YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

The official Raspberry Pi magazine

Issue 42 February 2016

raspberrypi.org/magpi

PI ZERO PROJECTS

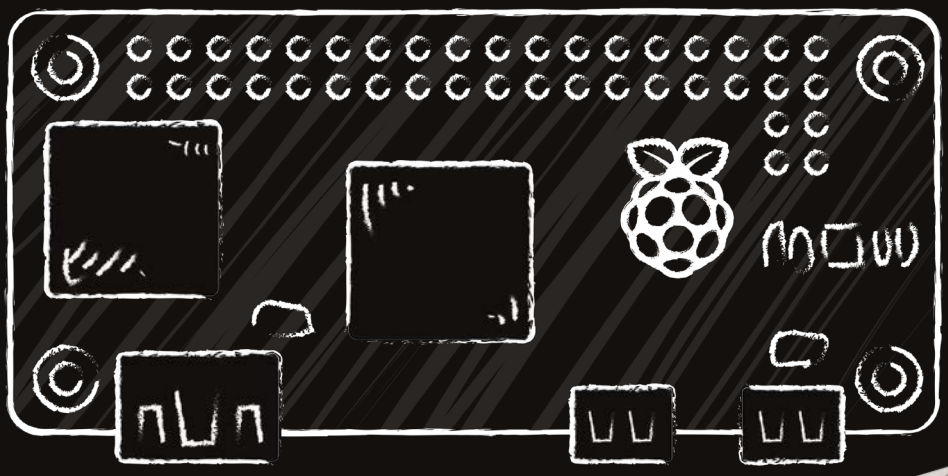
Five awesome projects you can build from start to finish

PI ZERO LIGHTSABER

Join the fight against the Dark Side with the Pi-Saber

ACTIVITY MONITOR

Keep up with your colleagues with this traffic light system



RESET SWITCH

Add a handy reset switch to your Pi Zero

QUAKE III MOUSE

Put a PC in your mouse in easy steps

Also inside:

- 10 AMAZING HAT ADD-ON BOARDS
- CONTROL MINECRAFT WITH NODE-RED
- BUILD A BINARY CLOCK WITH THE SENSE HAT
- SEND TWEETS WITH THE WOLFRAM LANGUAGE



RETRO GAMING BLISS

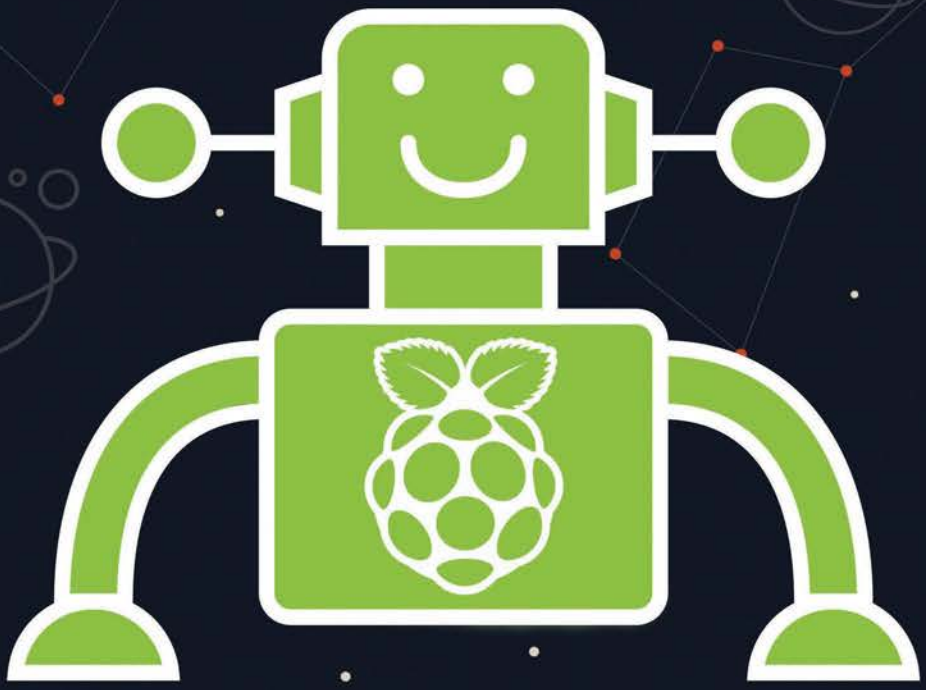
Make your own console from a controller

THE ONLY MAGAZINE WRITTEN BY THE COMMUNITY, FOR THE COMMUNITY

Issue 42 • Feb 2016 • £5.99

9 772051 998001 02

VNC SDK now available for RASPBERRY PI!



With VNC SDK you can connect VNC Viewers with VNC Servers easily over the Internet. Using VNC Cloud, no complicated network configuration at either end is required!

What will you create with VNC SDK?
Check it out here: https://developer.realvnc.com/

REAL VNC

VNC DEVELOPER

Getting connected: www.realvnc.com/products/vnc/raspberrypi/
For more information contact vncdeveloper@realvnc.com

Expand your Pi

Stackable expansion boards for the Raspberry Pi

Serial Pi Plus

RS232 serial communication board. Control your Raspberry Pi over RS232 or connect to external serial accessories.

Breakout Pi Plus

The Breakout Pi Plus is a useful and versatile prototyping expansion board for the Raspberry Pi

ADC Differential Pi

8 channel 18 bit analogue to digital converter. I²C address selection allows you to add up to 32 analogue inputs to your Raspberry Pi.

IO Pi Plus

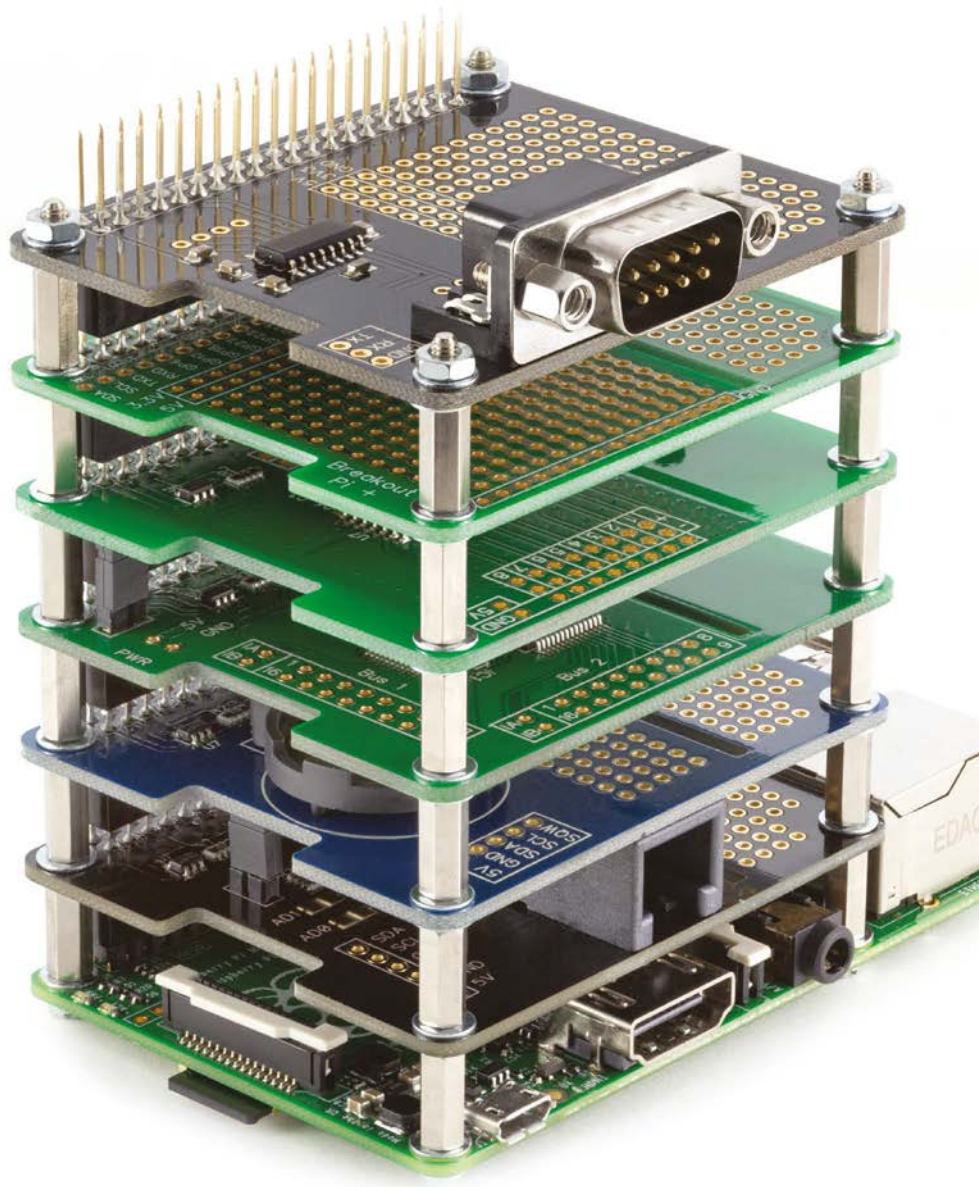
32 digital 5V inputs or outputs. I²C address selection allows you to stack up to 4 IO Pi Plus boards on your Raspberry Pi giving you 128 digital inputs or outputs.

RTC Pi Plus

Real-time clock with battery backup and 5V I²C level converter for adding external 5V I²C devices to your Raspberry Pi.

1 Wire Pi Plus

1-Wire® to I²C host interface with ESD protection diode and I²C address selection.



We also stock a wide range of expansion boards for the original Raspberry Pi models A and B

AB electronics UK

www.abelectronics.co.uk

DEXTER INDUSTRIES

SAVE 15% "MagPi15" discount code



BrickPi

Build a LEGO robot with your Raspberry Pi!

GoPiGo

Everything you need to build a Raspberry Pi robot!





GrovePi

Connect hundreds of sensors to your Raspberry Pi!

www.dexterindustries.com

WELCOME TO THE OFFICIAL PI MAGAZINE!

The Raspberry Pi Zero has really taken the world by storm. Online stores can't keep their limited stock on their virtual shelves for more than a few minutes and the production line still isn't keeping up with demand. That said, many tens of thousands of you are already proud owners of the smallest and cheapest computer in the world, and I know many of you are crying out for some ideas and project inspiration. That's why we've hand-picked five of our current favourite Pi Zero projects and turned them into guides you can build from start to finish in our cover feature, starting on page 14.

We've got everything from adding a reset switch to your Pi Zero and creating an Internet of Things gizmo, to building a fully fledged retro gaming powerhouse out of nothing more than a simple pocket-change USB controller and a few household tools – we hope you enjoy them!

We've put the Pi Zero feature near the front of the magazine so you shouldn't miss it, but before you jump too far ahead we've also got some details of the Raspberry Pi's fourth birthday party for your attention. It's happening in Cambridge (at the actual birthplace of the Pi, no less) in the first weekend in March. Tickets are already in limited supply, so get ordering ASAP and we'll see you there. See page six for details.

Enjoy the issue!

Russell Barnes



THIS MONTH:

6 COME TO THE PI BIRTHDAY BASH!
Raspberry Pi turns four, and you can help us celebrate

14 MORE #PIZERO PROJECTS
We've got more excellent tutorials for you to try with Zero

36 ZOE THE ZERO QUADCOPTER
The first Pi Zero-powered drone is here and ready to fly

70 TOP 10 RASPBERRY PI HATS
Improve your Pi with these ten fantastic add-ons

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

The MagPi

Available on the App Store

GET IT ON Google Play

CC BY NC SA

EDITORIAL
Managing Editor: **Russell Barnes**
russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
Sub Editors: **Laura Clay, Phil King, Lorna Lynch**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**
Font used & inspired by: **PurePixel**

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
Sam Aaron, Boris Adryan, Mike Cook, Dana Flinn, Brett Haines, Gareth Halfacree, Lucy Hattersley, Richard Hayler, Phil King, Simon Monk, Matt Richardson, Jon Silver & Richard Smedley

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.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.



Contents

Issue 42 February 2016 raspberrypi.org/magpi

TUTORIALS

> BUILD A BINARY CLOCK **44**
Dr Simon Monk helps you build a very digital clock

> SURFING RANDOM STREAMS **48**
Put some randomisation into your Sonic Pi creation

> NODE-RED MINECRAFT PART 2 **50**
Control Steve without using Python code

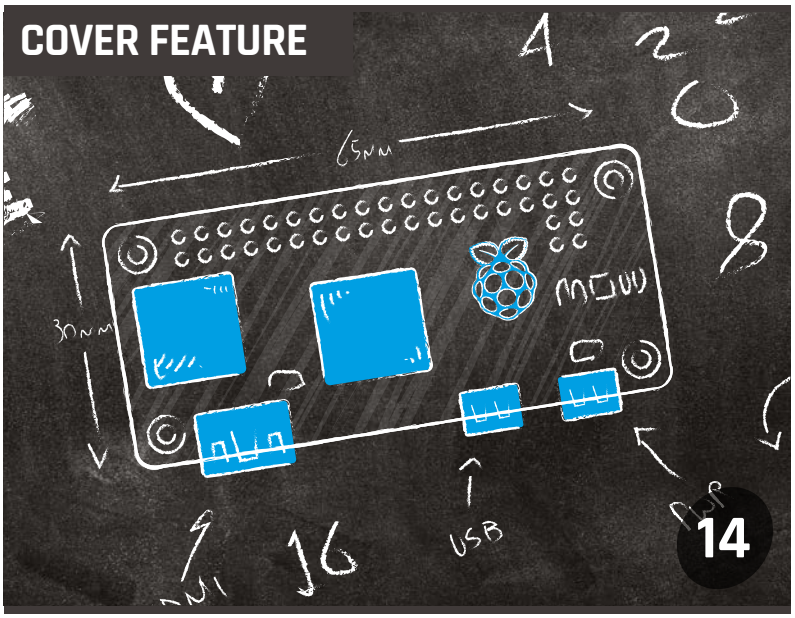
> PI BAKERY: HUNT THE SNARK **52**
Create a fun, distracting game for the kids

> USE SENSE HAT WITH FUZE **58**
Even BASIC works with the sensor board now

> TWEET WITH WOLFRAM **62**
Access Twitter very simply via Wolfram Language

> BUILD YOUR OWN HAT **64**
Your custom HAT, with the components you need

COVER FEATURE



MORE PI ZERO PROJECTS!

Five more really cool things to do with your Raspberry Pi Zero, from a simple switch to a full-on games console in a controller

IN THE NEWS



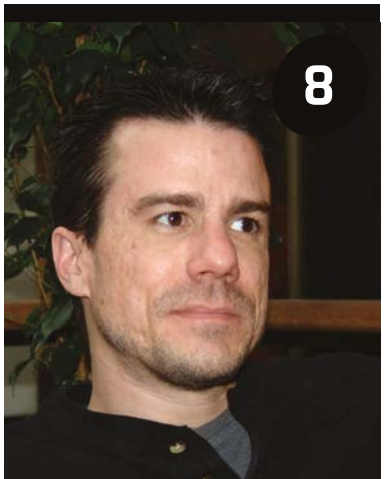
Come and celebrate Raspberry Pi's 4th birthday

The Big Birthday Weekend in March is open to everyone – find out why you should come along

ASTRO PI



Tim Peake has successfully made it to the ISS – here's how life aboard has been for him so far



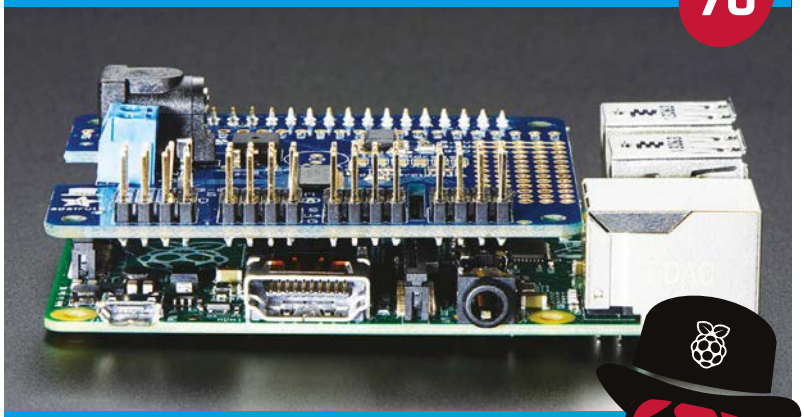
Ian Murdock

The creator of Debian, a giant in the Linux and open source community, passes on

Contents

THE BIG FEATURE

70

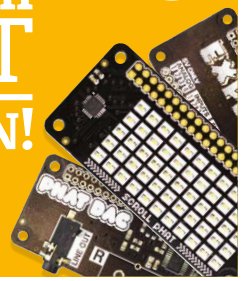


TOP 10 HATS

Not stylish headwear, but excellent hardware you can add to the Raspberry Pi. Ten of them, in fact!

5 OF EACH Pimoroni pHAT MUST BE WON!

95



Don't miss your chance to win big with The MagPi this issue!

4BORG ROBOT REVIEW

78

Check out the affordable and super-cool kit from the robot masters over at PiBorg. Maybe a late Christmas present to yourself?



YOUR PROJECTS



32

CLIMBING KILIMANJARO WITH PI

How Kurt Hunter let people follow his climb progress using a 3D printed model

Zoe the Zero 36

A custom-made quadcopter is rebooted with the help of a Raspberry Pi Zero



Sniffur 38

Dog-tracking software helps this owner know where his dog is at all times



Balance Bot 40

This sociable two-wheeled robot can recognise faces and react to them



REGULARS

> NEWS 6

The biggest stories from the wide world of Raspberry Pi

> TECHNICAL FAQS 68

Got a problem? Our experts answer all your questions

> BOOK REVIEWS 84

The latest computer books reviewed and rated

> THE FINAL WORD 96

Matt Richardson tells us how important it is to have fun

COMMUNITY

> THIS MONTH IN PI 86

The Pi community continues to make excellent things

> EVENTS 88

Find a community gathering near you in the coming weeks

> LETTERS 92

Have your say on the magazine and the community

REVIEWS

> LCD CONTROL CASE 76

> 4BORG 78

> UBUNTU PI FLAVOUR MAKER 80

> MEARM 82

CELEBRATE WITH RASPBERRY PI ON ITS BIRTHDAY!

The Raspberry Pi Big Birthday Weekend is on 5 and 6 March – come and join in the festivities as Pi turns four

YOU ARE CORDIALLY INVITED...

THE RASPBERRY PI BIG BIRTHDAY WEEKEND

Date:	Saturday 5 March & Sunday 6 March 2016
Time:	10am to 5pm
Location:	Cambridge Computer Laboratory, CB3 0FD
Party:	Saturday 5 March, 5:30pm to 7:30pm
Price:	Free for the day, £5 for the party
Tickets:	magpi.cc/1Sx4meX

On 29 February 2012, pre-orders opened for the Raspberry Pi. The original production run was for just 10,000 units. Now, nearly four years later, the Pi has become a global phenomenon and is still continuing to grow.

To mark its fourth birthday, another big celebration has been planned for the weekend of 5-6 March in Cambridge, the birthplace of the Pi, and entry is free! Like last year, the event will be held in the Cambridge Computer Lab and has been organised by the co-creators of the Cambridge Raspberry Jam (CamJam), Michael Horne and Tim Richardson.

"Tim Richardson and I are jointly responsible for organising the activities during Saturday and Sunday," Michael tells us. "This includes getting volunteers to give talks, run and assist with workshops, exhibit in the show and tell area, and take part in our marketplace."

Indeed, with those four main elements, the event will be similar in format to last year's party. There will be different topics, of course, and much of it done by the community according to Michael: "We rely very much on the community to dictate the content of much of the talks, workshops, and show and tell. We are expecting to





THE VENUE

Cambridge Computer Laboratory – William Gates Building, 15 JJ Thomson Ave, Cambridge CB3 0FD

"The William Gates Building is a modern building – very airy, lots of glass," says Michael Horne. "The Street, which is how the ground floor is known, is long and narrow, and we pack it with as much Pi-related goodness as possible. Above The Street, we have several workshop and prep rooms, which are very much as you'd expect from university seminar rooms... we're very lucky to use the venue, which is the same as last year. We also held Pi Wars there this year and found it to be an extremely welcoming, flexible place to hold an event."



Photo by Christian Richardt CC BY 2.5

be overwhelmed with input from them and hopefully we'll be able to take up most of the offers of help."

Michael says they're still working out exactly what talks will take place and which workshops will be run. "Judging from last year, you'll get a broad cross-section of speakers and tutors from all around the community and from the Foundation. There will be an Education track for teachers on the Saturday, and you can also expect plenty of content surrounding programming, hardware hacking, robotics, and home/business automation. Basically, we're hoping that the celebration weekend can give people a feel for what the community is like and what is going on 'out there'."

The event itself is open to everyone, which is why they want to give an idea of everything that is out there in Pi Land. Much as Raspberry Jams are for anyone interested in the Raspberry Pi, so too is the birthday party, albeit with a definite celebratory twist.

"We are aiming it at everybody!" explains Michael. "We hope to have activities and exhibits that will appeal to everybody who attends. There will be things that are aimed at the beginner, some that are aimed at a more intermediate level, and some for the experienced hacker. We are also hoping that many teachers from different education levels will attend, and some of the content will be aimed at them specifically... we hope to have something for everyone."

> "Judging from last year, you'll get a broad cross-section of speakers and tutors"

The event will be a fun way to meet new people, learn something about the Raspberry Pi, or just celebrate its birthday. Hopefully, there will be some special surprises there as well. After the success of the previous year's birthday bash, the team are confident they can do it again.

"It went down a storm," recalls Michael. "Almost everybody enjoyed the event and we got asked many, many times if there was going to be another one. We're hoping that was a good indication that everything went well!"

Tickets are expected to go fast, so you should check out the information for the event (**raspberrypi.org/birthday**) and decide when you want to go. The main event will take place on both Saturday and Sunday and is free, while a special party is being held on Saturday evening, with tickets priced at £5, so you can celebrate the Pi's birthday in style. We hope to see you there!




Photo by Ilya Schurov, Computerra Weekly

IAN MURDOCK, CREATOR OF DEBIAN, PASSES AWAY

A giant in the Linux and open-source community leaves behind a legacy of code

Debian is one of the most important versions (or distributions) of Linux. It powers Ubuntu and helped pave the way for a friendlier Linux that was much easier for people to get started with. It's the basis for Raspbian, the main Raspberry Pi OS, for this very reason.

With this in mind, it's very sad to hear of the passing of Ian Murdock, the creator of Debian and its ethos. His death on 28 December 2015 came as a shock to the Linux and open-source community at large, especially as he was only 42.

> "Though Ian's legacy goes beyond just Debian, it is where he first made a name for himself"

Debian Manifesto

Though Ian's legacy goes beyond just Debian, it is where he first made a name for himself. While studying at Purdue University, Indiana, Ian created the Debian Manifesto, a basic premise as to why Linux should be packaged as distributions and how the best way to go about that was. "The Debian design process is open to ensure that the system is of the highest quality and that it reflects the needs of the user community," Ian wrote in 1994. "By involving others with a wide range of abilities and backgrounds, Debian is able to be developed in a modular fashion. Its components are of high quality because those with expertise in a certain area are given the opportunity to construct or maintain the individual components of Debian involving that area. Involving others also ensures that valuable suggestions for improvement can be incorporated into the distribution during its development; thus, a distribution is created based on the needs and wants of the users rather than the needs and wants of the constructor. It is very difficult for one individual or small group to anticipate these needs and wants in advance without direct input from others."

Many of these concepts are now fairly standard in Linux distribution development, keeping them open and allowing for the best work possible to go into them. Debian was first released in September 1993 and, 22 years later, is still going very strong.

Linux Foundation

After his work on Debian, Ian was appointed CTO of the then Free Standards Group (FSG) in 2006. When the Linux Foundation – the leading Linux advocacy group – formed from FSG a year later, he continued in this role.

In the same year, Ian joined what was then Sun Microsystems and created an open-source version of Solaris, called OpenSolaris. Based on the 'lessons of Linux', it included many GNU tools, as well as a GNOME desktop and other Linux-like trappings. While there, he was vice president of emerging platforms.

When Sun Microsystems merged with Oracle, Ian left and joined Salesforce Marketing Cloud as vice president of platform and developer community. Salesforce creates analytics software for marketing companies.

Shortly before his death, Ian joined Docker. His name will be remembered in the Linux community, and not just because the '-ian' in Debian is from his own name. His philosophy lives on in his projects and in the others that he inspired, and will do for a long time to come.

FUZE

Teaching kids to code

"The FUZE is what the Raspberry Pi was designed for"

THE NATIONAL MUSEUM OF COMPUTING

The FUZE T2 Case Special Edition

- Protect your Pi from physical & static damage
- UK keyboard* & 4 Extra USB ports
- FUZE I/O Board with 40 way GPIO pass-through
- Clearly labelled input output ports
- 2 Amp power supply and on/off switch!
- Adds analogue ports, 4 in & 1 out
- 840 pin solderless breadboard (black)



Reviewed model FUZE T2-R

Computer *act!ve*

BUY IT AWARD

PC PRO Recommended

...makes the Pi more accessible than ever

micro mart

EDITOR'S CHOICE

... it's certainly the **best** we've **ever** tested

£99.99

FUZE T2-C-SE

Compatible with Raspberry Pi V1 & V2

BONUS

Includes 8GB SD pre-configured with FUZE BASIC

* USA & German keyboard layouts are also available. Prices include VAT but not shipping - see fuze.co.uk for details

©2015 FUZE & the FUZE logo are trademarks of FUZE Technologies Ltd. Raspberry Pi and the Raspberry Logo are trademarks of the Raspberry Pi Foundation and are used with permission. All rights reserved.

FUZE Technology Ltd
+44 (0) 1844 239 432 - contact@fuze.co.uk

Available from www.fuze.co.uk



WALKING IN SPACE

From launch and docking, to breathtaking spacewalk, **Tim Peakes**'s adventures have gripped the nation – but now we wait for the Astro Pi experiments to start...

THE STORY SO FAR



Looking for a way to inspire children about the possibilities of science and technology, the Raspberry Pi Foundation met with representatives of the UK space industry and the UK Space Agency early in 2014, and came up with the idea of taking a Pi – equipped with add-on sensors – up to the ISS, carrying software written by UK schoolchildren.

Launched at 2015 BETT (the education technology show), the coding competition was a huge success, with winning entries making novel and ingenious use of the Pi and the Sense HAT for everything from *Minecraft* data visualisation to radiation detection, and even some fun games to test how astronaut reaction times change during several months in space. The judges – a panel from all of the organisations mentioned – were particularly impressed by the appreciation of the scientific method shown in the primary school entries, as well as the quality of work that had gone into the older pupils' entries. You can read more about the winners in *The MagPi* #34 and #37.

Last month we brought you news that two Raspberry Pis and Sense HATs – the Astro Pis – were safely aboard the International Space Station (ISS). After bad weather had delayed the launch of the Cygnus cargo freighter, British ESA astronaut Major Tim Peake safely followed the Welsh-made computer boards aboard the station a few days later. Many people watched the launch, and the docking – including the schools which supplied the experiments to be run on the Astro Pis – but even more tuned in during January's epic spacewalk.

The UK and the world were captivated by Tim Peake's breathtaking spacewalk, carried out to fix a voltage regulator on one of the banks of solar panels that provides the ISS with its electricity. Although cut slightly short by the build-up of water in fellow astronaut Tim Kopra's helmet, the walk was a success. To effect repairs, Tim had gone right to the far end of the space station where, as he put it, "perched on the farthest handrail of the ISS, to my right was nothing but the vast blackness of space." Replacement of the regulator had to be carried out in darkness (every 90 minutes, the ISS's orbit takes it through day and night), when the solar array was producing no electricity.



Above Tim Peake arrives on the ISS. Picture taken by Space Station commander Scott Kelly



Although necessary for maintenance work on the space station, the spacewalk is also one of the most sought-after tasks for an astronaut: a chance to get as close as is humanly possible to the vast blackness of space, with the beauty of the Earth below

Science ahead

Tim has a busy schedule of science experiments ahead, from studying colloids in microgravity to the effects of low gravity on brain pressure, and how that affects astronauts' vision. Although only eight hours are officially allocated for education activities during Tim's stay aboard the ISS, and he has a packed schedule of experiments during his 40-hour working week, he plans to spend Saturday afternoons performing Astro Pi experiments. To speed things along, the Master Control Program (MCP) – written to control the multiple Astro Pi programs – will take care of the timings of experiments, and help with transferring data back to mission control, where Libby Jackson, the UK Space Agency's astronaut flight education programme manager, will share it with the rest of us.

Astronauts running code and low-cost hardware that we can all run back on Earth should make a stronger connection between Pi users and space education, but this will apply most strongly to the British schoolchildren whose experiments are carried on the Principia Mission Astro Pis (see 'The Story So Far' boxout). Those Astro Pis, in envy-inspiring aerospace cases (possibly the most requested accessory ever: we're sure they'll become available some time this year), are now ready and waiting for Tim to carry out some of the experiments. With Tim quite active on Twitter and Facebook throughout the mission, you will be able to share in the news when the experiments start.

> "As we go to press, the Astro Pis have been unwrapped"

As we go to press, the Astro Pis have been unwrapped and are ready for the experiments to begin. By the time you read this, the first Pis to go beyond the Earth's atmosphere could be running code from UK schools, inspiring a new generation of scientists who'll be taking us further into space in decades to come.

WALKING BACK AGAIN



If you missed the spacewalk, you can view the whole thing on Tim Peake's Principia Mission blog, at **magpi.cc/1JW6jii**. Just try to imagine performing the same movements in a swimming pool, with weights attached to you, to get the full feel of the endurance involved in carrying out these engineering tasks!

COLUMBUS



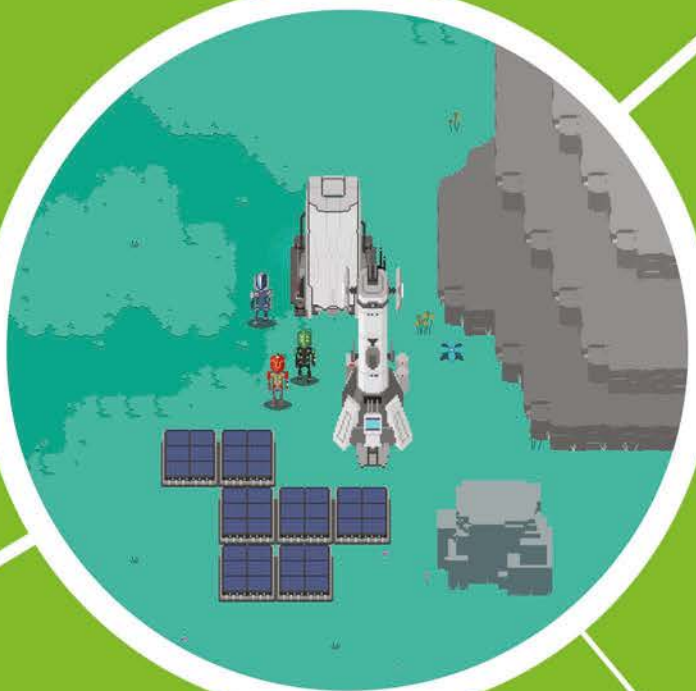
Tim Peake is working in Columbus – the first European laboratory dedicated to long-term experimentation in microgravity – which has been attached to the ISS since 2008. For more information about the Columbus research laboratory, visit **magpi.cc/1QbgqlQ**.

CEEDuniverse

LEARN
PLAY
CREATE

Discover a world of science fiction grounded in computing reality with **pi-top**'s flagship learning software!

Learn to code, make circuits and create hardware with your friends using **CEED**universe a massive multiplayer online game. Perfect for added engagement to STEAM classes and home learning! **CEED**universe comes free with all **pi-top** devices.





Build an outpost with your friends!



Learn to build circuits in game!



Uncover hidden secrets and learn to code!

What will you do in **CEED**universe?

WWW.PI-TOP.COM

pi-top

AVAILABLE AT **WWW.PI-TOP.COM**

WORLDWIDE SHIPPING
Colours in Green or Grey

World's First Raspberry Pi laptop.

12Hr Battery Life
13.3" HD Screen

Accessable Modular Components & RPi

Compatible with all models of Raspberry Pi

$299 excluding VAT

MagPi Review

★★★★☆



pi-topCEED

pi-topCEED is the easiest way to use your Raspberry Pi!

13.3" HD Screen

Front facing access to the Raspberry Pi and components

Compatible with all models of Raspberry Pi

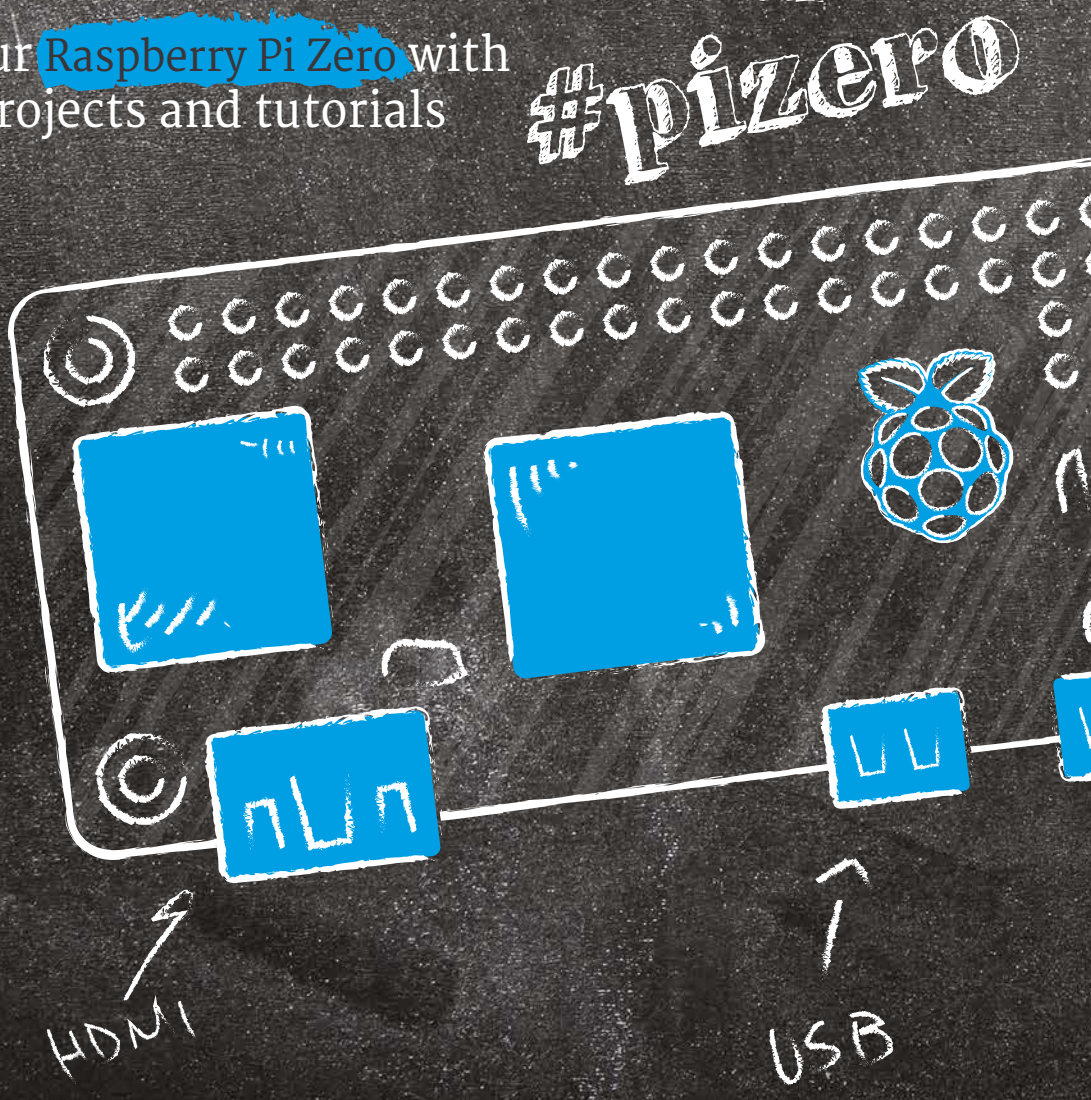
$99 excluding VAT



"I'd love to see a **pi-top**CEED in every classroom."
— Eben Upton, Founder & CEO of RPi

More #PIZERO PROJECTS!

Do more with your Raspberry Pi Zero with these excellent projects and tutorials



If you've read issue 41 and have been following us online, you'll know that ever since the Raspberry Pi Zero was released, there have been lots of people using it for many different projects. From simple little utilities to full-blown drones, the uses of the Pi Zero seem endless thanks to its ridiculously tiny size and power.

In tribute, we thought we'd come up with more of our own ideas for readers to try out. Hopefully, this will help get your creative project gears whirring, so you can have a stab at that dream project you've always wanted to do.

These projects will also technically work with any other Raspberry Pi, although you might run into space issues with some of them. So, grab the Pi Zero you got free with issue 40 and let's begin.

CONTENTS

Where do you want to start?

