THE OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

The official Raspberry Pi magazine

Issue 56 April 2017

raspberrypi.org/magpi

INCREDIBLE RASPBERRY PI PROJECTS

Get started with digital making today

COMMAND LINE SUPPORT

The terminal will help you, if you know how to ask

POWER UP

The safest ways to power your Pi

HOT STUFF!

Build an LED thermometer

AMAZING ART WITH PYTHON

Use a DJ controller to mix a masterpiece

Also inside:

- > CELEBRATING 5 YEARS OF RASPBERRY PI
- > SIMULATE MONOPOLY TO BUY WISELY
- > BUILD A SIMPLE REACTIONS GAME
- > LEARN THE LANGUAGE OF LIGHT

EVERY OPERATING SYSTEM TESTED!

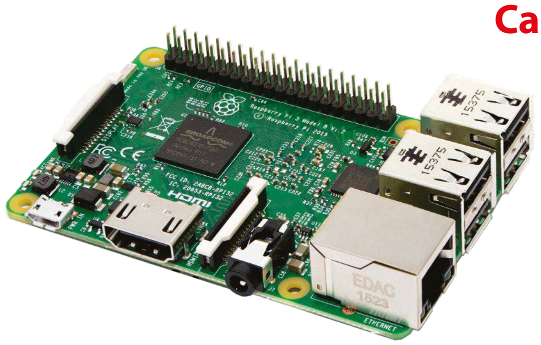
Find the right Raspberry Pi distro for the right task with our 10-page super-test

Issue 56 • Apr 2017 • £5.99

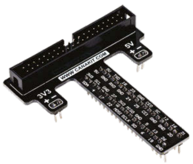
04

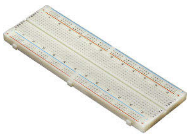
9 772051 998001

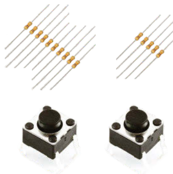
THE ONLY MONTHLY MAGAZINE WRITTEN BY AND FOR THE PI COMMUNITY



CanaKit Raspberry Pi Quick-Start Guide







WELCOME TO THE OFFICIAL MAGAZINE

Digital making is fun, but it's also incredibly important. Whether you're building something from scratch, or taking an existing gadget and stripping it down to add your own touch of magic, it's vital that we think about how things actually work.

Whether we like it or not, we live in a world where small impenetrable black monoliths deliver our entertainment and information. We don't have to think about how they work and their designers don't want us to.

Digital making with the Raspberry Pi is the perfect antidote, and we've put together a selection of easy and intermediate projects for you to try in our cover feature starting on page 14. Elsewhere this issue we're also building some great devices, including an LED thermometer, a gadget that turns Python code into art, and a simple reactions game, among other things.

Happy making!

Russell Barnes
Managing Editor

THE OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
INCREDIBLE RASPBERRY PI PROJECTS

FREE PI ZERO W WITH A SUBSCRIPTION! PAGE 26

SUBSCRIBE & SAVE 25%
SEE PAGE 26 FOR DETAILS

THIS MONTH:

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

The MagPi

Available on the App Store

GET IT ON Google Play

EDITORIAL
Managing Editor: **Russell Barnes**
russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
News Editor: **Lucy Hattersley**
Sub Editors: **Rachel Churcher and Phil King**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Daiva Bumelyte, and Mike Kay**
Illustrator: **Sam Alder**

PUBLISHING
For advertising & licensing:
russell@raspberrypi.org +44 (0)7904 766523
Publisher: **Liz Upton**
CEO: **Eben Upton**

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

SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337
Bournemouth
BH1 9EH | **+44 (0)1202 586 848**

CONTRIBUTORS
William Bell, Henry Budden, Mike Cook, Phil King, Simon Long, Spencer Organ, Les Pounder, Matt Richardson, Richard Smedley, and Patricio Vivo

recycle
When you have finished with this magazine, please recycle it.

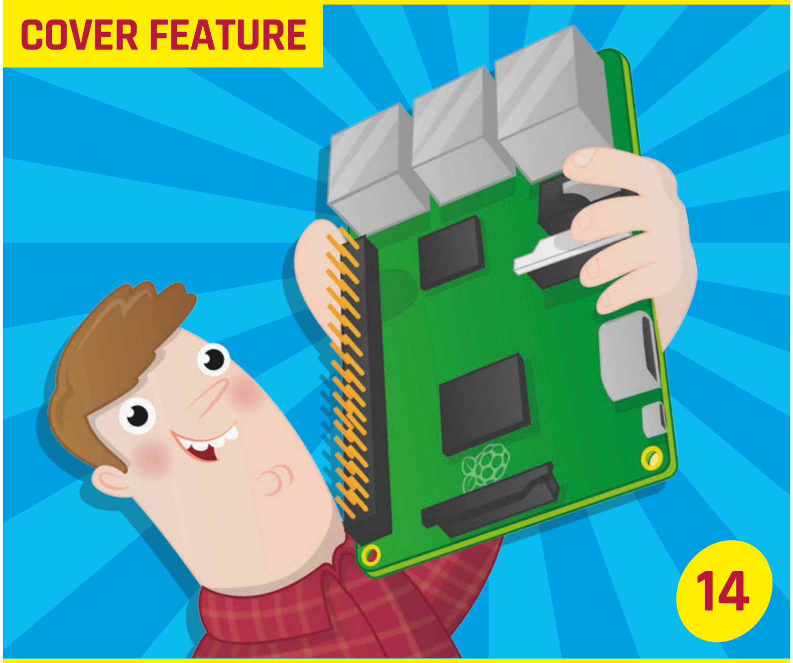
This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Contents

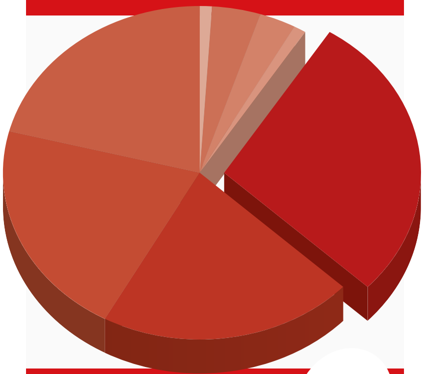
Issue 56 April 2017 raspberrypi.org/magpi

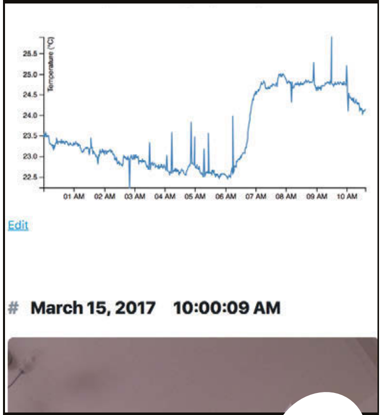
TUTORIALS

COVER FEATURE



IN THE NEWS

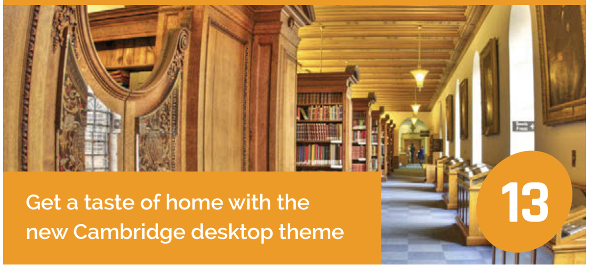




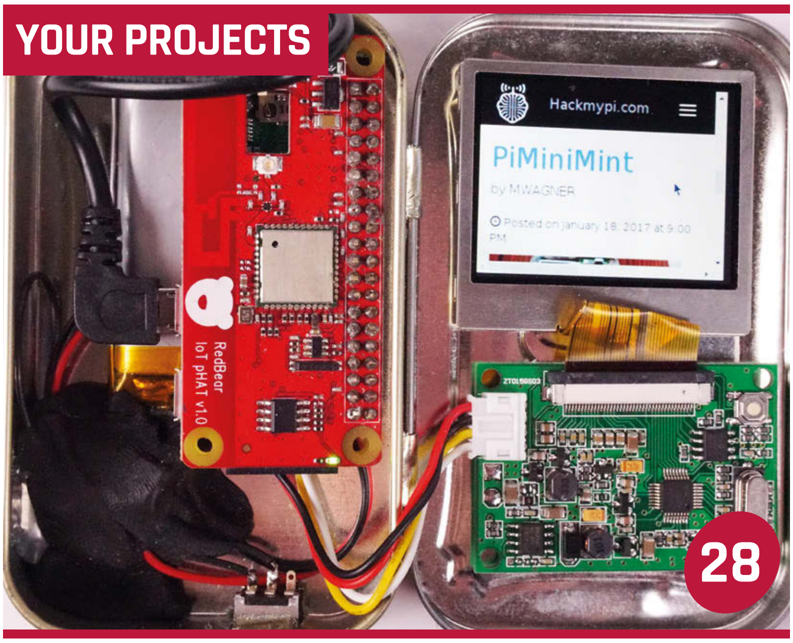
Edit

March 15, 2017 10:00:09 AM

THE BIG FEATURE



YOUR PROJECTS



REGULARS

COMMUNITY

REVIEWS

FIFTH BIRTHDAY PARTY

Raspberry Pi celebrates its first five years

Philip Colligan welcoming people to the Big Birthday Party 2017

Nearly 1,500 members of the Raspberry Pi community attended the Fifth Birthday Party for the organisation.

Held at Cambridge Junction arts centre on 4/5 March, the Big Birthday Weekend 2017 saw hundreds of Raspberry Pi community members gather together to share projects and cake.

The Twitter hashtag #piparty even became a Twitter Moment, gaining global coverage.

The party featured workshops and talks from the Raspberry Pi Foundation, Code Club, and active members of the community.

The MagPi team had a stand at the party. We showed off recent projects including the PiGRRL2 from the cover of issue 55.

Sam Aaron and Ben Smith's Sonic Pi workshop was especially impressive. They demonstrated live coding with a mind-blowing live performance (watch online: **magpi.cc/2m6mlQm**).

"We describe our mission as putting the power of digital making in the hands of people all over the world," said Philip Colligan, CEO of Raspberry Pi Foundation.

In the main hall, dozens of digital makers showed off their latest builds. Party-goers played Spanner Spencer's four-player Gauntlet, Steve Upton's Pi Bash, and NuStem's Technology Wishing Well.

Below Digital makers shared inspiring projects with the Raspberry Pi community

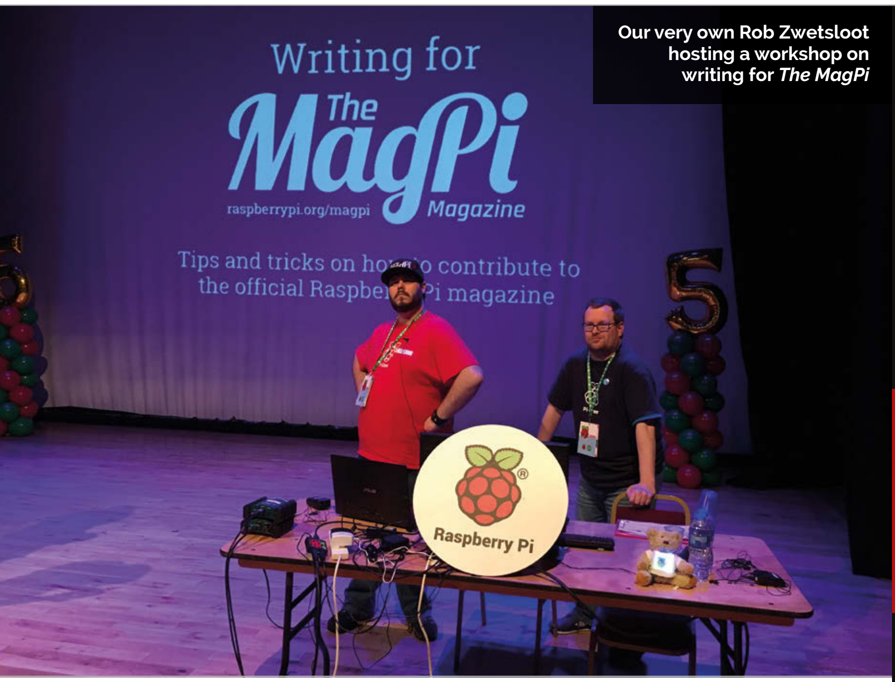
A special year

"It's been another interesting, weird, and surprising year for us at Raspberry Pi," said Eben Upton, founder and CEO of Raspberry Pi Trading.

The Raspberry Pi is an incredibly important computer because it exists in a world of computers that are expensive or inaccessible. "You can't let your child play with the family computer any more than you let your child take the family car apart," explained Eben. The Raspberry Pi is a fantastic platform for digital making and experimentation. It's fun and low cost.

One interesting development is that Raspberry Pi has become an advocate for British manufacturing. "I didn't think that is what we'd be when we built our first units five years ago," said Eben

"We are not building units in the UK because we are patriotic," he explained. "We are building


Writing for The MagPi Magazine
raspberrypi.org/magpi
Tips and tricks on how to contribute to the official Raspberry Pi magazine
Raspberry Pi

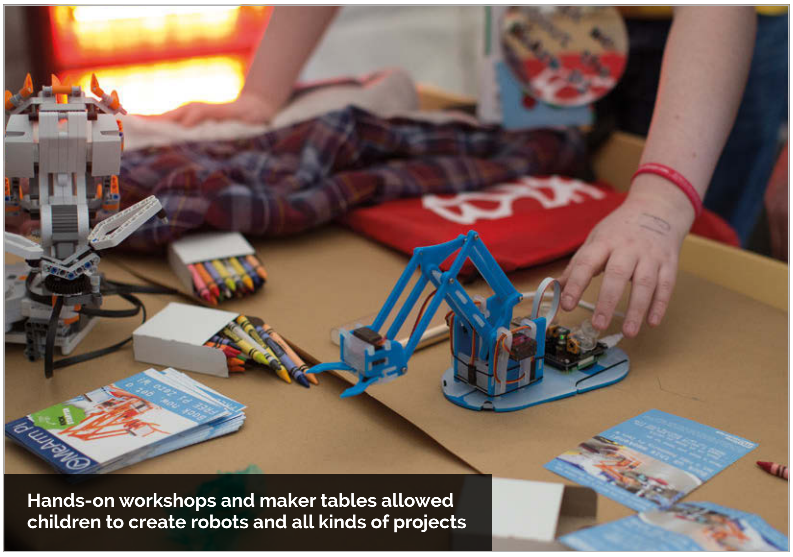
Our very own Rob Zwetsloot hosting a workshop on writing for *The MagPi*

them in the UK because we are cheapskates who want to build cheap but high-quality products." It turns out that the cheapest place to build products is here in the United Kingdom.

> We are as much a movement as we are an organisation

"We have to remember," said Eben, "that we are as much a movement as we are an organisation. For every one person inside the organisation there are 10 or 100 more people out there in the world using the things we make, using the resources, and the infrastructure we create, to make the difference."

Fran Scott ended the party with a Live Coding Explosions show, using a Raspberry Pi to blow up hydrogen balloons on stage (**magpi.cc/2m6ArBn**). It was "absolutely wonderful to join Raspberry Pi and Code Club to celebrate their fifth birthday," remarked Fran. "Thanks so much for the invite."



Hands-on workshops and maker tables allowed children to create robots and all kinds of projects

PI PARTY IN NUMBERS

1,800 GUESTS & VOLUNTEERS

31 TALKS

22 EXHIBITORS

6 CONTROLLED EXPLOSIONS BY FRAN SCOTT

1,000 CUPCAKES

SALES SOAR ABOVE COMMODORE 64

Raspberry Pi becomes world's third-best selling computer platform

Recent Raspberry Pi sales figures revealed that the humble board is now the world's third-best selling general-purpose computer platform, edging past the sales of the Commodore 64.

"The Commodore 64 had, until recently, the distinction of being the third most popular general-purpose computing platform," Eben Upton, CEO of Raspberry Pi Trading told a crowd at the fifth birthday party. "That's what I'm here to celebrate," he said, "we are now the third most popular general-purpose computing platform after the Mac and PC."

The Commodore 64 was a hugely popular 8-bit computer platform, first built in 1982

> "Eben also revealed that the Pi Zero W sold about 100,000 in its first four days"

There are approximately 1.25 billion PCs sold. Apple's CEO Tim Cook has claimed sales of 80 million Intel Mac computers using the OS X platform.

Outselling the Commodore 64 cements Raspberry Pi in the annals of history. It's not bad considering that the original plan was to produce between 10,000 and 20,000 boards.

"This is really cool," said Eben. "And we did it together, and it's kind of wonderful."

Sales rise

Eben also revealed that the Pi Zero W sold about 100,000 units in its first four days, so it looks as if the Raspberry Pi computer platforms will keep selling for many years to come.

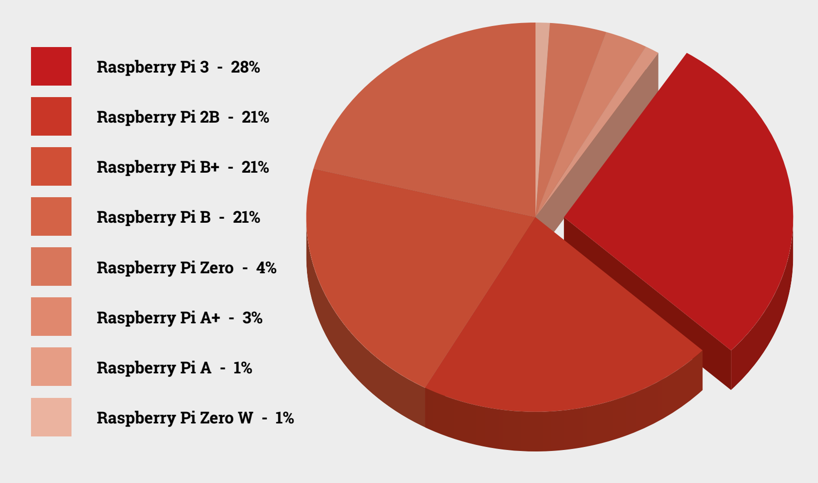
Our analysis shows that the Raspberry Pi 3 has accounted for almost a third of all Raspberry Pi boards sold.

The Pi 3 sits next to its immediate predecessor, the Raspberry Pi 2 (which has the same board shape but a slightly slower CPU). These two boards account for over half of all Raspberry Pi boards sold.

The rest of the sales are shared between older models. The original Model A accounts for just two percent of sales.

There is some debate surrounding the total sales of the Commodore 64. The 12.5 million figure comes from an analysis of serial numbers by Michael Steil (**magpi.cc/2mI2Egg**).

Michael's analysis of Commodore 64 platform sales is based on 16 different board revisions, used in a range of '64' computers such as the PET 64, Educator 64, and C64GS games console. However, it does not include the Portable SX-64 and Commodore 128, as these had a different architecture.



Raspberry Pi 3
Model B
element14
WD LABS
375GB

NEW PIMORONI KITS

Four new projects for Pi Zero W

Pimoroni has released four new project kits to sit alongside the new Pi Zero W board. Each kit contains a Pi Zero W board and GPIO pin header, requiring some basic soldering.

"As soon as we heard about the Pi Zero W, we knew that its features would make IoT projects much more approachable," says Paul Beech, Pimoroni co-founder. "These kits are made to appeal to everyone, to inspire people to try IoT and crafting."

The Starter Kit (£32) contains a Raspberry Pi, microSD card, Pibow case, and Blinkt! LED strip.

> "These kits are made to appeal to everyone"

A Mood Light Kit (£30) comes with a Pimoroni Unicorn pHAT (a grid of 32 programmable RGB NeoPixel LEDs) and a mood light stand.

The Scroll Bot Kit (£35) comes with a Pimoroni Scroll pHAT HD and robot stand. Finally, the Pirate Radio Kit (£40) comes with a pHAT Beat DAC, 5W speaker, and radio stand.

More information can be found on the Pimoroni website (**magpi.cc/2mbbAMK**).

The Pirate Radio Kit enables you to build your own internet radio

Light up your room by programming the multicoloured LEDs

RX300 THIN CLIENT

Access Windows and cloud computing from a Raspberry Pi device

Desktop virtualisation firm NComputing has used the Raspberry Pi 3 to build a thin client computer. Thin clients are lightweight computers that access a remote server, typically a cloud-based computer or virtualisation environment. The server does all the heavy computational lifting, while the thin client provides the interface.

The RX300 supports virtual desktops from nine popular Windows operating systems, including Windows 10 and Windows Server 2016.

"Looking at the IT industry, Raspberry Pi has established itself as a robust and reliable device, one increasingly becoming viable within an enterprise setting," comments Christos Golias, head of sales at Tarox AG, Germany. "The virtualisation environment provided by NComputing's vSpace Pro 10 software coupled with the affordability of an RX300 offers an attractive IT deployment solution for businesses."

Young Song, NComputing CEO, explains: "Our vision is to make easy-to-use, powerful, and centrally managed computing solutions even more affordable. The Raspberry Pi single-board computer is renowned for its innovation, its portable size, versatility, and economic viability, making it the perfect thin client. Matched with our vSpace Pro software, we know it will deliver what IT admins need, and what end users want."

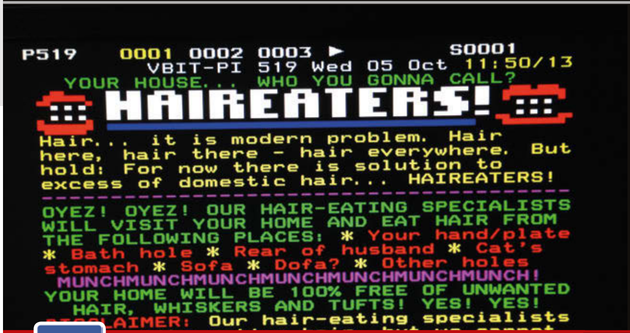
You can boot the RX300 into two modes: Thin Client Mode or Raspbian, enabling users to access the Linux OS.

The RX300 costs $99 excluding taxes, and includes a one-year subscription to vSpace Pro 10.

The RX300 is a thin client computer that provides cloud access to Windows on a Raspberry Pi board

NOW TRENDING

The stories we shared that flew around the world

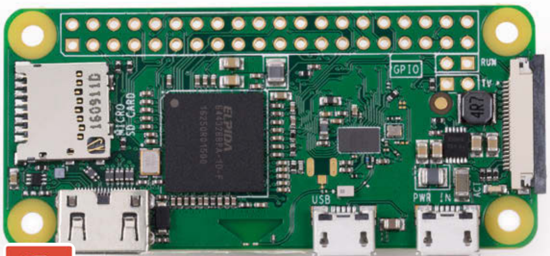


```
P519    0001 0002 0003 ▶          S0001
          VBIT-PI 519 Wed 05 Oct 11:50/13
   YOUR HOUSE... WHO YOU GONNA CALL?
      HAIREATERS!
Hair... it is modern problem. Hair
here, hair there - hair everywhere. But
hold: For now there is solution to
excess of domestic hair... HAIREATERS!
----------------------------------------
OYEZ! OYEZ! OUR HAIR-EATING SPECIALISTS
WILL VISIT YOUR HOME AND EAT HAIR FROM
THE FOLLOWING PLACES: * Your hand/plate
* Bath hole * Rear of husband * Cat's
stomach * Sofa * Dofa? * Other holes
 MUNCHMUNCHMUNCHMUNCHMUNCHMUNCHMUNCH!
YOUR HOME WILL BE 100% FREE OF UNWANTED
  HAIR, WHISKERS AND TUFTS! YES! YES!
DISCLAIMER: Our hair-eating specialists
```

PRESS THE TEXT BUTTON ON YOUR RASPBERRY PI

magpi.cc/2ign2q3

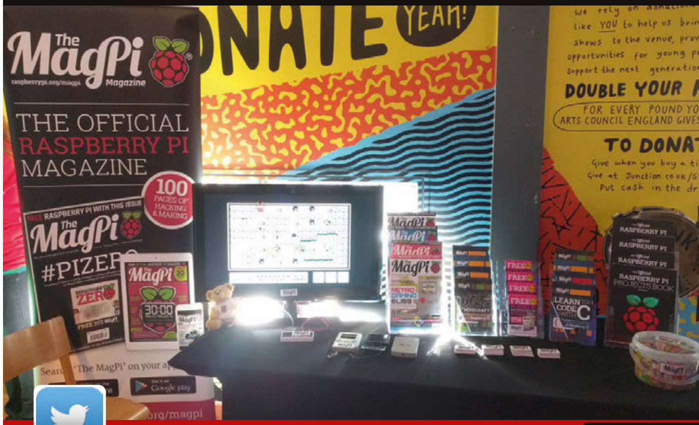
There was lots of love for our story on bringing Teletext back with the help of the Raspberry Pi.



WIRELESS WONDER

magpi.cc/2lylJhX

How the Raspberry Pi team managed to squeeze wireless networking and Bluetooth onto the tiny Pi Zero W board.



PIPARTY

magpi.cc/2mmx8mn

Miss the #PiParty? We've got a little gallery of photos from the day!

BLOG IN A BOX

Creators of WordPress.com build Raspberry Pi platform

Blog In A Box is a new tool for Raspberry Pi from Automattic, the company that developed **WordPress.com**.

The software turns your Raspberry Pi into a blogging platform that serves up a website, similar to WordPress.

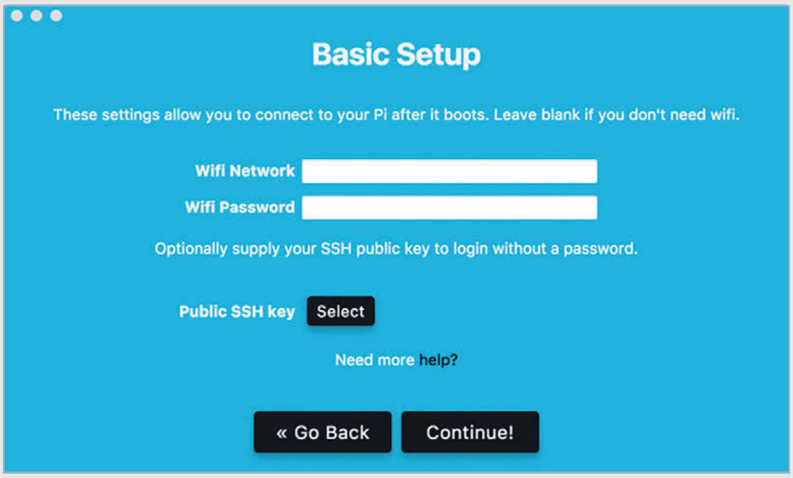
However, Blog In A Box takes full advantage of the Raspberry Pi hardware. You can automatically capture photographs from the Pi Camera Module or access data from the Sense HAT.

"We wanted to do something with the Raspberry Pi," says team member John Godley.

"We decided we'd really like an easy single package that would put a fully working WordPress system onto a Raspberry Pi in a way that will be useful for people to experiment and learn with," reveals John.

"We like to tinker ourselves," he adds.

Blog In A Box enables makers to "connect to Raspberry Pi sensors and hardware, and have it all viewable in a smart-looking way," according to John.



Basic Setup

These settings allow you to connect to your Pi after it boots. Leave blank if you don't need wifi.

Wifi Network

Wifi Password

Optionally supply your SSH public key to login without a password.

Public SSH key Select

Need more help?

« Go Back Continue!

Above A custom installer is used to burn a bootable SD Card image pre-loaded with your network details

The installer tool is available for Mac OS X, Windows, and Linux. Using it, you configure the settings for your Raspberry Pi (WiFi details, admin password).

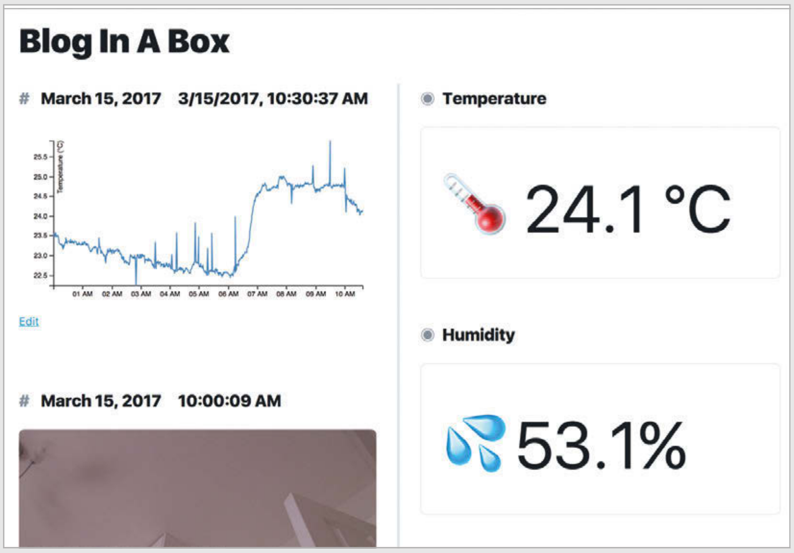
> Blog In A Box takes full advantage of the Raspberry Pi hardware

It then burns an image file direct to a microSD card. "Boot this card and you have a working WordPress system with MySQL, PHP, and Nginx all installed and configured," explains John. "When it's finished booting, WordPress is immediately available at **bloginabox.local**."

The WordPress plugin and tools give you access to anything you connect to your Raspberry Pi. "We've added built-in support for the Camera Module and a Sense HAT, both of which can be controlled from WordPress and set to capture data and post it automatically."

John tells us, "Everything is open source and we're hoping people can use it as a good base for all kinds of purposes."

Download the Blog In A Box installer tool from **inabox.blog**.



Blog In A Box

March 15, 2017 3/15/2017, 10:30:37 AM

Edit

March 15, 2017 10:00:09 AM

Temperature

24.1 °C

Humidity

53.1%

Left Data captured from a Raspberry Pi Sense HAT can be automatically posted to a blog

CAMBRIDGE THEME FOR PIXEL

New desktop images for Raspberry Pi produced to honour Cambridge roots

The Raspberry Pi Foundation has released a selection of new wallpapers and a screensaver for the Raspbian with PIXEL operating system.

The optional new look for Raspbian is based on the city of Cambridge, UK.

"The connection between Raspberry Pi and Cambridge runs a lot deeper than mere geography," says Simon Long, UX engineer.

He explains, "Raspberry Pi was founded with the aim of increasing the number of applicants to study computer science at the University of Cambridge. The processor core which powers the Raspberry Pi was developed in the city by ARM, the hugely successful microprocessor company which itself grew out of Acorn, one of the original pioneers of the 1980s home computer revolution, and another Cambridge success story."

A few months ago, Eben Upton and Simon were looking at the beautiful city flyover videos that Apple offers as screensavers on the Apple TV. "We thought that it would be great if we could do something similar for Raspbian, with Cambridge as the subject," recalls Simon. "So we enlisted the help of Cambridge Filmworks (**magpi.cc/2mHzz1D**)."

Cambridge Filmworks specialises in filming with drones, and they put together a video showing the best of Cambridge's architecture, recorded from the air.

The new screensaver is gorgeous. Once installed, you access it via Menu > Preferences > Screensaver.

To get the new theme, first update the Raspbian package list in Terminal:

```
sudo apt-get update
```

To install the wallpapers:

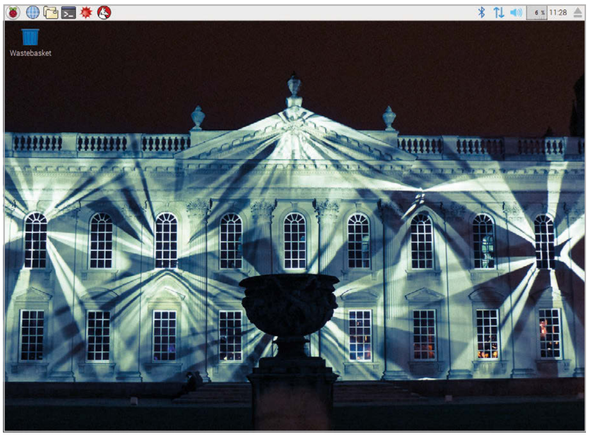
```
sudo apt-get install cantab-wallpaper
```

To install the screensaver:

```
sudo apt-get install cantab-screensaver
```

To install both:

```
sudo apt-get install cantab-theme
```



Left The Senate House of the University of Cambridge, lit up at the e-Luminate Festival 2017

DEXTER
INDUSTRIES
GoPiGo
20+ new projects for your Pi robot car.
BrickPi3
LEGO MINDSTORMS + RASPBERRY PI
www.dexterindustries.com

INCREDIBLE RASPBERRY PI PROJECTS

Get started with digital making with these incredible projects

Digital making is really important to the Raspberry Pi community. It's not enough for us to just buy a Raspberry Pi and just set it up. We have to make stuff too.

Here at *The MagPi*, we want to help people make things. Making is fun. It's about getting your creative juices flowing.

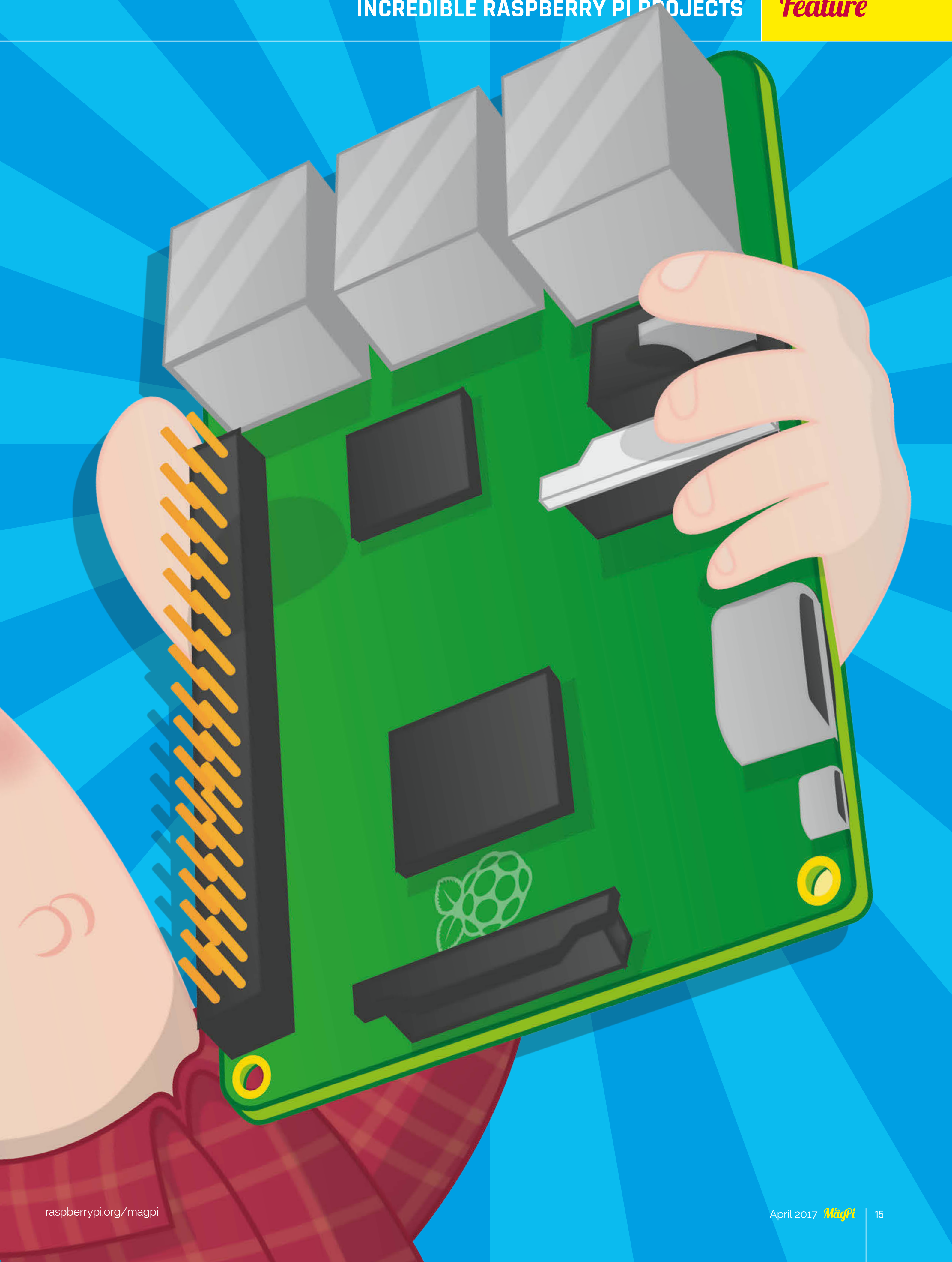
In the following pages, we present a stack of ideas for things to research and make with your Raspberry Pi.

These are practical projects that anybody can make. Some are simple kits like the LEGO NES case; others are *Blue Peter*-style hacks like the Pi-powered binoculars.

They're designed to be challenging, but definitely doable.

More advanced makers will want to look at Sixlab's YouTube Drone. This combines drone technology with a Pi Camera and 4G connectivity. It's the cutting edge of maker-tech.

There's little more important than being a digital maker in the modern world. The world is full of the 'new shiny': toys and trinkets that people pay for but never truly understand. It's way more important to make your own things. They may not always be as shiny, but they're worth so much more.



Beginner

BUILD AN EASY ROBOT

Get a CamJam robotics kit and make a wheeled rover in an afternoon

MIKE AND TIM

Michael Horne and Tim Richardson run the Cambridge Raspberry Jam and are active members of the Raspberry Pi community.
camjam.me

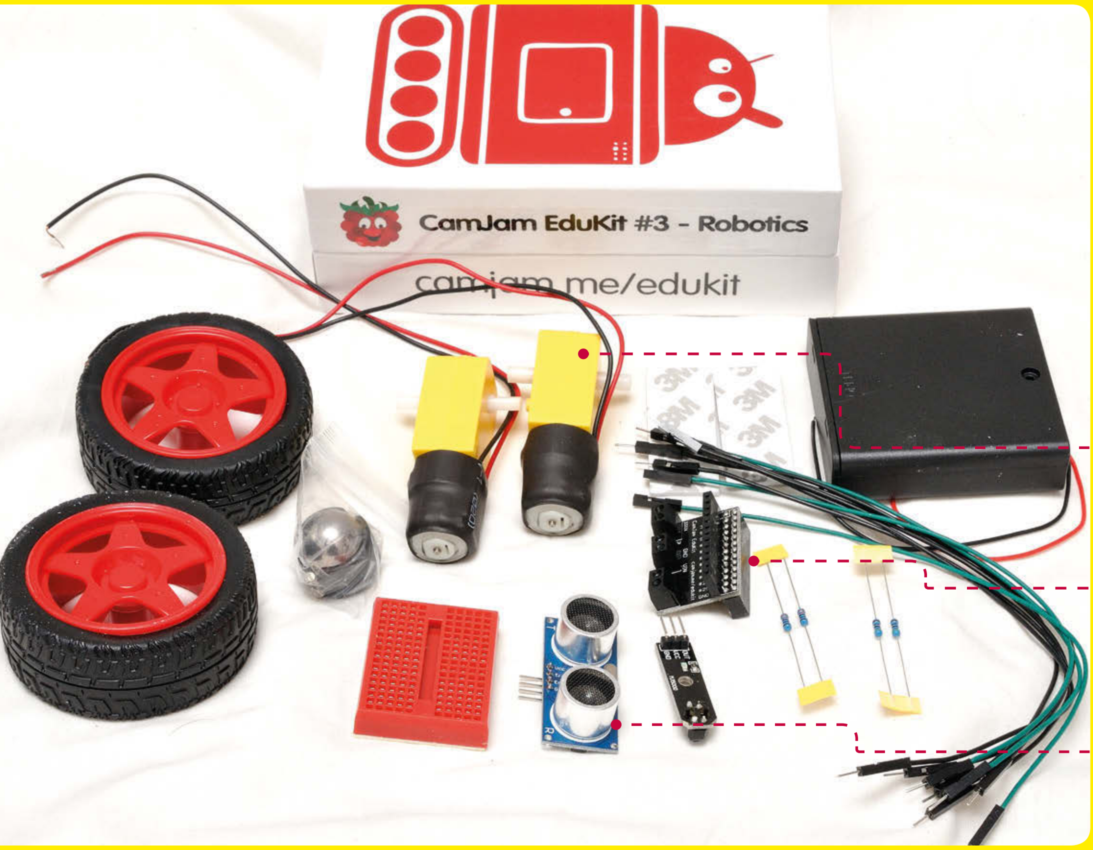
Building a robot is a dream for many new Raspberry Pi owners, and it's way easier to get started than you think.

One of our favourites is the EduKit 3 Robotics by Cambridge Raspberry Jam. The kit is designed by Michael Horne and Tim Richardson, the guys who run CamJam and Pi Wars.

Inside the box, you get a custom motor controller made by 4Tronix (**4tronix.co.uk**). This sits between the two DC motors and a battery compartment, which takes four AA batteries – make sure you buy high-quality batteries, as cheap ones often don't have enough juice to move the wheels.

The kit comes with two sensors: an SR-04 distance sensor and a line-following sensor. The only thing you don't get is a chassis. This is part of its charm, though, as you can build a robot out of any box capable of containing the Raspberry Pi. It's even possible to use the box that all the components come in.

What makes the CamJam EduKit 3 better than other robotics projects is the quality of the instructional material. There are ten different worksheets, covering everything from building the robot to driving the motors and setting up the sensors.

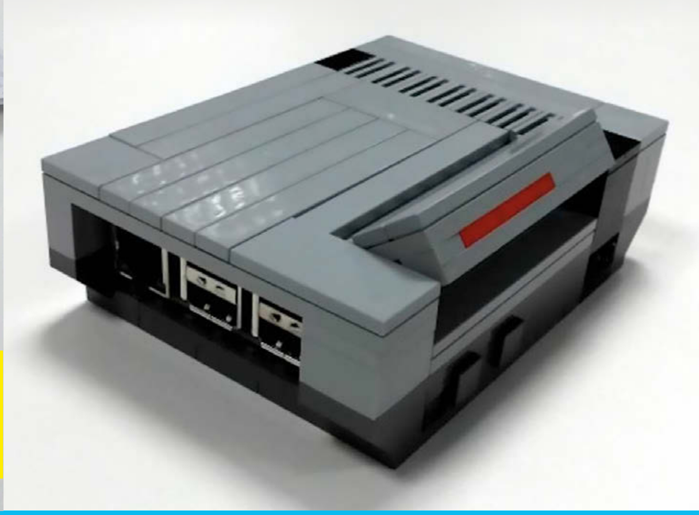


These two yellow motors power the red wheels and are used to move the robot around

A custom motor control unit is included. This enables you to control the robot using Python code

Two sensors are included with the kit. A distance sensor lets the robot stop before hitting a wall, and a line sensor enables the robot to follow a line drawn underneath it

The NesPi kit contains 111 pieces of LEGO and a 30-page manual. Like all LEGO kits, it's a lot of fun to put together as you build the case



Putting the parts together results in this Raspberry Pi case. It looks like a classic NES console, but with access to the Raspberry Pi ports. The lid (used for cartridges in the original NES) provides access to the GPIO pins

BUILD A NES FROM LEGO

Make a retro gaming console from toy bricks

The Danish company's plastic bricks are the toys that many makers love the most. LEGO has provided a sturdy start in life for many a maker.

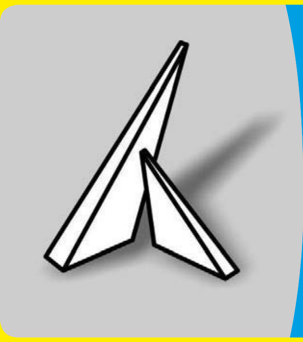
The idea of building a case for the Raspberry Pi from LEGO is by no means unique. There are many, many designs on the internet for Raspberry Pi LEGO cases.

But we like this NesPi design the best. Made from 111 pieces of regular LEGO, it is a tender homage to the 1985 Nintendo Entertainment System.

> A tender homage to the 1985 Nintendo Entertainment System

The result is a design that looks close to a NES, but has openings on the side enabling access to all the ports on your Raspberry Pi. This makes it possible to connect USB peripherals and the HDMI cable.

You can pick up a NesPi for just under €35 (including VAT) from the Spanish retailer, RaspiPC (**magpi.cc/2mj6QAJ**).The website is in Spanish, but it's easy enough to navigate (leave the telephone field blank, as it may struggle with international phone numbers).



BERNARDO MARTINEZ

Bernardo is a digital maker who runs RasPi PC, a Spanish retailer specialising in Raspberry Pi equipment. He designed the NesPi kit.
magpi.cc/2mj9fLJ

Alternatively, if you already have the required LEGO pieces, it's possible to create a similar-looking NES case from scratch. One maker, André Rinas, used the LEGO Digital Designer (**ldd.lego.com**) to create a similar case. You can find the instructions on his website (**magpi.cc/2mjhM1b**).

Intermediate

CREATE A WALL DISPLAY

Turn an old monitor or picture frame into a connected wall display with DAKboard

DAN KING

Dan is a software developer and digital designer from Syracuse, New York.

magpi.cc/2mJz9e4

YOU'LL NEED

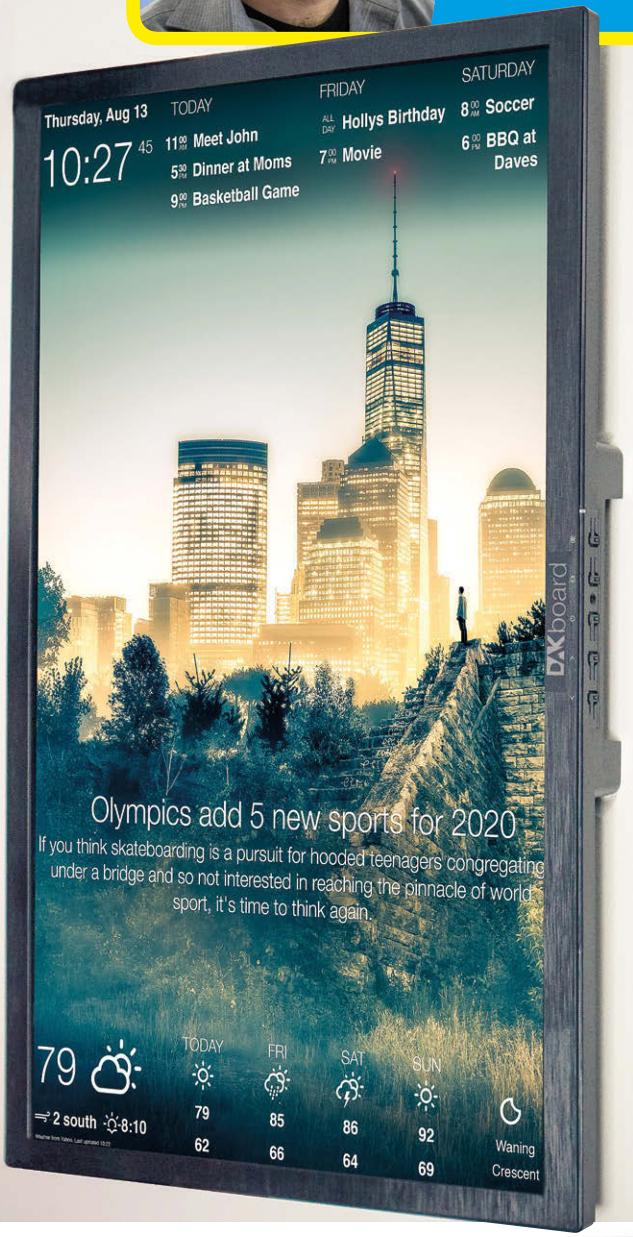
- Raspberry Pi computer (plus WiFi dongle, if using the Pi version 1 or 2)
- 8GB+ SD/micro SD card
- Micro USB charger (for Raspberry Pi)
- Power extension cord
- Photo frame wire

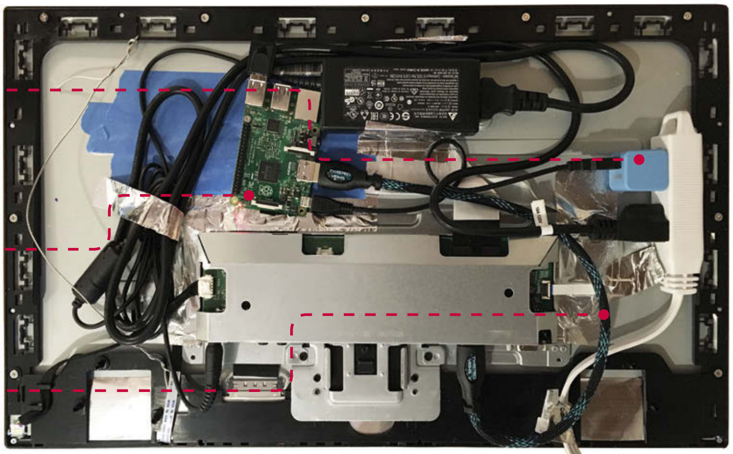
Hooking up a Raspberry Pi to a monitor is one of the easiest things to do. In fact, that's pretty much what you do when you first set up a Raspberry Pi.

That's one of the things that makes DAKboard such an enticing project. DAKboard is a gorgeous web interface that displays photographs, weather and other information (such as events from your calendar or Wunderlist to-do list).

Start with a fresh installation of the Raspbian OS on your Raspberry Pi.

Begin by connecting to a wireless network. Click on the networking





A power adapter is attached to the rear of the display, allowing both the monitor and USB adapter to be hooked up to a single cable

The Raspberry Pi is fixed to the rear of the display. The DAKboard software is set up to launch in full-screen mode when the power is turned on

A short HDMI cable is used to connect the Raspberry Pi directly to the HDMI input on the display icon in the Menu bar and choose your wireless network.

Now open the Chromium web browser and set up an account at **dakboard.com**. You'll be taken to the DAKboard app interface. Before going any further, click the Options tab in DAKboard. Click on your account name (in the menu to the left) and choose Account. See the bit where it says 'Private URL' – you'll need this URL later, so minimise Chrome for now.

Configure Raspbian

Back in Raspbian, choose Menu > Preferences > Raspberry Pi Configuration. Ensure that Boot is set to Desktop and Auto Login is ticked.

We're going to turn SSH on, so it's a good idea to click Change Password first. Enter a new password for your wall display so you can access it remotely. Choose Interfaces and set SSH to Enabled.

Finally, click Localisation and Set Locale. Ensure that the Country is set to your location; it's 'GB: Great Britain' by default. Click on Set Timezone and choose the correct time zone for your locale; it's UTC (which is the same as GMT) by default.

Now we'll hide the mouse pointer after a few moments of inactivity. Open Terminal and enter:

```
sudo apt-get install unclutter
```

Next, make a couple of system configuration changes.

```
sudo nano /boot/config.txt
```

...and add:

```
# Display orientation.
Landscape = 0, Portrait = 1
display_rotate=1

# Use 24 bit colors
framebuffer_depth=24
```

Use **CTRL+O**, **ENTER**, and **CTRL+X** to save and exit Nano.

