrypi:~ $
```

The eagle-eyed reader may have noticed the weird backslash character: '\'. You can't have spaces in file names, so you use a backslash followed by a space at the command line. Most of the time you'll also use the **TAB** button to quickly enter long file names (see 'Tab completion').

File paths come in two types: relative and absolute. Relative paths are 'relative' to your working directory, which is **/home/pi/** when you start. Entering **`ls`** alone shows the contents of the current directory. You can view the contents of a directory inside your working directory using **`ls`** and its name:

```
ls Documents
```

You can also view the contents of the directory above you using two dots (..):

```
ls ..
```

This displays files relative to where you currently are in the file system. If you moved into the **Downloads** folder and entered **`ls Documents`**, it'd cause an error, because there is no **Documents** directory inside the **Downloads** folder.

An absolute path, on the other hand, always starts with a slash '/', which is the root directory (the base of your hard drive). Enter:

```
ls /
```

...to view the root directory. Here you'll see all the directories and files that make up Linux. You'll see directories like **bin** (for binaries), **boot** (files used to start up the system), and **home**. Enter:

```
ls /home/pi
```

...and you'll view the contents of your home folder, just as if you had entered **`ls`** from within it.

You can use absolute paths no matter what your working directory might be, because they always start from the root.

## Moving around

Up until now we've stayed in the home folder and looked around using **`ls`**. You move from one directory and another using the **`cd`** (change directory) command:

```
cd Documents
```

Now enter:

```
pwd
```

...and you'll see a different working path: **/home/pi/Documents**. To move back up a directory (known as the 'parent' directory), you use two dots.

```
cd ..
```

Enter **pwd** again and you're back in the home folder. Now try it using an absolute path. Enter:

```
cd /
```

...and you'll be in the root directory. Enter **ls** to view the folders at the base of your hard drive. Enter:

```
cd /home/pi
```

...to move back to your home folder. There's a shortcut for this:

```
cd ~
```

The tilde (~) character is a shortcut for your home folder. You can use it at the start of an absolute directory too.  For instance, entering:

```
cd ~/Downloads
```

...moves to your **Downloads** folder no matter where you are in the system.

## Files

Throughout the file system, you'll find various types of files. A good selection is in the **python_games** folder, so enter:

```
cd ~/usr/share/python_games
ls -l
```

The **-l** part (an option) makes it use 'long listing' mode, which displays items with lots of information:

```
-rw-rw-r--  1  pi  pi  973  Jan 27 2015
4row_arrow.png
```

From left to right, each item is:

- **Permissions:** The users and groups that can access a file.
- **Hard links:** The number of files that are linked to this file.
- **Owner:** The person who owns the file. Usually either **pi** or **root**.
- **Group:** The group the file belongs to.
- **File size:** The name of the file.
- **Modification:** When the file was last changed.
- **File name:** The name of the file.

The most obscure item is the list of letters and hyphens that make up the permissions. The first letter will be either a '–' or a 'd' depending on whether it's a file or a directory. Our **4row_arrow.png** is a file, so it's a '–'.

After that are nine letters arranged into three groups of three (see **Figure 1** overleaf):

- **Owner:** Typically this will be the person who created the account.
- **Group:** This is a group of users. You have only one group, **pi**, by default, containing just one user (also **pi**).
- **Other:** These are users from other systems.

Each of the three groups contains letters: 'rwx'. These letters are always in that order and each is either the letter or a hyphen. A letter indicates that the person, group, or other has access to read, write, or execute the file. A hyphen means they don't have that level of access.

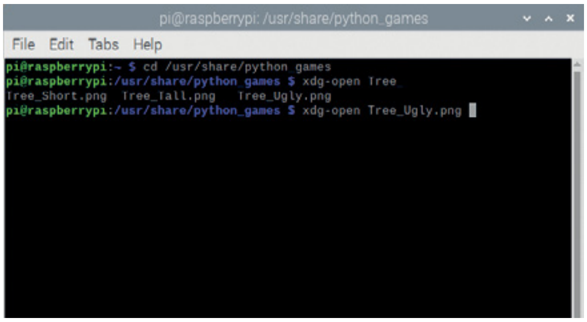
Some examples include:

- **rwx** read, write, and execute
- **rw-** read, write, but don't execute
- **r-x** read and execute
- **r--** read only

Now that you've discovered how to move around the file system from the command line, it's time to learn what else you can do.

## Take command

One of the first commands you need to learn is **mkdir**. This stands for 'make directory'. Move to the home folder and create a new directory called **test**:



```
pi@raspberrypi:~ $ cd /usr/share/python_games
pi@raspberrypi:/usr/share/python_games $ xdg-open Tree_
Tree_Short.png  Tree_Tall.png   Tree_Ugly.png
pi@raspberrypi:/usr/share/python_games $ xdg-open Tree_Ugly.png
```

## TAB COMPLETION

The single most useful tip you'll ever learn for the command line is tab completion. Pressing **TAB** at any time when entering a working path attempts to complete the file or directory name for you. Use **cd ~/usr/share/python_games** and enter **xdg-open Tr**, then press the **TAB** key. Notice how it fills it out to '**xdg-open Tree_**'. There are three files starting with Tree. Quickly press **TAB** twice and you'll see them: **Tree_Short.png**, **Tree_Tall.png** and **Tree_Ugly.png**. Enter **S**, **T**, or **U** and press **TAB** again to fill out the whole file name. Press **RETURN** to open it.

Tab completion can be invaluable for entering long file names packed with letters, numbers, and punctuation.

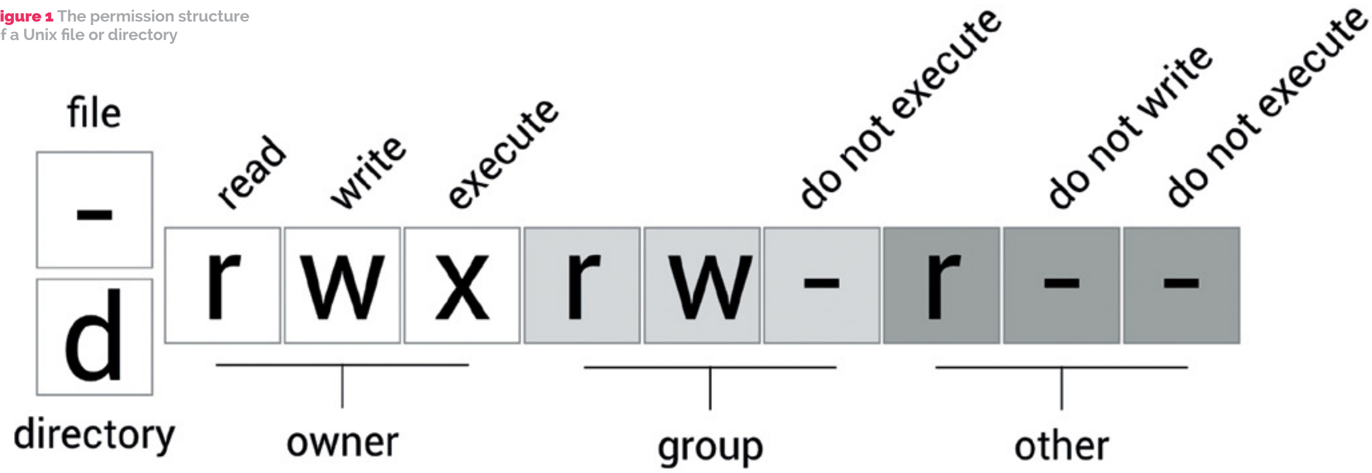


**Figure 1** The permission structure of a Unix file or directory

```
cd ~
mkdir test
cd test
```

To create files, you use the **touch** command; if the file doesn't exist, it creates an empty one. Enter:

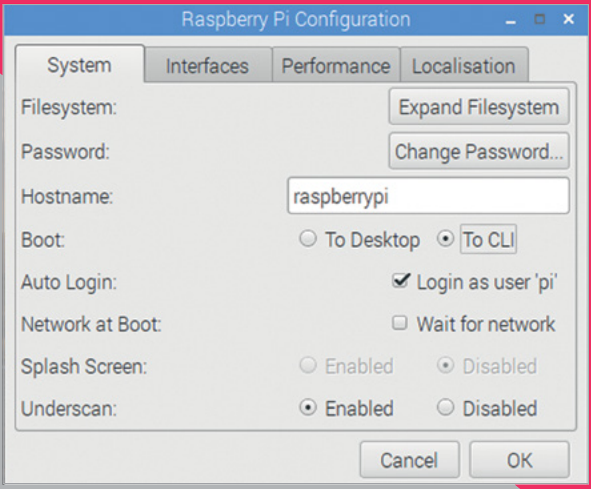
```
touch test.txt
```

You'll create a blank file called **test.txt**. Enter **ls -l** and you'll see the new file along with all its details. Notice that the file size is 0. This is because the file is completely empty. We can edit the contents of the file using a text editor called nano:

```
nano test.txt
```

You can enter and edit text in nano. Enter a single line, 'Hello World', and press **CTRL+O** followed by **ENTER** to save the file. Now press **CTRL+X** to exit.

Enter **ls -l** again; you'll notice that the file size has changed from 0 to 12. This is one for each letter (including space) and a newline marker at the end (you can see this character using **od -c test.txt** if you're curious).



Raspberry Pi Configuration

System | Interfaces | Performance | Localisation

Filesystem: Expand Filesystem

Password: Change Password...

Hostname: raspberrypi

Boot: ○ To Desktop ● To CLI

Auto Login: ☑ Login as user 'pi'

Network at Boot: □ Wait for network

Splash Screen: ○ Enabled ● Disabled

Underscan: ● Enabled ○ Disabled

Cancel OK

You can set Raspbian to boot into the command line (instead of the graphical interface) from the Raspberry Pi Configuration settings

Let's try deleting files. This command removes the file:

```
rm test.txt
```

Now move up to its parent directory and use another command, **rmdir**, to remove the empty **test** directory.