ESSENTIAL KIT

As well as listing some of the essential materials needed to make some of our projects, we've also included a whole host of accessories for your Raspberry Pi Zero. Whether you just want a nice neat case or a different way to connect USB devices, we should have something for you.

PG 16



PI CONSOLLER

An upgrade to the NES Zero controller we made last issue, the SNES Zero controller also has a built-in rechargeable battery, allowing for a single cable solution. You can also play a few extra games on it thanks to more buttons, and not just those on SNES.

PG 18



RESET SWITCH

Starting off with something simple yet incredibly useful, this reset switch makes use of a couple of unpopulated pins on the board that hook directly to the Pi Zero's chip. This allows you to reset the Pi when needed, without unplugging, and it doesn't take up space anything else can use.

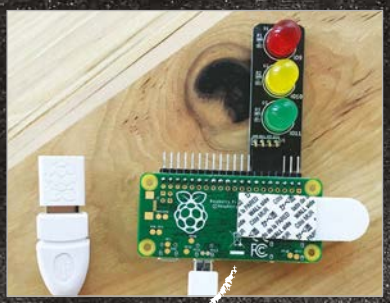
PG 22



SLACKTIVITY MONITOR

Slack is an instant messaging service that's currently very popular with businesses and organisations. Here we have a simple project that taps into Slack and tells you how busy the chat rooms are at any given time, with the use of LED traffic lights.

PG 24



PI ZERO LIGHTSABER

Bring an elegant weapon from a more civilised age into reality, and make it safe as well. The Pi-Saber uses a foam pool noodle and some LED lights to create the laser sword of space wizards from a galaxy far, far away. Fortunately, it can't chop off a limb by accident.

PG 26



QUAKE III MOUSE

Play *Quake III* entirely with a mouse, no keyboard required! This is actually something Eben Upton desperately wanted, so we decided to create one to let him know it could exist. Buttons on the mouse replace WASD keys for a one-device solution.

PG 28



#PIZERO ACCESSORIES

What you can get for your Raspberry Pi Zero to make using it and creating projects that much easier?

As there's a bit of a Raspberry Pi economy now, when new gear comes out there's a much faster reaction when making new accessories for it. Even though the Raspberry Pi Zero is only a few months old, this means that there are already a few accessories you can get your hands on to improve your Pi Zero experience.

There's also a bit of kit you'll need to make some of the projects in this issue, and even start making your own incredible builds.

ZERO CASES

Keep your Pi Zero in style with these trendy cases

ZEBRA ZERO

magpi.cc/1Klkt7o

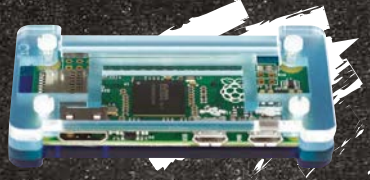
A sturdy case made up of two layers of wood along with a thick acrylic centre, the Zebra Zero adds a touch of class to your Raspberry Pi Zero. It also has a massive heatsink, which may be a little superfluous. You can also get a Zebra Zero Plus with a full breadboard attached for easy testing.



PIBOW ZERO CASE

magpi.cc/1KljtQz

A special Pi Zero version of the classic laser-cut Pibow is already available. With its unique, chip-hugging design, it leaves the ports very open for use with any custom project and even some of the Pimoroni HATs – although you'll have to insert the SD card before construction.



MODMYPI ZERO CASE

magpi.cc/1KlgEiu

A matte black, laser-cut, layered case for the Raspberry Pi Zero. This keeps access open to the four main ports to get the Zero working (HDMI, SD, USB, and power), as well as keeping the GPIO pins free. Even the reset and video out pins are accessible through it.

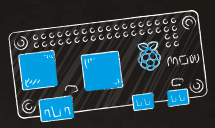


ADAFRUIT PI PROTECTOR

magpi.cc/1ZEZbhn

A much simpler case than the others featured here, the Protector covers the bottom and part of the top of the Pi Zero, not quite offering the full protection of its rivals. It does, however, have more space open around the GPIO than others, allowing for greater access to it in case you have tight space in your project.





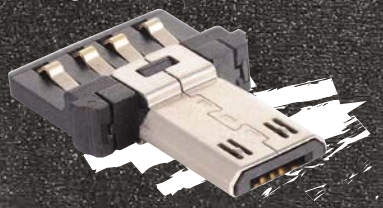
MISCELLANEOUS ADD-ONS

Other bits and pieces to improve your Raspberry Pi Zero

MINI USB OTG ADAPTOR

magpi.cc/1JT9aZc

There are a few of these available – we've had some success with the meZmory brand one, but there are several others. It makes plugging in single USB devices to the Pi Zero much easier – great for projects that just require wireless and nothing else.



RIGHT-ANGLE GPIO HEADER

magpi.cc/1ZFoH31

We've shown you how to solder a standard GPIO header to the Pi Zero, but this cool alternative allows for the GPIO pins to be parallel with the board. This can be ideal for awkward projects or installations where space restrictions are fixed, and it solders in like any other GPIO header.



ESSENTIAL RASPBERRY PI ZERO KIT

magpi.cc/1RSULPN

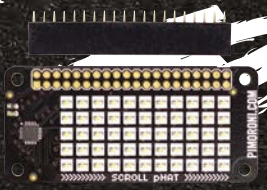
A nice selection of components that can be used with the Raspberry Pi Zero , including extra adaptors, several GPIO header options, and even little stick-on feet to raise the Pi Zero off the ground. It all comes in a little mint-like tin which can also fit the whole Pi Zero for a makeshift carry case.



SCROLL PHAT

magpi.cc/1JT9cQT

This actually works on other Raspberry Pis as well, but it's designed in the same form factor as the Pi Zero. This board gives you 55 LEDs in a rectangular formation and works similarly to the LED matrix on the Sense HAT. Using a Python script, you can control the LEDs or have it scroll text across the length of the board.



EXPLORER PHAT

magpi.cc/1Pk5SdN

This add-on board for the Zero (and other Pis) allows it to communicate with different objects more easily than wiring them directly to the Pi. This includes motors and sensors – we've seen Explorers used in robots due to this. It allows for analogue inputs as well as other inputs and outputs, and has its own Python library for quick development. Great for tiny projects.



ESSENTIAL KIT

You'll need these basics to create amazing Pi Zero projects

Components

LEDs, wires, resistors, and maybe even some light, heat or motion sensors. All of these are available for very cheap in bundles from a local electronics store, or online from a distributor such as RS Components. You can also get a lot of them in an electronics starter kit, like the one pictured from ModMyPi (**magpi.cc/1RvzWta**). For the tutorials in this issue, we'll make sure you know exactly what sort of components you need before you go out and buy them.



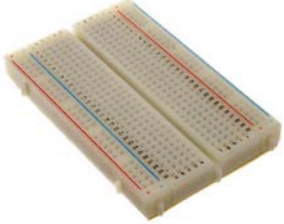
Soldering iron

An important piece of equipment for making more advanced projects, or building some kits available for the Raspberry Pi. As well as the actual soldering iron, you'll need good quality solder to work with. A decent stand for the soldering iron is also ideal – one that you're not likely to burn yourself with, containing a good sponge. Soldering can be dangerous, so make sure to supervise younger makers when they're building.



Breadboard

A perfect tool for prototyping and testing your circuits. If you've never used a breadboard, what it does is give you a load of pins that are aligned in columns, acting as connectors so you don't have to solder components together. This means you can quickly change a circuit to get it right, or reuse the breadboard over and over again for new and different circuits. You probably don't need to start with a massive one, but get one with positive and negative rails on each side.



Wire stripper

If you decide to get a spool of wire rather than a load of pre-cut pieces, you'll need a good way of getting wire off the spool. Attacking slowly and carefully with a pair of scissors or a knife gets extremely boring fast, and investing in a decent pair of wire cutters that will also strip the plastic will improve your life by at least 10% (don't quote us on that).

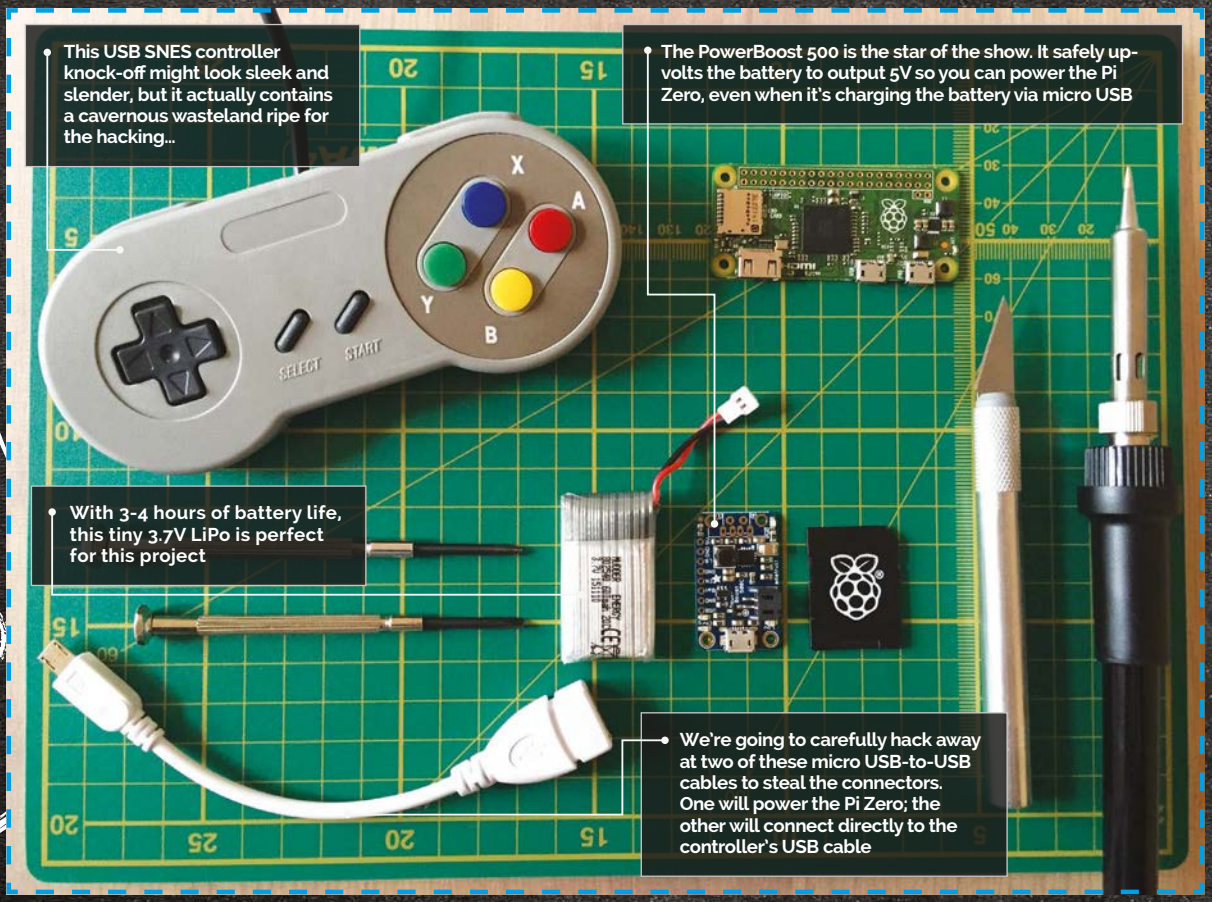


PI ZERO CONSOLLER

The Pi Zero's diminutive stature allows us to make an entire retro gaming console out of a simple USB controller for the first time. Here's how...

You'll Need

- USB SNES controller – eBay
- PowerBoost 500 **makersify.com**
- 3.7V 500mAh LiPo battery – eBay
- RetroPie **magpi.cc/Retro-Pie**
- Mini HDMI-to-HDMI cable (3m) **amazon.com**
- 2× micro USB cables **amazon.com**



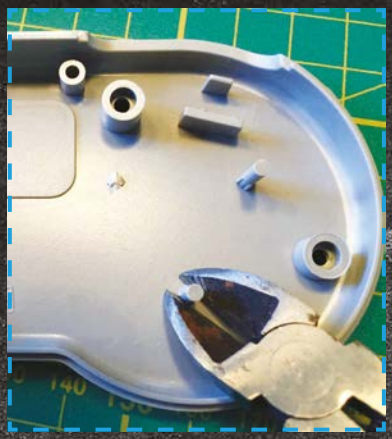
Tip

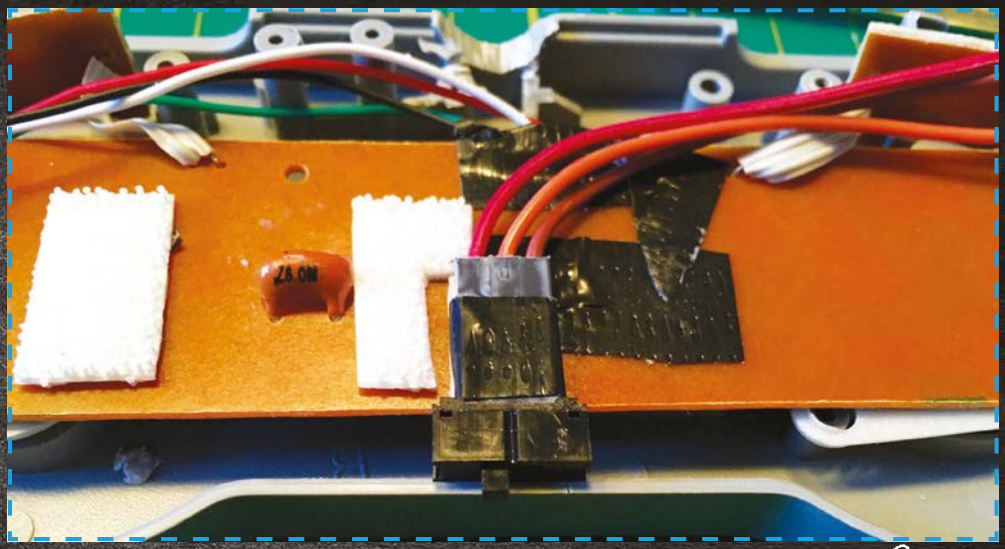
SNES CONTROLLER

Since they're so cheap, we actually picked up a couple of USB SNES controllers from eBay.

This project was inspired by a similar one undertaken by the magazine's features editor, Rob Zwetsloot, a couple of issues back. Rob took a modern USB version of a retro Nintendo Entertainment System controller, opened it up and put the Pi Zero inside. While it worked perfectly, running classic NES titles via RetroPie, we thought that the project could be taken one step further. Classic controllers don't have two cables coming out of them (in this case HDMI and USB for power), so our new target was to create a completely self-contained console in a controller, with only one cable coming out to connect to the HDMI port on a TV or monitor. Thus the Pi Zero-powered SNES Consoller was born...

There's a surprisingly large amount of technology to squeeze inside the controller for this project to work. Fortunately, most USB controllers are largely empty on the inside, using just four tiny USB colour-coded cables connected to a slender circuit board that deals with the inputs (D-pad and six buttons). How do we take inputs from the controller and get them to the Pi Zero? It actually couldn't be easier – all we're going to do is snip off the controller's USB cable and solder the wires directly to a micro USB connector that will slot straight into the USB port of the Pi Zero. Since USB cabling is entirely universal, it just works.





Carving the case

Before we do anything else, though, we need to create some more space in the controller itself. We'll work on the back plate to snip off the support struts first; your mileage may vary depending on which USB controller you buy (there are hundreds), but ours has three supports on each side of the controller. Snip them with some side cutters or chop them with a craft knife, and then sand down any sharp edges so they're flush to the case.

We'll also likely need to remove a couple of the screw hole supports on the controller itself, but don't get carried away yet – you should play it by ear as we try to fit everything inside, so we remove as few of the screw holes as possible.

Once complete, we'll need to snip off the four USB wires connected to the controller's control board, leaving enough play in the cables to ultimately connect the other end to your Pi Zero. To keep things really simple, we're going to connect a slimline mini HDMI-to-HDMI cable to the Zero inside the controller. With our controller, it's actually really easy just to line up the HDMI cable with the USB cable exit on the controller itself, but you'll need to use a craft knife to carve a larger opening and carefully strip back the housing of the mini HDMI connector to get it to fit snugly.

Securing your Pi Zero

The next thing to do is to secure the Pi Zero in place. Its location in the case is decided by the HDMI cable. We want to fix it directly to the controller's control circuit board. There's a good chance there's a couple of raised components on the back, so we're using sticky Velcro Command Strips. This gives us the clearance we need to steer clear of the components, as well as the flexibility to remove or vary the Zero's placement during the build process. Once it's securely Velcroed in place, we can also work on the placement of the battery and the PowerBoost 500, safe in the knowledge everything fits together as it should. We plan to position the PowerBoost board just under the right side of the Pi Zero, so the micro USB power connector on it sits against the bottom-right straight edge at the base of the controller. This allows us to add a power switch in the bottom-centre of the controller and place the battery in the right-hand compartment, which is currently completely vacant.

Since the PowerBoost 500 automatically powers on when you connect a battery to it, we need to fit a power switch to turn our SNES Consoller on and off. We're using a very standard three-pin switch that we got from a basic electronics kit, and three colour-coded male-to-female jumper cables – for power (red), ground (brown), and control (orange) – with snipped-off male ends so they can be soldered to the PowerBoost board later. Next, tape the female ends of the jumper cables together and then simply slot the switch into it. We position



Tip

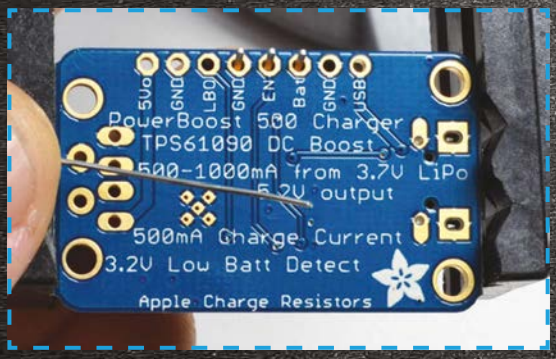
TOOLS YOU'LL NEED

Make sure you have a very sharp craft knife, a decent soldering iron, and a set of small screwdrivers to hand – you won't get far without them.



SOLDERING THE POWERBOOST 500

Using the PowerBoost 500 to power your project is ridiculously easy and convenient, but we do heartily recommend you refer to Adafruit's excellent guide at **magpi.cc/1PmIDWL**. You'll learn about all its features and capabilities, and also find out how to solder the board correctly. As you can see from the image, we need to solder the three highlighted pins for our power switch to work correctly. Once complete and with the battery connected, you'll find that when you flick the switch from nearest the ground pin (brown) to the power pin (red), the unit will power up and down. You'll also find a handy micro USB port on the PowerBoost board – position this against an outside edge of your Consoller, carve a port out of the plastic, and you'll be able to charge your battery - even while you're playing.



the switch in the middle-bottom of the board (under the Pi Zero and next to where the PowerBoost 500 will go), carve out space for the switch to fit flush, and stick it down to the controller's control board to keep it in place. Refer to the boxout on the left to see how to solder it correctly.

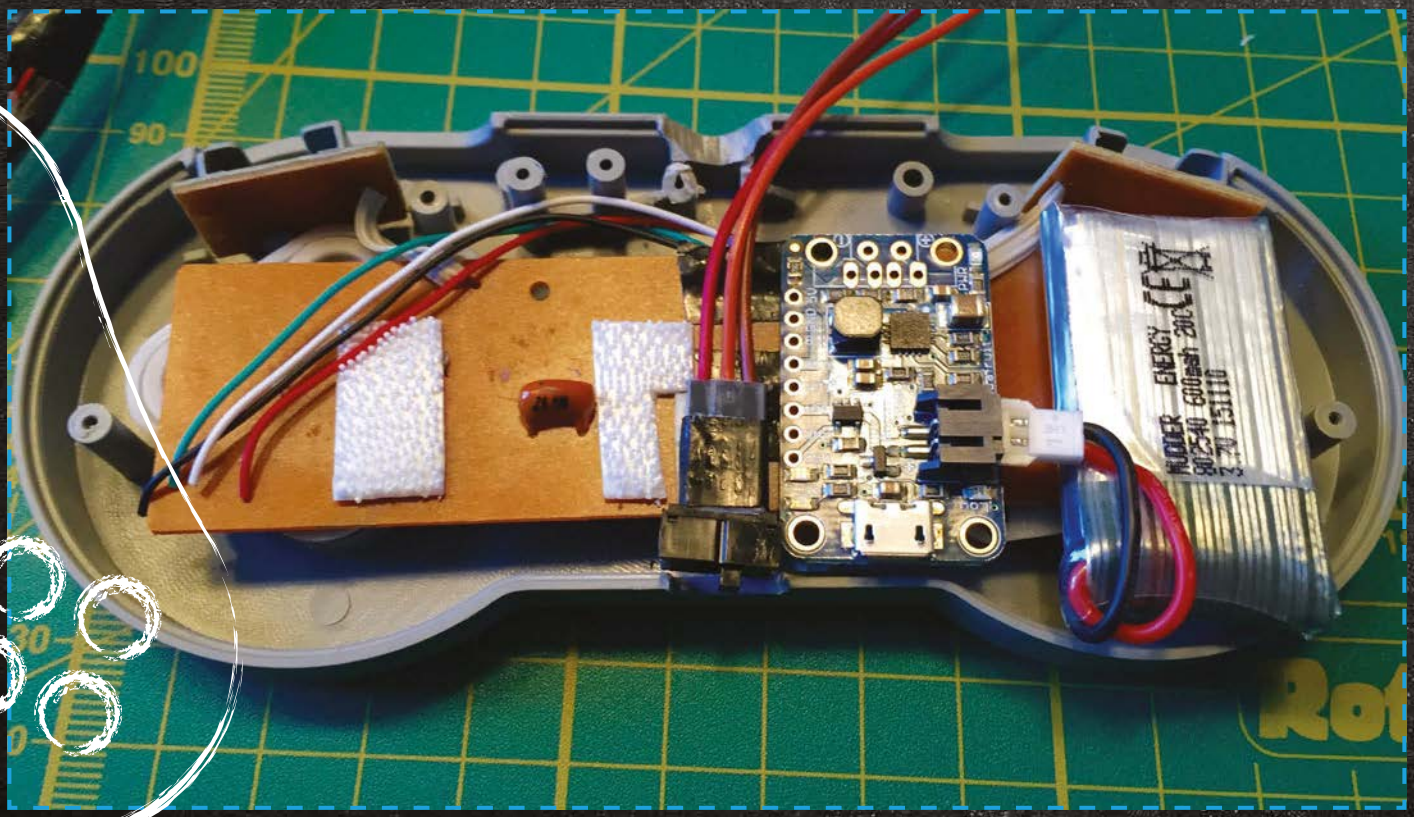
> "At this point, it's wise to ensure you can actually still close the case and find any sticking points"

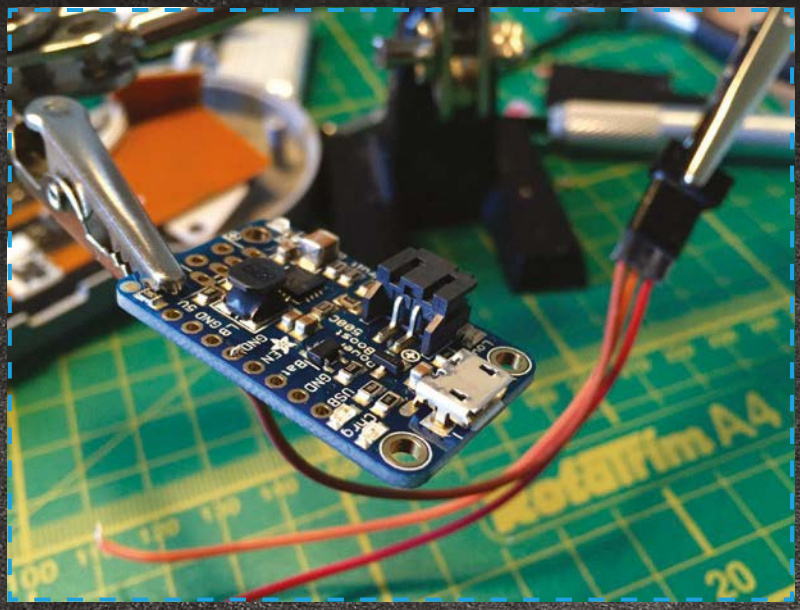
Positioning the PowerBoost and battery

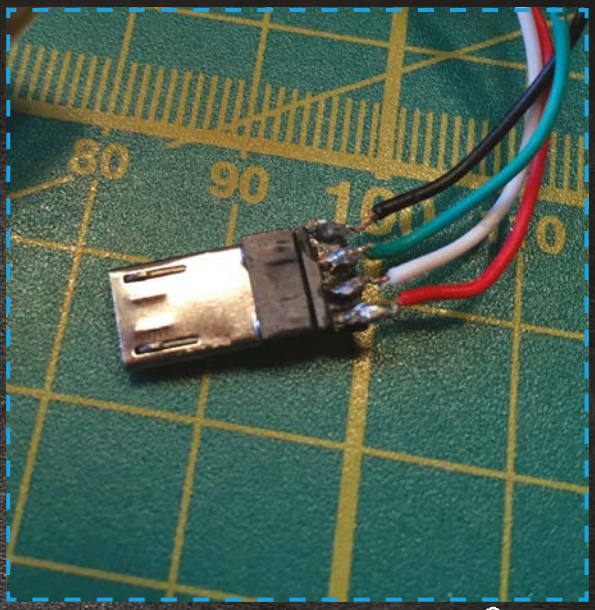
With the switch and PowerBoost prepared, we can work on the positioning of both the PowerBoost and the 3.7V 500mAh battery itself. This is probably the trickiest part of the project. Usually, the power cables on these LiPo batteries are quite short, so you might have fun getting it to fit correctly. Also, since we want to utilise the charging capabilities of the PowerBoost board, the micro USB port needs to be facing a clean edge of the controller so you can carve a suitable hole for it.

We've positioned the PowerBoost right next to our reset switch, with the micro USB facing the bottom edge of the board. Helpfully, this means the battery connector is already facing the correct direction to allow the battery to take up the rest of the compartment on the right. Since part of the PowerBoost sits under the right edge of the Pi Zero when it's in the case, we're using two double-sided sticky tabs to connect the PowerBoost to the controller's circuit board so it actually sits lower than the Zero. The battery itself doesn't need fixing in place, since its size means it doesn't move around.

At this point, it's wise to ensure you can actually still close the case







and find any sticking points that need adjustment. We found that we had to remove some of the support struts for the screws, but we made sure at least two (the far left and right ones) were left intact so we could still securely fasten the case.

Soldering the micro USBs

In terms of practical hands-on work left to do, we're nearly there. The final task is to physically connect the Pi Zero to the USB cables and the PowerBoost. With the power, it's perfectly fine to solder the 5V and ground pins from the PowerBoost directly to the 5V and ground pins on your Pi Zero's GPIO, but since you'll likely to want to use your Pi Zero for something else in the future, we've decided to take the less invasive route of ripping up a separate micro USB cable and soldering that to the PowerBoost instead.

The great thing about USB is that the four colour-coded cables are the same, regardless of USB size: two for power and two for control.



To power the Pi Zero, we only need the red and black cables: power and ground, respectively. These go to the two outer connectors on your micro USB cable – with the longest flat edge of the connector facing up and towards you, ground is on the left. Just search for 'micro USB cable connectors' for a handy diagram to refer to. Simply solder the ground and power cables to the 5V and ground pins on your PowerBoost – it's really that easy.

Connect it to your Zero for testing – assuming you have a correctly configured SD card in your Pi Zero, it should light up as normal when the power switch is turned on.

Connecting the controller micro USB is just as easy. With the longest edge of the micro USB header facing up and towards you, the colour-coded USB cables you snipped off earlier are soldered from left to right as black, green, white, and then red. With the micro USB connector soldered, all you need to do now is preconfigure your Pi Zero with RetroPie and you're ready to go!

> "The great thing about USB is that the four colour-coded cables are the same, regardless of USB size"

Setting up RetroPie

That's it – you've done the hard part! Now all you've got to do is follow the instructions at **magpi.cc/Retro-Pie** to set it up on the microSD card to slot into your Pi Zero. It already has a version that supports Pi Zero. Since there's no room for WiFi in your Consoller, we heartily recommend you set it up using a USB hub with keyboard and mouse, making sure you load all the games you want and scraping the internet for synopses and images in advance. We can't help you acquire games for obvious reasons, but search engines are most certainly your friend. Once complete, simply load it in your project Pi Zero, close up the case and power up for some serious retro gaming action!

Tip
RETROPIE

We love RetroPie. It's the Kodi or openELEC (home theatre front-end) of the retro gaming world. Grab it from **magpi.cc/Retro-Pie**

Tip
3.7V 500MAH BATTERY

You'll need to carefully check the dimensions of your case against the dimensions of your chosen battery to make sure it fits correctly. It'll be snug, but it's easily done.

PI ZERO Reset SWITCH

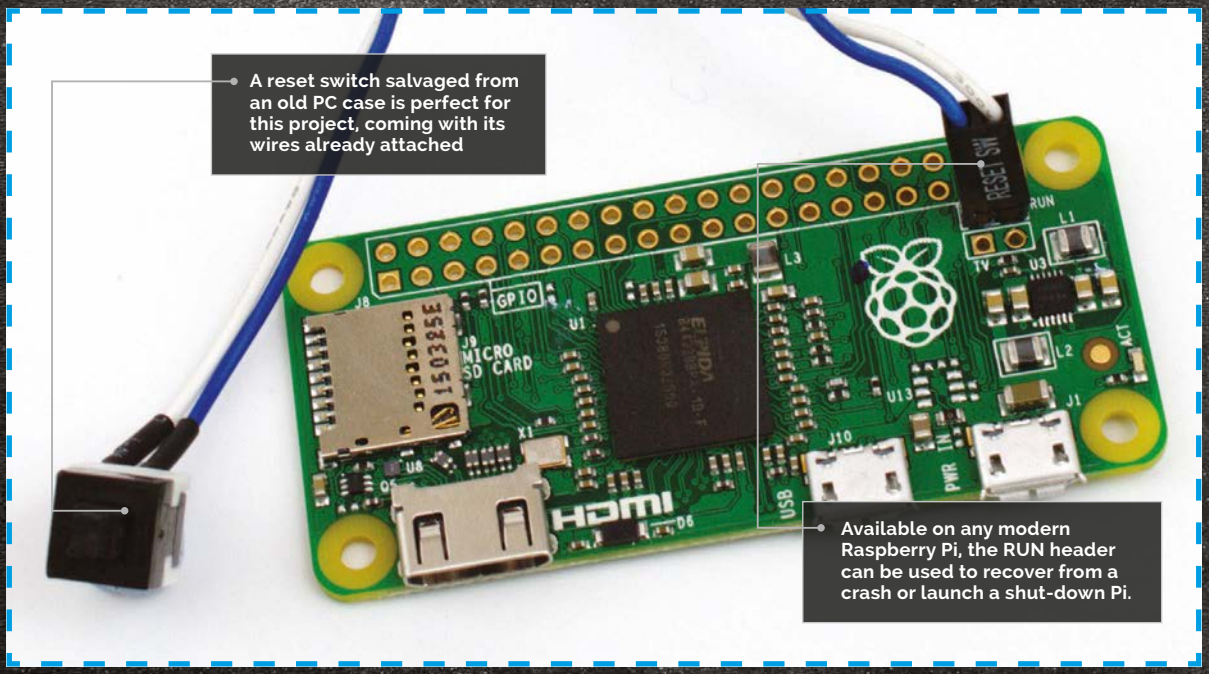
An easy first soldering project, adding a reset switch to your Pi Zero – or full-size Pi – is a great way to prevent crashes and hangs from ruining your day

You'll Need

- Soldering iron & solder **magpi.cc/1Oa5ksX**
- 2× 2.54mm male header pins **magpi.cc/1PCpMVa**
- Recycled PC case reset switch

Or

- Momentary switch, normally open **cpc.farnell.com/SW04563**
- 2× female jumper wires **magpi.cc/1QecOMP**



The Pi Zero's diminutive size, low cost, and tiny power requirements mean it's a great fit for embedded projects. There's only one problem: how to quickly get things back up and running if your code hits an infinite loop or otherwise crashes the Pi Zero, especially in projects where the Pi Zero is locked away from prying hands and powered by a hard to interrupt internal battery.

The solution? A reset switch, wired to the RUN header on the Pi Zero, which can restart the Pi in the event of a crash, or power it on from a shut-down state.

>STEP-01

Find the RUN Header

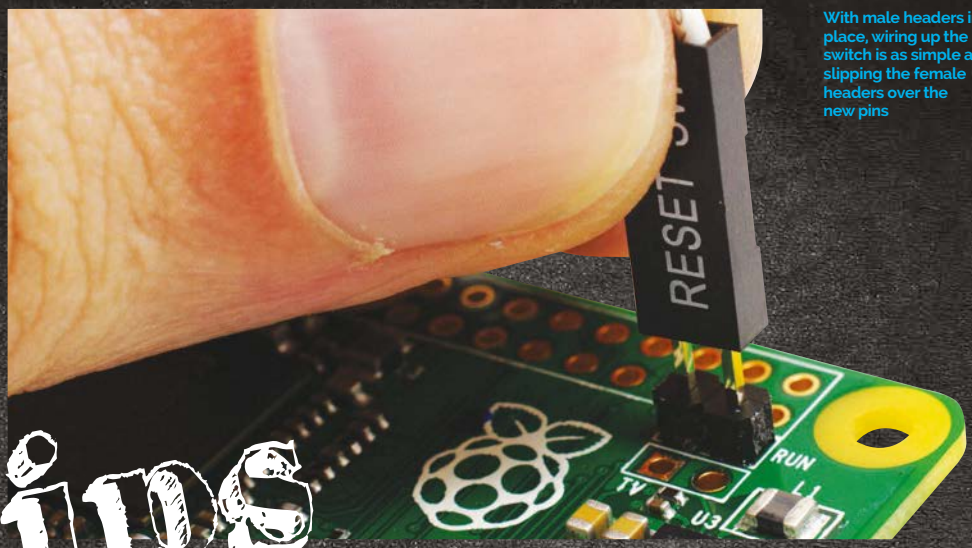
Known as header P6 on early models, the RUN header is linked to a pin in the processor which keeps it in standard running mode – hence the name RUN. When pulled to ground by shorting one contact on the header to the other, the processor leaves the RUN state. To all intents and purposes, this has the same effect as briefly cutting the power to the processor; if the Pi was running, it will reboot, and if the Pi was shut down, it will power on.

The RUN header can be found on the Pi Zero at the right-hand side of the board, sandwiched between the GPIO header and the composite video header (marked 'TV').

>STEP-02

Insert the male headers

The RUN header is unpopulated in all Raspberry Pi models, meaning



With male headers in place, wiring up the switch is as simple as slipping the female headers over the new pins

it doesn't have any pins inserted. While it's possible to solder a switch directly to the header, doing so will block the GPIO and TV headers; it's better to solder male pins into the header and run wires to the reset switch.

Take two of the pins in a bundle and snap them off by placing your thumbnail into the divot in the plastic between pins and pushing. Insert these into the Pi Zero's RUN header from the top, and secure them with tape or Blu-Tack, ready for soldering.

>STEP-03
Prepare the iron

Make sure your iron is safely in its stand, then plug it into a socket. If you're using a cleaning sponge, wet this under a running tap before wringing it out so it's damp but not sodden; if you're using brass shavings, no water is required.

Give the iron a few minutes to heat up, then wipe the tip on the sponge or brass shavings. When clean, 'tin' the iron by melting a small amount of solder onto the tip, then wiping it on your cleaning material again. The tip should look shiny, with no sign of black residue or discolouration.

>STEP-04
Solder the pins

Turn the Pi Zero over to expose the pins you stuck through the RUN header in step 2. Press the iron against the base of one pin, making sure it's also touching the shiny copper ring or square on the circuit board. Count to three then, with your other hand, push some solder against the base of the pin without touching the iron's tip.

The solder should melt then appear to be 'sucked' into the header's hole. If it doesn't, remove both solder and iron, clean the iron's tip, adjust your positioning, and try again.

>STEP-05
Inspect the joint

With both pins soldered, inspect your handiwork before trying to use the Pi Zero. The solder joints should appear to flow smoothly up from the circuit board to the pin, without any gaps. If you can still see copper on the circuit board or the solder isn't sticking to the leg of the pin, apply the iron again to 'reflow' the solder and, if required, add more.

Make sure you haven't connected the two pins together with excess solder. If you have, use a desoldering pump or braid to remove the excess, otherwise the Pi will fail to boot.

>STEP-06
Wire the switch

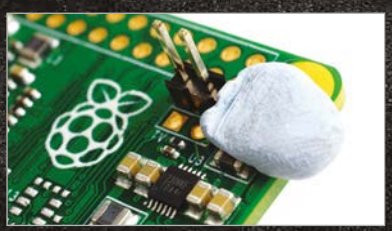
If you're using a reset switch from a PC case, the female header can be inserted directly onto the RUN header's new pins in either orientation. Otherwise, use a pair of female-to-female jumper wires to connect the pins to any switch described as 'momentary, normally open'. These can take many forms, from cheap tactile switches to more expensive vandal-proof devices.