Now we want to force the screen to stay on, and load the Chromium browser with the DAKboard website on boot. Enter this in Terminal:

```
sudo nano ~/.config/lxsession/
LXDE-pi/autostart
```

Delete the contents of the **autostart** file and replace it with this:

```
@xset s off
@xset -dpms
@xset s noblank
@chromium-browser
--noerrdialogs --incognito
--kiosk http://dakboard.com/
app/?p=YOUR_PRIVATE_URL
```

After saving the file and exiting nano, enter **sudo reboot**. The Raspberry Pi will restart and when it boots, you will see DAKboard running in full-screen mode (without a mouse pointer).

The next step is to customise the DAKboard interface. Click on Options and use the Date/Time, Calendars, Photos, Weather, and News options to customise the display.

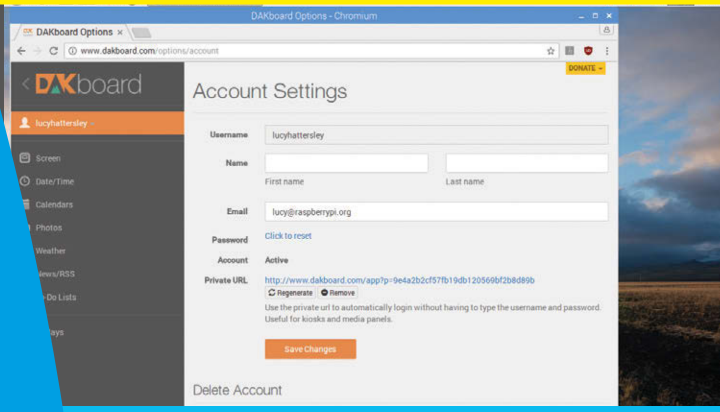
Now you need to fix your Raspberry Pi to the rear of your monitor and hang it on the wall. Dan from DAKboard suggests an IPS display: "You'll still be able to see it when viewing at an angle greater than 90 degrees," he explains.

If you plan to wall-mount the monitor, make sure the USB ports face down and not straight out (or the cables will stick into the wall).

Dan recommends a Dell S2340L 23-inch screen. "The plastic case is perfect for this setup," he says. "There's almost no bezel on the front, and the back cover pops off easily, leaving the frame which I then attached the photo frame wire to use for hanging."

You can find more details on the build on Dan's website (**magpi.cc/2mKaDcO**). DAKboard also sells a pre-built model, which is a monitor containing a Raspberry Pi, for $299.

SET UP DAKBOARD



DAKboard Options - Chromium

www.dakboard.com/options/account

Account Settings

Username: lucyhattersley

Name: First name / Last name

Email: lucy@raspberrypi.org

Password: Click to reset

Account: Active

Private URL: http://www.dakboard.com/app?p=9e4a2b2cf57fb19db120569bf2b8d89b

Use the private url to automatically login without having to type the username and password. Useful for kiosks and media panels.

Delete Account

>STEP-01

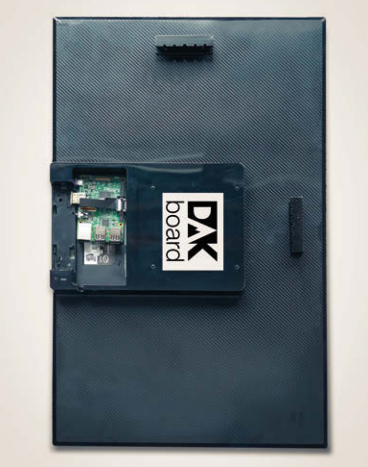
Set up DAKboard

Using a regular monitor and keyboard, set up the DAKboard software on your Raspberry Pi. Make sure it's connected to your wireless network and set up SSH so you can access it remotely.

>STEP-02

Fix the Raspberry Pi

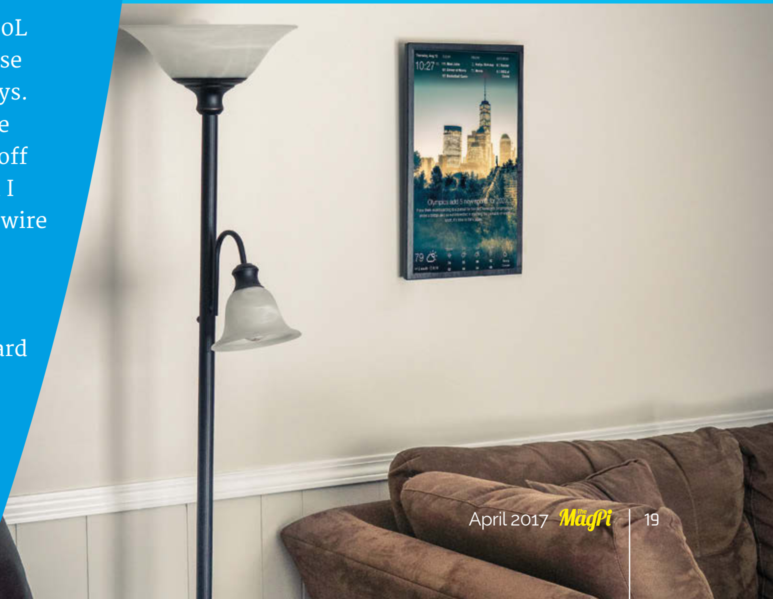
Attach the Raspberry Pi to the rear of the monitor. Using a short HDMI lead will help to contain the cables. A professional edition uses the DSI cable on the Raspberry Pi.



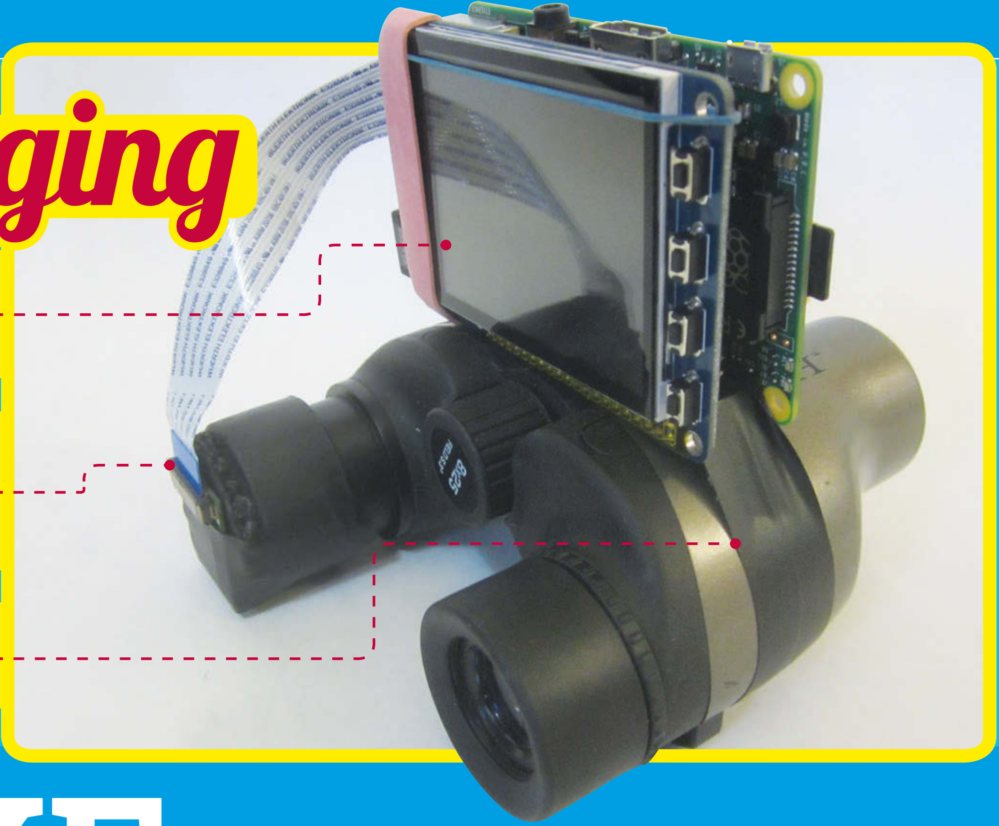
>STEP-03

Wall-mount the display

Now mount the monitor on your wall. Most monitors have brackets available for this purpose. You can also remove the bezel and use photo frame wire to hang the screen.



Challenging



An AdaFruit touchscreen is held firmly in place with two rubber bands (red and blue). This enables you to control the device out in the field

The Camera Module (connected via the CSI cable) is held in place using a circle of foam. More duct tape is used to hold the Camera Module firmly in front of the eyepiece

Duct tape is used to secure the Raspberry Pi and touchscreen display to the binoculars

MAKE PINOCULARS

The PiNoculars project is an excellent way to recycle an old pair of binoculars into a high-tech zoom recording device

YOU'LL NEED

- Binoculars
- Raspberry Pi
- Camera Module
- Adafruit touchscreen display
- Foam

The Raspberry Pi Camera Module is a great tool for digital making. With it you can quickly add an 8-megapixel camera to the Raspberry Pi board.

It connects directly to the Raspberry Pi using the CSI camera interface, a thin cable socket on the Raspberry Pi board.

The Camera Module is fantastic for video and still photography projects, such as time-lapse cameras. OpenCV (**opencv.org/**) software makes it easy to add computer vision to your projects. With this, you can train a Raspberry Pi to identify objects and react accordingly.

The PiNoculars project is one of our favourite Camera Module hacks.

Created by digital maker Josh Williams, PiNoculars are a regular set of binoculars, with a Raspberry Pi Camera module fixed over one of the eyepieces.

The Raspberry Pi is connected to the top of the binoculars along with a touchscreen display. The whole assembly is powered by a Goal Zero AA battery pack (**magpi.cc/2m9w6xr**).

"I was on 18-hour road trip back from Colorado to Michigan with my wife, and I was restless," Josh tells us. "I had brought the Raspberry Pi and Pi Camera Module along to play around with time-lapse photos in the mountains."

When Josh looked at his pair of binoculars, he had a brainwave. "Raspberry Pi and Camera Module and duct tape made for a crude prototype," says Josh.

When he got home, Josh set about refining the build. He now has

JOSH WILLIAMS

Josh helps run local maker spaces (All Hands Active and Maker Works). He builds amazing places for people to work on their projects.
magpi.cc/2mgA4pz

The finished project enables you to record long-distance video footage using an intelligent device

detailed instructions for two different types of PiNoculars. One follows the duct tape and foam route, while the second is a more complex build using laser cutting to create a mount for the Raspberry Pi and touchscreen.

"I used Adafruit's [PiTFT] capacitive touchscreen," says Josh. "Their tutorials made it incredibly easy to attach to the Pi (**magpi.cc/2mCpVy3**). Josh suggests that makers read Adafruit's DIY WiFi Raspberry Pi touchscreen camera tutorial by Phillip Burgess and the Ruiz Brothers (**magpi.cc/2m9Bxfv**).

The most time-consuming part of the build is creating a mount that wraps around the pair of binoculars, but you can skip all this by using foam and duct tape.

After setting up the Raspberry Pi with the Camera Module and Adafruit touchscreen, the whole kit is mounted on top of the binoculars.

The first step is to mark up and cut out a circle of foam. This serves as a mount for the Pi camera, to hold it in front of the eyepiece.

"The camera mount should barely cover the eyepiece," says Josh. It doesn't have to be perfect, but try to position the camera mount as close to the centre of the eyepiece as possible.

Josh then uses rubber bands to hold the Raspberry Pi and screen unit in place on top of the binoculars, and duct tape to fix the binoculars and Raspberry Pi together. "Be careful not to crush your LCD," he says. More duct tape is used to attach the foam mount over the PiNoculars eyepiece.

> There are a number of people who've combined the Raspberry Pi with microscopes and telescopes

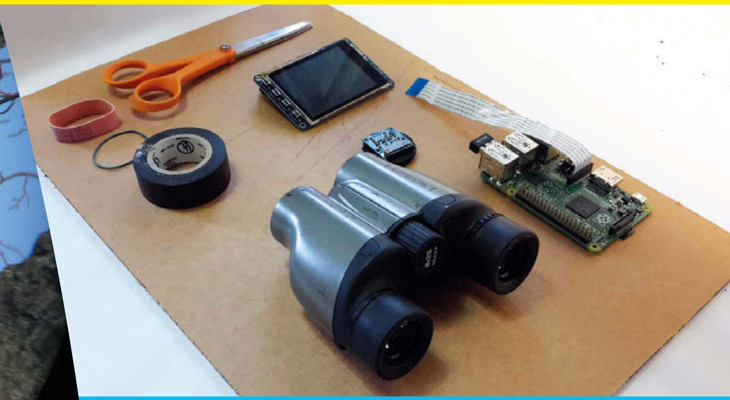
Then it's just a case of moving everything around until it is firmly fixed, and the camera can record a well-defined circle.

If you're a perfectionist, there's a much more detailed method, involving precision design with Inkscape (**inkscape.org**) and a laser cutter. But we like a quick hack, and this is a great project for making something quick and impressive.

"There are a number of people who've combined the Raspberry Pi with microscopes and telescopes," says Josh, who's fascinated by them.

"Remember to measure twice," he warns us, "and callipers are beautiful tools."

ATTACH A CAMERA MODULE

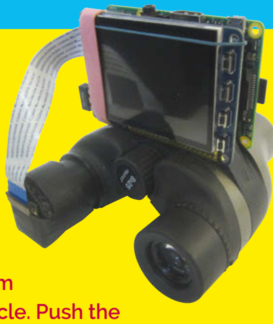


>STEP-01

Set up the camera and touchscreen

First, attach a Camera Module to your Raspberry Pi board and set up a capacitive touchscreen display. This will enable you to control the project on the move.

>STEP-02



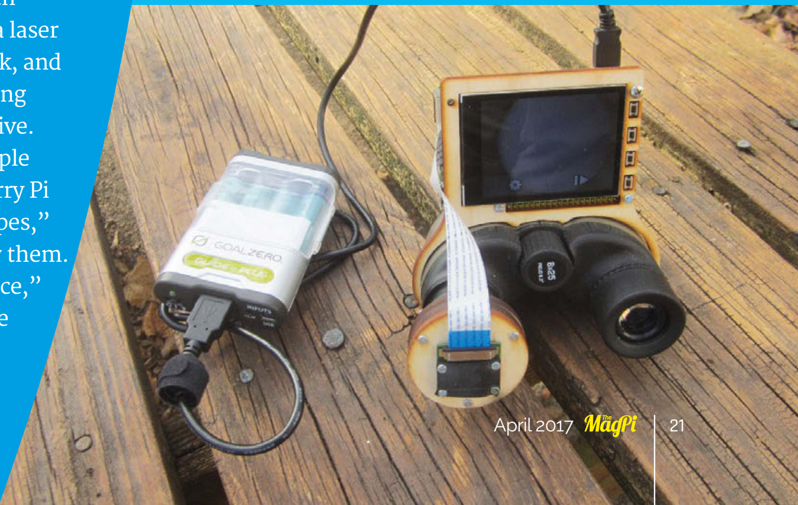
Attach the camera

Cut out a foam circle the same size as the eyepiece on your binoculars (it needs to cover the eyepiece completely). Mark an X in the centre and draw an 8×8mm square in the centre of the circle. Push the Camera Module into the foam and position it in front of the eyepiece.

>STEP-03

Fit it together

Use rubber bands to hold the Raspberry Pi and touchscreen display vertically on top of the binoculars, with the screen facing towards the eyepieces. Now use duct tape to secure the Raspberry Pi firmly, and a smaller piece to secure the foam-encased Camera Module. A portable power back enables you to take the PiNoculars out in the field for long-range video recording.



Challenging BUILD A YOUTUBE DRONE

Add a Camera Module to a drone to broadcast footage straight to YouTube

MAHMUT

Mahmut is a computer engineer and maker. He is working with his friend, Metin, on Sixfab Raspberry Pi 4G/LTE projects.
magpi.cc/2nSlpvp

YOU'LL NEED

- Drone Kit (and RC controller)
- Sixfab 4G/LTE shield
- Camera Module

This smart drone broadcasts footage over a cellular data connection direct to YouTube.

The YouTube Drone is a great project because it incorporates so many different Raspberry Pi features: video recording, flight control (either RC or automated), YouTube integration, and data network access.

It's also fundamentally inspiring. Hooking up a drone to a Raspberry Pi with network coverage lets you test a lot of future technologies.

There's potential to integrate it with computer vision technology, as well as automatic flight.

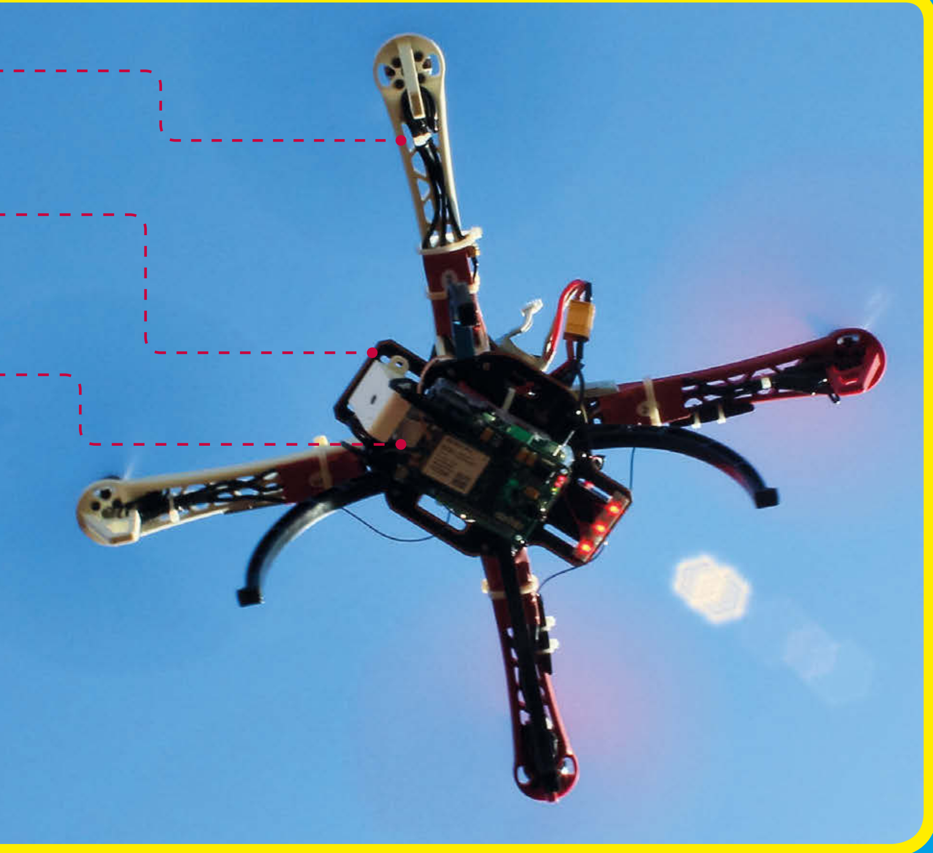
This project was built by Mahmut and Metin, a computer engineer from Sixfab and his maker friend.

"The Sixfab 4G/LTE shield offers high-speed internet connection to Raspberry Pi," says Mahmut.

Mahmut and Metin started with a drone kit: a Super Multicopter Set – Make Yourself Drone Kit from Robotistan (£180/$220, **magpi.cc/2mgXMjV**).

The drone set needs an RC controller with four channels. Robotistan recommends a RadioLink AT9 model RC controller (**magpi.cc/2mh8gQ8**).

The drone comes with a KK2 Multi-rotor LCD flight controller board (**magpi.cc/2mh0CWa**). The flight controller has GPIO pins, so it is possible to create



The drone is built using a kit model. This is controlled remotely with an RC controller

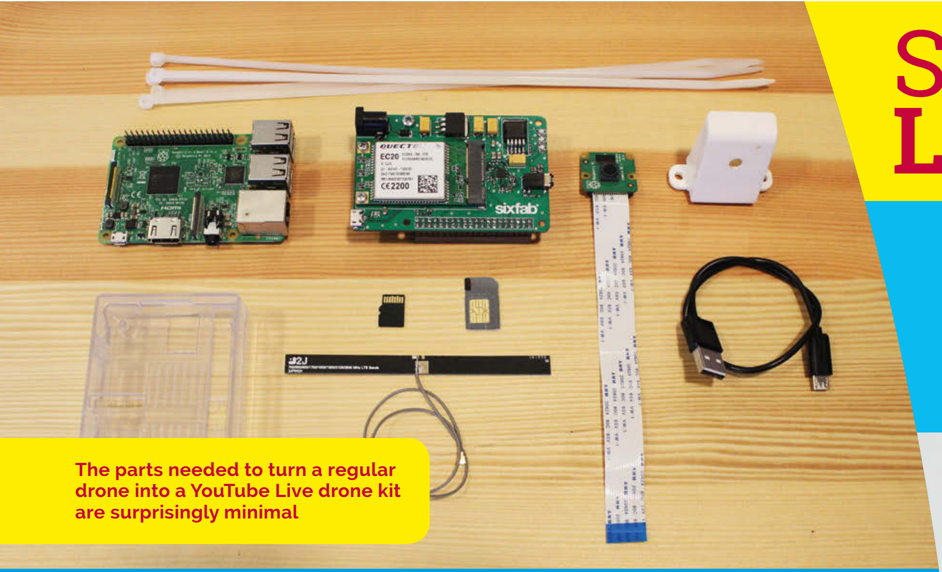
A Camera Module is used to record footage from the drone. This is shared directly to YouTube Live using the data connection

A Raspberry Pi and Sixfab 4G/LTE shield are mounted on the underside of the drone. This provides a permanent data connection

UK DRONE LAW

The UK Civil Aviation Authority's 'The Drone Code' is a guide to flying drones for fun (**magpi.cc/2nRZAw2**).

In a nutshell: Keep your drone under 120m and 50m away from other people, vehicles, or buildings. Stay clear of airports, airfields, and aeroplanes. Keep your drone in your field of vision (within 500m) at all times.



The parts needed to turn a regular drone into a YouTube Live drone kit are surprisingly minimal

an automated drone, although Mahmut and Metin's project currently flies via remote control.

The Raspberry Pi community discusses using a Pi with the KK2 flight controller in this forum thread: **magpi.cc/2nScpGB**.

You're not limited to the Super Multicopter. "You can use any drone capable of carrying an extra 200-250 grams," explains Mahmut.

After building the drone kit, the Raspberry Pi is connected to the Sixfab 4G/LTE shield. The Camera Module is attached to the Raspberry Pi and the whole assembly is mounted to the underside of the drone. Mahmut has produced a video tutorial demonstrating the build (**youtu.be/F44R5PaV25M**).

And an SD card image has been produced especially for the project (**magpi.cc/2nRY7FQ**).

Setting up the Sixfab 4G/LTE shield is relatively easy. A driver is available direct from the Sixfab website (**magpi.cc/2nRW9W3**). Copy the files to **/lib/modules/4.4.21-v7+/kernel/drivers**, reboot the Raspberry Pi, and run **`dmesg`** in a Terminal window.

Prospective makers will find this Sixfab tutorial helpful: YouTube/Facebook Live Video Streamer with Sixfab 3G/LTE shields (**magpi.cc/2nRZJzp**).

Please be careful when building a drone. It is important to not to make the connections between propellers and motors before it is complete and fully secured. Wear safety glasses when testing the drone, as spinning blades are dangerous.

"It is very easy for people with all the materials to do the project," says Mahmut. "Because it's very easy to use the Sixfab products and everyone can take it and do whatever they want with ease."

"The first time we did it we used a smaller drone and it would not go up," he tells us. "Other than that, we have not had a problem. But during the flight, the wind made our work very difficult.

"Those who want to do the project should not hesitate. They can easily complete it with the videos and the image file."

The current project uses the Pi and Sixfab shield 4G/LTE connection for YouTube Live streaming. "But our goal is to also control the drone via the network connection," says Muhmut.

> "Those who want to do the project should not hesitate. They can easily do it with the videos"

STREAMING LIVE VIDEO

>STEP-01

Set up the shield

A Sixfab 4G/LTE shield is connected to the Raspberry Pi. This provides a persistent data connection, enabling the Raspberry Pi to stream data to YouTube and other websites.

>STEP-02

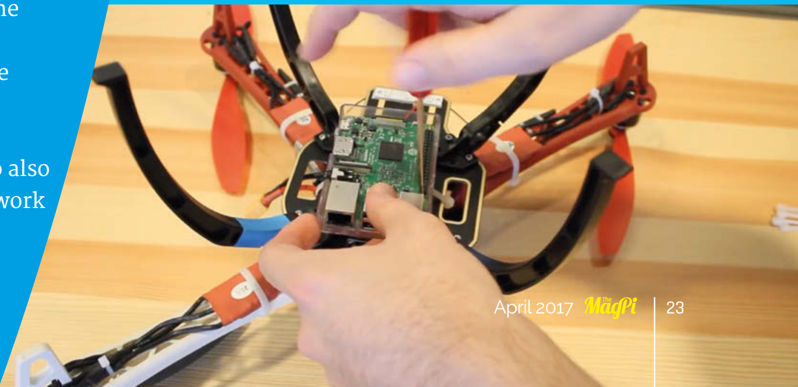
Attach the camera

A Camera Module is attached to the Raspberry Pi board. Meanwhile, a USB cable is used to connect the Sixfab shield to the Raspberry Pi. The LTE band provides a more powerful antenna for the shield, so it maintains a steady data connection.

>STEP-03

Mount the device

The Sixfab shield and Camera Module are mounted on the underside of a quadcopter. A battery unit is mounted on the top of the drone to provide steady power.



THE TOP TEN RASPBERRY PI KITS

Want to build a project, but don't want to make everything from scratch? Here are some of the best kits around. Everything from retro gaming systems to wildlife cameras can be bought in kit form

PICADE

£180
$239

magpi.cc/2nS0fvv

Creating a retro arcade system is a dream for many makers. Building a cabinet from scratch can be a costly and difficult enterprise. Picade makes it all a lot easier, though. Its powder-coated cabinet feels like a real arcade cabinet. It comes with a PCB, joysticks, and 12 arcade buttons.

RASPBERRY PI 3 RETRO GAMING BUNDLE

£65
$79

magpi.cc/2nSXHPs

If you're looking for a retro gaming console, then look at this Retro Gaming Bundle from The Pi Hut. It contains a Raspberry Pi 3 and two SNES-style USB gamepads. You also get a long HDMI cable, an official Raspberry Pi power supply, and a 16GB SD card. All you need to do is install RetroPie (**retropie.org.uk**).

PIRATE RADIO

£40
$49

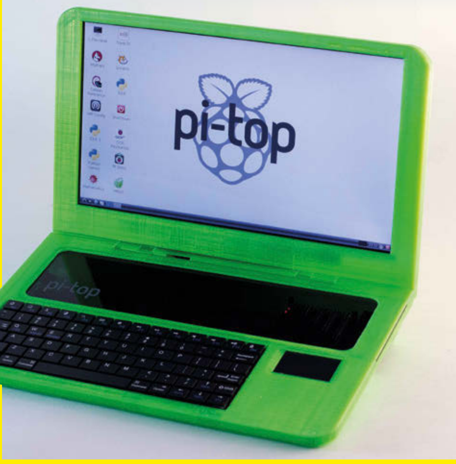
magpi.cc/2nSE0fE

This kit has everything you need to create a radio with your Raspberry Pi. It contains the new Pi Zero W and a pHAT BEAT (DAC and stereo amplifier). You also get a 5W speaker. The plastic case has a VU meter so you can view the sound levels. You will need to solder the GPIO header on to the Pi Zero W, and a female header to the pHAT BEAT.

PI-TOP

pi-top: £229.19 / $264.99
pi-topCEED: £109.99 / $164.99

pi-top.com



The pi-top and pi-topCEED are projects to turn your Raspberry Pi into a laptop or desktop computer. Unlike a regular laptop or desktop, these are module DIY computers that are ideal for mobile making. You can 3D-print the case and buy the components, or get the whole lot in kit form. It's not cheap, but it's designed to teach the basic architecture of a computer, and there's plenty of documentation and tutorials.

PI ZERO CCTV KIT (LITTLE BRO)

£24
$29

magpi.cc/2nSMYVz

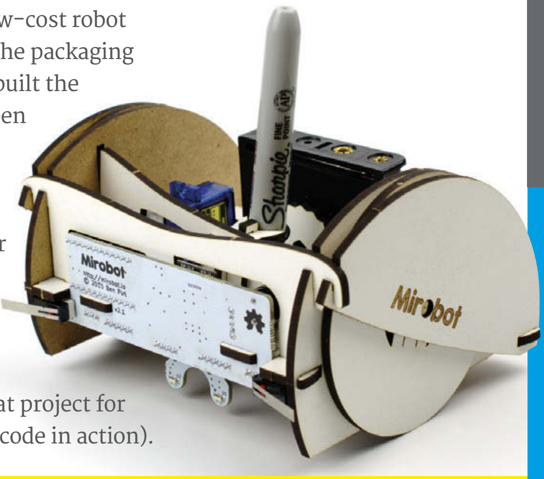
Start your mini surveillance state with this sign that houses a Pi Zero and a Camera Module. The camera logo on the sign has a hole for the Camera Module. You need to buy the latter separately, but can combine it with OpenCV computer vision to create a smart CCTV Camera that recognises people.

MIROBOT V2

£60 $75

magpi.cc/2nSJMt0

The Mirobot is a small, low-cost robot that you assemble using the packaging it comes in. Once you've built the robot, you add a felt-tip pen and use it to move and draw. The Mirobot acts like your average turtle robot. It can raise or lower the pen to draw shapes on paper. Turtle robots have a long history in computing and maker culture, and they're a great project for learning logic (and to see code in action).



WILDLIFE CAM KIT

£130 $158

naturebytes.org

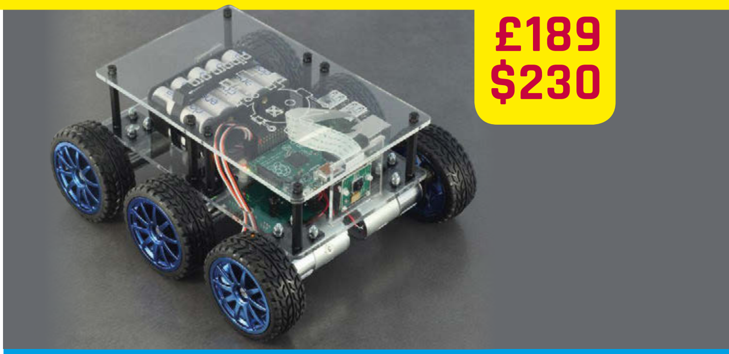
If you've ever wanted to record the critters in your garden, then the Wildlife Cam Kit is the way to go. Its PIR sensor detects movement and triggers the Pi Camera Module to take a stealthy snap. It's ideal for educational use. The Cam Kit is also very versatile and can be used for time-lapse photography, night-time shots (with a Pi NoIR camera and IR LED lighting), or even a live video feed.

LIV PI

£240 $250

livpi.com

LiV Pi comes from Hong Kong, where pollution has been a problem for many years. With air quality increasingly a concern in cities across the world, it's a great way to learn more about pollution (and keep an eye on levels in your area). Inside the kit are three sensors: carbon dioxide, temperature/humidity, and air pressure. It's not a cheap project, but it is a professional air-monitoring system.



£189 $230

DIDDYBORG

magpi.cc/2mHbmbp

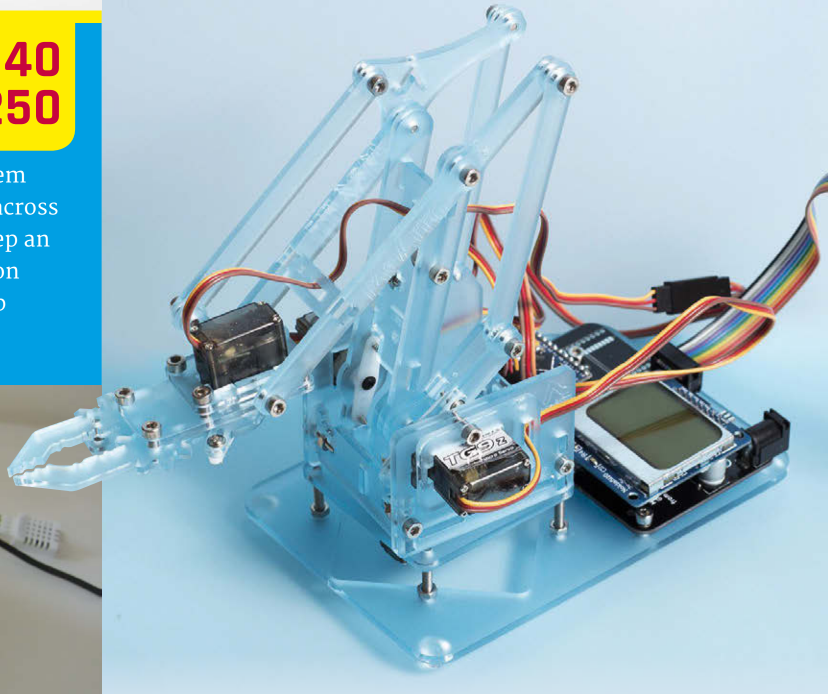
There are many robotics kits for the Raspberry Pi, but PiBorg's DiddyBorg is perhaps the most comprehensive. It has a robust laser-cut chassis with six 60rpm motors. The DiddyBorg comes with a PicoBorg Reverse motor controller and a BattBorg power management unit. You need to add a Raspberry Pi and Camera Module, and you have the option to control the robot using a PlayStation 3 gamepad.

MEARM

£32.99 $41.21

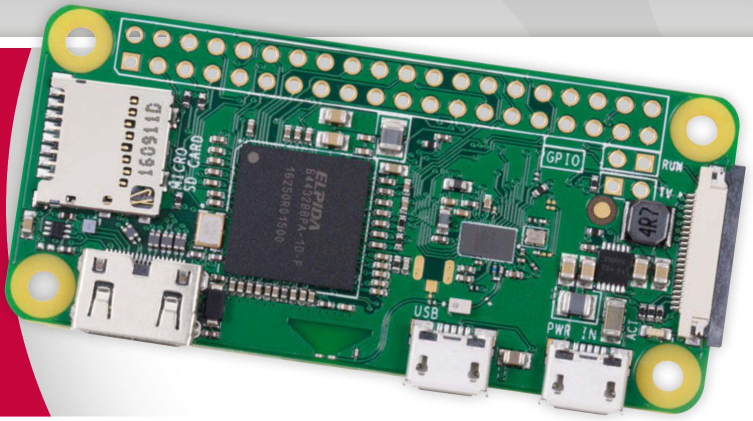
magpi.cc/2nS0lDo

Not all robots have wheels. The MeArm is a flat-pack robot arm kit that you build. It can then be controlled via Python or directly with a joystick. It's very easy to assemble, using just a screwdriver, and we think this is a great kit for anyone wanting to step into the world of digital making.



SUBSCRIBE TODAY AND RECEIVE A

FREE PI ZERO W



PLUS OFFICIAL PI ZERO CASE WITH 3 COVERS

AND FREE CAMERA MODULE CONNECTOR AND USB / HDMI CONVERTER CABLES

Subscribe in print for 12 months today and receive:

- A free Pi Zero W (the latest model)
- Free Pi Zero W case with 3 covers
- Free Camera Module connector
- Free USB and HDMI converter cables

Other benefits:

- Save up to 25% on the price
- Free delivery to your door
- Exclusive Pi offers and discounts
- Get every issue first (before stores)

THE OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
INCREDIBLE RASPBERRY PI PROJECTS
Get started with digital making today
NEW RASPBERRY PI ZERO W OUT NOW!
RETRO GAMING BLISS
PI ZERO NOW HAS WIRELESS!
BUILD A MAGIC MIRROR

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO W!

Get three issues:

£12.99 (UK) (Direct Debit)

$37.50 (US) (quarterly)

How to subscribe:

- magpi.cc/Subs-2 (UK / ROW)
- imsnews.com/magpi (USA)
- Call +44(0)1202 586848 (UK/ROW)
- Call 800 428 3003 (USA)

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

YES! I'd like to subscribe to The MagPi magazine and save money

This subscription is: □ For me □ A gift for someone*

Mag#56

YOUR DETAILS Mr □ Mrs □ Miss □ Ms □

First name .... Surname ....

Address ....

....

Postcode .... Email ....

Daytime phone .... Mobile ....

If giving The MagPi as a gift, please complete both your own details (above) and the recipient's (below).

GIFT RECIPIENT'S DETAILS ONLY Mr □ Mrs □ Miss □ Ms □

First name .... Surname ....

Address ....

Postcode .... Email ....

PAYMENT OPTIONS

1 DIRECT DEBIT PAYMENT £12.99 every 3 issues (UK only)
Instruction to your bank or building society to pay by Direct Debit

DIRECT Debit

Please fill in the form and send to:
The MagPi, Select Publisher Services Ltd,
PO Box 6337, Bournemouth BH1 9EH

Service user number 8 3 8 7 7 3

Name and full postal address of your bank or building society:

To: The Manager Bank/building society ....

Address ....

....

.... Postcode ....

Name(s) of account holder(s) ....

Branch sort code □□□□□□ Account number □□□□□□□□

Reference □□□□□□□□□□□□□□□□□□ (Official use only)

Instruction to your bank or building society
Please pay Select Publisher Services Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit Guarantee. I understand that this instruction may remain with Select Publisher Services Ltd and, if so, details will be passed electronically to my bank/building society.

Signature .... Date □□/□□/□□

Banks and building societies may not accept Direct Debit instructions for some types of account.

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

6 ISSUES	□ UK £30	□ Europe £45	□ Rest of World £50
12 ISSUES	□ UK £55	□ Europe £80	□ Rest of World £90

2 CHEQUE
I enclose a cheque for .... (made payable to Select Publisher Services Ltd)

3 CREDIT/DEBIT CARD □ Visa □ MasterCard □ Maestro □ Switch

Card number □□□□□□□□□□□□□□□□

Expiry date □□ □□ Valid from □□ □□ (if shown)

Issue number □□ (if shown) Security number □□□
(last 3 digits on the back of the card)

Signature .... Date □□/□□/□□

I would like my subscription to begin from issue .... (month + year)

RETURN THIS FORM TO:
MagPi Magazine Subscriptions, Select Publisher Services Ltd, PO Box 6337,
Bournemouth BH1 9EH

□ Please tick this box if you DO NOT want to receive any other information from Select Publisher Services Ltd.

□ Please tick this box if you DO NOT want to receive any other information from other companies.

□ Please tick this box if you DO NOT want to subscribe to The MagPi newsletter.

MATT WAGNER

Matt spends his time developing new Raspberry Pi projects, for a more connected world. In the future he hopes to spread this passion for a full-time profession.
hackmypi.com

PIMINIMINT

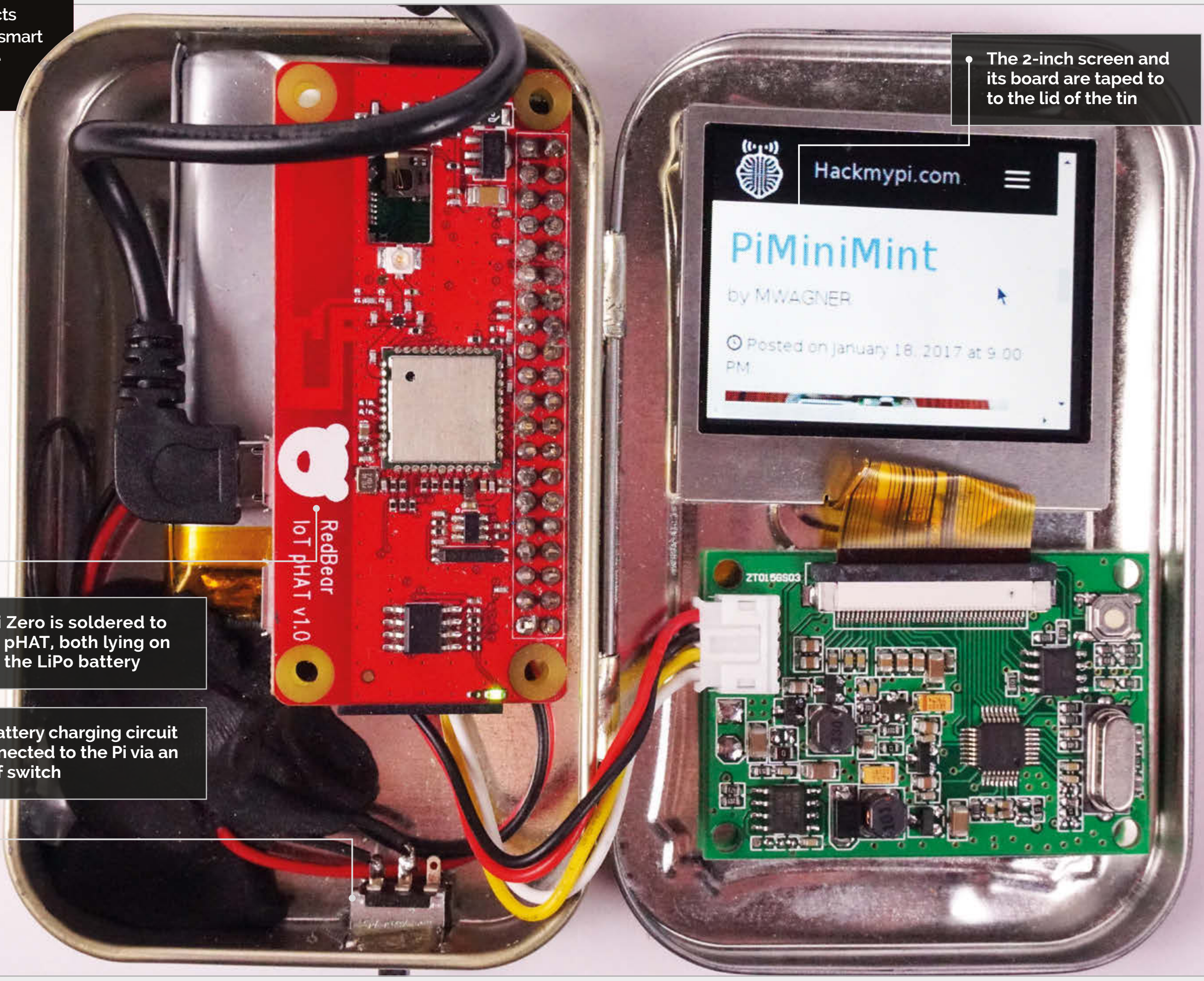
A fully functioning computer and display crammed into a mints tin

Quick Facts

- The original version featured a Pi NoIR camera
- Sugru stops the charger port moving around
- Hot glue is added to secure solder points
- Matt's new model will feature a phone SIM
- Planned Pi projects include smart luggage tracking

Matt Wagner first thought about fitting a computer into an Altoids tin in 2012, shortly after the launch of the original Raspberry Pi. While it wasn't quite possible back then, the release of the considerably smaller Pi Zero changed everything, although not immediately for Matt. "I was deep into another project, and the low availability of Zeros made it challenging to find and use for anything at the time," he recalls. "However, I now work in an environment [Micro Center store] surrounded with DIY materials, so the idea just came back to me one day."

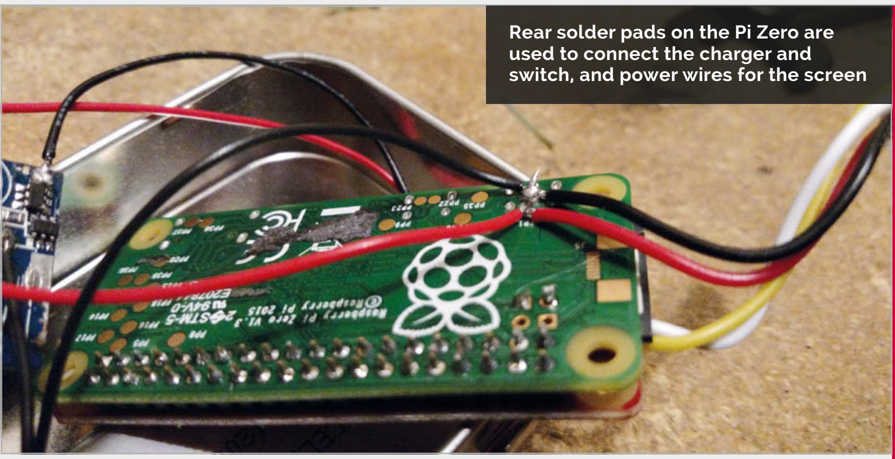
The result is the PiMiniMint, containing a Pi Zero connected to a 2-inch LCD display and – thanks to a RedBear IoT pHAT – equipped with WiFi and Bluetooth. "The original design took a few weeks, between prototyping and sourcing parts," Matt tells us. "The concept of it is simple: strap together already available components. In actuality, though, sourcing parts was a bit of an issue



The 2-inch screen and its board are taped to to the lid of the tin

The Pi Zero is soldered to an IoT pHAT, both lying on top of the LiPo battery

The battery charging circuit is connected to the Pi via an on/off switch



Rear solder pads on the Pi Zero are used to connect the charger and switch, and power wires for the screen

(shipping costs, and buying pre-made parts gets very expensive), so I had to find creative ways to solve problems."

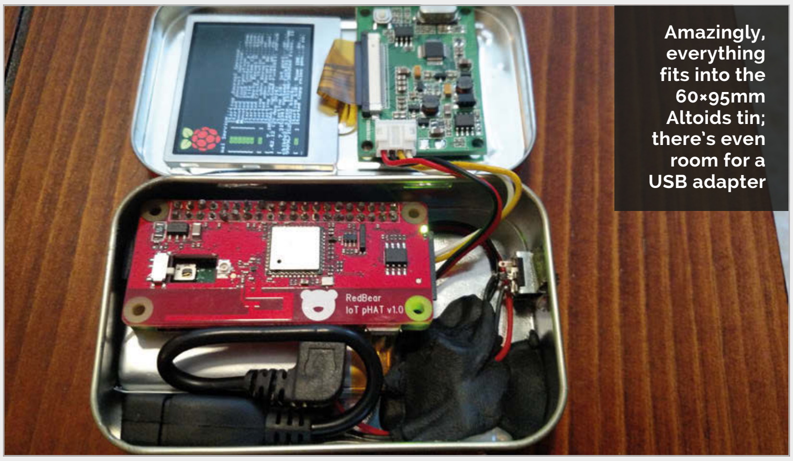
This included a solution for rechargeable battery power. "In my first iteration, I had no way to charge the battery," explains Matt. "An Adafruit PowerBoost would have been ideal, as it puts out enough amperage, but the cost would have been tight for me. The [Pimoroni] Zero LiPo is amazing, and I am currently using it in an ongoing project of mine; however, it sits too high on the GPIO and wouldn't allow for the lid to close with the screen inside."

In the end, at the suggestion of a co-worker, Matt opted to modify a standard USB phone charger, extracting its charging circuitry to use with a slimline LiPo battery. He also added an on/off switch. "I had some issues regarding wiring and power needs, but those were related to where in the circuitry I put my switch, and the kind of switch I used," he reveals. "Changing to a better switch did the trick." Matt's 2500mAh LiPo provides impressive battery life: "I haven't done a full 100%-0% test and timed it, but I have had it on for six-plus hours at a time."

One drawback of using a built-in battery was that there was no longer room in the tin for the infrared camera and full-size exterior USB port featured in the original design. "However, I do still have an interior full-size USB [via an adapter] that is available when opened."

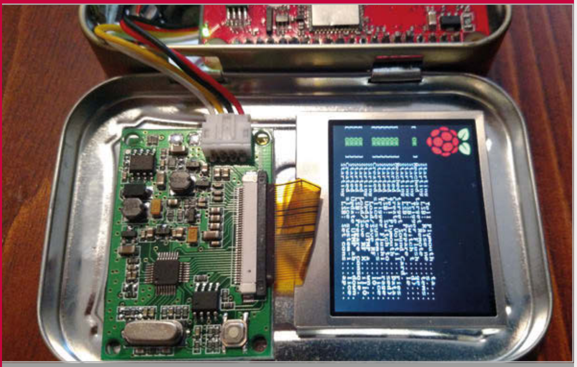
Matt is currently making a second PiMiniMint for a friend, featuring a few modifications: "His is going to have two on-board WiFi antennas and a dedicated SIM for transmitting data over 3G/4G."

As well as it being a cool pocket PC, Matt thinks the PiMiniMint is ideal for portable retro gaming. "Additionally, I have used it as a WiFi access point for saving files to, a WiFi repeater by plugging a second WiFi antenna into the USB port, and I'm sure there are more ideas out there!"



Amazingly, everything fits into the 60×95mm Altoids tin; there's even room for a USB adapter

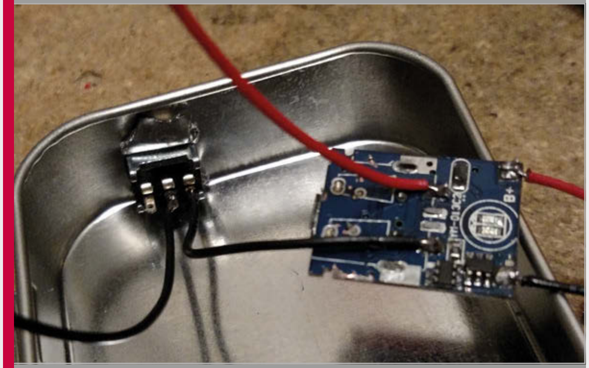
HOW TO TIN A PI



>STEP-01

Mount the screen

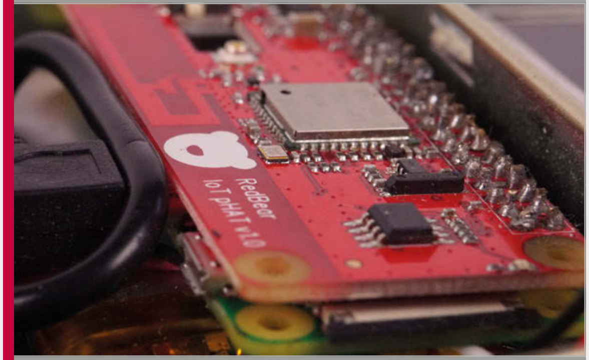
The 2-inch screen is attached to the lid of the Altoids tin using double-sided tape. The display board's white charge port needs to be near the hinge to avoid stretching the wires.



>STEP-02

Make a charger

Matt stripped the mini circuit board from a standard phone charger and removed its USB port. This board is then soldered to a slimline LiPo battery, and connected via a switch to the Pi Zero.