```
cd ..
rmdir test
```

You'll rarely use **rmdir** to remove directories, however, as it will fail to work if they have files in them. The solution is to use **rm** with the **-R** option. This option stands for 'recursive' and means it goes inside each directory (and subdirectory) and removes every file and directory. Be careful when using **rm -R**, as it's permanent and deletes everything inside. Enter:

```
rm -R test
```

The **test** directory and everything inside it will disappear. Unlike the desktop environment, there is no Wastebasket in the command line. When you remove files, they're instantly gone for good.

## Options

Most commands have options that affect how they work. It's common to use these three options with the **ls** command:

```
ls -lah
```

Options start with a single hyphen '-' followed the letter for each option. The three options used here are:

- l = long listing format

- a = all including hidden files
- h = human-readable (makes large file sizes more readable)

Options are case-sensitive. So **ls -l** and **ls -L** are two different things (small 'l' is long listing format; large 'L' is dereference mode). Sometimes options are listed out in full. These start with two hyphens and have a single hyphen for spaces. This command is the same as **ls -lah**:

```
ls -l --all --human-readable
```

But it's more common to see (and use) the single letter approach.

## Sudo

Sudo stands for 'substitute user do', although it's often also called 'superuser do'. If you have multiple users on your system, it can be used to perform commands as another user.

It's mostly used to get root access to your Linux installation. There is an account that controls your pi user, called 'root'. This is an all-powerful account, which can change just about anything on your system.

Your default account can view files in the root of the hard drive, but it can't create or delete files at the root. Enter:

```
cd /
touch test.txt
```

You'll see **touch: cannot touch 'test.txt': Permission denied**. However, enter:

```
sudo touch test.txt
```

...and the **test.txt** file will be created on the root of your hard drive. You can see it using **ls -l**.

Now try to delete it:

```
rm test.txt
```

It will say **rm: remove write-protected regular empty file 'test.txt'?"** Enter **Y** and it'll say **rm: cannot remove 'test.txt': Permission denied**.
You need to use sudo to remove the file:

```
sudo rm test.txt
```

You can see why sudo is such a powerful tool. Without it, you couldn't install software using apt or apt-get. But with it, you can remove or delete vital system files. Enter **ls /bin** and you'll see many programs (known as 'binaries') used by Linux. These include the **ls** command you just used. Accidentally deleting these files could make your system unstable.

So use sudo with care. In Raspbian you don't need to enter the password to use sudo. On many other Linux systems, however, you will be asked for the password before you can use sudo.

## What's up, man?

There are lots of ways of getting help inside the command line. The first command you should turn to is **man**. This stands for 'manual' and gives you instructions on Linux commands and tools. Enter:

```
man ls
```

...and you'll see the manual for the list command. Notice under the SYNOPSIS it says:

```
ls [OPTION]... [FILE]...
```

This shows you the structure of the command. Almost all commands are in the 'command, option, argument' structure, although some arguments have more than one [FILE] argument (such as copy, which requires a source and destination file).

Press the space bar to move down the instructions. Here you will see a list of all the available options. With **man**, you can get detailed information on just about every tool on the command line. You can even get a manual for the **man** command with:

```
man man
```

If you need a quick reminder on how to use a command, try using it with **-h** or **--help** as an option:

```
touch --help
```

...tells you what options are available with the **touch** command. You can use this with many command-line tools to get a quick refresher on how they work.

Moving from a GUI to a command line is a vital skill for hackers and coders. Everything on your computer, from programs to preferences, is stored in the file system somewhere. Learning to use the command line makes you a more capable Raspberry Pi user.

So, the next time you make a file, move a file, or delete something, don't head to the File Manager. Open Terminal and perform tasks from the command line. Soon it'll be second nature.

# Raspberry Pi 3A+

Gone too long, the A+ is back with the power of Raspberry Pi 3 behind it

The last time the Raspberry Pi family had a new A-form-factor design, more compact and lightweight than the full-size B and B+ ranges, it was Raspberry Pi A+, released four years ago. Since then, the family has grown considerably: we've seen Raspberry Pi 2, 3, Zero, Zero W and WH, 3B+ and, most recently, Raspberry Pi 4.

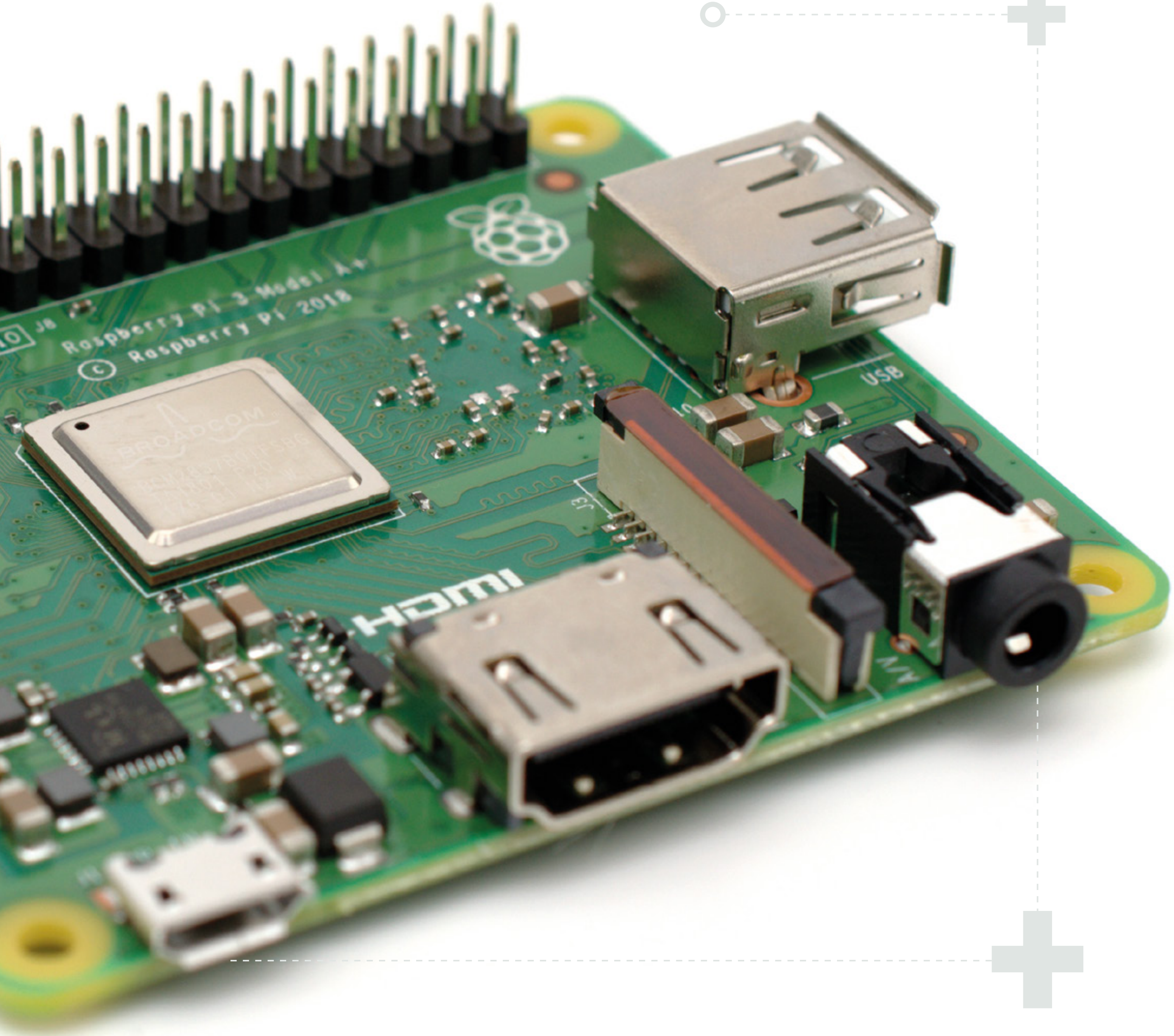
> Packs the power of Raspberry Pi 3B+ into the same footprint as the original Raspberry Pi A+

Launched in November 2018, Raspberry Pi 3A+ packs the power of Raspberry Pi 3B+ into the same footprint as the original Raspberry Pi A+. Read on for the lowdown on what Eben Upton describes as "the best product we can make."

## UK suppliers:

The Pi Hut
magpi.cc/zVTpyC

Pimoroni
magpi.cc/LhuKdS



CPC
magpi.cc/algkOF

RS Components
magpi.cc/nDkZGZ

For a complete list of resellers globally where the new Raspberry Pi 3 Model A+ can be purchased, please see rpf.io/products

# Meet Raspberry Pi 3A+

This mini marvel squeezes a lot into its compact footprint

## SPECS

**SoC:** Broadcom BCM2837B0 quad-core A54 (ARMv8) 64-bit @ 1.4GHz

**GPU:** Broadcom VideoCore IV

**NETWORKING:** 2.4GHz and 5GHz 802.11b/g/n/ac wireless LAN

**RAM:** 512MB LPDDR2 SDRAM

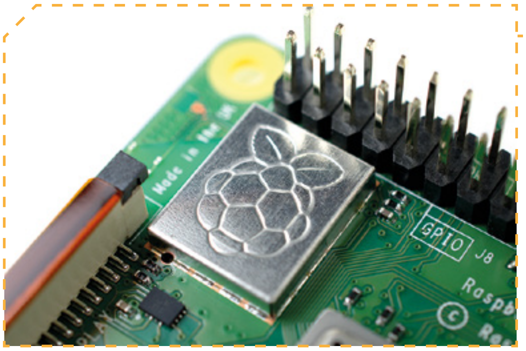
**BLUETOOTH:** Bluetooth 4.2, Bluetooth Low Energy (BLE)

**GPIO:** 40-pin GPIO header, populated

**STORAGE:** microSD

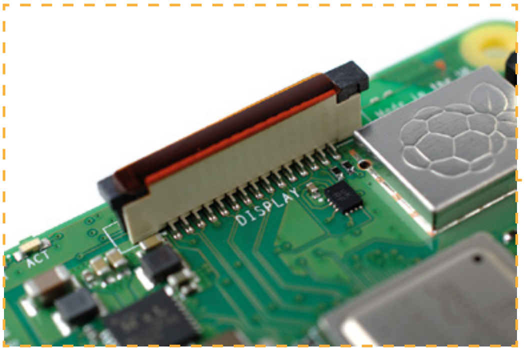
**PORTS:** HDMI, 3.5 mm analogue audio-video jack, 1 × USB 2.0, Camera Serial Interface (CSI), Display Serial Interface (DSI)

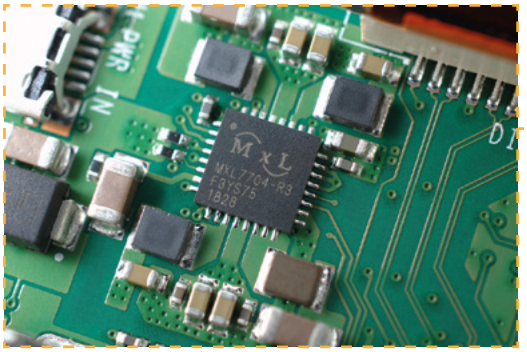
**DIMENSIONS:** 67×56×11.5 mm



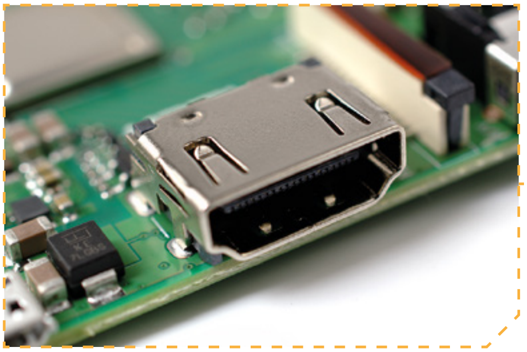
▲ The radio, originally developed for 3B+, makes Raspberry Pi 3A+ the first A-model to feature on-board networking

▼ Full-size display (DSI) and camera (CSI) ports make connecting peripherals a snap





◀ Raspberry Pi 3A+ benefits from the same clever power management chip (PMIC) as 3B+



▲ Unlike the Zero family, Raspberry Pi 3A+ includes both full-size HDMI and analogue audio-video outputs

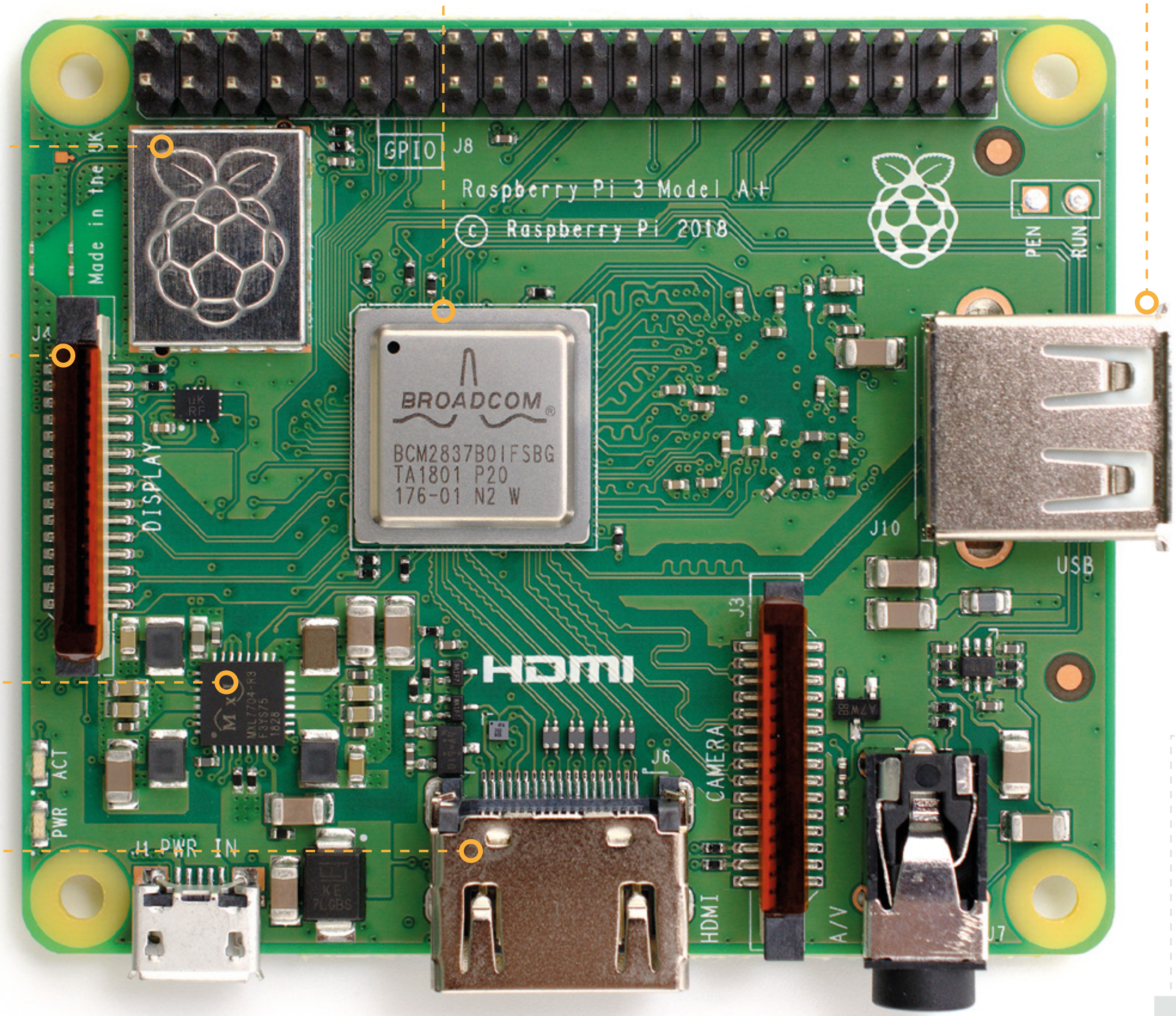
## QuickStart guide

Raspberry Pi 3A+ is directly compatible with 3B+ and all other Raspberry Pi models. If you've already got a power supply and microSD card with a new version of Raspbian installed, you're ready to go. If 3A+ is your first Raspberry Pi, check out pages 6–13, or see our online QuickStart guide to get up and running fast.

**magpi.cc/quickstart**

◀ Raspberry 3A+ has the same 1.4GHz system-on-chip (SoC) as 3B+, with no loss of performance

▶ The biggest change from Raspberry Pi 3B+: the reduction from four to one USB 2.0 ports



# Benchmarking
## Raspberry Pi 3A+

Small but mighty, the reduction in size doesn't mean Raspberry Pi 3A+ loses any performance compared to 3B+

For projects where a full-size Raspberry Pi is simply too bulky, there hasn't been much choice: aside from the Compute Module 3, which is targeted at industrial users, the choices were the four-year-old Pi A+ or ultra-compact Pi Zero family – and in either case it meant a dramatic drop in performance.

Raspberry Pi 3A+ comes with the promise of full-size performance in a small, lightweight, and less power-hungry package, but the only way to see if that is true or not is to put it through its paces in a range of benchmarks.

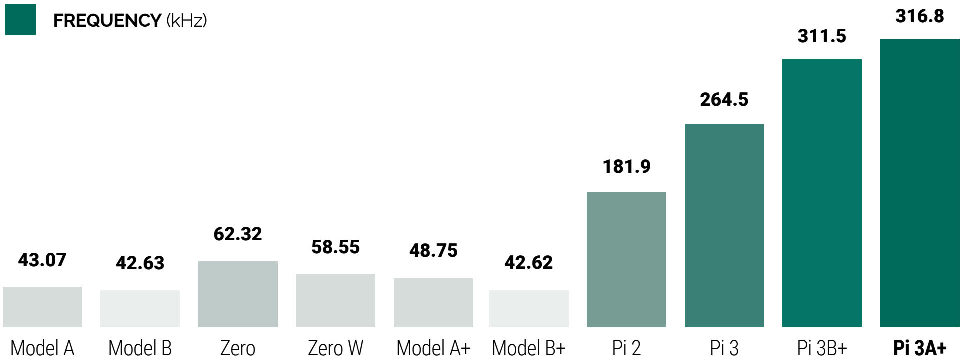
### Spec comparison

At its heart, Raspberry Pi 3A+ is a cut-down version of 3B+. It has the same Broadcom BCM2837B0 system-on-chip (SoC) roughly in the centre of the board, hidden under a metal heat-spreader, which runs at the same 1.4GHz frequency. While 512MB of RAM is less than the 1GB of a 3B+, the smaller Raspberry Pi 3A+ can certainly hold its own.

Looking back at the original Raspberry Pi A+, it's hard to imagine they're from the same family: from a single-core 32-bit 700MHz processor and no networking to a quad-core 64-bit 1.4GHz processor with built-in wireless LAN and Bluetooth. What an upgrade!

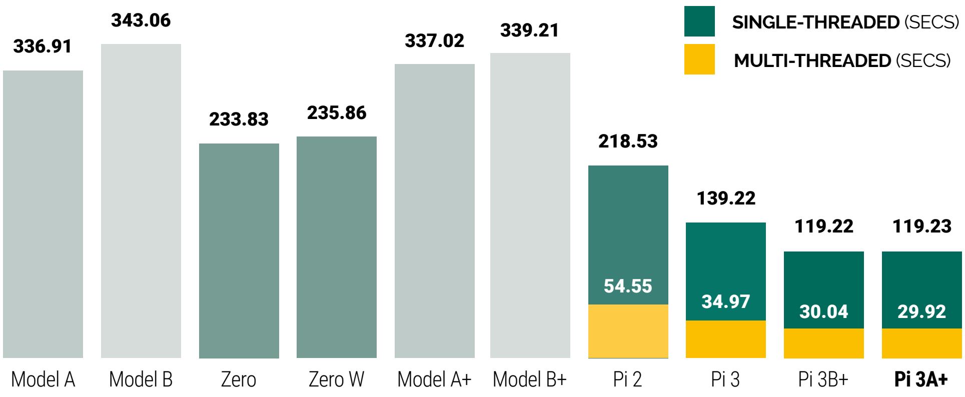
### Python GPIO
**Higher is better**

Designed to highlight a real-world bottleneck, the Python GPIO benchmark switches a single pin on and off while a frequency counter measures how quickly the pin is toggled. The faster the processor, the faster the pin can be toggled before the processor hits its limit.



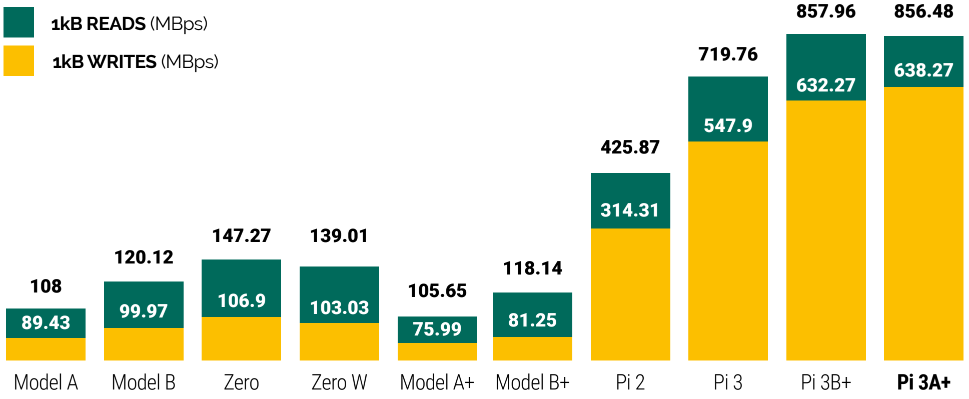
### SysBench CPU
**Lower is better**

Focusing on the central processor's performance, the SysBench CPU benchmark tests how quickly a Raspberry Pi can perform prime number calculations. For Raspberry Pi models with quad-core processors, the test is run twice: once with a single thread using only one of the cores, and again with four threads using all four cores.



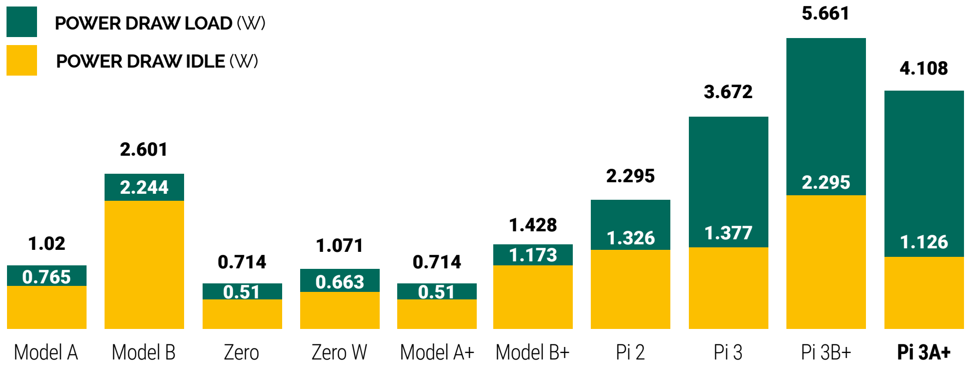
## SysBench
Memory Throughput
**Higher is better**

Processor performance is only part of the puzzle regarding overall system performance: in the SysBench Memory Throughput test, measurements show how quickly a Raspberry Pi can read and write to the random-access memory (RAM) in 1kB chunks, reported in megabytes per second (MBps).



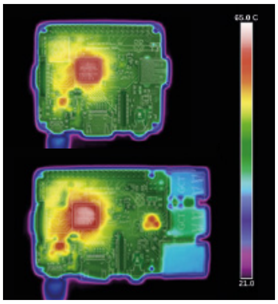
## Power Draw
**Lower is better**

More performance typically means more power used. Here, each Raspberry Pi is connected to an HDMI display, wireless keyboard, and, where applicable, a WiFi or wired Ethernet network before two measurements are taken: power used while Raspberry Pi is sat idle at the desktop, and again while running a CPU-heavy application.



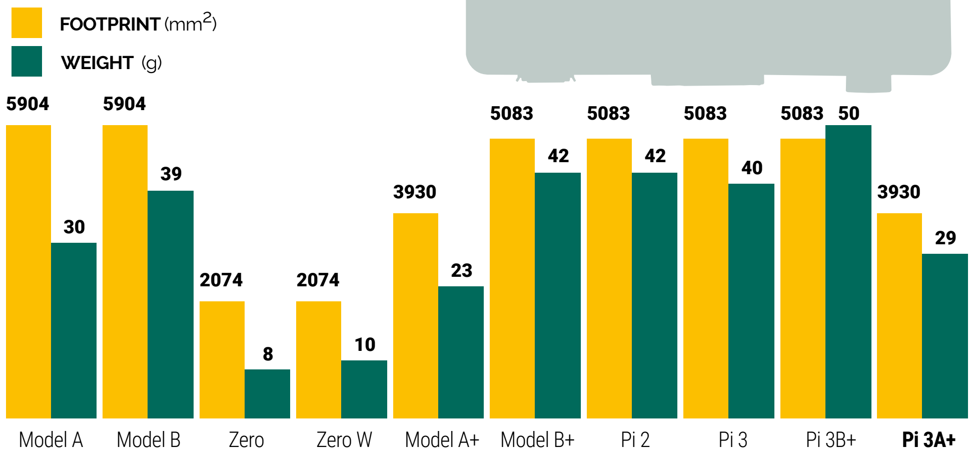
## Thermal Performance
**Cooler is better**

Raspberry Pi 3B+ benefited from a change to the way the system-on-chip (SoC) is attached to the circuit board, allowing it to better dissipate heat. With Raspberry Pi 3A+ having a smaller board, this test captures thermal images under heavy CPU load to show how well the two designs cope.



## Size & Weight
**Lower is better**

The biggest difference between Raspberry Pi 3B+ and 3A+ is their respective sizes. Here, the footprint of each model is measured from its widest points – to include the size of the ports which sit proud of the board – and its weight measured, both important aspects for embedded and robotics applications.



# Eben Upton and Roger Thornton on Raspberry Pi 3 A+

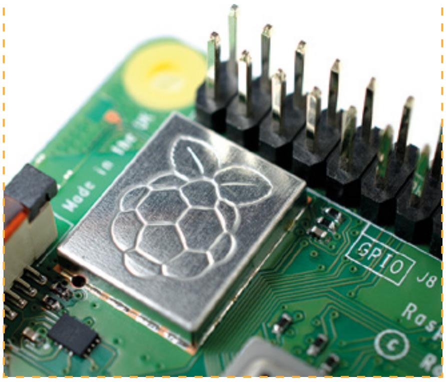
Eben and Roger talk about Raspberry Pi 3 A+'s inspiration and launch timing

## Simplifying certification

"What's the cost to do a certification campaign," Eben asks. "Couple of hundred thousand?"

"For a five-gig one, yeah, it's about 250… 300 000," answers Roger – the cost of clearing a radio-containing product like the Raspberry Pi 3 for general sale.

Raspberry Pi 3B+ moved to a modular radio certification, so the existing compliance work remains largely valid for 3A+. "A lot of the work we'd done to modularise the board meant that we could copy over pretty much all of our compliance work," Roger explains, "so thankfully this product's been just a bit of paperwork. There's definitely been an improvement in time-to-market."



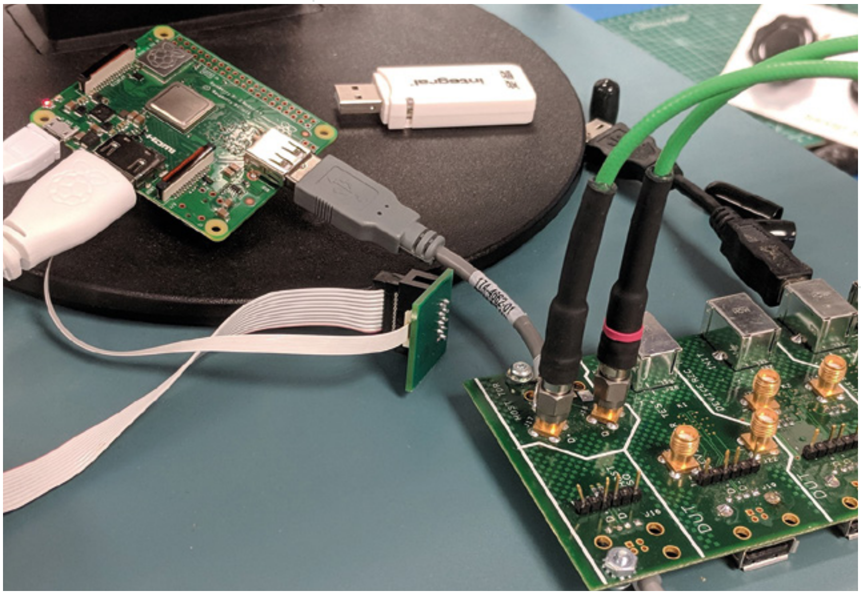