Briefly press the switch to reset when the Pi is running, but be warned this can cause SD card corruption if the Pi was writing to it at the time. If the Pi has been shut down, press the switch to power it back on – this is always safe.

QUICK TIPS

KEEP THE TIP CLEAN

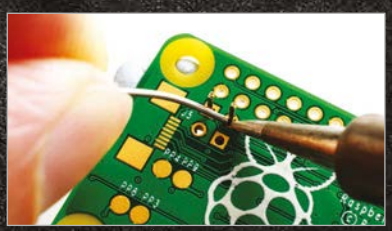
A dirty iron tip will make for difficult soldering. Clean the tip regularly while soldering, and re-tin with fresh solder every so often.



A small piece of Blu-Tack or masking tape can help keep a component in position while you flip the board for soldering

EMERGENCY USE ONLY

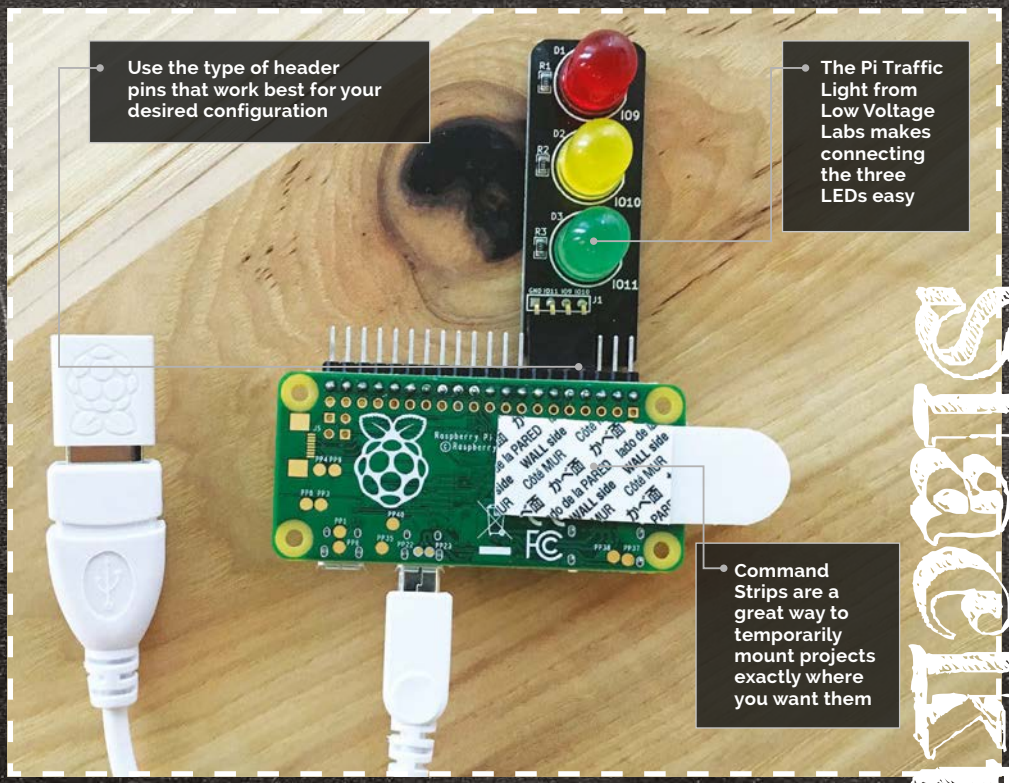
Just like cutting the power, using the reset switch can cause SD card corruption. Use it only when shutting the Pi down another way is impossible!



Any momentary-type normally open switch will work fine for this project, including this high-quality metal anti-vandal switch

SLACKTIVITY MONITOR

Track Slack chat activity with this LED display. You'll know at a glance if there's quiet, light chatter, or raucous debate



To communicate within the Raspberry Pi Foundation we use Slack, the IRC-like chat service. There can be a lot of activity, and it's taken time for people to get their notification settings just right. But we wanted more than what Slack's notifications can provide. We wanted to be able to see at a glance how much chatter was going on among colleagues. Luckily, Slack's API and Python libraries are powerful and easy to work with. Assembly is easy and the materials inexpensive, especially if you use a Pi Zero.

Initial configuration

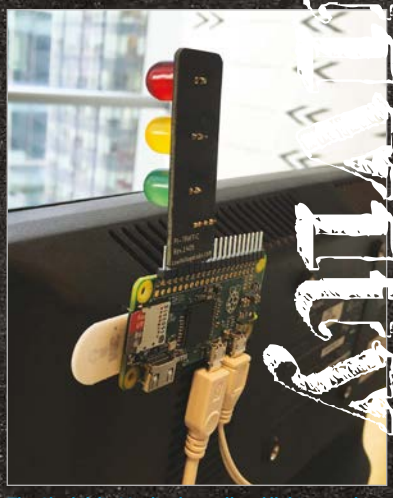
To set up your own Slacktivity Monitor, start from a fresh install of Raspbian. For the Raspberry Pi Zero, you need to use Raspbian Jessie, which can be found on NOOBS or be written to the SD card using the image found on the Raspberry Pi download page (**raspberrypi.org/downloads**). Like with other Raspberry Pi Zero projects, it's probably easiest to get everything configured and installed with a Raspberry Pi 2 or B+ and then move the microSD card to the Pi Zero when you're ready. Make sure to do the necessary updates by going into the terminal and using:

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

You'll need to set some basic configuration options. From the desktop, go to Raspberry Pi Configuration in the Preferences menu. Under the System tab, set your Pi to boot to the CLI

You'll Need

- Pi Traffic Light **magpi.cc/1VamjoL**
- WiFi adaptor
- Right-angle header pins
- Soldering iron and solder
- Blu-Tack or Command Strips



The Slacktivity Monitor is small and light enough to be mounted to a display with Blu-Tack or Command Strips.

(command line interface) and enable 'Wait for network'. You can also set these options with the command line utility raspi-config. Make sure you've also configured your Pi to connect to your WiFi network. Alternatively, you can use a USB Ethernet adaptor with your Pi Zero.

Software setup

From the command line, download the Python Slack Client library:

```
sudo pip install slackclient
```

Download the project code:

```
git clone https://github.com/mrichardson23/slacktivity-monitor.git
```

You'll need to use your own Slack API token in the project code so that the Slacktivity Monitor can authenticate itself. While logged into Slack, go to **magpi.cc/1Val8OF** and click Create Token for the user and team that you want to monitor. Copy that token code into your code, replacing the text SLACK TOKEN HERE.

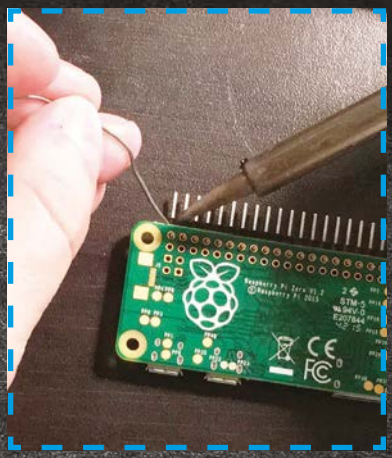
The code keeps track of roughly the last minute of activity. If there have been no messages sent within that time period, then the red LED will illuminate. If there have been any messages (but less than 15) within the last minute, then the yellow LED will illuminate. If 15 or more messages have been sent within the last minute, the LED will turn green. You may need to adjust this threshold to match the typical levels of activity in your Slack team.

Set up the script to run at boot. From the command line, execute:

```
sudo nano /etc/rc.local
```

Before the `exit 0` line, add a line to boot your code:

```
python /home/pi/slacktivity-monitor/main.py &
```



In order to connect the Pi Traffic Light to a Raspberry Pi Zero, you'll need to solder header pins.

When you're done, save the file and shut down the Pi so that you can solder the headers, add the Pi Traffic Light, and attach it where you need it.

Physical build

Solder header pins onto the Pi Zero and connect the Pi Traffic Light to those pins. The code uses pin 15 for red, 14 for yellow, and 28 for green. This is because they're all next to each other along with a ground pin, so the Pi Traffic Light will plug right into the header pins and work like a charm. Depending on how you want to arrange your Slacktivity Monitor, you may want to change the pin numbers and colours.

Use Blu-Tack or Command Strips to attach the Slacktivity Monitor to your display.

When you power up the Pi, the script should start automatically and connect to your team's instance of Slack. Even if there's no recent activity, you may see the traffic light turn yellow initially because the Slack API will return the most recent message when a new client connects.

Going further

The code ignores everything except messages, so you could also adjust the Slacktivity Monitor to react to different types of events. For instance, you can use the red light to indicate when your boss is offline, yellow when she's idle, and green when she's active.

Main.py

Language
>PYTHON

github.com/mrichardson23/slacktivity-monitor

```
import time
from slackclient import SlackClient
from gpiozero import LED

token = "SLACK TOKEN HERE" # found at
# https://api.slack.com/web#authentication

red = LED(15)
yellow = LED(14)
green = LED(18)

red.off()
yellow.off()
green.off()

sc = SlackClient(token)

message_counts =[]

if sc.rtm_connect():
    while True:
        message_count = 0
        events = sc.rtm_read()
        for event in events:
            if event['type'] == 'message':
                message_count = message_count + 1
                message_counts.insert(
0, message_count)

            if len(message_counts) > 600:
                message_counts.pop(600)
                total_message_count = 0

            for i in range(
0,len(message_counts)):
                total_message_count =
total_message_count + message_counts[i]

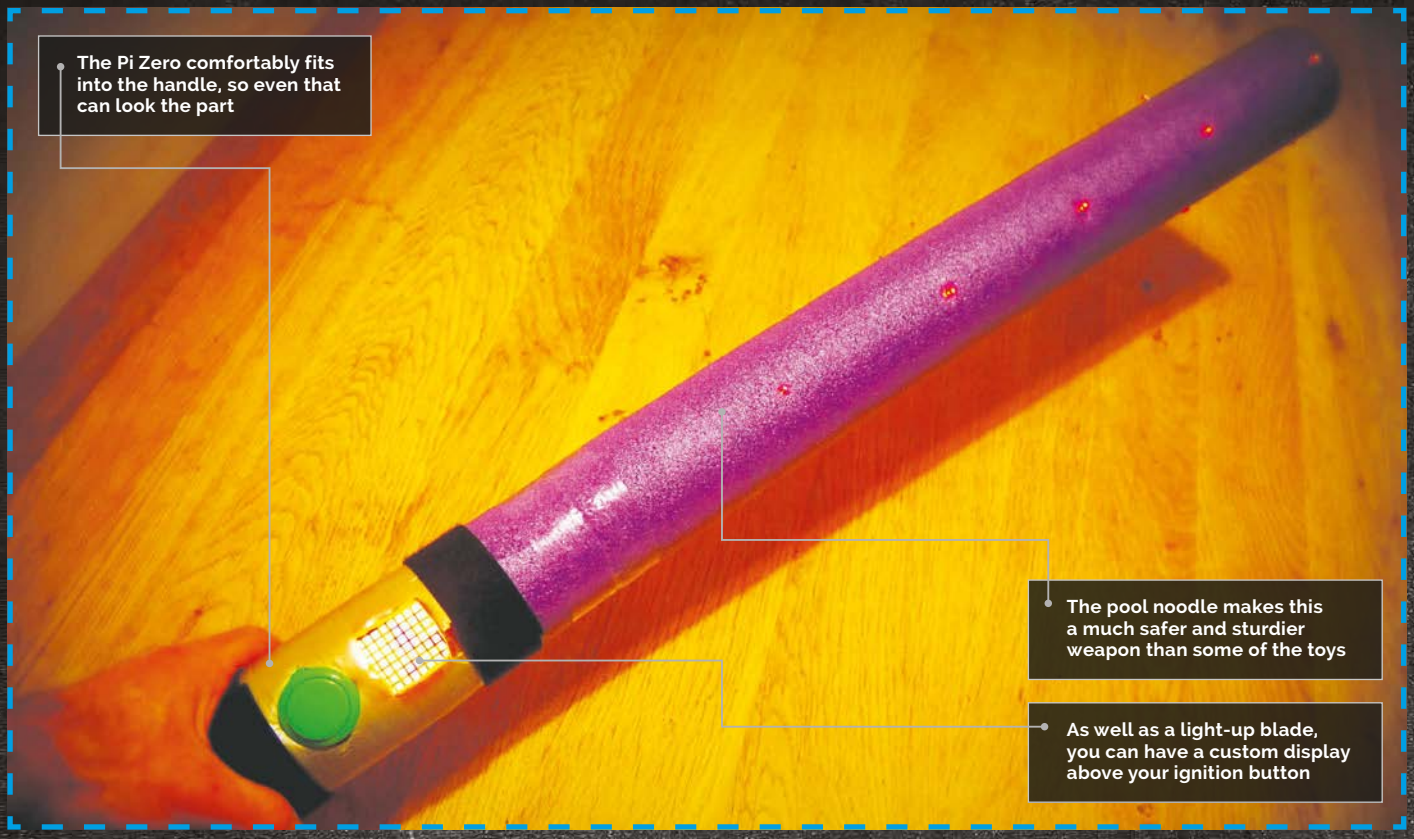
            if total_message_count == 0:
                red.on()
                yellow.off()
                green.off()

            if total_message_count > 0
and total_message_count < 15:
                red.off()
                yellow.on()
                green.off()

            if total_message_count >= 15:
                red.off()
                yellow.off()
                green.on()

            time.sleep(.1)

else:
    print "Error. Invalid token?"
```



PI-SABER

The Force is strong with the Pi Zero, so how about using one to power your own Pi-Saber?

You'll Need

- Adafruit bicolour 8×8 LED matrix **magpi.cc/1ZMEBM4**
- A big button **magpi.cc/1ZMEGPM**
- Battery-powered 3v LED Christmas lights
- Adafruit ADXL345 triple-axis accelerometer **magpi.cc/1ZMEMHh**
- A swimming noodle/woggle
- A power bank
- Soldering kit, jumper wires, and stripboard
- Some glue, a craft knife, duct tape, and some Velcro strips

Recreating your favourite Jedi battles is lots of fun, but many toy lightsabers can be quite delicate. This Pi-Saber should be tough enough to survive some energetic re-enactments. The LEDs that produce the powered-up effect will also fluctuate their brightness as the Pi-Saber is swooshed around, to maximise the excitement.

We're assuming you have a GPIO header attached. However, for a more robust Pi-Saber, you could solder each connection directly onto the Pi Zero.

Padawan's first steps

First, prepare Raspbian on Pi Zero so we can use Python properly:

```
sudo apt-get install git python-pip python-dev python-imaging python-smbus
sudo pip install gpiozero
```

Now install and test the accelerometer. We're assuming you have an ADXL345, but there are plenty of alternatives that will work with some modifications.

First, install the handy Python library:

```
git clone https://github.com/pimoroni/adxl345-python.git
sudo cp adxl345-python/adxl345.py /usr/local/lib/python2.7/dist-packages/
```

The accelerometer uses the I²C bus, so you'll need to activate that by going into the Raspberry Pi Config program and selecting the 'Enable I2C' option from the advanced options. After a reboot, connect the accelerometer to the Pi as shown in **Fig 1**. Change into the **adxl345** directory and run:

```
python example.py
```

Wiggle the accelerometer around and verify that the displayed values for x, y and z change accordingly.

Lights for an apprentice

The Pi-Saber will have a funky display on the handle that also turns on when you press the main button. There are plenty of neat Adafruit 8×8 LED matrix units that come with their own backpack to make

Python control simple. For testing, you can simply connect them via a breadboard, but for the finished product you'll need a more robust method. A simple option is to use a small piece of stripboard and either solder the wires directly or make use of a male header. As usual, there is a great Python library for the LED matrix:

```
git clone https://github.com/adafruit/Adafruit_Python_LED_Backpack.git
```

Then change into that directory and run the installer:

```
sudo python setup.py install
```

There are several example scripts in the repo that you can use for testing.

The other LED components for the Pi-Saber are the ones that will cover the main shaft. There are plenty of options, but Christmas lights are a great and cheap way of getting a string of pre-wired, low-power LEDs. Obviously, the more LEDs you get, the more dramatic the Pi-Saber effect. However, installing each one into the noodle can be quite tedious and time-consuming, so a string of 20 is a good compromise between aesthetics and sanity preservation.

Snip the wires just before the battery holder, strip back the insulation, and add a female jumper housing to the end of each one. Connect them up to the 3V and GND pins on the GPIO and check that they light up. Then switch one wire from the 3V pin to GPIO pin 17.

Jedi Master saber

The final element of the Pi-Saber is a nice big on/off button. Make up some wires with suitable connections, or simply solder directly onto the button's contacts. Then connect it up to the Pi Zero's GPIO pins (26 and one of the GNDs), as shown in **Fig 1**.

Now it's time to put everything together. Check everything is wired up as shown in **Fig 1** and then power up the Pi. Download the **PySaber.py** code (**magpi.cc/1ZMFVi3**), or type it into a text editor, and run it.

```
sudo python PiSaber.py
```

Nothing should happen until you press the button: the LED matrix should show a quick pattern and then start 'sparkling'. Meanwhile, the LEDs on the string should all light up. Pick up the accelerometer, jiggle it around, and you should see the brightness of the LEDs fluctuate along the shaft.

Once everything is working, add the line:

```
python /home/pi/PiSaber.py &
```

…to the **/etc/rc.local** file (above the 'exit 0' line) so that the code will run whenever the Pi is booted.

Now chop the noodle in two and then slice one half open lengthways like a hotdog bun – don't cut all the way through. Measure your noodle and make a sketch to work out how your particular power bank and button will fit inside. Once you've got this figured out, completely remove that much of one half of the noodle, leaving a wedge. Then mark out where the components go and cut away those chunks of the noodle. While you're at it, cut out a small trench all the way along the rest of the inner noodle for the LED cable.

Now calculate how far apart the LEDs should be to give an even spacing all the way round, and then poke them through; make some pilot holes first with a meat skewer or something similar. Once everything is in place and the wires are tidy, glue the slit shaft back together all the way along the length. Cover the handle parts with duct tape and then strap them together with some Velcro.

That should be it – now you're all set for some epic battles.

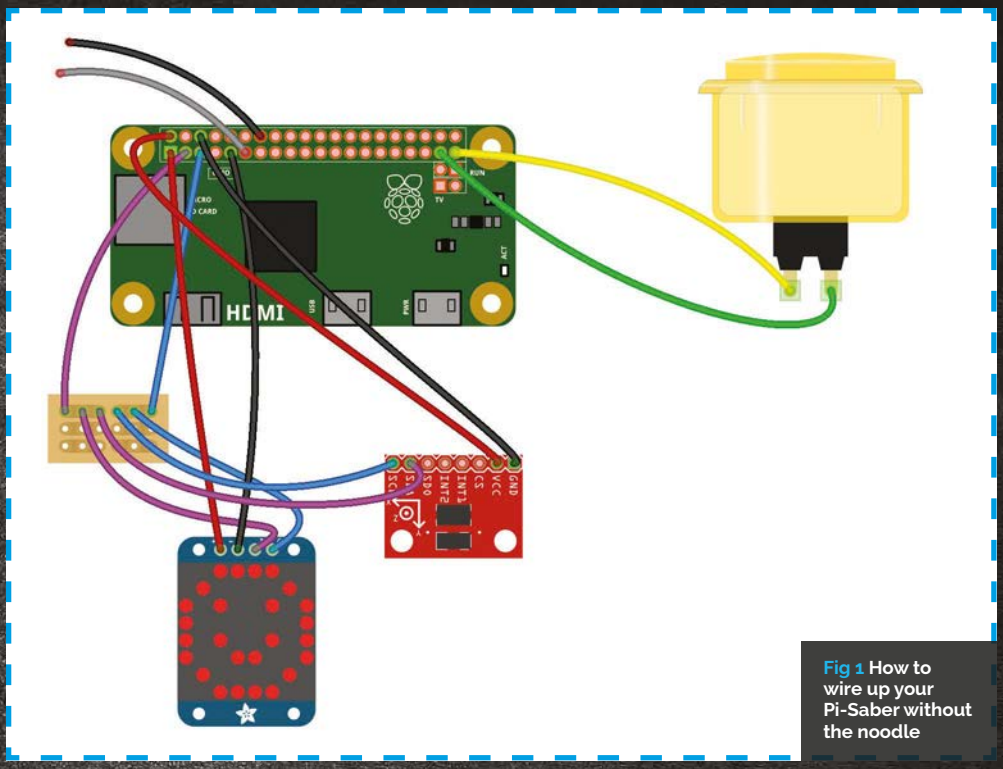


Fig 1 How to wire up your Pi-Saber without the noodle

> "Nothing should happen until you press the button: the LED matrix should show a quick pattern and then start 'sparkling'"

Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1ZMFVi3

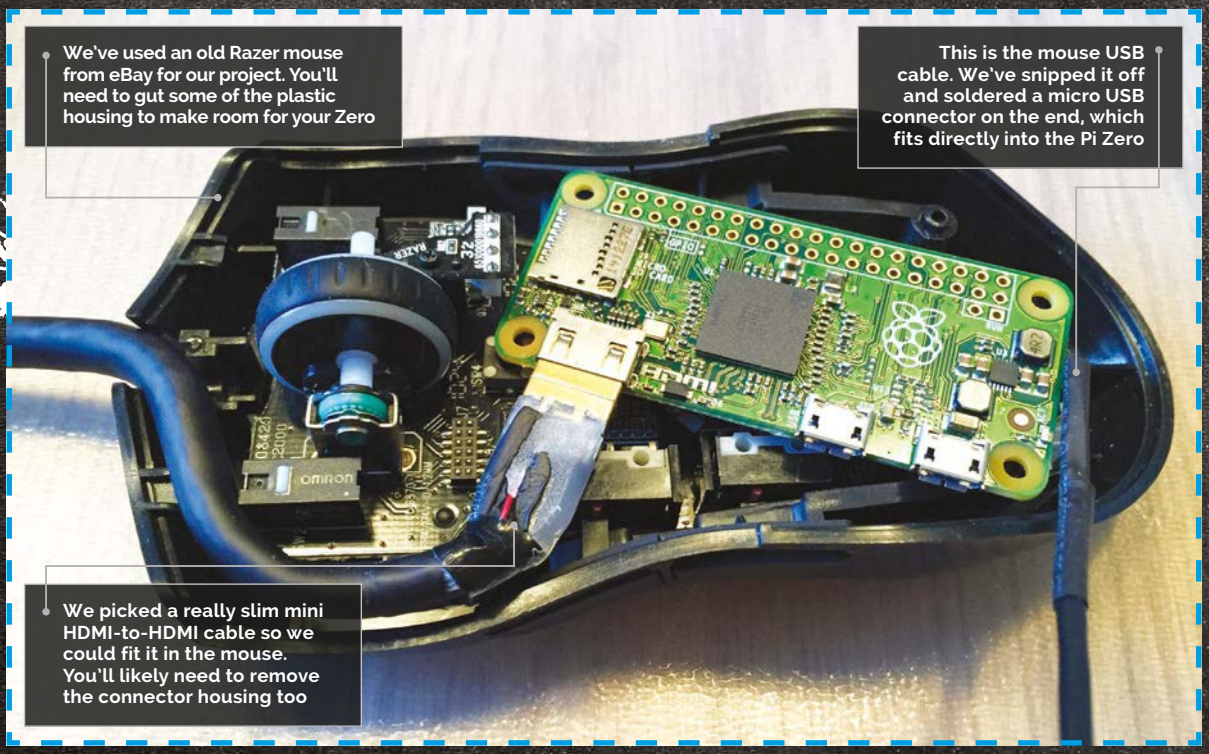
QUAKE III MOUSE

Wouldn't it be great if you could put an entire computer into a gaming mouse? Yes. Yes, it would...

You'll Need

- Gaming mouse – eBay
- Mini HDMI cable **amazon.com**
- 2× micro USB-to-USB cables **amazon.com**
- Quake III files (that you own legitimately)

#pizero



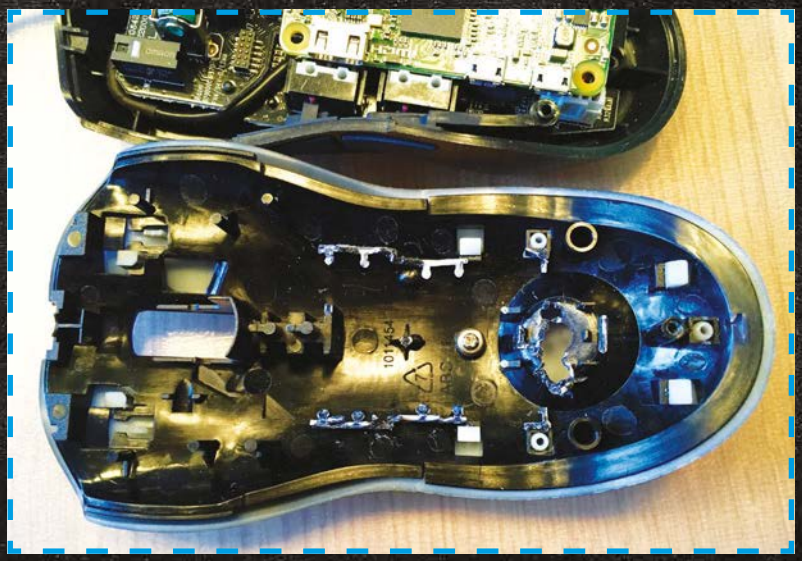
When the Pi Zero was first announced internally, Raspberry Pi's Eben Upton came up with the idea of putting one in a gaming mouse and installing *Quake III* on it. Our heads exploded. Since modern gaming mice have so many buttons, it's actually pretty trivial to tweak the control settings so that running and jumping can be controlled without the need for a keyboard at all. In essence, a lot of what we're doing with this project is an extension of the SNES Consoller project elsewhere in this feature: gut a gaming mouse (most of it is just empty, ergonomically shaped plastic anyway), connect its USB to the Zero internally, and you're pretty much done and dusted.

Of course, these things are invariably easier said than done, and it did take a few shopping trips on eBay to find a suitable mouse contender. We settled on an old Razer Lachesis from around 2009. It has a scroll wheel, two buttons directly behind said wheel, and two buttons on each side of the mouse. It also has enough dead space inside to fit our Pi Zero and HDMI cable. Perfect.

Since there's a lot more going on inside a gaming mouse than a retro controller, we had to do away with our plan to battery-power the *Quake III* mouse quite early on. The end result means that along with the HDMI cable coming out of the front in place of the original USB connector, we've also got a slimline (but ultra-long) micro USB-to-USB cable. Said cables are easily sleeved or tape-wrapped together, though, and most TVs these days offer USB ports that supply enough power to run the Pi Zero. Ultimately, it's a small compromise.

>STEP-01
Gut the mouse
It sounds gruesome, but it's actually easier than you think once you can actually get into the thing. Our Razer Lachesis had a solitary cross-head screw hidden under a sticker on the bottom of the mouse. With that removed, we were able to remove the LED light boards from the roof of the mouse, and trim down some of the support struts and plastic moulding using side cutters. Razer had conveniently used connectors for the LED lights and even for the USB port, so it was trivial to remove everything to create space.

>STEP-02
Soldering the micro-USB
Next, we snipped the USB cable for the mouse, leaving enough play to solder a micro USB connector on the end that fits straight into the Pi Zero. It just works. The cabling is straightforward – with the longest flat edge of the micro USB connector facing up and towards you, the cabling is soldered from left to right – black, green, white, and red. Since USB is entirely universal, there's no way you can get it wrong – just search for 'micro USB cable connector' for example diagrams. It can be fiddly to solder, but with a decent iron and some helping hands to hold it steady, it's pretty straightforward stuff.



>STEP-03
Fitting the Zero and mini HDMI
Now we want to fit the Pi Zero, taking into account that there's going to be a (comparatively) massive HDMI cable attached to it. Our best advice here is to buy the slimmest mini HDMI-to-HDMI cable you can find, but you'll need to use a craft knife to carefully remove the housing without damaging the cable itself. With our Razer mouse, we found that the Zero sat nicely at an angle, with the HDMI cable snaking around the chassis to the front port where the old USB cable was. At this point, it's probably wise to ensure you can seal up the mouse securely, since you might need to carve out a little more of the plastic housing to accommodate it.

>STEP-04
Power your Pi Zero
Sadly, there's just not enough room in the mouse to house a rechargeable battery and accompanying battery controller board. Instead, we're going to run a slimline micro USB cable from the Zero's power port out of the front of the mouse, alongside the HDMI cable. Once the project is complete and working, you can either tape-wrap the two cables together or get some decent heat-shrink sleeving to hold them securely together. You'll need to ensure both cables are long enough so that there's plenty of play at the other end, and both the HDMI and USB cables can be plugged in.

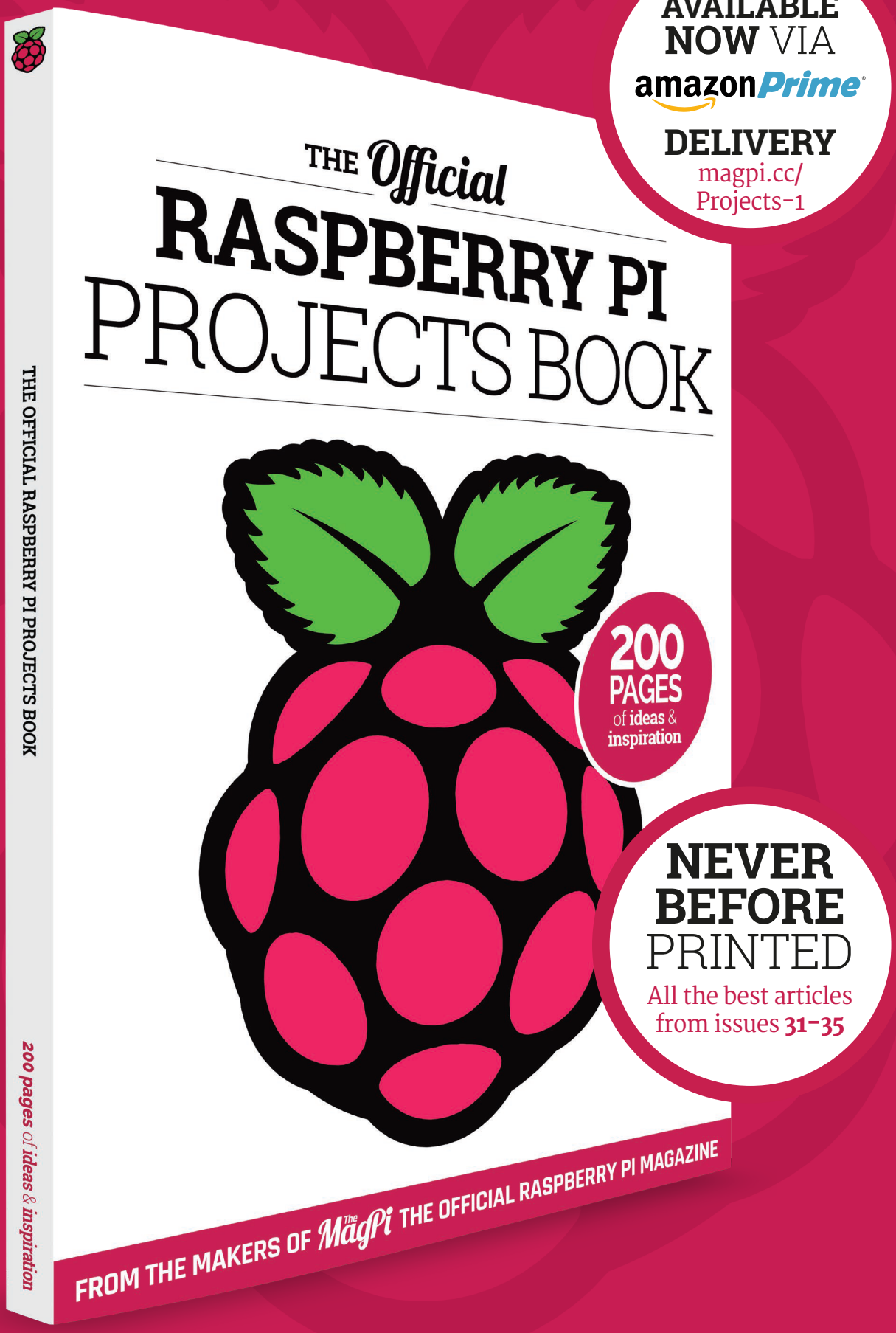


>STEP-05
Installing the software
The next step is to set up a Raspbian Jessie image on your SD card; we advise taking your Zero out of your mouse to do so, and using a USB hub with connected keyboard, mouse, and WiFi dongle. With Raspbian Jessie installed, you need to set up your game (in this case, *Quake III*). Installation isn't exactly a breeze, but you can find full instructions on the process at **magpi.cc/QuakeIII**. You need to recompile the game and acquire the pac files for *Quake III* from a reputable source. Sorry – our lawyers won't let us be more specific than that!

>STEP-06
The ultimate gaming mouse
With power and USB connectivity sorted, and an up-to-date install of Raspbian Jessie and *Quake III* set up, all that's left to do is to pop into the options menu and reassign control to as many of the mouse buttons as you can. We opted to use the wheel to switch weapons, the left two buttons for forward and back, and the opposite two sides for strafing. It takes a while to get used to, but you soon get the hang of it.

Tip

QUAKE III
Raspberry Pi Beginners has a really neat YouTube video showing you an alternative method of installing *Quake III* on your Raspberry Pi. **youtu.be/btSGRnXuAAA**



AVAILABLE NOW VIA amazon Prime DELIVERY
magpi.cc/Projects-1

NEVER BEFORE PRINTED
All the best articles from issues 31–35

£12.99
200 pages of
Raspberry Pi

THE Official
RASPBERRY PI PROJECTS BOOK

Amazing hacking and making projects
from the makers of The MagPi magazine

Inside:

- How to get started with Raspberry Pi
- The most inspirational community projects
- Essential tutorials, guides, and ideas
- Expert reviews and buying advice

Available now
SWAG.RASPBERRYPI.ORG
plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store
GET IT ON Google Play

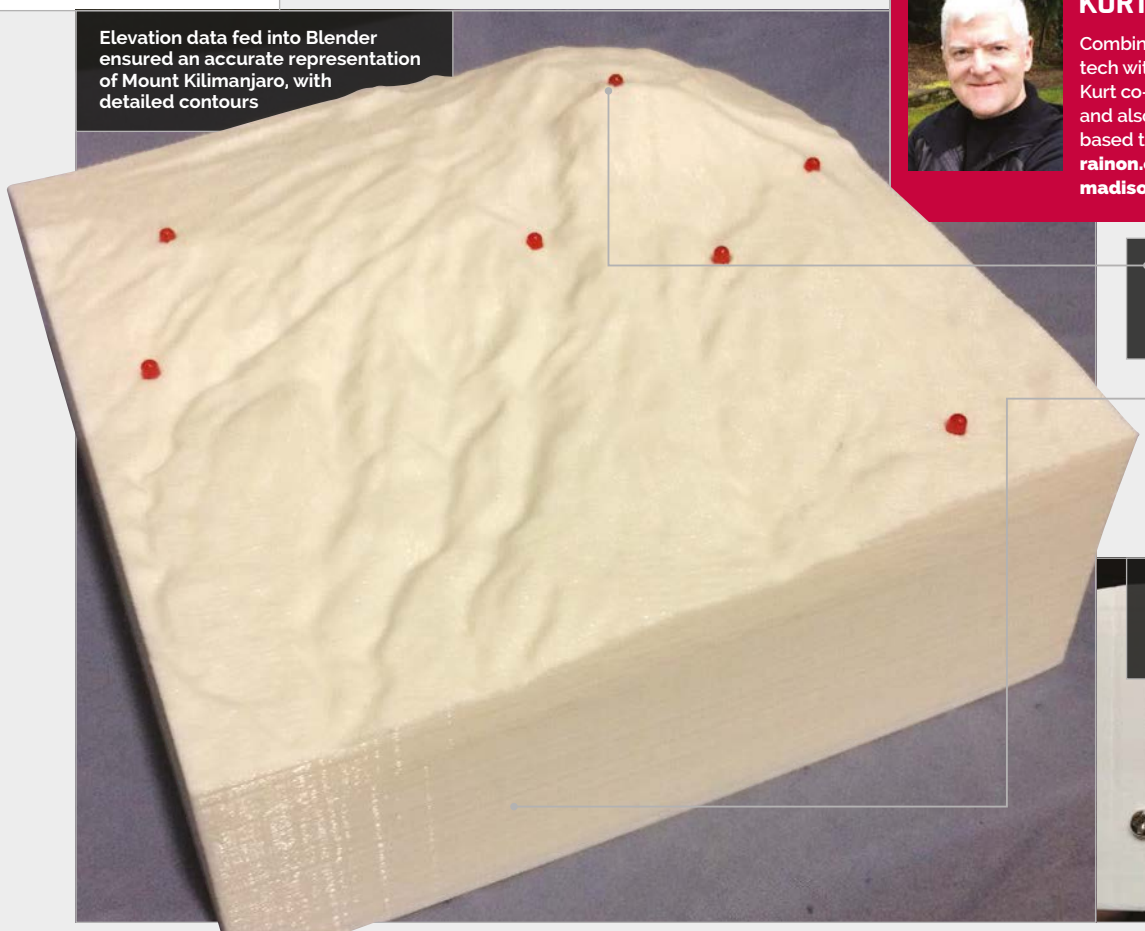


KURT HUNTER

Combining decades of experience in high tech with a passion for alpine climbing, Kurt co-founded Madison Mountaineering and also developed his own RainOn web-based tracking solution.

rainon.com

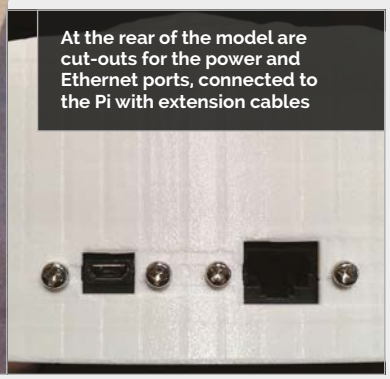
madisonmountaineering.com



Elevation data fed into Blender ensured an accurate representation of Mount Kilimanjaro, with detailed contours

Each of the seven LEDs marks the position of a camp or the summit, and lights up when the climber approaches that point

The Raspberry Pi and all the cabling are completely enclosed within the model, which was hollowed out after printing



At the rear of the model are cut-outs for the power and Ethernet ports, connected to the Pi with extension cables

LOCATION-TRACKING MOUNTAIN MODEL

When **Kurt Hunter** wanted to create a smart 3D model to show family and colleagues his progress while climbing Mount Kilimanjaro, he turned to the Raspberry Pi

Quick Facts

- With its summit 5,895m above sea level, Kilimanjaro is Africa's highest peak
- Two models were created: one for Kurt's home and another for his office window
- The scale used is 1:100,000, so 10mm on the model equates to 1km
- Kurt's previous mountain climbs include Aconcagua and Denali
- He also acted as a Mount Everest base camp manager for a major 2014 expedition

An LED lights up on a scale model placed on a kitchen counter in Washington State, USA, to signify that over 9,000 miles away in Tanzania, Kurt Hunter has reached the summit of Mount Kilimanjaro. Climbing Africa's highest peak had been a long-held ambition for Kurt. "Ever since moving to Seattle 20 years ago, I would look up at Mount Rainier and dream of climbing it," he tells us. "After getting myself in shape in 2004, I did indeed climb it and got hooked on climbing. That year I set a goal of climbing Kilimanjaro, but never got around to it until [2015]."