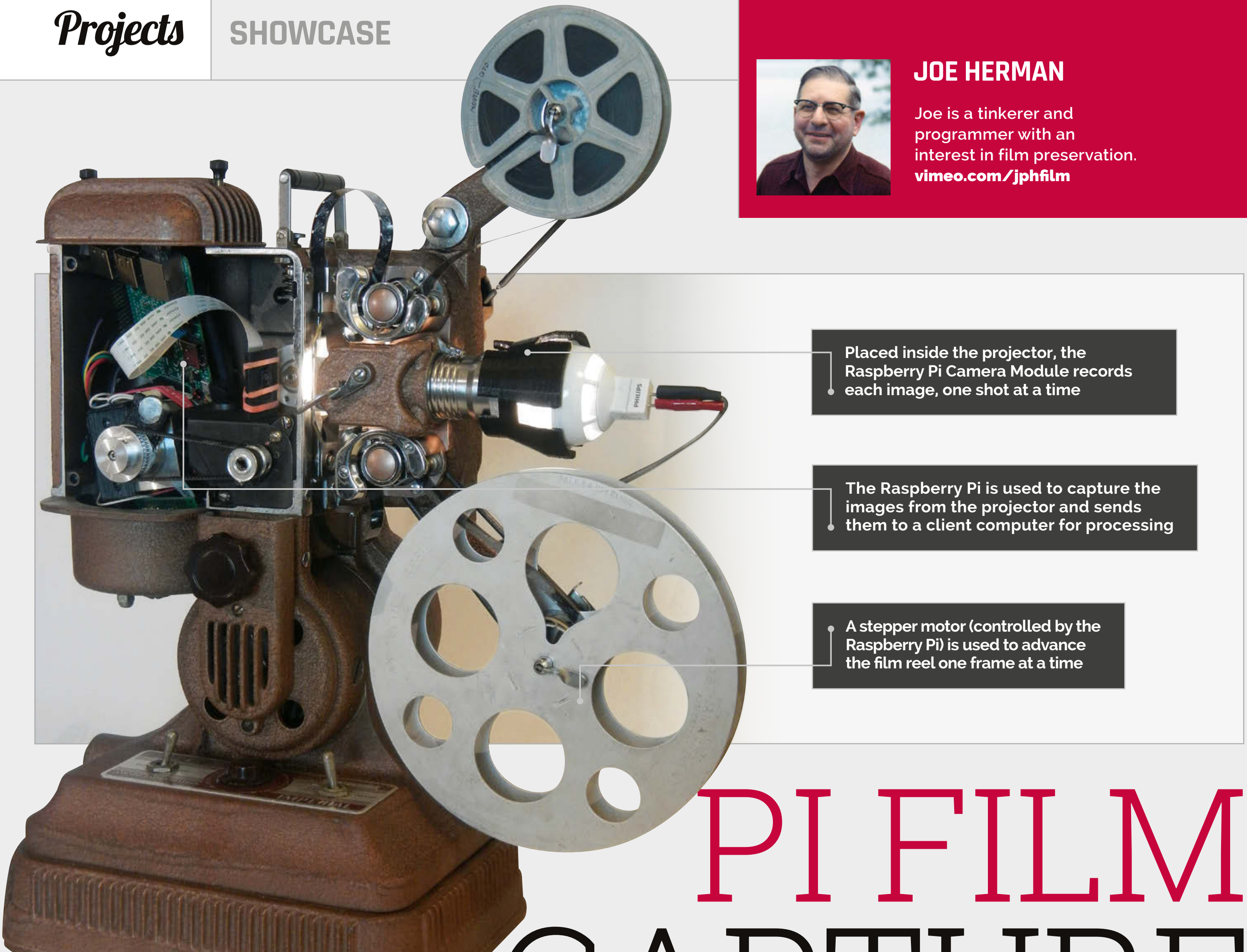
>STEP-03

Add connectivity

A headerless RedBear IoT pHAT is soldered directly via a male 40-pin header to the Pi Zero to provide WiFi and Bluetooth connectivity. If using the new Pi Zero W, this could be omitted.

JOE HERMAN

Joe is a tinkerer and programmer with an interest in film preservation.
vimeo.com/jphfilm



Placed inside the projector, the Raspberry Pi Camera Module records each image, one shot at a time

The Raspberry Pi is used to capture the images from the projector and sends them to a client computer for processing

A stepper motor (controlled by the Raspberry Pi) is used to advance the film reel one frame at a time

PI FILM CAPTURE

Expertly converting old film reels to digital video one frame at a time

Quick Facts

- A five-minute film reel can produce 4,000 still images
- 8mm captures need a 25mm focal length magnifying lens
- The projector uses a NEMA 17 motor and a DRV8825 chip
- Joe 3D-printed mounts for the lamp and motor
- It uses the original 5MP Camera Module

Restoring home movies captured on film can be a real challenge. Film conversion is a time-consuming and often expensive task.

One maker, Joe Herman, took matters into his own hands by hooking up a Raspberry Pi Camera Module to an old projector.

Joe's cousin and uncle uncovered a box of 130 reels recorded by his grandfather. Leo's moves "dated back to 1938," says Joe.

The preservation of these memories was important for Joe's large extended family. But to have them professionally restored would have cost "many thousands of dollars."

The obvious approach with a project like this would be to run the movie and record it using a camcorder. The results from that approach are very poor, though.

Professionals photograph one still image at a time and then stitch it all together, which is why the process is so expensive.

Pi Film Capture is a smart film transfer system that uses a Raspberry Pi and Pi Camera Module with Bell & Howell 8mm and Super 8 projectors, or an Ampro Imperial 16mm.

The setup employs a frame-by-frame approach. The Raspberry Pi takes a still photo of a film frame and sends it to a remote client computer for processing.

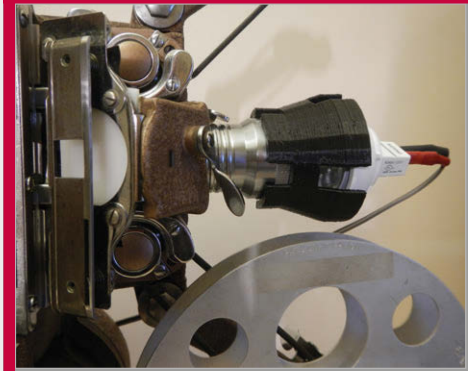
The Raspberry Pi is connected to a stepper motor that moves the frame to the next cell. It then repeats the process until the entire reel is captured. "Most old projectors would work," reckons Joe. "They were built to last."

Once the process is completed, the result is a folder of images. "I can combine these quickly into a viewable movie using FFmpeg (**ffmpeg.org**)," says Joe. If a more sophisticated clean-up is required, he uses a program called AviSynth (**avisynth.nl**).

"A key design goal for me was to use easily obtainable parts," reveals Joe. "Mostly because when I began the project, I was unsure whether it would go anywhere and didn't want to invest too much in expensive equipment."

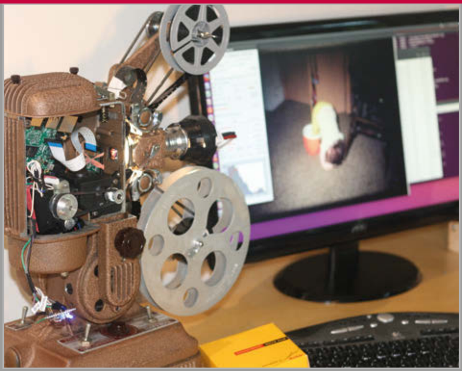
Joe solved "endless" challenges while designing Pi Film Capture.

BUILDING A CINE-CONVERTER



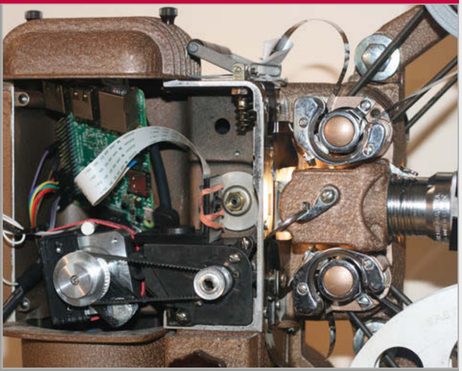
>STEP-01
Direct capture

A Pi Camera Module is stripped back and placed directly inside the lens housing. This approach removes any interference from outside light.



>STEP-02
Client computing

Each still frame of the reel is captured, one frame at a time. The Raspberry Pi sends the images to a client computer for storage and editing.



>STEP-03
Stepper motor

After capturing the frame and sending it to the client, the Raspberry Pi controls a stepper motor that moves the film to the next frame.

"Most were due to my inexperience and ineptitude," he tells us. "Or bad programming techniques."

The first proof-of-concept device drove the projector and triggered a capture as each frame advanced. "It fooled me into thinking this would be an easy project," recalls Joe. But frames were often under- or over-exposed. Sometimes it would miss frames altogether.

As Joe iterated through different versions of the project, he added a separate client computer: a PC running Ubuntu Linux.

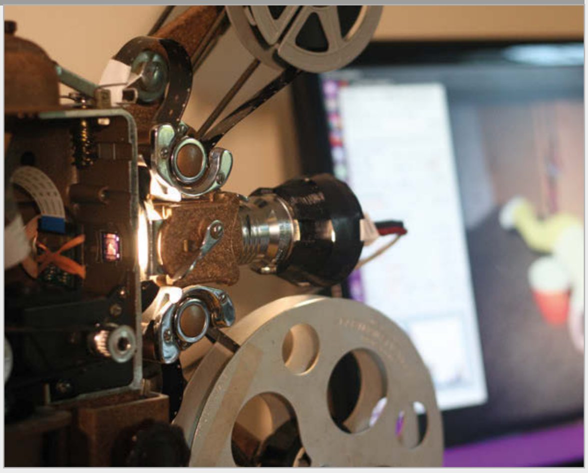
The client computer runs a control panel that lets the user adjust the camera, and displays the captured frames as they come off the Raspberry Pi. It also sends some simple projector control commands to the Pi.

The latest version of the Pi Film Projector uses OpenCV (**opencv.org**). This software adds computer vision and multi-core processing to the project.

"OpenCV provides functions similar to the HDR mode on cell phone cameras," explains Joe. Given several photos at different exposures, it can blend them into a single image that preserves detail in both bright and dark areas.

"For some films, this can make an enormous difference," says Joe.

Joe has captured over 100 reels of film recorded by his grandfather

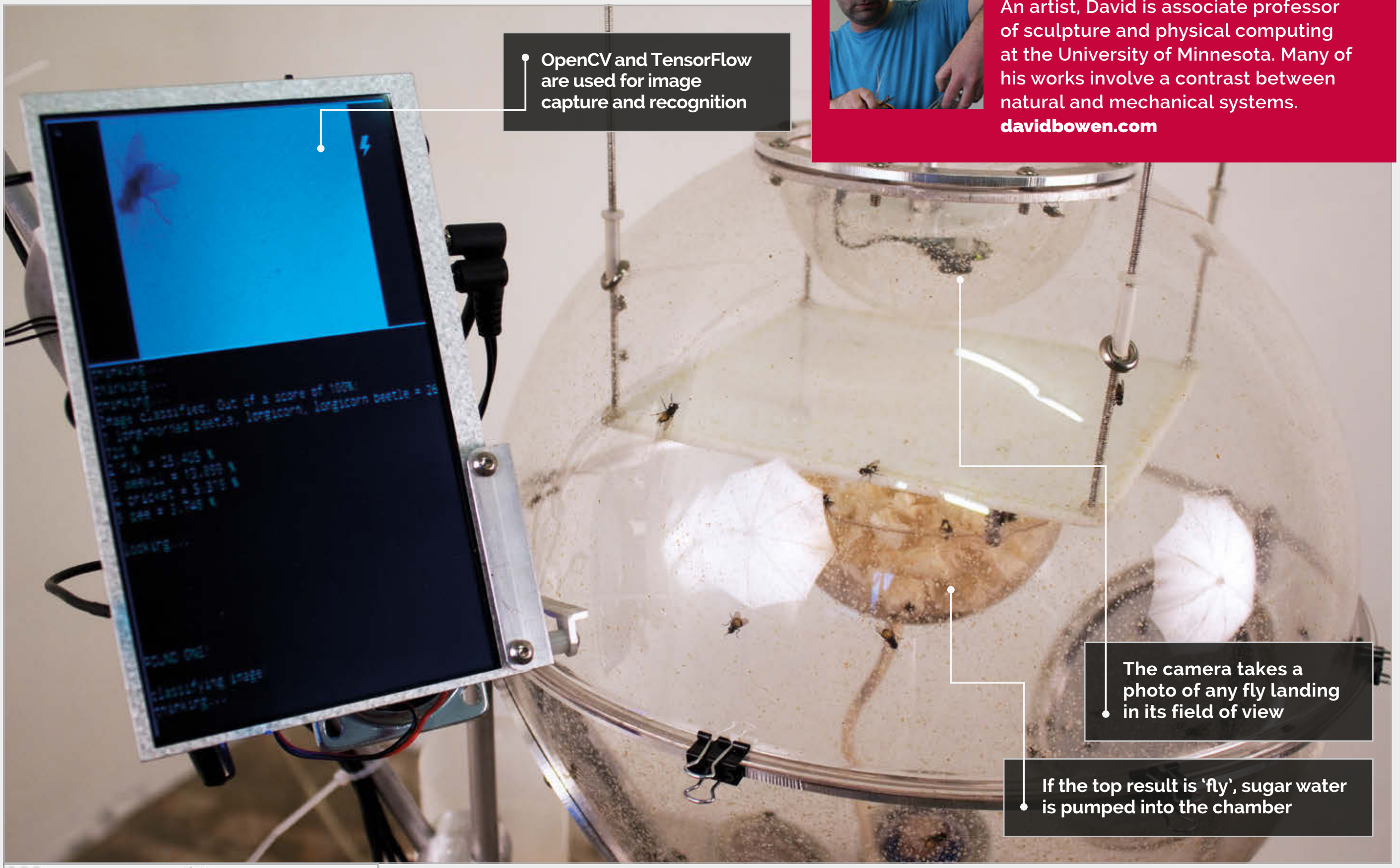


Above As the project has progressed, the image editing software has become more advanced, creating higher-quality recordings

"The downside is it slows down my captures by nearly half as I take more photos, so I try only to use it when necessary."

Joe advises prospective makers to have patience with the build. "It has a lot of pieces," he says. The captures take a lot of time, and getting good results can take a lot of practice.

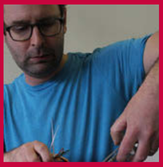
The results are worth it. Joe has converted over 100 of his grandfather's reels. "Bringing some family history to life again and sharing it has made this the most rewarding tech project I've ever worked on," he tells us. "I hope it inspires other people to bring that old box of film cans out of the closet and reconnect with their history."



OpenCV and TensorFlow are used for image capture and recognition

The camera takes a photo of any fly landing in its field of view

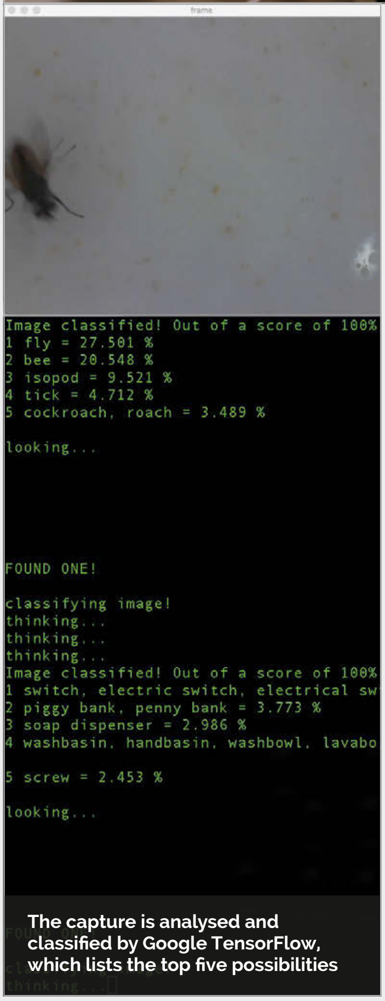
If the top result is 'fly', sugar water is pumped into the chamber



DAVID BOWEN

An artist, David is associate professor of sculpture and physical computing at the University of Minnesota. Many of his works involve a contrast between natural and mechanical systems.

davidbowen.com



```
Image classified! Out of a score of 100%
1 fly = 27.501 %
2 bee = 20.548 %
3 isopod = 9.521 %
4 tick = 4.712 %
5 cockroach, roach = 3.489 %

looking...




FOUND ONE!

classifying image!
thinking...
thinking...
thinking...
Image classified! Out of a score of 100%
1 switch, electric switch, electrical sw
2 piggy bank, penny bank = 3.773 %
3 soap dispenser = 2.986 %
4 washbasin, handbasin, washbowl, lavabo

5 screw = 2.453 %

looking...
```

The capture is analysed and classified by Google TensorFlow, which lists the top five possibilities

FLYAI

The fate of the whole colony depends on a Raspberry Pi

Quick Facts

- The flies can live for 30-40 days
- David says he'd intervene if needed
- The work isn't anti-AI, just a "commentary"
- Previous projects include tweeting flies
- He's hoping to exhibit flyAI in the autumn

Flies crawl around inside a transparent sphere, spied on by a camera, as a disembodied voice reads out image recognition data that determines whether or not they will be fed. This is the unsettling theme of flyAI (**magpi.cc/2n7pijV**), a Pi-powered art installation by David Bowen, who was inspired to create it after reading philosopher Nick Bostrom's *Superintelligence: Paths, Dangers, Strategies.* The book explores the prospective effects of future super-intelligent AI on the human race and while there are some utopian possibilities, David says there are also dangers: "The AI could end up repurposing our atoms for food!"

David tells us why he uses houseflies in some of his work: "They're often a stand-in for the absurd or, thinking about collective behaviour, the will of the collective," although their actions aren't quite as random as you might expect: "They do like to work their way to the tops of containers; they're attracted to light.

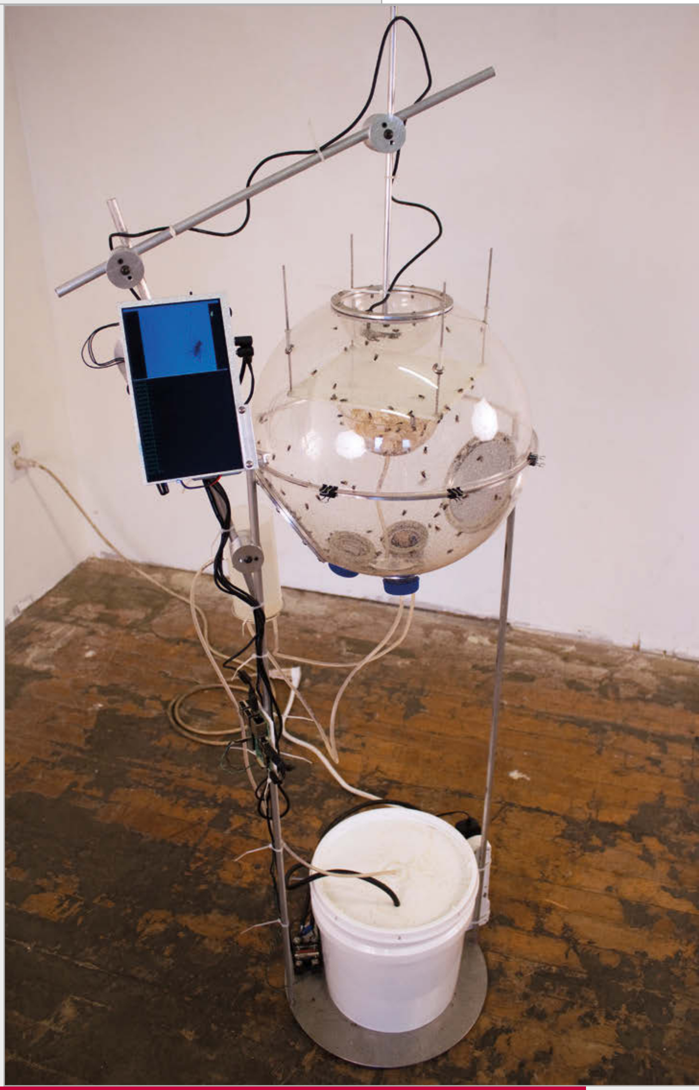
"Also, they don't elicit as much empathy from viewers as another living creature," adds David. That's probably just as well given the nature of flyAI, in which the flies' survival depends on the accuracy of image recognition. A regular USB webcam positioned at the top of the custom-built fly chamber is connected to a Raspberry Pi 3 which employs OpenCV to determine when a fly is in view and take a still. The photo is then classified by the Google TensorFlow image recognition algorithm which ranks the top five possibilities in order, along with their percentages. The results are revealed gradually on an LCD screen, as if being typed, and also read out aloud using the Festival text-to-speech system.

If 'fly' is ranked number one by the algorithm – which happens around 60% of the time according to David – the flies are given some food. "So their fate is determined by how accurate this AI is." The feeding process is also controlled by the Pi, which triggers a relay to start a pump to supply sugar water from a bucket, up through tubes into the middle of the chamber.

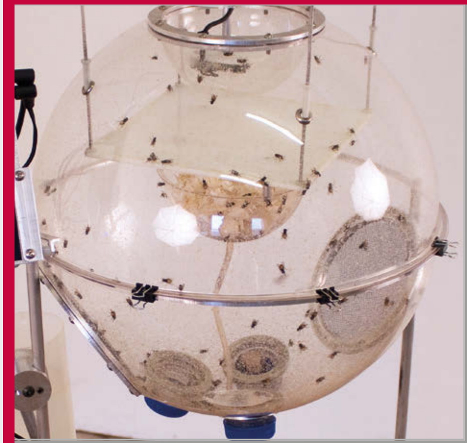
Having run the experiment for six weeks in his workshop, David reveals that the survival rate is good: "They live for about 30-40 days, which I'm told is longer than they live in the wild… they were getting kind of geriatric!" This is despite them interfering with the image recognition in a certain way: "The flies poop all over the inside of the sphere, including where the camera is." While David obtains his flies from an online source, he thinks they could possibly reproduce within the chamber if given the right conditions to lay their eggs.

> "The flies are a stand-in for the absurd or, thinking about collective behaviour, the will of the collective"

Looking to add further elements to the project, David is exploring the possibility of using the fly corpses to provide material for a microbial fuel cell that he's working on, which in turn will power the Raspberry Pi. He would also like to implement machine learning so that the image recognition gets smarter, "so the two [Pi and flies] could be more interdependent."

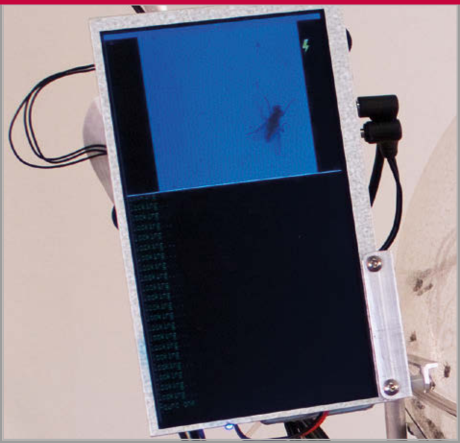


MAKE A FLY MACHINE



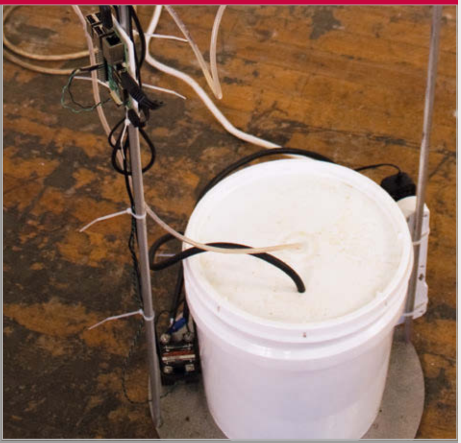
>STEP-01
Fly chamber

Made from two plastic hemispheres, the chamber was adapted with the addition of ports to create a nice airflow. The camera bracket was CNC-milled for a precise look and positioning.



>STEP-02
Data display

A standard LCD screen shows the captured still from the camera above the 'hand-typed' image recognition results, which are also read out aloud via a speaker at the rear.



>STEP-03
Feed the flies

Running all the software, the Raspberry Pi 3 has two GPIO pins connected to a solid-state relay. This triggers a water pump to send sugar water up the tube and into the fly chamber.

GET HELP FROM THE COMMAND LINE

The basic commands you need to get advice from the Linux command line

In our beginner's guide to the command line (*The MagPi* #54), we looked briefly at 'man', the manual you can access from the command line.

The man tool is so important that we think it deserves a more thorough explanation. And man isn't alone in offering help on the command line. Other commands like **`whatis`**, **`info`**, and **`apropos`** all offer support and assistance. And let's face it, support and assistance are what you will often need at the command line.

Even seasoned coders don't always know the correct command to type into the Linux terminal. This guide is all about the various ways to get help at the command line, so no matter what command you come across, you'll be able to find out more information on how to use it.

```
man
```

Your first point of call for getting help on the command line is man (short for 'manual'). Enter **`man`** followed by the name of a command to get detailed information about it. For instance, enter:

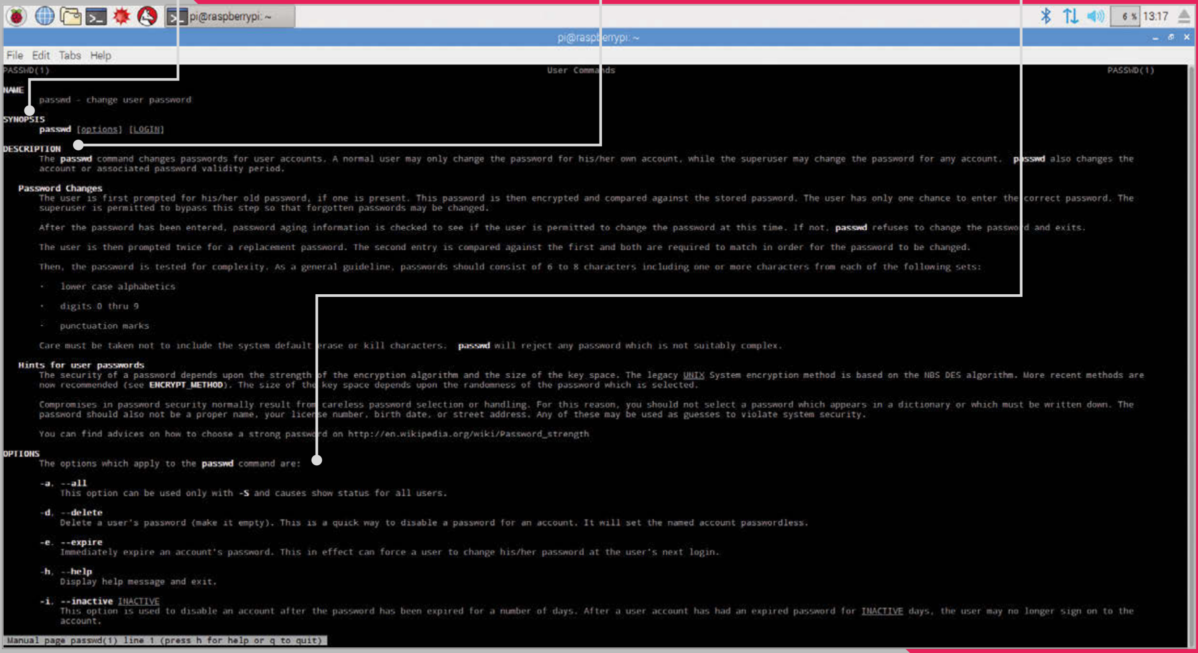
```
man passwd
```

...and you will see detailed information about the tool used to change your password. Man screens are displayed one page at a time. Press the **SPACE** bar to

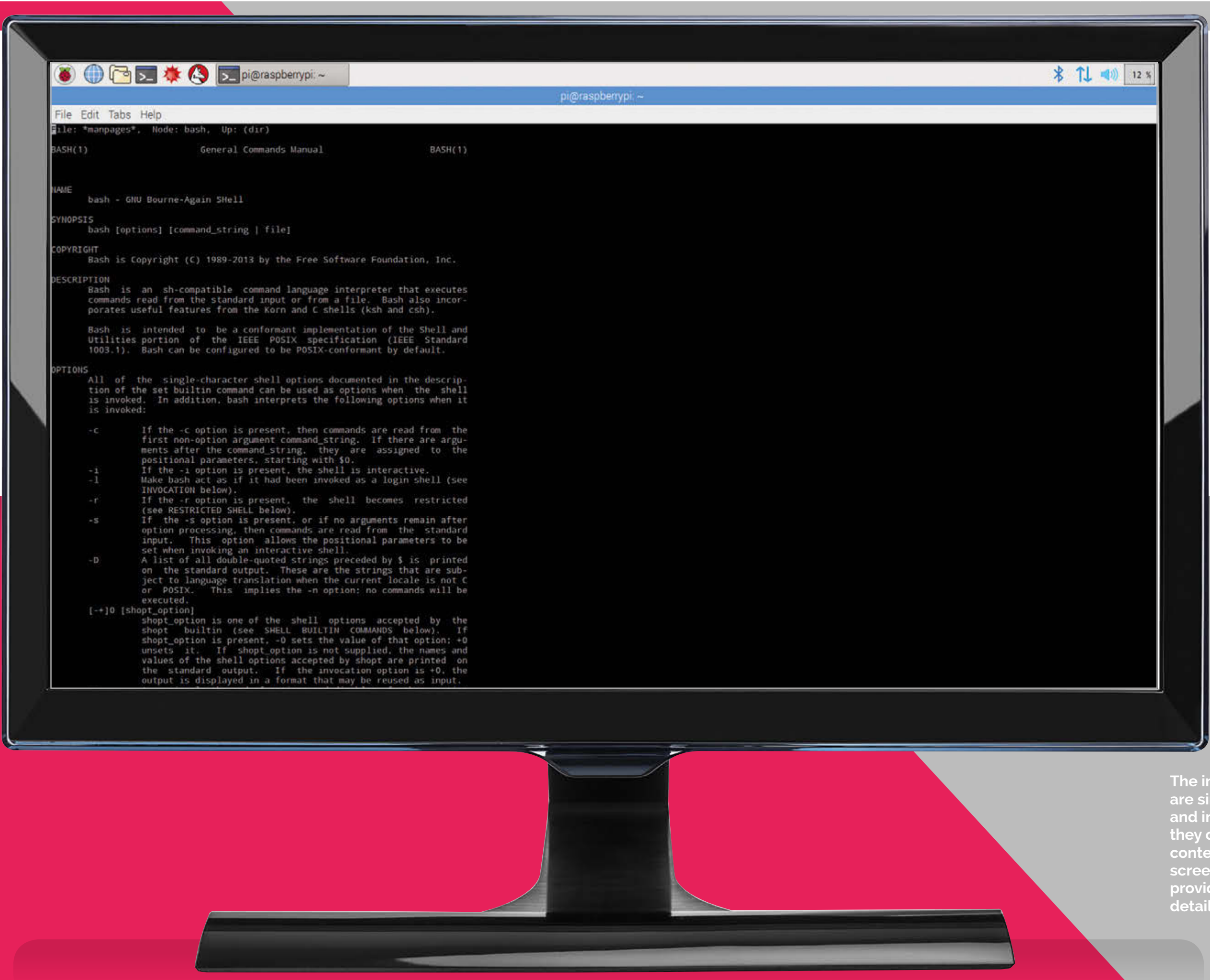
The synopsis gives a brief outline of how to use the command. In this case, you need to enter the command, an option, and an account name. The parts in brackets are optional

The description gives a detailed outline of the tool. It also offers information on how it works

Many commands have options, typically a hyphen followed by a letter (or double-hyphen followed by a word). You'll find each option outlined in detail in the man page



```
PASSWD(1)                                  User Commands                                  PASSWD(1)

NAME
       passwd - change user password

SYNOPSIS
       passwd [options] [LOGIN]

DESCRIPTION
       The passwd command changes passwords for user accounts. A normal user may only change the password for his/her own account, while the superuser may change the password for any account.  passwd also changes the
       account or associated password validity period.

   Password Changes
       The user is first prompted for his/her old password, if one is present. This password is then encrypted and compared against the stored password. The user has only one chance to enter the correct password. The
       superuser is permitted to bypass this step so that forgotten passwords may be changed.

       After the password has been entered, password aging information is checked to see if the user is permitted to change the password at this time. If not, passwd refuses to change the password and exits.

       The user is then prompted twice for a replacement password. The second entry is compared against the first and both are required to match in order for the password to be changed.

       Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to 8 characters including one or more characters from each of the following sets:

       ·   lower case alphabetics

       ·   digits 0 thru 9

       ·   punctuation marks

       Care must be taken not to include the system default erase or kill characters.  passwd will reject any password which is not suitably complex.

   Hints for user passwords
       The security of a password depends upon the strength of the encryption algorithm and the size of the key space. The legacy UNIX System encryption method is based on the NBS DES algorithm. More recent methods are
       now recommended (see ENCRYPT_METHOD). The size of the key space depends upon the randomness of the password which is selected.

       Compromises in password security normally result from careless password selection or handling. For this reason, you should not select a password which appears in a dictionary or which must be written down. The
       password should also not be a proper name, your license number, birth date, or street address. Any of these may be used as guesses to violate system security.

       You can find advices on how to choose a strong password on http://en.wikipedia.org/wiki/Password_strength

OPTIONS
       The options which apply to the passwd command are:

       -a, --all
           This option can be used only with -S and causes show status for all users.

       -d, --delete
           Delete a user's password (make it empty). This is a quick way to disable a password for an account. It will set the named account passwordless.

       -e, --expire
           Immediately expire an account's password. This in effect can force a user to change his/her password at the user's next login.

       -h, --help
           Display help message and exit.

       -i, --inactive INACTIVE
           This option is used to disable an account after the password has been expired for a number of days. After a user account has had an expired password for INACTIVE days, the user may no longer sign on to the
           account.
 Manual page passwd(1) line 1 (press h for help or q to quit)
```



```
File: *manpages*,  Node: bash,  Up: (dir)

BASH(1)                   General Commands Manual                  BASH(1)



NAME
       bash - GNU Bourne-Again SHell

SYNOPSIS
       bash [options] [command_string | file]

COPYRIGHT
       Bash is Copyright (C) 1989-2013 by the Free Software Foundation, Inc.

DESCRIPTION
       Bash  is  an  sh-compatible  command language interpreter that executes
       commands read from the standard input or from a file.  Bash also incor-
       porates useful features from the Korn and C shells (ksh and csh).

       Bash  is  intended  to  be a conformant implementation of the Shell and
       Utilities portion  of  the  IEEE  POSIX  specification  (IEEE  Standard
       1003.1).  Bash can be configured to be POSIX-conformant by default.

OPTIONS
       All  of  the  single-character shell options documented in the descrip-
       tion of the set builtin command can be used as options when  the  shell
       is invoked.  In addition, bash interprets the following options when it
       is invoked:

       -c        If the -c option is present, then commands are read from  the
                 first non-option argument command_string.  If there are argu-
                 ments after the command_string,  they  are  assigned  to  the
                 positional parameters, starting with $0.
       -i        If the -i option is present, the shell is interactive.
       -l        Make bash act as if it had been invoked as a login shell (see
                 INVOCATION below).
       -r        If the -r option is present,  the  shell  becomes  restricted
                 (see RESTRICTED SHELL below).
       -s        If  the -s option is present, or if no arguments remain after
                 option processing, then commands are read from  the  standard
                 input.   This  option  allows the positional parameters to be
                 set when invoking an interactive shell.
       -D        A list of all double-quoted strings preceded by $ is  printed
                 on  the standard output.  These are the strings that are sub-
                 ject to language translation when the current locale is not C
                 or  POSIX.   This  implies the -n option; no commands will be
                 executed.
       [-+]O [shopt_option]
                 shopt_option is one of the  shell  options  accepted  by  the
                 shopt   builtin  (see  SHELL  BUILTIN  COMMANDS  below).   If
                 shopt_option is present, -O sets the value of that option; +O
                 unsets  it.   If  shopt_option is not supplied, the names and
                 values of the shell options accepted by shopt are printed  on
                 the  standard  output.   If  the invocation option is +O, the
                 output is displayed in a format that may be reused as input.
```

The info screens are similar to man, and in many cases, they offer the same content. But some info screens (like bash) provide much more detailed content

move to the next page, and press **Q** to exit the page and return to the command prompt.

Man pages can be a bit tricky to read at first, but you'll soon get the hang of it.

At the top are the Name, Synopsis, and Description sections. Read these to get an overview of the command. Below them you'll find options and parameters; read these carefully to discover ways to expand your usage of each command. It's a good idea to use man on any commands you know, and read the manual for any new Linux commands you come across.

You can even read a man page for man:

```
man man
```

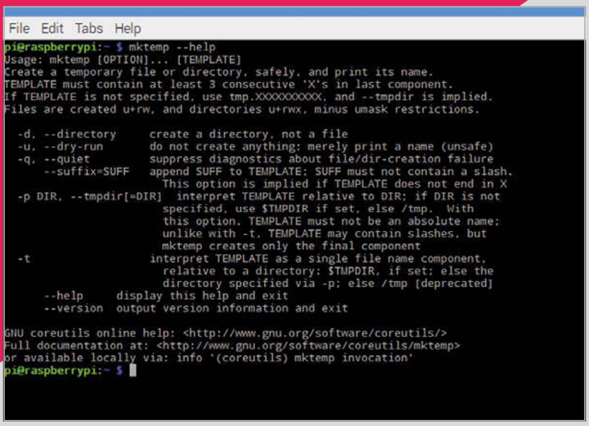
Press **H** in the man screen to view a summary of navigational key presses. These are worth learning so you can do more than press space to move to the next page.

Man's lesser-known partner is 'info', which is used to display information pages associated with commands. Sometimes these are the same as the man pages. In other cases they provide a different description. Try these:

```
man bash
info bash
```

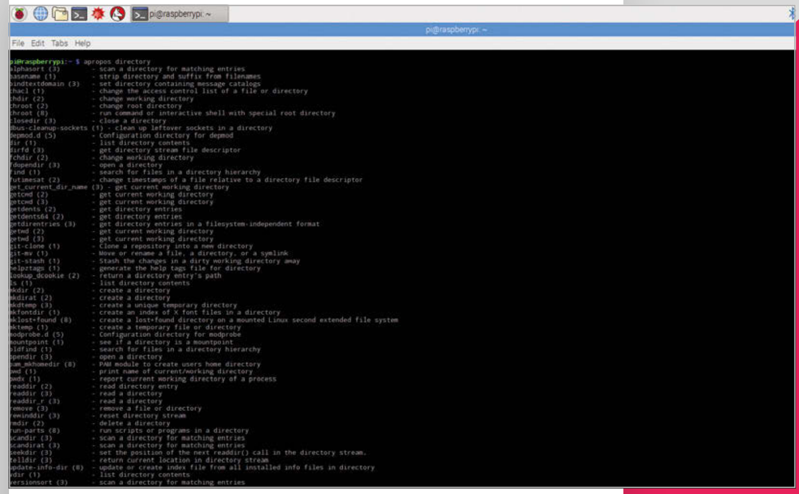
While `man bash` gives you a brief description of the GNU Bourne-Again Shell and the options used with the bash command; `info bash` gives the whole history and hundreds of pages of detailed information.

Press **H** on an info screen to view the controls for navigating such long documents. As well as **SPACE** to move down, you use **DELETE** to go back a screen, **TAB** to highlight links, and **RETURN** to use them. Press **Q** to exit the help screen.



```
pi@raspberrypi:~ $ mktemp --help
Usage: mktemp [OPTION]... [TEMPLATE]
Create a temporary file or directory, safely, and print its name.
TEMPLATE must contain at least 3 consecutive 'X's in last component.
If TEMPLATE is not specified, use tmp.XXXXXXXXXX, and --tmpdir is implied.
Files are created u+rw, and directories u+rwx, minus umask restrictions.

  -d, --directory     create a directory, not a file
  -u, --dry-run       do not create anything; merely print a name (unsafe)
  -q, --quiet         suppress diagnostics about file/dir-creation failure
      --suffix=SUFF   append SUFF to TEMPLATE; SUFF must not contain a slash.
                        This option is implied if TEMPLATE does not end in X
  -p DIR, --tmpdir[=DIR]  interpret TEMPLATE relative to DIR; if DIR is not
                        specified, use $TMPDIR if set, else /tmp.  With
                        this option, TEMPLATE must not be an absolute name;
                        unlike with -t, TEMPLATE may contain slashes, but
                        mktemp creates only the final component
  -t                  interpret TEMPLATE as a single file name component,
                        relative to a directory: $TMPDIR, if set; else the
                        directory specified via -p; else /tmp [deprecated]
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/mktemp>
or available locally via: info '(coreutils) mktemp invocation'
pi@raspberrypi:~ $ █
```

Many commands feature a built-in help option, accessed with `-h` or `--help`. Using it offers a brief outline



```
pi@raspberrypi:~ $ apropos directory
alphasort (3)         - scan a directory for matching entries
basename (1)          - strip directory and suffix from filenames
bindtextdomain (3)    - set directory containing message catalogs
chacl (1)             - change the access control list of a file or directory
chdir (2)             - change working directory
chroot (2)            - change root directory
chroot (8)            - run command or interactive shell with special root directory
closedir (3)          - close a directory
dbus-cleanup-sockets (1) - clean up leftover sockets in a directory
depmod.d (5)          - Configuration directory for depmod
dir (1)               - list directory contents
dirfd (3)             - get directory stream file descriptor
fchdir (2)            - change working directory
fdopendir (3)         - open a directory
find (1)              - search for files in a directory hierarchy
futimesat (2)         - change timestamps of a file relative to a directory file descriptor
get_current_dir_name (3) - get current working directory
getcwd (2)            - get current working directory
getcwd (3)            - get current working directory
getdents (2)          - get directory entries
getdents64 (2)        - get directory entries
getdirentries (3)     - get directory entries in a filesystem-independent format
getwd (2)             - get current working directory
getwd (3)             - get current working directory
git-clone (1)         - Clone a repository into a new directory
git-mv (1)            - Move or rename a file, a directory, or a symlink
git-stash (1)         - Stash the changes in a dirty working directory away
helpztags (1)         - generate the help tags file for directory
lookup_dcookie (2)    - return a directory entry's path
ls (1)                - list directory contents
mkdir (2)             - create a directory
mkdirat (2)           - create a directory
mkdtemp (3)           - create a unique temporary directory
mkfontdir (1)         - create an index of X font files in a directory
mklost+found (8)      - create a lost+found directory on a mounted Linux second extended file system
mktemp (1)            - create a temporary file or directory
modprobe.d (5)        - Configuration directory for modprobe
mountpoint (1)        - see if a directory is a mountpoint
oldfind (1)           - search for files in a directory hierarchy
opendir (3)           - open a directory
pam_mkhomedir (8)     - PAM module to create users home directory
pwd (1)               - print name of current/working directory
pwdx (1)              - report current working directory of a process
readdir (2)           - read directory entry
readdir (3)           - read a directory
readdir_r (3)         - read a directory
remove (3)            - remove a file or directory
rewinddir (3)         - reset directory stream
rmdir (2)             - delete a directory
run-parts (8)         - run scripts or programs in a directory
scandir (3)           - scan a directory for matching entries
scandirat (3)         - scan a directory for matching entries
seekdir (3)           - set the position of the next readdir() call in the directory stream.
telldir (3)           - return current location in directory stream
update-info-dir (8)   - update or create index file from all installed info files in directory
vdir (1)              - list directory contents
versionsort (3)       - scan a directory for matching entries
```

Apropos is used to search the manual for matching words. With it, you can find commands based on subjects, such as directory, password, or links

FINDING COMMANDS

As you become more familiar with man and info, you'll start searching for commands to look up. Here, the **man -k** command comes in useful. In particular, try this:

```
man -k directory | more
```

This command lists all available man entries. Press **SPACE** to run through them one at a time. The **man -k** option is worth remembering. If you use **man man**, it tells you the **-k** option is 'equivalent to apropos'. Apropos is used to search manual page names and descriptions. It's a handy way to find commands when you don't know their names.

For instance, enter:

```
apropos directory
```

...and you'll get a list of all the commands that have the word 'directory' in their description or page name. Here you'll find common commands such as **ls**, **cd**, and **pwd**, but you'll also find less obvious commands, such as **mktemp**.

Next to each command is a number, like (1) or (2). These correspond to the section numbers of the manual (view using **man man**).

The section numbers are useful for guiding you to the commands that can be used on the command line. As a general rule, **1:** Executable programs or shell commands, and **2:** System calls, both tend to be worth investigating. Higher numbers are for library calls, special files, and kernel routines for advanced users.

You can find out more information about any command using man:

```
man mktemp
```

This command gives you detailed information on how to create temporary directories.

TAB AUTOCOMPLETE

Another way to find files is to use 'tab autocomplete'. By pressing the **TAB** key, you can automatically complete commands, files, and directories on the command line. If you're not doing so already, learn to press **TAB** a lot on the command line: it's a good way to discover new commands.

Take the **apt** tool, for example. There are **apt-get** and **apt-cache**, but did you know about **apt-config** and **apt-key**?

Enter:

```
apt
```

And press the **TAB** key twice. It will display all the different types of apt available.

You can even run through the letters of the alphabet. Enter the letter 'a':

```
a
```

And press **TAB** twice to view all the commands beginning with 'a'. You can then use man to look up commands. It's a great way to broaden your knowledge of the command line.

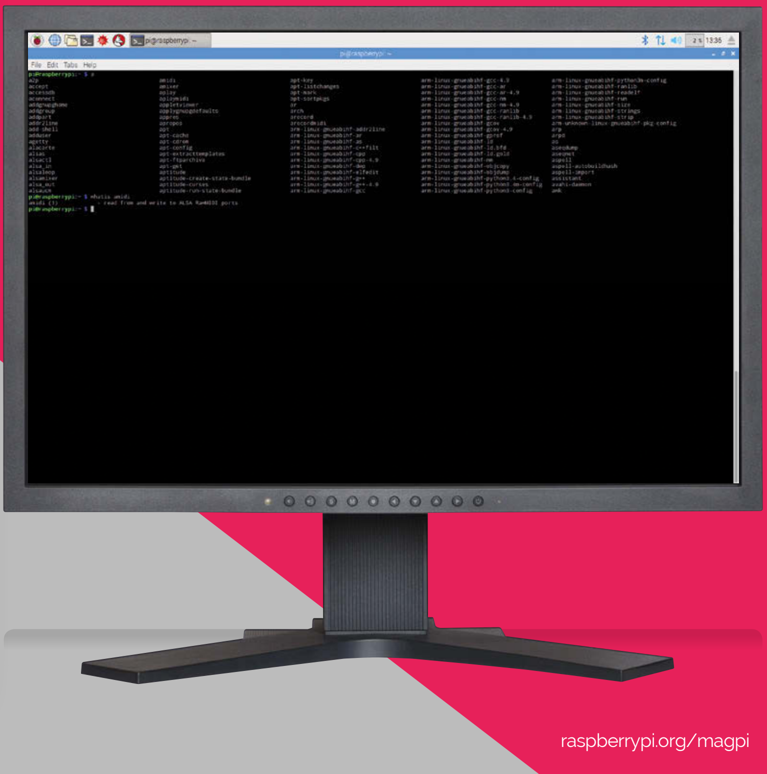
EXPRESS HELP

Many commands also offer a help feature as an option. Help is typically accessed using **-h** or **--help**:

```
mktemp --help
```

This command displays the options offered by the mktemp command. It's the same as the first page of mktemp's man file, but saves you digging in and out of the full document.

Enter a single letter and press TAB twice to view all the commands available in that letter. Then use whatis to get a short one-line description of each command



```
pi@raspberrypi:~ $ a
a2p                  amidi                         apt-key                              arm-linux-gnueabihf-gcc-4.9            arm-linux-gnueabihf-python3m-config
accept               amixer                        apt-listchanges                      arm-linux-gnueabihf-gcc-ar             arm-linux-gnueabihf-ranlib
accessdb             aplay                         apt-mark                             arm-linux-gnueabihf-gcc-ar-4.9         arm-linux-gnueabihf-readelf
aconnect             aplaymidi                     apt-sortpkgs                         arm-linux-gnueabihf-gcc-nm             arm-linux-gnueabihf-run
addgnupghome         appletviewer                  ar                                   arm-linux-gnueabihf-gcc-nm-4.9         arm-linux-gnueabihf-size
addgroup             applygnupgdefaults            arch                                 arm-linux-gnueabihf-gcc-ranlib         arm-linux-gnueabihf-strings
addpart              appres                        arecord                              arm-linux-gnueabihf-gcc-ranlib-4.9     arm-linux-gnueabihf-strip
addr2line            apropos                       arecordmidi                          arm-linux-gnueabihf-gcov               arm-unknown-linux-gnueabihf-pkg-config
add-shell            apt                           arm-linux-gnueabihf-addr2line        arm-linux-gnueabihf-gcov-4.9           arp
adduser              apt-cache                     arm-linux-gnueabihf-ar               arm-linux-gnueabihf-gprof              arpd
agetty               apt-cdrom                     arm-linux-gnueabihf-as               arm-linux-gnueabihf-ld                 as
alacarte             apt-config                    arm-linux-gnueabihf-c++filt          arm-linux-gnueabihf-ld.bfd             aseqdump
alias                apt-extracttemplates          arm-linux-gnueabihf-cpp              arm-linux-gnueabihf-ld.gold            aseqnet
alsactl              apt-ftparchive                arm-linux-gnueabihf-cpp-4.9          arm-linux-gnueabihf-nm                 aspell
alsa_in              apt-get                       arm-linux-gnueabihf-dwp              arm-linux-gnueabihf-objcopy            aspell-autobuildhash
alsaloop             aptitude                      arm-linux-gnueabihf-elfedit          arm-linux-gnueabihf-objdump            aspell-import
alsamixer            aptitude-create-state-bundle  arm-linux-gnueabihf-g++              arm-linux-gnueabihf-python3.4-config   assistant
alsa_out             aptitude-curses               arm-linux-gnueabihf-g++-4.9          arm-linux-gnueabihf-python3.4m-config  avahi-daemon
alsaucm              aptitude-run-state-bundle     arm-linux-gnueabihf-gcc              arm-linux-gnueabihf-python3-config     awk
pi@raspberrypi:~ $ whatis amidi
amidi (1)            - read from and write to ALSA RawMIDI ports
pi@raspberrypi:~ $
```

Not all commands make use of **--help**. Some, like **ls --help**, display the full man document (you can pipe this through **less**:

```
ls --help | less
```

...but it's typically easier to use **man ls**. Some commands don't implement the help option at all.

```
pwd --help
```

...returns 'invalid option'. But it's worth trying when you are experimenting with new commands.

One final command worth using when searching for commands is whatis:

```
whatis pwd
```

This example returns 'print name of current/working directory'. Often, this brief description is enough to let you know what it does, or at least tell you if it's something you'd like to investigate further with man or info.

These are just some of the tools you can use to get help at the command line in Linux. While the command line may seem intimidating at first, you're far from alone in this text-only environment.

WEB SEARCH

One of the advantages when using a desktop interface, like Raspbian, is that a web browser – and a search engine – is just a click away.

Getting online from the command line is a lot easier than you'd imagine. There are many different text-based web browsers that enable you to access Google, Bing, DuckDuckGo, and other websites without having to boot into the PIXEL desktop interface.

We're going to use:

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

Now you can open the web browser from the command line using:

```
elinks
```

The elinks interface is full-screen, so it replaces the command line. Press **g** to open a URL field. You can enter full URLs, such **http://www.google.com** or just shortened versions, such as **raspberrypi.com**.