"We describe our product design process as resembling a bunch of trains driving along, because we're dealing with a bunch of subsystems," Eben Upton, co-founder of the Raspberry Pi Foundation, explains of the development process that has delivered Raspberry Pi 3 Model A+. "Some subset of the trains will arrive at the station, and when enough of them arrive then we've got a new product we can make. We don't have a big team of marketing guys who make PowerPoints about our roadmap. It's much more coffee-table-oriented than that."

## Bringing back A+

"The A+ form factor was always a good form factor," says Roger Thornton, Principal Hardware Engineer. "A lot of people have asked us for an A+ again, and we were able to make that, so we did."

With both Raspberry Pi 2 and 3 having launched exclusively in larger B-variant form factors, the A+ has been absent from the line-up for a full two generations. "It was skipped for Raspberry Pi 2 because it simply wasn't around

▼ As with any Raspberry Pi, 3A+ undergoes in-depth testing before it's released to the public



"If we have an opportunity to bring our best tech down a bit, then we do," explains Eben Upton

"A lot of people have asked us for an A+ again, and we were able to make that," says Roger Thornton of the launch

for long enough," recalls Eben. "It normally takes six to nine months for us to get our feet under ourselves, and by that time it was very clear that the 3B was imminent."

"The Zero W happened the year after the 3B," Roger adds, "and to some extent addressed the gap that an A+ might have occupied."

The Zero family isn't a complete replacement for the A+, though, as the launch of Raspberry Pi 3B+ hammered home.

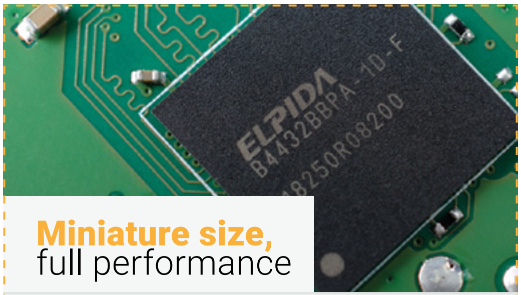
## Filling a hole

"There's a massive performance gulf between the 3B+ and Zero," explains Eben. "3A+ represents us bringing our best technology to the lowest possible price point. We can't bring this technology down to the Zero price point, at the moment, but we can bring it down a bit.

"We don't just like to sit there at $35 feeling pleased with ourselves. If we have an opportunity to bring our best tech down a bit, then we do. I mean, this is the best product we can make. We don't know how to make anything better at the moment. It's lovely, actually. We like to make physically beautiful objects. They sometimes cost a bit of money to make, or cost a bit of blood, I think, and this… I mean, for me, once we knew we could make it, I think we kind of had to.

"There may be another motivation for doing it: it's kind of tidying up 'classic' Raspberry Pi," Eben admits. With their attention turning to Raspberry Pi 4 – since launched in June 2019 – the hardware design team were starting the research phase of figuring out what they might do next. "Tying up loose ends is an important part of that, and the lack of an A+ form-factor product is a loose end," says Eben.

> "This is us bringing our best technology to the lowest possible price point"



### Miniature size, full performance

"We're confident that the performance you saw on a 3B+ you'll be able to get on a 3A+," says Roger. "There's no reduction because of the board size change; you should still be able to run the processor at the same speed. We have had to reduce the memory to hit the price point we're after, but it still runs nicely."

"When we launched the original A+, we launched it with half the RAM of the original B+," Eben adds. "At the time that meant 256MB; now halving the RAM still leaves you with half a gig, which is kind of fun."

Raspberry Pi

## Raspberry Pi 3 Model A+/B+ Safety and User guide

Designed and distributed by
Raspberry Pi
Maurice Wilkes Building
Cowley Road
Cambridge
CB4 0DS
UK
www.raspberrypi.org

**Raspberry Pi** Regulatory compliance and safety information
Product name: Raspberry Pi 3 Model A+/B+
IMPORTANT: PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE.

Warnings

- Any external power supply used with the Raspberry Pi shall comply with relevant regulations and standards applicable in the country of intended use. The power supply should provide 5V DC and a minimum rated current of 2.5A.

Instructions for safe use

- This product should not be overclocked.
- Do not expose this product to water or moisture, and do not place it on a conductive surface whilst in operation.
- Do not expose this product to heat from any source; it is designed for reliable operation at normal room temperatures.
- Operate this product in a well-ventilated environment, and do not cover it during use.
- Place this product on a stable, flat, non-conductive surface while in use, and do not let it contact conductive items.
- Take care while handling this product to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling this product while it is powered. Only handle by the edges to minimize the risk of electrostatic discharge damage.
- Any peripheral or equipment used with the Raspberry Pi should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. Such equipment includes, but is not limited to, keyboards, monitors, and mice.

For all compliance certificates and numbers, please visit www.raspberrypi.org/compliance.

### 汉语

Raspberry Pi 3 代A+/ B+型
重要提示：请保留此信息以供将来参考。

警告

- Raspberry Pi 使用的任何外置电源应符合所在国家的相关法规和标准。 电源应提供 5V DC 和 2.5A 最小额定电流。

安全使用说明

- 此产品不可超频。
- 请勿将本产品暴露在水或潮湿环境中；当其运行时，请勿将其置于导电表面上。
- 请勿将此产品暴露于任何热源；此产品仅适合在正常室温中使用，以确保可靠运行。
- 在通风良好的环境中运行此产品, 在使用过程中请勿覆盖。
- 使用时，请将本产品放在稳定、平坦、绝缘的表面上，请勿让它接触导电物品。
- 拿放本产品时请小心，以免对印刷电路板和连接器造成机械或电气损坏。
- 本产品通电时避免接触、拿放。 拿放时应只接触产品的边缘，以最大限度地降低静电放电损坏的风险。
- Raspberry Pi使用的任何外设或设备应符合使用国家的相关标准，并进行相应标记，以确保满足安全和性能要求。

如需查询所有合规证书及编号，请访问 www.raspberrypi.org/compliance 。

### Čeština

Raspberry Pi 3 Model A+/ B+
DŮLEŽITÉ: TUTO INFORMACI SI PONECHTE PRO POUŽITÍ V BUDOUCNU.

Varování

- Každý externí napájecí zdroj použitý s Raspberry Pi musí splňovat příslušné předpisy a normy platné v zemi určení. Napájecí zdroj by měl poskytovat stejnosměrné napětí 5V a minimální jmenovitý proud 2,5A.

Pokyny pro bezpečné používání

- Tento výrobek by neměl být přetaktován.
- Výrobek nevystavujte vodě ani vlhkosti a za provozu ho neumisťujte na vodivý povrch.
- Výrobek nevystavujte teplu z jakéhokoli zdroje; je navržen pro spolehlivý provoz při normálních pokojových teplotách.
- Výrobek používejte v dobře větraném prostředí a za provozu ho nepřikrývejte.
- Výrobek při používání ponechte na stabilním, plochém a nevodivém povrchu a zabraňte jeho dotyku s vodivými předměty.
- Při manipulaci s výrobkem dbejte na to, abyste zabránili mechanickému nebo elektrickému poškození desky plošných spojů a konektorů.
- Vyvarujte se manipulace s výrobkem, když je napájen. K manipulaci používejte pouze okraje, abyste minimalizovali riziko poškození elektrostatickým výbojem.
- Veškerá periferní a další zařízení používaná s Raspberry Pi by měla být v souladu s příslušnými normami země použití a měla by být odpovídajícím způsobem označena, aby se zajistilo, že splňují požadavky na bezpečnost a výkon.

Všechna osvědčení o shodě a čísla najdete na www.raspberrypi.org/compliance.

### Dansk

Raspberry Pi 3 Model A+/  B+
VIGTIGT: OPBEVAR DENNE INFORMATION FOR FREMTIDIG REFERENCE.

Advarsler

- Eksterne strømforsyninger, der anvendes til Raspberry Pi skal være i overensstemmelse med relevante bestemmelser og standarder, som er gældende i det land, hvor anvendelsen er tiltænkt. Strømforsyningen skal give 5 V jævnstrøm og en nominel strømstyrke på mindst 2,5 A.

Instruktioner for sikker brug

- Dette produkt må ikke overophedes.
- Udsæt ikke dette produkt for vand eller fugt, og sæt det ikke på en ledende overflade under drift.
- Udsæt ikke dette produkt for varme fra nogen kilder; det er designet til pålidelig drift ved normal stuetemperatur.
- Anvend dette produkt i et godt ventileret miljø, og tildæk det ikke under brug.
- Anbring dette produkt på en stabil, flad og ikke-ledende overflade under brug, og lad det ikke komme i berøring med ledende genstande.
- Vær forsigtig ved håndtering af dette produkt for at undgå mekanisk eller elektrisk beskadigelse af printkort og stik.