In preparing for the two-week expedition, Kurt set himself another challenge. He wanted to combine his passion for 3D printing and programming in an interesting way, to enable his family and co-workers to follow along with his climb in a fun, visual experience. To this end, he set about creating a Raspberry Pi 2-powered 1:100,000 scale model of Kilimanjaro, complete with LEDs to mark the spots of all his planned camps, along with the summit itself. The location data would be supplied by a personal GPS locator device, via his own RainOn web-based tracking solution.

Model making

Kurt spent a weekend designing the 3D model in the Blender CAD program, using accurate elevation data for Kilimanjaro gleaned from Viewfinder Panoramas (**magpi.cc/1Sd7psw**) – see **magpi.cc/1Joosv3** for a tutorial on the technique – and calculating the physical offsets for the surface holes required for the LEDs. The model needed to be tall enough to contain the Raspberry Pi, mounted on a removable base panel, and include cut-outs for its power and Ethernet connections (via extension cables). The final

150×150mm model was then 3D printed using a PrintrBot Metal Simple, a process that took 26.5 hours to complete.

Once the support scaffolding was removed to hollow out the model, it was time to wire it up. "The biggest labour component was soldering up the individual LEDs and resistors to the GPIO ribbon cable," reveals Kurt. "The LEDs are held in place on the model by friction and silicon sealant." While he didn't encounter any major problems getting it working, since he'd already made a simple prototype with a breadboard, Kurt did discover a difference between the GPIO pin assignments on Windows IoT Core and Raspbian.

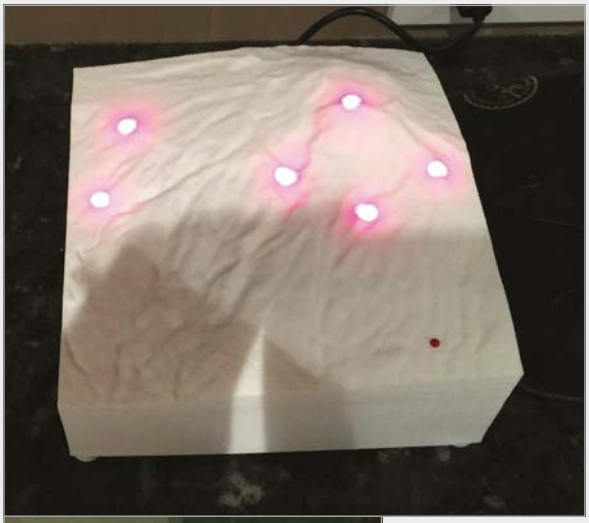
> "Essentially, the system acquires real-time location data from a personal locator device"

Trip tracking

When developing the Pi code for the project, Kurt was keen to evolve the RainOn Adventure Tech (**rainon.com**) web-based tracking system he'd developed for previous adventures as a Microsoft Azure solution. "Essentially, the system acquires real-time location data from a personal locator device, such as DeLorme inReach or SPOT, for a provisioned 'trip' like the Kilimanjaro expedition. It then provides geofence-based waypoint proximity notifications, among other features."

Each trip is a combination of an objective (typically the latitude/longitude of a mountain summit), a route (which can include a list of waypoints), and a start and end date/time. "The system polls the services of the personal locator manufacturer and retrieves the last reported location for each device assigned to an active trip. That retrieved location is tested against the geofence for each of the registered set of waypoints associated with the trip."

The trips and waypoint information are created in advance. "For example, Kilimanjaro using the Machame Route that we climbed has six





Above An LED was lit as Kurt reached each waypoint

Left A second model was put on display in Kurt's office window

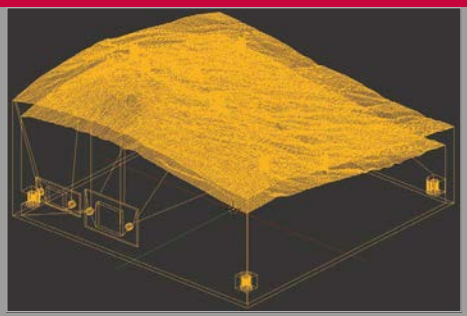
CONSTRUCTING THE 3D MODEL



>STEP-01

Elevation data

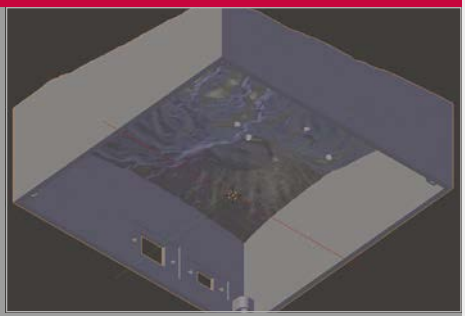
To ensure the accuracy of the 3D model, Kurt obtained DEM (digital elevation model) data from Viewfinder Panoramas (**magpi.cc/1Sd7psw**) for a 15km square area around Kilimanjaro, including locations of the summit and six camps.



>STEP-02

Model in Blender

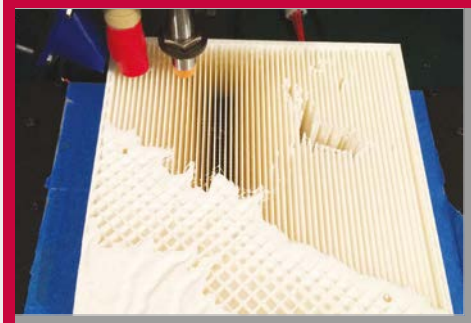
The elevation data was then fed into the Blender CAD program. The height of the model was set to provide the minimum vertical clearance required to accommodate a Raspberry Pi 2 and cabling. A square base was also modelled.



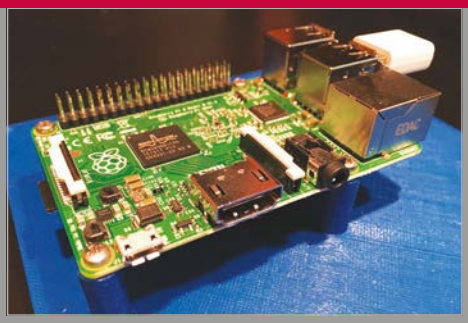
>STEP-03

Punching holes

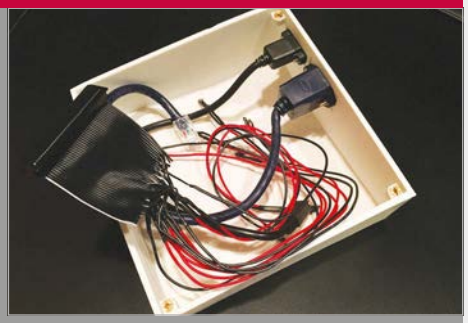
Surface holes for the 3mm LEDs were 'Boolean punched' out of the surface layer. Cut-outs for the power and Ethernet ports, extended from the Pi with cables, were added using 3D model data from DataPro (**magpi.cc/1Sd9ynZ** / **magpi.cc/1Sd9x3l**).



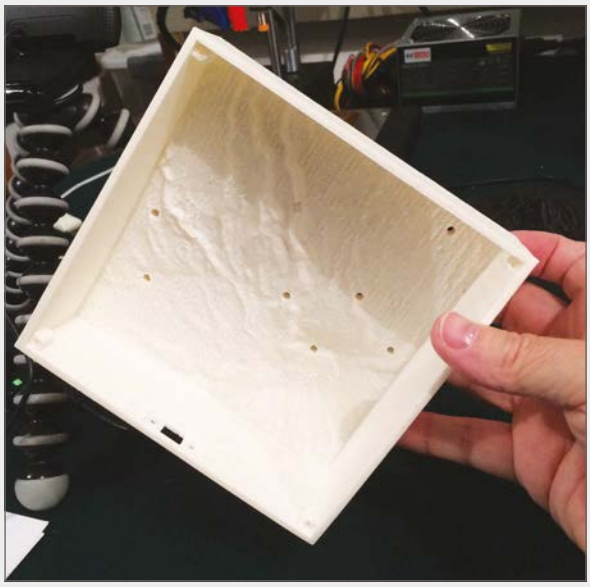
>STEP-04
3D printing
The 3D print of the 150×150mm primary model was completed on a PrintrBot Metal Simple printer, using Microsoft 3D Builder, in just over 26.5 hours. The interior scaffolding was then removed manually to hollow it out.



>STEP-05
Mounting the Pi
To enable easy removal of the Raspberry Pi from the model, it was mounted to the square base via the latter's four bosses, which accommodated McMaster-Carr press-fit threaded inserts to take the M2.5 Pi mounting screws.

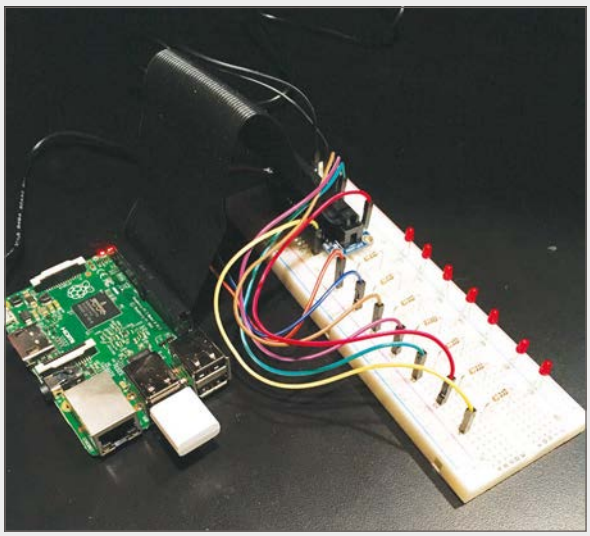


>STEP-06
Wiring up
All LED wiring to the Pi's GPIO pins was via a single 40-pin socket, using a pre-made ribbon cable with one end cut off, and the relevant GPIO pin lines soldered to power and ground hook-up wires for the LEDs, along with resistors.



Above The 3D printed model with scaffolding removed

Below A breadboard prototype was used to test the system



camps (in fixed locations) and the summit, plus the start and end locations, so that trip has nine associated waypoints, seven of which are on my 3D model. All of the data is stored in a SQL Server database which also supports spatial data, so geofence tests and other geospatial calculations are super easy."

> "The system worked in real-time and the waypoint LEDs stay lit after I had reached that location"

Coding the Pi

When it came to programming the Raspberry Pi to process the location data, Kurt ended up using the Windows 10 IoT Core operating system. As a senior partner engagement manager in his day job at Microsoft, Kurt was already well versed in Windows and its Visual Studio developer tools. Even so, he tells us he also developed a Python version of the code under Raspbian, but found the Windows 10 IoT Core / Visual Studio route much easier. "Three things were very attractive about using Windows 10 IoT Core: 1) Visual Studio is awesome (and familiar), 2) I could develop in C#, and 3) the remote debugging/deployment was super easy to set up and use."

Based largely on Microsoft's Blinky example code (**magpi.cc/1ndOkcN**), Kurt's program calls the RainOn Azure service API every ten minutes to obtain JSON location data. If this is within a geofence, set at a 200m radius, for one of the camps or the summit, the corresponding LED is lit.

Testing it out

To check everything was working properly before leaving for Tanzania, Kurt used the Postman test tool to call on his RainOn APIs and simulate data from the personal locator device. "I also use Google Earth quite a bit to find the lat/long of various places and measure distances. So I tested the system by sending simulated locations near the waypoints. When I first developed the geofence/waypoint feature, I did actually use local locations and drive around in my car with the device to test it."





When it came to the actual Kilimanjaro climb, everything worked, although Kurt admits he could have used more error checking and recovery. "The system worked in real-time and the waypoint LEDs stay lit after I had reached that location. However, if the power went out then only future waypoints would light up, as I didn't store the previously reached waypoints in the Pi, nor did I report the history in my web APIs. So, sometime between camp three and four, this happened to the setup I had in my office window, but the one at home worked for the whole climb."

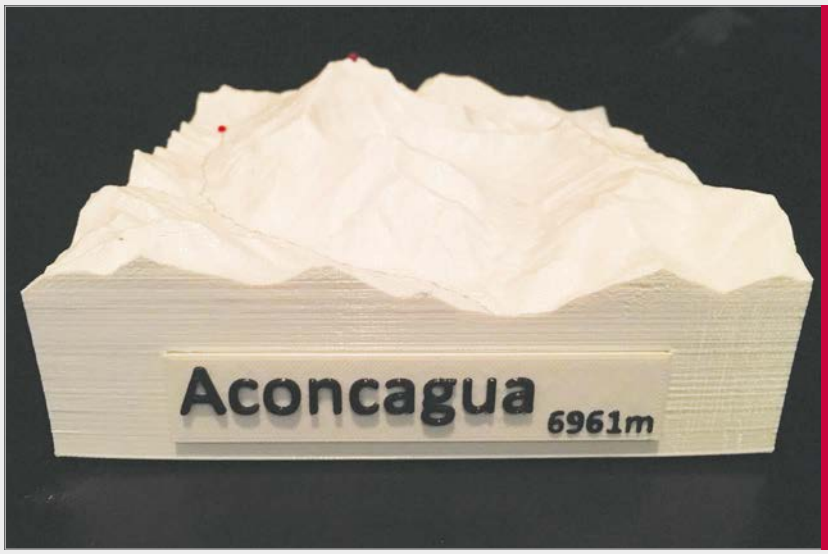
Always another mountain

Kurt admits that a few shortcuts and compromises had to be made in the last-minute rush to get the project completed in time for the Kilimanjaro trip. Given more time, Kurt says he would have liked to use the Pi's HDMI output to provide an animated map view and real-time data and graph displays. He plans to make improvements to the system for future expeditions. "I would very much like to figure out a way to show more discrete climbing progress on the model, rather than a single LED per day. I've thought about a number of ways that I've found to be impractical: light pipe projector from an LCD panel, laser, tight string of tiny LEDs… I'm looking for an idea!"

As for the experience of climbing Kilimanjaro, Kurt tells us it was awesome. "The climb itself takes seven days and you hike through four climatic zones before reaching the summit: rainforest, heather and moorland, alpine desert, and finally arctic above 5,000m – the summit is 5,895m. It's really quite amazing and the local staff make the experience so enjoyable. Every afternoon, when we reached camp, the staff would sing and dance with us… The view from the summit is spectacular as you arrive at sunrise, well above the clouds, and can see for miles and miles."

Above Success! Kurt and his fellow climbers reach the summit

Above left Kilimanjaro rises majestically from the African plain

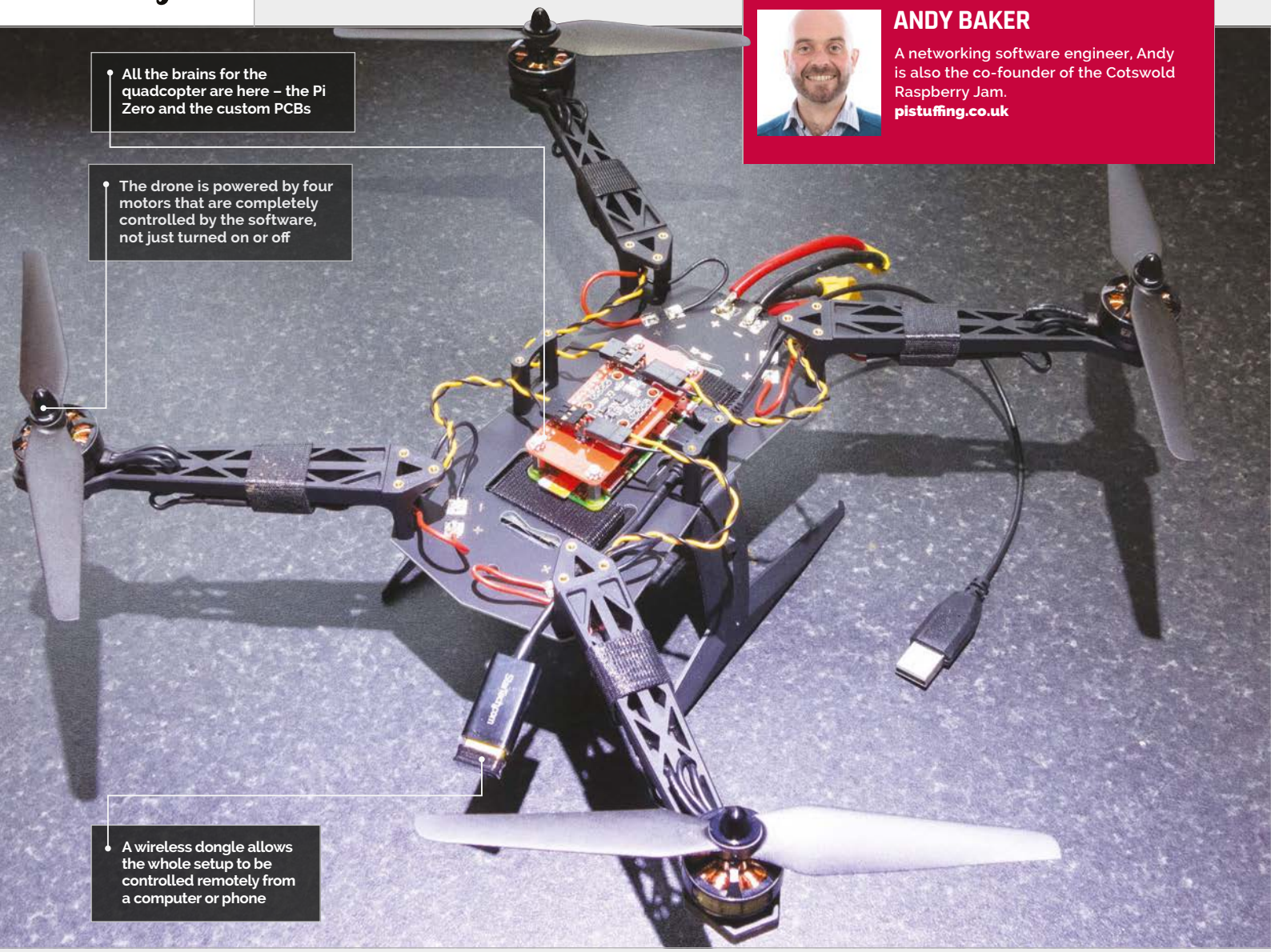


ACONCAGUA EXPEDITION

Kurt has since created another model for a January 2016 expedition to Mount Aconcagua in the Andes – at 6,961m, it's the highest peak in the Americas, and indeed anywhere outside of Asia. While Kurt didn't take part in this expedition, he has climbed the mountain previously. His 1:100,000 scale model of Aconcagua was constructed in a similar fashion to the Kilimanjaro one, with the Raspberry Pi enclosed within a hollowed-out 3D printed model. Six LEDs mark the positions of the summit and camps. Madison Mountaineering, one of the world's leading mountain guide services, are transmitting their real-time expedition locations using an inReach device – **magpi.cc/1J0013Y**.

ANDY BAKER

A networking software engineer, Andy is also the co-founder of the Cotswold Raspberry Jam.
pistuffing.co.uk



Quick Facts

- Andy claims it's taken him three years so far
- An older version appeared in *The MagPi* #19
- It couldn't actually fly for the cover image, though
- Raspbian Jessie fixed a number of programming errors
- It's only his second Pi project

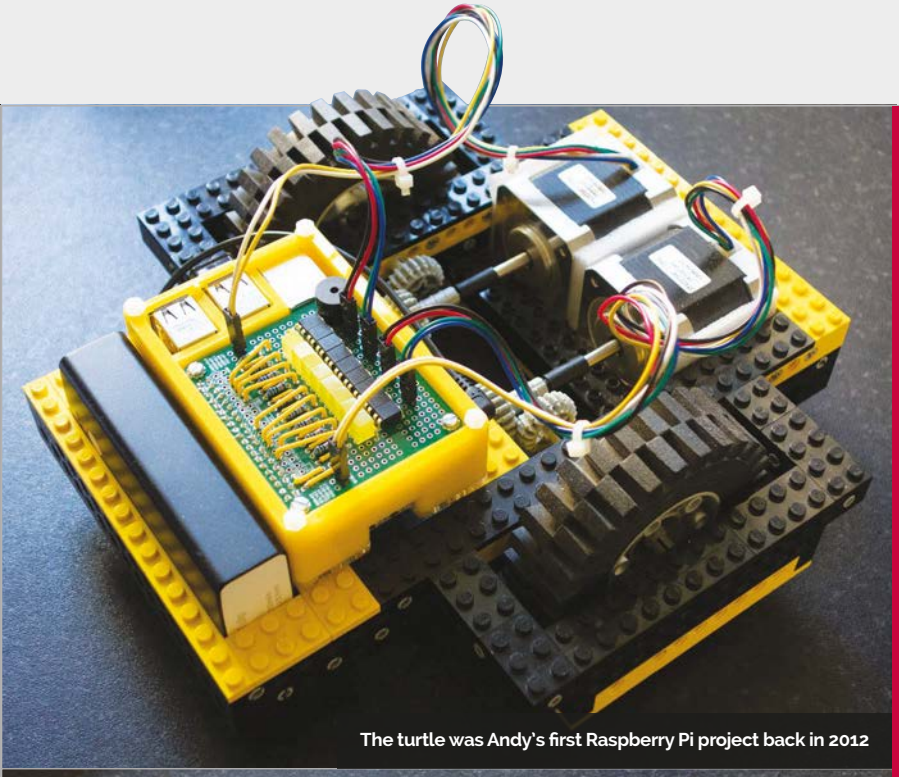
ZOE THE ZERO

The first in what will more than likely be a swarm of Pi Zero-powered drones, Zoe the Zero is the continuation of a long-held dream

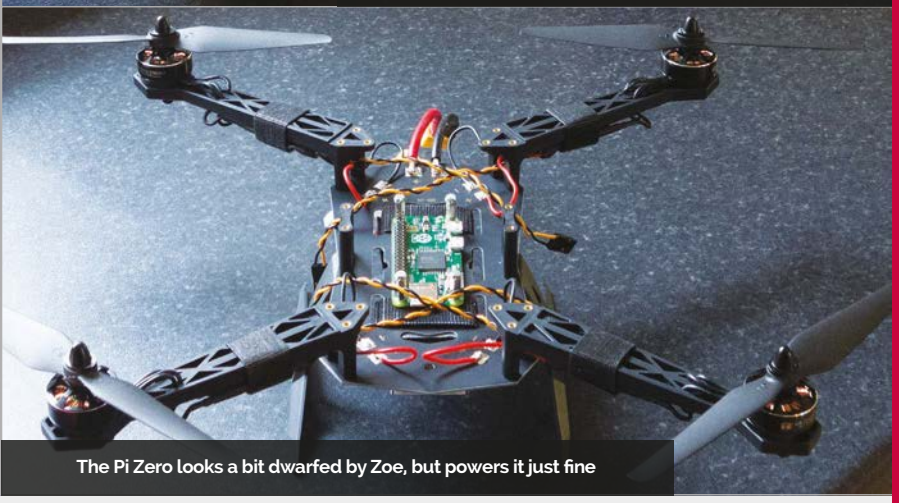
When we were planning issue 40 of *The MagPi* with its cover-mounted Pi Zero, we had a big brainstorm about the kind of projects we could do for features, and maybe also as future covers. One thing that we noted was that the Pi Zero could be great for powering quadcopters and other drones. In just over a month since the issue hit shelves, Andy Baker managed to start the trend and create his own drone.

"I completed my first Pi project (a wheeled 'Turtle' robot) in December 2012," Andy tells us. "I wanted to move on to something that was a real challenge, and quadcopters were just starting to appear commercially. I was completely ignorant of how they worked, so I wanted to fill that gap in my understanding and have some fun at the same time."

Andy has spent the last few years building, testing, and experimenting to create his very own quadcopter from scratch: "I haven't finished yet, primarily because I actually wanted to work it out all by myself, without borrowing anyone else's code or guidance. There was a lot of experimentation and learning involved, including a lot of wasted time taking the wrong direction and backing up when I hit a metaphorical brick wall – or, more precisely, when I couldn't stop



The turtle was Andy's first Raspberry Pi project back in 2012



The Pi Zero looks a bit dwarfed by Zoe, but powers it just fine

the quad from hitting a physical brick wall."

The Raspberry Pi Zero helped Andy bring Zoe back and make it operational again after a few tweaks. The years-long creation wasn't always so quick or simple, though.

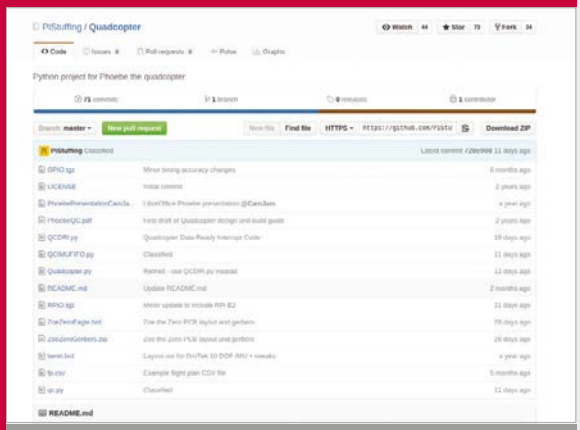
"Other than having a custom HAT PCB, the physical build is very simple due to the off-the-shelf combination of frame, motors, and props," says Andy. "But the software interfacing to hardware was a major challenge and the testing is expensive; the innate instability of a quadcopter means any imperfections ultimately end with a crash and corresponding hardware damage. Despite it 'only' costing about £200 to build the Pi Zero quadcopter from scratch, it's cost me more than ten times that over the last few years to reach this stage."

Interestingly enough, Andy doesn't plan to really use the drone as such, instead treating it as a challenge to just make it work.

"If it ever reaches that stage, I'll almost certainly shelve the project as complete and start on something new – perhaps a fusion reactor, or infinite improbability drive. But at the moment, there's plenty of interesting stuff I can add to the very basic quad I have now to keep me interested for a long time yet."

While it may have taken Andy years, there's now a lot more info and kits available to take a few shortcuts with. Hopefully, in the future, we can show you how to create your own quadcopter.

OPERATING ZOE



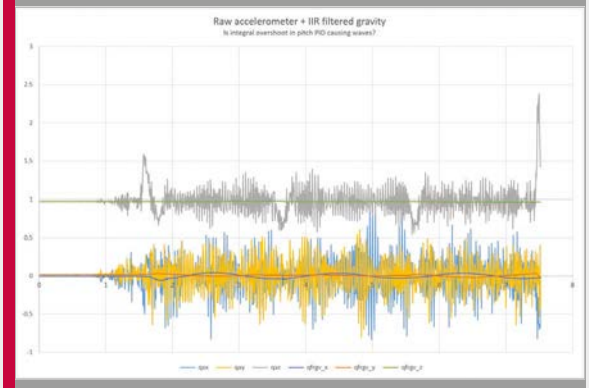
>STEP-01
Starting flight

A preprogrammed flight plan begins for Zoe, with targets for what it should do. This includes ascending at a specific speed for a specific amount of time, hovering, etc.



>STEP-02
In-flight checks

The sensors on the drone provide input as to what it's doing, as a way to make sure everything is operating as it should during the plan.



>STEP-03
Flight corrections

A lot of maths 'glues' the flight plan and data input together to make sure everything is on track, correcting any mismatch in the props/motors.



MATT REED

Matt Reed lives in Nashville, Tennessee and is a creative technologist at Redpepper. **mcreed.com**



Quick Facts

- The battery in a beacon can work for up to a year
- Each beacon has a unique ID number
- The signal strength is used to detect Bean's distance
- It uses trilateration, not triangulation, to detect Bean
- Sniffur can detect Bean in a 50-metre range area

SNIFFUR

Matt always knows where **Bean** the dog is, thanks to his beacon-powered Sniffur system – by **Lucy Hattersley**

We love animals here at *The MagPi*, so when we heard about a Raspberry Pi dog-tracker called 'Sniffur', we had to learn more.

Sniffur uses tracking technology called 'beacons'; these are relatively new devices being placed in department stores to track the precise location of customers.

Sniffur flips this idea on its head, by putting the beacon on a moving object and using three Raspberry Pis to detect where it is. The object in question is Bean, an adorable greyhound adopted by Redpepper, a marketing agency in Nashville, Tennessee.

"We're a culture-first company," says Matt Reed, the project's maker. "A lot of us have dogs or pets in some form, so part of us being happy is knowing our pets are happy. It doesn't hurt that our CEO also has two dogs.

"Bean is absolutely the sweetest dog. Very shy, timid, loving. She wanders around into people's offices and just stands next to their desks waiting for rubs, then just wanders on to the next spot."

But all dogs like to go out and play. "When [greyhounds] do, they are very hard to catch because they are so fast," Matt tells us. "They are the second-fastest land animal after the cheetah: top speed 43mph. Greyhounds also have a high prey drive; if they see a squirrel then it's 'bye bye'.

"The need to know where she is at any moment and see if she's close to the front doors is the reason Sniffur was built," continues Matt. "We have done retail experiments using Estimotes (**estimote.com**) and constantly keep a stock of beacons available in our lab for experimentation.

"I grabbed one of the Estimote beacons from the lab and zip-tied it to Bean's collar. Usually, beacons are attached in stationary



The Estimote beacon transmits a steady ping that can be picked up by the Rasbperry Pis

physical locations, but Bean is now a mobile beacon transmitting a polling signal everywhere she goes.

"I put a Bluetooth USB dongle into three Raspberry Pis and coded them to listen for Bean's Unique Beacon ID," reveals Matt.

One value transmitted by a beacon is RSSI (received signal strength indicator). This is the distance of a beacon from the antenna. "I've referred to it as triangulation because everyone knows what that means," says Matt, "but I've recently learned on Hackaday that 'trilateration' is the correct term, because I'm calculating distance instead of geometry."

> "'Trilateration' is the correct term, because I'm calculating distance instead of geometry"

Matt built an app for the iPad to display Bean's location. "It's just an HTML, JavaScript and CSS page that connects over the local WiFi network to the three Raspberry Pis," he explains.

"I think this could be a fun project for anyone looking to familiarise themselves with beacons. A cheaper version would be to use one Pi to generate a single radius distance from that antenna. You could also set an antenna by an exit and just throw an alarm, flash lights or lock the door, whenever the roaming beacon comes within range.

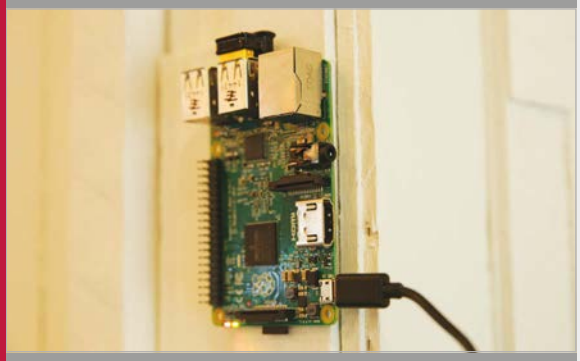
"Hmm," ponders Matt, "that is actually not a bad idea. I think Sniffur is fun from a visualisation standpoint, but from a practical standpoint something more proactive might be better."

BUILDING A SNIFFUR SYSTEM



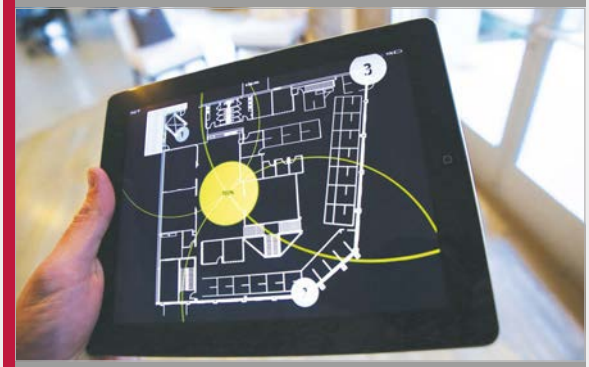
>STEP-01
Beacon

A Estimote beacon is attached to a collar and placed around the dog's neck. Powered by a single cell battery, it'll give away her position for up to a year.



>STEP-02
Three Raspberry Pis

Three Raspberry Pis are set up, with WiFi and Bluetooth dongles to act as antennas. They measure the distance of Bean to each Raspberry Pi, which is used to determine her position.



>STEP-03
Tracker app

A webpage (wrapped as an iPad app) connects to the local WiFi network and obtains the values from each Raspberry Pi. These are used to display the position of Bean on a map.



KRIS TEMMERMAN

A freelance developer who creates interactive displays for museums and advertising agencies. He's building a robot in his spare time.
neuroproductions.be

BALANCE BOT

A sociable, balancing robot built with a Raspberry Pi to power it. How tricky is balancing a robot via code, though?