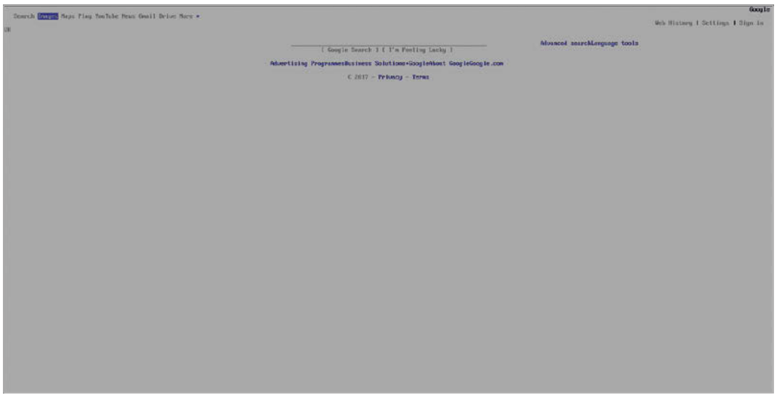
Better yet, there are a few key bindings for helpful sites. Press **g** then enter these shortcuts:

d - dict.org search

sd - Slashdot

g - Google search

You can also enter Google search terms in the URL field. Press **g**, then enter 'g the magpi' to search for our website in Google.


Search Images Maps Play YouTube News Gmail Drive More ▾
Google
Web History | Settings | Sign in
UK
Advanced searchLanguage tools
[Google Search] [I'm Feeling Lucky]
Advertising ProgrammesBusiness Solutions+GoogleAbout GoogleGoogle.com
© 2017 - Privacy - Terms

Other keyboard shortcuts can be used to navigate the program:

g - Goto URL

Down Arrow - Next link

Up Arrow - Previous link

Return - Select link

Left Arrow - Back

u - Forward

q - Quit

. - Toggle link numbering

% - Toggle colours

t - New tab

T - Open link in new tab

> - Next tab

< - Previous tab

c - Close tab

POWER YOUR RASPBERRY PI

Don't let your Raspberry Pi suffer from a shortage of volts

You'll Need

- Raspberry Pi
- Raspbian OS
- Power supply

Powering your Raspberry Pi is, on the surface, a remarkably simple affair. All Raspberry Pi boards are designed to use the same micro USB power socket as many smartphones. All you need is a spare USB adapter, and it will provide power to your Raspberry Pi.

That's the basic requirement. Of course, this being a board for makers and hackers, there's far more to it than that. Learning how to properly supply the right amount of power to your Raspberry Pi board is important as you start to create complex projects.

A 5V micro USB typically powers the Raspberry Pi. But how much current (in milliamps or amps) the Pi requires to function depends on your usage.

The recommended amount is between 700mA for a Raspberry Pi Model A, and up to 2.5A for a Raspberry Pi 3 Model B (see 'Power Supply Requirements' box).

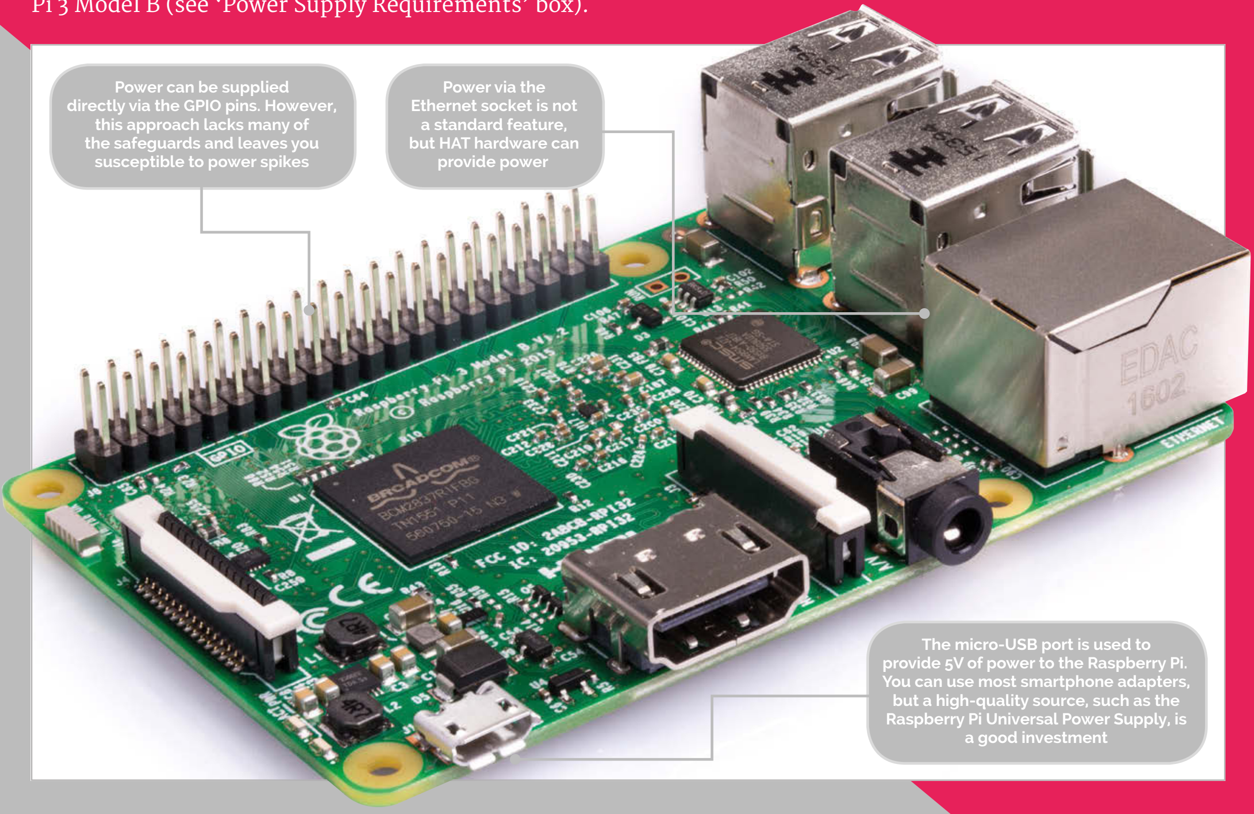
The Raspberry Pi boards typically draw much lower amounts, between 200 and 500mA.

Usage depends on what you're doing with the Pi. Playing video and browsing the web draws more power than idling and booting. It also depends on what devices you have connected; some keyboards and mice draw more power than others.

Low power warning

So, the Raspberry Pi can function on lower power supplies, but it may have problems when you start to do demanding tasks or add peripherals.

The Raspbian operating system comes with a low voltage indicator icon. This yellow lightning bolt appears in the top-right of the screen. It is used to indicate an under-voltage warning.



Power can be supplied directly via the GPIO pins. However, this approach lacks many of the safeguards and leaves you susceptible to power spikes

Power via the Ethernet socket is not a standard feature, but HAT hardware can provide power

The micro-USB port is used to provide 5V of power to the Raspberry Pi. You can use most smartphone adapters, but a high-quality source, such as the Raspberry Pi Universal Power Supply, is a good investment

The Raspberry Pi Universal Power Supply is a reliable source of power for your board

Drawing too much power from the Raspberry Pi board isn't dangerous, but it can cause erratic behaviour. If you want to attach several performance-heavy peripherals to your Raspberry Pi, it's best to use a powered USB hub.

Official supply

The Raspberry Pi Universal Power Supply (**magpi.cc/2a14pye**) is a dependable power source. It'll keep feeding your Pi the steady 2A it needs for proper performance. If you're worried about getting stable power, then it's a good idea to invest in a suitable adapter.

Power via GPIO

A more technical (and dangerous) way to power the Raspberry Pi is via the GPIO pins.

The 5V GPIO pins on a Raspberry Pi are connected to the 5V rail. Typically they provide the remaining power from the Raspberry Pi (that isn't being used to power the board itself). So you can hook up the GPIO 5V pins to a 5V power source and feed power directly to the board.

```
Connect a 5V source to Pin #2 (5V).
Connect the ground of that source to Pin #6 (GND).
```

Please be aware that there is no regulation or fuse protection on the GPIO to protect from over-voltage or current spikes.

If an incorrect voltage is applied, or a current spike occurs on the line, you can permanently damage your Raspberry Pi.

Power Usage

	Pi 1 (B+)	Pi 2 B	Pi 3 B	Zero
Boot	Max 0.26A	0.40A	0.75A	0.20A
	Avg 0.22A	0.22A	0.35A	0.15A
Idle	Avg 0.20A	0.22A	0.3a0A	0.10A
Video playback (H.264)	Max 0.30A	0.36A	0.55A	0.23A
	Avg 0.22A	0.28A	0.33A	0.16A
Stress	Max 0.35A	0.82A	1.34A	0.35A
	Avg 0.32A	0.75A	0.85A	0.23A

Test conditions used a standard Raspbian image (26 Feb 2016), at room temperature, connected to an HDMI monitor, USB keyboard, and mouse. For the Model 3B it was connected to a WiFi access point. All these power measurements do not take into account power consumption from additional USB devices; these measurements can easily be exceeded with multiple additional USB devices connected or when using a HAT.

Power via HAT

If you're looking to power the Raspberry Pi via the GPIO pins, then it's best to go via a HAT. The hardware sits on top of the GPIO pins and adds the safety features you need.

Power over Ethernet

Supplying power to your Raspberry Pi via the Ethernet cable is an attractive proposition, but it's not provided as standard. If you're interested in providing power to your Raspberry Pi via a network cable, then take a look at the Pi PoE Switch HAT (**magpi.cc/2lhnDDT**).

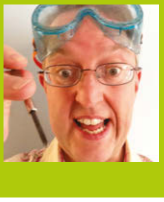
Mobile power

It is possible to provide power to the device from a battery pack designed for mobile phone charging. Companies such as Anker (**anker.com**) and Poweradd (**ipoweradd.com**) are known for their mobile charging solutions.

But if you want something a little more serious, we reviewed the Zero LiPo in issue 51 and found it a smart solution to portable power. It supplies power from lithium batteries via the GPIO pins with safety management features (**magpi.cc/2coJyZ2**).

Power Supply Requirements

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	~400mA



PHIL KING

When not sub-editing *The MagPi* and writing articles, Phil loves to work on Pi projects, including wheeled robots.

@philking68

CREATE AN LED THERMOMETER

Read a temperature sensor and display its value as a bar graph

You'll Need

- GPIO Zero Python library
- 1× solderless breadboard
- 1× MCP3008 ADC chip
- 1× TMP36 temperature sensor
- 5× LEDs (red, yellow, green)
- 5× 330Ω resistors
- 1× 1uF capacitor
- 11× male-to-female jumper wires
- 8× male-to-male jumper wires

Analogue readings on the Pi require an analogue-to-digital converter (ADC) chip such as the MCP3008. It's a pretty popular ADC, so like many other things you'll find it in the GPIO Zero Python library. We'll use one to display the current temperature on the screen, then add some LEDs and use GPIO Zero's handy **LEDBarGraph** class to get them to light up according to the temperature.

>STEP-01

Enable SPI

The analogue values from the ADC chip will be communicated to the Pi using the SPI protocol. While this will work in GPIO Zero out of the box, you may get better results if you enable full SPI support. Open a terminal window and enter:

```
  sudo apt-get install python3-spidev
python-spidev
```

Click OK and reboot the Pi.

>STEP-02

Connect the ADC

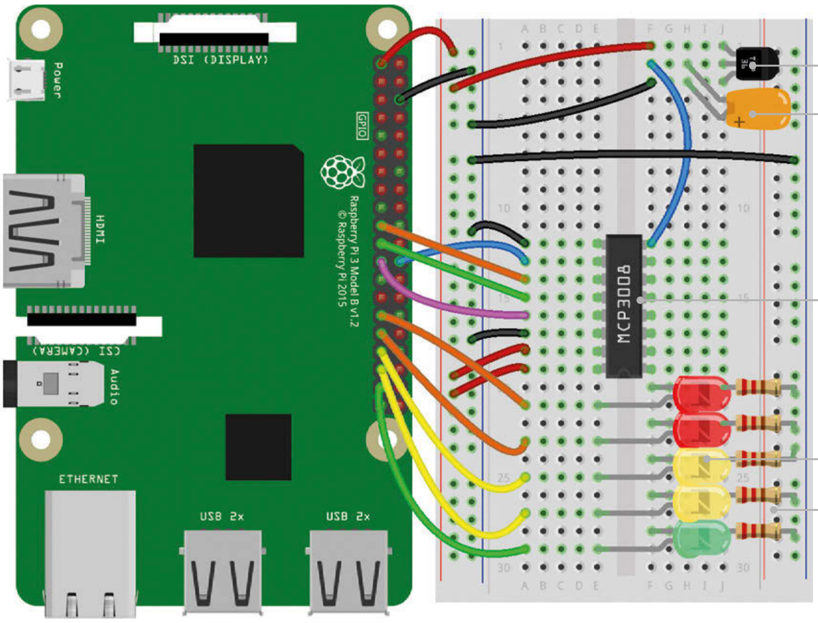
It's best practice to turn off the Pi while creating our circuit. As you can see from the diagram below, there's quite a lot of wiring required to connect the MCP3008 ADC to the Pi's GPIO pins.

First, place the MCP3008 in the middle of the breadboard, straddling its central groove. Now connect the jumper wires as in the diagram. Two go to the '+' power rail, connected to a 3V3 pin; two others are connected to a GND pin via the '–' rail. The four middle legs of the ADC are connected to GPIO pins 8 (CE0), 10 (MOSI), 9 (MISO), and 11 (SCLK).

>STEP-03

Add the sensor

Now that the ADC is connected to the Pi, you can wire devices up to its eight input channels, numbered 0 to 7. Here, we'll connect a TMP36 analogue temperature sensor. It's vital that this is wired up correctly, otherwise it'll overheat. With its flat face towards



It's vital to get the wiring correct for the TMP36 sensor, otherwise it will quickly overheat

Link the sensor's ground and output pins with a capacitor to help stabilise its readings

The MCP3008 ADC chip straddles the central groove; the side shown without wires comprises eight input channels

We use five LEDs to display a bar graph relating to the temperature; each LED is connected to a different GPIO pin

Each LED's shorter leg is connected to the ground rail via a resistor, to limit the current you, the left-hand leg is for power, so connect this to the '+' power rail. The right-hand leg is connected to the '–' ground rail. Its middle leg is the output; here we're connecting to channel 7 (the nearest one) of the MCP3008. Finally, to help stabilise the readings which might otherwise be erratic, we'll add a capacitor to link its output and ground legs.

>STEP-04

Take the temperature

We can now read the sensor's value in Python. On the Pi, open IDLE from the main menu: Menu > Programming > Python 3 (IDLE). Create a new file, enter the code from **ch10listing1.py**, and save it. At the top, import the **MCP3008** class from GPIO Zero, then the **sleep** function from the **time** library. Next, we define a function that converts the sensor reading into degrees Celsius. We then assign the **adc** variable to channel 7 of the MCP3008. Finally, we use a **for** loop to display the converted temperature on the screen, updating it every second. Note: if you've just been handling the sensor, it might take a little while to settle down.

>STEP-05

LED bar graph

Next, we'll add our line of five LEDs to the circuit, as in the diagram. From green to red, we've connected their longer legs to the following GPIO pins: 26, 19, 13, 6, and 5. In IDLE, create a new file, enter the code from **ch10listing2.py** , and save it. At the start, we import the **LEDBarGraph** class from GPIO Zero; this will enable us to use the LEDs to display a bar graph, saving a lot of complex coding. We assign the **graph** variable to our LEDs on the GPIO pins mentioned previously, and also enable PWM so that we can adjust their brightness for a more accurate display. We then set **graph.value** to various fractions between 0 and 1 to light the relevant number of LEDs from green to red, including partially lit ones for precision. Note that if the value is negative, it will light the LEDs from the other end, red ones first.

>STEP-06

Display the temperature

So, we've got our temperature sensor and LED bar graph set up. Let's combine them to display the temperature on the LED bar graph. In IDLE, create a new file, enter the code from **ch10listing 3.py**, and save it. At the top, we import GPIO Zero's **MCP3008** and **LEDBarGraph** classes, along with the **sleep** function from the **time** library. We then define a function to convert the temperature sensor's readings to degrees Celsius. We assign the **adc** variable to channel 7 of the MCP3008 and **graph** to our LEDs' GPIO pins, setting PWM to **true**. Finally, in our **for** loop, we add a **bars** variable to determine how many LEDs are lit in the bar graph. In this example, we've divided **temp** by 35, which is around the maximum temperature for the UK, so if it gets to 35°C, all the LEDs should light up fully. Naturally, you can adjust this number to suit your own location's climate. When ready, run the code and see those LEDs light up to show the current temperature.

Language

>PYTHON 3

DOWNLOAD:
magpi.cc/2bhwQbJ

ch10listing1.py

```python
from gpiozero import MCP3008
from time import sleep

def convert_temp(gen):
  for value in gen:
    yield (value * 3.3 - 0.5) * 100

adc = MCP3008(channel=7)

for temp in convert_temp(adc.values):
  print("The temperature is", temp, "C")
  sleep(1)
```

ch10listing2.py

```python
from gpiozero import LEDBarGraph
from time import sleep

graph = LEDBarGraph (26, 19, 13, 6, 5, pwm=True)

graph.value = 1/10
sleep(1)
graph.value = 3/10
sleep(1)
graph.value = -3/10
sleep(1)
graph.value = 9/10
sleep(1)
graph.value = 95/100
sleep(1)
graph.value = 0
```

ch10listing3.py

```python
from gpiozero import MCP3008, LEDBarGraph
from time import sleep

def convert_temp(gen):
  for value in gen:
    yield (value * 3.3 - 0.5) * 100

adc = MCP3008(channel=7)
graph = LEDBarGraph (26, 19, 13, 6, 5, pwm=True)

for temp in convert_temp(adc.values):
  bars = temp / 35
  graph.value = bars
  sleep(1)
```

SIMON LONG

Works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves crosswords.

raspberrypi.org

AN INTRODUCTION TO C PART 10
FILE INPUT AND OUTPUT

Reading from and writing to files

In last issue's tutorial, we looked at how to get input from the user at the console. In this instalment, we'll look at the other common method of input and output in C; reading and writing files.

Many programs need to be able to access files on the computer's disk.

In C, files are accessed by use of *file pointers*, which contain all the information required to access a file – its name and location on the disk, and a current position within the file at which data will be read or written.

So the first thing we need to do is to get a file pointer. This is done using the C function **fopen**, which takes two arguments. The first argument is the path to the file; the second is the *file access mode* – this indicates whether you intend to read from the file or write to it.

Here's an example of reading a file. Use your text editor to create a file called **input.txt** in the **/home/pi** directory, and type anything you like into it. Save it, and then create and run the following program:

```
#include <stdio.h>

void main (void)
{
  FILE *fp;
  int value;

  fp = fopen ("/home/pi/input.txt", "rb");
  if (fp)
  {
    while (1)
    {
      value = fgetc (fp);
      if (value == EOF) break;
      else printf ("%c", value);
    }
    fclose (fp);
  }
}
```

We create a file pointer using the command **fopen** (short for 'file open') – we open the file at **/home/pi/input.txt**, with the mode **rb**, for 'read binary'. This creates the file pointer and initialises it to the beginning of the file.

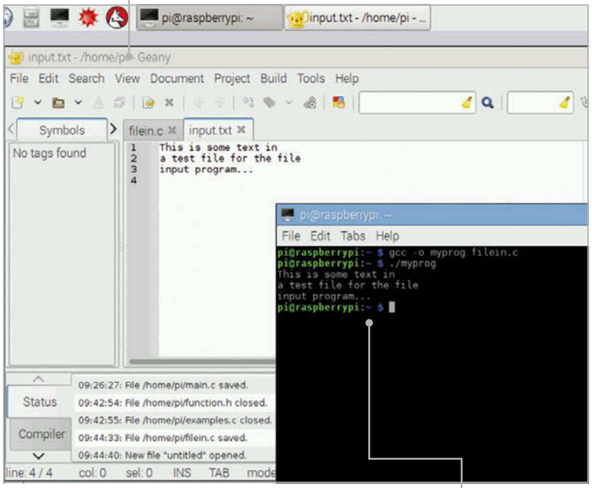
We then check to see if the file pointer is non-zero – if the pointer is returned as zero, the file wasn't successfully opened. (For a read, this usually indicates that the file doesn't exist.)

We then repeatedly call the function **fgetc** (for 'file get character') – each time this function is called, it reads a single byte from the file, and then advances the file pointer to the next byte. When the file pointer reaches the end of the file, it returns the special value **EOF** ('end of file'). So we print the value returned by **fgetc** each time until it returns **EOF**.

Once we have finished reading the file, we stop access to it by calling **fclose** (for 'file close').

Note that while **fgetc** reads characters, it returns an integer – this is because the code for **EOF** falls outside the valid range of a char variable (0-255). Unless at end of file, **fgetc** returns an integer value which can be treated as a char.

Inputted text is read from a special text file created for the task


pi@raspberrypi: ~
input.txt - /home/pi - ...
input.txt - /home/pi - Geany
File Edit Search View Document Project Build Tools Help
Symbols
No tags found
filein.c
input.txt
Status
Compiler
line: 4 / 4 col: 0 sel: 0 INS TAB mode
pi@raspberrypi: ~
File Edit Tabs Help

```
This is some text in
a test file for the file
input program...
```

```
pi@raspberrypi:~ $ gcc -o myprog filein.c
pi@raspberrypi:~ $ ./myprog
This is some text in
a test file for the file
input program...
pi@raspberrypi:~ $
```

```
09:26:27: File /home/pi/main.c saved.
09:42:54: File /home/pi/function.h closed.
09:42:55: File /home/pi/examples.c closed.
09:44:33: File /home/pi/filein.c saved.
09:44:40: New file "untitled" opened.
```

The information from the text file can then be read from the script, as well as changed

READ AND WRITE THE SAME FILE

You can open a file for simultaneous reading and writing with the same pointer. Set the file access mode to **rb+** to read an existing file and to overwrite it; set it to **wb+** to create a new file and be able to read back what you have written to it; set it to **ab+** to open a file to both append at the end and read from it.

CHECK THE LIBRARY

The C library offers a wide range of functions for reading and writing data from and to files; we've just looked at some of them. If you need to access a file, have a look at some of the other library functions – like **fread**, **fwrite**, **fscanf**, **fputs**, and **fgets** – to see if they are more appropriate than the basic **fputc** and **fgetc** we've used here.

Writing a file

To write to a file, we use a file pointer in exactly the same way, but we open it in a mode for writing.

```
#include <stdio.h>

void main (void)
{
  FILE *fp;
  int value;

  fp = fopen ("/home/pi/output.txt", "wb");

  if (fp)
  {
    for (value = 48; value < 58; value++)
    {
      fputc (value, fp);
    }
    fclose (fp);
  }
}
```

We open the file **/home/pi/output.txt** with the mode **wb**, for 'write binary'. We then call the function **fputc** (for 'file put character') in a loop, writing the bytes 48, 49…57 to the file. (These are the character codes for the digits 0, 1…9). We then close the file pointer.

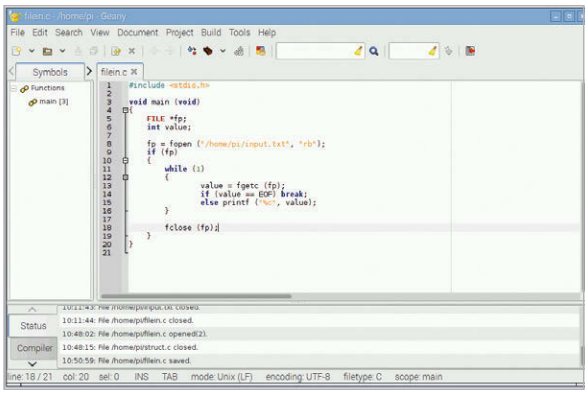
After running this, you should find the file **output.txt** in your home directory, containing the string **"0123456789"**.

Formatted output

fputc is useful for writing bytes to a file, but it's an inconvenient way of writing text to a file. For this, we can use the **fprintf** function (short for 'file print formatted').

```
#include <stdio.h>

void main (void)
{
  FILE *fp;

  fp = fopen ("/home/pi/output.txt", "wb");

  if (fp)
  {
    fprintf (fp, "This is some text.\n");
    fclose (fp);
  }
}
```

fprintf works in exactly the same way as **sprintf**, but the first argument is a file pointer rather than a string.


filein.c - /home/pi - Geany
File Edit Search View Document Project Build Tools Help
Symbols
filein.c
Functions
main [3]
Status
Compiler

```
#include <stdio.h>

void main (void)
{
    FILE *fp;
    int value;

    fp = fopen ("/home/pi/input.txt", "rb");
    if (fp)
    {
        while (1)
        {
                value = fgetc (fp);
                if (value == EOF) break;
                else printf ("%c", value);
        }

        fclose (fp);
    }
}
```

```
10:11:43: File /home/pi/input.txt closed.
10:11:44: File /home/pi/filein.c closed.
10:48:02: File /home/pi/filein.c opened(2).
10:48:15: File /home/pi/struct.c closed.
10:50:59: File /home/pi/filein.c saved.
```

line: 18 / 21 col: 20 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filetype: C scope: main

Above Example code and how it should look on Raspberry Pi

Moving around a file

To add to the end of an existing file, open it with the mode **ab** (for 'append binary'). If the file exists, output will then be added to the end of the file; if the file does not exist, it will be created and output will start at the beginning.

Sometimes when accessing a file, we don't want to start at the beginning. The **fseek** function is used for this.

```
#include <stdio.h>

void main (void)
{
  FILE *fp;
  int value;

  fp = fopen ("/home/pi/input.txt", "rb");
  if (fp)
  {
    fseek (fp, 10, SEEK_CUR);
    while (1)
    {
      value = fgetc (fp);
      if (value == EOF) break;
      else printf ("%c", value);
    }
    fclose (fp);
  }
}
```

fseek (fp, 10, SEEK_CUR) moves the file pointer 10 bytes ahead of the current position, so this program will print all but the first ten characters in the file. The first argument to **fseek** is the file pointer; the second is how many bytes to move the pointer. This can be positive or negative – so **fseek (fp, -5, SEEK_CUR)** moves the pointer 5 bytes back from the current position.

The third argument to **fseek** allows you to choose a position relative to the start of the file (**SEEK_SET**) or the end of the file (**SEEK_END**) rather than the current position (**SEEK_CUR**).

REMEMBER TO FCLOSE

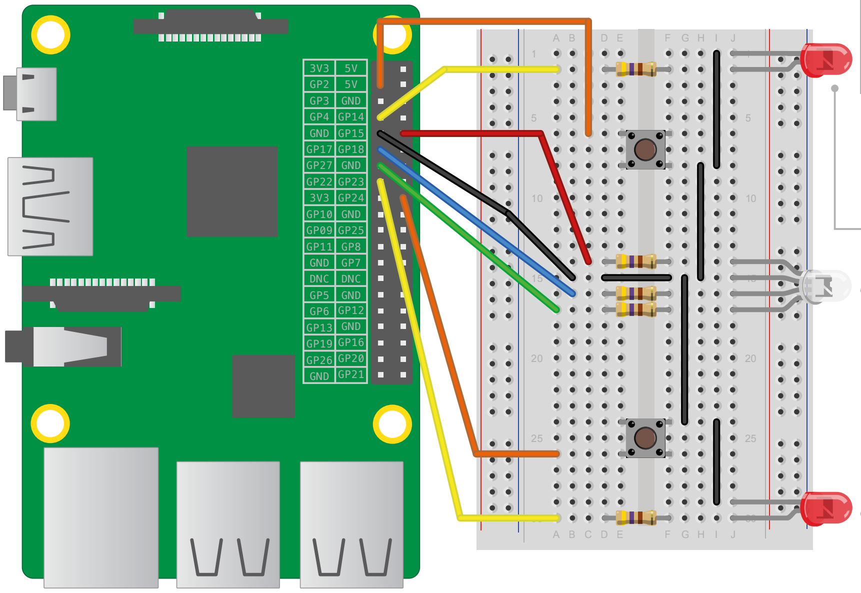
It's easy to forget to call **fclose** on your file, but it is important to do so. On some systems, when writing to the file system, the write doesn't actually complete until **fclose** is called – if your program doesn't call **fclose**, you might find that you write to files and nothing shows up in them.

ALWAYS CHECK YOUR FILE POINTER

Never assume that **fopen** has worked – always check that the value it returns is a valid pointer (i.e. not zero). If you try to read from a zero pointer, you'll get random nonsense; if you write to a zero pointer, you'll probably crash the system!

HENRY BUDDEN

Following the release of the Raspberry Pi when he was 12, he taught himself to code and use electronics, and has shared this process with the world.
magpi.cc/2eCbaMf



The red LEDs light up to indicate which player has the most points and is in the lead!

This RGB LED can create any colour in the visible light spectrum!

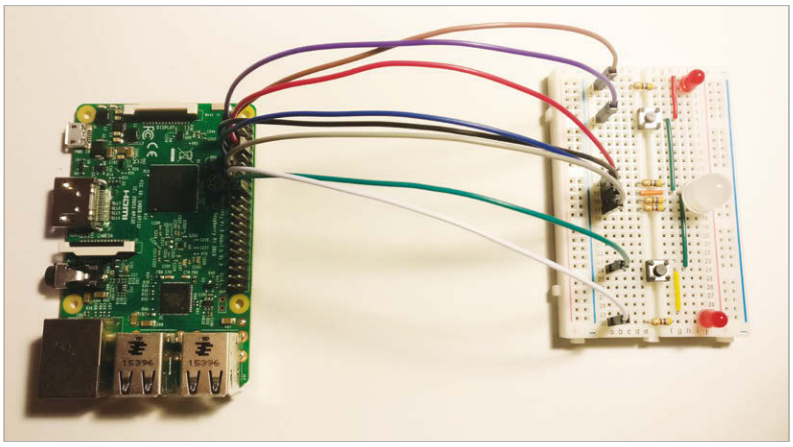
BUILD A REACTION GAME

Make a fast-paced reaction game and challenge your friends to find out who has the best ninja reflexes!

You'll Need

- 2× red LEDs
- RGB LED
- 2× Tactile push buttons
- 5× 470Ω resistors
- 8× Male-to-female jumper wires
- 4× Male-to-male jumper wires
- Alternatively, Monk Makes Raspberry Pi Electronics Starter Kit **magpi.cc/2lWlZHp**

Using a breadboard and some simple components, make a super-fun reaction game while learning how to use a breadboard to make an electronic circuit, as well as programming in Python to make your game even better! The game involves a race against an opponent, but there are a handful of trick signals thrown in for good measure, so don't get too complacent. The project is designed as an extension to the projects found in the Monk Makes Raspberry Pi Electronics Starter Kit.



Above It's a simple circuit, but you could easily turn it into a physical game with some 3D-printed parts

>STEP-01

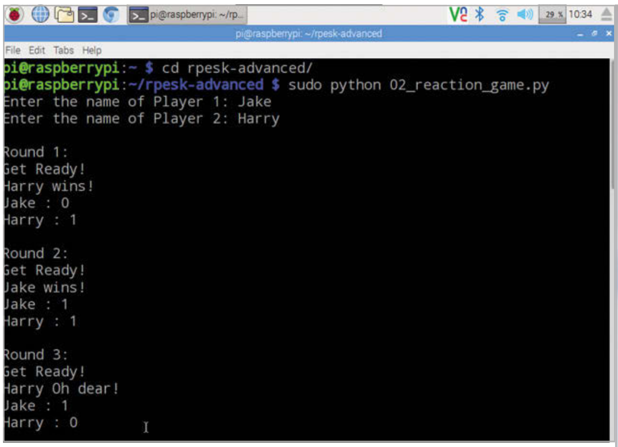
Connect the LEDs

Gather the components as listed on the left. Following the breadboard diagram, attach the two red LEDs to the breadboard, at either end. You must ensure that the long leg of the LED is facing down, on both LEDs. This is the positive leg, which will be connected to a positive output on the Raspberry Pi. The big RGB LED can be connected in the middle of the breadboard, on the right-hand side, as shown. There is one leg longer than the rest, which will be connected to the second hole from the top (shown by a red circle on the diagram). This is the negative leg.

>STEP-02

Add the resistors

In order for the LEDs to work correctly, a resistor needs to be added in series to each one, limiting the flow of current so they don't melt! You will need five resistors, as the RGB LED counts as three separate LEDs that share the same cathode (negative leg). Connect three resistors to the LED's positive legs, bridging the gap across the centre of the breadboard, as shown in the diagram. All of the pins in each row are connected to each other inside the breadboard, with a gap in the centre.



```
pi@raspberrypi:~ $ cd rpesk-advanced/
pi@raspberrypi:~/rpesk-advanced $ sudo python 02_reaction_game.py
Enter the name of Player 1: Jake
Enter the name of Player 2: Harry

Round 1:
Get Ready!
Harry wins!
Jake : 0
Harry : 1

Round 2:
Get Ready!
Jake wins!
Jake : 1
Harry : 1

Round 3:
Get Ready!
Harry Oh dear!
Jake : 1
Harry : 0
```

Above Current scores are displayed in the terminal window on the Raspberry Pi

>STEP-03

Add the buttons

This project needs two buttons: one for each player. Some small buttons have four legs, as shown in the diagram, and a connection is made between the diagonal pins when it is pressed. If you have a button with two legs, simply connect it to the breadboard with one leg on either side of the central gap, ensuring that the jumper wires are connected to the same row as the button. As you can see from the diagram, the button will be connecting a Raspberry Pi input pin to ground when pressed.

>STEP-04

Finish up

Using the five male-to-male jumper leads, follow the diagram to connect all the negative legs on the LEDs and the buttons together, which can then be connected to a ground pin on the Raspberry Pi. The eight male-to-female jumper leads can then be used to connect the positive sides of the LEDs to the Pi's output pins, as well as connecting the two buttons and the common ground connection. Ensure that each jumper lead is connected to the correct pin on the Raspberry Pi, as this is crucial for the code to run correctly with the hardware.

>STEP-05

Install the code

Using the latest version of Raspbian (PIXEL) on your Raspberry Pi, you can now boot up. If you are not using this version of Raspbian, you will need to install the GPIO Zero library if you haven't already. You'll need to make sure that your Pi is connected to the internet, via WiFi or Ethernet. When booted, open the terminal window and type: **`git clone https://github.com/henrybudden/rpesk-advanced/`**. This will download the Python files from GitHub, onto your Raspberry Pi. Once this is completed, you can navigate to the folder using **`cd repsk-advanced`** and then run the game by typing **`sudo python 02_reaction_game.py`**.

Language

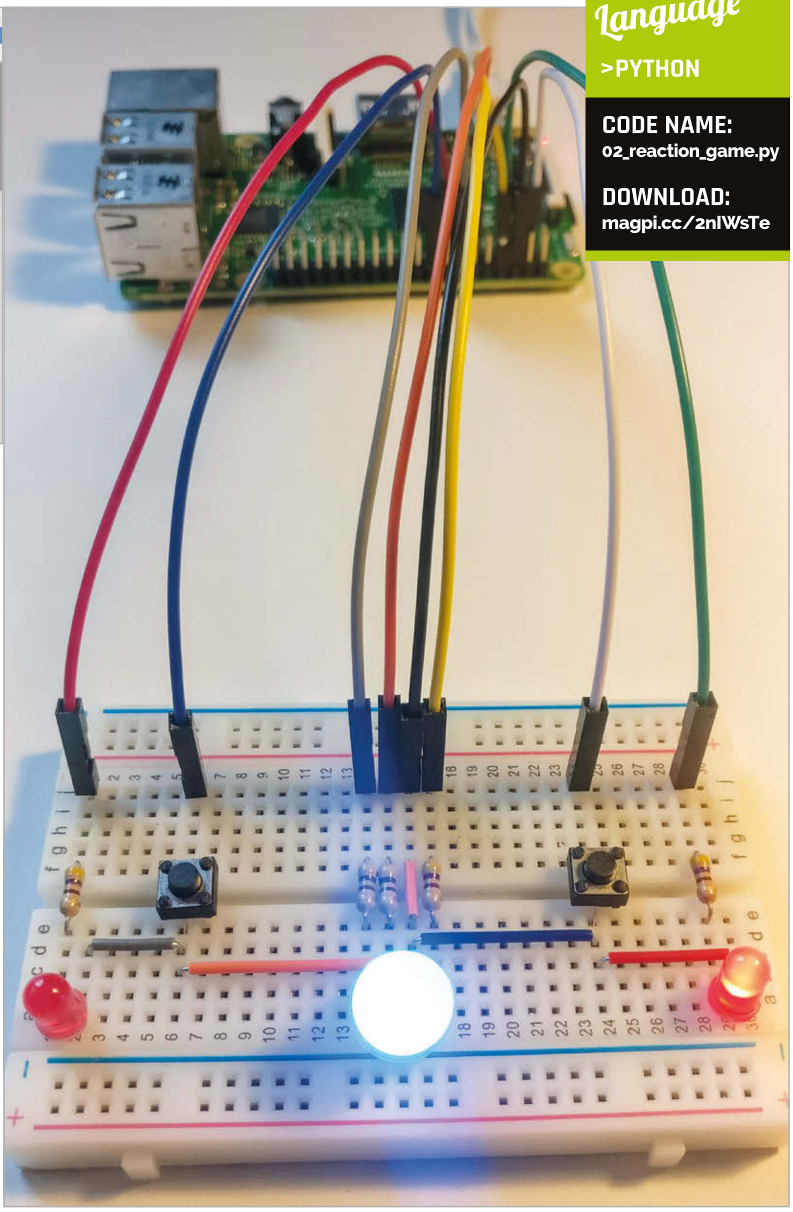
>PYTHON

CODE NAME:

02_reaction_game.py

DOWNLOAD:

magpi.cc/2nIWsTe



Above For a tidy breadboard, trim the legs of the components and jumper wires so that they sit neatly on the surface

>STEP-06

How to play

When you run the program, you will first need to type the names of the two players. Player one's button is located at the bottom of the breadboard and player two's at the top (as shown on the diagram). Whenever the RGB LED in the centre of the breadboard lights up in any colour other than red, press your button as fast as you can. The first player to hit their button will gain a point. However, beware! If the RGB LED lights up in red, do not press your button. If you do, one point will be deducted from your score! After each round, the score for each player is displayed on the screen, and one of the small red LEDs on the breadboard will light up on the side of the player in the lead. If both LEDs are lit up, you are tied. Each round will begin automatically, so keep going until you get too tired!

PART 01

WILLIAM H. BELL

Engineering consultant, particle physicist, computer software developer, IT administrator, magazine editor, and education enthusiast.

whbell.net

CREATE A MONOPOLY SIMULATION

Hack the board game of Monopoly by figuring out which properties are best to buy. This tutorial simulates a game and figures out the winning squares

You'll Need

- Raspberry Pi
- Raspbian
- Matplotlib

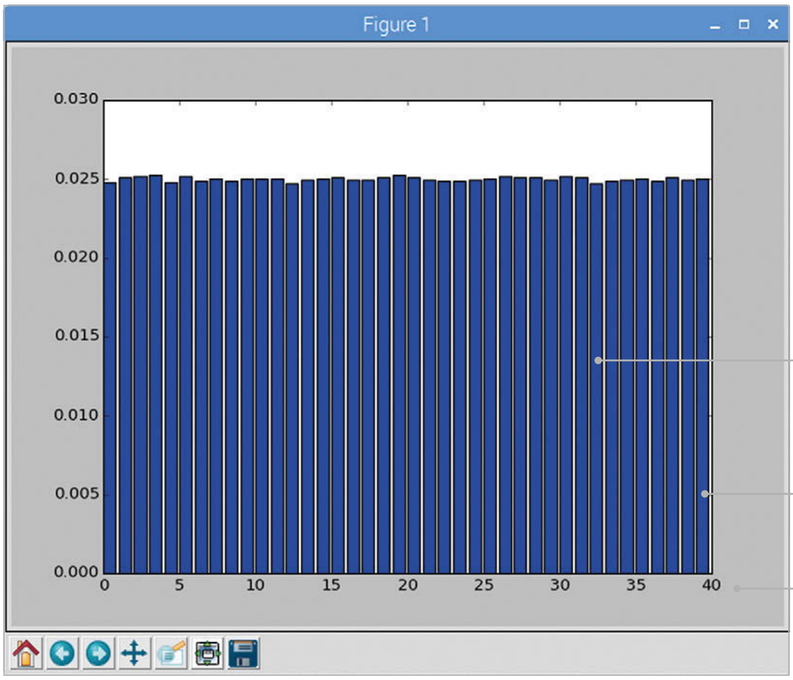
Monopoly is a great board game that can be the subject of many arguments. After playing the game for a long time, players seem to land on some board squares more than others. Some people seem to favour the second street (light blue), while others choose the fourth (orange). Therefore, to settle these arguments, a simple discrete event simulation can be built that describes the game and predicts where the players will land on average. In this series, the simulation will be built from simple components into a complex model of the board that predicts the probability of where players will land.

Welcome to the world of 'discrete event simulations'. These are used to model all sorts of effects, from when trains arrive to particle collisions at the highest energies.

The simulations are constructed from an understanding of the expected probability distributions. These are combined together to form the final event. In the case of trains, the train timetable and the distribution of arrival times can be used as an input probability distribution. For particle physics simulations, the initial probability distributions are taken from theoretical models.

In the case of our Monopoly game, the random elements are the dice and the order of the cards in the Chance and Community Chest decks. Once you've got these, you can play games of Monopoly and discover which squares are landed on the most. Armed with this info you can be sure to buy these squares, and improve your chance of winning a game.

In this article, a simple simulation of just the board and the dice is introduced. The first building block in this is the simulation of rolling two dice.



The chart currently shows the probability of landing on each of the Monopoly board squares, where no additional effects (like Go To Jail) have been added. We start with everything even

The vertical bars are the probability of landing on a Monopoly board square

The 40 bars are the Monopoly board squares, from 'GO' (0) to the last square (39)

Rolling the dice

Players move around a Monopoly board by throwing two dice. An ideal six-sided die has the same probability of landing on each side. Therefore, the probability of throwing any of the numbers is the same:

```
P(1) = P(2) = P(3) = P(4) = P(5) = P(6)
```

The total probability for all allowed states is defined as one. Therefore, each of the sides of the die has a probability of 1/6 of being selected.

In the Monopoly board game, two dice are rolled rather than one. The probably of throwing a total value on two dice is not the same for the all of the possible total values, since there is more than one way of producing some of the outcomes. The probability of rolling a total of twelve is:

```
P{6,6} = P(6) x P(6) = 1/36
```

...since there is only one way to create this total value. However, the probability of rolling a total value of five is the sum of the probability for each combination:

```
P{2,3} + P{3,2} + P{1,4} + P{4,1} = 4/36 = 1/9
```

Simulating two dice

The effect of throwing a perfect six-sided die can be simulated by using a random number generator to generate integers between one and six. The Python random library provides a function, **randint**, that allows integers to be generated between two limits.

twoDice.py

Language

>PYTHON

DOWNLOAD:
github.com/williamhbell/MonopolySimulation

```
# Import the random package
import random

def rollTwoDice():
  # Generate two random numbers within the
  # range 1 <= i <= 6 and add them together
  return random.randint(1,6) + random.randint(1,6)


# A list to contain the total value rolled.
# There are 12 elements, because total value you can roll is 12.
counters=[0.]*12

# Roll two dice 100 times.
nRolls = 100
for i in range(nRolls):
  totalValue = rollTwoDice()  # roll the dice

  # Python list indices count from zero. Therefore, have to remove
  # one from the totalValue to put it into the right index in the list.
  # count this total value
  counters[totalValue-1] = counters[totalValue-1] + 1.


# Total probability is always defined as 1.
# Therefore, have to divide by the total number of counted values.
for i in range(len(counters)):
  counters[i] = counters[i] / float(nRolls)

# Now print out the probabilities for each of the combinations
print("The probabilities of rolling a total value using two dice:")
for i in range(len(counters)):
  # Need to add one, since Python counts from zero.
  print(" P("+str(i+1)+")="+str(counters[i]))
print(
"where P(n) is the probability of rolling a total of n on two dice.")
```

The standard random number generator that is packaged with Python is not perfectly random, but is sufficiently random to be used for this project. It generates numbers from a random series, which is initialised from a random number seed. If the seed is not set within the program, then a different seed is chosen each time the Python program that uses the library starts.

Rolling two dice can be simulated by running the **randint** function twice and adding the result together, where each **randint** function call produces an independent value.

The **twoDice.py** program simulates rolling two dice 100 times. The program contains a simple function called **rollTwoDice** that calls the **randint** function twice, adds the return values together and returns the result. When the program starts, it creates a list called **counters**. This list has twelve elements set to zero.

A floating point value is used in this list, since the final probability values are floating point numbers. Once the list has been created, the program goes into a loop calling **rollTwoDice** and recording the values returned in the **counters** list. Once the simulated dice have been rolled 100 times, all of the counter values are divided by the number of dice rolls. This creates a probability distribution function for a total value on two dice.

Try running the **twoDice.py** program by typing:

```
./twoDice.py
```

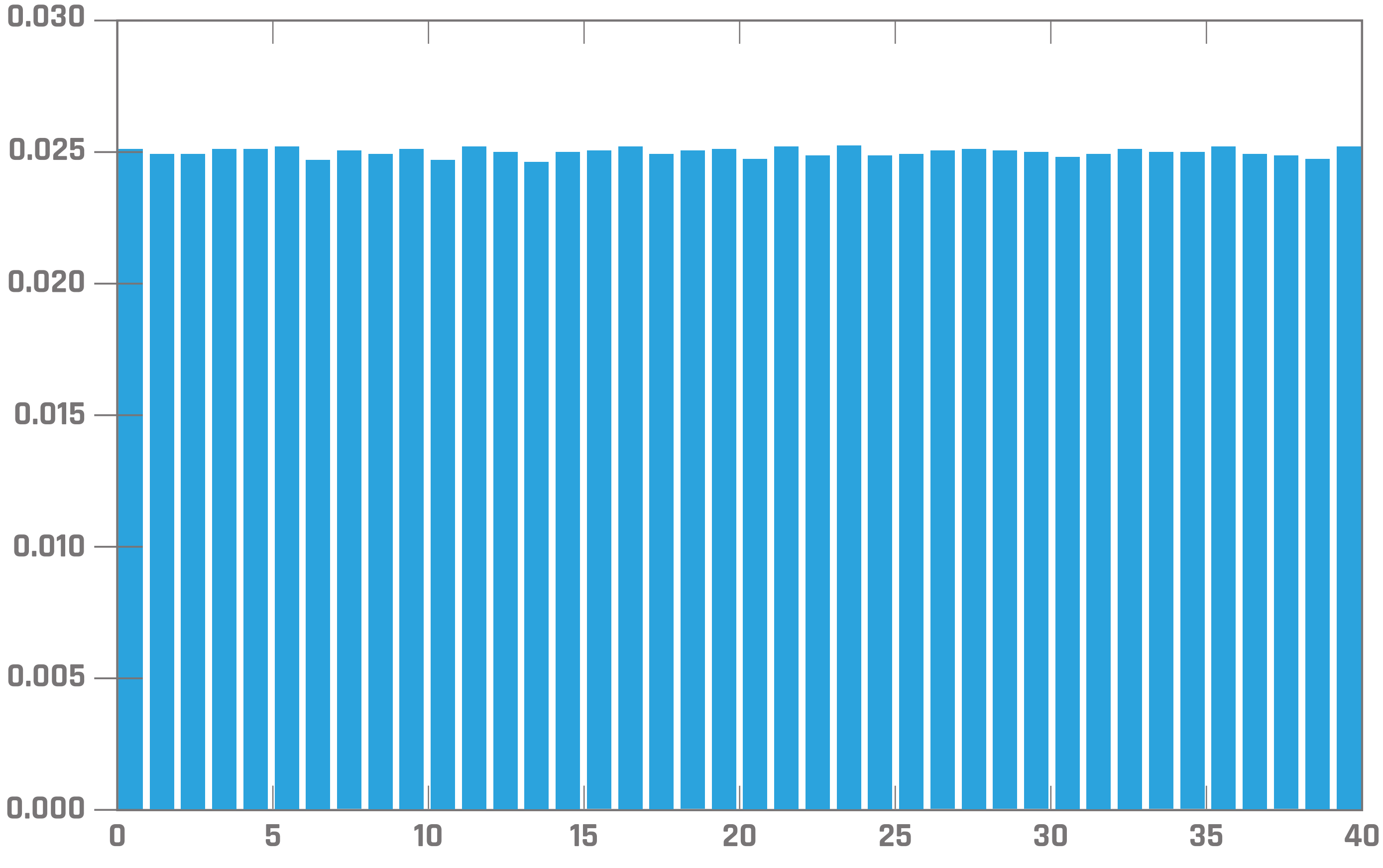
Then use a text editor, such as IDLE, to increase the value of **nRolls** by a factor of 100 or 1000 and rerun the program. As the number of rolls becomes very large, the probability distribution function reported by the program approaches the theoretical probability values for the different results of rolling two dice.

The difference between the theoretical probability values and the observed probability values occurs due to statistical uncertainties. As the number of rolls increases, the statistical uncertainty on the resulting probability distribution function drops.

Modelling the board

The Monopoly board has 40 squares. A player starts from the 'GO' square and then moves around the board in a clockwise direction until the game finishes. Each of these squares can be given a number from zero to 39, where the 'GO' square is zero and the last square on the board is 39. As with the **twoDice.py** program, the number of times each square is chosen can be counted using a Python list with 40 elements. The only other information that is needed is the current player position.

A simulation of the Monopoly board without any other effects is given in the **boardOnly.py** program. This program includes the **rollTwoDice** function, as well as a function called **plot**. The latter uses the **pyplot** functions **bar** and **show** to produce a bar chart that is shown on the screen. The **pyplot** function is part of Matplotlib. Therefore, before running this program, the Matplotlib library must be installed.



```
sudo apt-get install python-matplotlib
```

The program starts by initialising a Python list called **counters** with 40 elements that all contain zero. The values are set to be floating points, since the values will be divided by the total number of entries at the end of the program. The number of dice rolls is set to be 1,000,000, so that the entries in the resulting probability distribution function have a low statistical uncertainty. The current position is set to be zero, which corresponds to the 'GO' square, then the program loops until the number of dice rolls has been achieved.

Each time the program loops, the **rollTwoDice** function is used to create a new total value. The total value from the two simulated dice rolls is then added to the current position. If the new position is beyond the 39th square, the total number of squares is removed so that the new position is still on the board. Then the new position is recorded in the **counters** list.

After the player has gone around the board a few times, the frequency distribution recorded in the **counters** list approaches the theoretical distribution. The final frequency distribution is then divided by the total number of moves recorded to form the probability distribution function for the board. Rather than just printing out the different probability values, they can also be shown as a histogram using the Matplotlib library. This is achieved by calling the **plot** function. The resulting plot stays on the screen until the associated window is closed.

Try running the **boardOnly.py** program by typing:

```
./boardOnly.py
```

The simulation takes several seconds to run, since it is set to run a large number of dice rolls by default. When it finishes, the Matplotlib window will appear, with the histogram of the different probability states. To make the program as simple as possible, the plot is not labelled. However, the x-axis label of 'Monopoly board square' and the y-axis label of 'P(square)' can be set by calling other Matplotlib functions.

The probability distribution generated by running the **boardOnly.py** program is not flat because of statistical uncertainty on each counter value. As the number of rolls becomes bigger, the distribution becomes flatter.