- Undgå håndtering af dette produkt, mens det er tændt. Må kun håndteres ved at holde i kanterne for at minimere risikoen for skader ved elektrostatisk udladning.
- Alt perifert udstyr eller udstyr, der anvendes til Raspberry Pi skal overholde relevante standarder i landet for anvendelse og mærkes i overensstemmelse hermed for at sikre, at kravene for sikkerhed og ydeevne er opfyldt.

For alle overensstemmelsescertifikater og numre, gå på www.raspberrypi.org/compliance.

### Nederlands

Raspberry Pi 3 Model A+/ B+
BELANGRIJK: BEWAAR DEZE INFORMATIE VOOR TOEKOMSTIGE VERWIJZING.

Waarschuwingen

- Elke externe voeding die met de Raspberry Pi wordt gebruikt, moet voldoen aan de relevante voorschriften en normen die van toepassing zijn in het land van het beoogde gebruik. De voeding moet 5V DC en een minimale nominale stroom van 2,5A leveren.

Instructies voor veilig gebruik

- Dit product mag niet overklokt worden.
- Stel dit product niet bloot aan water of vocht en plaats het tijdens gebruik niet op een geleidend oppervlak.
- Stel dit product niet bloot aan warmte van welke bron dan ook; het is ontworpen voor betrouwbare werking bij normale kamertemperatuur.
- Gebruik dit product in een goed geventileerde omgeving en dek het niet af tijdens gebruik.
- Plaats dit product tijdens het gebruik op een stabiel, plat, niet-geleidend oppervlak en laat het niet in contact komen met geleidende items.
- Wees voorzichtig met het gebruik van dit product om mechanische of elektrische schade aan de printplaat en connectoren te voorkomen.
- Gebruik dit product niet terwijl het wordt gevoed. Alleen aan de randen vasthouden om het risico op schade door elektrostatische ontlading te minimaliseren.
- Alle randapparatuur of apparatuur die met de Raspberry Pi wordt gebruikt, moet voldoen aan de relevante normen voor het land van gebruik en dienovereenkomstig worden gemarkeerd om ervoor te zorgen dat wordt voldaan aan de veiligheids- en prestatie-eisen.

Ga naar www.raspberrypi.org/compliance.

### Suomi

Raspberry Pi 3 Malli A+/ B+
TÄRKEÄÄ: SÄILYTÄ NÄMÄ TIEDOT MYÖHEMMÄN KÄYTÖN VARALTA.
Varoituksia

- Kaikkien ulkoisen Raspberry Pi -laitteessa käytettyjen virtalähteiden on noudatettava käyttömaassa sovellettavia asiaankuuluvia asetuksia ja standardeja. Virtalähteen virran on oltava 5V DC minimin nimellisvirran ollessa 2,5A.

Ohjeet turvallista käyttöä varten

- Tätä tuotetta ei saa ylikuormittaa.
- Älä altista tätä tuotetta vedelle tai kosteudelle, äläkä aseta sitä johtavalle pinnalle sen ollessa toiminnassa.
- Älä altista tätä tuotetta mistään lähteestä aiheutuvalle kuumuudelle; se on suunniteltu luotettavaa toimintaa varten normaaleissa huonelämpötiloissa.
- Käytä tätä tuotetta hyvin ilmastoidussa lämpötilassa, äläkä peitä sitä käytön aikana.
- Aseta tämä tuote vakaalle, tasaiselle, ei-johtavalle pinnalle sen ollessa käytössä, äläkä anna sen koskettaa johtavia kohteita.
- Noudata varovaisuutta tätä tuotetta käsiteltäessä mekaanisen tai sähköisen vaurioitumisen estämiseksi painetulle piirilevylle ja liittimille.
- Vältä tämän tuotteen käsittelyä sen ollessa kytkettynä virtalähteeseen. Käsittele vain reunoista sähköstaattisen purkautumisen vaaran minimoimiseksi.
- Kaikkien Raspberry Pi -laitteiden kanssa käytettävien oheislaitteiden ja muiden laitteiden on oltava käyttömaan asianmukaisten standardien mukaisia, ja niiden on oltava merkittyjä sen varmistamiseksi, että turvallisuus ja suorituskykyvaatimukset täytetään.

Lisätietojen saamiseksi kaikista vaatimustenmukaisuussertifikaateista vieraile verkkosivustolla www.raspberrypi.org/compliance.

### Français

Raspberry Pi 3 Modèle A+/ B+
IMPORTANT: VEUILLEZ CONSERVER CETTE INFORMATION POUR VOUS Y REPORTER ULTÉRIEUREMENT.

Avertissements

- Toute alimentation électrique externe utilisée avec le Raspberry Pi doit être conforme aux réglementations et normes applicables dans le pays d'utilisation. L'alimentation électrique doit fournir 5 V CC et un courant nominal minimum de 2,5 A.

Consignes pour une utilisation en toute sécurité

- Ce produit ne doit pas être utilisé à une vitesse supérieure à celle prévue pour son usage.
- N'exposez pas ce produit à l'eau ou à l'humidité et ne le placez pas sur une surface conductrice pendant le fonctionnement.
- N'exposez pas ce produit à la chaleur quelle qu'en soit la source; il est conçu pour un fonctionnement fiable à des températures ambiantes normales.
- Faites fonctionner ce produit dans un environnement bien ventilé et ne le couvrez pas pendant l'utilisation.
- Placez ce produit sur une surface stable, plane et non conductrice pendant son utilisation et ne le laissez pas en contact avec des éléments conducteurs.

- Faites attention lors de la manipulation de ce produit pour éviter tout dommage mécanique ou électrique au niveau de la carte de circuit imprimé et des connecteurs.
- Évitez de manipuler ce produit lorsqu'il est sous tension. Ne manipulez que par les bords afin de minimiser les risques de dommages dus aux décharges électrostatiques.
- Tout périphérique ou équipement utilisé avec le Raspberry Pi doit être conforme aux normes applicables dans le pays d'utilisation et être marqué en conséquence pour garantir la sécurité et les performances.

Pour tous les certificats et numéros de conformité, veuillez consulter www.raspberrypi.org/compliance

### Deutsch

Raspberry Pi 3 Modell A+/ B+
WICHTIG: BITTE BEWAHREN SIE DIESE INFORMATIONEN FÜR ZUKÜNFTIGE REFERENZ.

Achtung

- Jedes externe Netzteil, das mit dem Raspberry Pi verwendet wird, muss den einschlägigen Vorschriften und Normen entsprechen, die im Bestimmungsland gelten. Die Stromversorgung sollte 5 V Gleichstrom und einen minimalen Nennstrom von 2,5 A liefern.

Anweisungen für die sichere Verwendung

- Dieses Produkt sollte nicht übertaktet werden.
- Setzen Sie dieses Produkt nicht Wasser oder Feuchtigkeit aus und stellen Sie es während des Betriebs nicht auf eine leitfähige Oberfläche.
- Setzen Sie dieses Produkt keiner Wärmequelle aus. Es ist für einen zuverlässigen Betrieb bei normalen Raumtemperaturen ausgelegt.
- Betreiben Sie dieses Produkt in einer gut belüfteten Umgebung und decken Sie es während des Gebrauchs nicht ab.
- Stellen Sie dieses Produkt während des Gebrauchs auf eine stabile, flache, nicht leitende Oberfläche und lassen Sie es nicht mit leitfähigen Gegenständen in Berührung kommen.
- Seien Sie vorsichtig beim Umgang mit diesem Produkt, um mechanische oder elektrische Schäden an der Leiterplatte und den Anschlüssen zu vermeiden.
- Vermeiden Sie die Handhabung dieses Produkts während der Stromversorgung. Produkt nur an den Rändern anfassen, um das Risiko von elektrostatischen Entladungsschäden zu minimieren.
- Alle Peripheriegeräte oder Geräte, die mit dem Raspberry Pi verwendet werden, müssen den geltenden Normen für das jeweilige Land entsprechen und entsprechend gekennzeichnet sein, um zu gewährleisten, dass die Sicherheits- und Leistungsanforderungen erfüllt werden.

Alle Konformitätszertifikate und -nummern finden Sie auf www.raspberrypi.org/compliance.

### Italiano

Raspberry Pi 3 Model A+/ B+
IMPORTANTE: CONSERVARE QUESTE INFORMAZIONI PER RIFERIMENTO FUTURO.

Avvisi

- Tutti gli alimentatori esterni utilizzati con il Raspberry Pi devono essere conformi alle normative e agli standard pertinenti applicabili nel paese di utilizzo previsto. L'alimentatore utilizzato dovrà fornire 5 V CC e una corrente nominale minima di 2,5 A.

Istruzioni per l'utilizzo in sicurezza

- Questo prodotto non deve essere overcloccato.
- Non esporre questo prodotto all'acqua o all'umidità e non collocarlo su una superficie conduttiva mentre è in funzione.
- Non esporre questo prodotto a fonti di calore. Il prodotto è progettato per un funzionamento affidabile solo alla normale temperatura ambiente.
- Utilizzare questo prodotto in un ambiente ben ventilato e non coprirlo durante l'uso.
- Per l'uso, collocare questo prodotto su una superficie stabile, piana e non conduttiva. Evitare che venga in contatto con oggetti conduttivi.