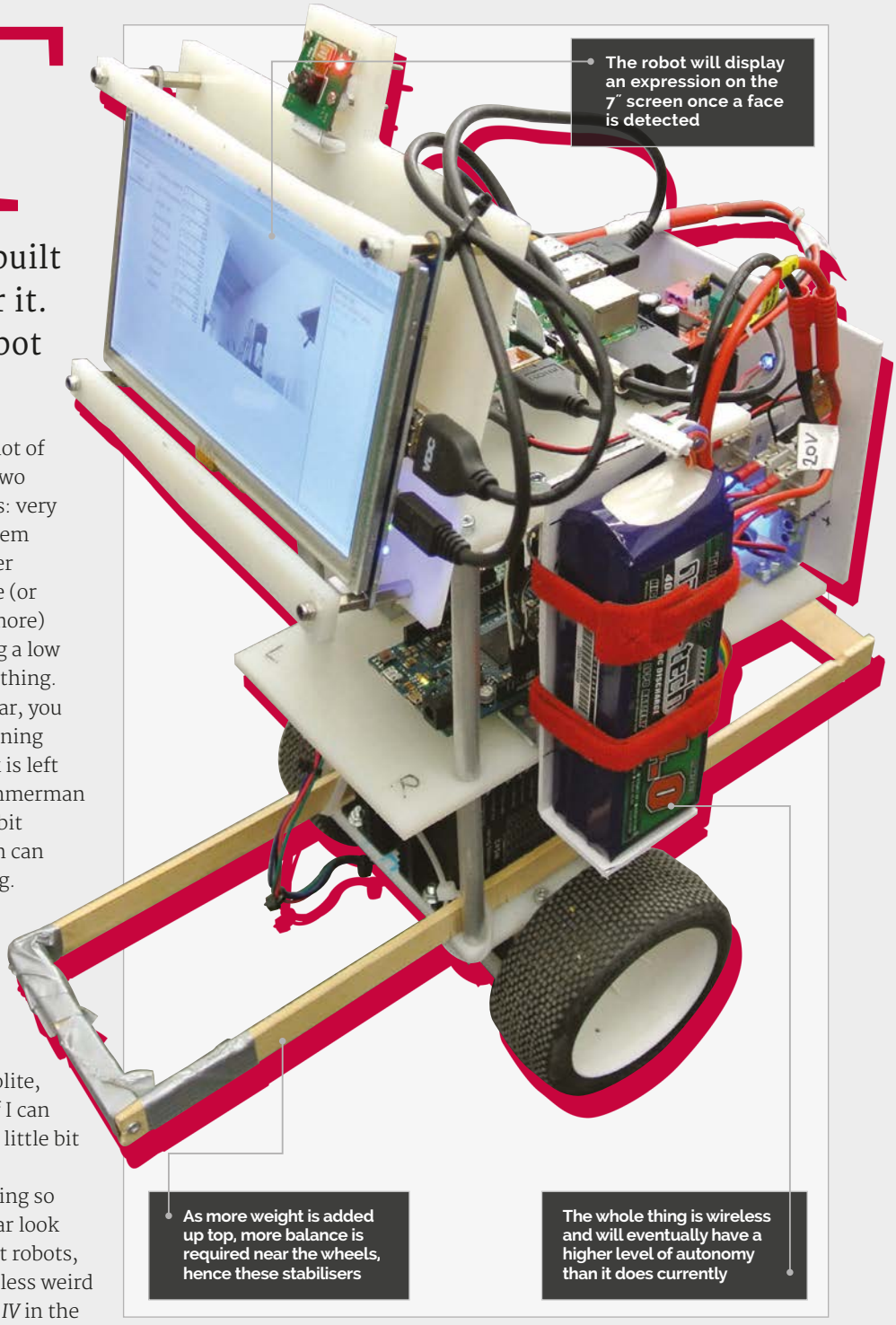
Quick Facts

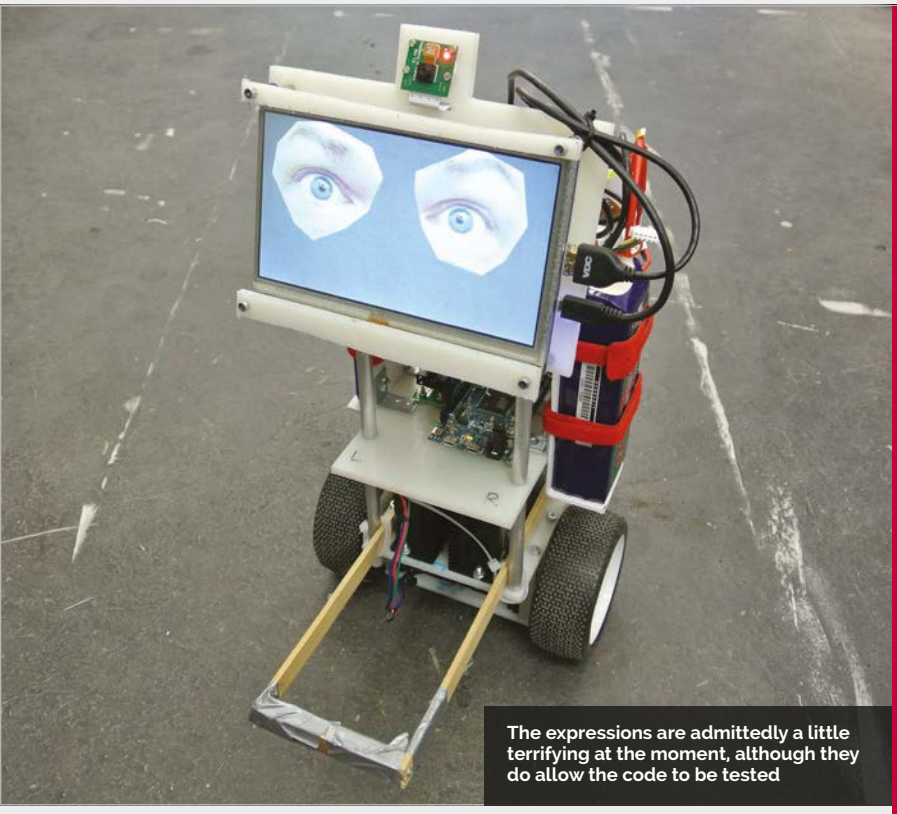
- The robot has currently taken 20 days to build
- Its reactions to detected people still need work
- The screen displays faces when detecting people
- Arms are planned for the robot
- It uses a Pi camera for the face recognition

There's a reason a lot of robots don't use two wheels or two legs: very simply, it's difficult for them to balance. It's much easier and sturdier to be on three (or more) wheels or four (or more) legs; generally, just having a low centre of gravity is a good thing. For two wheels in particular, you need some way of maintaining balance. Usually, this task is left up to humans; as Kris Temmerman shows us, though, a good bit of coding and construction can accomplish the same thing.

"The idea came after seeing some social robots, like Nao and Pepper," Kris tells us. "Technologically they're great, but I think they are also kind of boring. Too polite, too cute. I wanted to see if I can make a social robot with a little bit more character."

The robot is self-balancing so that it can achieve a similar look to these personal assistant robots, and hopefully look a little less weird than the robot from *Rocky IV* in the





The expressions are admittedly a little terrifying at the moment, although they do allow the code to be tested

process. It has facial recognition built in so it can detect and react to people properly.

"For me it's a sandbox – a way to try all kinds of hardware and software," Kris explains. Since he wanted to do some intensive computations, in combination with a display and sound, he opted to use a Raspberry Pi 2. "The quad-core [CPU] came in really handy. I have the face recognition running in a separate thread on its own core, so the main process that handles the graphics, sound, and motion can run smooth and uninterrupted. And I still have two cores left over for future improvements."

> "It has facial recognition built in so it can detect and react to people properly"

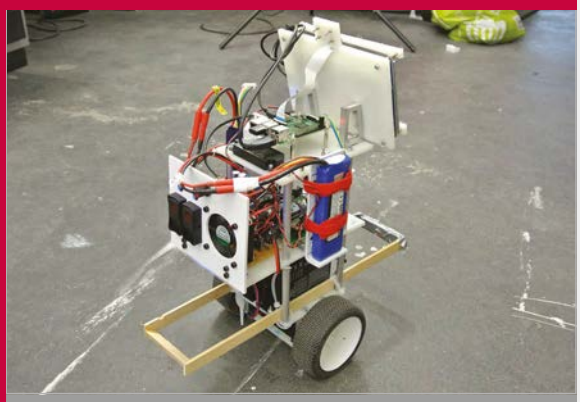
While Kris says it's certainly not a project for beginners, it's not as hard as it seems. "If you have a reasonable amount of programming experience, it's definitely something you can make. You can find plenty of information about balancing robots online. For the face recognition I use OpenCV, an open-source computer vision library that has the hard parts already solved for you."

Kris made lot of the body and chassis himself with nylon sheets, together with some spare motors and drivers. There's an Arduino layer in the build as well, to aid with the balancing.

That works well and there are a few online videos you can watch of the little robot racing around on its two wheels (**magpi.cc/1URY7zT**). The robot still has some work left to be done. In particular, Kris is looking at mounting a lidar onto it so that it can have a better sense of its environment, allowing it to move around autonomously.

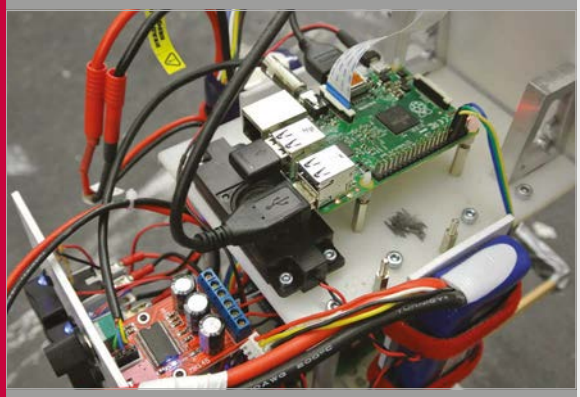
"This was my first project with a Raspberry Pi, but I really like that it has a full-blown Linux distro… I'm definitely going to use it more in the future."

OPERATE A ROBOT



>STEP-01
Balancing act

Once turned on, the robot starts to balance itself to stay upright. Much like the Boston Dynamics robot, pushing it will cause it to right itself.



>STEP-02
Remote control

The robot is not currently fully autonomous, so to control it you have to use a wireless keyboard connected to the Raspberry Pi. It will maintain balance as you drive it around, though.



>STEP-03
I see you

The camera can detect faces. If one appears in its line of sight, the robot will look in the direction of the face and react to it with noises.

SUBSCRIBE TODAY!

Subscribe to the Official Raspberry Pi mag today for a whole host of benefits

Subscription benefits

- Save up to 25% on the price
- Free delivery to your door
- Never miss a single issue
- Get it first (before stores)

PI ZERO #40 IS STILL ON OFFER
Call today!

SAVE UP TO **25%**



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)

Direct Debit

£12.99 (UK) (quarterly)

How to subscribe:

- **magpi.cc/Subs1** (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 & save money

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

Mag#42

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.

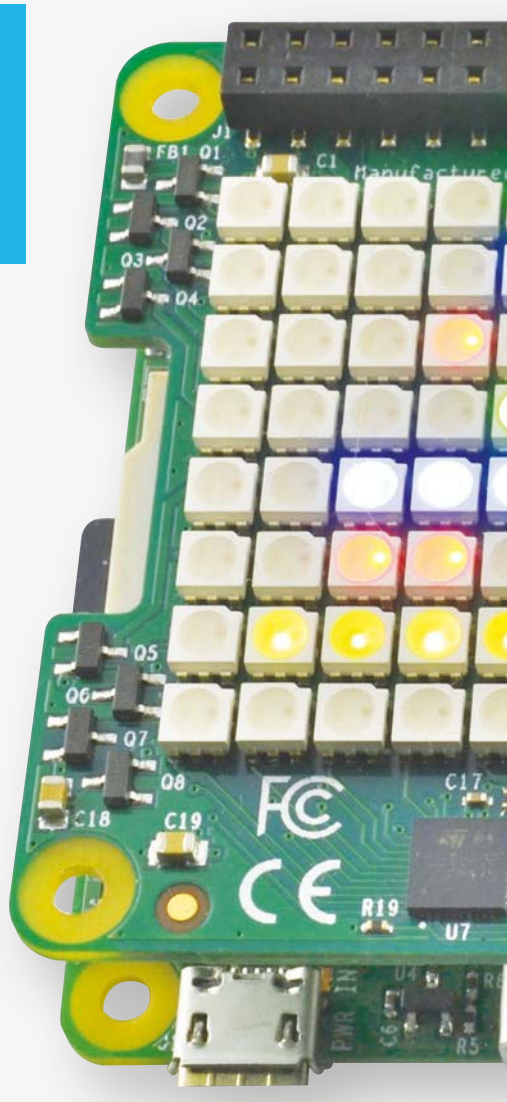
EVERYDAY ENGINEERING PART 12



SIMON MONK

Simon Monk is the author of the *Raspberry Pi Cookbook* and *Programming the Raspberry Pi: Getting Started with Python*, among others.
simonmonk.org
monkmakes.com

BINARY CLOCK



You'll Need

- Raspberry Pi with 40-pin GPIO header
- Sense HAT or Pimoroni Unicorn HAT
- USB WiFi adaptor or Ethernet connection

Solve real-world electronic and engineering problems with your Raspberry Pi and the help of renowned technology hacker and author, **Simon Monk**

This is a very simple project to make. In fact, all you really need to do is plug the Sense or Unicorn HAT onto the Raspberry Pi's GPIO header and install the software.

This project displays the current time in binary. The rows of the display are (from top to bottom) the last two digits of the year (e.g. 15), the month number, the day of the month, the hour (24-hour format), the minute, the second, and finally hundredths of a second.

Reading the binary clock

To read a binary clock, you either need to be good at recognising binary numbers or use a bit of arithmetic to calculate the time, by adding up the binary digit values for the LEDs that are lit on a particular row.

For example, the first row displays the year in two-digit format. So, if LEDs at positions 8, 4, 2, and 1 are all lit, then the year is 8+4+2+1 = 15. In 2016, just the digit position 16 will be lit.

The colours of the LEDs have no significance – they are different to make the display more interesting.

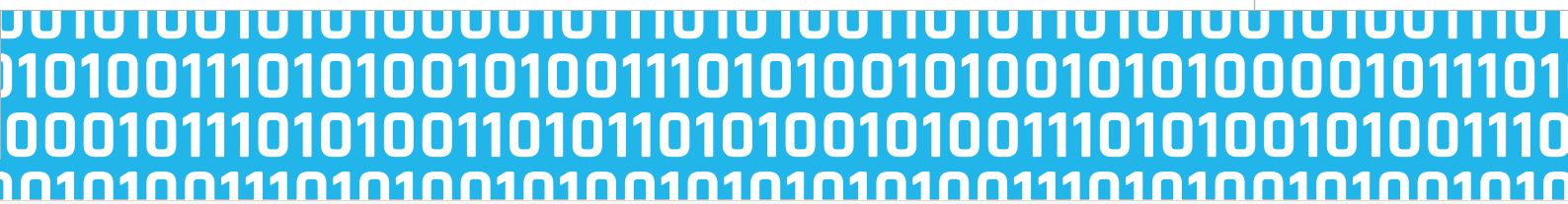
If you're using the Sense HAT, you'll need to install the software for the HAT using the following commands:

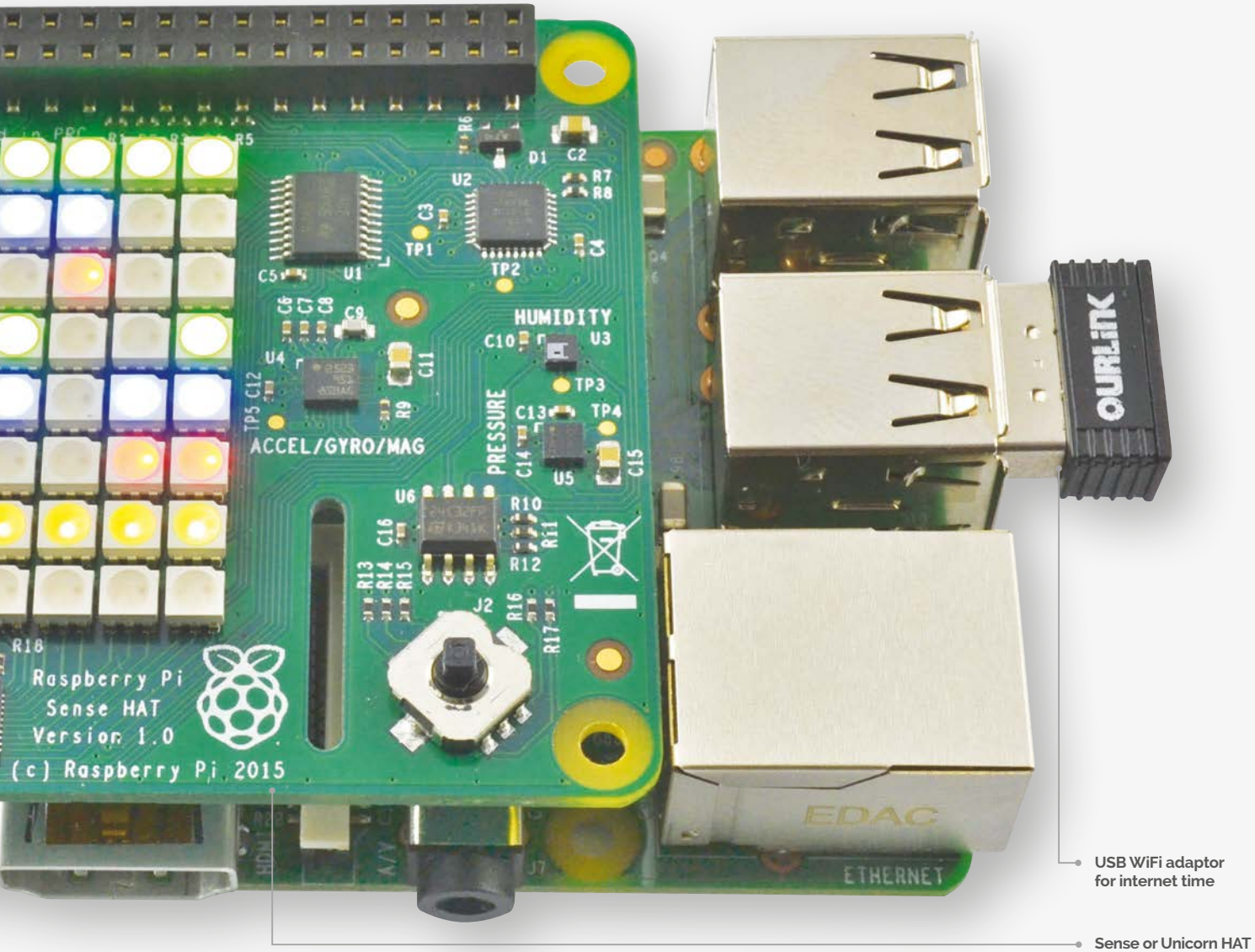
```
sudo apt-get update
sudo apt-get install sense-hat
sudo pip-3.2 install pillow
```

If you're using a Unicorn HAT, then run the following commands:

```
sudo apt-get install python-pip python-dev
sudo pip install unicornhat
```

There are two versions of the software for this project – one for the Sense HAT and one for the Pimoroni Unicorn HAT – so be sure to use the right version of the program for your hardware. The Sense





HAT version is called **binary_clock_sh.py**, and the Unicorn version **binary_clock_uni.py**.

You can download the programs for this project from your Raspberry Pi command line using the command:

```
git clone https://github.com/simonmonk/pi_magazine.git
```

This command will actually bring down the code for all the projects in Simon's *The MagPi* series, so if you've already issued this command for one of the earlier articles, change directory to **pi_magazine** and run the following command to update your directory with this project (**12_binary_clock**):

```
git pull
```

To run the program, change to the directory where the code for this project lives and then run the program using the commands below:

```
cd /home/pi/pi_magazine/12_binary_clock
sudo python binary_clock_sh.py
```

Or, if you are using the Unicorn HAT:

```
sudo python binary_clock_uni.py
```

binary_clock_uni.py

```
#!/usr/bin/env python

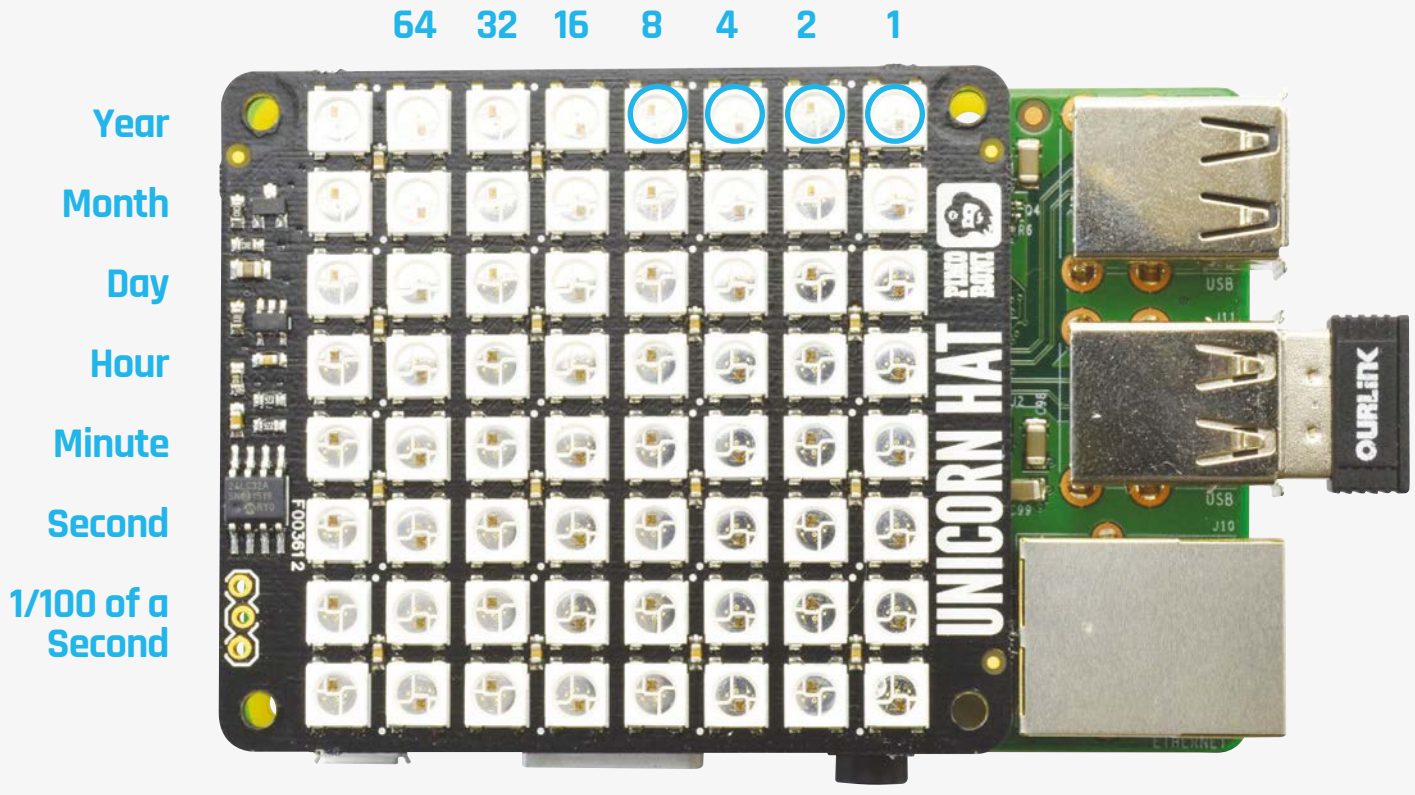
import unicornhat as hat
import time, datetime

year_color = (0, 255, 0)
month_color = (0, 0, 255)
day_color = (255, 0, 0)
hour_color = (0, 255, 0)
minute_color = (0, 0, 255)
second_color = (255, 0, 0)
hundrefths_color = (127, 127, 0)
off = (0, 0, 0)

hat.clear()
hat.brightness(0.5)

def display_binary(value, row, color):
        binary_str = "{0:8b}".format(value)
        for x in range(0, 8):
                if binary_str[x] == '1':
                        hat.set_pixel(x, row, color[0], color[1], color[2])
                else:
                        hat.set_pixel(x, row, 0, 0, 0)

while True:
    t = datetime.datetime.now()
    display_binary(t.year % 100, 0, year_color)
    display_binary(t.month, 1, month_color)
    display_binary(t.day, 2, day_color)
    display_binary(t.hour, 3, hour_color)
    display_binary(t.minute, 4, minute_color)
    display_binary(t.second, 5, second_color)
    display_binary(t.microsecond / 10000, 6, hundrefths_color)
    hat.show()
    time.sleep(0.0001)
```



binary_clock_sh.py

Language
>PYTHON

DOWNLOAD:
magpi.cc/1P1182l

```
#!/usr/bin/env python

from sense_hat import SenseHat
import time, datetime

hat = SenseHat()

year_color = (0, 255, 0)
month_color = (0, 0, 255)
day_color = (255, 0, 0)
hour_color = (0, 255, 0)
minute_color = (0, 0, 255)
second_color = (255, 0, 0)
hundrefths_color = (127, 127, 0)
off = (0, 0, 0)

hat.clear()

def display_binary(value, row, color):
        binary_str = "{0:8b}".format(value)
        for x in range(0, 8):
                if binary_str[x] == '1':
                        hat.set_pixel(x, row, color)
                else:
                        hat.set_pixel(x, row, off)

while True:
        t = datetime.datetime.now()
        display_binary(t.year % 100, 0, year_color)
        display_binary(t.month, 1, month_color)
        display_binary(t.day, 2, day_color)
        display_binary(t.hour, 3, hour_color)
        display_binary(t.minute, 4, minute_color)
        display_binary(t.second, 5, second_color)
        display_binary(t.microsecond / 10000, 6, hundrefths_color)
        time.sleep(0.0001)
```

How the code works

The two versions are actually very similar to each other. Both start by importing the libraries that they need for their displays. They then define a number of colour variables that you can change, if you like, in order to set different colours for each row of the clock.

Most of the action takes place in the **display_binary** function. This takes three parameters. The first is the value to be displayed as binary, the second is the row on which it should be displayed (0 to 7), and the final parameter is the colour to be displayed.

The value is converted into a binary string representation using String's **format** function. The **for** loop then iterates over each column (x value) and if that digit contains a 1, turns on the pixel in that row and column to the colour specified, otherwise the pixel is cleared.

The main loop fetches the current time and then splits it into its component parts, displaying each on a separate row. To display the year in two-digit format, we use modulo 100 (**t.year % 100**), which is the remainder after dividing it by 100.

Using your binary clock

Having a keyboard, mouse and monitor attached to your clock is fine while you are constructing it, but it would be better to have the program start automatically when the Raspberry Pi first boots up. To do this, run the following command to make the program executable:

```
sudo chmod +x binary_clock_*.py
```

Then, edit the file **/etc/rc.local** with the command:

```
sudo nano /etc/rc.local
```

Add the following line after the first block of comment lines that begin with #. Change the line to the **_uni** version if you are using a Unicorn HAT.

```
sudo /home/pi/pi_magazine/12_binary_
clock/binary_clock_sh.py &
```

Restart your Raspberry Pi and this time the program should start up automatically.

Note that you will still need an internet connection for the Pi to pick up the correct time.

NEXT MONTH

In the next project in this series, you will discover how to have a Raspberry Pi control a powerful linear actuator, capable of opening or closing a window or door.



Sonic Pi PART 7



SAM AARON

Sam is the creator of Sonic Pi. By day he's a Research Associate at the University of Cambridge Computer Laboratory; by night he writes code for people to dance to.
sonic-pi.net

SURFING RANDOM STREAMS

In this month's Sonic Pi guide, its creator, **Sam Aaron**, demonstrates the incredible power of randomisation in live-coded music...

You'll Need

- Raspberry Pi running Raspbian
- Sonic Pi v2.7+
- Speakers or headphones with a 3.5mm jack
- Update Sonic Pi: `sudo apt-get update && sudo apt-get install sonic-pi`

Back in episode four of the series, we looked at randomisation while coding some sizzling synth riffs. Since randomisation is an important part of my live-coding DJ sets, I thought it would be useful to cover the fundamentals in detail. So get your lucky hat on; let's surf some random streams!

There is no random

The first surprise when playing with the randomisation functions is that they're not random. What does this mean? Let's try some tests. Imagine a number in your head between 0 and 1, but don't tell me. Now let me guess... was it '0.321567'? No? I'm no good at this. Let me try again, but let's ask Sonic Pi to choose a number. Start Sonic Pi v2.7+ and ask it for a random number, but don't tell me:

```
print rand
```

Now for the reveal... was it '0.75006103515625'? Yes! I can see you're sceptical; it was a lucky guess. Let's try again. Press the Run button again and see what we get... The same again? This clearly can't be random! You're right, it's not.



What's going on? The fancy computer science term is determinism. This means that nothing is by chance and everything is destined to be. Sonic Pi is destined to always return '0.75006103515625' in the program above. It may sound useless, but in fact it's one of the most powerful parts of Sonic Pi. If you persevere, you'll learn how to rely on the deterministic nature of Sonic Pi's randomisation as a fundamental building block to your compositions and live-coded DJ sets.

A random tune

When Sonic Pi boots, it actually loads into memory a sequence of 441,000 pre-generated random values. When you call a random function such as **rand** or **rrand**, this random stream is used to generate your result. Each call to a random function uses a value from this stream, so the tenth call to a random function will use the tenth value from the stream. Also, every time you press Run, the stream is reset for that run. That's why I could predict the result to **rand** and why the 'random' tune was the same every time. Everybody's version of Sonic Pi uses the same random stream, which is important when we start sharing our pieces with each other.

Let's use this knowledge to generate a repeatable random tune:

```
8.times do
  play rrand_i(50, 95)
  sleep 0.125
end
```

Type this into a spare buffer and press Run. You'll hear a tune consisting of 'random' notes between **50** and **95**. When it's ended, press Run again to hear exactly the same tune again.

Handy randomisation functions

Sonic Pi comes with a number of useful functions for working with the random stream. Here are some of the most useful:

- **rand** Simply returns the next value in the random stream
- **rrand** Returns a random value within a range
- **rrand_i** Returns a random whole number within a range
- **one_in** Returns true or false with the given probability
- **dice** Imitates rolling a dice and returns a value between 1 and 6
- **choose** Chooses a random value from a list

Check out their documentation in the Help system for detailed information and examples.

Resetting the stream

While repeating a sequence of notes is essential for replaying a riff on the dance floor, it might not be the one you want. Wouldn't it be great if we could try a number of different riffs and choose the one we liked best? This is where the real magic starts.

We can manually set the stream with the function **use_random_seed**. In computer science, a random seed is the starting point for a new stream of random values. Let's try it:

```
use_random_seed 0
3.times do
  play rrand_i(50, 95)
  sleep 0.125
end
```

Great, we get the first three notes of our tune above: **84**, **83**, and **71**. We can now change the seed to something else, like this:

```
use_random_seed 1
3.times do
  play rrand_i(50, 95)
  sleep 0.125
end
```

We get **83**, **71**, and **61**. You may spot that the first two numbers are the same as the last two numbers before – no coincidence.

The random stream is just a giant list of 'pre-rolled' values. Using a random seed simply jumps us to a point in that list. Imagine a big deck of pre-shuffled cards. Using a random seed is cutting the deck at a particular point. The great part of this is that it's this ability to jump around the random list which gives us power when making music.

Let's revisit our random eight-note tune with this new power, and also add a live loop so we can experiment live while it's playing:

```
live_loop :random_riff do
  use_random_seed 0
  8.times do
    play rrand_i(50, 95), release: 0.1
    sleep 0.125
  end
end
```

While it's still playing, change the seed value from **0** to something else. Try **100**, or **999**. Try your own values and experiment – see which seed generates the riff you like best.

Bringing it all together

This tutorial has been quite a technical dive into the workings of Sonic Pi's randomisation functionality. Hopefully, it's explained how it works and how you can start using randomisation reliably to create repeatable patterns in your music. Crucially, you can use repeatable randomisation *anywhere*: the amplitude of notes, the timing of the rhythm, amount of reverb, current synth, the mix of an FX, etc. In the future we'll take a close look at these, but for now I'll end with a short example.

Type the code below into a spare buffer, press Run, then change the seeds. Press Run again while it's playing, and explore the different sounds, rhythms, and melodies you can make. When you find a nice one, note the seed number so you can return to it. Once you've found a few seeds you like, put on a live-coded performance by simply switching between your favourite seeds to make a full piece.

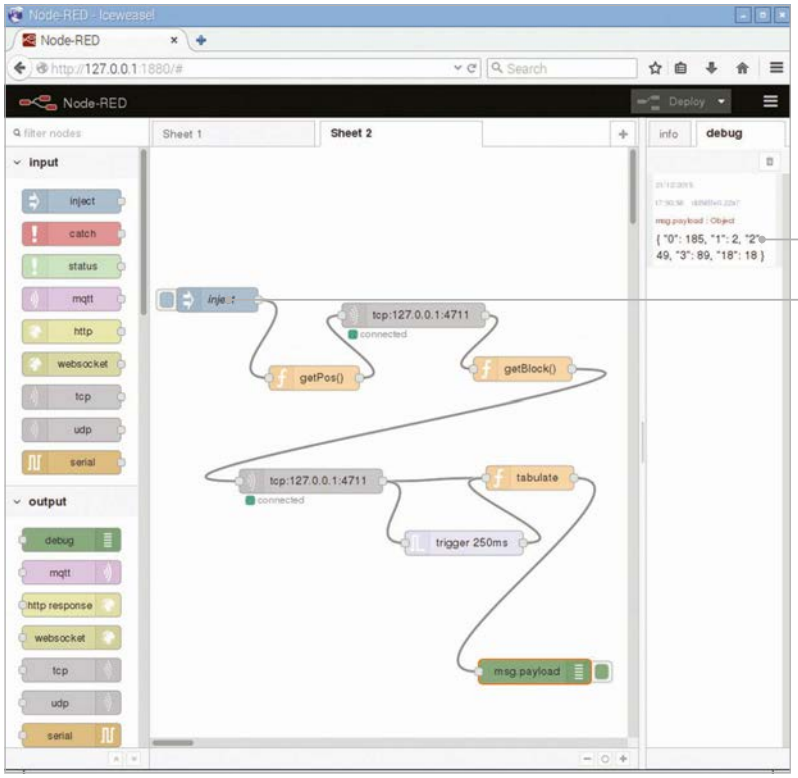
```
live_loop :random_riff do
  use_random_seed 10300
  use_synth :prophet
  s = [0.125, 0.25, 0.5].choose
  8.times do
    r = [0.125, 0.25, 1, 2].choose
    n = (scale :e3, :minor).choose
    co = rrand(30, 100)
    play n, release: r, cutoff: co
    sleep s
  end
end

live_loop :drums do
  use_random_seed 2001
  16.times do
    r = rrand(0.5, 10)
    sample :drum_bass_hard, rate: r, amp: rand
    sleep 0.125
  end
end
```



BORIS ADRYAN

Boris doesn't enjoy building in Minecraft as much as he likes doing data science with block inventories.
iot.ghost.io / @BorisAdryan



Below The 'getBlock()' function. Lines 8-16 populate an array with 343 (73) elements, each carrying a payload with a world.getBlock statement for the TCP node



PART 2 NODE-RED AND CONTROLLING MINECRAFT WITH JAVASCRIPT

With Node-RED you can mine like a pro. Give Steve an additional sense to see if there are any exciting blocks in the neighbourhood

You'll Need

- Raspbian **raspberrypi.org/downloads**
- Node-RED – pre-installed on Raspbian **nodered.org**
- Iceweasel – installed via terminal **wiki.debian.org/Iceweasel**

REAL-WORLD INTERACTION

Look at issue 41's cover feature for inspiration on how to get *Minecraft* working with the real world.

Part one of this mini-series introduced linear flows with Node-RED and *Minecraft*, and you should go through it before embarking on the next adventure. In part two, you'll dive deeper into Node-RED and JavaScript. Learn how to react to events, loop over a set of variables, and retain values from previous executions of a node in your flow. With this tutorial, we've got you well covered for any further explorations with Node-RED!

This tutorial takes off where we left in part one, which can be found in *The MagPi* issue 41, on page 20. In that tutorial, we taught you how to make a *Minecraft* 'Hello World' and how to read data from it.

Preparation

To get back to where you were, start *Minecraft* and open a world. Fire up Node-RED from the *Programming* sub-menu. Open a web browser (Iceweasel works best with Node-RED) and direct it to **127.0.0.1:1880** to see the programming environment. Arrange the windows so you can see what's going on in Node-RED and in the *Minecraft* world.

You should see your last flow, the one that retrieved Steve's position and printed the coordinates in the debug panel. If not, you can download the Player Position flow from the Node-RED flows directory (**magpi.cc/1Qr4Xht**) and import it before you proceed.

Calculate and store the coordinates

Edit the **conversion** function node, and rename it to **getBlock()** to reflect its new role. There's a bit of magic in this one. The `replace()` and `split()` functions remove the trailing line break from the *Minecraft* message and break the components of Steve's position into character strings. These are put into an array – a variable that saves a few values in a list – which is called `arr`. Convert the strings from the list into whole numbers and assign them to variables `x`, `z`, and `y` with `parseInt()` to be able to do calculations on them.

Define a variable `cube_size` that keeps half of the size of a cube in whose centre we're located. With three nested `for` loops, iterate over this cube that spans from *x* - *cube_size*, *z* - *cube_size*, *y* - *cube_size* to *x* + *cube_size*, *z* + *cube_size*, *y* + *cube_size*. For each tuple (`h`, `k`, `l`), create a `payload` object that carries a string of the format `"world.getBlock(h,k,l)\n"`, and store it as an element of an array, `coord_set`. The `function` node returns `[coord_set]`, which

makes Node-RED invoke the next flow element for each list element.

Drag and drop another **TCP request** node from the function section of the node panel. As before, the *Server* is 127.0.0.1 and *port* 4711, and we expect the returned message to be finished with **\n**.

Drag and drop a **trigger** node. Configure it such that it sends '*nothing*' and '*waits for*' 250ms, and then sends '*the string payload*' **"trigger".** Check the '*extend delay*' option. This essentially means that as long as there are incoming messages at least every 250ms, the node will remain silent, but if there are no new messages, we send the trigger message. In our case, this will indicate that the TCP request node is done with all requests.

A final function node called **tabulate** does a few clever bits. First, it checks if the incoming payload is **"trigger"**. If that's the case, it publishes the results of a variable **`context.table`** as payload, but deletes **`context.table`**. Remember, that's only happening after all TCP requests are finished. If TCP requests are still incoming, the second part after the **`else`** statement is relevant: the request buffer with a *Minecraft* block ID is converted into a string. Have a look at **magpi.cc/1PvHAh4** to see what they mean; for example, 0 is air and 17 is wood. If it's the first time we're entering this part of the code, we'll create an empty variable **`context.table`**: *context* is a special variable in Node-RED that is persistent between different times a node is called up. If there's no table entry for ID, we'll create one and set it to 0, otherwise we take its current value; then we add 1. With that, the node expects more incoming IDs until the trigger message arrives.

Ultimately, we'll have a directory of elements which we can print with a debug node; this is happening in the **`return`** statement in the **tabulate** node. The block type IDs are shown in quotes; the number of occurrences follows the colon. These should add up to 343, the volume of the cube around us.

We hope that this tutorial has helped you understand a few concepts that are not immediately obvious when using Node-RED for the first time. In part one, we promised interaction with the real world... well... just add the GPIO node. We're sure you'll figure it out!