The **boardOnly.py** program produces a basic probability distribution function shape that will no longer be flat when the other effects of the Monopoly board game are added.

In next issue's article, the most important effect in the Monopoly board game will be added. Then other features will be added towards the complexity of the full game. As with all complex simulations, this will allow each of the effects to be observed and quantified. By the end of the series, it will be clear which properties should be bought to win the game.

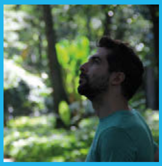
Language

>PYTHON

DOWNLOAD:
github.com/williamhbell/MonopolySimulation

boardOnly.py

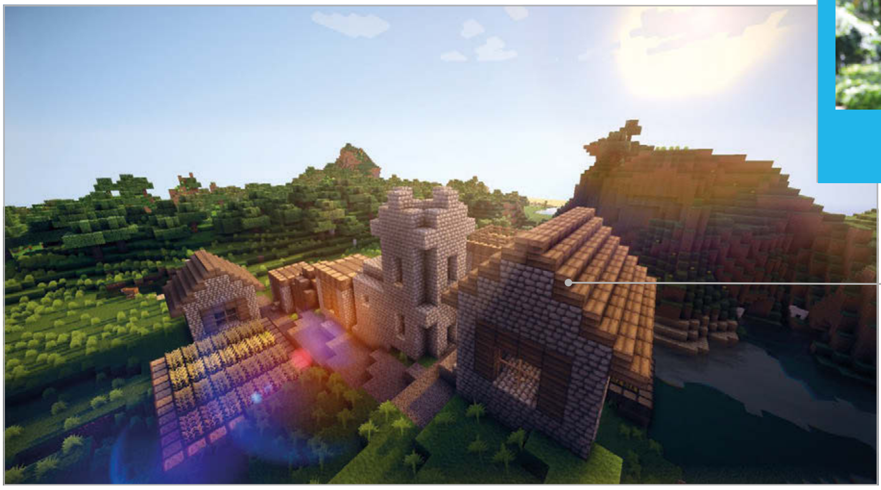
```
# Import the random package
import random

# Import the matplotlib pyplot package
import matplotlib.pyplot as pyplot

def rollTwoDice():
  # Generate two random numbers within the range
  # 1 <= i <= 6 and add them together
  return random.randint(1,6) + random.randint(1,6)

# A function to create a histogram
def plot(x, y):
  pyplot.bar(x, y)
  pyplot.show()

# The number of squares on the board
nsquares = 40

# A list to contain the total value rolled.
counters=[0.]*nsquares

# A variable to hold the current position
currentPosition = 0

# Set the number of rolls
nRolls = 1000000

# Print a message
print("Rolling two dice " + str(nRolls) + " times...")

# Roll the dice
for i in range(nRolls):
  # roll the dice
  totalValue = rollTwoDice()

  # Move the player to the next position
  currentPosition = currentPosition + totalValue

  # If the player has moved past last square, wrap board around.
  if currentPosition >= nsquares:
    currentPosition = currentPosition - nsquares

  # Count the current position on the board
  counters[currentPosition] = counters[currentPosition] + 1.

# Total probability is always defined as 1.
# Therefore, have to divide by the total number of counted values.
for i in range(len(counters)):
  counters[i] = counters[i] / float(nRolls)

# Now print out the probabilities for each of the combinations
print("The probabilities of landing on a given Monopoly square
after " + str(nRolls) + " rolls")
for i in range(len(counters)):
  # Need to add one, since Python counts from zero.
  print(" P("+str(i)+")="+str(counters[i]))
print("where P(n) is the probability of landing on the nth Monopoly
board square")

# Create a bar chart display
plot(range(len(counters)),counters)
```



PATRICIO GONZALEZ VIVO

Artist and graphic engineer, and author of *The Book of Shaders*, Patricio uses the language of light to turn data into stunning landscapes. He believes in developing a better together.

patriciogonzalezvivo.com
@patriciogv



Here's a GLSL shader mod running on Minecraft – a big difference from normal Minecraft!

GLSL SHADERS

THE COMPUTER LANGUAGE OF LIGHT

Make state-of-the-art special visual effects and animations for your video games and projects

You'll Need

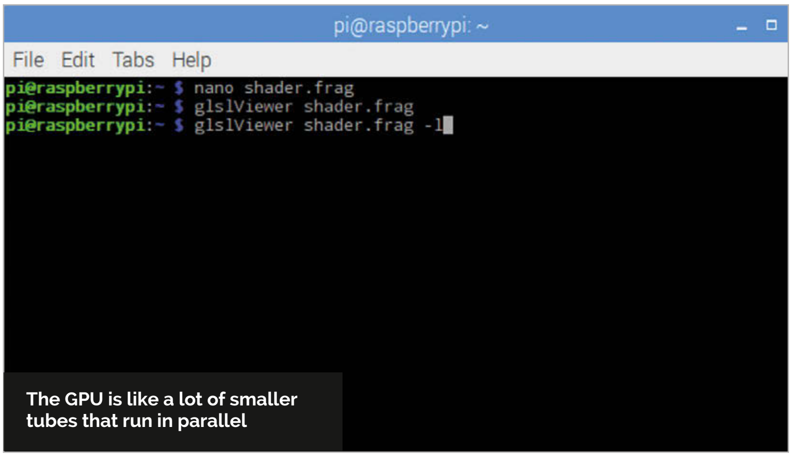
- glslViewer **magpi.cc/2hNP4r3**
- The Book of Shaders **magpi.cc/2nJ5dR2**

Since the 1960s, computers have had graphic interfaces: mouse pointers, windows, icons, sidebars, all the things that are now common on home PCs. It hasn't been until the last few decades that video games have pushed the limits of visual content on the computer. Video games have reshaped the computer from the inside out, and one particular piece of hardware is responsible for the leap: the graphics processing unit (GPU). Thanks to the GPU, you can enjoy realistic 3D worlds with incredible cinematographic effects on them. GPUs can be programmed, too: the small pieces of code that run on them are called 'shaders'. An example of this is the GLSL Shaders mod pack for Minecraft (**magpi.cc/2ikJsEN**).

Shaders are a dry, C-like computer language designed with the main purpose of rendering an image pixel by pixel without sacrificing performance. We can think of shaders as the computer language of light.

So how do shaders work? And how are they different from other computer languages? Graphics are quite hard for regular computer programs to process. Regular programs are written to be run on the CPU; they are designed to do one task after the other – like Python, for example, where the program follows a list of orders, one line at a time.

Processing a digital image requires assigning a colour to each pixel on the screen. Let's say you have an old 800×600 monitor: that requires around 480,000 calculations. Now let's say your image is actually an animation that needs to change 24 times per second to produce the illusion of movement. That is more than 11,520,000 calculations per second – enough calculations to freeze a CPU.



```
pi@raspberrypi:~ $ nano shader.frag
pi@raspberrypi:~ $ glslViewer shader.frag
pi@raspberrypi:~ $ glslViewer shader.frag -l
```

The GPU is like a lot of smaller tubes that run in parallel

A better solution

That's why engineers came up with a different way to process graphics on computers: they call it a GPU. You've probably heard it described as 'the graphics card'. The main design difference is parallelization. The chip inside the GPU runs multiple tasks in parallel at the same time, each one independent from the others. In the first chapter of *The Book of Shaders* (**thebookofshaders.com**), you can find a more detailed explanation of this.

Learning how to program your own shaders is not simple, but it doesn't need to be hard any more! You can find a program called glslViewer (**magpi.cc/2hNP4r3**) on the latest Raspbian distribution. This little program is made to view shaders and follow the examples in *The Book of Shaders*. This gentle step-by-step guide will walk you through learning about this abstract language of light. You can print a version of the book by following the instructions in this appendix chapter: **magpi.cc/2iwOwWW**. You will need a web browser with WebGL to both view and print the book.

Live coding with glslViewer

Running directly from the command line, glslViewer is a flexible renderer for shaders that doesn't require a windows manager.

To live-code a shader, you just need a text editor to do the coding and glslViewer will display the changes immediately. Let's try it out!

Open a Terminal window and install glslViewer:

```
sudo apt-get install glslviewer
```

Next, open a new file call **shader.frag** in your favourite editor (like gedit, nano, vi, or Emacs):

```
nano shader.frag
```

...and write in the following code:

```
#ifdef GL_ES
precision mediump float;
#endif

void main() {
  gl_FragColor = vec4(1.0,0.0,1.0,1.0);
}
```

Save the file, exit the editor, and give it a try on glslViewer:

```
glslViewer shader.frag -l
```

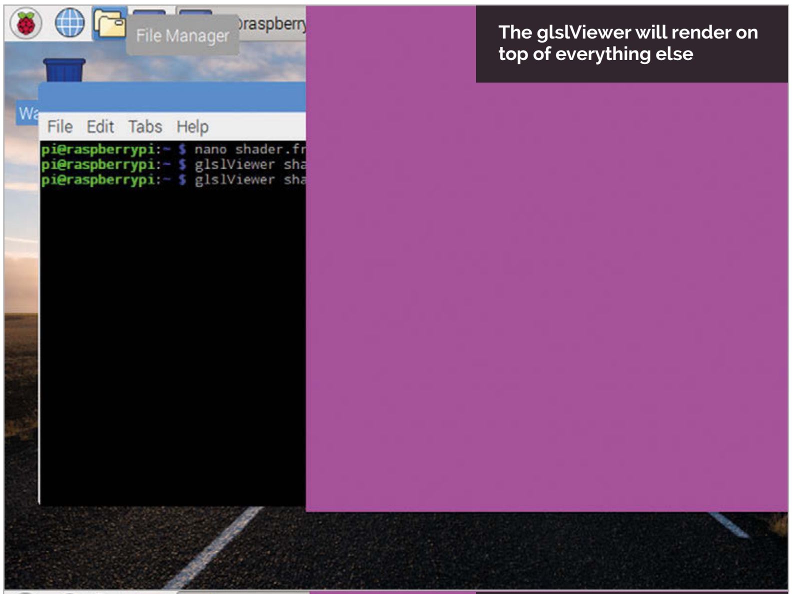
You should see a magenta rectangle at the top-left corner of your screen. It's not much, just a starting point.

Note: the **-l** is to open it in live-code mode; this will open it not in full-screen, so you can keep editing it. You can also change the position and size of the viewport by using **-x**, **-y**, **-w**, and **-h**.

Let's push glslViewer to run in the background by pressing **CTRL+Z**. Once back in the Terminal, open your file again and you can keep editing!

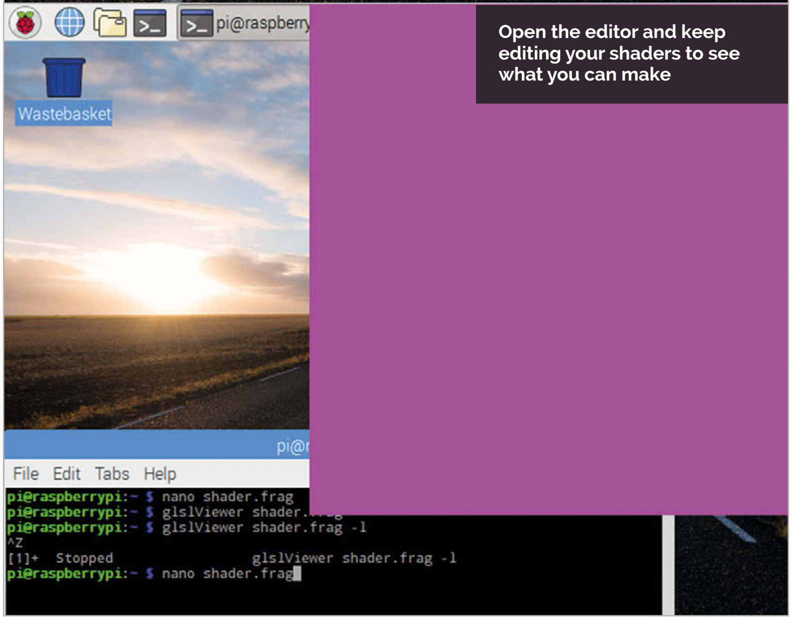
```
nano shader.frag
```

By following along with the 'Hello World' chapter in *The Book of Shaders* (**magpi.cc/2ikHZhZ**), you can learn more about what's happening here, and how to change your code to create more complex colours and animation. Once you have finished editing, you can close your editor and close glslViewer. For that, you need to bring it from the back to the foreground by typing **fg** into the Terminal, and then close it with the letter **Q**.



```
pi@raspberrypi:~ $ nano shader.fr
pi@raspberrypi:~ $ glslViewer sha
pi@raspberrypi:~ $ glslViewer sha
```

The glslViewer will render on top of everything else



```
pi@raspberrypi:~ $ nano shader.frag
pi@raspberrypi:~ $ glslViewer shader.
pi@raspberrypi:~ $ glslViewer shader.frag -l
^Z
[1]+  Stopped                 glslViewer shader.frag -l
pi@raspberrypi:~ $ nano shader.frag
```

Open the editor and keep editing your shaders to see what you can make

Playing examples

There is a Python script, supplied with glslViewer, that will help you to download and play the examples in the book. To do this, check out the examples below and you'll be able to get shaders working on your Raspberry Pi in no time:

```
glslLoader https://thebookofshaders.com/10/ikeda-00.frag
glslLoader https://thebookofshaders.com/10/ikeda-01.frag
glslLoader https://thebookofshaders.com/10/ikeda-02.frag
glslLoader https://thebookofshaders.com/10/ikeda-03.frag
glslLoader https://thebookofshaders.com/10/ikeda-04.frag
```

MIKE'S PI BAKERY

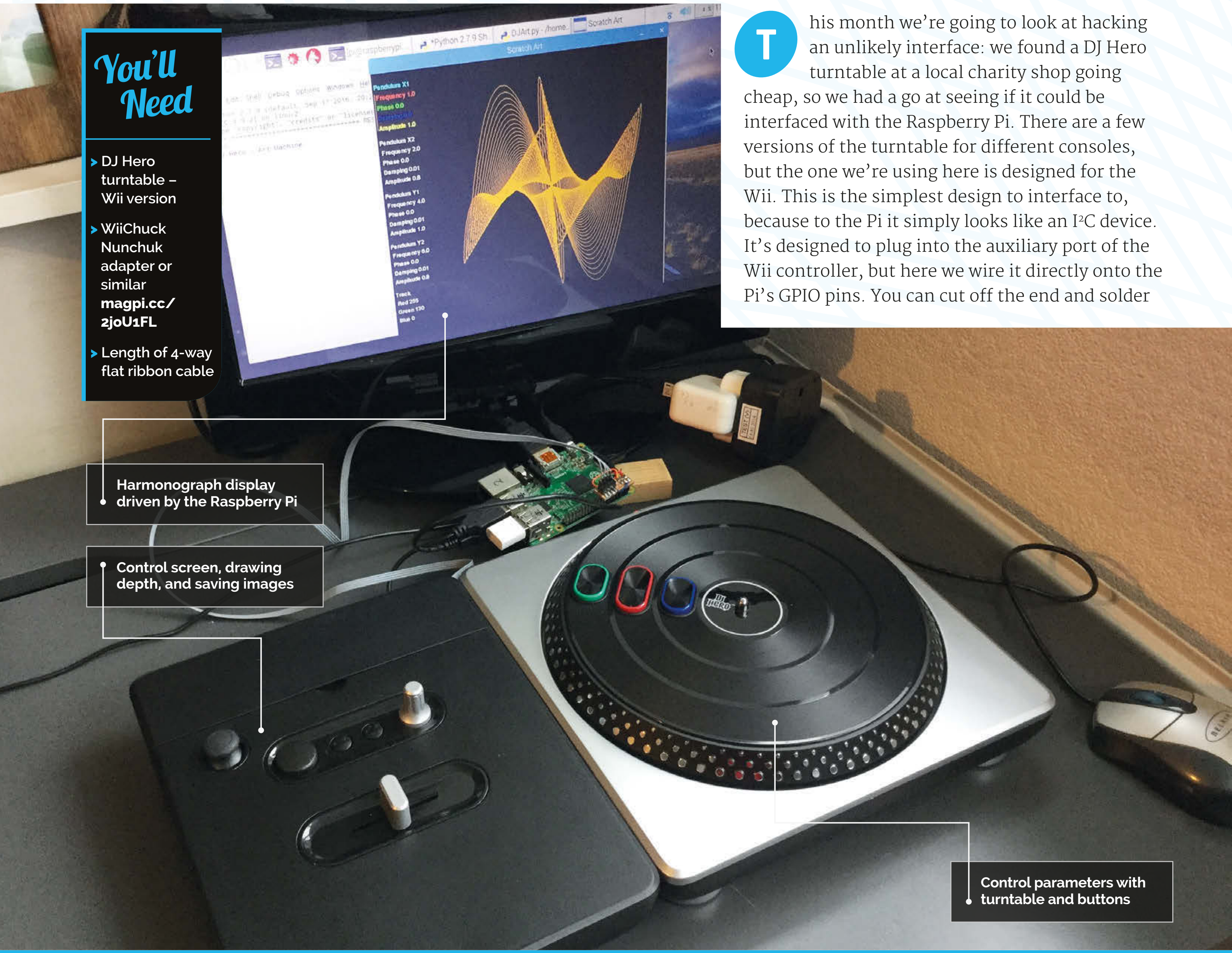
MIKE COOK

Veteran magazine author from the old days and writer of the Body Build series. Co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/259aT3X

MAKE STUNNING SCRATCH ART

Hack a DJ Hero turntable to simulate a harmonograph and make intricate line art by scratching



Harmonograph display driven by the Raspberry Pi

Control screen, drawing depth, and saving images

Control parameters with turntable and buttons

You'll Need

- DJ Hero turntable – Wii version
- WiiChuck Nunchuk adapter or similar **magpi.cc/2joU1FL**
- Length of 4-way flat ribbon cable

This month we're going to look at hacking an unlikely interface: we found a DJ Hero turntable at a local charity shop going cheap, so we had a go at seeing if it could be interfaced with the Raspberry Pi. There are a few versions of the turntable for different consoles, but the one we're using here is designed for the Wii. This is the simplest design to interface to, because to the Pi it simply looks like an I²C device. It's designed to plug into the auxiliary port of the Wii controller, but here we wire it directly onto the Pi's GPIO pins. You can cut off the end and solder

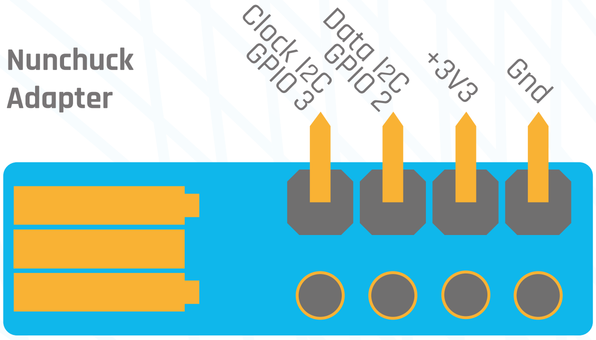


Fig 1 Wiring the Wii Nunchuk adapter to the Pi

extension wires to it, or get a small 'Wii Nunchuk adapter' PCB and wire that to the Pi. **Fig 1** shows the wiring you need; as there are only four wires, you can use a length of ribbon cable to attach it to the Pi, with the cover neatly clipping over the top.

Talking to the DJ Hero

The DJ Hero comprises a collection of buttons, knobs, a joystick, and a turntable. **Fig 2** shows these controls labelled up as the functions they're going to perform in this project. The I^{2}C address of this device is **`0x52`** and consists of a number of registers. To initialise the device on power-up, you must write the value **`0x55`** to register **`0xF0`**. Note that each successive read will increment the address of the register read from. So, to read the interface status, you need to set the register to zero and then read data six times – these six bytes contain the information from the DJ deck. The assignment of bits is shown in **Fig 3** overleaf, and at first sight looks a bit complex. However, this diagram has been simplified, in that the DJ Hero is capable of having a second turntable on the left-hand side, and the bits corresponding to this second turntable have been omitted.

Language

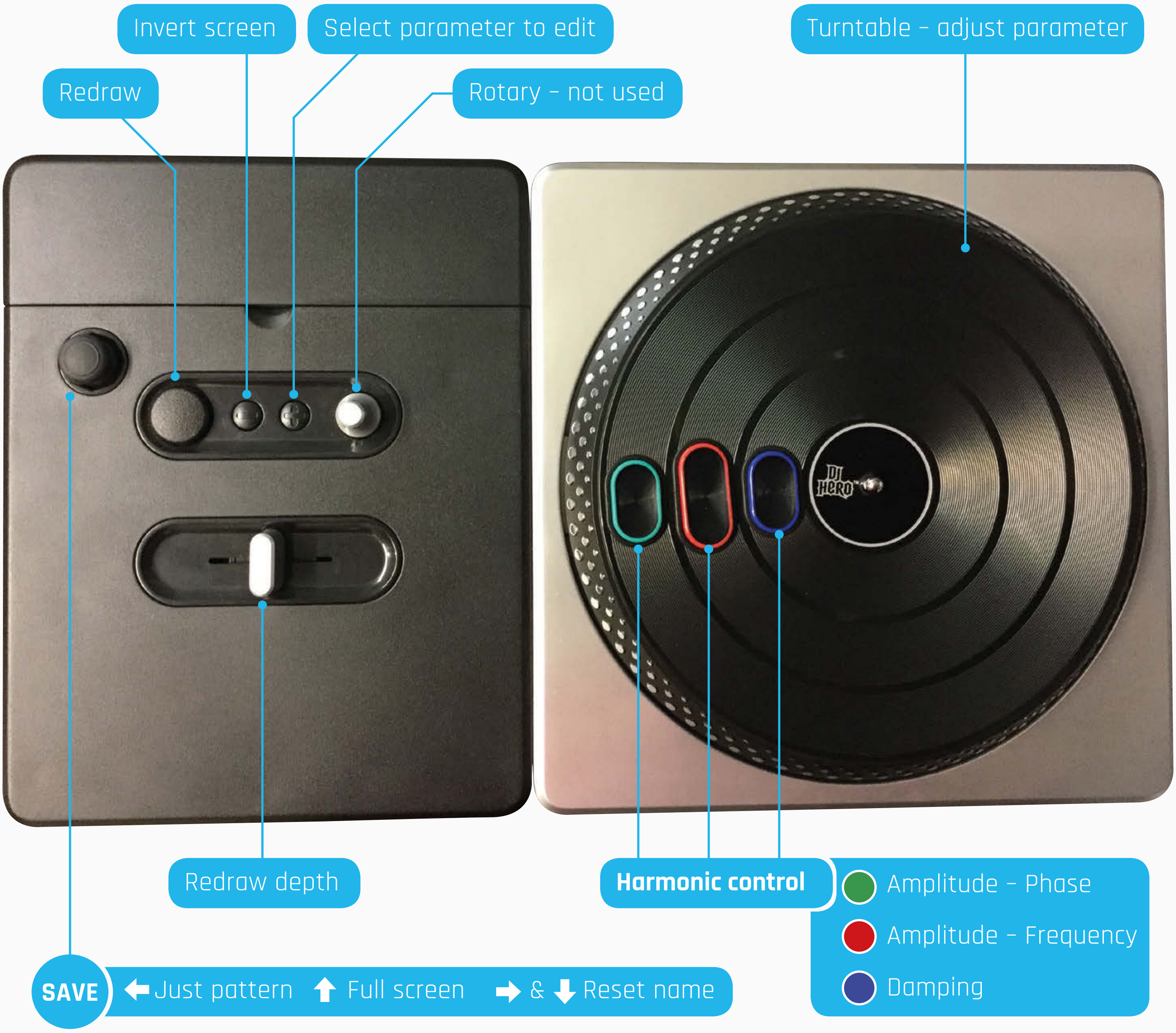
>PYTHON 2

DOWNLOAD THE CODE:
magpi.cc/1NqJjmV

PROJECT VIDEOS

Check out Mike's Bakery videos at:
magpi.cc/1NqJnTz

Fig 2 The full set of controls on the DJ Hero controller



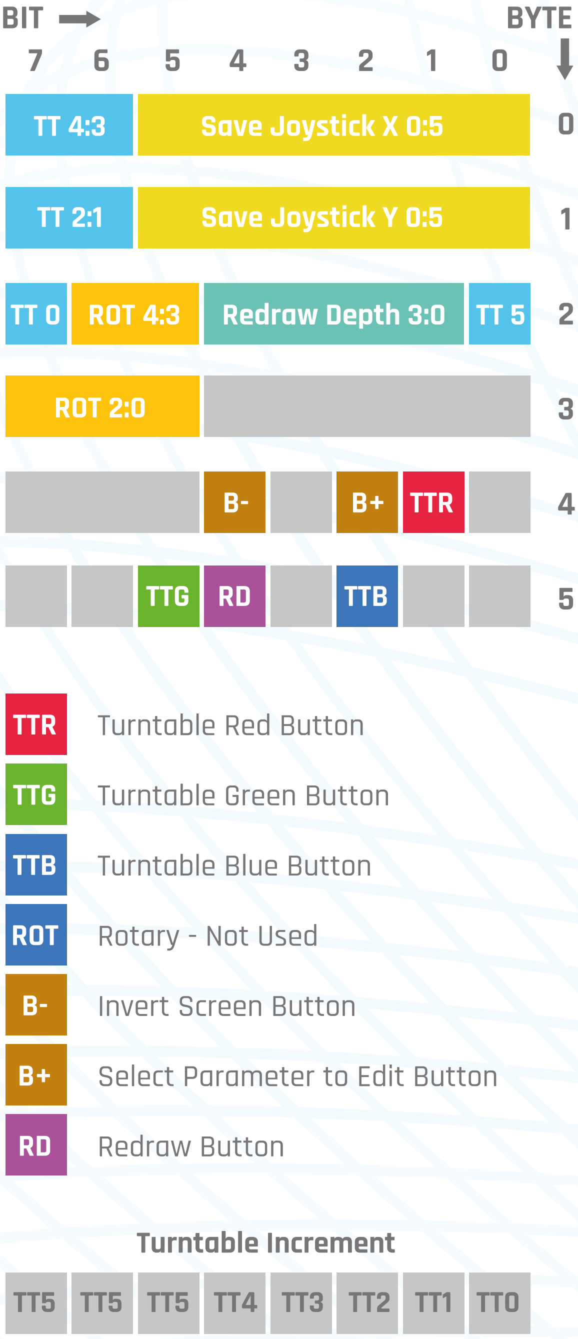

BIT
7 6 5 4 3 2 1 0
BYTE
0: TT 4:3 | Save Joystick X 0:5
1: TT 2:1 | Save Joystick Y 0:5
2: TT 0 | ROT 4:3 | Redraw Depth 3:0 | TT 5
3: ROT 2:0
4: B- | B+ | TTR
5: TTG | RD | TTB
TTR Turntable Red Button
TTG Turntable Green Button
TTB Turntable Blue Button
ROT Rotary - Not Used
B- Invert Screen Button
B+ Select Parameter to Edit Button
RD Redraw Button
Turntable Increment
TT5 TT5 TT5 TT4 TT3 TT2 TT1 TT0
Sign Extend

Fig 3 The data block returned from the DJ Hero

The buttons are simple: one bit in one byte of the returned data block, and other controls are spread over several bits. The two axes of the joystick are the first six bits in the first two bytes, whereas the values from the slider are in bits 4 to 1 of byte number 2. The rotary control's five bits are split up over two bytes, whereas the turntable is scattered all over the place in four locations. You'll need a bit of software to pull out the data you want from this block.

The turntable returns a value, not of its position, but of the change in the position since the last time it was read. This is a five-bit signed value, with the most significant bit being the sign bit. To be useful, this must be converted into the sort of number that Python can understand, so bit 5 of the turntable number is propagated to all the higher bits in the word; this is known as sign extension. Also, for negative values, the number must be in the 'one's complement' format; that is, with bits 0 to 4 inverted and 1 subtracted. This format is universally used in computer languages and prevents you from having two different bit patterns for plus zero and minus zero, meaningless concepts.

The graphics

We're going to use this controller to control a simulation of a harmonograph. These devices were originally a very popular Victorian contraption, consisting of a pen on the end of a compound pendulum that was sent swinging and produced Lissajous-like patterns. Here, however, we're going to simulate a system with four pendulums that will exceed the flexibility of any mechanical device. Two pendulums control the X position of the pen, and two control the Y position, the final position being the simple sum of each of the two pendulums on each axis. There are four parameters that control the exact path of each pendulum.

01. **The Amplitude or extent of the swing**

02. **The Frequency of the swing, determined by the pendulum's length**

03. **The Phase of the swing, a fixed value added to the frequency**

04. **The Damping or decay, which is the slow reduction of amplitude as the friction takes energy out of the pendulum**

It's the subtle interaction of these four parameters that produces the near-infinite number of patterns that can be obtained. There's also the factor of how long you let it swing for, producing a less or more dense pattern. Eventually the damping will make all the pendulums stop, although in this simulation you can set the damping value to zero if you wish.

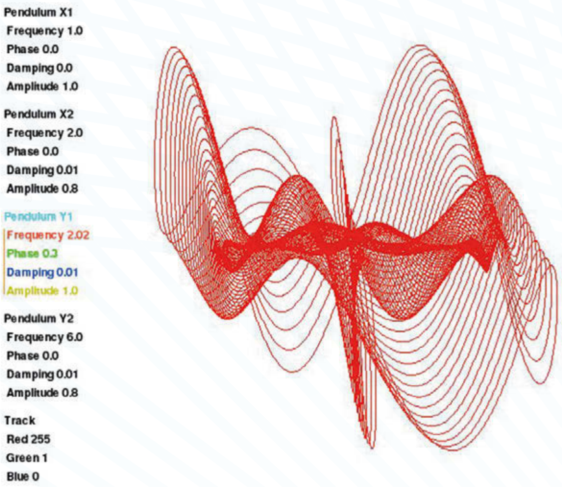
The software

The code for this is **DJArt.py** and is written under the Pygame framework. The drawing area is set by the **screenSize** variable, and you could make the window size larger simply by altering this value if you have a higher-resolution monitor. In addition, the window is wider by 100 pixels given by the **controlBar** variable, in order to have room for the editable parameters. The **swing** function is the one that actually draws the path of the pendulums on the screen, with the points calculated by the **calcNewPoint** function. Rather than showing every point on the screen as it is plotted, we speed things up considerably by drawing for 40ms and then displaying the result. All the rest of the code is devoted to editing the parameters of the program. The **initValues** function sets the startup parameters; you might want to change this as you explore new combinations of pendulum frequencies.

The DJ Hero peripheral is handled by the **readDJ** function, and the **buttonFix** function returns a one if the button is pressed, or else it returns a zero. The raw data returns the bit as zero when a button is pressed, and a one when not. The **adjustPends** function imposes a wraparound value limiting the range on the parameters. Keeping the amplitude at one or less ensures that the drawing stays within the screen area.

Using the software

When running, the software starts out by having a starting set of parameters producing a pattern on a black background. The - (minus) button will toggle the background between black and white. **Fig 4** shows the full program window against a white background. The Redraw button will run the simulation for a time determined by the 'Drawing Depth' slider; with the slider to the left, you get the shortest time. The + button allows you to edit the parameters of each of the four pendulums and the track colour. The selected pendulum's parameters are displayed in that colour, with the others shown in the foreground colour. Changing a parameter is simple: just press the appropriate colour button on the turntable and move it clockwise for an increase, or anticlockwise for a decrease. For example, the pendulum's frequency is shown in red, so holding down the red button and moving the turntable will change the frequency. However, there are four parameters and only three buttons. The fourth parameter is amplitude and coloured yellow; to change this, simply hold down both the red and the green keys at the same time.



Pendulum X1
Frequency 1.0
Phase 0.0
Damping 0.0
Amplitude 1.0

Pendulum X2
Frequency 2.0
Phase 0.0
Damping 0.01
Amplitude 0.8

Pendulum Y1
Frequency 2.02
Phase 0.3
Damping 0.01
Amplitude 1.0

Pendulum Y2
Frequency 6.0
Phase 0.0
Damping 0.01
Amplitude 0.8

Track
Red 255
Green 1
Blue 0

Fig 4 The control screen, and the result of different parameters taken from the controller

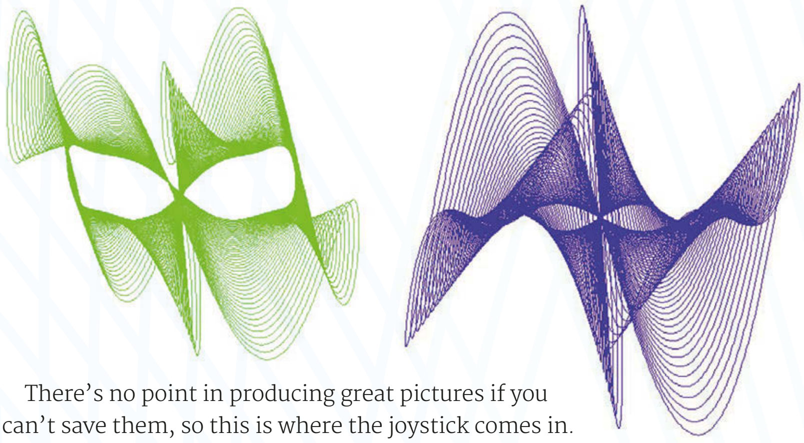


Fig 5 Designs created using the DJ Hero controller in this tutorial

There's no point in producing great pictures if you can't save them, so this is where the joystick comes in. Move it fully left to save just the picture and fully up to save the picture along with the parameter values. The first time you do this, a dialogue box will prompt you for the name and place to store the image. After the first one is saved, subsequent saves append a number to the file name. Once a picture is saved, there is a 'saved' message in the lower left-hand corner of the window which lasts until you press another control. If you want to use a new name or new folder, then move the joystick fully to the right or fully down and you will get the file dialogue box next time you save an image.

Getting good results

The patterns look best on the screen with a black background, but if you're going to print it out then it looks a lot better on a white background. We think the best patterns are obtained when the frequency of two pendulums are set as a simple ratio of each other, or slightly off a simple ratio, like **1:2.01**. Unrelated frequency values tend to produce scribble-like, chaotic results. It's amazing how small changes in phase or damping can radically change the picture. This is especially true of the phase, which wraps round every cycle; as this is shown in radians, this is at a value of **5.3**. When changing a parameter, there's a short real-time preview of the resulting shape. Other parameters have lower limits of either zero or one, while the colour has an upper limit of 255. You can actually get Lissajous figures if you want by setting the amplitude of pendulums 1 and 3 to zero, and the amplitude of the other two to one, along with a decay of zero. **Fig 5** shows a selection of patterns we have obtained so far.

Taking it further

The only control not used is the rotary one. You might like to try to add some code so that this controls the thickness of the plotted line. Altering the short variable value to a larger number alters the depth of the pattern preview; you might like to make this bigger if you're using a Raspberry Pi 3, because it can draw faster. You could also add software to change the colour of the track according to the absolute X or Y position or change it between two colours, either abruptly or fading from one to the other over a length of track.

SPENCER ORGAN

Chemistry and physics teacher, Raspberry Pi enthusiast and Certified RPi Educator from the West Midlands with a passion for running workshops and building fun, educational, and practical things with the Pi!
magpi.cc/2bkQ53q

BUILD THE MARAUDER'S CLOCK

Know when your family get home with this beautiful Harry Potter-inspired clock

You'll Need

- Servos (continuous and position)
- Watch hands
- Arduino Uno **magpi.cc/2lO4uNA**
- 2× Mini breadboard **magpi.cc/2lOjISN**
- Transparent medium breadboard **magpi.cc/2lOfNVW**
- Pimoroni Explorer HAT Pro **magpi.cc/2lOn5ZZ**
- 2× 5V mini DC relay **magpi.cc/2lOere6**
- 4× LED, male-to-male and male-to-female jumper cables
- A box or clock body
- Background graphics

We all love the many visual elements in the Harry Potter movies, and this project combines two of the most iconic: the Marauder's Map and the Weasley 'who is home?' clock. The project uses a Raspberry Pi to detect the presence of people on the network and an Arduino Uno to control the clock hands.

>STEP-01

Preparing the setup

We start the project by setting up the Raspberry Pi and Explorer HAT Pro. Install the libraries for the Explorer HAT Pro using the tutorial on the Pimoroni GitHub page here: **magpi.cc/2lOfDOk**. In this project we will be using two of the 5V DC outputs from the HAT to activate a relay. Note: These ports don't actually output 5V, but connect to ground when activated.

>STEP-02

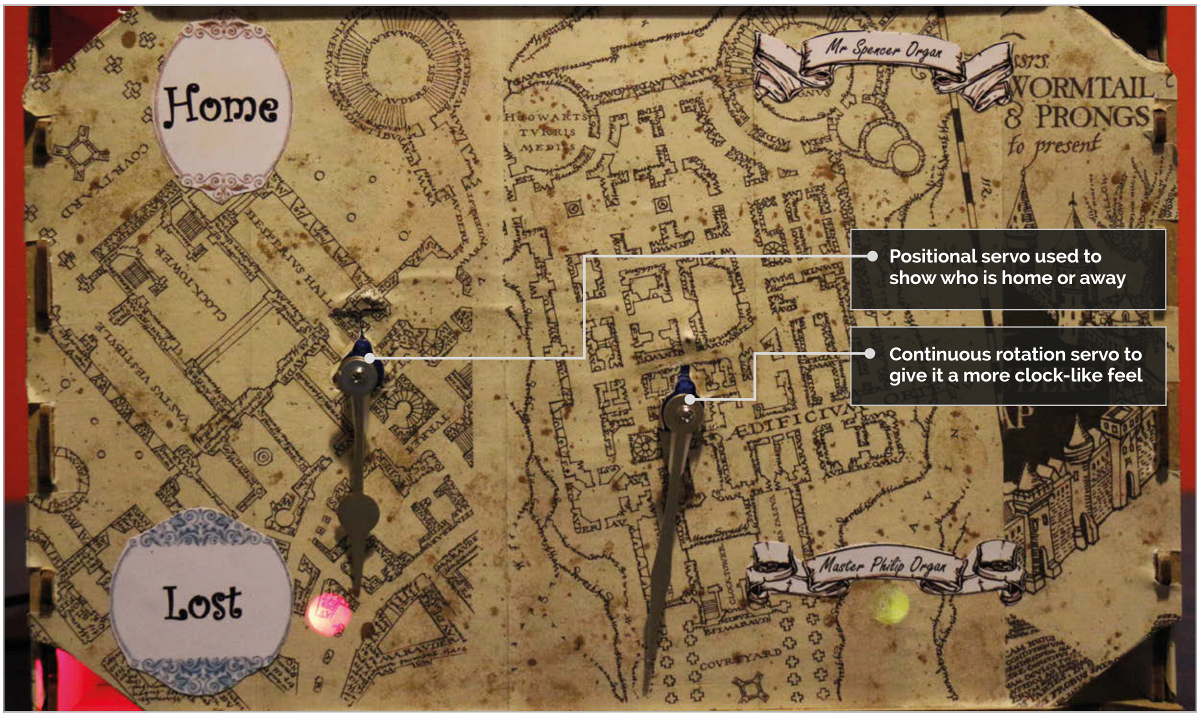
Configure the code

Clone the Python code from **magpi.cc/2lOfkTJ**. You will need to find the MAC addresses for each of the devices you will be detecting on the network, and substitute them into the code. It is important that all the devices are on the same network as the Pi. On an iOS device you can find your MAC address under Settings > General > Wi-Fi address; on an Android device, it's in Settings > About phone (or tablet) > Status.

>STEP-03

Connect the servos

We then need to connect the two servos to the Arduino. Use a breadboard to connect a common 5V and ground line from the Arduino. Connect the red power cables on each of the servos to the common 5V line, and the brown wire to the common ground. Use jumper cables to connect the orange signal wire of the continuous servo to digital pin 12 on the Arduino, and the orange signal wire of the position servo to digital pin 11. The continuous rotation servo will become our second hand on the clock face, and the position servo will show whether people are home or away.

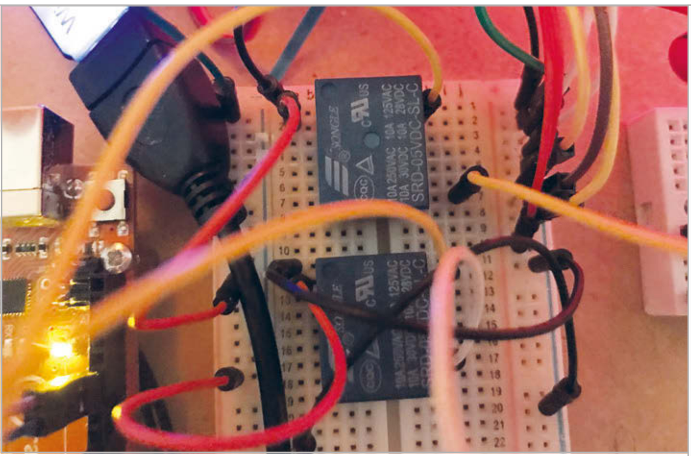


Positional servo used to show who is home or away

Continuous rotation servo to give it a more clock-like feel

TESTING THE CLOCK MECHANISM

The clock mechanism can be tested by connecting digital input 1 on the Arduino to ground.



Above The coil connectors of the relay are connected to 5V and output on the Explorer HAT Pro. The switch connectors are connected to digital input 1 (and 2 for the second relay) and the ground on the Arduino

>STEP-04

Connect the LEDs

We will be using four LEDs to light up different parts of the Marauder's map. Connect the short leg (cathode) to a common ground. Connect the long leg (anode) to digital pins 5,6,7, and 8.

The following LED pins are used:

Digital Pin 5	= Person 2
Digital Pin 6	= Away notification
Digital Pin 7	= Person 1
Digital Pin 8	= Home notification

>STEP-05

Link the Pi and Arduino

Before connecting the relays, we will need to look at the pin guide and identify the two pins for the coil and the two pins for the switch. One side of the coil on both relays should be connected to the 5V output on the Explorer HAT Pro. The other side of the of the relay coil should then be connected to output 1 and 2 on the Explorer HAT Pro. The common switch connector on both relays should then be connected to the common ground on the Arduino. The switch connector on the first relay is connected to Digital 1 on the Arduino, and the switch connector on the second relay is connected to Digital 2. We will be using the switch connector, which is activated when the relay receives power.

>STEP-06

Assembling the project

We will need to download the code for the Arduino and upload it to the Uno from here: **magpi.cc/2lOfkTJ**. We then need to assemble our project. In our example we have used a fruit box, but you could use an old clock, a cardboard box, or a hobby box. We need to (carefully) screw clock hands to the two servos; take care not to move the servo as it can cause damage. Mount the two hands so that that the continuous rotation servo can move freely and the positional servo can move up and down.

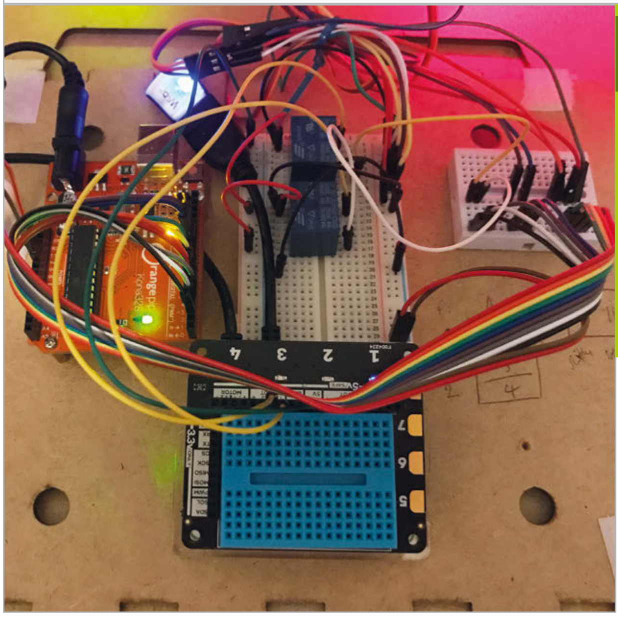
clock.py

Language

>PYTHON

DOWNLOAD:
magpi.cc/2lOfkTJ

```
!/usr/bin/env python

import subprocess
import time
import explorerhat

occupant = ["Person 1 device","Person 2 device"]

# MAC addresses for our devices - in order of people above
address = ["XX:XX:XX:XX:XX:XX","XX:XX:XX:XX:XX:XX"]

while True:
  print("starting loop")
  output = subprocess.check_output(
"sudo arp-scan -l", shell=True)
  print ("starting scan")
  for i in range (len(address)):
    if address[i] in output:
      print address[i]
      print occupant[i]

    if "XX:XX:XX:XX:XX:XX:XX"  in output:
      #Person 1 MAC address
      explorerhat.output[0].on()

    else:
      explorerhat.output[0].off()


    if "XX:XX:XX:XX:XX:XX:XX"  in output:
      #Person 2 MAC address
      explorerhat.output[1].on()

    else:
      explorerhat.output[1].off()
  time.sleep(60)
```



CONNECTING THE RELAYS

If the output from the relay is reversed (i.e. on when it should be off), try swapping the pole of the switch on the relay.

Left The Arduino, Raspberry Pi, and relays are mounted on the rear of the case with long jumper cables so that the box can be opened

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved…

NEED A PROBLEM SOLVED?

Email **magpi@raspberrypi.org** or find us on **raspberrypi.org/forums** to feature in a future issue.

RASPBERRY PI SECURITY

HOW SECURE IS THE RASPBERRY PI?

Password

Raspbian uses a default password that is quite well known – this is to allow kids to have easy access to their Raspberry Pis. Other operating systems use different passwords and even force you to set a new password after install.

SSH

By default, SSH is turned off on Raspbian. The default password and connection name are well known, so SSH could let people snoop in on the Pi and find out what you're doing. This can be turned back on very easily using the configuration tool.

Network

Since Raspbian and most Raspberry Pi OSs are run on Linux, they're very secure from the majority of network attacks. Especially with SSH turned off or the default server name changed, you're unlikely to find a problem day-to-day.

HOW CAN I IMPROVE SECURITY?

Change password

Changing the default password for your operating system is a good first step to adding further security to your Raspberry Pi: just make sure you remember what it is! To do this, open the Terminal and type `passwd`. Enter the current password and then the new one you want to use.

Encryption

The SD card stores all your data, and this can be easy to read on other computers or Raspberry Pis. You can easily encrypt the file system using a tool called eCryptfs. There's a good tutorial on how to do this, which can be found here: **magpi.cc/2nv57wr**.

HATs

There are a few HATs and add-ons that allow for further security, privacy, and encryption on the Raspberry Pi. These are very popular with enterprise users of the single-board computer and add a physical layer of protection to the Pi.

ONLINE SECURITY

Browser

The default browser in Raspbian is Chromium, which is the base of Google's Chrome OS. It's quite secure, and you can easily add pop-up blockers or extensions that deal with malicious ads. There's also the benefit that Raspbian runs on Linux, so website malware has less chance to affect your system.

Scratch

Online Scratch does allow you to create games and programs and publish them anywhere, but there is a login system that allows you to access your projects more easily. This can also be used to create a login system in your Scratch game.

Python

Python is able to scrape data off the web, such as weather or time or stock numbers. Be careful with the online information you use and make sure it's parsed through Python, which can make everything into a string of text. Try not to run this information outside the Python environment.

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

WHERE CAN I BUY A RASPBERRY PI?

You can buy a Raspberry Pi from our main distributors, Premier Farnell/element14 and RS Components/Allied Electronics. Both distributors sell all over the world. There are also many resellers of Raspberry Pis, both online and in bricks-and-mortar stores.

HOW MUCH DOES IT COST?

The Model A+ costs $20, the Model B+ costs $25, the Pi 2 costs $35, the Pi 3 costs $35, the Pi Zero costs $5, and the Pi Zero W costs $10, plus local taxes and shipping fees.

WHAT DO I GET WHEN I BUY ONE?

You get the Raspberry Pi board itself. A power supply and SD card are not included, but can be purchased at the same time from most places that sell the Raspberry Pi. You can also purchase pre-loaded SD cards. We recommend buying these from us or our licensed distributors rather than from third parties on eBay, as our software is updated regularly and cards sold by third parties can quickly become outdated.

WHY IS THE PRICE IN US DOLLARS? YOU ARE A UK COMPANY!

The components we buy are priced in dollars, and we negotiate manufacturing in dollars. Because currency markets are so volatile, we price the final board in dollars too, so we don't have to keep changing the price.

CAN I BUY A RASPBERRY PI KIT?

Raspberry Pi resellers produce some fantastic bundles for people who would rather get everything they need from a single source. In 2016, we put together our own Raspberry Pi Official Starter Kit. The kit is available to order online in the UK from our partners element14 and RS Components, priced at £99+VAT, and from distributors and resellers around the world.

RASPBERRY PI OS GALLERY

Take a look at every operating system available for the Raspberry Pi

The Raspberry Pi is amazing. We're probably a little biased, but it's hard to argue against it. While the hardware is great, it's important to not forget the software that makes it so fantastic, and at the very core of this is a good operating system.

There are a fair few to choose from on the Raspberry Pi. While you'll be safe with Raspbian the vast majority of the time, there's plenty of choice if you want to tweak your Raspberry Pi to be truly optimal for your build or project.

Not sure where to begin? Have no fear, here's our list of every current Raspberry Pi OS.

GLOSSARY

HARD FLOAT

Also known as armhf, this makes use of the Raspberry Pi's processor for maximum efficiency. This is preferred.

SOFT FLOAT

If not using hard float, an OS for Raspberry Pi is using soft float. This type of OS doesn't use the full hardware of the Pi and can reduce performance.

DISTRIBUTION

A specific release of Linux, which while not an operating system itself, can become one with programs added on top.

SPIN

A specific version of a distribution, usually with slightly different software included.

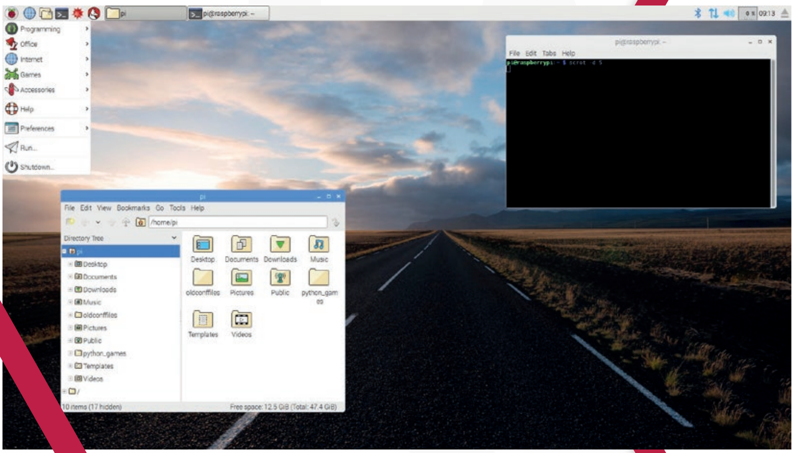
RASPBIAN

The all-purpose operating system

Let's start with the premier Raspberry Pi operating system, Raspbian. It's a specific spin of the ARM version of Debian 7.0 Jessie that is optimised for the Raspberry Pi. Debian is a very easy-to-use Linux distribution that is well supported: you have lots of software to choose from and it's simple to install.