- Durante l'uso o lo spostamento del prodotto prestare attenzione ad evitare danni meccanici o elettrici al circuito stampato e ai connettori.
- Evitare di maneggiare questo prodotto mentre è alimentato. Afferrare solo dai bordi per ridurre al minimo il rischio di danni da scariche elettrostatiche.
- Tutte le periferiche e le apparecchiature utilizzate con il Raspberry Pi devono essere conformi agli standard pertinenti per il paese di utilizzo ed essere dotate del relativo marchio a garanzia della conformità con i requisiti di sicurezza e prestazioni necessari.

Per informazioni su numeri e certificati di conformità, visitare www.raspberrypi.org/compliance.

### 日本語

Raspberry Pi 3 モデルA+/ B+
重要: 将来参照できるようこの情報は保管しておいてください。

警告

- Raspberry Pi と共に使用する外部電源は使用対象国内の規制や基準に準拠したものにしてください。 電源の出力は 5V DC で最低定格電流が 2.5A でなければなりません。

安全な使用のための説明

- 本製品をオーバークロックしてはなりません。
- 本製品を水や湿気にさらしたり、動作中に導電性の面に置いてはなりません。
- 本製品をどんな熱源からの熱にもさらさないでください。本製品は通常の室温で動作するように設計されています。
- 本製品は通気性の良い環境で使用し、使用中に密閉しないでください。
- 本製品の使用中は本製品を平らな安定した非導電性の面に置き、導電性の物に接触させないでください。
- 本製品を扱う際は慎重に取り扱い、プリント基板およびコネクター類への物理的または電気的損傷を避けてください。
- 電源が入っている状態で本製品に触れることは避けてください。 静電放電による損傷のリスクを最小限にするため、持つ際は端を持ってください。
- 安全と性能の要件を満たすため、Raspberry Pi と共に使用する周辺機器または装置は使用国内の規制や基準に準拠したものにし、該当する表記のあるものにしてください。

すべての順守証明書および番号については www.raspberrypi.org/compliance を参照してください

### 한국어

Raspberry Pi 3 모델A+/ B+
중요: 추후 참조를 위해 이 정보를 보관하십시오.

경고

- Raspberry Pi 와 함께 사용되는 모든 외부 전원 공급 장치는, 해당 국가에서 적용되는 관련 규정 및 표준을 준수해야합니다. 전원 공급 장치는 5V DC, 최소 정격 전류 2.5A를 공급해야 합니다.

안전한 사용을 위한 지침

- 본 제품을 '오버클럭' 해서는 안됩니다.
- 본 제품을 물이나 습기에 노출시키지 말고, 작동 중에 전도성 표면 위에 놓지 마십시오.
- 본 제품을 모든 소스의 열에 노출시키지 마십시오. 일반적인 실내 온도에서 안정적인 작동을 하도록 설계되었습니다.
- 본 제품을 통풍이 잘되는 환경에서 사용하고, 사용 중에는 덮지 마십시오.
- 이 제품을 사용하는 동안 안정적이고 편평한 비전도성 표면에 놓고, 전도성 물품에 접촉시키지 마십시오.
- 인쇄회로기판 및 커넥터의 기계적 또는 전기적 손상을 방지하기 위해, 본 제품을 취급할 때 주의하십시오.
- 전원이 공급되는 동안에는 본 제품을 다루지 마십시오. 정전기로 인한 손상 위험을 최소화하기 위해 가장자리만 잡으십시오.
- Raspberry Pi 와 함께 사용되는 모든 주변 장치 또는 장비는, 해당 사용 국가의 관련 표준을 준수해야 하며 또한 안전 및 성능 요구사항을 충족하도록 표시해야 합니다.

모든 준수 인증서 및 번호는, 다음 사이트를 참조하십시오. www.raspberrypi.org/compliance.

### Polski

Raspberry Pi 3 Model A+/ B+
WAŻNE: PROSIMY ZACHOWAĆ TE INFORMACJE NA PRZYSZŁOŚĆ.

Ostrzeżenia

- Wszelkie zewnętrzne źródła zasilania używane z Raspberry Pi powinny być zgodne z odpowiednimi przepisami i normami obowiązującymi w kraju przeznaczenia. Zasilacz powinien zapewniać napięcie 5V DC i minimalny prąd znamionowy 2,5A.

Instrukcje bezpiecznego użytkowania

- Ten produkt nie powinien być przetaktowany.
- Nie należy wystawiać tego produktu na działanie wody ani wilgoci, ani umieszczać go na powierzchni przewodzącej podczas pracy.
- Nie należy wystawiać tego produktu na działanie ciepła z jakiegokolwiek źródła; produkt zaprojektowano tak, aby działał niezawodnie w normalnej temperaturze pokojowej.
- Używać w dobrze wentylowanym otoczeniu i nie zakrywać podczas użytkowania.
- Podczas użytkowania należy umieścić produkt na stabilnej, płaskiej, nieprzewodzącej powierzchni i nie dopuścić do kontaktu z przedmiotami przewodzącymi prąd.
- Należy zachować ostrożność podczas obchodzenia się z produktem, aby uniknąć mechanicznego lub elektrycznego uszkodzenia płyty z obwodami drukowanymi i złączy.
- Nie należy przenosić produktu, gdy jest podłączony do zasilania. Trzymać wyłącznie za krawędzie, aby zminimalizować ryzyko uszkodzenia w wyniku wyładowań elektrostatycznych.
- Wszelkie urządzenia peryferyjne lub sprzęt używany z Raspberry Pi powinny być zgodne z odpowiednimi normami dla kraju użytkowania i być odpowiednio oznakowane, aby zapewnić spełnienie wymagań bezpieczeństwa i wymogów eksploatacyjnych.

Wszystkie certyfikaty zgodności i numery można znaleźć na stronie www.raspberrypi.org/compliance.

### Português do Brasil

Raspberry Pi 3 Modelo A+/ B +
IMPORTANTE: POR FAVOR, GUARDE ESTAS INFORMAÇÕES PARA REFERÊNCIA FUTURA.

Avisos

- Qualquer fonte de alimentação externa usada com o Raspberry Pi deve cumprir os regulamentos e normas aplicáveis no país de utilização. A fonte de alimentação deve fornecer 5V CC e uma corrente nominal mínima de 2,5A.

Instruções para o uso seguro

- Este produto não deve ser usado em overclock.
- Não exponha este produto à água ou à umidade, e não o coloque em uma superfície condutora durante a operação.
- Não exponha este produto ao calor de qualquer fonte; Ele é projetado para operação confiável à temperatura ambiente.
- Opere este produto em um ambiente bem ventilado e não o cubra durante o uso.
- Coloque este produto em uma superfície estável, plana e não condutora durante o uso, e não deixe que entre em contato com dispositivos que conduzem eletricidade.
- Tome cuidado ao manusear este produto para evitar danos mecânicos ou elétricos à placa de circuito impresso e aos conectores.
- Evite manusear este produto enquanto estiver ligado. Somente manuseie pelas bordas (laterais) para minimizar o risco de dano por descarga eletrostática.
- Qualquer periférico ou equipamento usado com o Raspberry Pi deve cumprir os padrões de fabricação e uso relevantes para o país e assim garantir que os requisitos de segurança e desempenho sejam atendidos.

Para todos os certificados conformidade e números, visite www.raspberrypi.org/compliance.

### Русский

Raspberry Pi 3 Модель A+/ B+
ВАЖНО: СОХРАНИТЕ ЭТУ ИНФОРМАЦИЮ ДЛЯ БУДУЩЕГО ИСПОЛЬЗОВАНИЯ.

Внимание!

- Любой внешний источник питания, используемый с Raspberry Pi, должен соответствовать соответствующим нормам и стандартам, применяемым в стране предполагаемого использования. Источник питания должен обеспечивать 5 В постоянного тока и минимальный номинальный ток 2,5 А.

Инструкции по безопасному использованию

- Этот продукт не должен использоваться вопреки нормативам произодителя.
- Не подвергайте этот продукт воздействию воды или влаги и не размещайте его на проводящей поверхности во время работы.
- Не подвергайте этот продукт воздействию тепла из любого источника; он предназначен для надежной работы при нормальной комнатной температуре.

- Эксплуатируйте этот продукт в хорошо проветриваемой среде и не накрывайте его во время использования.
- Поместите этот продукт на устойчивую, плоскую, непроводящую поверхность во время использования и не позволяйте ему контактировать с проводящими изделиями.
- Соблюдайте осторожность при обращении с этим продуктом, чтобы избежать механического или электрического повреждения печатной платы и разъемов.
- Избегайте обращения с этим продуктом во время его питания. Используйте только края, чтобы свести к минимуму риск повреждения электростатического разряда.
- Любое периферийное устройство или оборудование, используемое с Raspberry Pi, должно соответствовать соответствующим стандартам для страны использования и быть соответствующим образом маркировано для обеспечения соблюдения требований безопасности и производительности.

Для всех сертификатов соответствия и номеров, пожалуйста, посетите  www.raspberrypi.org/compliance .

**Español**

Raspberry Pi 3 Modelo A+/  B +
IMPORTANTE: POR FAVOR CONSERVE ESTA INFORMACIÓN PARA FUTURA REFERENCIA.

Advertencias

- Cualquier fuente de alimentación externa utilizada con la Raspberry Pi deberá cumplir con las correspondientes regulaciones y normas aplicables en el país de uso previsto. La fuente de alimentación debe proporcionar 5V DC y una corriente nominal mínima de 2.5A.

Instrucciones para un uso seguro

- Este producto no debe ser usado con una frecuencia de reloj superior a la nominal. (No se debe overclockear).
- No exponga este producto al agua o a la humedad, y no lo coloque sobre una superficie conductora mientras está en funcionamiento.
- No exponga este producto a ningún tipo de fuente de calor; está diseñado para un funcionamiento fiable a temperatura ambiente normal.
- Utilice este producto en un ambiente bien ventilado, y no lo cubra durante el uso.
- Coloque este producto sobre una superficie estable, plana y no conductora mientras esté en uso, y no permita que entre en contacto con elementos conductores.