Above The 'tabulate' function. Lines 1-4 are executed if the trigger node doesn't receive any more results from the TCP node.

Language
>NODE-RED/ JAVASCRIPT

DOWNLOAD:
Player Position
magpi.cc/1Qr4Xht

Block Inventory
magpi.cc/1n59yKr

Player Position


```
[{"id":"945e5f77.182da8","type":"function",
"z":"a59a50d2.5a65b","name":"function","fun
c":"msg.payload = \"player.getPos()\\n\";\
nreturn msg;","outputs":1,"noerr":0,"x":128,"
y":153,"wires":[["ec2d4ccc.7365b","c5a42bac.
ef9ba"]]},{"id":"c5a42bac.ef9ba","type":"debug"
,"z":"a59a50d2.5a65b","name":"","active":true,"
console":"false","complete":"false","x":378,"y":54,"wires":[]},{"i
d":"f1098e71.8eb9d","type":"inject","z":"a59a50d2.5a65b","name":"i
nject","topic":"","payload":"Hello World","payloadType":"string","
repeat":"","crontab":"","once":false,"x":90,"y":54,"wires":[["945e
5f77.182da8"]]},{"id":"ec2d4ccc.7365b","type":"tcp request","z":"a
59a50d2.5a65b","server":"127.0.0.1","port":"4711","out":"char","sp
litc":"\\n","name":"","x":244,"y":253,"wires":[["f4f5b8bd.154d68"]-
]},{"id":"f4f5b8bd.154d68","type":"function","z":"a59a50d2.5a65b",
"name":"conversion","func":"msg.payload = msg.payload.toString();\
nreturn msg;","outputs":1,"noerr":0,"x":367,"y":154,"wires":[["c5a4
2bac.ef9ba"]]}]
```


Block Inventory

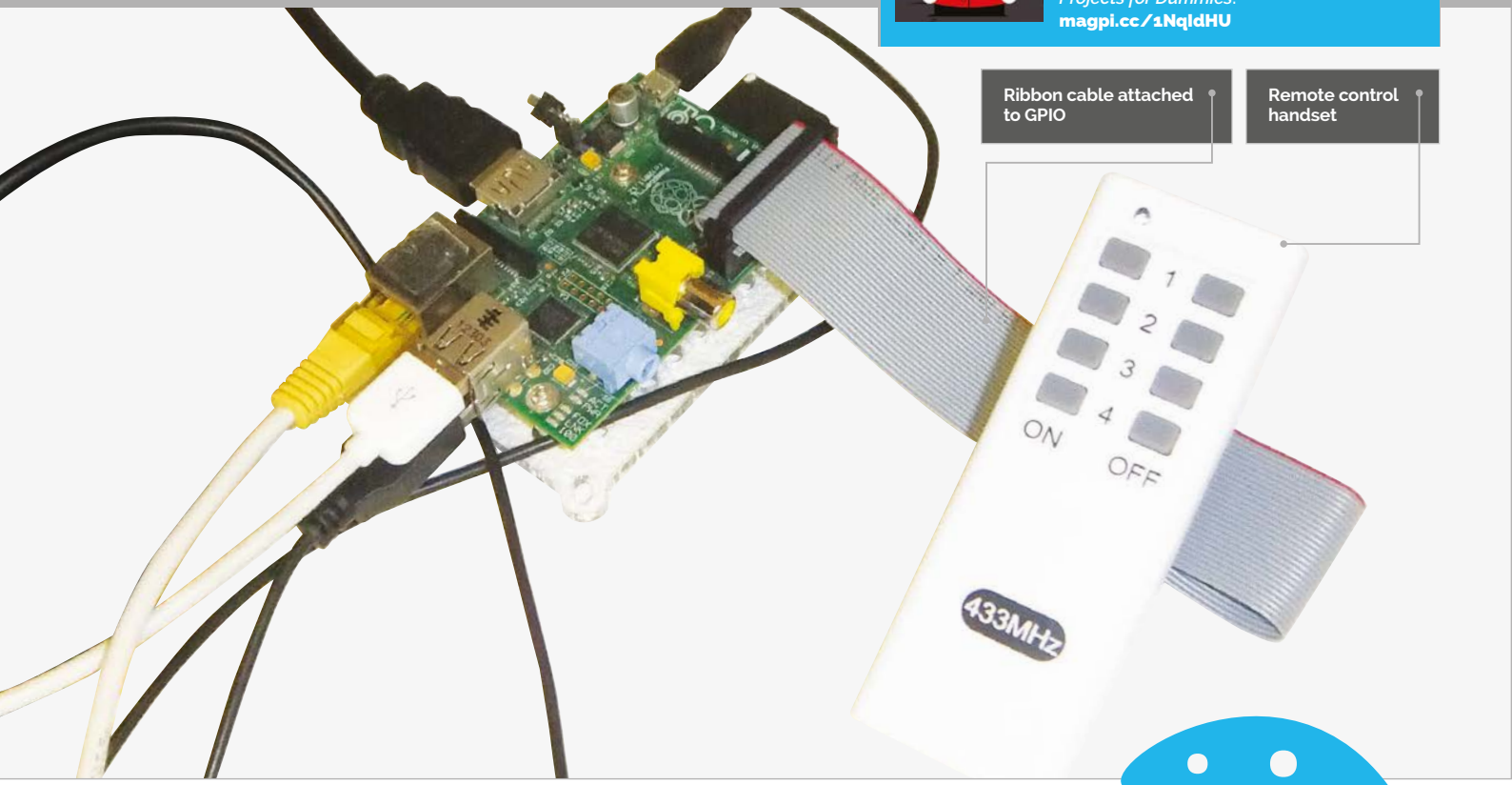

```
[{"id":"533edf4.facc12","type":"function","z":"e13b1b02.1ec4e8","name":
"getPos()","func":"msg.payload = \"player.getPos()\\n\";\nreturn msg;"
,"outputs":1,"noerr":0,"x":207,"y":151,"wires":[["71429f10.8ebd6"]]},{"
id":"dd58ffe0.22a7","type":"debug","z":"e13b1b02.1ec4e8","name":"","act
ive":true,"console":"false","complete":"payload","x":642,"y":278,"wire
s":[]},{"id":"cc9ed7e5.336128","type":"inject","z":"e13b1b02.1ec4e8","
name":"inject","topic":"","payload":"trigger","payloadType":"string","
repeat":"","crontab":"","once":false,"x":98,"y":53,"wires":[["533edf4.
facc12"]]},{"id":"71429f10.8ebd6","type":"tcp request","z":"e13b1b02.1
ec4e8","server":"127.0.0.1","port":"4711","out":"char","splitc":"\\n"
,"name":"","x":329,"y":74,"wires":[["a9afe2c4.56502"]]},{"id":"a9afe2
c4.56502","type":"function","z":"e13b1b02.1ec4e8","name":"getBlock()"
,"func":"var arr = msg.payload.toString().replace(/(\\n|\\r)+$/, '').
split(\",\");\n\nvar x = parseInt(arr[0]);\nvar z = parseInt(arr[1]);\
nvar y = parseInt(arr[2])\nvar cube_size = 3;\n\nvar coord_set =
[]\n  for (var h = x-cube_size; h < x+cube_size+1; h++) {\n      for
(var k = z-cube_size; k < z+cube_size+1; k++) {\n          for (var
l = y-cube_size; l < y+cube_size+1; l++) {\n              coord_set.
push({payload: \"world.getBlock(\"+h+\",\"+k+\",\"+l+\")\\n\"});\n
}\n      }\n  }\nreturn [coord_set];","outputs":1,"noerr":0,"x":463,
"y":140,"wires":[["9713fdf5.68ec"]]},{"id":"9713fdf5.68ec","type":"t
cp request","z":"e13b1b02.1ec4e8","server":"127.0.0.1","port":"4711",
"out":"char","splitc":"\\n","name":"","x":203,"y":277,"wires":[["4958
e94e.b6a718","e8e39686.171c68"]]},{"id":"4958e94e.b6a718","type":"func
tion","z":"e13b1b02.1ec4e8","name":"tabulate","func":"if (msg.payload
=== \"trigger\") {\n  msg.payload = context.table;\n  context.table =
undefined;\n  return msg;\n} else {\n  var ID = msg.payload.toString().
replace(/(\\n|\\r)+$/, '');\n  \n  context.table = context.table ||
{};\n  context.table[ID] = context.table[ID] || 0;\n  context.table[ID]
= context.table[ID]+1;\n}","outputs":1,"noerr":0,"x":470,"y":278,"wires
":[["dd58ffe0.22a7"]]},{"id":"e8e39686.171c68","type":"trigger","z":"e1
3b1b02.1ec4e8","op1":"1","op2":"trigger","op1type":"nul","op2type":"val
","duration":"250","extend":true,"units":"ms","name":"","x":362,"y":354
,"wires":[["4958e94e.b6a718"]]}]
```

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/1NqIdHU

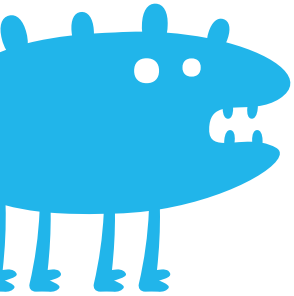


You'll Need

- Maplin remote-controlled mains sockets 3 Pack (Code: N79KA) **magpi.cc/1Qkzahn**
- 12-way dual-row surface-mount header socket
- 10× 3K3 resistors
- 10× General-purpose NPN transistors
- Raspberry GPIO connection lead

HUNT THE SNARK





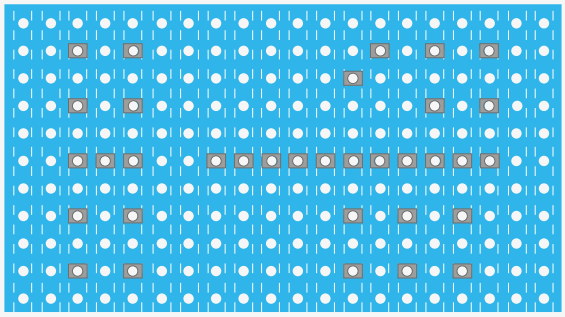
Get control of the mains to allow you to 'Hunt the Snark'

Remember the children's game 'hunt the thimble'? Well, in Hunt the Snark the object of the hunt, like the Snark itself, is not actually defined. The form the Snark takes on can be anything, providing it's a mains-powered device; it could be anything from lamps to air fresheners. The idea is that the children can rush about the house, excitedly cataloguing which Snarks are on and which are off, and the first to report back to you is the winner. Plan it right and you could have up to two minutes' rest between rounds. Of course, this requires you to have remote access to mains-controlled devices. There have been a few projects that allow the Raspberry Pi to control mains devices, but none of them were implemented the way we'd do it here at the Bakery. So this is our take on adapting a commercial product to the Raspberry Pi. Previous attempts at similar projects have involved buying a new transmitter and coercing it into producing the correct codes. What we've done instead is to just get the Raspberry Pi to press the keys on a remote control.

The project

The remote-control mains socket we used was quite a generic 'Made in China' device, so we imagine it's available in other places as well. Even so, the principles used here should be applicable to any number of remote controls. We wanted to have a remote control that could not only be attached to the Raspberry Pi, but also still be capable of operating as a standalone remote controller. The key to that

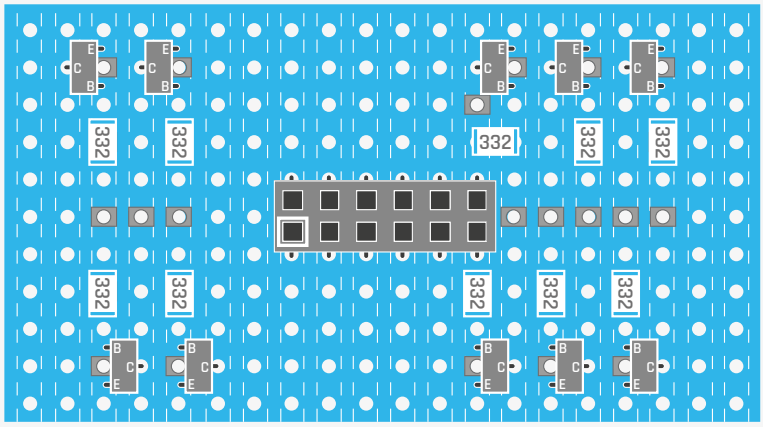
MODIFYING THE REMOTE



>STEP-01

Prepare the stripboard

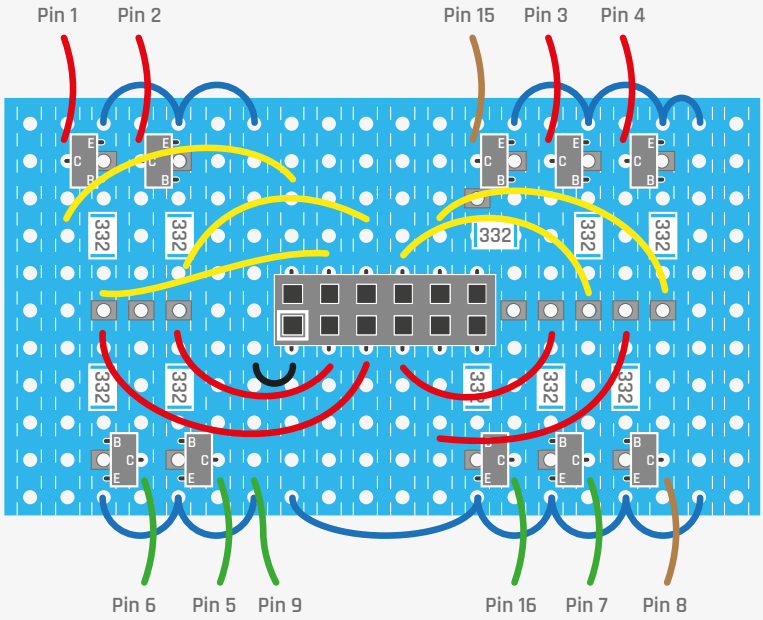
Cut a piece of stripboard, 20 columns of 11 hole tracks, and break the track at the places shown. Do this with a scalpel so that the minimum amount of copper on each side of the hole is removed. Then, using solder braid, remove the four-position slider switch from the remote's circuit board.



>STEP-02

Mount the components

Solder the 3K3 resistors across the breaks in the tracks and add the transistors. The pinout for the transistors shown is the most common one; check that the transistors you're using match this. Finally, solder the double row pin-header socket between the track breaks.



>STEP-03

Wire it into the remote

Add the wires to the board. There are two sorts: those that connect between points on the board, and those that go from the board to the pins of the encoder chip on the remote's main board. Use thin wires for this; we used 28 AWG solid wire for the point-to-point ones, and very thin multi-stranded flexible wires from board to chip. Mark the lower-left socket with a white marker pen; this is the ground connection.

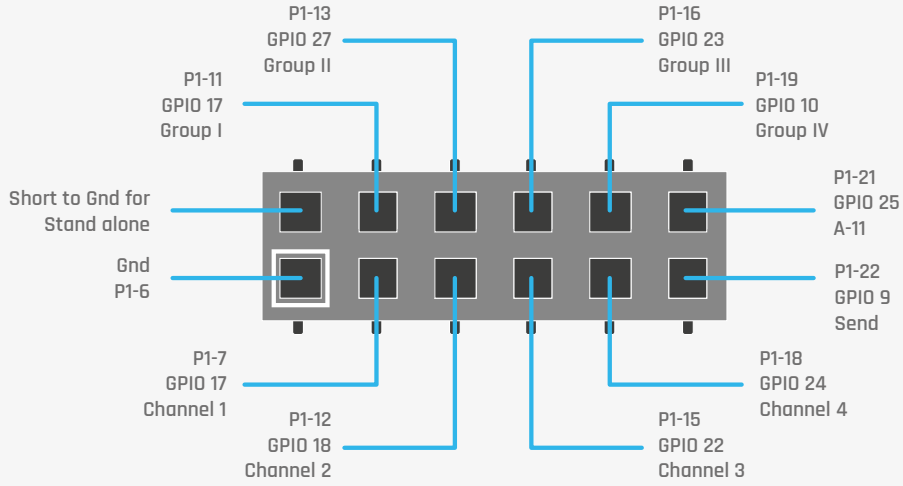
was to understand what was happening in the circuit, so it was out with the screwdrivers to prise the two halves of the case apart. This was the hardest part of the project and the result is shown in **Fig 1** (overleaf). Inside was a single-chip encoder SC5262, and a crystal oscillator which acts as the 433.92MHz transmitter. Remote controllers have a small grid of tracks covered in dark material, and carbon pills on a membrane are pressed down to short these tracks together. This remote, however, was a little unconventional, in that there were three sets of wires that were all shorted together when a button was pushed. We traced the wiring and, with reference to the encoder's data sheet, worked out what was happening.

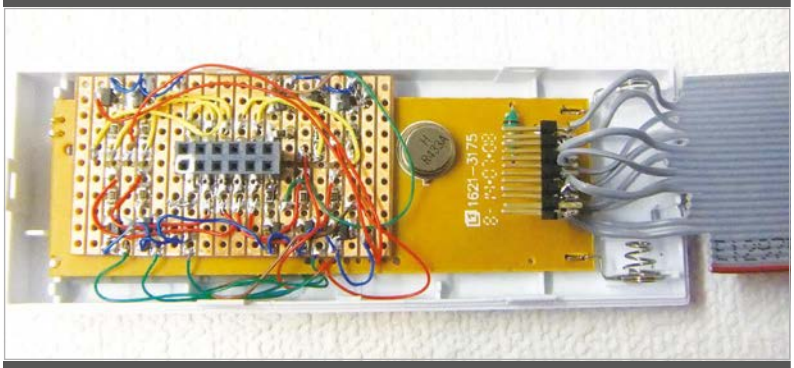
The design

Basically, for each channel there's an on and off button. Pressing either shorts one of four data lines (one for each channel) to ground and at the same time shorts out the transmit-enable line or another data line, depending on whether it's the on or off column of the channel. The key to understanding the operation was to spot that there's a diode that also pulls the transmit-enable line down when that data line is pulled down. While the remote runs off a very small 12V battery, the only signals changed by the key presses are to either let a pin float, or connect it to ground. This makes the interface circuit very simple indeed. All that's needed is an open collector

>STEP-04
Make the Pi GPIO lead

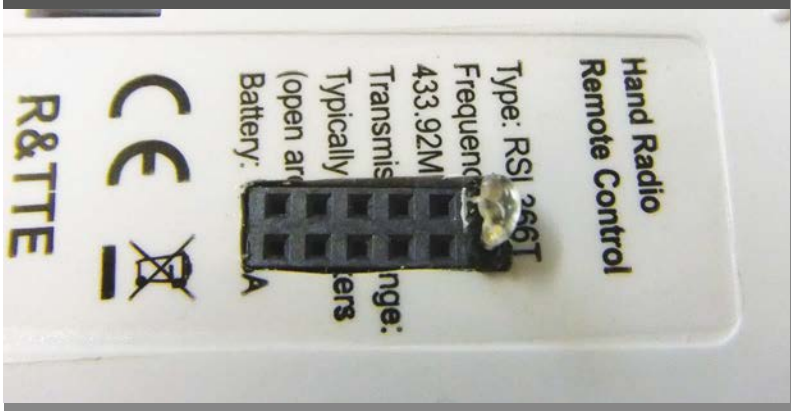
Make a lead from the remote's socket to the Raspberry Pi's GPIO pins. This can be done in many ways; we used an IDC socket for a 26-pin GPIO Pi and wired up the other end to a 12-way two-row pin header. Note: the picture shows the GPIO number along with the pin number of the GPIO header, as well as the wire's function.





>STEP-05
Finished wiring

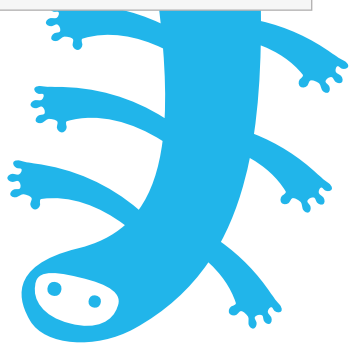
You can see the wiring and the remote end of the Pi lead. After testing, we smothered this end in hot-melt glue to act as insulation and strain relief. The interface board should be attached to the remote's circuit board with double-sided sticky tape. You'll need to snip about 1.5mm off the four plastic moulded pillars in the back shell to compensate for the extra thickness of the stripboard.



>STEP-06
Cut a slot in the remote's back shell

Finally, cut a slot in the remote's back shell to align with the pin-header socket; this will sit flush with the back shell. We used a combination of a drill and scalpel; the plastic is thin enough so that it can be cut. Cut a two-pin strip and wire the pins together, put a dab of hot-melt glue over the joint, and insert it in the white outlined connector. This allows the remote to operate as a standalone Group I controller.





transistor pulling the pin to ground or letting it float; any generic NPN transistor will do, but we used the 2N2222 type because these were to hand – you can use much cheaper ones if you wish. In addition to these push buttons, there's a slider switch on the back which selects one of four possible groups, denoted by Roman numerals.

> "The software is a simple enough Pygame program"

We decided to make the remote programmable for all possible channels, so that gives you a maximum of 16 controlled mains outlets, and the schematic is shown in **Fig 2**. In order for it to fit neatly inside the existing remote case we used surface-mounting components, but there's no absolute need for this and you can make it with conventional leaded components if you wish. See the 'Modifying the remote' step-by-step guide for full details.

The software

The software is a simple enough Pygame program. Each controllable plug is represented by a photograph of the plug, with the channel and group settings from the back of the plug displayed on the front. The LED lights up red when the plug is on, and is black when it's off. Also, each plug can be given a name so you can identify it more easily. Clicking on a plug will toggle its state. And for Snark hunting, pressing the **H** key will change the state of a random number of plugs.

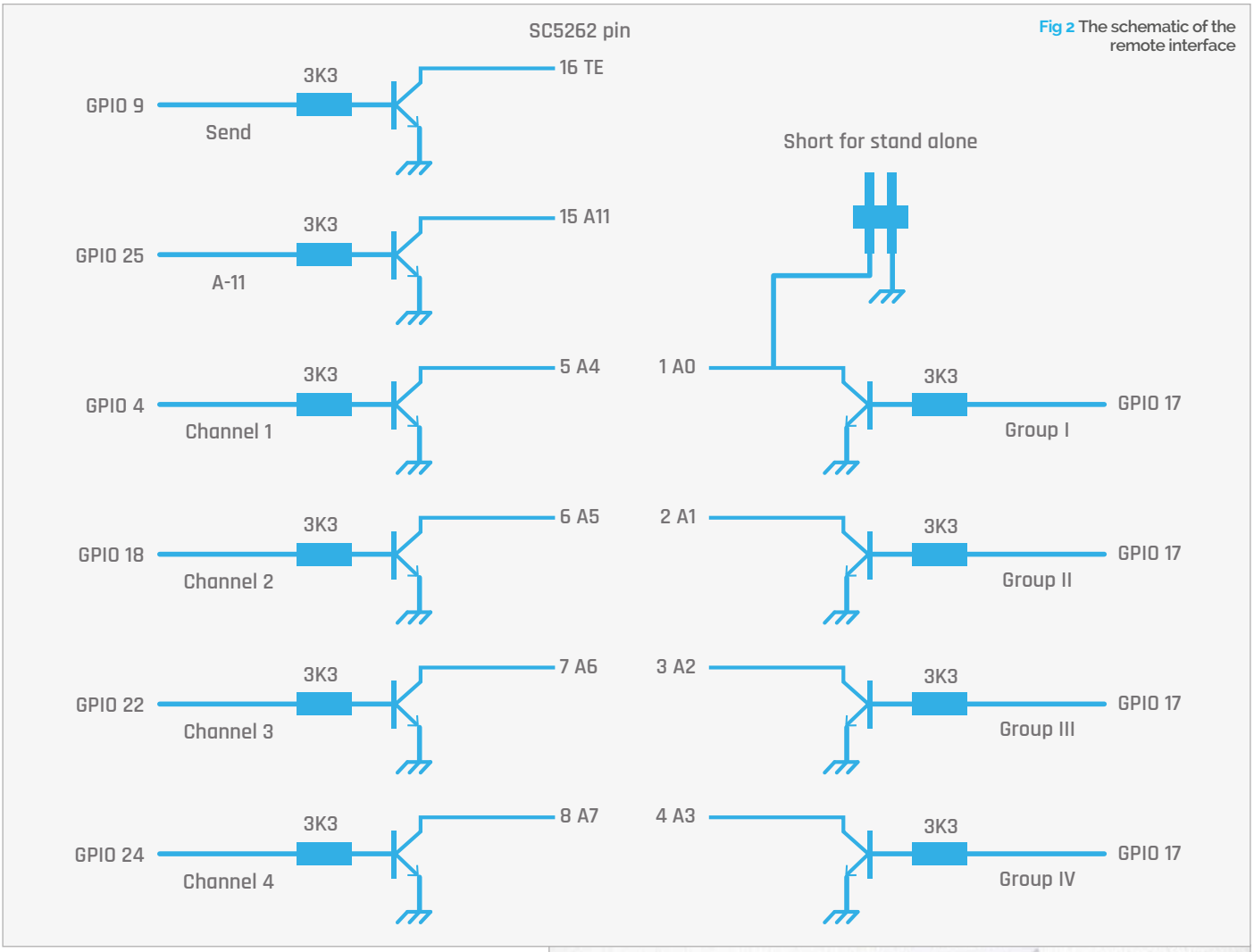


Fig 2 The schematic of the remote interface

The code

The code (**Mains.py**) overleaf is configured for the full possible set of 16 remote plugs. There's a version on GitHub that covers just the basic four plugs if you need it, along with the image files required. The **`plugName`** list can be altered to suit your installation. When you close the program window, all plugs are set to off – you can miss this bit out of the **`terminate`** function if this is not how you want it to work. It takes just 0.8 seconds to change the state of a plug; when you have 16 of them, however, this is nearly 13 seconds and it can seem that the code has hung. This is why lists are used, so that you only change the state of plugs that need changing. The **`setPlug`** function is the key to controlling the keys on the remote handset and can be taken into your own programs. First, the appropriate group and channel pins are shorted to ground, by writing a logic 1 to the transistor. Then, depending on whether it's an on or off command for that plug, the lines are set and a delay entered. The pins are released and another delay ensures the encoder stops transmitting before releasing the control and group pins. Making the delays shorter reduced the reliability of the plug switching, in our experience.

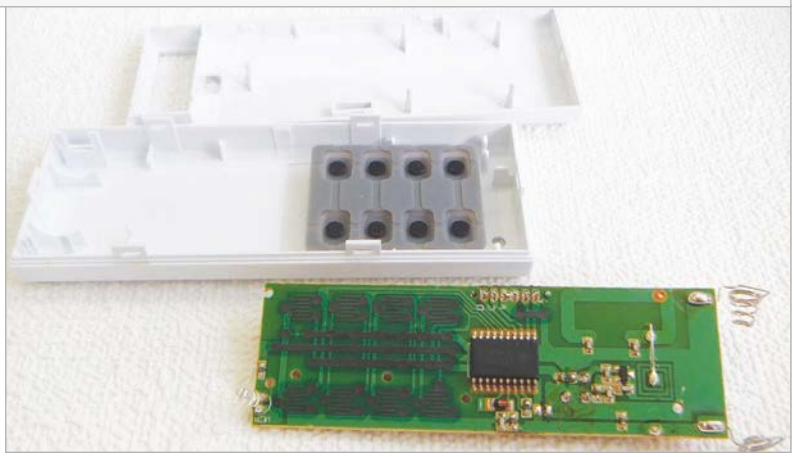
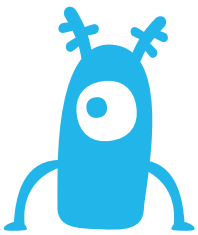
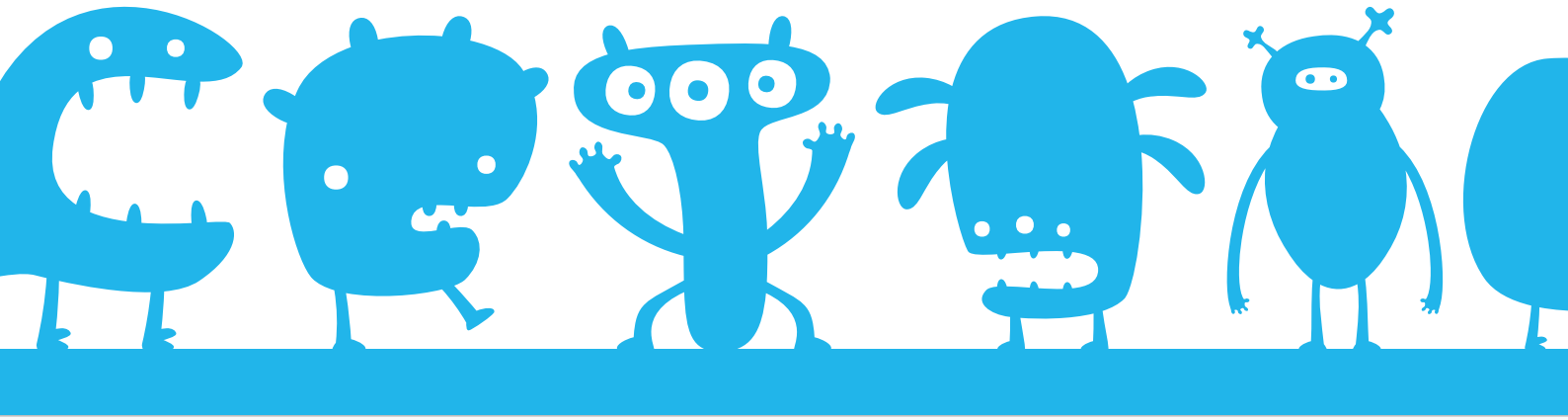


Fig 1 The remote's internal circuit board

Taking it further

The Snark hunting can have several rules and variations, from just finding the one device that's on to having to remember what devices were on, or even solving a puzzle with the on devices as a clue. The basic control of mains can be used in many ways, such as switching the lights on and off in a realistic fashion in an unattended house – in this way, any potential burglars will know that you have a computer setup really worth stealing.





Mains.py

```
# Mains control - Hunt the Snark
# By Mike Cook - January 2016


import pygame, time, os, random
import wiringpi2 as io
from copy import deepcopy

pygame.init()        # initialise graphics interface
os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
pygame.display.set_caption("Control Central")
pygame.event.set_allowed(None)
pygame.event.set_allowed([pygame.KEYDOWN,pygame.
QUIT,pygame.MOUSEBUTTONDOWN])
screen = pygame.display.set_mode([1050,480],0,32)
textHeight = 22
font = pygame.font.Font(None, textHeight)
random.seed()

try :
   io.wiringPiSetupGpio()

except :
   print"start IDLE with 'gksudo idle' from command line"
   os._exit(1)

plugState = [ False for n in range(0,16)]
groupPins = [17,17,17,17, 27,27,27,27, 23,23,23,23,
10,10,10,10]
channelPins = [4, 18, 22, 24, 4, 18, 22, 24, 4, 18, 22,
24, 4, 18, 22, 24]
A_11Pin = 25
sendPin = 9
plugName = ["Lounge1","Lounge2","Dining
Room","Kitchen","Bedroom 1","Bedroom 2",
"Bedroom 3","Hall 1","Landing","Garage","Porch","Attic",
"TV Lamp","Kitchen Lamp","Hall 2","Computer Room"]

plugPicture = pygame.image.load(
"images/plug.png").convert_alpha()
scPicture = [pygame.image.load(
"images/sc"+str(n)+".png").convert_alpha()
for n in range(1,5)]
sgPicture = [pygame.image.load(
"images/sg"+str(n)+".png").convert_alpha()
for n in range(1,5)]

def main():
   print"Mains Controller - Hunt the Snark"
   print"Press h - to hunt"

   initGPIO()
   showPicture()
   while True:
     checkForEvent()
     time.sleep(0.2)

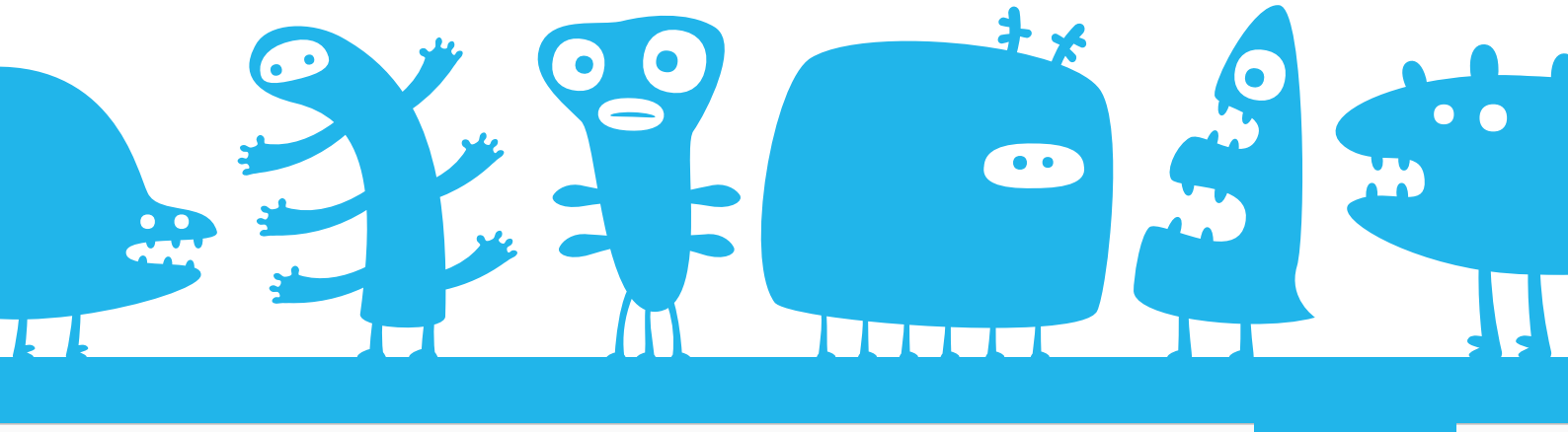
def togglePlug(click):
   global plugState
   plug = (click[0] / 130) + (8 *(click[1] /240))
   plugState[plug] = not(plugState[plug])
   setPlug(plug,plugState[plug])
   showPicture()

def setPlug(p,on):
   io.digitalWrite(groupPins[p],1)
   io.digitalWrite(channelPins[p],1)
   if on :
      io.digitalWrite(A_11Pin,0)
      io.digitalWrite(sendPin,1)
      time.sleep(0.32)
      io.digitalWrite(sendPin,0)

   else :
      io.digitalWrite(A_11Pin,1)

      time.sleep(0.32)
      io.digitalWrite(A_11Pin,0)

   time.sleep(0.2)
   io.digitalWrite(channelPins[p],0)
   io.digitalWrite(groupPins[p],0)
```



Language
>PYTHON 2.7

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

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

```
def huntSet():
   global plugState
   print"Setting the Snark"
   changeState = deepcopy(plugState)
   changed = False
   maxiumChanges = 8 # alter for more potential changes
   while not(changed):
      for n in range(0,random.randint(1,maxiumChanges)):
        p=random.randint(0,15)
        changeState[p] = not(changeState[p])
      for n in range(0,16):
         if changeState[n] != plugState[n] :
            plugState[n] = changeState[n]
            setPlug(n,plugState[n])
            changed = True
            showPicture()
   print"Ready for the hunt"

def initGPIO():
   for pin in range (0,16):
      io.pinMode(groupPins[pin],1)
      io.digitalWrite(groupPins[pin],0)
      io.pinMode(channelPins[pin],1)
      io.digitalWrite(channelPins[pin],0)
   io.pinMode(A_11Pin,1)
   io.digitalWrite(A_11Pin,0)
   io.pinMode(sendPin,1)
   io.digitalWrite(sendPin,0)

def showPicture():
   for row in range(0,2):
      for plug in range(0,8):
         screen.blit(
scPicture[plug % 4],[(plug * 130)+35,(row * 240)+9])
         screen.blit(sgPicture[((plug / 4)+ row*2)],[(
plug * 130)+41,(row * 240)+152])
         screen.blit(plugPicture,[plug * 130,row * 240])
   for plug in range(0,16):
      x = (plug % 8)*130
      y = (plug / 8)*240
      drawWords(plugName[plug],x+15,y+97)
      if plugState[plug]:
         pygame.draw.circle(
screen,[180,0,0],[x+66,y+80],6,0)
   pygame.display.update()
   time.sleep(0.4)

def drawWords(words,x,y) :
        textSurface = pygame.Surface(
(2+len(words)*12,textHeight))
        textRect = textSurface.get_rect()
        textRect.left = x
        textRect.top = y
        textSurface = font.render(
words, True, (19,104,229), (200,205,208))
        screen.blit(textSurface, textRect)

def terminate(): # close down the program
    print ("Closing down, please wait")
    for plug in range(0,16): #Turn plugs off-delete if needed
       if plugState[plug] == True :
          setPlug(plug,False)
    pygame.quit() # close Pygame
    os._exit(1)

def checkForEvent(): # see if we need to quit
    global visited, duplicate
    event = pygame.event.poll()
    if event.type == pygame.MOUSEBUTTONDOWN:
       point = pygame.mouse.get_pos()
       togglePlug(point)
       # print point
       # print out position of click for development

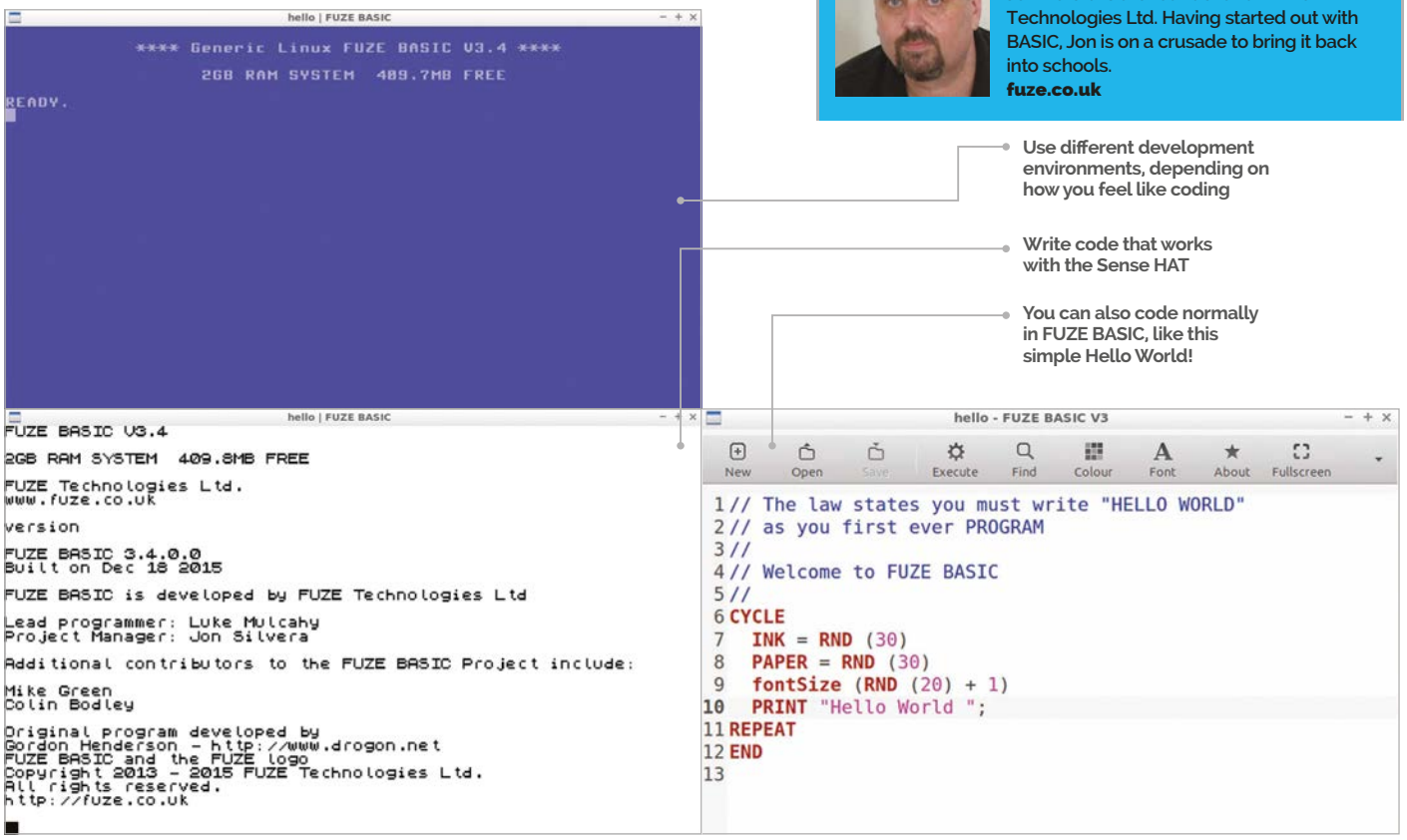
    if event.type == pygame.QUIT :
        terminate()
    if event.type == pygame.KEYDOWN :
       if event.key == pygame.K_ESCAPE :
          terminate()
       if event.key == pygame.K_h : # hunt the Snark
          huntSet()

# Main program logic:
if __name__ == '__main__':
    main()
```



JON SILVERA

Jon Silvera is the founder and CEO of FUZE Technologies Ltd. Having started out with BASIC, Jon is on a crusade to bring it back into schools.
fuze.co.uk



USE FUZE BASIC WITH THE SENSE HAT

You'll Need

- Raspbian **raspberrypi.org/downloads**
- FUZE Basic **fuze.co.uk/getfuzebasic**
- Sense HAT **magpi.cc/SenseHAT**

FUZE brings back the ways of the classic BASIC language, and now you can use it with the Sense HAT

Time for a new look at FUZE BASIC. There have been many significant improvements in FUZE BASIC recently. Along with many 'behind the scenes' adjustments, the most noticeable change is the new editor. Although enjoyed by many, the older FB editor was just too archaic. The lack of any real copy and paste functionality was too limiting. Now with Find (and Replace), text size, themes, and a better integrated Help system, along with many new commands and a significant speed boost, it's time to give FUZE BASIC a try. With its forgiving syntax, common-sense language, and powerful command set, you'll find no easier introduction to programming.