The core Raspbian system is not maintained by the Raspberry Pi Foundation itself but it is heavily promoted, and used by, Raspberry Pi. It's maintained in such a way that means it is compatible with Raspberry Pis running on all three chipsets, allowing for some level of backward compatibility. It does run best on Raspberry Pi 3, though, where it has a little more power to make use of.

While Raspbian does by default log into a desktop environment, you can easily modify it to log into the command line instead. This option can be found with several other options for enabling or disabling services such as SSH and the Camera Module, making it very easy to modify.



```
pi@raspberrypi:~ $ scrot -d 5
```

= Raspbian

magpi.cc/2ejN6sk

INFO

- armhf
- Linux distribution
- Based on Debian 7.0 Jessie
- In active development
- Recommended by Raspberry Pi

RASPBIAN WITH PIXEL

This is the main release of Raspbian that you get from the Raspberry Pi Foundation. It includes PIXEL (Pi Improved Xwindows Environment, Lightweight) – this is basically a skin for the desktop environment, but it's very pretty and optimised for the Raspberry Pi.

It also comes with some extra flourishes like links to Raspberry Pi resources in the menus (and a link to *The MagPi*!), as well as some of the Raspberry Pi-specific software like GPIO Zero already installed. This full-size version of Raspbian is the most popular and the one the majority of tutorials will be written for.

RASPBIAN LITE

Lite is a minimal version of the Raspbian image for the Raspberry Pi. This means it has less software installed on it, and fewer modules will load with the kernel, which results in the operating system using a lot less of the Raspberry Pi's resources. It will use less electricity this way and perform a little faster for very specific tasks, such as file servers or other uses where it never needs to use a monitor.

PROS

- Official Raspberry Pi software and hardware work best on it
- Easy to use
- Easy to set up
- Most tutorials online use it

CONS

- SD card image is on the larger side
- Uses more resources than other operating systems

DESKTOPS AND ALL-PURPOSE

General-use operating systems that you can use for normal computer tasks

RISC OS OPEN

riscosopen.org

RISC OS is a blast from the past, a continuation of the same operating system that ran on Acorn computers in the 1990s. Very familiar to people around the age of 30, it feels right at home on ARM chips, as that's what the old Acorn computers used to run on. It's not super-useful for day-to-day desktop tasks as it uses a lot of very different software, but it's great it exists.

PI-TOPOS: POLARIS

pi-topOS:Polaris

pi-top.com

The pi-top is a Raspberry Pi-powered laptop (and later desktop) build that was successfully crowdfunded less than two years ago. It has a huge educational focus and for that, the pi-top team have created a special operating system called pi-topOS to go with the hardware.

It allows for easy access to coding apps like Python, and also includes lessons and worksheets to follow so you can learn Python. There's also Minecraft Pi, Scratch, and Sonic Pi, just like on Raspbian. It also comes with its own special educational game called CEED Universe – an RPG-esque game where you solve puzzles using code, along with puzzles that involve physical electronics. It's all set up with a universal login, with progress and such tied to your online account so you can access most of your data from any pi-top or Raspberry Pi running pi-topOS.

While it can be used like a desktop, it's a lot more angled towards learning to code than desktop use. This may be preferable if you're giving it to a young person, but for others looking for a more pure desktop experience on Raspberry Pi, this is maybe not the place to look.

INFO

- Built for pi-top computers
- Designed for education
- Comes with CEED Universe educational game
- Easy to hack with

PROS

- Interface looks similar to smartphones
- Great learning resources
- CEED Universe is great

CONS

- No access to extra Raspbian programming libraries
- Not the best desktop experience

UBUNTU MATE

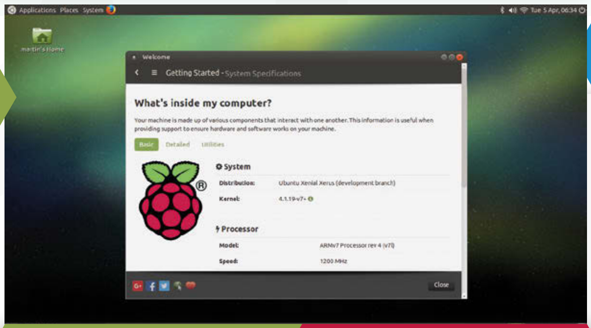
ubuntu-mate.org

Ubuntu is almost synonymous with Linux. It's one of, if not the most popular Linux distributions out there and it's managed to gain this reputation thanks to driving Linux-based operating systems to be more user friendly.

As well as the main version of Ubuntu, there are several popular spins of it that use different desktop environments along with other little tweaks and fixes, including Ubuntu MATE, which uses the more classical-looking MATE desktop environment. It's known for being lightweight and straightforward and it's gained popularity among the other spins on PCs for this very reason.

It's also the only desktop Ubuntu that works on Raspberry Pi, and while known for being lightweight and fast compared to its peers on PC, it's relatively resource intensive on the Raspberry Pi. That's not to say it is bad and slow – it's designed to work on the Raspberry Pi 2 and 3 and they have ample power to run it well enough. Raspbian runs just a little bit better, though.

It doesn't have all the amenities of Raspbian either – no GPIO Zero, no built-in Scratch, etc. It does have access to a wide range of software, though, and generally works a little better as a desktop PC.


Applications Places System
Tue 5 Apr, 06:34
martin's Home
Welcome
Getting Started - System Specifications
Basic
Detailed
Utilities
Close

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.

System

Distribution: Ubuntu Xenial Xerus (development branch)

Kernel: 4.1.19-v7+

Processor

Model: ARMv7 Processor rev 4 (v7l)

Speed: 1200 MHz

INFO

- Based on the very popular Ubuntu
- Debian core, like Raspbian
- More of a pure desktop experience
- Uses MATE desktop environment

PROS

- Good software library
- Good desktop experience
- Easy to get
- Well maintained

CONS

- Not as fast as Raspbian
- No access to extra Raspbian programming libraries

PURE LINUX DESKTOPS

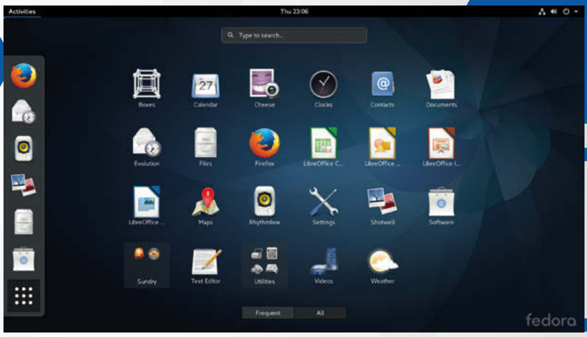
The stock Linux distributions built for Raspberry Pi

FEDORA

magpi.cc/2miaZUL

- ARM build of the popular Fedora OS
- Currently on version 25
- Made for Raspberry Pi 2 & 3
- Desktop-orientated as standard

Fedora is a classic Linux distribution much in the same way Ubuntu is, although Fedora is a little older. It's well known for using the latest and greatest Linux software, testing and using it before most other big Linux operating systems. While there have been a few Raspberry Pi specific Fedora spins in the past, these days stock Fedora is built using ARM and can run straight on a Raspberry Pi.



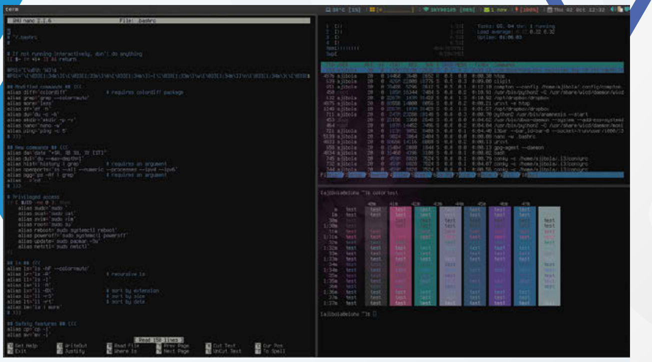
Fedora offers a clean desktop experience and is quite popular among programmers. It's also a good place to see the latest software available for the Raspberry Pi chipset.

ARCH LINUX

magpi.cc/2nf6YFk

- An early Raspberry Pi OS
- Very customisable
- Recommended for more advanced users

Arch Linux ARM is the spin of Arch that works on the Raspberry Pi – although there's a couple of versions specifically for the Raspberry Pi. One is optimised for the original versions (including the Pi Zero) while the other works well on Raspberry Pi 2 and 3. Arch is very barebones, offering a command-line interface as default and a minimal suite of software as standard. From here you can build it up and customise it as you wish, although there's a little less software available for Arch ARM than for the full PC version of Arch. It's great for software-heavy projects that need access to as many resources as possible from the Pi. Make sure you get the correct version for your Raspberry Pi chipset, though.



```
  GNU nano 2.x.x                 File: .bashrc

#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

## Modified commands ## {{{
alias diff='colordiff'              # requires colordiff package
alias grep='grep --color=auto'
alias more='less'
alias df='df -h'
alias du='du -c -h'
alias mkdir='mkdir -p -v'
alias nano='nano -w'
alias ping='ping -c 5'
# }}}

## New commands ## {{{
alias hist='history | grep'         # requires an argument
alias openports='ss --all --numeric --processes --ipv4 --ipv6'
alias pgg='ps -Af | grep'           # requires an argument
# }}}

# Privileged access
    alias sudo='sudo '
    alias scat='sudo cat'
    alias svim='sudo vim'
    alias root='sudo su'
    alias reboot='sudo systemctl reboot'
    alias poweroff='sudo systemctl poweroff'
    alias update='sudo pacman -Su'
    alias netctl='sudo netctl'

## ls ## {{{
alias ls='ls -hF --color=auto'
alias lr='ls -R'                    # recursive ls
alias ll='ls -l'
alias la='ll -A'
alias lx='ll -BX'                   # sort by extension
alias lz='ll -rS'                   # sort by size
alias lt='ll -rt'                   # sort by date
alias lm='la | more'
# }}}

## Safety features ## {{{
alias cp='cp -i'
alias mv='mv -i'
                    [ Read 158 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

GENTOO

magpi.cc/2neUygW

- Linux you can build, almost from scratch
- For advanced users familiar with Linux

Gentoo is a step beyond Arch. Whereas Arch gives you a basic, functioning OS that you can build upon, Gentoo has you build everything up as you go along. This goes as far as selecting kernel modules in the Linux kernel – core pieces of software that allow for different pieces of hardware to run. It's not for everyone, but if you have a mastery of Linux you can create a heavily customised operating system for your Raspberry Pi. Of course, you won't then have access to packages like GPIO Zero or Minecraft Pi.

SARPI

magpi.cc/2mirKip

- Supports hard float and soft float
- Slackware ARM on the Raspberry Pi
- Not an official release
- For advanced Linux users

While not an official release of Slackware, SARPi merely repackages Slackware ARM in a way that makes it installable on Raspberry Pi. Slackware is a very old Linux distribution that has been around for many years, focusing on stability above all. It comes with the most barebones install of all the Linux distros on this page – it doesn't even have a command-line interface as default. If you want to learn the nitty-gritty of Linux, running Slackware is a good place to start. It might take a little while to learn, though.

LIGHTWEIGHT OPERATING SYSTEMS

Make the most of your Raspberry Pi with these slimmed down OSs

DIETPI

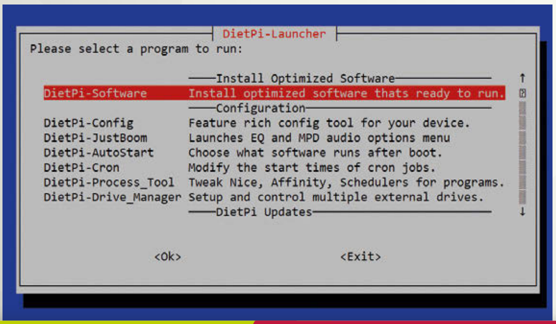
dietpi.com

- Lighter than Raspbian Lite
- Based on Debian Jessie
- Very light graphical interface over command line

Claiming to be three times lighter than even Raspbian Lite, DietPi is a very stripped down version of Raspbian, albeit with more than just a command line. It uses Whiptail menus – very simple graphical menus rendered in the CLI that barely use any extra resources, and definitely use a lot less than a traditional graphical desktop.

Being based on Raspbian means it does have access to a lot of the great programming software that comes with it – including easy libraries for controlling the GPIO ports with Python and such. One of the unique aspects of DietPi is its task manager, which allows you to easily tweak the priorities for different tasks running on your Pi – particularly useful if you've chosen DietPi to make sure certain tasks get as much power as possible.

It is also structured with certain types of project in mind, providing automation features and a comprehensive logging system. While this may all sound fairly advanced, it's actually quite easy to get your head around, while offering some easy-to-use advanced options.



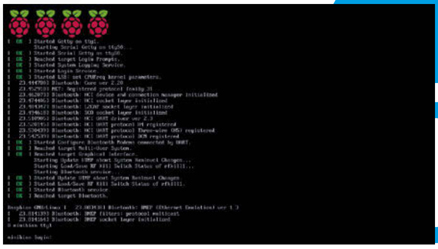
```
DietPi-Launcher
Please select a program to run:

                         ──Install Optimized Software──
  DietPi-Software        Install optimized software thats ready to run.
                         ──Configuration──
  DietPi-Config          Feature rich config tool for your device.
  DietPi-JustBoom        Launches EQ and MPD audio options menu
  DietPi-AutoStart       Choose what software runs after boot.
  DietPi-Cron            Modify the start times of cron jobs.
  DietPi-Process_Tool    Tweak Nice, Affinity, Schedulers for programs.
  DietPi-Drive_Manager   Setup and control multiple external drives.
                         ──DietPi Updates──

              <Ok>                        <Exit>
```

PROS	CONS
Embraces being lightweight	Whiptail menus are not for everyone
Raspbian core	Limited personal use in default state

MINIBIAN

magpi.cc/2mgkCDg

- Designed for embedded systems or server
- Command-line interface
- Based on Raspbian



```
[  OK  ] Started Getty on tty1.
         Starting Serial Getty on ttyS0...
[  OK  ] Started Serial Getty on ttyS0.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started System Logging Service.
[  OK  ] Started Login Service.
[  OK  ] Started LSB: set CPUFreq kernel parameters.
[   23.444790] Bluetooth: Core ver 2.20
[   23.452910] NET: Registered protocol family 31
[   23.462073] Bluetooth: HCI device and connection manager initialized
[   23.474486] Bluetooth: HCI socket layer initialized
[   23.484347] Bluetooth: L2CAP socket layer initialized
[   23.494618] Bluetooth: SCO socket layer initialized
[   23.510905] Bluetooth: HCI UART driver ver 2.3
[   23.520145] Bluetooth: HCI UART protocol H4 registered
[   23.530439] Bluetooth: HCI UART protocol Three-wire (H5) registered
[   23.542539] Bluetooth: HCI UART protocol BCM registered
[  OK  ] Started Configure Bluetooth Modems connected by UART.
[  OK  ] Reached target Multi-User System.
[  OK  ] Reached target Graphical Interface.
         Starting Update UTMP about System Runlevel Changes...
         Starting Load/Save RF Kill Switch Status of rfkill1...
         Starting Bluetooth service...
[  OK  ] Started Update UTMP about System Runlevel Changes.
[  OK  ] Started Load/Save RF Kill Switch Status of rfkill1.
[  OK  ] Started Bluetooth service.
[  OK  ] Reached target Bluetooth.

Raspbian GNU/Linux 8 [   23.803938] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[   23.814139] Bluetooth: BNEP filters: protocol multicast
[   23.814164] Bluetooth: BNEP socket layer initialized
 minibian tty1

minibian login:
```

Minibian takes a very similar approach to DietPi, although instead of offering a menu system like Whiptail, it's gone full command-line interface with the way it works. Once again, it's based on Raspbian and while it hasn't been updated in a year, the last release was created to make sure it was compatible with the Raspberry Pi 3.

It's very much designed as a release for people who want to use the Raspberry Pi as a server or an embedded system, the kind you'd find more commonly in businesses than at home. It is also compatible with all of the Raspbian software as well, allowing it to be used like Raspbian itself if the need arises. This release probably isn't for everyone, though.

PROS	CONS
Great for servers and headless projects	Command-line interface
Works with Raspbian software	Niche appeal

MOEBIUS

magpi.cc/2mgxJod

- Incredibly small
- Still hard float
- Designed for speed

Probably one of the smallest operating systems you can get for the Raspberry Pi, Moebius boasts that you only need a 128MB SD card to install it, and that it only takes up 20MB of RAM by default. The latest version is currently in beta testing, which adds support for Raspberry Pi 3 (although at the time of writing, the beta only works on Pi 3).

It's a much more extreme version of Minibian, even going so far as using some components from Raspbian to make sure it still makes use of hard float on the releases. It's an interesting OS, best suited to tasks that require as little power and space as possible. Hopefully the newer version will also work on Pi Zero for a truly minimal experience.

PROS	CONS
Only needs a 128MB SD card	Latest version still in development
20MB RAM usage	Extremely limited by default

KIOSK AND WEB BROWSERS

Looking to set up a public internet access point? Here's what could help…

CHROMIUM OS FOR ALL SBC

magpi.cc/2njuVeS

- Gives access to all Google products
- Uses cloud storage
- The core of Chrome OS laptops

Chrome OS should be familiar to a lot of people: it's Google's operating system based on the firm's browser and which comes loaded on its laptops and other computers. The idea is it's all cloud based, with storage via Google Drive and an office suite with Docs and such. It's had varying levels of success, but as it's built from the open Chromium base, it was open for being hacked a bit to work on Raspberry Pi.

Chromium OS for Raspberry Pi works just like Chrome OS, which is great, although it can be a little sluggish at times. For a web kiosk, you have to jump through extra hoops to get it working for multiple users. It seems more suited for home then, perhaps as a kitchen computer for quickly accessing recipes.

PROS

- Familiar interface
- Access to Google apps

CONS

- Active development seems to have stopped
- Tricky to modify for web kiosk use

LINUTOP

linutop.com

- Based on Ubuntu
- Paid-for web kiosk OS
- Compatible on PC and Pi

This is the only paid-for OS in this guide and it is something that a lot of businesses pay to use. Whereas with other operating systems you need to do some tweaking to get it working well as a kiosk, Linutop does all the hard work for you. They even sell boxes with OS pre-installed (the smallest of which is powered by a Raspberry Pi), requiring little setup to get it working.

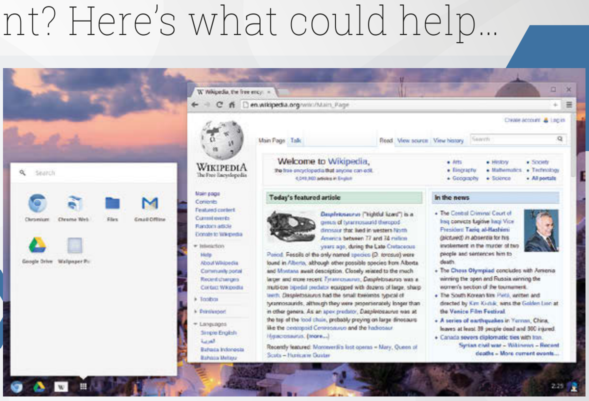
There's no login required with the Linutop either – you can just let people securely browse the web. All the settings are locked away and reset for each new session. They even provide remote-control software so you can manage your web kiosk from afar.

PROS

- Well supported
- Professional software

CONS

- Core OS is not Raspbian
- Not free



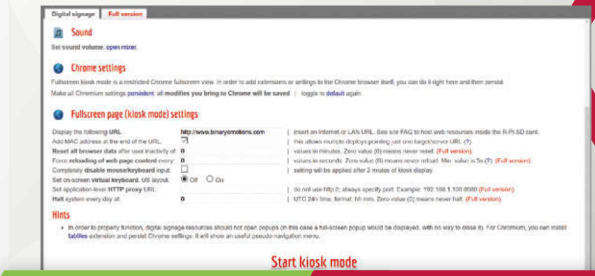
RASPBERRY WEBKIOSK

magpi.cc/2mgv3Xv

- Free web kiosk software
- Uses Chromium for browsing
- Allows for secure browsing from the Raspberry Pi

This is actually two operating systems in one – a pure, easy web kiosk OS for letting people browse the web in your cafe or hotel or whatever, and a digital signage version that is a touch more restrictive, only showing the browser itself.

Compared to Linutop, it is free and uses Chromium instead of Firefox for browsing: a minimal difference, but one that many people might prefer. It's unclear whether or not it is still being fully developed. The last release was in July 2016, so it supports the Raspberry Pi 3, which is probably the best model to use for a Pi-powered web kiosk.



Digital signage | Full version

Sound

Set sound volume: open mixer

Chrome settings

Fullscreen kiosk mode is a restricted Chrome fullscreen view. In order to add extensions or settings to the Chrome browser itself, you can do it right here and then persist.

Make all Chromium settings persistent: all modifies you bring to Chrome will be saved | toggle to default again.

Fullscreen page (kiosk mode) settings

Display the following URL:	http://www.binaryemotions.com	insert an Internet or LAN URL. See site FAQ to host web resources inside the R-Pi SD card.
Add MAC address at the end of the URL:	☑	this allows multiple deploys pointing just one target/server URL (?)
Reset all browser data after user inactivity of:	0	values in minutes. Zero value (0) means never reset. (Full version)
Force reloading of web page content every:	0	values in seconds. Zero value (0) means never reload. Min. value is 5s (?). (Full version)
Completely disable mouse/keyboard input:	□	setting will be applied after 2 minutes of kiosk display.
Set on-screen virtual keyboard. US layout:	● Off ○ On	
Set application-level HTTP proxy URL:		do not use http://; always specify port. Example: 192.168.1.100:8080 (Full version)
Halt system every day at:	0	UTC 24h time, format: hh:mm. Zero value (0) means never halt. (Full version)

Hints

- In order to properly function, digital signage resources should not open popups (in this case a full-screen popup would be displayed, with no way to close it). For Chromium, you can install tabIles extension and persist Chrome settings: it will show an useful pseudo-navigation menu.

Start kiosk mode

PROS

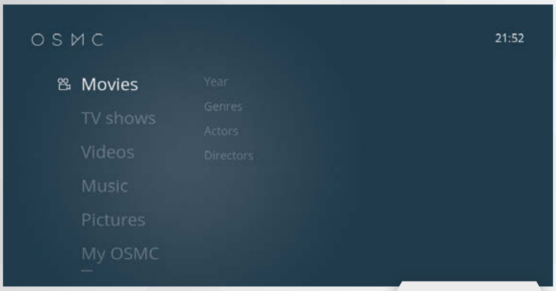
- Can support digital signage
- Free to use

CONS

- Core OS is not Raspbian
- No remote control

HOME ENTERTAINMENT

Whether you want to play some games or just watch some TV, there's a Pi OS for you



OSMC

osmc.tv

- Built on Kodi core
- Previously RaspBMC
- Uses custom interface by default

You'll hear a lot about Kodi on these pages – it's the media centre software used by many projects and products that allow you to play back media on your TV. It's an excellent core for any home theatre PC, so it's great that OSMC not only uses Kodi, but is completely up to date with the latest version.

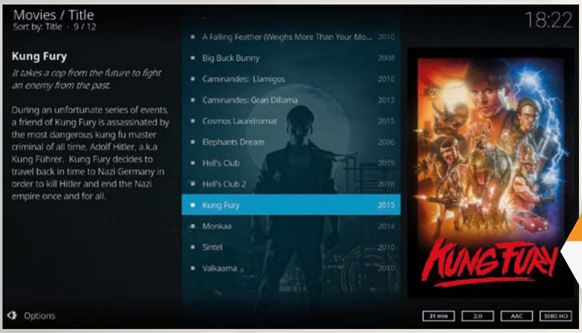
OSMC sets itself apart from stock Kodi by using its own interface as default, trying to make it easier to navigate your media or add it in the first place. Compared to its main alternative, LibreELEC, it's a little heavier resource-wise, although it doesn't make a difference in terms of actually playing your media. It'll run 1080p video with ease and is compatible with a huge array of codecs.

PROS

- Kodi base
- Playback is as good as the Pi can handle

CONS

- New default Kodi skin possibly better
- Fewer ties to Kodi dev team than LibreELEC


Movies / Title
Sort by: Title · 9 / 12

Kung Fury

It takes a cop from the future to fight an enemy from the past.

During an unfortunate series of events, a friend of Kung Fury is assassinated by the most dangerous kung fu master criminal of all time, Adolf Hitler, a.k.a Kung Führer. Kung Fury decides to travel back in time to Nazi Germany in order to kill Hitler and end the Nazi empire once and for all.

A Falling Feather (Weighs More Than Your Mo...	2010
Big Buck Bunny	2008
Caminandes: Llamigos	2016
Caminandes: Gran Dillama	2013
Cosmos Laundromat	2015
Elephants Dream	2006
Hell's Club	2015
Hell's Club 2	2016
Kung Fury	2015
Monkaa	2014
Sintel	2010
Valkaama	2010

LIBREELEC

libreelec.tv

- Close ties to Kodi dev team
- Successor to OpenELEC
- Stock Kodi experience

For a long time, OpenELEC was actually our Kodi OS of choice for the Raspberry Pi. Especially during the days of the original Raspberry Pi when resources were at a premium, having as light an OS as possible to run Kodi was great. Development has splintered a bit and LibreELEC has risen as the successor to OpenELEC, with just enough OS for Kodi.

It runs an almost pure stock Kodi, with a few extra LibreELEC menus on top to handle extra settings and updates and such. It also has the benefit of being optimised for the new interface for Kodi, which is definitely a marked improvement over the old one. It's still our preferred choice, especially on older Raspberry Pis and the Pi Zero.

PROS

- About as light an OS as possible for Kodi
- Great update system

CONS

- Slower off the mark with new Kodi releases
- AirPlay doesn't work well

XBIAN

xbian.org

- Veteran Raspberry Pi OS
- Based on Debian
- Has a custom installer

One of the original Kodi OSs on Raspberry Pi, Xbian has been around for nearly five years. It offers a stock Kodi experience on a pared-down version of Debian/Raspbian, which means you can then install other software underneath Kodi if you so wish. On the original Raspberry Pis this was a little risky as you only had limited resources, but with the Pi 2 and Pi 3 there's a bit more power to make use of.

Xbian still retains this ability, as well as being packaged with an easy-to-use installer that automatically burns an SD card for you. It gives you a bit more flexibility this way, in exchange for some optimisation, although you'll only notice issues on original Raspberry Pis and Pi Zero.

PROS

- Kodi base
- Easy install system

CONS

- Heavier than other Kodi OS
- Not great on Pi Zero

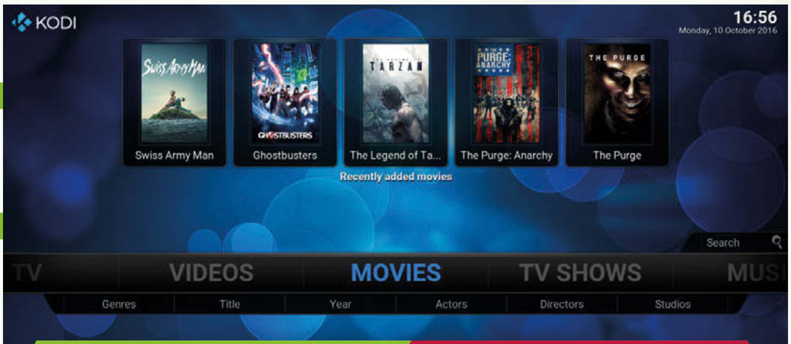
SLAXBMC

magpi.cc/2mgp451

- Slackware build with Kodi
- Comes with standard desktop
- Already built, just need to install

When we talked about Slack on an earlier page, we mentioned that it is great for building up the exact operating system you want. SlaxBMC is an excellent example of this: a Kodi OS for Raspberry Pi built on Slackware. You don't have to build it from scratch yourself, though: you can just grab the image and burn it to an SD card to use straight away.

SlaXBMC is a bit different in that it also lets you log into a standard graphical environment by exiting Kodi. This way you can turn your Pi into a living room PC if you wish. However, this does mean that it has a much larger image and uses a few more resources than other Kodi builds.



PROS

- Choice of install options
- Quick access to normal desktop

CONS

- Uses older Kodi
- Large image sizes
- Heavier on resources

RETROPIE

retropie.org.uk

- Easy to set up
- Retro game player
- Emulates tons of consoles and old PC games

The premier retro game emulation software for Raspberry Pi, RetroPie is the heart of many retro gaming projects. It allows emulation of basically everything pre-SNES and a few things after it, along with arcade games and some PC stuff like SCUMM. It did start off a little rough around the edges, but over the past couple of years it has sharpened up to become an amazing and easy-to-use piece of software.

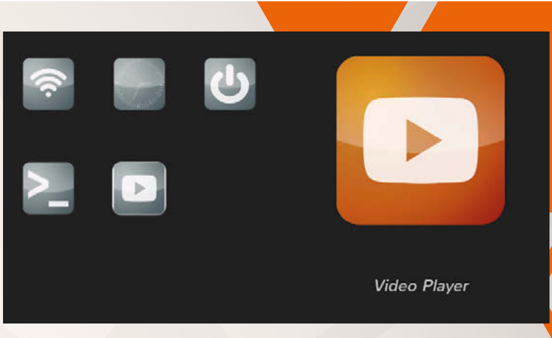
It also includes a few features for making sure it is kept up to date, and even supports netplay between RetroPie devices. No media playback at the moment, though (although it is being tested), so it's purely for games.

PROS

- Very easy to set up
- Extensive list of emulators

CONS

- No traditional media playback yet
- Need to source your own games



PIBOX

magpi.cc/2mgslMk

- Media server system
- Aimed more at business
- Back-end server and front-end player system

If you remember MythTV, you might be familiar with how PiBox works. A media server keeps all your media organised, which can then be securely sent to one or more media player boxes on the same network. Whereas with Kodi you're having to interact with Windows or Linux file servers, this is a lot more contained.

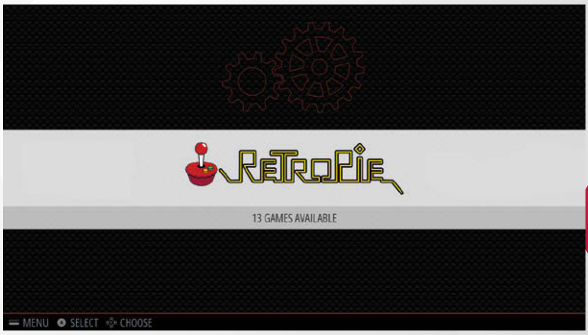
It does use the same hardware acceleration as Raspbian and Kodi does, and can even stream webcam video. It's not quite as easy to use overall as Kodi, although in many ways it offers more options in exchange.

PROS

- Makes use of hardware acceleration
- Same software for both ends of media playback

CONS

- No Kodi core
- Less easy to set up



PROVIDING A SERVICE

Cloud, server, or Docker – here's a selection of operating systems for different utilities

RASPBERRY OWNCLOUD

magpi.cc/2mu5HW4

- Uses popular ownCloud software
- Cloud storage solution
- Can stream media across network

OwnCloud is very good and while you could spend an evening setting it up on another Raspberry Pi distro, it's great to see it all ready to use with Raspberry ownCloud. It runs on ownCloud 9.1 and handily turns a Pi into a cloud server for your local network or beyond. It even connects to popular cloud storage providers like Google Drive and Dropbox. Additionally, it has UPnP and DLNA streaming set up so you can use it to stream media around your house.

PINET

pinet.org.uk

- Easy to use
- Available on NOOBS
- Centralised server for Raspberry Pi classroom

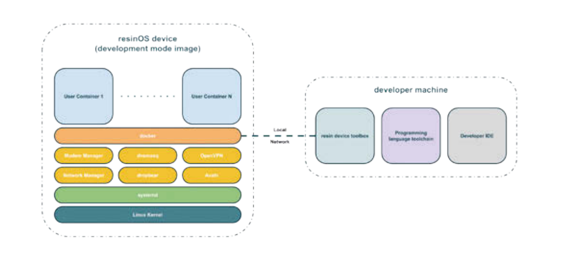
PiNet is a great piece of software, providing a server of students' user accounts in a classroom that uses Raspberry Pis to teach. It's all very well documented and kept up to date, and it means that kids don't have to carry their projects with them on SD cards. Outside the classroom, it could be used to create separate test environments for your software.

RESINOS

resinos.io

- OS for containers
- For Docker and dev work
- Powers resin.io

Docker and containers are excellent development and deployment tools, allowing you to test out different scenarios on different installs that are almost like virtual machines. ResinOS is an operating system for Raspberry Pi that allows you to use and create containers for Raspberry Pi installs. It's an excellent tool if you like to use more advanced Linux and programming methods.

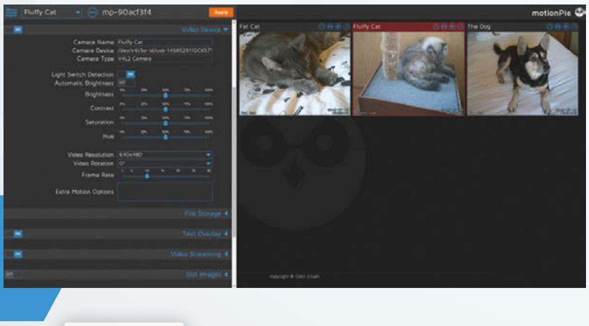


MOTIONEYEOS

magpi.cc/2nqLuFT

- CCTV software
- Allows for remote monitoring
- Motion activation features

We've used motionEyeOS in the past for a couple of projects and it's a great little operating system for any project that requires remote access to a camera. Whether it's full-on CCTV or just watching your cat flap, it's got a lot of great features to make sure all your surveillance needs are satisfied. It works with the official Camera Module as well as USB webcams, and supports multiple input streams.



Fluffy Cat mp-90acf3f4

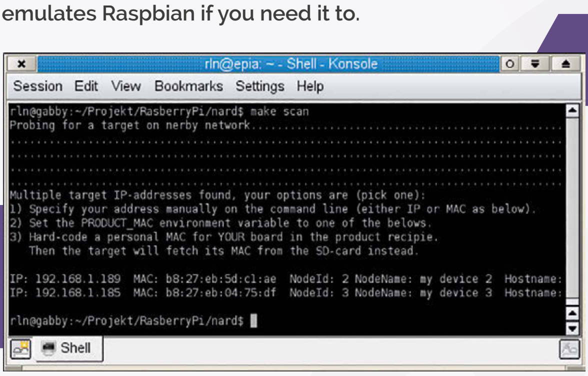
Camera Name: Fluffy Cat
Camera Device: /dev/v4l/by-id/usb-1458529110CK57
Camera Type: V4L2 Camera
Video Resolution: 640x480
Video Rotation: 0°

NARD SDK

magpi.cc/2nqWK4K

- Runs everything in RAM
- Designed for embedded systems

A very popular use for the Raspberry Pi is embedded systems – computers running all day, every day for great lengths of time that don't get looked at or serviced very often. These kinds of uses usually require a slightly different approach to the operating system and this is where Nard comes in. Importantly, it loads the system to memory so that if there's a power cut there's a much lower chance of the SD card becoming corrupted. It also has auto-repair tools for the SD card file system. It even emulates Raspbian if you need it to.



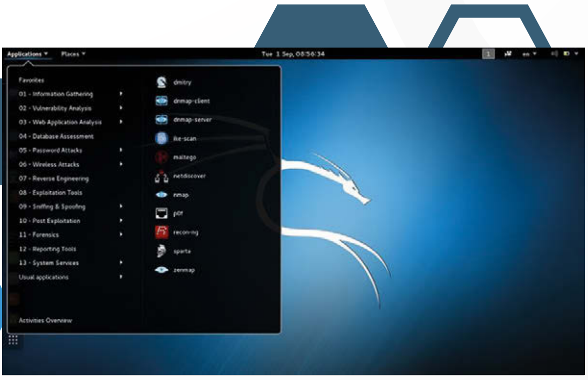
```
rln@gabby:~/Projekt/RasberryPi/nard$ make scan
Probing for a target on nerby network.........................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
Multiple target IP-addresses found, your options are (pick one):
1) Specify your address manually on the command line (either IP or MAC as below).
2) Set the PRODUCT_MAC environment variable to one of the belows.
3) Hard-code a personal MAC for YOUR board in the product recipie.
   Then the target will fetch its MAC from the SD-card instead.

IP: 192.168.1.189  MAC: b8:27:eb:5d:cl:ae  NodeId: 2 NodeName: my device 2  Hostname:
IP: 192.168.1.185  MAC: b8:27:eb:04:75:df  NodeId: 3 NodeName: my device 3  Hostname:

rln@gabby:~/Projekt/RasberryPi/nard$ 
```

KALI LINUX

magpi.cc/2nqTdUl

- Pen testing distro
- Used for testing security
- Portability on Raspberry Pi is useful

Kali has a reputation as one of the better pen testing Linux distros, aimed at helping people find vulnerabilities in their systems. Portability of Kali is pretty key, so a Raspberry Pi with Kali installed is a great tool for checking network vulnerability issues as well as problems with individual machines. The Raspberry Pi image is a streamlined version of Kali, but there is a full version you can install if you need it.



INTERNET OF THINGS

Use these special operating systems to power your projects

WINDOWS 10 IOT CORE

magpi.cc/1qOEEb1

Windows-based IoT has come a long way and the Windows 10 IoT Core offers a lot to the Raspberry Pi. You can program it using Visual Studio on a Windows 10 laptop, before uploading it to a Pi to control your project. You can also remotely monitor your project from your laptop.

SNAPPY UBUNTU CORE

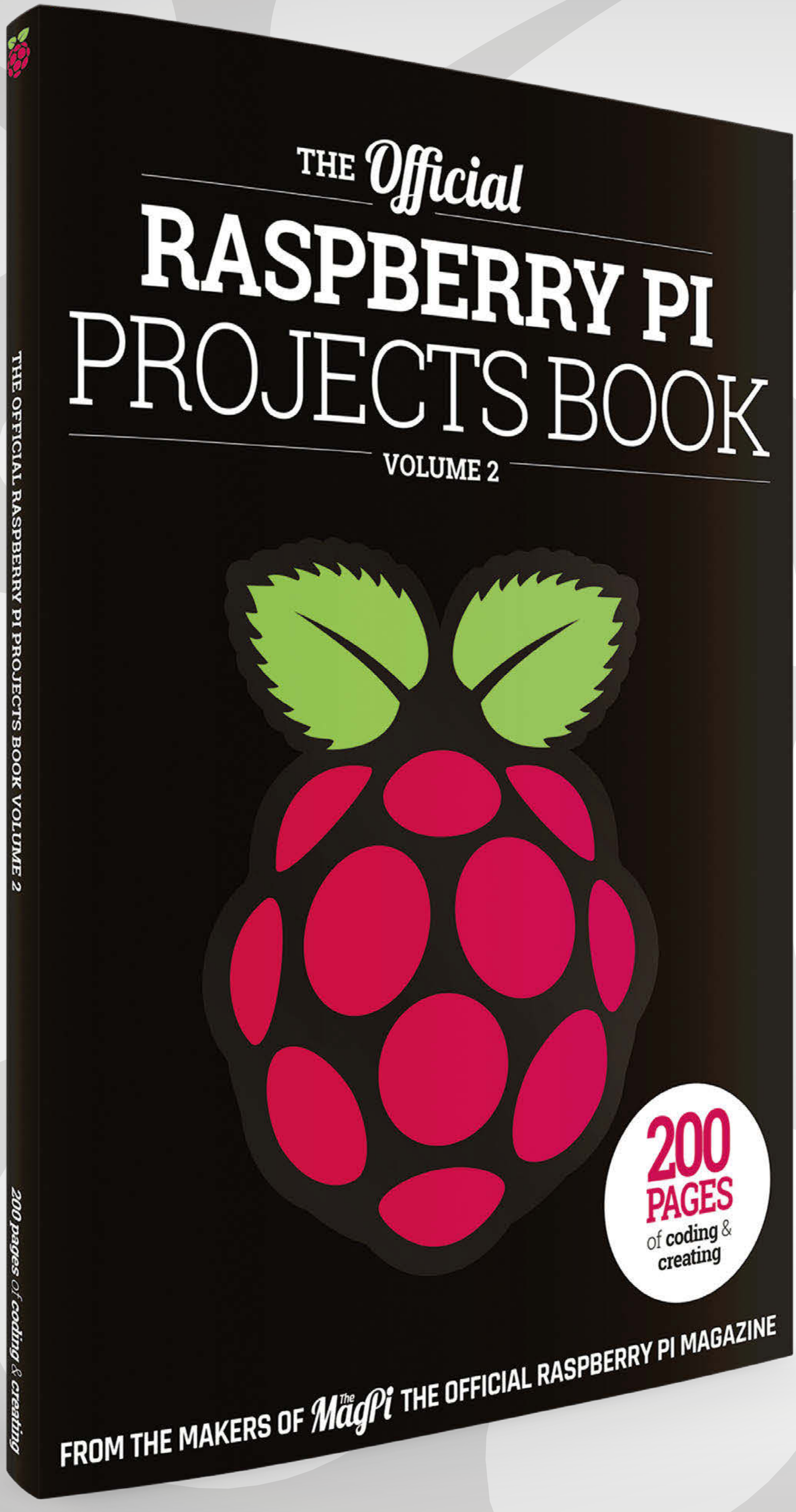
magpi.cc/2mvRB6U

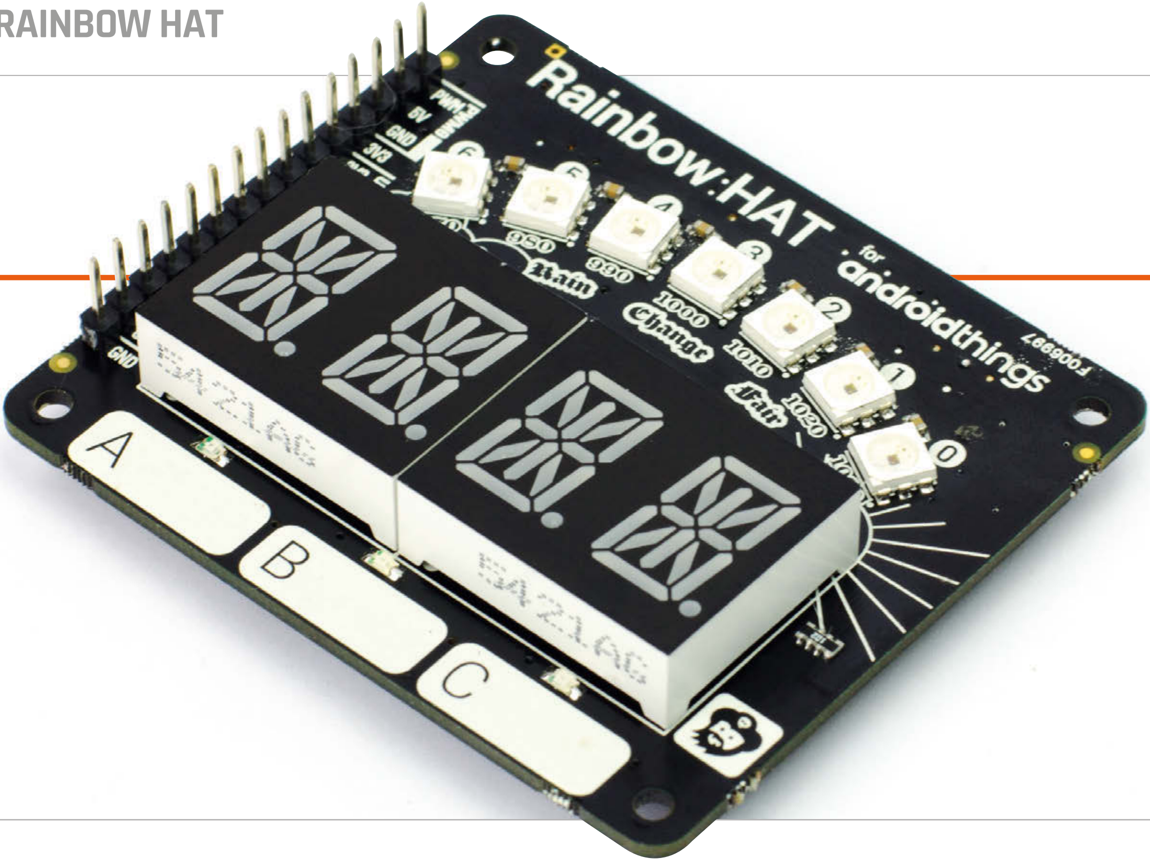
An ultra-lightweight version of Ubuntu that can be used for IoT, but also for embedded systems, cloud servers, and more. As it's Linux, it may be more familiar to people who are already programming and using Raspberry Pi in their projects. It's well supported and full of great software packages.

ANDROID THINGS

magpi.cc/2nqUFGi

Android powers a lot of hardware these days, so it's no surprise to see this very stripped-down version designed for embedded systems and IoT available for the Raspberry Pi. We used it in the review for the Rainbow HAT this issue and if you have Android dev experience, it's easy to pick up.


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 2
200 PAGES of coding & creating
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 2
200 pages of coding & creating



Maker Says

“A buffet of sensors, inputs, and displays to explore Android Things

Pimoroni

RAINBOW HAT

A versatile HAT to try out Android Things or use with Python

Originally known as Project Brillo, Google’s recently relaunched Android Things operating system is designed to be used with IoT devices including the Raspberry Pi 3. To save you the trouble of connecting various individual electronic components to your Pi to try out the new system, Pimoroni has designed the all-singing, all-dancing Rainbow HAT. Jam-packed with LEDs, buttons and sensors, it enables you to experiment with Android Things and use it with the wide range of protocols available on the Raspberry Pi, such as GPIO, I²C, SPI, and PWM.

To show numbers and letters, there’s an I²C-connected four-digit alphanumeric display, which is a cut above the standard type with 14 segments per digit, enabling it to accurately show the full alphabet. GPIO is used for the three capacitive touch buttons and their built-in LEDs (red, green, and blue respectively). The most visually impressive feature is the SPI-operated ‘rainbow’ arc of seven super-bright APA102 RGB LEDs that gives the HAT its name. PWM is employed for the piezo buzzer, which can be used to play buzzy tunes. I²C is also used for the BMP280 sensor, which measure temperature and pressure. Finally, there’s a line of 15 breakout pins for servo, I²C, SPI, and UART – all powered by 3V3. One notable omission, however, is an accelerometer/magnetometer, as featured in Pimoroni’s Enviro pHAT – possibly there wasn’t room on the board for one.

The Rainbow HAT can be programmed using Python in Raspbian, but we first tried it out with Android Things (see boxout for more details). Since the coding – via a remote computer – is mainly done in Java, it helps if you have a working knowledge of the language. Alternatively, like us, you can import one of the example projects from the Android Things GitHub repo (**magpi.cc/2mFNxTe**). The first time you do this, you’ll be prompted to install and update various tools and plug-ins, which is a bit of a hassle, but after that it works smoothly.

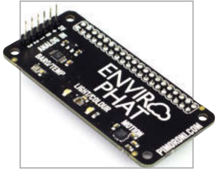
Android examples

To start with, we tried out the sample-button app from the Android Things repo. All it does is light up the LED of the Rainbow HAT’s capacitive button A when you press it. By digging into the code, you can change the GPIO pin numbers to use a different button and LED, although you’ll need to refer to **pinout.xyz** to find the correct numbers. Next, we imported the weather station project, which showed off more of the HAT’s capabilities. By default, the segment display displays the current temperature, although

Related

ENVIRO PHAT

This Zero-size board features four different sensors, for measuring temperature, pressure, light level, colour, motion, and direction, plus analogue inputs.

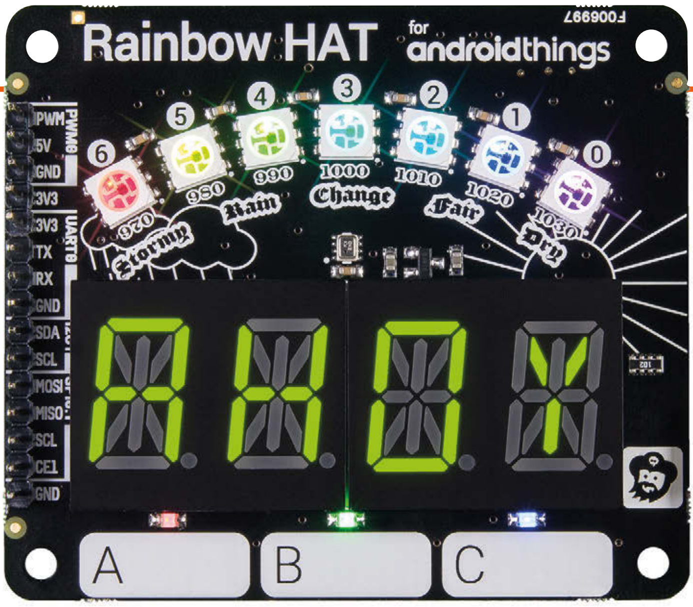


£16 / $20

magpi.cc/29NHB3T

magpi.cc/2lX1r6h

£24 / $25



we soon noticed it rising. As usual with a board-mounted sensor, this is due to the heat from the board and the Raspberry Pi just below it, but you could always calibrate it by comparing the real ambient temperature (using a standard thermometer) to discover the average difference. A press of a button switches the display to pressure, which is also indicated by the rainbow arc of LEDs. You can also configure this example app to publish sensor data to Google Cloud Pub/Sub, from which you can process it and visualise it using various Google tools.

Alternatively, in the Raspbian OS, Python can be used to program the HAT. There's a dedicated library, added using the one-line installer. Example programs include a pirate-themed demo that plays the *What Shall We Do with the Drunken Sailor?* sea shanty on the piezo buzzer while gradually changing the colour of the rainbow arc and showing AHOY, YARR or GROG on the alphanumeric display depending on the capacitive button pressed! Other examples include a temperature gauge and an RGB/HSV lamp whose shade is set using the touch buttons.

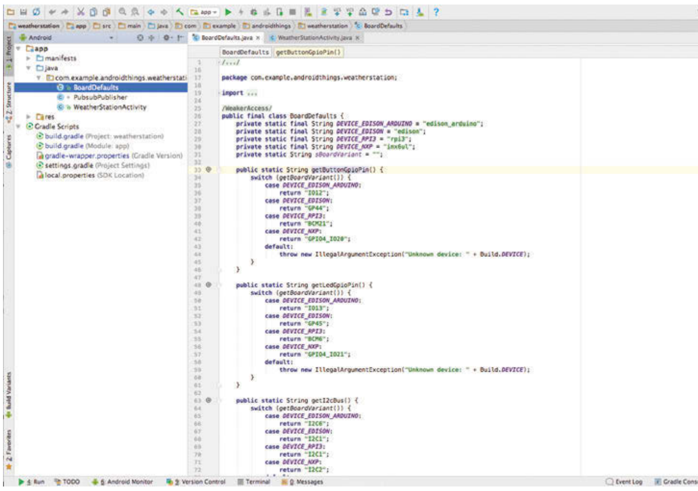
A look at the library example code – or the getting started tutorial at **magpi.cc/2lX6Jih** – reveals a simple syntax for using the HAT's features, such as **`weather.temperature`** to obtain temperature data and **`display.print_str('AHOY')`** to show a string on the four-digit alphanumeric display.

Whether programmed using Python or Android Things, the Rainbow HAT has an impressive number of features crammed into it, making it useful for all sorts of projects. Suggestions from Pimoroni include a weather station, stopwatch or countdown timer, three-note piano, reaction time game, and mood light.

> “The most visually impressive feature is the SPI-operated ‘rainbow’ arc”

Installing Android Things

Once you've downloaded the latest preview image of the Android Things OS (from **magpi.cc/2lWvvPg**), you can flash it to a microSD card to insert in the Raspberry Pi. On booting up, it shows a logo with network connection information underneath – you'll need to hook the Pi up to the router via Ethernet at first, as there's no way to set up wireless connectivity beforehand. With the Android Studio application installed on a remote computer, you can then connect to the Pi from a terminal using the adb tool from the SDK. You can then start building an app in Android Studio and deploy it to the Pi.