- Tenga cuidado al manipular este producto para evitar daños mecánicos o eléctricos en la placa de circuito impreso y en los conectores.
- Evite manipular este producto mientras está encendido. Sujételo solo por los bordes para minimizar el riesgo de daños por descargas electrostáticas.
- Cualquier periférico o equipo utilizado con la Raspberry Pi debe cumplir con las normas aplicables en el país de uso y debe estar marcado en consecuencia para garantizar que se cumplen los requisitos de seguridad y rendimiento.

Para obtener todos los certificados de conformidad y sus números de registro, visite www.raspberrypi.org/compliance.

**Svenska**

Raspberry Pi 3 Modell A+/ B +
VIKTIGT: BEHÅLL DENNA INFORMATION FÖR FRAMTIDA REFERENS.

Varningar

- Alla externa strömförsörjningar som används med Raspberry Pi måste uppfylla alla tillämpliga regler och standarder i det land där de används. Strömförsörjningen måste tillhandahålla 5 VDC och ha en lägsta märkström på 2,5 A.

Instruktioner för säker användning

- Produkten bör inte överklockas.
- Utsätt inte produkten för vatten eller fukt, och placera den inte på en ledande yta medan den är i drift.
- Utsätt inte produkten för värme från någon värmekälla. Den är utformad för tillförlitlig drift vid normal rumstemperatur.
- Använd produkten i en väl ventilerad miljö, och täck inte över den vid användning.
- Placera produkten på en stabil, isolerad yta vid användning, och låt den inte komma i kontakt med ledande föremål.
- Var försiktig när du hanterar produkten för att undvika mekaniska eller elektriska skador på kretskortet och kontakterna.
- Undvik att hantera produkten med strömmen på. Håll den endast i kanterna för att undvika elektrostatiska urladdningar.
- Eventuell kringutrustning och utrustning som används med Raspberry Pi måste uppfylla relevanta standarder i det land där den används, och den bör märkas så att säkerhets- och prestandakraven uppfylls.

Besök www.raspberrypi.org/compliance, för alla certifikat och nummer om överensstämmelse.

**EU**
**Radio Equipment Directive (2014/53/EU)**
**Declaration of Conformity (DoC)**

We, Raspberry Pi (Trading) Limited, 30 Station Road, Cambridge, CB1 2JH, United Kingdom, Declare under our sole responsibility that the product: Raspberry Pi 3 Model B+ to which this declaration relates is in conformity with the essential requirements and other relevant requirements of the Radio Equipment Directive (2014/53/EU).

The product is in conformity with the following standards and/or other normative documents:

SAFETY (art 3.1.a): **IEC 60950-1: 2005 (2nd Edition) and EN 62311: 2008** EMC (art 3.1.b): **EN 301 489-1/ EN 301 489-17 Ver. 3.1.1 (assessed in conjunction with ITE standards EN 55032 and EN 55024 as Class B equipment)** SPECTRUM (art 3. 2): **EN 300 328 Ver 2.1.1, EN 301 893 V2.1.0**

In accordance with Article 10.8 of the Radio Equipment Directive: The device ‘Raspberry Pi 3 Model B+’  operates in compliance with harmonised standard EN 300 328 v2.1.1 and transceives within the frequency band 2,400 MHz to 2,483.5 MHz and, as per Clause 4.3.2.2 for wideband modulation type equipment, operates at a maximum e.i.r.p. of 20dBm. The device ‘Raspberry Pi 3 Model B+’ also operates in compliance with harmonised standard EN 301 893 V2.1.1 and transceives within the frequency bands 5150-5250MHz, 5250-5350MHz, and 5470-5725MHz and, as per Clause 4.2.3.2 for wideband modulation type equipment, operates at a maximum e.i.r.p. of 23dBm (5150-5350MHz) and 30dBm (5450-5725MHz).

In accordance with Article 10.10 of the Radio Equipment Directive, and as per below list of country codes, the operating bands 5150-5350MHz are strictly for indoor usage only.

| | | | | |
|---|---|---|---|---|
| BE | BG | ⚠ | CZ | DK |
| DE | EE | | IE | EL |
| ES | FR | HR | IT | CY |
| LV | LT | LU | HU | MT |
| NL | AT | PL | PT | RO |
| SI | SK | FI | SE | UK |

The Raspberry Pi complies with the relevant provisions of the RoHS Directive for the European Union.

**WEEE Directive Statement for the European Union**
This marking indicates that this product should not be disposed with other household wastes throughout the EU. To prevent possible harm to the environment or human health from uncontrolled waste disposal, recycle it responsibly to promote the sustainable reuse of material resources. To return your used device, please use the return and collection systems or contact the retailer where the product was purchased. They can take this product for environmental safe recycling. Note: A full online copy of this Declaration can be found at www.raspberrypi.org/compliance/

**China**
Raspberry Pi 3 Model B+ CMIIT ID: 2018AJ2147

**FCC**

Raspberry Pi 3 Model B+ FCC ID: 2ABCB-RPI3BP
Model A+ FCC ID:2ABCB-RPI3AP

This device complies with Part 15 of FCC Rules, Operation is Subject to following two conditions: (1) This device may not cause harmful interference, and  (2) This device must accept any interference received including interference that cause undesired operation. Caution: Any changes or modifications to the equipment not expressly approved by the party responsible for compliance could void user s authority to operate the equipment. This equipment has been tested and found to comply within the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: • Reorient or relocate the receiving antenna  • Increase the separation between the equipment and receiver • Connect the equipment into an outlet on a different circuit from that to which the receiver is connected • Consult the dealer or an experienced radio/TV technician for help. For product available in the USA/Canada market, only channel 1~11 can be operated and these channel assignments deal with only the 2.4GHz range  This device and its antenna(s) must not be co-located or operation in conjunction with any other antenna or transmitter except in accordance with FCC’s multi-transmitter procedures. This device is going to be operated in 5.15~5.25GHz frequency range, it is restricted in indoor environment only
IMPORTANT NOTE: FCC Radiation Exposure Statement; This appliance and its antenna must not be co-located or operation in conjunction with any other antenna or transmitter. A minimum separation distance of 20cm must be maintained between the antenna and the person for this appliance to satisfy the RF exposure requirements. This raspberry Pi product is certified for usage as a Single Module under the requirements of 15.212.  The integration of this module into end products is subject to the following requirements:
USER MANUAL OF THE END PRODUCT:
In the user’s manual of the end of product, the end user has to be informed:

- To keep at least 20cm separation with the antenna while this end product is installed and operated.
- That the FCC radio-frequency exposure guidelines for an uncontrolled environment can be satisfied.
- That any changes or modifications not expressly approved by the manufacturer could void the user’s authority to operate this equipment.
- That, if the size of the end product is smaller than 8x10cm, then additional FCC part15.19 statement is required to be available in the user’s manual; This device complies with Part 15 of FCC Rules, Operation is Subject to following two conditions: (1) This device may not cause harmful interference, and (2) This device must accept any interference received including interference that cause undesired operation. Caution: Any changes or modifications to the equipment not expressly approved by the party responsible for compliance could void user's authority to operate the equipment.

LABEL OF THE END PRODUCT:
The final end product must be labelled in a visible area with the following " Contains TX FCC ID: 2ABCB-RPI3A/BP". If the size of the end product is larger than 8x10cm, then the following FCC part 15.19 statement has to also be available on the label: This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

ISED

Raspberry Pi 3 Model B+ IC: 20953-RPI3P
Raspberry Pi 3 Model A+ IC:20953-RPI3A