By way of an introduction, we thought we'd show off the new Raspberry Pi Sense HAT support, but before we jump in at the deep end, we'll have a little 'Hello World' warm-up exercise. Ready? Now flex those fingers...

Basic FUZE

Have you got the latest version of FUZE BASIC installed? It's now on to v3.4. If not, then hop on to **fuze.co.uk** and go straight to the 'Get FUZE BASIC' page. A few instructions and a few minutes installing will have you up and running in no time.

You'll now have a new item in your Raspbian main Menu under Programming; you can right-click this to add it to your Desktop if you prefer.

The first thing you'll see is the FUZE BASIC Direct mode. Some of you oldies will recognise the familiar Commodore 64 look and feel. Of course, this will have some of you squirming in your seat, reeling at the fact you're being forced into enemy lines. Worry not, as a simple tap of **SHIFT+TAB** will present you with a series of retro desktops.

In Direct mode, you can instruct FUZE BASIC to do many things. In fact, most program commands will

work in DM. For example, try typing this:

```
Apples = 10
Oranges = 10
Fruit = Apples + Oranges
Print Fruit
```

This will, unless an error occurred, result in '20' being displayed. OK, let's get on with the task at hand. Press **F2** to enter the editor and type in the following program:

```
LOOP
  INK = RND (30)
  PAPER = RND (30)
  fontSize (RND (20) + 1)
  PRINT "Hello World ";
REPEAT
END
```

Once you've finished, press **F3** to RUN the code (or click the 'Execute' icon). The first time you RUN it, you will be asked to save it.

Now, let's have a play with the Sense HAT. If you haven't already connected the Sense HAT to your Pi, turn it off and do so: you need to boot the Pi with the HAT attached, otherwise it's not likely to be recognised. So, reboot if you need to and get back to the FUZE BASIC Direct mode. If you're in the Editor, click the NEW icon and then **F2** to return to Direct. If you're in Direct mode, type in NEW and press **ENTER** to clear the memory.

The 'official' coordinates of the Sense HAT matrix are set as though the Pi is vertical, not horizontal. FUZE BASIC remaps the coordinates so that when it's attached to a FUZE computer case it makes more sense. Therefore, X is along the bottom and in line with the Raspberry Pi text, and Y starts down by the (FC) logo.

In Direct mode, try this:

```
sensePlot (0,0)
```

This will switch on the bottom-left LED. You'll soon get the hang of it if you try a few more **sensePlot** commands. To clear the matrix display, use the following command:

```
senseCLS
```

Now try:

```
senseColour = raspberry
```

…and then plot a few more LEDs:

```
SensePlot (2,3)
```

There are around 30 named colours in FUZE BASIC, so go wild. Or, if you like things a bit more flexible, you can use:



Above A very colourful rendition of Hello World, as rendered by FUZE Basic

```
senseRGBcolour (R, G, B)
```

… where R, G, and B are values between 0 and 255. With them set to '0, 0, 0', this is the equivalent of black, or in this case 0 light. With '255, 0, 0' it is pure red, while '255, 0, 128' is a nice dark pink. Type in:

```
Print senseTemperature
```

…and

```
Print sensePressure
```

…and

```
Print senseGyroX
```

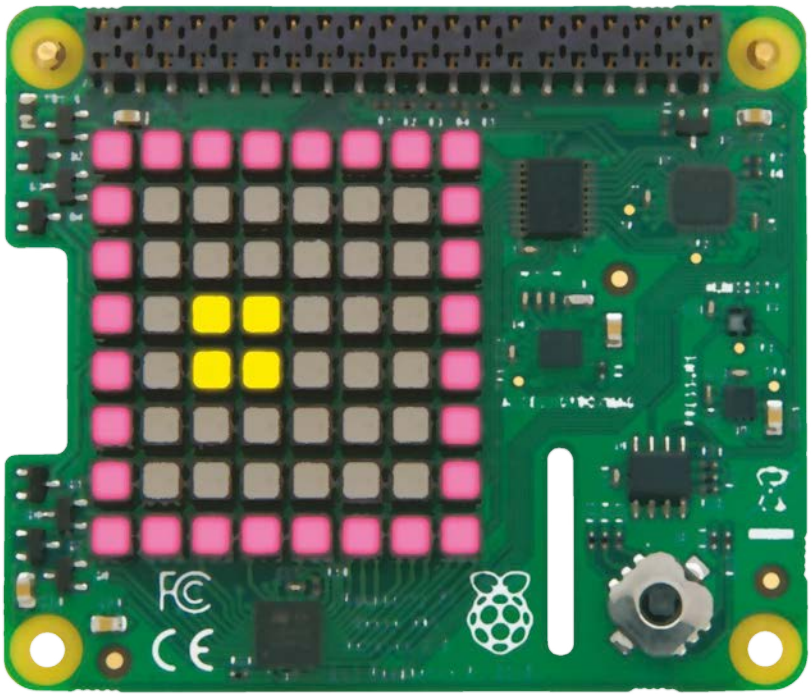
These return a value based on the Sense HAT's respective internal sensor readings. For a full list of the Sense HAT commands, check the FUZE BASIC Programmer's Reference Guide installed in the FUZE BASIC folder under Documentation.

Press **F2** to enter the editor and make sure it's empty (New icon). Now type in the following program:

```
x = 3
y = 3
LOOP
    senseCLS
    senseColour = Raspberry
    senseRect (0, 0, 7, 7, 0)
    senseColour = LightYellow
    senseRect (x, y, 1, 1, 1)
    IF senseGyroY < 3 THEN x = + 1
    IF senseGyroY > 3 THEN x = - 1
    IF senseGyroX < 3 THEN y = + 1
    IF senseGyroX > 3 THEN y = - 1
    IF x < 1 THEN x = 1
    IF x > 5 THEN x = 5
    IF y < 1 THEN y = 1
    IF y > 5 THEN y = 5
    WAIT (0.01)
REPEAT
```

FUZE HARDWARE

You can also buy a full FUZE machine that has a special OS and the ability to program components on the GPIO.



Above Tilt the Sense HAT to move the yellow 'ball' within the confines of the pink border

Now RUN (**F3**) the program. The matrix should display a pink square with a small yellow ball inside it. Gently move your Raspberry Pi by tilting it back and forth. The ball should roll around the matrix display.

We'll briefly explain what the code is doing. Firstly, a couple of variables are set. These are used to determine the coordinates of the ball on the matrix. Remember, computers always count from zero, so **3, 3** is actually the fourth LED in and up on the display.

> "Computers always count from zero, so 3, 3 is actually the fourth LED in and up"

Next, we start a main **LOOP**. The Sense HAT matrix is cleared, the active colour is set to Raspberry, and we draw the outer rectangle with **senseRect (0, 0, 7, 7, 0)**. Here, **0, 0** is bottom-left and **7,7** is top-right. The **0** at the end instructs the command to draw an empty shape, as opposed to a filled-in one.

Then we change the colour to LightYellow and draw the ball at the current x and y positions.

senseGyroX and **senseGyroY** report the current values held by the positional chip on the Sense HAT.

We check to see if the Raspberry Pi has been tilted by a measure of 3 or more. If you want to improve the sensitivity, lower this number to anything above 0. The sensor is incredibly sensitive, so even a bang on the table will register if you have the number set very low.

ANY PI

As long as you can hook a Sense HAT up to it, any Raspberry Pi will work with FUZE!

Finally, we check to make sure we're restrained to the inside of the rectangle, so if x or y goes below 1 or above 5 then they are fixed at that position.

A very short delay of a single hundredth of a second is used to keep things smooth and then the **REPEAT** statement marks the end of the **LOOP** so it returns back.

The Raspberry Pi Sense HAT is a wonderful little add-on, and FUZE BASIC makes it incredibly easy to use. We hope you have fun experimenting with it. FUZE BASIC has around 20 commands and counting. Currently, the following movement detection commands are available:

senseGyroX	Returns the current gyro value
senseGyroY	
senseGyroZ	
senseAccelX	Returns the current accelerometer value
senseAccelY	
senseAccelZ	
senseCompassX	Returns the current magnetometer value
senseCompassY	
senseCompassZ	

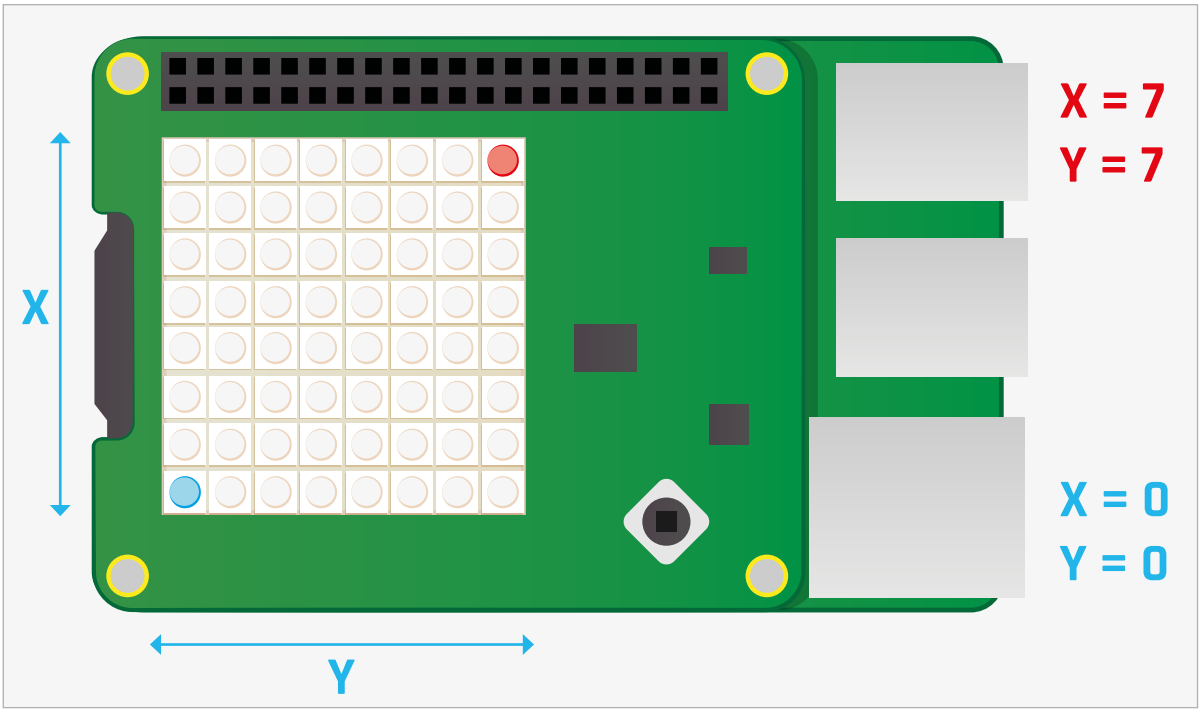
And the built-in sensor commands include:

senseHeight	Returns the current height above sea level – does not work well indoors, although it can be used to get relative height
senseHumidity	Returns the current humidity value
sensePressure	Returns the current air pressure value
senseTemperature	Returns the current temperature value in degrees

You can test all the above commands in Direct mode by typing PRINT in front of any of them and pressing **ENTER**. For example:

```
PRINT senseTemperature
```

...will display the current temperature as reported by the Sense HAT.



Left FUZE remaps the coordinates for each LED compared to the Python version; here's a guide

The LED matrix commands include:

senseCLS	Turns off all LEDs
senseHflip	Flips the matrix display horizontally
senseVflip	Flips the matrix display vertically
sensePlot (x, y)	Lights an LED in the current colour at the specified coordinates
senseLine (x1, y1, x2, y2)	Draws a line from coordinate one to coordinate two
senseRect (x, y, width, height, fill)	Draws a rectangle at the specified coordinates (1 = filled in, 0 = empty)
senseRGBcolour (red, green, blue)	Sets the colour using RGB values
senseColour =	Sets the colour using one of 30 built-in predefined names – 'teal', for example
senseScroll (xDirection, yDirection)	Scrolls the matrix either negatively or positively by the specified number of LEDs

You'll find these very easy to implement, but if you'd like to see them in action, check the Demo folder in the FUZE BASIC folder on the Desktop; this contains a few Sense HAT demos along with others. That's all for now, but please keep an eye on the FUZE website and the magazine for more tutorials.

Language
>BASIC

HelloWorld

```
LOOP
  INK = RND (30)
  PAPER = RND (30)
  fontSize (RND (20) + 1)
  PRINT "Hello World ";
REPEAT
END
```

SenseHAT

```
x = 3
y = 3
LOOP
    senseCLS
    senseColour = Raspberry
    senseRect (0, 0, 7, 7, 0)
    senseColour = LightYellow
    senseRect (x, y, 1, 1, 1)
    IF senseGyroY < 3 THEN x = + 1
    IF senseGyroY > 3 THEN x = - 1
    IF senseGyroX < 3 THEN y = + 1
    IF senseGyroX > 3 THEN y = - 1
    IF x < 1 THEN x = 1
    IF x > 5 THEN x = 5
    IF y < 1 THEN y = 1
    IF y > 5 THEN y = 5
    WAIT (0.01)
REPEAT
```



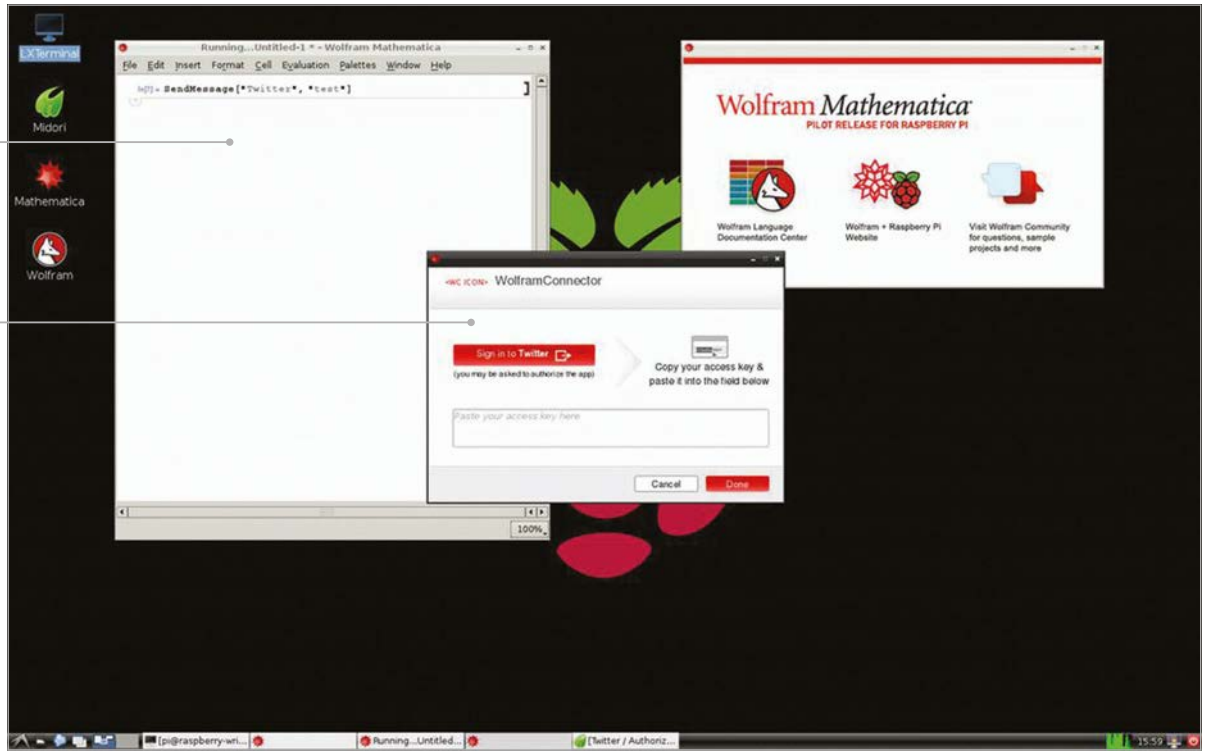


DANA FLINN & BRETT HAINES

Dana Flinn (left) is a project administrator in Public Relations at Wolfram Research, and Brett Haines (right) is a junior software engineer in the Research & Development department.
magpi.cc/1Otb5Af

A single function in the Wolfram Language can send a tweet from your Pi

Authorisation is only required once for the Wolfram Language to tweet on your behalf



You'll Need

- A Twitter account
- The Wolfram Language Documentation Center **magpi.cc/1Otbmn0**
- Internet access

TWEET FROM YOUR RASPBERRY PI WITH THE WOLFRAM LANGUAGE

Dana Flinn and **Brett Haines** walk you through how to easily send tweets in the Wolfram Language, using the SendMessage function

The Wolfram Language is a powerful knowledge-based language that makes computation accessible to everyone, and is an easy language to dip your toes into as a first-time programmer. To send tweets from Twitter on your Raspberry Pi, all you need is a single function. We'll be using SendMessage, which is a multipurpose function in the Wolfram Language that gives Raspberry Pi users the ability to seamlessly tweet from their device.

Using the SendMessage function, we'll walk you through how to apply the symbolic language and tweet using your Raspberry Pi. Don't feel limited to just tweeting with this function, though! SendMessage can also be used to generate an email, update a Facebook or LinkedIn status, and generate a beep.

PYTHON VS WOLFRAM

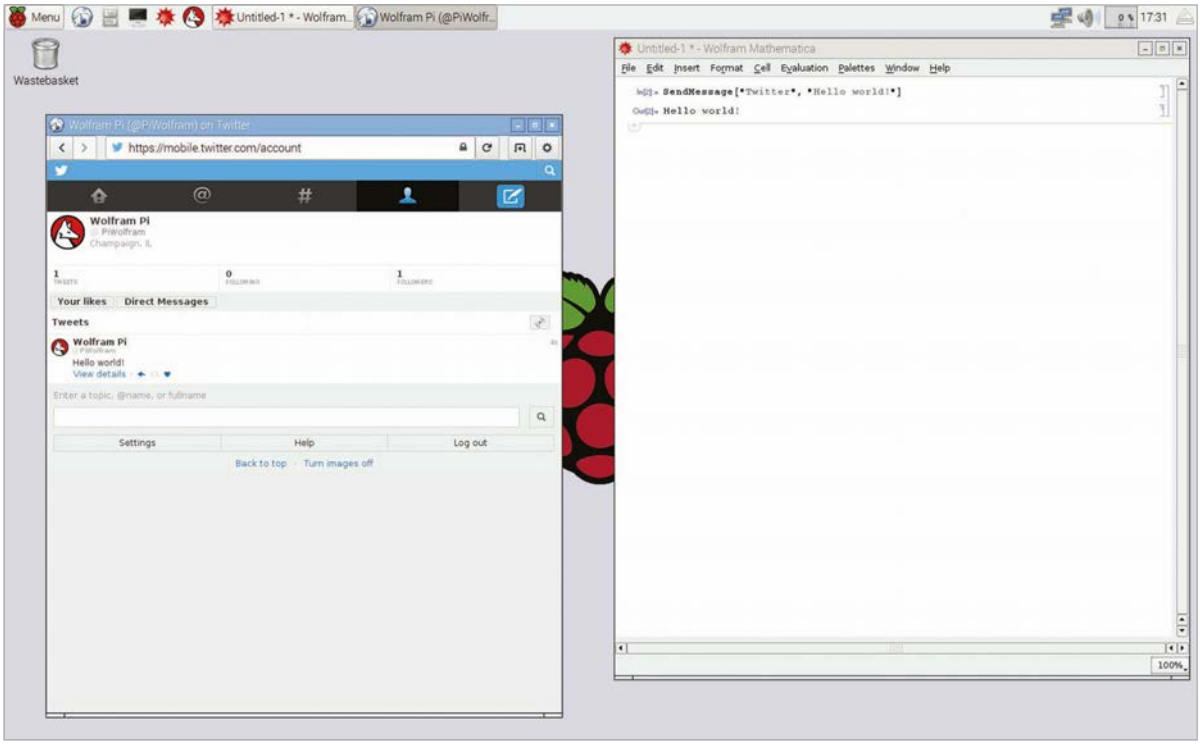
Wolfram allows for much easier access to Twitter compared to Python so, depending on your needs, Wolfram may be all you need.

>STEP-01

Prepare the syntax

First, you need to launch your copy of Mathematica on your Raspberry Pi and open a blank notebook. With over 5,000 functions built into the Wolfram Language, it can seem intimidating at first. However, using the Documentation Center will make it simple to find the right function for your project.

In the Wolfram Language, all functions use square brackets and have names that start with capital letters. The function needed for this project is SendMessage, which is rather straightforward. It's used to send a message to a specified channel, which in our case is Twitter.



Left You can tweet whatever message you want and it will appear instantly – limited to 140 characters, of course!

MORE IN WOLFRAM

Find more projects from the Wolfram team over on the Wolfram site: **magpi.cc/1SCpu2o**

>STEP-02

Setting up your code

To set up your code, open Mathematica on your Pi and type the SendMessage function, then the text you want your tweet to say. So, from the Documentation Center, we know it should be set up like this:

```
SendMessage["Twitter", "Tweet text"]
```

Where it says **"Tweet text"**, the text can be changed to anything you want posted to your Twitter account.

>STEP-03

Evaluate your code

Once you've typed that in, press **SHIFT+ENTER**. This will run your code in Mathematica and prompt an authorisation dialogue to allow the Wolfram Language to send tweets on your behalf. Once authorised, a tweet is sent to your account. Select 'Done' after entering the access key, and the tweet will appear in your Twitter stream.

>STEP-04

Create a variable

The previous example showed a static string being tweeted. However, this is far from all the Wolfram Language is capable of. Much like other programming languages, it allows users to set variables, in which we can store whatever we like. To create a variable, just type the name you want to give it, an equals sign, then whatever you want to assign to the variable, like so:

```
myVariable = "Some string here!"
```

Unlike several other programming languages, you don't need to specify what kind of data is being stored in your variables – the Wolfram Language is smart enough to figure out what it is and how to use it on its own!

>STEP-05

Using StringJoin

You can combine strings using the **StringJoin** function. Just like SendMessage, StringJoin is used by putting the strings you want to join in between square brackets ([and]), separated by commas. So, to join "Hello " and "world", you'd enter:

```
StringJoin["Hello ", "world"]
```

This isn't limited to static strings – variables can be joined to strings too, like so:

```
StringJoin["Hello ", myVariable]
```

>STEP-06

Combining it all

Now it's time to put it all together. We can take a variable, store a string to it, join it to another string, then send it to Twitter:

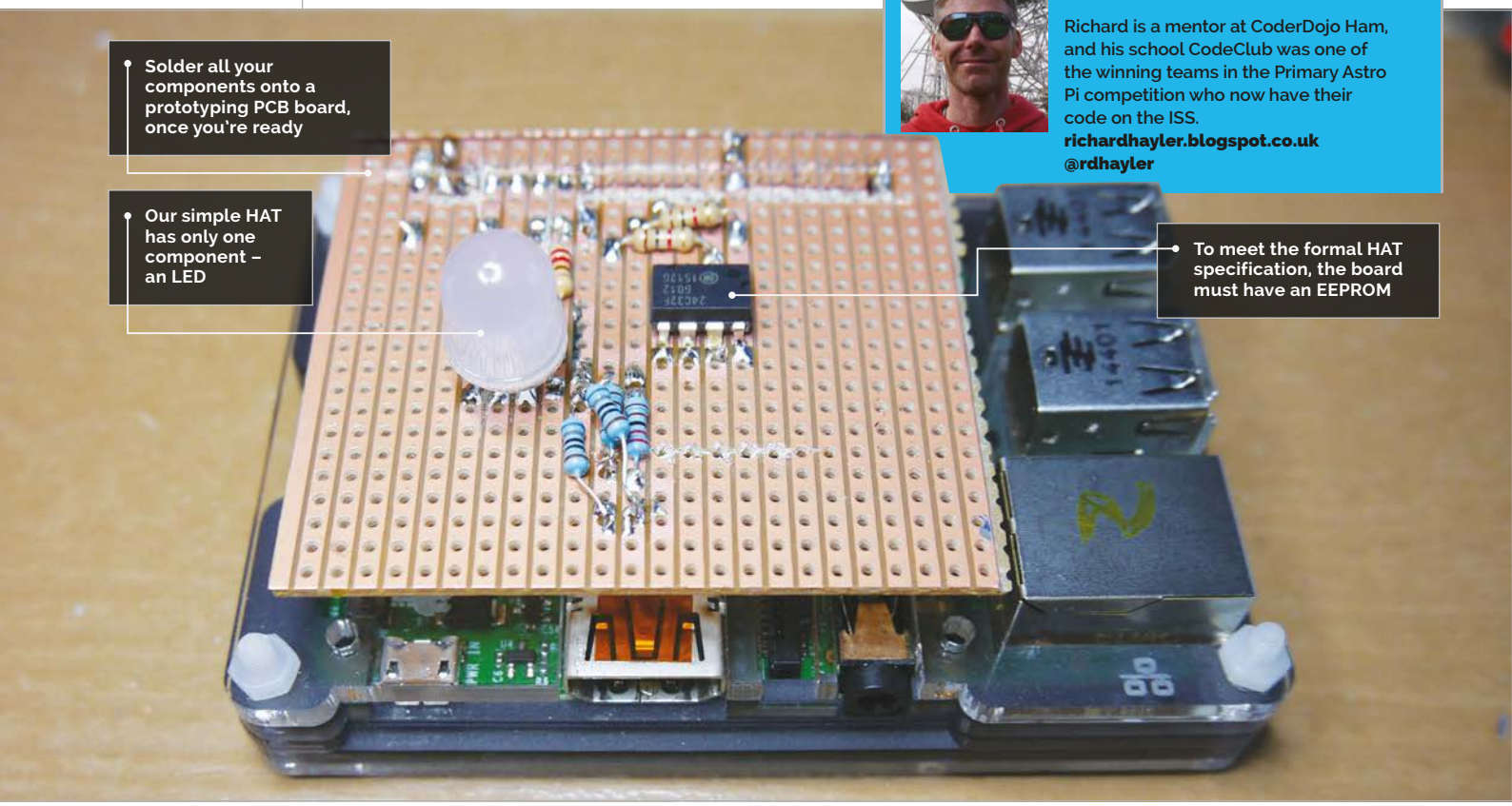
```
myName = "WolframLanguage"
SendMessage["Twitter", StringJoin["This
tweet brought to you by ", myName]]
```

Now that you know how functions work in the Wolfram Language, it's easy to dive into the documentation and explore the thousands of functions built in. Now go forth and code!



RICHARD HAYLER

Richard is a mentor at CoderDojo Ham, and his school CodeClub was one of the winning teams in the Primary Astro Pi competition who now have their code on the ISS.
richardhayler.blogspot.co.uk
@rdhayler



You'll Need

- An EEPROM (a CAT24C32 is recommended) **magpi.cc/1RwAaQP**
- A breadboard and some jumper leads **magpi.cc/1RwAdMt**
- Some LEDs **magpi.cc/1RwAiQi**
- The Pi HAT specification repository **magpi.cc/1OAHaGk**

MAKE YOUR OWN PI HAT

There are some amazing HATs available to enhance the hardware capability of your Pi. Ever wondered how to design and build a prototype HAT? This tutorial will show you how to get started

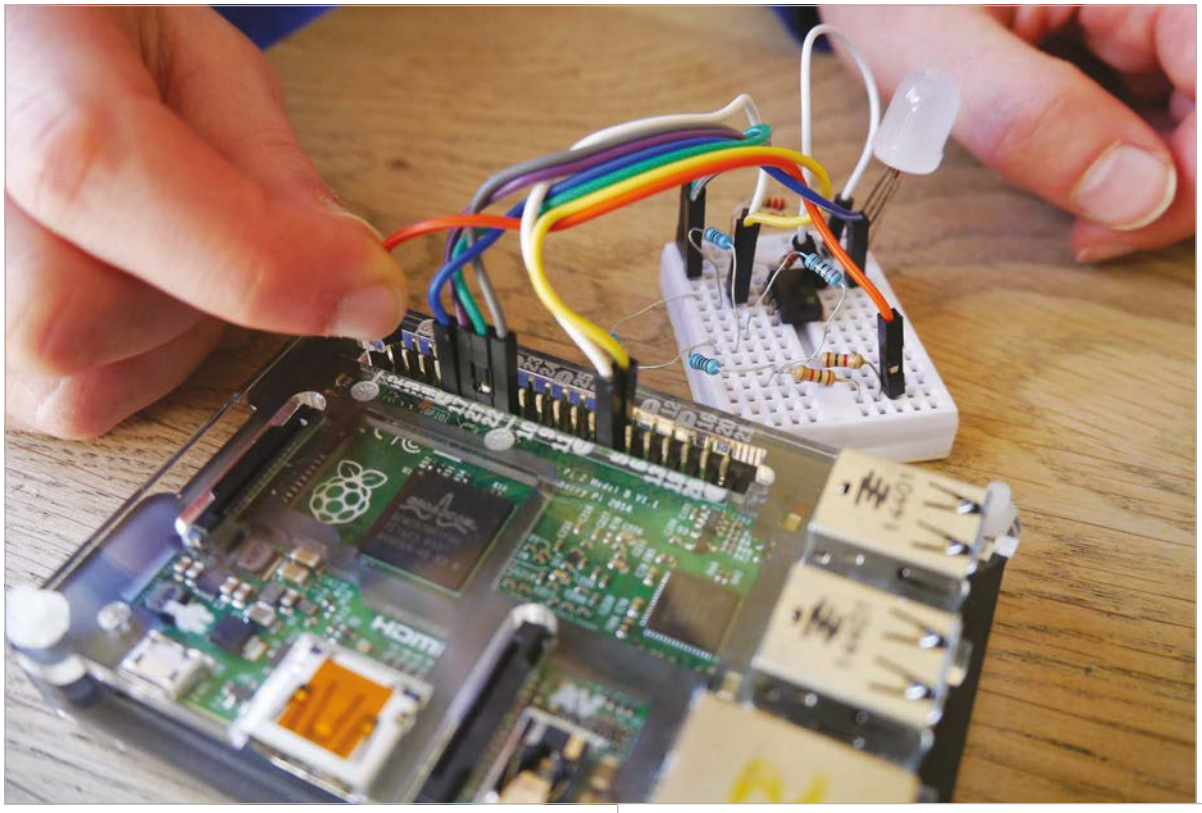
HATs first arrived with the launch of the Raspberry Pi Model B+ in 2014. Since then, there have been some amazing and wonderful HATs, many of which have featured in previous issues of *The MagPi*: theUnicorn HAT, Piano HAT, Sense HAT, and Explorer HAT. Officially, HAT stands for 'Hardware Attached on Top,' although your expert suspects that the acronym came after the name. What else would you call an add-on board that sits on top of the Pi?

Although you can manufacture and sell boards that don't conform to the HAT standard, there are many advantages in doing so. Having the same physical dimensions makes it easy to ensure your board will fit with most Pi cases, and the inclusion of an EEPROM allows the operating system to identify the HAT and set up any required hardware at startup.

These instructions will work with either the latest version of Raspbian (Jessie) or the older Wheezy. As usual, run a **`sudo apt-get update`** and **`sudo apt-get upgrade`** before starting. Then enable kernel support for I²C using:

```
sudo raspi-config
```

Choose 'Advanced Options', then tap down and select 'A7 I2C'. Make sure 'Yes' is highlighted for the



Left Build your circuit on a breadboard or Explorer HAT first to make sure it works, allowing you to make changes

WHAT IS AN EEPROM?

EEPROM stands for 'Electrically Erasable Programmable Read-Only Memory' and is a chip used in electronic devices to store small amounts of data.

answers to both questions. Alternatively, if you're using Jessie, you can set this up using the Raspberry Pi configuration tool available through the GUI.

You will then be prompted to reboot. Once your Pi has restarted, install the I²C tools package with:

```
sudo apt-get install i2c-tools
```

Now it's time to construct your EEPROM circuit on a breadboard. Put together the circuit as shown in **Fig 1**. If you're not using a 24C32, you may need to connect to different pins for SDA and SCL. Check the data-sheet for your IC to make sure you get this correct. You also need to give the HAT some functionality: in the interests of keeping things simple, the circuit of Fig 1 just contains a tri-colour RGB LED, but you can use whatever components you wish, depending on what you want your HAT to do.