```
/.../

package com.example.androidthings.weatherstation;

import ...

/WeakerAccess/
public final class BoardDefaults {
    private static final String DEVICE_EDISON_ARDUINO = "edison_arduino";
    private static final String DEVICE_EDISON = "edison";
    private static final String DEVICE_RPI3 = "rpi3";
    private static final String DEVICE_NXP = "imx6ul";
    private static String sBoardVariant = "";

    public static String getButtonGpioPin() {
        switch (getBoardVariant()) {
            case DEVICE_EDISON_ARDUINO:
                return "IO12";
            case DEVICE_EDISON:
                return "GP44";
            case DEVICE_RPI3:
                return "BCM21";
            case DEVICE_NXP:
                return "GPIO4_IO20";
            default:
                throw new IllegalArgumentException("Unknown device: " + Build.DEVICE);
        }
    }

    public static String getLedGpioPin() {
        switch (getBoardVariant()) {
            case DEVICE_EDISON_ARDUINO:
                return "IO13";
            case DEVICE_EDISON:
                return "GP45";
            case DEVICE_RPI3:
                return "BCM6";
            case DEVICE_NXP:
                return "GPIO4_IO21";
            default:
                throw new IllegalArgumentException("Unknown device: " + Build.DEVICE);
        }
    }

    public static String getI2cBus() {
        switch (getBoardVariant()) {
            case DEVICE_EDISON_ARDUINO:
                return "I2C6";
            case DEVICE_EDISON:
                return "I2C1";
            case DEVICE_RPI3:
                return "I2C1";
            case DEVICE_NXP:
                return "I2C2";
```

Above Android Studio helps you program the Rainbow HAT

Last word

A potpourri of LEDs, buttons, and sensors makes the Rainbow HAT very useful for experimenting with Android Things, or programming with Python in Raspbian. It may be overkill for some projects, but it's a versatile add-on that offers good value for money.

★★★★☆

magpi.cc/2mHACOX

£112 / $119

Maker Says

Fully compatible hardware and software

Western Digital

WD PIDRIVE COMPUTE CENTRE

Build a Raspberry Pi computer with a 375GB hard drive

Western Digital has been bringing its hard drive range to the Raspberry Pi thanks to a series of PiDrive devices.

These combine the Raspberry Pi microSD card with a larger WD hard drive to provide a Pi with a beefed-up storage system.

The WD PiDrive Computer Centre features a Raspberry Pi 3 and a WD PiDrive Foundation Edition 375GB package. You also get a wireless keyboard and mouse and a plastic case to hold both the Raspberry Pi and WD hard drive.

Assembling the hardware proved easy enough. Simply screw the Raspberry Pi board to the base of the case and then screw the hard drive in next to it. A USB cable splits out to provide power to the Raspberry Pi and the hard drive. Included are a power adapter and all cables.

The lid comes with a magnetic top, so it's securely slotted to enclose the drive, but can be quickly removed to access the Pi and hook the GPIO pins up to electronic components.

Foundation Edition

It's the Foundation Edition software that makes this system sing.

It would have been easy just to produce a kit that uses the hard drive as an external drive to a regular Raspberry Pi.

However, WD has taken the time to produce a custom edition of NOOBS that works directly with the main drive.

You select SDA1 during the installation process, and it walks through the process of installing Raspbian with PIXEL. From then on you have a version of Raspbian running on the full 375GB hard drive.

Not only that, but WD has produced a custom edition of BerryBoot alongside its Foundation Edition Software (**magpi.cc/2ns5lnA**). BerryBoot's boot selection screen makes it easy to add multiple operating systems to your Raspberry Pi, and the ample storage gives them space to breathe.

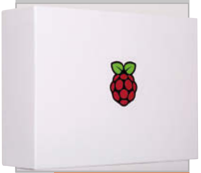
We added Ubuntu Mate 16.04, OpenELEC, and RetroPie, all alongside Raspbian. Each OS had plenty of space for files.

All in all, we liked this kit. Adding a large hard drive transforms the Raspberry Pi, and it's all enclosed in a nice mini desktop-style case.

Related

OFFICIAL STARTER KIT

If you want a Raspberry Pi computer kit but you're not interested in the larger hard drive, then take a look at the official kit instead.



£99 / $139

magpi.cc/2nrY33i

Last word

A lovely piece of kit that offers excellent Raspberry Pi integration with a larger hard drive.

★★★★★

magpi.cc/2l2BrFT

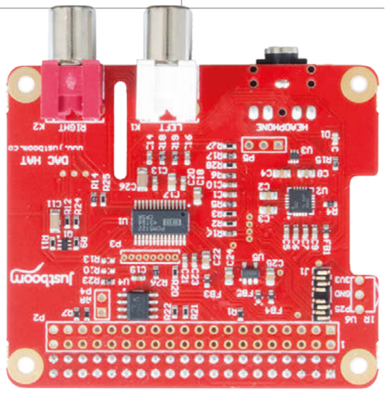
£30 / $40

Maker Says

A plug-and-play, high-resolution, digital-to-analogue converter for the Raspberry Pi

JustBoom

JUSTBOOM DAC HAT



Music sounds better with a HAT on

The JustBoom range of Pi products comprises three main types – DAC, Amp, and Digi – featuring different sets of outputs for different uses. Each is available in full-size HAT and Pi Zero-size pHAT form, ready-assembled. All are based on the I²S (inter-IC sound) standard.

Here we're taking a look at the DAC HAT, which delivers its sound via an amplified 3.5mm headphone socket, or twin RCA outputs connected to a hi-fi system or powered speakers. While the HAT features a 384kHz/32-bit DAC chip (the TI PCM5122), this is limited by the Pi's Linux drivers to 192kHz. Nevertheless, the sound quality proved mightily impressive in our tests. Both CD-quality 16-bit tracks and 24-bit HD audio files were played with crystal clarity, even via a modest Sony mini hi-fi system. Classical music in particular benefits from the enhanced clarity and detail.

The DAC HAT comes pre-assembled, so you just need to mount it securely on the Raspberry Pi using the supplied plastic spacers and bolts. As well as slots for Camera Module and touchscreen cables, this well-designed HAT features a full (unpopulated) GPIO breakout. The packaging includes an optional IR receiver, for you to solder on if you want to use a remote control. There's also the option to fit the HAT in a sleek plastic case (£13 / $17).

It's then a case of deciding which OS to use. A whole host are supported, including Volumio, OSMC, OpenELEC, Roon, and Max2Play (for which a 30-day free trial is included). Most are easy to set up with the HAT. In Raspbian, it involves commenting out one line in **/boot/config.txt** and adding three, one of which enables playback from Sonic Pi. A quick reboot and it's ready to use, although we found we needed to boost the digital volume in alsamixer for headphone playback.

One final trick up the DAC HAT's sleeve is the ability to stack a standalone Amp board (£60 / $79) on top, featuring block terminals so you can connect it directly to passive speakers. You'll need an additional power supply to get full power (30W RMS) out of it.

Related

IQAUDIO PI-DAC+

Based on the same TI PCM5122 chip, IQaudIO's HAT delivers similarly impressive sound quality. It can be stacked with a Pi-AMP+ to use passive speakers, too.

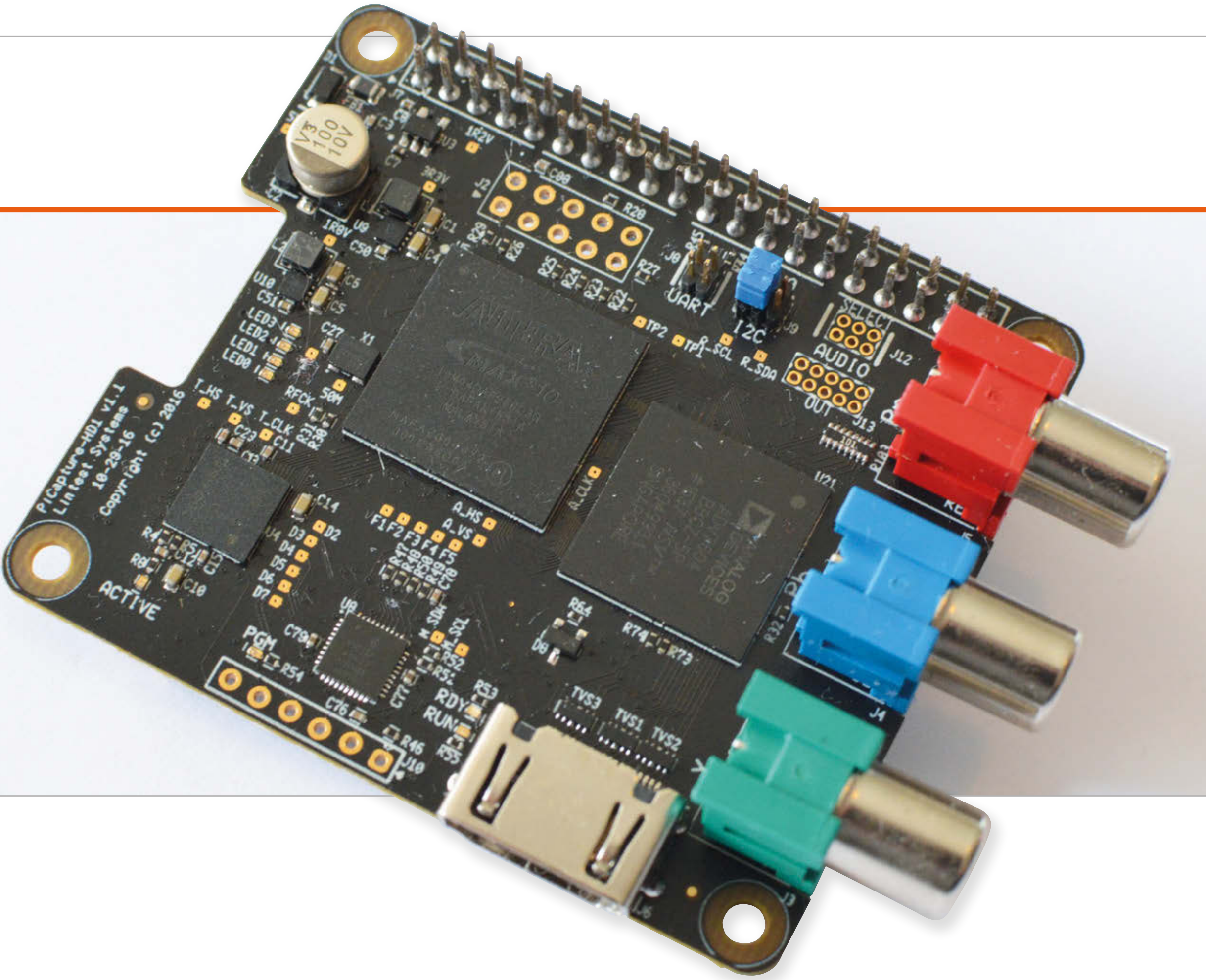
£30 / $40

magpi.cc/2mFj87h

Last word

Delivering audiophile sound at an affordable price, the well-designed DAC HAT is an impressive piece of kit. It offers a huge improvement in quality over the Pi's standard 3.5mm audio jack, and sounds particularly impressive when playing back lossless formats.

★★★★★



Maker Says

High-definition, low-latency video capture for the Raspberry Pi

Lintest Systems

PICAPTURE HD1

Capture video using your Raspberry Pi with this inexpensive capture card that also doubles as a HAT

This is a very interesting kit/HAT for the Raspberry Pi. Video capture usually requires machines that offer a decent amount of power so they can stream or convert an input signal into a video. However, due to the way the PiCapture is built and optimised, it connects straight to the Raspberry Pi's VideoCore graphics processor, and is able to make full use of it for video capture.

This is the HD version of the PiCapture, which supports input via HDMI and component video. The HD and SD versions share the same connection to the Pi through the GPIO ports, and via the camera connector (CSI) port. This is important, as the CSI has a direct (and very high bandwidth) connection to the main BCM chip on the Raspberry Pi, which contains VideoCore. Installed like a HAT on the Raspberry Pi, PiCapture draws power from the GPIO ports, and can connect via I²C or UART. It also handily extends the GPIO pins above the board so you can use them for other tasks.

It's an incredibly neat and well-thought-out design, and because VideoCore is the same in all versions of the Raspberry Pi, it will work on any Pi with a 40-pin GPIO port – even a Model A+. Raspberry Pi Zeros aren't included on the spec sheet, but with a little modification and some creative balancing (it's quite a bit wider than the Zero is), you could probably get it to work. With the form factor the way it is, though, you'd be better off with an A+.

With the hardware all attached, you can then record your video input. As this is done through the camera connector, you can use Camera Module software to preview and record the video; no extra software needed! It's an incredibly smart way to do it as it also means it's instantly usable in Python for more advanced projects and uses. A typical use would be to record video, and show a preview at the same time, so that you can watch the preview on your monitor or TV during recording.

Related

FEBON100 UVC CVBS

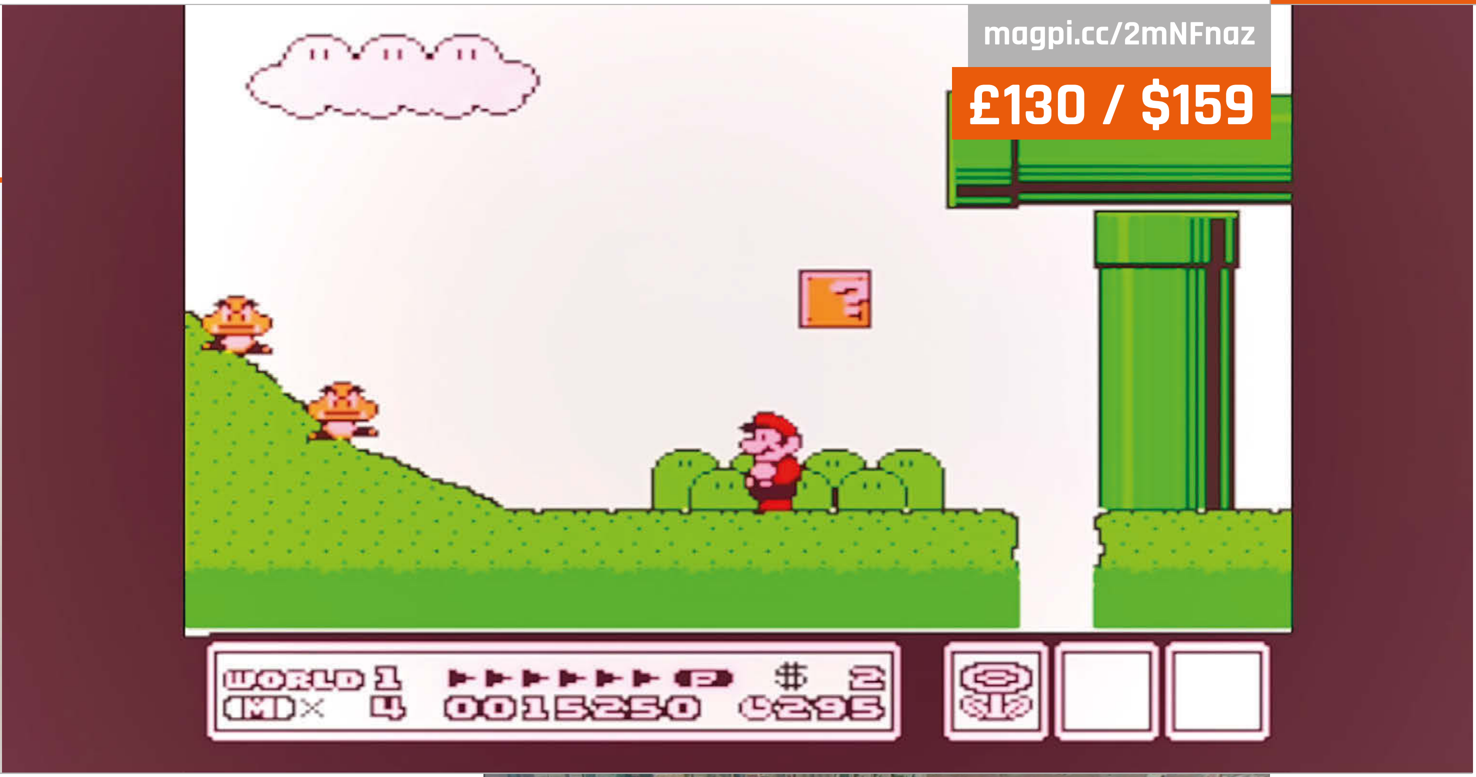
There are a few USB capture cards that will work with the Raspberry Pi, although these don't support very high resolutions.

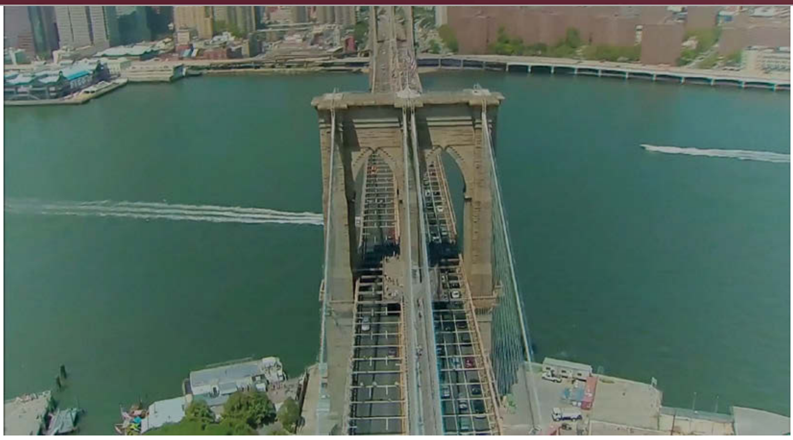
£62/$77

magpi.cc/2moR0s8

magpi.cc/2mNFnaz

£130 / $159





Above The colours are washed out and quite different from the original here

Left Depending on the video, the colour problems may not be as apparent, but they're still there

As well as Python, you can use raspistill/raspivid on it, which is pretty good for making sure the input is working on the board.

This is unfortunately where the PiCapture's limitations start to show. The video is smooth, the picture appears at the correct resolution, and there's no discernible input lag, even while playing games. However, the colours are all a little bit off.

It's a bit of a deal-breaker. With the right kind of video it won't be noticeable, and if you're a dab hand with the likes of Adobe Premiere, you should be able to tweak the colours to be more like the original. There's also the matter of there being no sound recording capability, so you'll have to marry up externally recorded sound with the video in an editing program. Unfortunately, there is no way for the CSI input to take sound as well as video, so it's not the PiCapture's fault.

> Compared to getting a full rig and video capture setup, this is a lot cheaper

It's a little bit of a shame. Yes, compared to getting a full rig and video capture setup this is a lot cheaper, but it does require a little more work to make it look anywhere near as good. For content that doesn't require exact colours it will do the job, but don't expect to record perfect 1080p60 video from a HDMI source, especially as at 1080p, you will be limited to a 30fps recording.

For what it is, it's still pretty amazing. It's just limited by the mixture of hardware it is using to complete the task. As it works with Python, it could still be used to monitor specific video streams for motion, for example in a CCTV project. It also theoretically means it will work a bit better with the input from something like a DSLR or other video camera, as it feeds it straight through the CSI port.

Last word

A unique and well-thought-out piece of kit that unfortunately stumbles somewhat when it comes to the actual task of video capture.

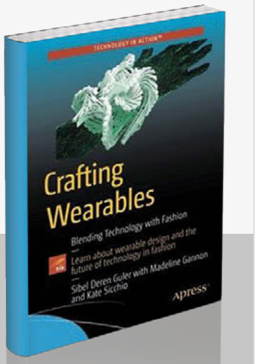
★★★☆☆

RASPBERRY PI BESTSELLERS

APRESS TECHNOLOGY IN ACTION

Apress brings a diverse range of maker topics to its 'Technology In Action' series

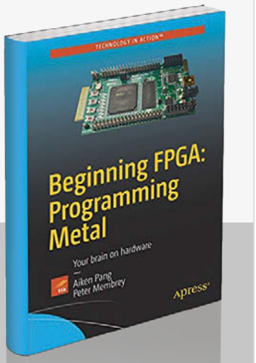
CRAFTING WEARABLES: BLENDING TECHNOLOGY WITH FASHION


Crafting Wearables
Blending Technology with Fashion
Sibel Deren Guler with Madeline Gannon and Kate Sicchio
Apress

Author: Sibel Deren Guler
Publisher: Apress
Price: £22.99
ISBN: 978-1484218075
magpi.cc/2nfbQKS

Guler's scattergun approach to this diverse topic has interesting facts and ideas in every section. It touches on the elementary skills you need, leaving you with more questions – and the urge to find the answers.

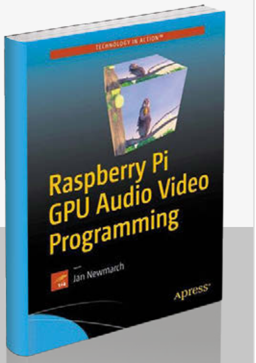
BEGINNING FPGA: PROGRAMMING METAL



Author: Aiken Pang
Publisher: Apress
Price: £29.50
ISBN: 978-1430262473
magpi.cc/2nfkUPK

If you want to really grasp logic gates, computer hardware, and close-to-the-metal programming, field-programmable gate arrays are your path to enlightenment. Challenging but fun.

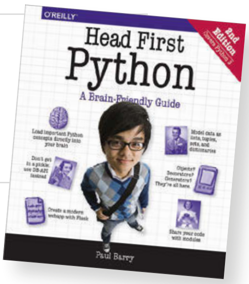
RASPBERRY PI GPU AUDIO VIDEO PROGRAMMING



Author: Jan Newmarch
Publisher: Apress
Price: £22.99
ISBN: 978-1484224717
magpi.cc/2nfenEA

The missing manual for GPU programming on the Pi – particularly with OpenMAX and OpenVG. You'll need a reasonable grasp of C, but Newmarch gives you everything else. Very code heavy.

HEAD FIRST PYTHON



Author: Paul Barry
Publisher: O'Reilly
Price: £39.99
ISBN: 978-1491919538
magpi.cc/2nfiscb

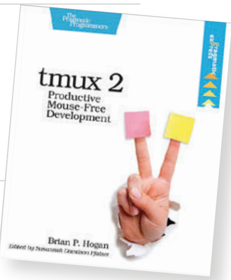
Whether or not you like the quirky, visual style of O'Reilly's Head First books, this is an excellent introduction to Python, well written and well paced. Aimed at those with at least a little coding experience in another language, Paul Barry's text starts iconoclastically by sidestepping Hello World, and diving into a more in-depth first program, bringing in the list data structure, importing from the standard library, and introducing the powerful *in* operator.

The data-first approach, with lists, then dictionaries, tuples, and sets, is a firmer foundation than starting with control flows. Functions follow, and making your own modules, which you then install in the cloud at PythonAnywhere, to support the web app you write with Flask. Difficult subjects like decorators, comprehensions, and generators are painlessly introduced, as our app is constantly refactored, and communicating with an SQL database is carefully taught.

PEP8 gets an early mention, but otherwise testing is left. Nor is there anything on Jupyter or doing data science, for example, but this is a book that teaches a lot, very quickly, and confident learners who find other introductory books too slow-paced will get a head start in Python with this Head First guide.

Score ★★★★★

TMUX 2 PRODUCTIVE MOUSE-FREE DEVELOPMENT



Author: Brian P Hogan
Publisher: Pragmatic Bookshelf
Price: £17.50
ISBN: 978-1680502213
magpi.cc/2nfgjQO

An 80-page book on a command-line tool might seem a negligible product, but when you see it's from Brian Hogan, author of *Exercises for Programmers* (five-star review in *The MagPi* #42), you know that its claims to make you more productive are worth investigating.

If you're used to tabs in browsers or terminals, you'll be familiar with doing more than one thing within a window. tmux is a terminal emulator which lets you do several command-line tasks within a terminal or virtual console. "Your mouse is slowing you down," says Hogan. Keyboard-controlled environments mean you don't have to break concentration on typing as you switch between tools.

tmux allows multiple 'windows' of command-line sessions, as well as dividing up the terminal window into panes like a tiled file manager. tmux sessions can exist independently of the terminal running them, reattaching them to your shell as required – great for running on headless Raspberry Pis or web servers, where you may want to leave quite complex command-line sessions running after you log out of SSH. There's plenty more to tmux, as Hogan outlines, from scripting to pair programming. Try it and boost your productivity.

Score ★★★★☆

DETECTIVE DOT

Authors: Sophie Deen & friends
Publisher: Bright Little Labs
Price: £12.99
ISBN: 978-0995636200
detectivedot.org

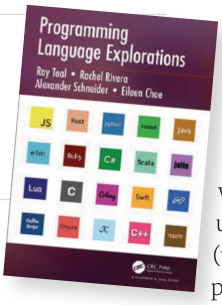
Detective Dot has high-tech gadgets, amusing sidekicks, and an arch-nemesis, but where she differs from most fictional detectives is not just that she is nine years old, and working for the Children's Intelligence Agency (CIA). She is also a coder and inventor.

Leveraging youngsters' addiction to screens, as well as being available for tablets and phones, with or without CIA membership, the book features heavy smartphone use, and in particular a modified selfie stick. Dot can talk to tech, as well as code. The code she writes during the book is all available at the end of the story, in both plain English algorithm form, and Python, along with her secret report and supplemental facts on both tech and the CIA.

Sophie Deen, former program manager at Code Club – and Code Club Pro – set up Bright Little Labs and wrote the first (of many, we hope) adventures of Detective Dot, to address a number of problems in literature, the media, and the world. Managing a fraction of this within a book that entertains pre-teens is a real achievement, and the CIA's Code of Conduct: 1. Question everything, 2. Think for yourself, feels like the summary of everything Dot does and discovers.

Score ★★★★☆

PROGRAMMING LANGUAGE EXPLORATIONS



Authors: Ray Toal, Rachel Rivera
Publisher: CRC Press
Price: £49.99
ISBN: 978-1498738460
magpi.cc/2nfnIMT

We all know that having learnt one or two programming languages well, better understanding will come through learning several more, particularly those embracing different paradigms, but it's hard to find the time. If only there were some way of concisely getting the most relevant features from other languages. Enter *Programming Language Explorations*, a book to open your eyes to theoretical concepts through practical lessons.

Twelve languages are covered, with an emphasis on ideas rather than features. A right-angled triangle generator opens each chapter, giving a useful starter in each language's style and approach. After the basics, languages are used to introduce particular concepts, or how features are expressed in that language – but every feature is not repeated unnecessarily in each language chapter.

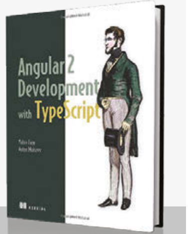
For each language, chapters end with where to go next for reading, understanding, and participating (with interesting open-source projects listed), as well as 20 or 30 exercises to embed the learning. After JavaScript, CoffeeScript, Lua, Python, Ruby, Julia, Java, Clojure, Elm, Erlang, Go, and Swift, a lightning tour of "the classics" describes what was/is important in everything from Fortran to APL. A summary of concepts covered and a choice of where to go next complete a very worthwhile investment in improving your coding abilities.

Score ★★★★☆

ESSENTIAL READING: MODERN WEB

New frameworks, better performance, and dependable standbys – web development demands a multi-skill approach

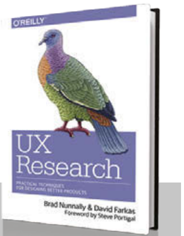
Angular 2 Development with TypeScript



Author: Yakov Fain
Publisher: Manning
Price: £36.62
ISBN: 978-1617293122
magpi.cc/2nfbSlP

Learn why Angular 2 really can live up to the hype, as you develop an example online auction app.

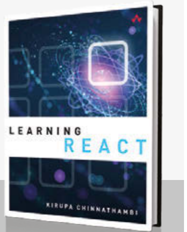
UX Research



Authors: Brad Nunnally, David Farkas
Publisher: O'Reilly
Price: £23.99
ISBN: 978-1491951293
magpi.cc/2nfbUKt

Usability research techniques weighted and compared. Thorough and practical, with great examples, and practitioner interviews. For any product you're planning.

Learning React



Author: Kirupa Chinnathambi
Publisher: Addison-Wesley
Price: £24.99
ISBN: 978-0134546315
magpi.cc/2nfmPnj

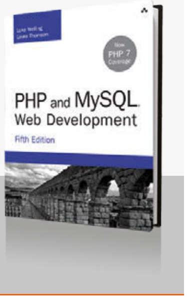
Thoroughly accessible introduction to the JavaScript library for building user interfaces (the V in MVC), and building apps with it.

High Performance Images

Authors: Colin Bendell, et al.
Publisher: O'Reilly
Price: £27.99
ISBN: 978-1491925805
magpi.cc/2nfmscm

From colour theory and image formats, to the problems of responsive images – an essential guide to a complex topic.

PHP and MySQL Web Development



Authors: Luke Welling, Laura Thomson
Publisher: Addison-Wesley
Price: £30.99
ISBN: 978-0321833891
magpi.cc/2nfptcK

PHP & MySQL are not trendy, but totally ubiquitous through WordPress, Drupal, and a dozen other essential LAMP projects. A comprehensive introduction.

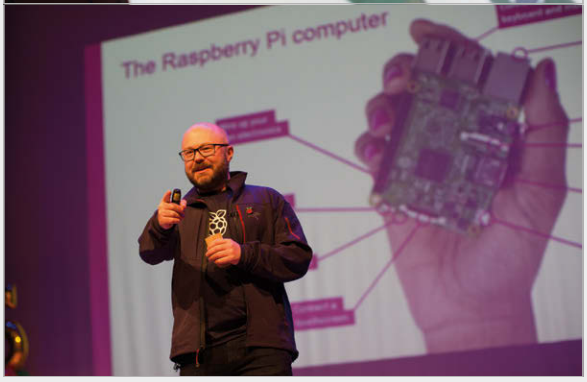
THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi

PARTY TIME!

The Raspberry Pi is five years old! Released on 29 February 2012, the Raspberry Pi has created a huge community of digital makers of all ages and genders from around the world. It has also helped reform the UK computing curriculum and sold in the millions. The Birthday Weekend at the start of March was an excuse to celebrate all of this. We were even there to show off the magazine! Here are some pictures from the various festivities.

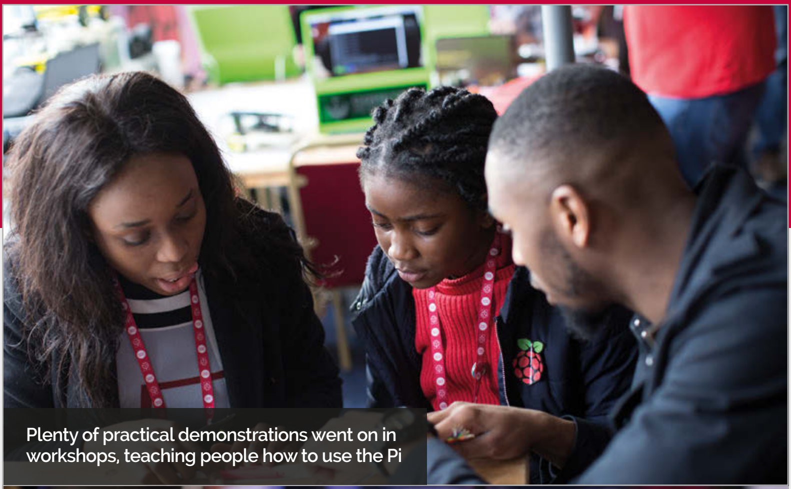
The official cake of the party changed colours as you tweeted it! There was a real cake, though. It was yummy



Top The venue for the party was the Cambridge Junction, a classy building that just about housed all the festivities
Middle Pi-powered games and gizmos were everywhere at the party, like this giant game based on Operation
Bottom Foundation CEO Philip Colligan gave a keynote speech on the Raspberry Pi

Code Club was well represented at the party – our stall was right next to theirs!

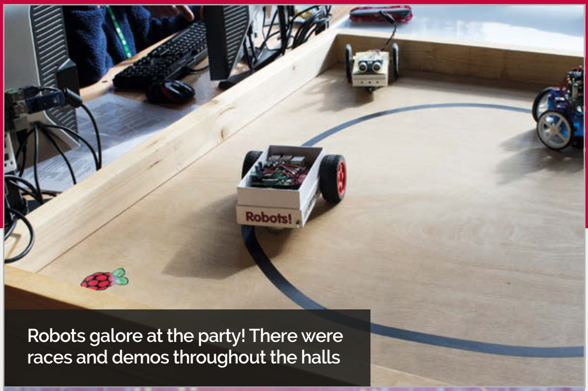
Eben enjoyed the face-painting stall



Plenty of practical demonstrations went on in workshops, teaching people how to use the Pi

This robot face tracked your movements. It was a little spooky

People of all ages were able to learn something at the workshops



Robots galore at the party! There were races and demos throughout the halls

Fran Scott came by to perform some explosions – real explosions – with code

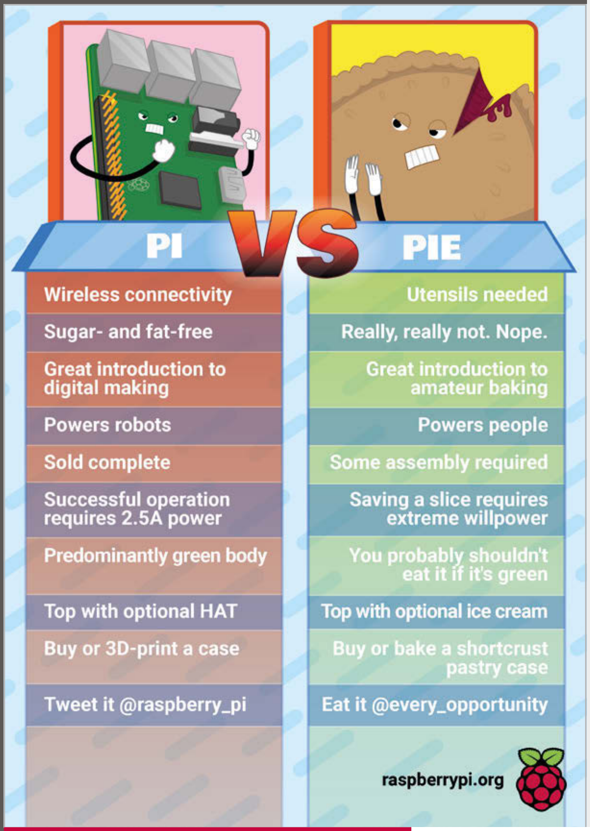
AUDIO VISUAL

Want to see more from the party? You can catch a glimpse of the talks and activities in the Big Birthday Weekend video, on the Raspberry Pi YouTube account here: **magpi.cc/2mAard6**. You might even spot The MagPi team...

PI VS PIE

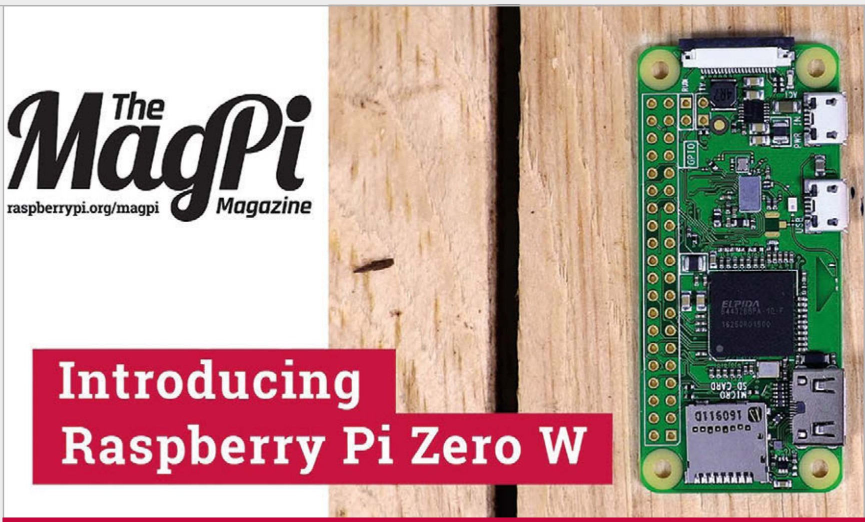
What's best on Pi Day? The Raspberry Pi or a raspberry pie?

Every year on 14 March, the internet comes together to celebrate Pi Day. In date formats where the month comes first (such as 2017-03-14), 3/14 is the first three digits of the mathematical constant pi. We always like to celebrate on Pi Day, not only because of the name but also because of the STEM connotations. This year, Raspberry Pi illustrator Sam Alder and *The MagPi* contributor Alex Bate put together a very important list, showing the differences between pies and Pis. No one should go without reviewing its contents.



PI	PIE
Wireless connectivity	Utensils needed
Sugar- and fat-free	Really, really not. Nope.
Great introduction to digital making	Great introduction to amateur baking
Powers robots	Powers people
Sold complete	Some assembly required
Successful operation requires 2.5A power	Saving a slice requires extreme willpower
Predominantly green body	You probably shouldn't eat it if it's green
Top with optional HAT	Top with optional ice cream
Buy or 3D-print a case	Buy or bake a shortcrust pastry case
Tweet it @raspberry_pi	Eat it @every_opportunity

raspberrypi.org



MAGPI VIDEOS!

We're very proud to announce that here at *The MagPi* we're now doing videos! As well as the wonderful words we write, you can now see and hear us on the Raspberry Pi YouTube channel. We launched with our very own introduction to the Raspberry Pi Zero W, hosted by our very own and very handsome features editor, Rob. Look forward to tutorials, reviews, unboxings, and even interviews in the future for *The MagPi*. Watch our first video right here: **magpi.cc/2mAbzxm**.

> "Look forward to tutorials, reviews, unboxings, and even interviews"

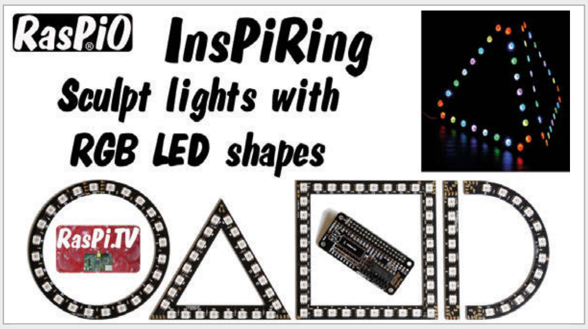
BEST PIES

We asked the Twitter followers of The MagPi the most improtant question of all: what is your favourite pie? After 24 hours and roughly 500 votes, we got these definitive results. We're very disappointed that pecan pie came last, though.

44%	24%	17%	15%
APPLE PIE	STEAK & ALE PIE	SHEPHERD'S PIE	PECAN PIE

CROWDFUND THIS!

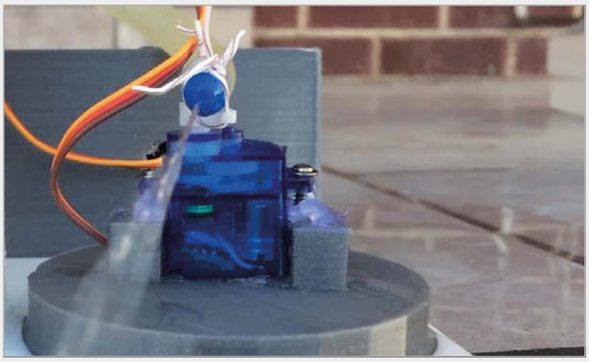
The best crowdfunding hits this month for you to check out…



INSPIRING

kck.st/2lZkKst

There are many ways to use LEDs with the Raspberry Pi, and Alex Eames has come up with yet another excellent idea for how to use them: by creating strips and shapes of LEDs that can be attached in different ways and used to easily build up a project in any way you desire. It's already hit its goal, so it's a good time to help fund it further and get your hands on some.



RASPBERRY PI WATER GUN

kck.st/2nleu1J

Ever wanted a fun Raspberry Pi kit that teaches you coding, making, and gives you an excellent prank at the end of it? The Raspberry Pi Water Gun is a kit you can build that lets you create a trap that squirts would-be pranksters with water. It's fun and daft and we love the idea, so head on over to the Kickstarter if you want to give it a hand.

BEST OF THE REST

Here are some other great things we saw this month

magpi.cc/2nvTEMU

SWITCH BOOTH

Did you manage to get a Nintendo Switch? Dragging yourself away from Zelda to read the mag? Well it seems that Nintendo of America's PR department likes the Raspberry Pi, as it powers their Switch kiosks that can be found in stores across the US.

magpi.cc/2nw3oXK

PIMAC

What we love about this project is that even though it's not the original iMac monitor, it fooled us at first and made us think it was. The whole thing is built inside the case, and the old power button turns on the new monitor. Clever.

magpi.cc/2nw46Es

WINDOWS 98 WATCH

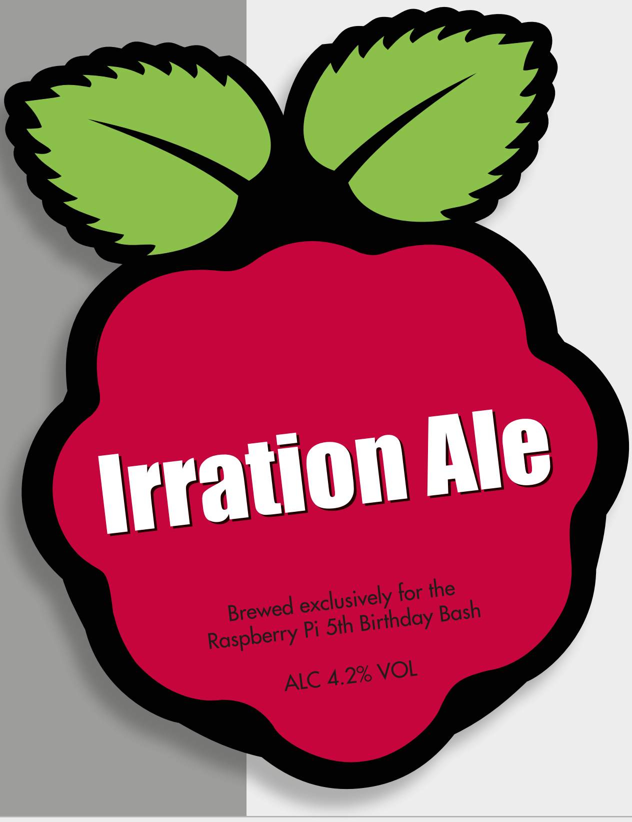
This may be pretty daft, but goodness is it also amazing? In an era of crazy Apple and Android smartwatches, let's bring it back a step back to the 1990s and power a smartwatch with the venerable Windows 98. 20 years old now, and available to run slowly on your wrist.

COMMUNITY PROFILE

BEN CROSTON

Ben Croston

Category: Maker
Day job: Brewery owner
Website: magpi.cc/2n16xey
twitter.com/CrostonBen



If you have ever used Python to control the GPIO pins then you have Ben to thank, as he created the RPi.GPIO library

Ben has always been a maker at heart. For his first project he created a home automation system while still in school, and learnt to build a robot in the late 1980s, using one of the Usborne books and a BBC Micro!

Ben Croston loves to brew beer. In fact, when he took voluntary redundancy from British Aerospace he saw it as his chance to take a passion and turn it into a business. At British Aerospace, Ben had worked on data acquisition and simulation systems using engineering principles and knowledge to trick the avionics of an aircraft to think that it was flying while recording the data for later analysis. So when Ben took voluntary redundancy, coincidentally just as the Raspberry Pi was first being announced, it was natural for him to think about the brewing process and how it could be improved using technology. As Ben frequently mentions during our chat, "Brewing is scientific and it requires a high degree of consistency in order to reproduce the same beer time and again."

So how can technology help the process? Ben has employed a Raspberry Pi to manage the heating system used to brew the beer. It provides a web interface for remote access and monitoring of the temperature in the hot liquor tank. This uses a DS18B20 temperature sensor, the same sensor that is used in the CamJam Edukit #2. It even provides a fail-safe option to ensure that the water is maintained at the correct temperature – all part of the complex brewing process.

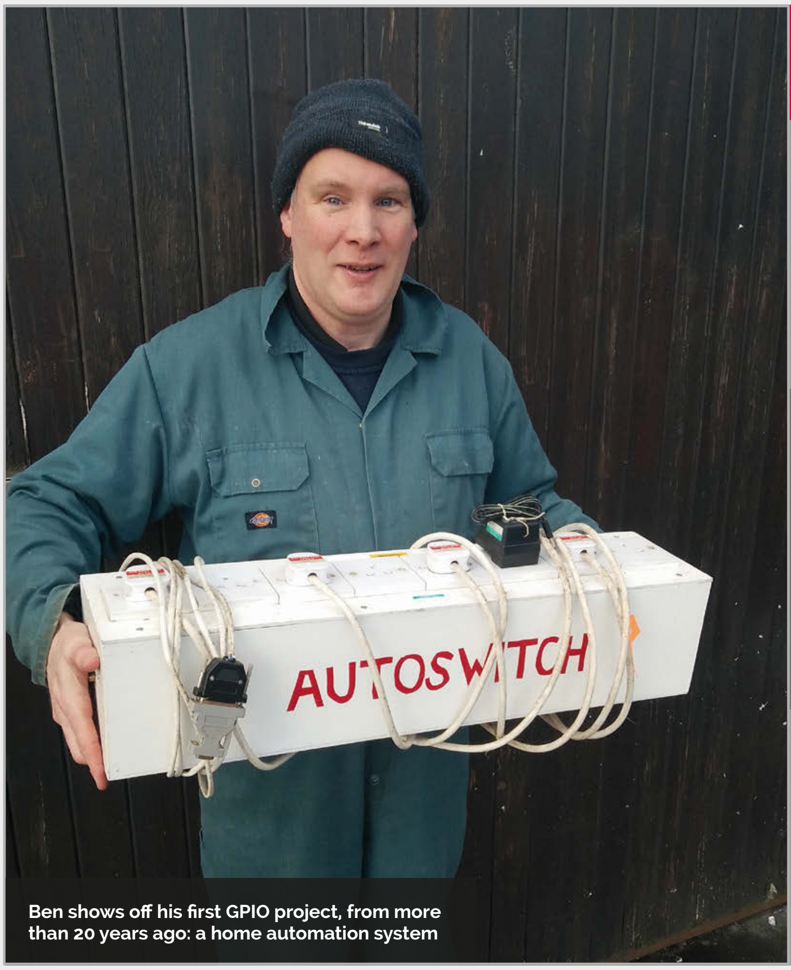
His original idea was to use RPi.GPIO to control the heaters and motors using a series of sensors. Ben has plans to add further Raspberry Pis to his brewing systems, including the fermentation system. It takes a week for the natural sugars to become alcohol. During the fermentation process, the yeast requires careful temperature control in order to produce a product of a consistent quality.

The GPIO and Ben's library have enabled anyone to build their own physical projects, but did Ben foresee this success? The simple answer is no. The popularity of the Raspberry Pi and his library could not have been predicted, and this is something which humbles Ben. He is amazed at the number of projects that have been created using his library. Every time he opens *The MagPi* and looks on a blog he sees something new, created using a library that he created to "scratch his own itch."

These constant innovations and projects show that there is a need for children and adults to have access to an easy-to-use tool for physical computing, and that by abstracting some of the complexities we can help everyone to get started with digital making. Ben is a keen advocate of children learning to code. He is a STEM Ambassador, and visits local schools to talk about the scientific processes that are required to brew beer at a commercial level.

> Ben has plans to add further Raspberry Pis to his brewing systems

Above Ben transfers the freshly brewed beer to a 'copper' tank, where it will pass through a heat exchange



Ben shows off his first GPIO project, from more than 20 years ago: a home automation system

HIGHLIGHTS

PI PARTY BEER

Every year for the Raspberry Pi Birthday Party, Ben brews a special recipe beer called Irration Ale. This beer is only for the party and is provided free of charge to the adults at the evening event. It's nice to know that we are celebrating with a product that a Raspberry Pi has helped to create.

magpi.cc/2n14qaw

RPI.GPIO

The original library that has powered hundreds of thousands of Python GPIO projects around the world. Created as a means for Ben to interface his brewing equipment with the Raspberry Pi to ensure consistent brewery practices, this library has helped many to realise their own physical computing projects with Raspberry Pi and Python.

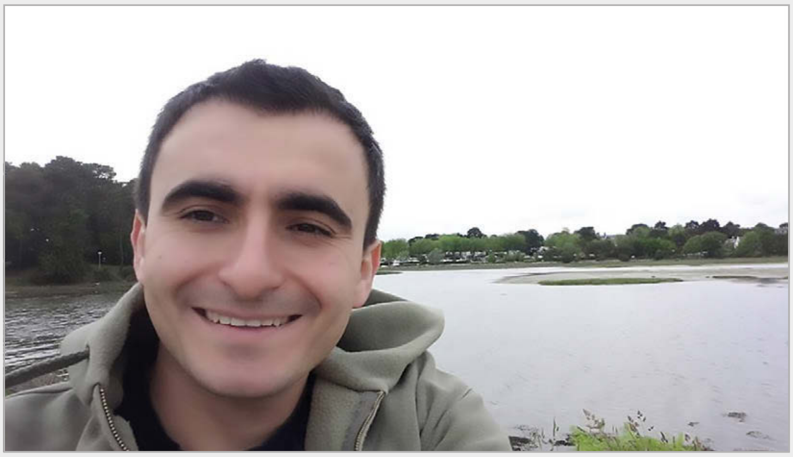
EARLY ADOPTER

Ben has always been tinkering with hardware, even when he was at school. Here (left) we see Ben with his own version of an early home automation system, which used 1990s PC technology and a series of transistors to control the power to a number of mains outlets.

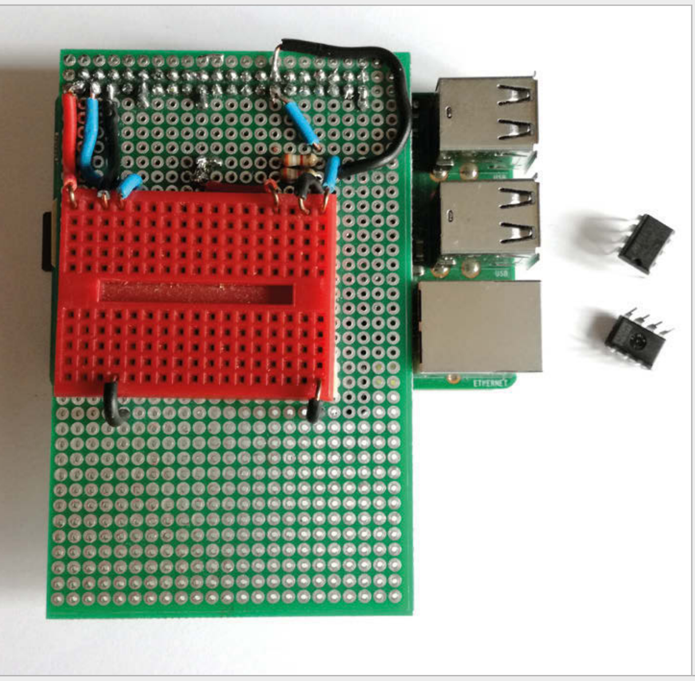
KEEPING YOUR HAT OPEN

Is there a way to use only open-source tools to design and manufacture Raspberry Pi accessories? **Leon Anavi** was determined that there should be for his Pi HAT

It's relatively easy to put together a few discrete components, and perhaps a chip or two, on a breadboard, to prototype and build your own accessory to connect up to the Pi's GPIO pins. But how about something neater and more professionally packaged? The Pi HAT format (see 'Pi HAT?' box) is a piece of standardisation that has resulted in several dozen popular add-ons, but it's also a clear specification that can be used to make your own HAT. We even published an introductory tutorial on this in *The MagPi* #42.



Leon Anavi, a programmer from Plovdiv in Bulgaria, was inspired to try this for himself, as his first hardware project. "I'm a software guy," he admits. "Take anything that I say about hardware with a pinch of salt." Later, he self-deprecatingly tells us, "The combination of a software engineer and a soldering iron could be dangerous."



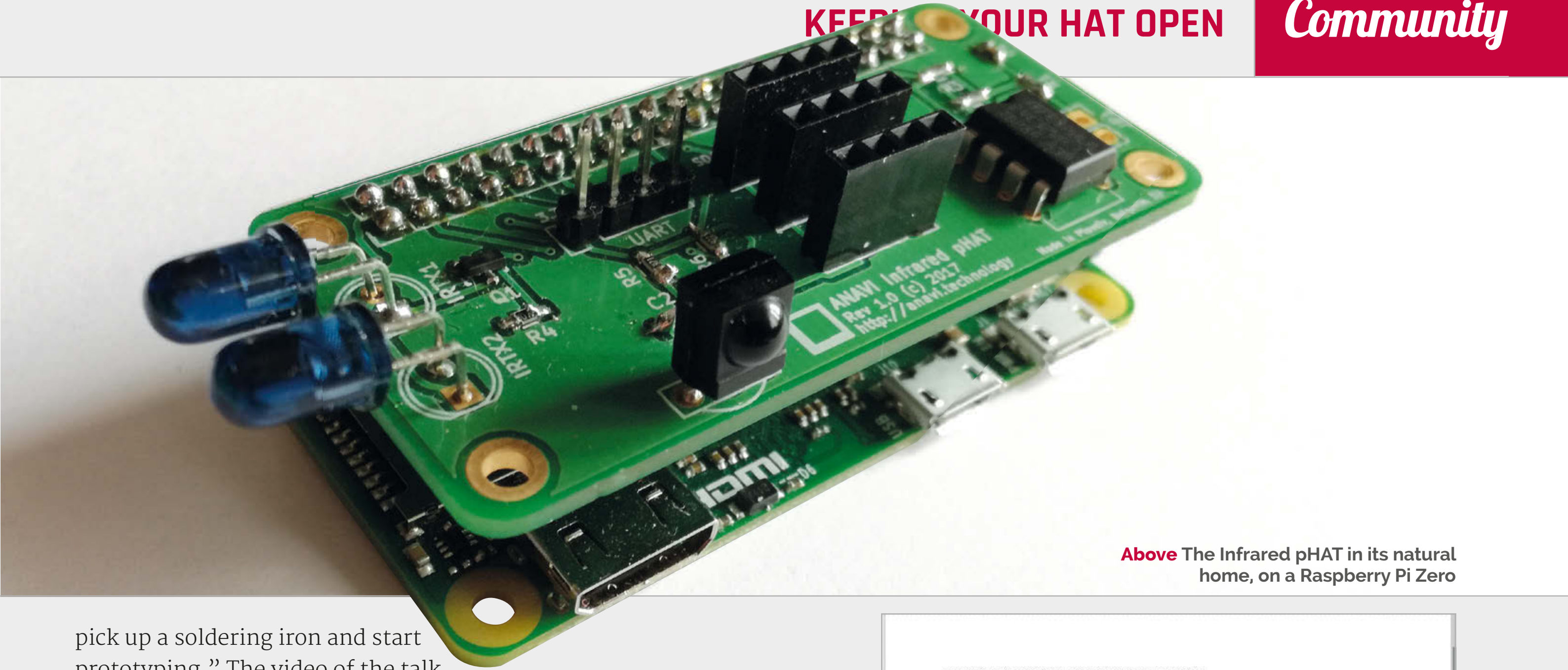
Below A simple home-made board for flashing the HAT's EEPROM saves having to buy a professional tool

Having bought the first ever HAT – the Sense HAT – for some IoT experiments, he discovered some limitations. Given sufficient processor load – something not difficult to achieve, particularly on single-core Pis – the heat generated by the CPU actually affects the Sense HAT's temperature sensor, which proved a particular problem when Leon built a weather station. Another problem for prototyping is the Sense HAT's 8×8 LED array – very pretty, but not all that suitable for a text display. Leon decided to try to make his own board, with a better-placed temperature sensor.

Leon is a senior software engineer with Konsulko Group, which provides open-source consultancy in embedded software, particularly for the automotive industry – an industry where you'll often find the Pi in a prototyping role. Long a fan of free and open-source software – and inspired by presentations about open-source hardware at FOSDEM and Hackaday Belgrade, given by Tsvetan Uzunov – Leon brought his passion for all things open source to his new interest in hardware, challenging himself to use open-source tools throughout the project.

Prototyping

Leon presented his experiences at FOSDEM [see *The MagPi* #55, pp82-83], where we caught up with him. He spoke of the purpose of his talk: for people to "get inspired – as soon as you get back from FOSDEM, [I hope] you'll



Above The Infrared pHAT in its natural home, on a Raspberry Pi Zero

pick up a soldering iron and start prototyping." The video of the talk is up on **fosdem.org**, but to save your bandwidth we'll summarise here, and add in what Leon told *The MagPi* afterwards.

For prototyping your own HAT, you don't need a great deal of fancy equipment, but you do need an idea. This could be as simple as blinking an LED, the 'Hello, World!' program of the hardware world, used in our issue 42 tutorial. While you can breadboard the circuit for your idea, to test it out in HAT format, the best way to try out your idea is by using an Adafruit Perma-Proto HAT.

You'll need an EEPROM if you want to conform to HAT standard. Leon outlined the device tree, originating in open firmware, which is used to describe the hardware using key-value pairs in a tree of nodes. The device tree is found in all modern ARM devices. The CAT24C32 is the recommended EEPROM: an 8-pin DIP, talking to the I²C interface. Flashing the chip takes a few simple steps, from the software point of view, but you do need a suitable device. As befits a hobby project, Leon made his own EEPROM flasher, using a breadboard to attach pins 3 and 5 of the EEPROM to GPIO 2 and 3 for flashing, and to GPIO 27 and 28 to read it back [see picture] – accomplished here by moving it from the left side to the right side of the breadboard, powering off the Pi in between.

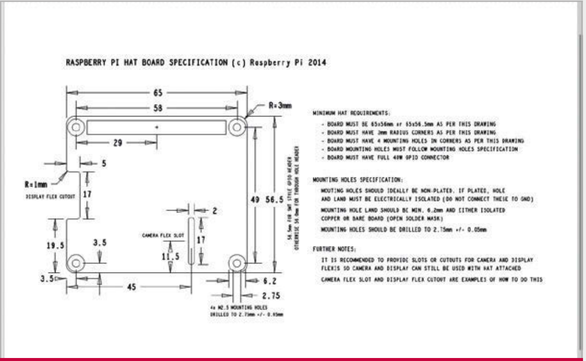
Mentoring

Having made a working prototype, Leon's next step was to design a PCB – first with a cardboard mock-up, to think through the placement of the components, then with CAD (computer-aided design) software. Many hobbyists use cut-down, free-of-charge versions of professional proprietary software, but as Leon wanted to work entirely with free and open-source software, his first choice was KiCad.

This choice was influenced not just by the advantages outlined in the 'KiCad' box (overleaf), but by local hardware company Olimex – developer of the TERES-I DIY laptop [see our FOSDEM coverage in *The MagPi* #55] – being heavy users and supporters of KiCad, and running some workshops in its use. A particular influence was local KiCad expert, Rangel Ivanov, who also helped with Leon's next project (described later).

When trying something out for the first time, particularly in a new field, never underestimate the utility of having someone local you can turn to for help – whether their mentoring is just a reassuring presence, or detailed debugging assistance. The range of help given within the Raspberry Pi and wider maker community is something we find quite inspiring.

Leon acknowledges plenty of help in creating his first DIY HAT.


RASPBERRY PI HAT BOARD SPECIFICATION (c) Raspberry Pi 2014

MINIMUM HAT REQUIREMENTS:
- BOARD MUST BE 65x56mm or 65x56.5mm AS PER THIS DRAWING
- BOARD MUST HAVE 3mm RADIUS CORNERS AS PER THIS DRAWING
- BOARD MUST HAVE 4 MOUNTING HOLES IN CORNERS AS PER THIS DRAWING
- BOARD MOUNTING HOLES MUST FOLLOW MOUNTING HOLES SPECIFICATION
- BOARD MUST HAVE FULL 40W GPIO CONNECTOR

MOUNTING HOLES SPECIFICATION:
MOUTING HOLES SHOULD IDEALLY BE NON-PLATED. IF PLATED, HOLE AND LAND MUST BE ELECTRICALLY ISOLATED (DO NOT CONNECT THESE TO GND)
MOUNTING HOLE LAND SHOULD BE MIN. 6.2mm AND EITHER ISOLATED COPPER OR BARE BOARD (OPEN SOLDER MASK)
MOUNTING HOLES SHOULD BE DRILLED TO 2.75mm +/- 0.05mm

FURTHER NOTES:
IT IS RECOMMENDED TO PROVIDE SLOTS OR CUTOUTS FOR CAMERA AND DISPLAY FLEXIS SO CAMERA AND DISPLAY CAN STILL BE USED WITH HAT ATTACHED
CAMERA FLEX SLOT AND DISPLAY FLEX CUTOUT ARE EXAMPLES OF HOW TO DO THIS

PI HAT?

A Pi HAT is a board that sits on top of the Pi – plugging into all of those GPIO pins – to add extra functionality. HAT stands for Hardware Attachment on Top, as well as referring to its position atop the Pi board. The more you spend time with computer scientists, the more you notice their love of (usually very bad) puns.

The HAT spec was introduced along with the expansion to a 40-pin GPIO with the Pi Model B+ in 2014, followed by frenzied discussion on the correct pronunciation of EEPROM. Third-party, commercial HATs were swiftly introduced. The best travelled is the Sense HAT, which made its way to the International Space Station, carried by Tim Peake, and runs code developed by UK schoolchildren.

Pi HATs must follow official specs to use the name HAT: a form factor to fit on top of the 40-pin GPIO header, on a 65 by 56mm board, and with an EEPROM containing a device tree fragment, identifying the board to the Pi. You can see the full details at: **magpi.cc/1OAHaGk**.

HAT templates are freely available for KiCad and Eagle, saving plenty of time over transferring the outline dimensions and connector holes from the Pi Foundation specification document.

Another useful site is **pinout.xyz/boards** which lists HATs and pHATs, with details of the pins that they use.

Nevertheless, he claims to have repeated a lot of common mistakes along the development path, from not complying with recommended trace space sizes, to placing a component – a potentiometer – where it could be shorted out by the Pi's HDMI connector when the HAT was moved slightly by users pressing the button mounted on it.

It's always quite comforting to hear about smart people making some of the same mistakes as the rest of us (if you are in the Tizen or Yocto communities, you'll have encountered Leon, and know just how smart he is), and it's something to be remembered when you encounter problems along the way in developing your own board.

To get the prototype board made, Leon chose the US company OSH Park, widely used in the maker community because of its combination of excellent quality and relatively low prices, although cheaper Chinese alternatives are available. Leon later looked for a local manufacturer, giving the added advantage of quick shipping.

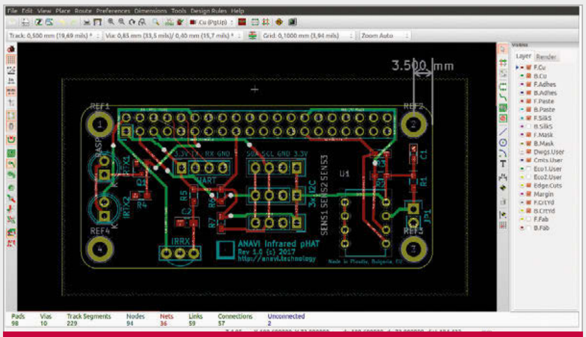
Complete package

Originally called the RabbitMax Flex HAT, legal complications led to it being renamed as the ANAVI Flex. This Raspberry Pi HAT includes an IR photo sensor, RGB and IR LEDs, a slot for an LCD character display, and a usefully specced relay (rated at 10A, 250V), as well as slots for up to five plug-and-play I^{2}C sensors – anything that i2cdetect works with shouldn't give you any problems.

During September last year, as prototype boards were distributed to bloggers and developers, software examples were written using the wiringPi library and LIRC. As a C coder, Leon was slightly taken aback by the demand for example code in Python, but provided it – using RPi.GPIO – along with C examples, and even a user manual, with all of it released under open-source and Creative Commons licences.

It's a neat little board, very useful not just for that weather station application without the temperature measurement being skewed by processor load, but a good board for general experimentation across electronics, embedded hardware, and the ubiquitous Internet of Things. It is useful if you have an idea you'd like to try out and you want to get started without having to build the hardware for it.

Following feedback on the prototypes, small revisions were made, and a low-volume manufacturing run was released and quickly sold out. The next step was a crowdfunding campaign, which led to a successfully funded larger manufacturing run – and to Leon trying his hand at designing a pHAT.



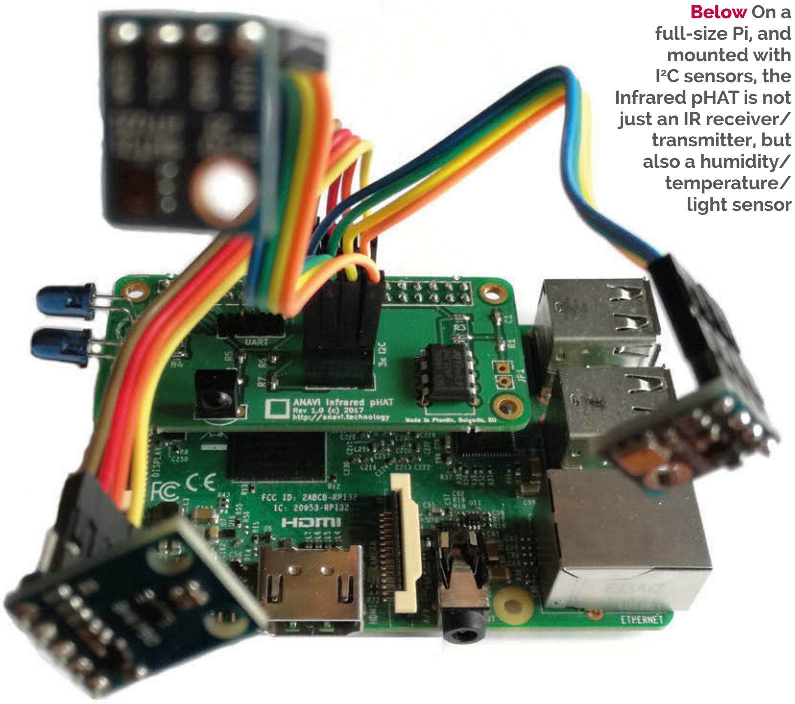
KICAD

One of the tools essential to creating a new prototype board is a CAD – computer-aided design – program. Traditionally, these have been niche, expensive proprietary products, but one free and open-source alternative, KiCad, has long been highly rated.

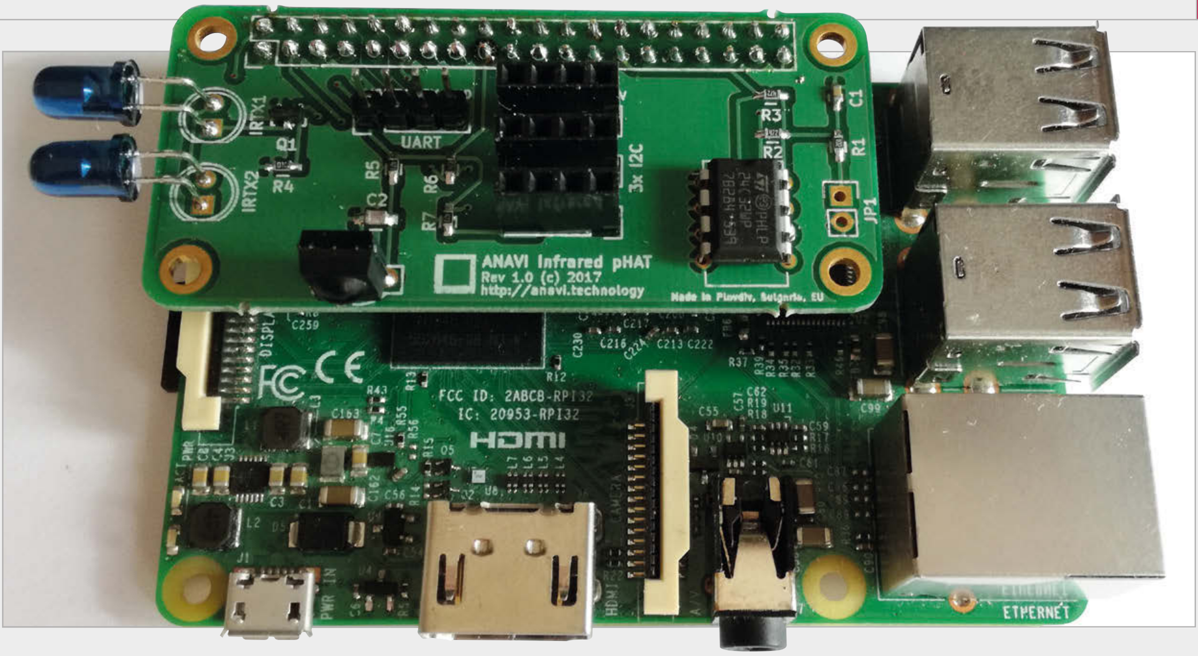
Handling up to 32 layers, KiCad also features a 'push and shove' router to clear traces out of the way as you draw a new track, or reroute the track when that is not possible. A rotatable 3D viewer enables a clearer picture of the results of your design decisions. KiCad is also a complete suite for electronic design automation (EDA), covering everything from schematic capture to a bill of materials.

The project has received generous development sponsorship, including from CERN, which introduced the push-and-shove router capabilities. The Raspberry Pi Foundation has itself donated to the CERN and Society Foundation, for the development of KiCad's differential pair routing and track-length control features.

The KiCad forum on the website features several HAT discussions and tips. Also of interest is the showcase of projects created by KiCad users: **magpi.cc/2mjclJt**



Below On a full-size Pi, and mounted with I^{2}C sensors, the Infrared pHAT is not just an IR receiver/transmitter, but also a humidity/temperature/light sensor



Left The pHAT works well on all Raspberry Pi models

Phat pHAT

Most readers will be familiar with the Pi's HAT accessories – see the 'Pi HAT?' box if you are not – but those who don't use the Pi Zero may have missed the arrival of the HAT's smaller, and unofficial, cousin. A pHAT is a partial HAT, lacking the full, offical Pi HAT specification. In particular, few have the EEPROM, and the size is smaller, matched to the Pi Zero (65 by 30mm, with four mount holes). They can be used with the Zero to make all sorts of handy gadgets – but, as with a regular HAT, they'll work with other models of Pi from the B+ onwards (with 40-pin GPIO). pHATs are also, somewhat charmingly, called 'Bonnets' over at Adafruit in the US.

> “The purpose is for people to get inspired – pick up a soldering iron and start prototyping”

Straight from the success of steering through his DIY Flex HAT project, Leon has launched a crowdfunding initiative for the ANAVI Infrared pHAT. A lot more specialist than the Flex, it's designed as a programmable IR controller to use with the LIRC software. It has an IR receiver, transmitter, UART, and three I^2C slots for sensors: temperature and barometric pressure (BMP180), temperature and humidity (HTU21), and light (BH1750).

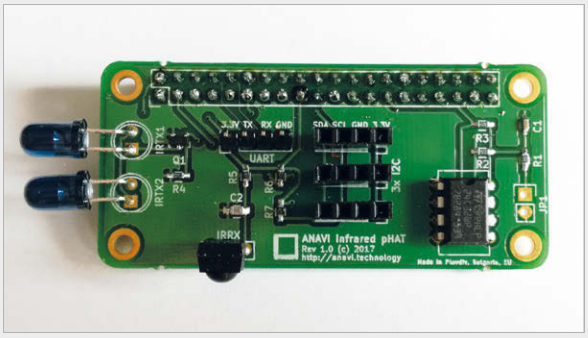
As an IR controller, it is close to plug-and-play, simply snapping into place, and coming with sample applications that make it easy to run on Raspbian. If you want to go deeper into the software, both Leon's add-ons support his IoT GNU/Linux distribution, which is based on Poky. Built with the Yocto Project and Openembedded, it features an open-source daemon for communicating with other Internet of Things (IoT) objects through the lightweight machine-to-machine connectivity protocol, MQTT.

Documentation covers the I^2C sensors, along with software installation. Possible uses range beyond a universal remote control to the receiver in all types of Pi-based devices. There's plenty of community documentation and tutorials on capturing infrared remote control signals, as LIRC is a mature and widely used library.

Leon wanted to continue with the open-source workflow in the new project, but was also keen to see manufacure take place locally. We ask him how he's got on. “I established a contact with a local manufacturing company,” he replies. “So it is going to be an open-source hardware product entirely designed, manufactured and assembled here in my hometown of Plovdiv, Bulgaria. Of course, the design is done with free and open-source tools only, like KiCad.”

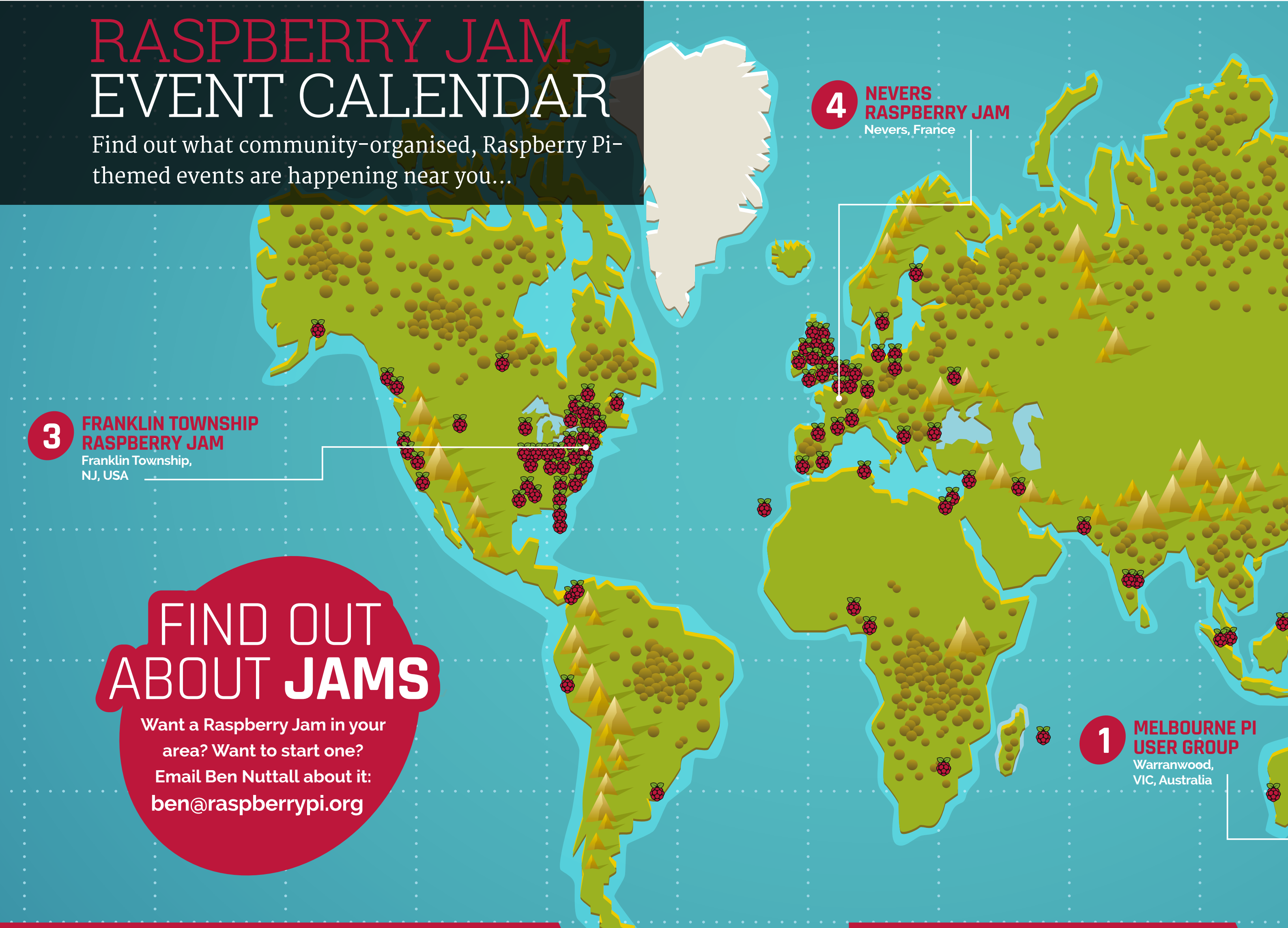
As we go to press, the Infrared pHAT crowdfunding campaign on IndieGoGo has massively surpassed its modest target. “It is amazing!” says Leon. “It is just a small hobby open-source project, so a week ago I thought that almost no one will be interested in it. The positive crowdfunding campaign has been already beyond my wildest expectations. People all over the world are supporting the project.”

Below The Infrared pHAT is perfectly designed to sit on top of a Pi Zero



RASPBERRY JAM EVENT CALENDAR

Find out what community-organised, Raspberry Pi-themed events are happening near you…


4 NEVERS RASPBERRY JAM
Nevers, France
3 FRANKLIN TOWNSHIP RASPBERRY JAM
Franklin Township, NJ, USA
1 MELBOURNE PI USER GROUP
Warranwood, VIC, Australia

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:
ben@raspberrypi.org

HIGHLIGHTED EVENTS

1 MELBOURNE PI USER GROUP

When: Tuesday 18 April
Where: Melbourne Rudolf Steiner School, Warranwood, VIC, Australia
mpug.clarkson.id.au

The Melbourne Pi User Group aims to cater to those interested in the Raspberry Pi and all it has to offer.

2 QUANTUM TECHNOLOGY CLUB

When: Thursday 20 April
Where: Cottage Lane Mission, Ormskirk, UK
quantumtech.club

Experiment, learn, and share various technologies: computing (hardware and coding), electronics, Raspberry Pis, and Arduinos.

3 FRANKLIN TOWNSHIP RASPBERRY JAM

When: Saturday 22 April
Where: Franklin Township Public Library, Franklin Township, NJ, USA
magpi.cc/2mwVIzu

Love the Raspberry Pi? Interested in learning more about it? If you'd like to come and tinker, feel free!

4 NEVERS RASPBERRY JAM

When: Friday 28 April
Where: 124 Rue de Marzy, Nevers, France
magpi.cc/2ns6zzy

A French Raspberry Jam that is being put on following the success of a previous 'National Meeting' of Pi fans.

REGULAR EVENTS

5 CORNWALL TECH JAM

When: Saturday 8 April
Where: Bodmin Library, Bodmin, UK
cornwalltechjam.uk

For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming (Scratch, Python, Minecraft, and much more).

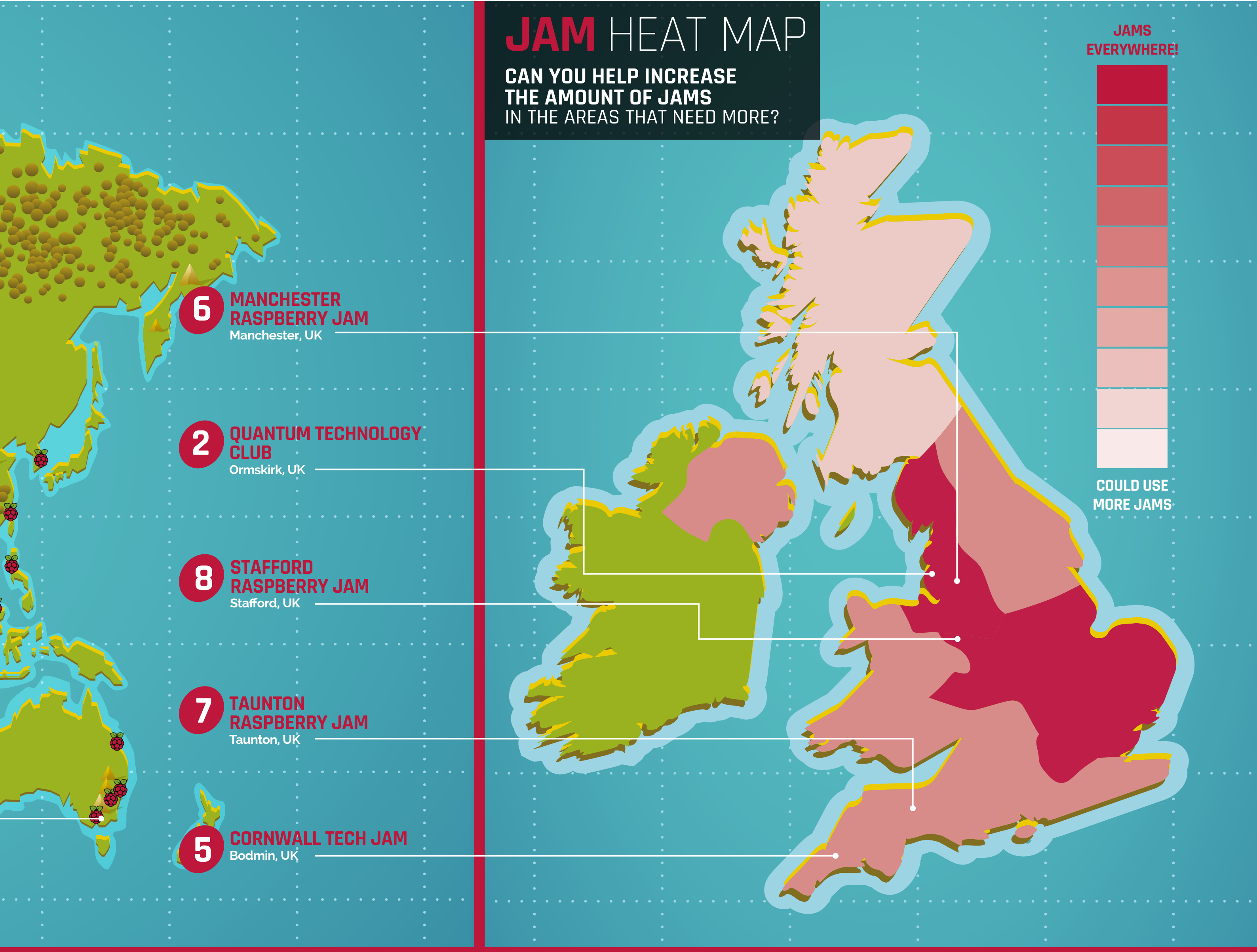
6 MANCHESTER RASPBERRY JAM

When: Saturday 8 April
Where: The Shed, Manchester, UK
magpi.cc/2k476Wn

Bring your Raspberry Pi along and take part in the workshops. There's also a getting started session for newcomers.

JAM HEAT MAP

CAN YOU HELP INCREASE THE AMOUNT OF JAMS
IN THE AREAS THAT NEED MORE?



7

TAUNTON RASPBERRY JAM

When: Sunday 9 April
Where: Taunton Library, Taunton, UK
magpi.cc/2mwTTm8

More chances to get hands-on with more digital making activities through workshops and a hackspace area to share projects.

8

STAFFORD RASPBERRY JAM

When: Tuesday 11 April
Where: Stafford Library, Stafford, UK
magpi.cc/2lzwyQu

A big meet-up of Pi enthusiasts where they all share ideas, help each other, and most of all have fun!

MORE WAYS TO LEARN

SHOULD YOU START A CODE CLUB?

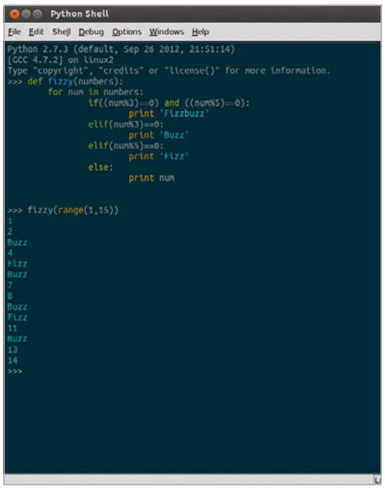
Code Clubs are amazing places for kids to learn how to code, but don't just take our word for it - listen to the kids that go to them with the latest video on the Raspberry Pi YouTube channel: **magpi.cc/2nrY9IC**. Our favourite quote is "it makes me all excited inside", which is an excellent reaction to digital making. To find out more, go to **codeclub.org.uk**

YOUR LETTERS

Highlighting code

Hi folks, love your magazine. Can you tell me what IDE you use when writing the Python code that appears in *The MagPi*? Or, how can I hack IDLE to reproduce it? I really like the way the code is highlighted in *The MagPi*. The IDLE provided with the Pi uses a different, less detailed, and (in my opinion) less helpful colour scheme. It would be great to have the code in IDLE highlighted in exactly the same colours as in *The MagPi*.

Lawrence