This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :(1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

For product available in the USA/Canada market, only channel 1~11 can be operated. Selection of other channels is not possible.
Pour les produits disponibles aux États-Unis / Canada du marché, seul le canal 1 à 11 peuvent être exploités. Sélection d'autres canaux n'est pas possible.

This device and its antenna(s) must not be co-located with any other transmitters except in accordance with IC multi-transmitter product procedures. Referring to the multi-transmitter policy, multiple-transmitter(s) and module(s) can be operated simultaneously without reassessment permissive change. Cet appareil et son antenne (s) ne doit pas être co-localisés ou fonctionnement en association avec une autre antenne ou transmetteur. This radio transmitter has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.Le présent émetteur radio (IC: 20953-RPI3P) a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Dynamic Frequency Selection (DFS) for devices operating in the bands 5250- 5350 MHz, 5470-5600 MHz and 5650-5725 MHz. Sélection dynamique de fréquences (DFS) pour les dispositifs fonctionnant dans les bandes 5250-5350 MHz, 5470-5600 MHz et 5650-5725 MHz. The device for operation in the band 5150–5250 MHz is only for indoor use to reduce the potential for harmful interference to co-channel mobile satellite systems. les dispositifs fonctionnant dans la bande 5150-5250 MHz sont réservés uniquement pour une utilisation à l'intérieur afin de réduire les risques de brouillage préjudiciable aux systèmes de satellites mobiles utilisant les mêmes canaux. The maximum antenna gain permitted for devices in the bands 5250-5350 MHz and 5470-5725 MHz shall be such that the equipment still complies with the e.i.r.p. limit.le gain maximal d'antenne permis pour les dispositifs utilisant les bandes 5250-5350 MHz et 5470-5725 MHz doit se conformer à la limite de p.i.r.e.

The maximum antenna gain permitted for devices in the band 5725-5850 MHz shall be such that the equipment still complies with the e.i.r.p. limits specified for point-to-point and non-point-to-point operation as appropriate. le gain maximal d'antenne permis (pour les dispositifs utilisant la bande 5725-5850 MHz) doit se conformer à la limite de p.i.r.e. spécifiée pour l'exploitation point à point et non point à point, selon le cas. For indoor use only. Pour une utilisation en intérieur uniquement.

IMPORTANT NOTE:
IC Radiation Exposure Statement:
This equipment complies with IC RSS-102 radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body. Cet équipement est conforme aux limites d'exposition aux rayonnements IC établies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé avec un minimum de 20 cm de distance entre la source de rayonnement et votre corps.

IMPORTANT NOTE:
This module is intended for OEM integrator. The OEM integrator is responsible for the compliance to all the rules that apply to the product into which this certified RF module is integrated. Additional testing and certification may be necessary when multiple modules are used. Any changes or modifications not expressly approved by the manufacturer could void the user's authority to operate this equipment.

USERS MANUAL OF THE END PRODUCT:
In the user's manual of the end product, the end user has to be informed to keep at least 20cm separation with the antenna while this end product is installed and operated. The end user has to be informed that the IC radio-frequency exposure guidelines for an uncontrolled environment can be satisfied

The end user has to also be informed that any changes or modifications not expressly approved by the manufacturer could void the user's authority to operate this equipment. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

LABEL OF THE END PRODUCT:
The final end product must be labelled in a visible area with the following " Contains IC: 20953-RPI3P / 20953-RPI3A ".The Host Model Number (HMN) must be indicated at any location on the exterior of the end product or product packaging or product literature which shall be available with the end product or online.

**Australia and New Zealand**

Class B Emissions Compliance Statement
Warning: This is a Class B product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

**Brazil**

Este equipamento não tem direito à proteção contra interferência prejudicial e não pode causar interferência em sistemas devidamente autorizados. Este produto está homologado pela Anatel, de acordo com os procedimentos regulamentados pela Resolução nº242/2000 e atende aos requisitos técnicos aplicados, incluindo os limites de exposição da Taxa de Absorção Específica referente a campos elétricos, magnéticos e eletromagnéticos de radiofrequência, de acordo com as Resoluções nº 303/2002 e 533/2009.

**Mexico**

La operación de este equipo está sujeta a las siguientes dos condiciones: (1) es posible que este equipo o dispositivo no cause interferencia perjudicial y (2) este equipo o dispositivo debe aceptar cualquier interferencia, incluyendo la que pueda causar su operación no deseada. Este equipo esta diseñado para operar con las antenas que enseguida se enlistan y para una ganacia máxima de 3.5 dBi. El uso de Antenas con este equipo no incluidas en esta lista o que tengan una ganancia mayor a 3.5 dBi quedan PROHIBIDAS.

**Taiwan**

| 根據NCC低功率電波輻射性電機管理辦法 規定: | |
|---|---|
| 第十二條 | 經型式認證合格之低功率射頻電機，非經許可，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能 |
| 第十四條 | 低功率射頻電機之使用不得影響飛航安全及干擾合法通信；<br>經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。<br>前項合法通信，指依電信法規定作業之無線電通信。<br>低功率射頻電機須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。 |

應避免影響附近雷達系統之操作。
高增益指向性天線只得應用於固定式點對點系統。

R 007-AG0046

Raspberry Pi 3 Model B+
**CHILE SUBTEL 5219/DO**

Raspberry Pi 3 Model B+
**CHILE SUBTEL 5219/DO**

CCAO18LP0240T9

# SUBSCRIBE TODAY FROM ONLY £5

## SAVE UP TO 35%

WIN! 1 OF 10 RASPBERRY PI 4 CASES
The MagPi
The official Raspberry Pi magazine
Issue 85 | September 2019 | magpi.cc
RASPBERRY PI 4
YOUR NEXT DESKTOP PC
Can you do everything with a $55 computer?
7-DAY TEST DIARY
SUPER-FAST NAS DRIVE
Build a gigabit file server with Raspberry Pi 4
BACK TO SCHOOL
Use Raspberry Pi to learn & teach coding
PLUS! LEARN WEB DEVELOPMENT
MAKE AMAZING 8-BIT SPRITES
ROCKYBORG ROBOT REVIEWED!
40 PAGES OF TUTORIALS

### Subscriber Benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 35% compared to stores

### Rolling Monthly Subscription

- Low monthly cost (from £5)
- Cancel at any time
- Free delivery to your door
- Available worldwide

### Subscribe for 12 Months

£55 (UK) £90 (USA & Rest of World)
£80 (EU)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspbery Pi Zero Kit with Rolling Monthly Subscription)

☎ Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

# FREE Raspberry Pi Zero W Starter Kit

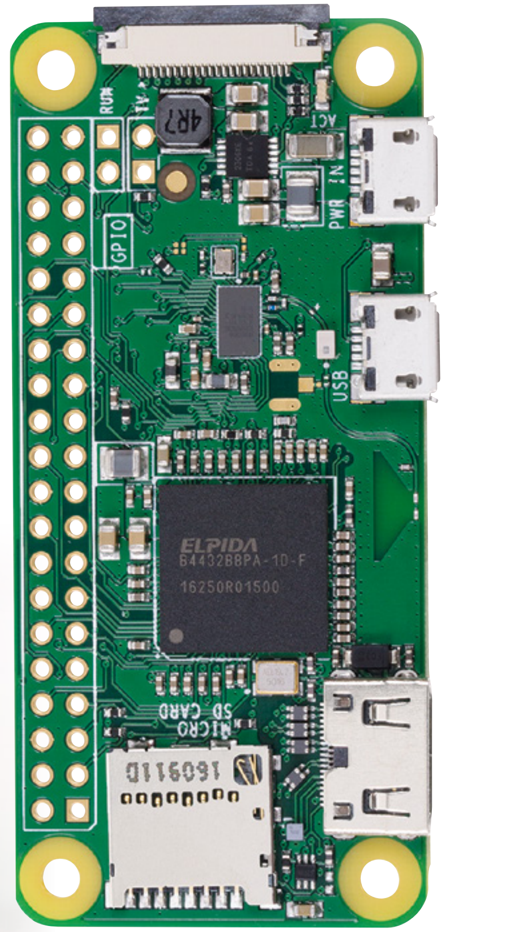
WITH YOUR SUBSCRIPTION

**Subscribe in print for 12 months today and you'll receive:**

- Raspberry Pi Zero W
- Raspberry Pi Zero W case with three covers
- USB and HDMI converter cables
- Camera Module connector

Offer subject to change or withdrawal at any time

WORTH £20



Buy now: magpi.cc/subscribe

**SUBSCRIBE on** app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

# GET STARTED WITH RASPBERRY PI

The world's favourite single-board computer, Raspberry Pi is packed with potential: it can be a digital camera, a programmable games machine, or even a sensor for the International Space Station. While creating a Raspberry Pi project, you'll learn computing, coding, and electronics. It's fun, and creative, and it's changing the world.

## MASTER YOUR RASPBERRY PI IN EASY STEPS

- Set up your Raspberry Pi 3A+ for the first time
- Discover amazing software built for creative learning
- Learn how to program in Scratch and Python
- Control electronics: buttons, lights, and sensors

Raspberry Pi PRESS

magpi.cc