```
Python 2.7.3 (default, Sep 26 2012, 21:51:14)
[GCC 4.7.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> def fizzy(numbers):
        for num in numbers:
                if((num%3)==0) and ((num%5)==0):
                        print 'Fizzbuzz'
                elif(num%3)==0:
                        print 'Buzz'
                elif(num%5)==0:
                        print 'Fizz'
                else:
                        print num


>>> fizzy(range(1,15))
1
2
Buzz
4
Fizz
Buzz
7
8
Buzz
Fizz
11
Buzz
13
14
>>>
```

Above This Solarized theme is just one of many you can find for IDLE

We actually colour the code by hand in the magazine, so it's not exactly something you'd be able to use as an IDE for writing Python scripts. IDLE does have a colour scheme that you can edit to a limited extent in the settings, if you're happy just to change a few colours. However, if you do a quick Google search, you will easily be able to find a range of custom highlighting themes for Python. Follow the install instructions and you should be able to produce something closer to *The MagPi* colour scheme.

Pi Zero W price

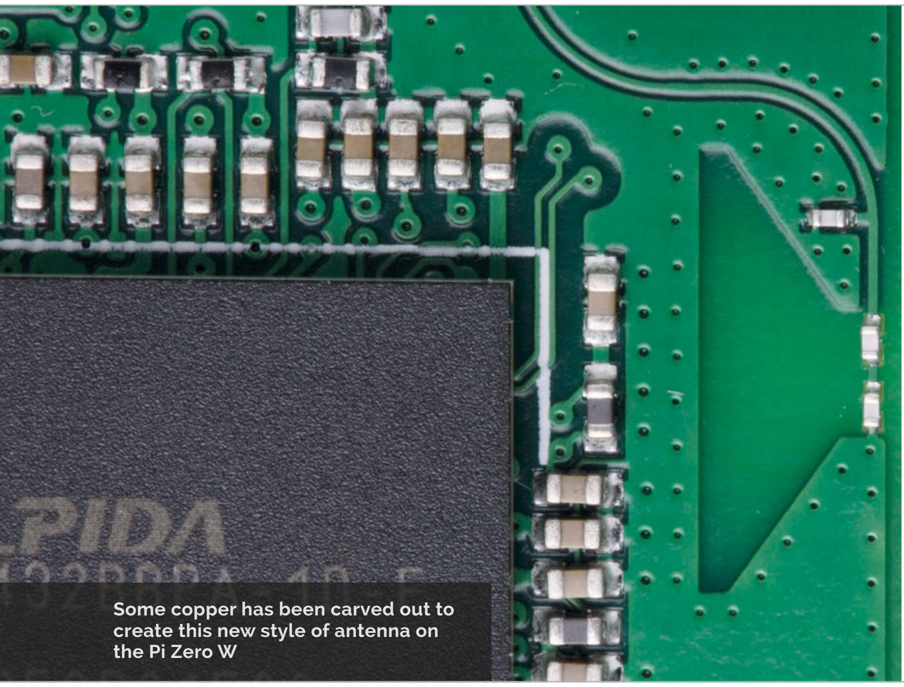
I'm a little confused. When the Raspberry Pi Zero first came out it was $5, which translated to £4. Now the Raspberry Pi Zero W is out and it's $10, the price for it is £9.60 in the UK. How come doubling the dollar price means more than double for the UK price? It's all quite confusing!

Michael B

This is all because Raspberry Pis are built at a dollar price, as all the costs of its parts are negotiated in dollars. To guarantee that the Pi is as cheap as possible, the Pis are sold at this cost converted to local currency.

For the Raspberry Pi Zero, when it came out in November 2015, $5 was about £3.40 after a straight conversion. Add VAT of 20% to that and you get to £4. If you kept an eye on Pi Zero prices throughout 2016, you'd have seen it change a bit as the pound fluctuated. This brings us to now, with the Pi Zero W costing $10. The pound is now worth a little less against the dollar, so at launch the new Zero W's pound value is higher than it would have been at the end of 2015.

Always remember that the VAT price is included on UK items, but sales tax is added afterwards in the US!



Some copper has been carved out to create this new style of antenna on the Pi Zero W

Wireless worries

I was very excited to hear about the new Raspberry Pi Zero W when it was announced. I've been wanting integrated wireless on the Pi Zero since it came out! I was surprised to see a very different type of antenna, though, compared to the Raspberry Pi 3. While it does look very intriguing, I am worried about how good it is as an antenna compared to the older one? Surely a little bit of cut-out board can't be as good? Hope you can allay my fears.

Les Bourne

We actually talked to Roger Thornton, one of the lead hardware engineers, about the antenna and he told us that in theory the antenna can be a lot more powerful than it currently is on the Raspberry Pi Zero W, but needed to be reined in a bit to make sure it passed compliance. Basically, it's as powerful as it's allowed to be, so you don't have to worry about it.

FROM THE FORUM: ARTICLE DATABASE UPDATE

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

Hello. It would be very useful to have a table of contents for all issues (regular issues and all special topic issues) of *The MagPi* which could be searched.

For example, it would be useful when I would like to find all issues of *The MagPi* which are part of a guide to learn C (which would be issues 47, 48, and 49).

Is something like this already available? If not, would it be possible to produce something like this?

Best regards,
Supermerlin

This was a letter we originally had in issue 51 and at the time we mentioned it would be great if there was an entire catalogue/database of our articles. A reader from the community called 'ozono' has since delivered, with a table of contents from all our issues! You can find the link on the forum here: **magpi.cc/2mHmzZM**.

We'll try and get our own version sorted one day as well!

WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org**
or on The MagPi section of the forum at:
raspberrypi.org/forums

In association with PIMORONI
http://pimoroni.com

WIN ONE OF FIVE SETS OF RASPBERRY PI ZERO W PROJECT KITS!

Three cute project kits for the Raspberry Pi Zero W. Solder and assemble, then connect and code them to make them unique to you.

PIMORONI
Raspberry Pi Zero W Project Kit
Scroll Bot
Solder, assemble, and code this scrolling notification kit. Program it to notify you of lots of different things like Twitter mentions or the ever entertaining news headlines.
PIMORONI
Raspberry Pi Zero W Project Kit
Mood Light
Solder, assemble, and code this colour changing light kit. Set it to change depending on your recent Tweets or the time of day.
PIMORONI
Raspberry Pi Zero W Project Kit
Pirate Radio
Solder, assemble, and code this wireless digital audio player. Stream your favourite tunes from your device or direct from the internet.

Which pHAT does the Pirate Radio project kit use?

Tell us by 26 April for your chance to win!

Simply email **competition@raspberrypi.org** with your name, address, and answer!

Terms & Conditions

Competition closes on 26 April 2017. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.

WELCOME TO THE MAGAZINE
NEW!
(Hello World)
Issue 1 | Spring Term 2017 helloworld.cc
JOIN THE CLUB
CODE CLUB CO-FOUNDER CLARE SUTCLIFFE SHARES HER THOUGHTS ON BREAKING THE 5,000 UK CLUBS MILESTONE, WORLDWIDE EXPANSION, AND WHAT 2017 HAS IN STORE
FOR COMPUTING & DIGITAL MAKING EDUCATORS
PAPERT'S LEGACY
What we can learn from the father of Logo, and constructionism's role today
LEARN NEW COMPSCI SKILLS
Start plugging knowledge gaps today
PROJECT QUANTUM
Do your pupils really understand? Quantum aims to help you ask the perfect question
LESSON PLANS AND TUTORIALS
Ideas to inspire and help you and your students
CREATE A MAKERSPACE
How to introduce a digital making space in your school
PLUS IS COMPUTING BECOMING MORE EXCLUSIVE? • ITALY PUTS MAKERSPACES IN EVERY SCHOOL • GRADUATE FROM SCRATCH TO PYTHON • CODING INTERACTIVE FICTION • PIONEERS PROGRAMME LAUNCHES • YOUR DIGITAL MAKING QUESTIONS ANSWERED • 3D GRAPHICS WITH FREE SOFTWARE • INSIDE PICADEMY USA • THE IMPORTANCE OF PLAYFUL COMPUTING

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

MAKING ART WITH CODE

Matt Richardson delves into the world of Processing, the programming language for visual artists

One way we achieve our mission at the Raspberry Pi Foundation is to find an intersection between someone's passion and computing. For example, if you're a young person interested in space, our Astro Pi programme is all about getting your code running on the International Space Station. If you like music, you can use Sonic Pi to compose songs with code. This month I'd like to introduce you to some interesting work happening at the intersection of computing and visual arts.

Processing is a programming language and development environment that sits perfectly at that intersection. It enables you to use code to generate still graphics, animations, or interactive applications such as games. It's based on the Java programming language and runs on multiple platforms and operating systems. Thanks to the work of the Processing Foundation, and in particular contributor Gottfried Haider, Processing runs like a champ on the Raspberry Pi.

When I want to communicate how cool Processing is while speaking to members of the Raspberry Pi community, I usually make this analogy: with Sonic Pi, you can use one line of code to make one note; with Processing, you can use one line of code to draw one stroke. Once you've figured that out, you can use computational tools such as loops, conditions, and variables to make some beautiful art.

And even though Processing is intended for use in the realm of visual arts, its capabilities can go beyond that. You can make applications that interact with the user through a keyboard or mouse input. Processing also has libraries for working with network connections, files, and cameras. This means that you don't have to create artwork with Processing. You can also use it for almost anything you need to code.

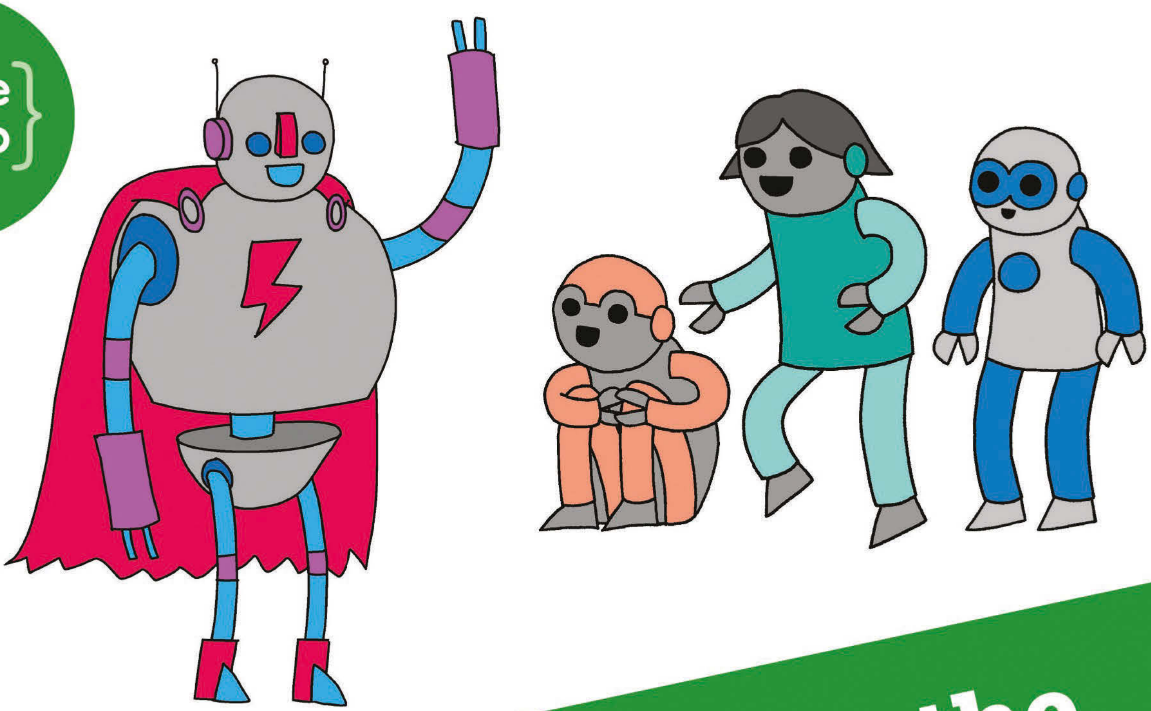
Physical process

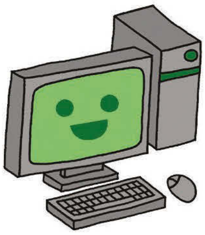
Processing is especially cool on the Raspberry Pi because there's a library for working with the Pi's GPIO pins. You can therefore have on-screen graphics interacting with buttons, switches, LEDs, relays, and sensors wired up to your Pi. With Processing, you could build a game that uses a custom controller that you've built yourself. Or you can create a piece of artwork that interacts with the user by sensing their proximity to it.

Best of all, Processing is created with learning to code in mind. There are tons of built-in examples that you can use to learn about many different programming and drawing concepts. The documentation on Processing's website is very thorough and – like Raspberry Pi – there's a very supportive community around it if you run into any trouble. Additionally, the Processing development environment is powerful but also very simplified. For these reasons, it's perfect for someone who is just getting started.

To get going with Processing on Raspberry Pi, there's a one-line install command. You can also go to **Processing.org** and download pre-built Raspbian images with Processing already installed. To help you on your journey, there's a resource on Raspberry Pi's website for getting started with Processing. It includes a walkthrough on how to access the GPIO pins to combine physical computing and visual arts.

When you launch Processing, you will be presented with a blank file where you can start keying in your code… don't let that intimidate you. All of the world's greatest pieces of art have started with a raw slab of marble, a blob of clay, or a blank canvas. It just takes one line of code at a time to generate your own masterpiece.





Pi-DAC+
Pi-DigiAMP+
DACZeroHP
Pi-DACZero
Pi-Digi+
Pi-AMP+
Pi-DACPRO

NEW RASPBERRY PI ZERO W OUT NOW!
The MagPi
RETRO GAMING BLISS
PI ZERO NOW HAS WIRELESS!
BEGINNER'S GUIDE TO CODING
MAKE A NIGHT-VISION NATURE CAMERA

The MagPi ESSENTIALS

LEARN | CODE | MAKE

OUT NOW IN PRINT

ONLY £4/$6

from

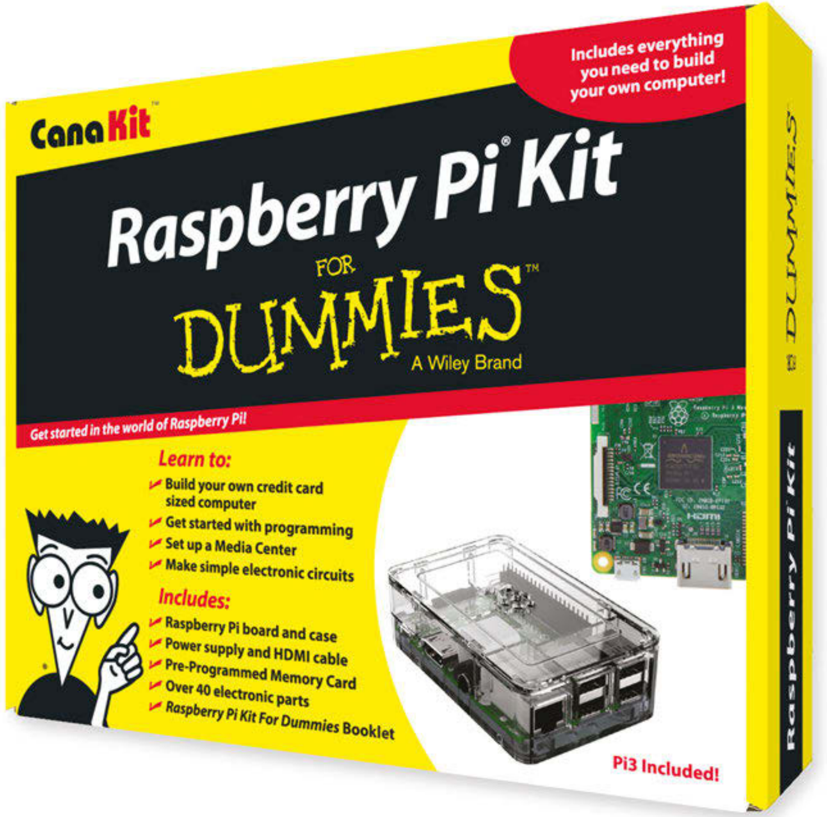
raspberrypi.org/magpi

The MagPi ESSENTIALS From the makers of the official Raspberry Pi magazine

GET THEM DIGITALLY:

Available on the App Store

GET IT ON Google Play


Includes everything you need to build your own computer!
CanaKit
Raspberry Pi® Kit
FOR DUMMIES™
A Wiley Brand
Get started in the world of Raspberry Pi!
Learn to:
Build your own credit card sized computer
Get started with programming
Set up a Media Center
Make simple electronic circuits
Includes:
Raspberry Pi board and case
Power supply and HDMI cable
Pre-Programmed Memory Card
Over 40 electronic parts
Raspberry Pi Kit For Dummies Booklet
Pi3 Included!
Raspberry Pi Kit
DUMMIES