Power up the Pi. You may notice that the LED is already on (perhaps just faintly). This is because the Pi doesn't know what is connected and so the GPIO output is 'floating'. This is something our HAT will deal with by preconfiguring the pins that it uses. The code snippet overleaf (**myoh-rgb.py**) provides some simple Python for interacting with the LED; you'll need to install the wonderful gpiozero library to use it.

Next, check to see what I²C devices have been detected on the second I²C bus:

```
i2cdetect -y 1
```

Fig 1 The EEPROM should be connected to SDA and SCL for programming. If you're not using a 24C32, you may need to connect to different pins (check the data-sheet for your IC)

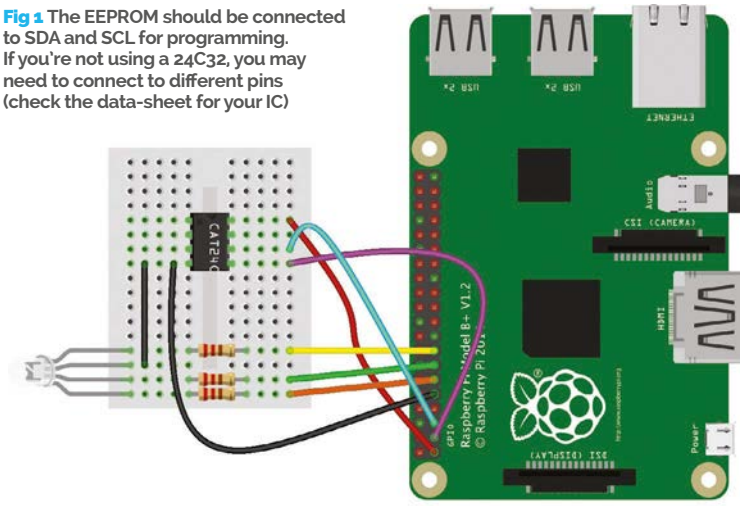
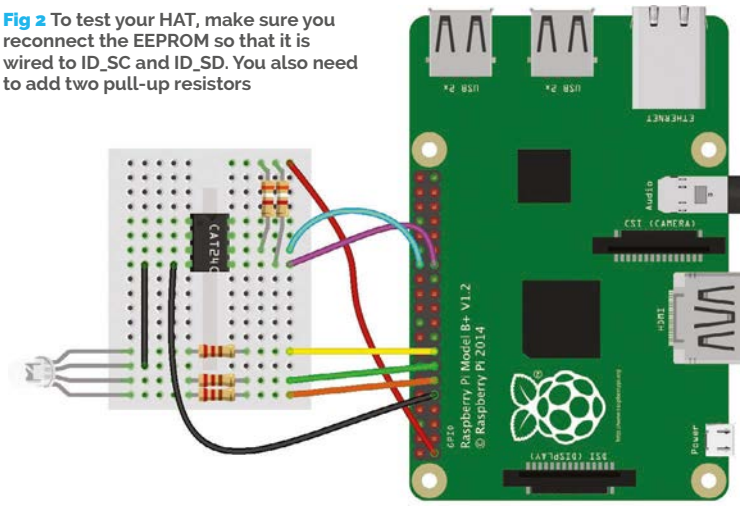


Fig 2 To test your HAT, make sure you reconnect the EEPROM so that it is wired to ID_SC and ID_SD. You also need to add two pull-up resistors



STATIC WARNING!

EEPROMs can be easily damaged by static electricity, so try to ground yourself before handling them. Ideally, wear an anti-static wrist strap, or at the very least leave that nylon sweater in the wardrobe.

You should see a table displaying the devices found on your I²C bus. Assuming you have nothing else connected, all but one of the entries should be empty, as represented by 2 dashes: – –. Normally, your EEPROM should show up in the leftmost column (0) as the number 50, next to the 50 on the horizontal axis.

If you don't see this or the command returns an error ('No such file or directory'), try the other bus:

```
i2cdetect -y 0
```

If your EEPROM still isn't showing up, check your connections and that you have I²C support enabled.

Now you need some software. Clone the HATs reference material and tools from GitHub (**magpi.cc/1OAHaGk**). It's well worth reading this documentation, as it explains the thinking behind the HAT specification in lots of detail.

You're also going to use the handy EEPROM tools to flash your chip. First of all, compile the eepmake tool:

```
cd hats/eepromutils
make
```

Then modify the eepflash script, which assumes that the EEPROM is on the first I²C bus; if yours is, then you can obviously skip this step. Copy the **eepflash.sh** and then modify with your favourite text editor:

```
cp eepflash.sh eepflash1.sh
nano eepflash1.sh
```

Change all mentions of **i2c-0** to **i2c-1**, and **i2c-0/0-0050** to **i2c-1/1-0500**, then save the file.

Now you need to modify the supplied template with your own settings. Open the **eeprom_settings.txt** file in your favourite editor and modify the various fields. Most are self-explanatory and you can read more about each one in the specification. However, the UUID is particularly important and must comply to RFC 4122, so that every HAT can be uniquely identified and can therefore be used as a per-board serial number. It also allows HATs to be stacked.

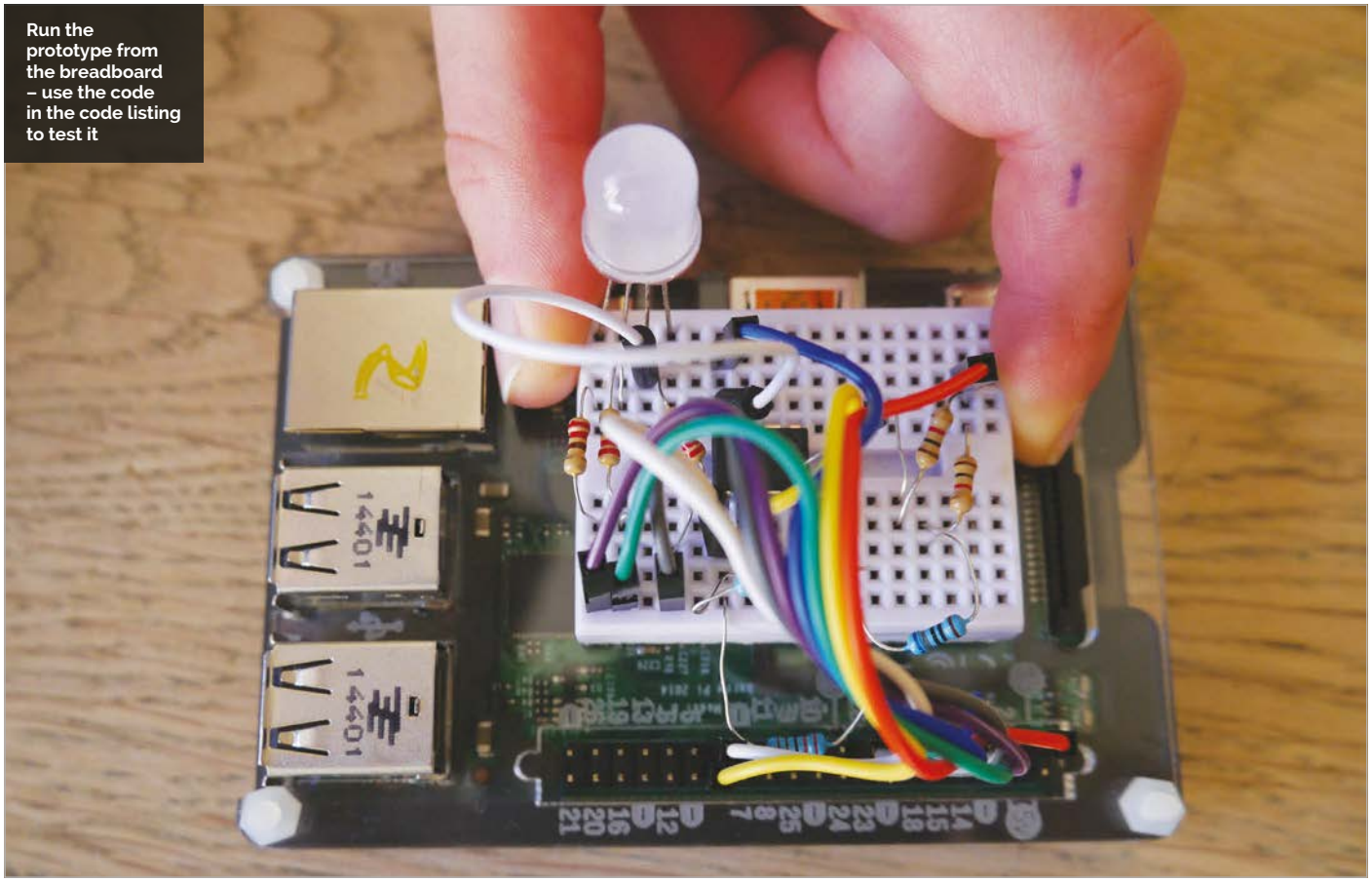
Once that's done, you need to convert the human-readable text file into binary data that can be written to the EEPROM.

```
./eepmake eeprom_settings.txt eeptest.eep
```

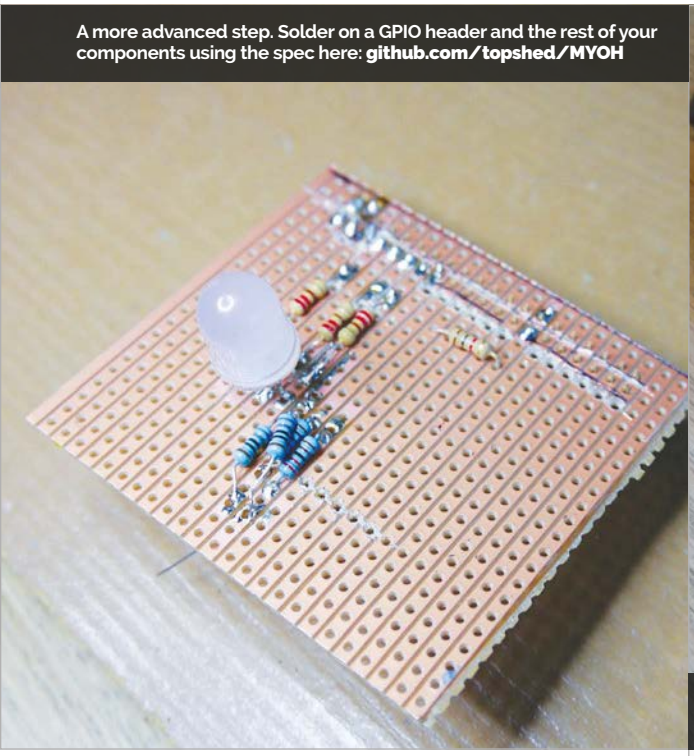
Then dump this data to the EEPROM – this is the actual flashing bit and you'll see a warning asking you if you're sure you want to go ahead:

```
./.eepflash1.sh -w -f=eeptest.eep -t=24c32
```

Make sure you use the version of the script we modified earlier. If you're using a different type of EEPROM, change the **-t** parameter accordingly.



Run the prototype from the breadboard – use the code in the code listing to test it



A more advanced step. Solder on a GPIO header and the rest of your components using the spec here: **github.com/topshed/MYOH**



Slice up some stripboard to the correct size for a HAT that fits over the Pi

To test your new creation, you need to modify your circuit so that the EEPROM is connected to the Pi as it would be if it were part of a HAT. GPIO pins 27 and 28 are reserved for HAT detection, so power down your Pi again and move the appropriate wires as shown in **Fig 2**. Then add the 3.9kΩ pull-up resistors which ensure that the pins are not 'floating'.

Now reconnect the power to your Pi and let it boot. If everything has worked, the directory **/proc/device-tree/hat** should be present, and the contents of its files should match the details in the default **eeprom_settings.txt** file. You should also see that the GPIO pins used by the RGB LED have now been preset, so it should not be lit at all.

> See what other amazing add-ons you can create!

Congratulations! You have now programmed an EEPROM that the Pi recognises as meeting the specification for a HAT.

As a final test, run the **myoh-rgb.py** code and check that the RGB LED flashes through its sequence of colours. Clearly, this is a very basic HAT; there are loads more possibilities for using the ability to have the Pi preload the necessary drivers, so that more complicated hardware can be easily supported. The Sense HAT is a great example of this technique.

Now that you've made a simple HAT and got to grips with the process, see what other amazing add-ons you can create!

If you want to completely follow the whole HAT design process, you now have the option to make the transition from breadboard to proper circuit board. The Raspberry Pi Foundation specification also sets the physical properties of the board, right down to the radius of the rounded corners and the inclusion of slots for the camera and display cables. For prototyping, you probably needn't worry about this too much; a simple stripboard design is available in the GitHub repository for this tutorial, and is pictured in the main image. If you want to take things even further, the marvellous Fritzing CAD software has a built-in PCB template that can help you produce a final design suitable for manufacture.

myoh-rgb.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1RwC040

```
from gpiozero import RGBLED
from time import sleep

led = RGBLED(22,27,17)

led.on()
sleep(0.5)
led.off()
led.red = 1
sleep(0.5)
led.red =0
led.green=1
sleep(0.5)
led.green = 0
led.blue = 1
sleep(0.5)
led.blue=0
```

FREQUENTLY ASKED QUESTIONS

NEED A PROBLEM SOLVED?

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

Your technical hardware and software problems solved...

HOW CAN I BACK UP MY RASPBERRY PI?

Save files from SD card

The most important part of your installation is the files you keep on your Raspberry Pi. You can simply copy these files, documents and programs from the home folder to a USB stick, or via card reader to a computer, regularly.

Cloud backup

You can install Dropbox on the Raspberry Pi. However, it's a bit more complicated than on other computers. With the right settings, you can make sure your important files are always kept on the cloud so they can be restored if needed.

Clone the SD card

A slightly longer but more thorough method involves creating an image from your SD card. This saves all your settings and software as well, and can be reinstalled to your SD card much like you would install an operating system to it.



HOW DO I UPDATE MY RASPBERRY PI?

Software upgrade

Sometimes you'll need to install newer versions of software to make sure everything works properly. Open up the terminal and run the following two commands to update the software: **`sudo apt-get update`** followed by **`sudo apt-get upgrade`**.

Firmware upgrade

The kernel and system software can be updated using the firmware updater rpi-update. You won't need to do this often, but it can be completed by opening the terminal and typing: **`sudo rpi-update`**.

Reinstall

In the case of upgrading your Raspberry Pi operating system to the newest version of Raspbian, called Jessie, you'll need to completely reinstall. Look at our backup FAQ to find out how to save any of your important files.

WHICH OPERATING SYSTEM SHOULD I USE ON RASPBERRY PI?

Raspbian

The main operating system for the Pi is lightweight and runs well. A lot of tutorials and projects run on it, so it's very useful to people of all skill levels. It also gets the most support from the Raspberry Pi Foundation.

OpenELEC for media

If you want to use your Raspberry Pi to power a TV's media centre, than OpenELEC is one of the best solutions. It's very small and quick to install, while also coming with all the software and extras you'd want in order to stream or play media on a TV, such as support for remotes.

Ubuntu MATE

A more traditional desktop experience for your Raspberry Pi, although it doesn't run as quickly as Raspbian. With a few tweaks, you can have it work as a replacement desktop quite easily, and with full access to Ubuntu software as well.

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

Does the Raspberry Pi support networking?
The Model B, Model B+, and Model 2 B versions of the device have built in 10/100 wired Ethernet. There's no Ethernet on the Model A, Model A+, or Pi Zero versions of the board.

Is there built-in WiFi?
No model of the Raspberry Pi has built-in WiFi, but all models can support a USB WiFi dongle. The Foundation offers its own branded WiFi dongle which has been fully tested for use with the Raspberry Pi. It's available through the Swag Store (**magpi.cc/1PCpVb5**). You can, of course, use a dongle from another provider if you wish, a selection of which has been tested on the eLinux wiki (**magpi.cc/1P2pwPH**).

Why is there no Gigabit Ethernet on the Raspberry Pi?
The Ethernet is attached via the USB 2.0 bus, so the upstream bandwidth would not support Gigabit.

Will there ever be a built-in WiFi option?
Unlikely. The SoC does not support native WiFi, and adding an additional built-in WiFi chip would greatly increase the cost of the Raspberry Pi.

Does the Raspberry Pi have support for any form of netbooting or PXE?
The Raspberry Pi does not support PXE booting or network booting without an SD card. If you want to network-boot multiple Raspberry Pis, you could use PiNet (**pinet.org.uk**). This is a free and open-source community-based project initially designed for schools. Each Raspberry Pi boots off a small set of startup files on an SD card, and fetches the rest of the data it needs from the PiNet server, thereby allowing you to maintain a single operating system image for all the Raspberry Pis. PiNet also adds network user accounts, shared folders, and automated backups.

THE MAGPI APP

Having trouble with *The MagPi* on the App Store or Google Play? Here are your most common questions answered:

How do I find *The MagPi* on Google Play or the App Store?
All you have to do is go to the search bar and type 'The MagPi' or 'Raspberry Pi' to find us.

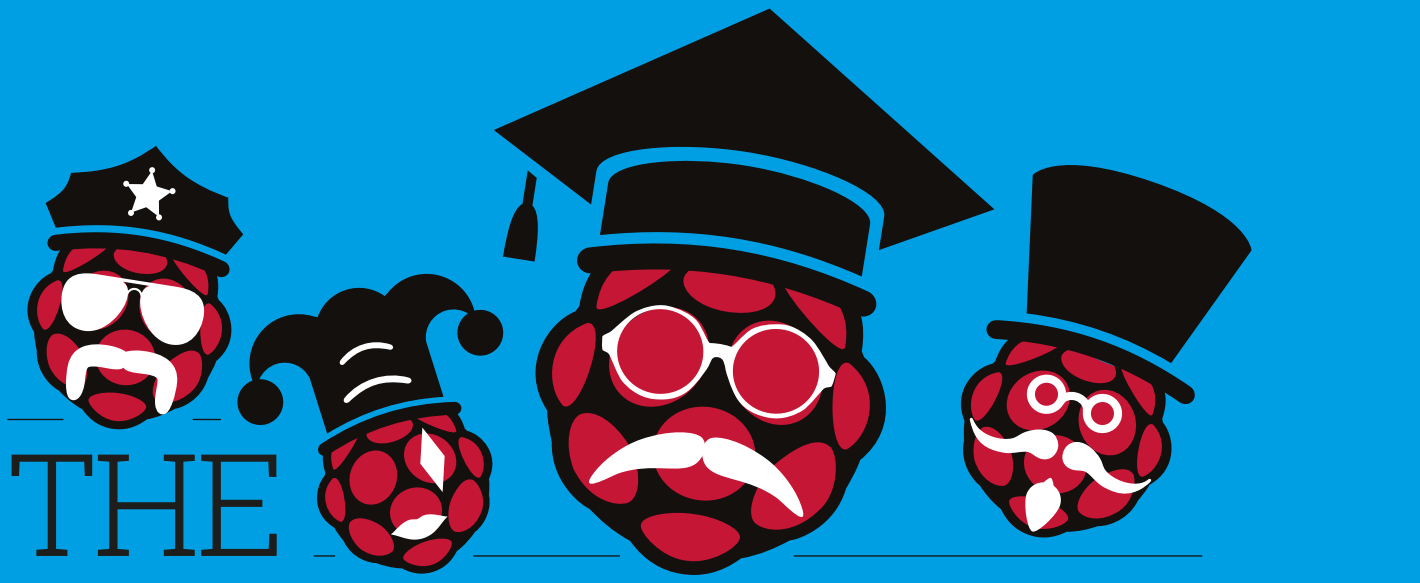
I've subscribed to the digital edition and I can't sign in to restore my purchases. Please help!
Since your *The MagPi* purchases are linked to your Google or Apple accounts, there's no need to sign in at all. If you'd like to re-download your purchases on your current device, or make your purchases available on other devices, all you need to do is press 'Subscribe' on the home screen, then 'Restore Purchases' on the next screen.

How can I search the digital magazine for keywords?
Finding direct references is really easy with *The MagPi* app – all you have to do is tap the screen to get the app's GUI to show, and then press the small magnifying glass icon in the top-right corner of the screen. Now, just type in your search term to find the relevant results.



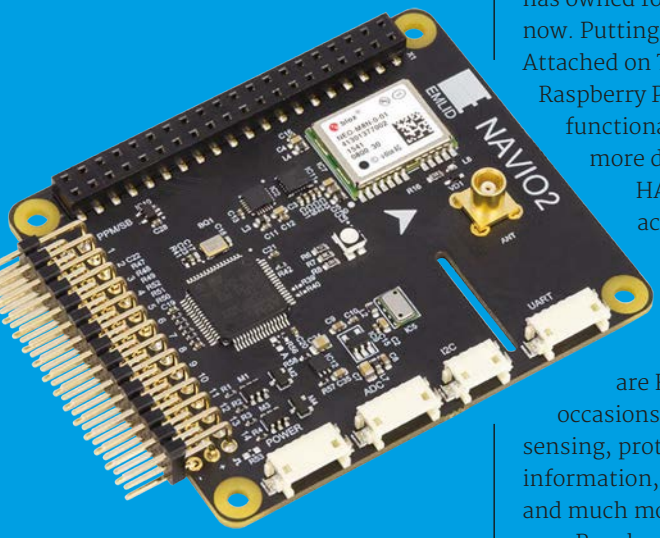
GET IT ON Google Play

Available on the App Store



THE 10 BEST RASPBERRY PI HATS

Your Raspberry Pi looks a lot cooler wearing a HAT, and it becomes more functional too. **Here are the 10 best HATs for the Raspberry Pi**



Being able to pull off a good hat is a great life skill, and it's one the Raspberry Pi has owned for a couple of years now. Putting a HAT (Hardware Attached on Top) device on your Raspberry Pi makes it more functional, more powerful, and more dapper.

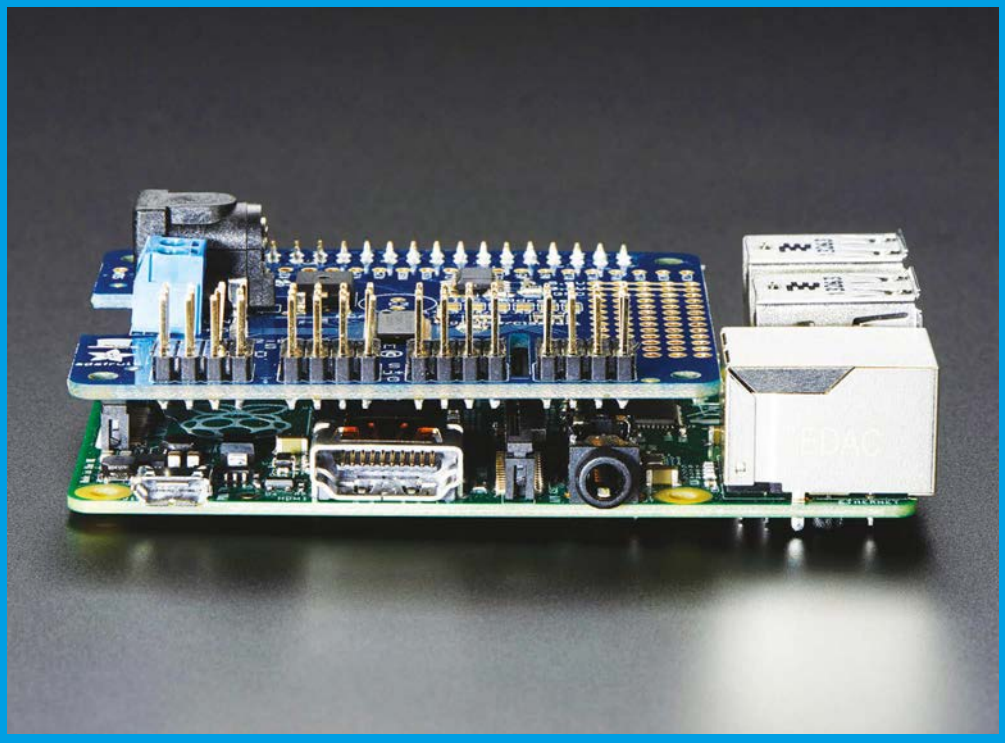
HATs are a type of accessory for the Raspberry Pi, designed to be easy to set up and easy to use. There are HATs for all kinds of occasions: you can buy HATs for sensing, prototyping, displaying information, controlling robots, and much more. A HAT can make your Raspberry Pi experience much more fun because it enables you to get on with building your project without having to worry about whether your hardware will work.

HATs are also designed to sit on top of the Raspberry Pi (Model A+, B+, or 2), plugging into its GPIO header, with optional extra attachment via the four mounting holes. The HAT sits on top of the Raspberry Pi, hence the name.

You can't just put any old board on top of a Pi and call it a 'HAT', though. HATs have to meet a set of rigorous standards laid down by the Raspberry Pi Foundation.

As well as fitting on top of the Raspberry Pi, a HAT has to work seamlessly. The idea behind a HAT is that your Pi should be able to detect when one is attached.

Other accessories typically require you to follow detailed instructions to install drivers and get the hardware up and running;



> Putting a HAT on your Raspberry Pi makes it more functional, more powerful, and more dapper

a HAT should start working the instant it's connected.

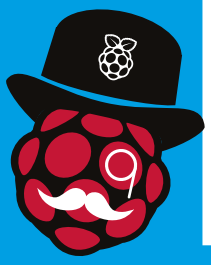
Raspberry Pi achieves this using two specific GPIO pins found on the Model A+, B+, and 2: ID_SD and ID_SC (pins number 27 and 28). These pins are reserved to connect to an EEPROM (Electrically Erasable Programmable Read-Only Memory) chip on the HAT. The EEPROM chip contains code in a .dtf file, known as a 'device tree', that outlines all the chips and hardware on the HAT.

When a HAT is connected to a Raspberry Pi, the Pi detects it, reads the .dtf file, and difascovers what hardware is attached. Raspbian then loads the required drivers for the hardware automatically, sets up the GPIO pins accordingly, and the HAT is ready to use.

Not every piece of hardware you buy for the Raspberry Pi is a HAT. Some types have special setup requirements, and others don't fit on top of the Pi like a HAT. So HATs aren't necessarily better than non-HATs, but if a piece of hardware is called a HAT then you know it'll be relatively easy to set up and use.

"While we cannot force anyone to follow our minimum requirements or HAT specification," says James Adams, director of hardware at Raspberry Pi, "doing so will make users' lives easier, safer, and will make us more likely to recommend a product."

HATs are great, and you can get so much more out of a Raspberry Pi using HAT hardware. Here are our top ten favourite HATs...



Behind THE HAT

Anybody can make a HAT, but they have to follow a specification laid down by Raspberry Pi. Here are three things that make a HAT:


B+ ADD-ON BOARDS AND HATs

NOTE All references to GPIO numbers within this document are referring to the BCM2835 GPIOs (NOT pin numbers on the Pi GPIO header).

Introduction

The Raspberry Pi B+ has been designed specifically with add-on boards in mind. For B+ we are introducing 'HATs' (Hardware Attached on Top). A HAT is an add-on board for B+ that conforms to the HAT specifications. HATs are not backward compatible with models A and B.

There are obviously a lot of add-on boards designed for the original model A and B boards (which interface to the original 26 way GPIO header). The first 26 pins of the B+ GPIO header are identical to those of the original models, so most existing boards will still work.

As well as now being a 40W GPIO header (originals on B and A are 26W) the biggest change with B+ add-on boards versus older boards designed for models A and B is that the B+'s header has 2 special pins (ID_SC and ID_SD) that are reserved exclusively for attaching an 'ID EEPROM'. The ID EEPROM contains data that identifies the board, tells the B+ how the GPIOs need to be set up and what hardware is on the board. This allows the add-on board to be automatically identified and set up by the Pi software at boot time including loading of the necessary drivers.

While we cannot force anyone to follow our minimum requirements or HAT specification, doing so will make users lives easier, safer, and will make us more likely to recommend a product. Likewise if one of the minimum requirements is ignored we are unlikely to look on a product very favourably.

Specifications

The HAT hardware must follow the specifications set by the Raspberry Pi Foundation. "We want to ensure consistency and compatibility with future add-on boards, and to allow a much better end-user experience..." says the README.md document. **magpi.cc/1OAHaGk**



Design

The HAT must have dimensions of 65mm × 56mm, connect to the GPIO pins (using a full 40-pin header), and feature four corner holes for mounting on the Pi Model A+, B+, and 2. **magpi.cc/1RPfNig**

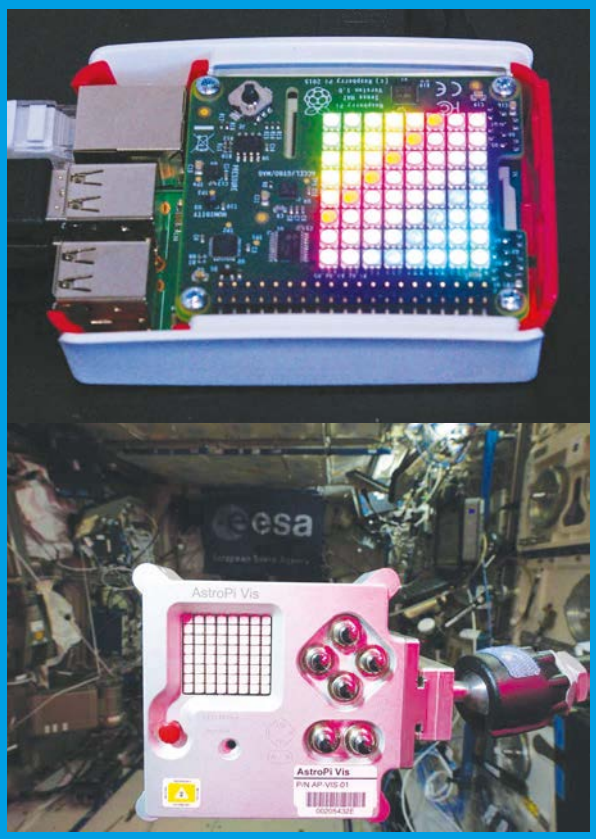
EEPROM



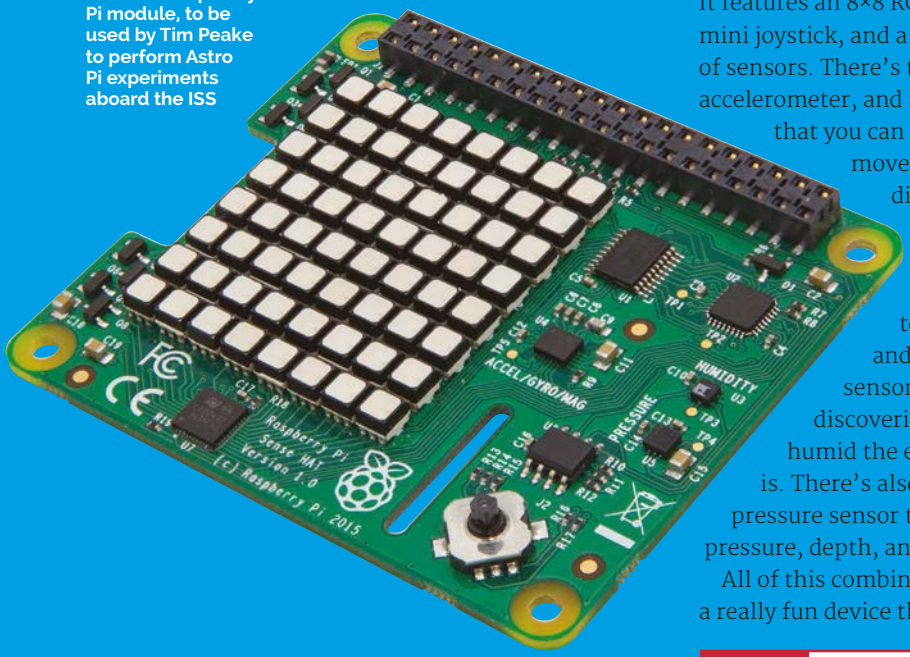
The HAT must contain an EEPROM that details the hardware contained on the board. This allows the Pi to identify a connected HAT and automatically configure the GPIOs and drivers for the board.

SENSE HAT

Made for space and just as much fun on Earth, the Sense HAT is the coolest add-on around



Above The Sense HAT and Raspberry Pi module, to be used by Tim Peake to perform Astro Pi experiments aboard the ISS



We couldn't do a top ten of Raspberry Pi HATs and not start off with the Sense HAT. Developed for the Astro Pi mission (**astro-pi.org**), the Sense HAT went into space shortly before Tim Peake, who will be running experiments with it aboard the ISS.

Cool doesn't even come close to describing the Astro Pi mission, and having a Sense HAT enables us all to get a bit closer to living the dream of being an astronaut.

> "Cool doesn't even come close to describing the Astro Pi mission"

It helps that the Sense HAT is a really good piece of hardware. It features an 8×8 RGB LED display, mini joystick, and a whole bunch of sensors. There's the gyroscope, accelerometer, and magnetometer that you can use to detect movement and direction of the Raspberry Pi. Alongside these sit temperature and humidity sensors, for discovering how hot and humid the environment is. There's also a barometric pressure sensor to measure air pressure, depth, and altitude.

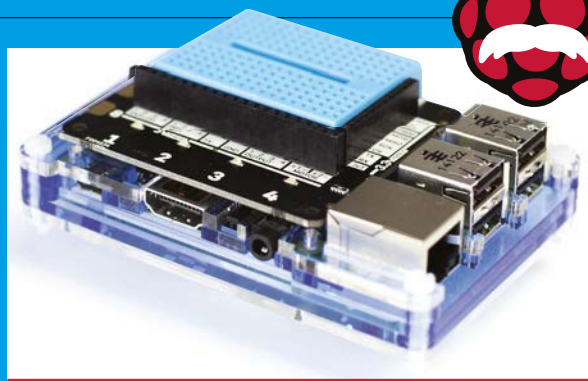
All of this combines to create a really fun device that can be used for an incredibly diverse range of projects. You can create a compass that uses the LED matrix to point north, measure the environment around you, or use the joystick and screen to create a game of Snake. Some Sense HAT projects are super simple, while others are extremely complex data logs of movement and environmental changes.

The great thing about Astro Pi (aside from the coolness of space) is that it has pushed an incredible level of community support to the Sense HAT. Because it's formed part of Tim Peake's high-profile mission to space, it's got a lot of user-generated projects, as well as a great collection of tutorials curated by Raspberry Pi itself (**astro-pi.org/resources**).

Like all HATs, it's easy to get started with, but you'll get more from it in the long term. You can program the Sense HAT in Python or Scratch, so it's great for beginners, and you get a lot of additional Raspberry Pi functionality for the price.

Now that Tim Peake's up in space, and soon to be running his tests on a Raspberry Pi (and sharing photos with us back on Earth), there isn't a better time to pick up a Sense HAT and join in with the experiment.

£23 magpi.cc/1TGGUt5



EXPLORER HAT

Electronic projects are made perfectly simple with this prototyping HAT

The Explorer HAT makes it much easier to build prototypes on top of your Raspberry Pi. It provides buffered 5V tolerant inputs and outputs, so you can attach and detach components to/from your Pi without worrying about damage.

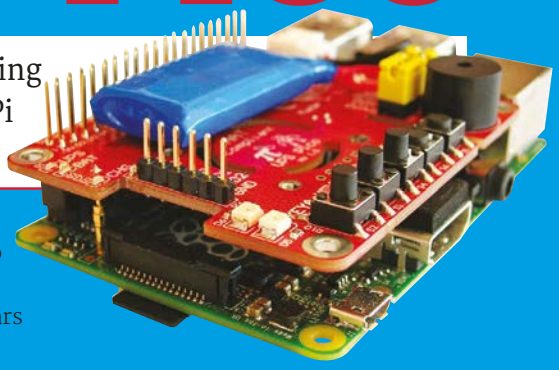
It also has capacitive touch pads and crocodile clip pads, along with four coloured LEDs to serve as alert lights. A PRO version (£18) adds analogue inputs and motor drivers. All of this sits beneath a mini breadboard that's ideal for working on smaller projects.

At £10, it's a great way to add a little extra functionality to your Raspberry Pi and make it easier to learn electronics projects.

£10 magpi.cc/10AKy46

UPS PICO

Keep the power running with this Raspberry Pi power solution



When building real-world devices that are designed to just work, the prospect of interrupted power often rears its ugly head.

Fortunately, the UPS Pico (Uninterrupted Power Supply) saves the day. Its main function is to keep the Raspberry Pi safe during a power cut. The basic model will safely shut down the Raspberry Pi during a power cut; the advanced model will keep it up and running for eight hours then shut it down safely. It keeps itself charged up via the GPIO pins, and an embedded measurement system detects if a power problem has occurred.

€25.99 (£19.83) pimodules.com

PIANO HAT

Discover your inner Mozart with this musical HAT



Piano HAT started out as an Indiegogo project by Zachary Igielman, who wanted to create a musical add-on board for the Raspberry Pi that would teach soldering skills and the exploration of music and sound generation.

Pimoroni picked up the project and worked with Zach to turn it into the stylish Piano HAT. It slots on top of the Raspberry Pi and adds 13 capacitive piano keys, along with two keys for changing the octave and one for switching the instrument. It comes with a software library, so you can play example tracks and use the Piano HAT to explore music.

£15 magpi.cc/10ALwNT



UNICORN HAT

Turn your Raspberry Pi into a rainbow of colour with this stylish Unicorn HAT

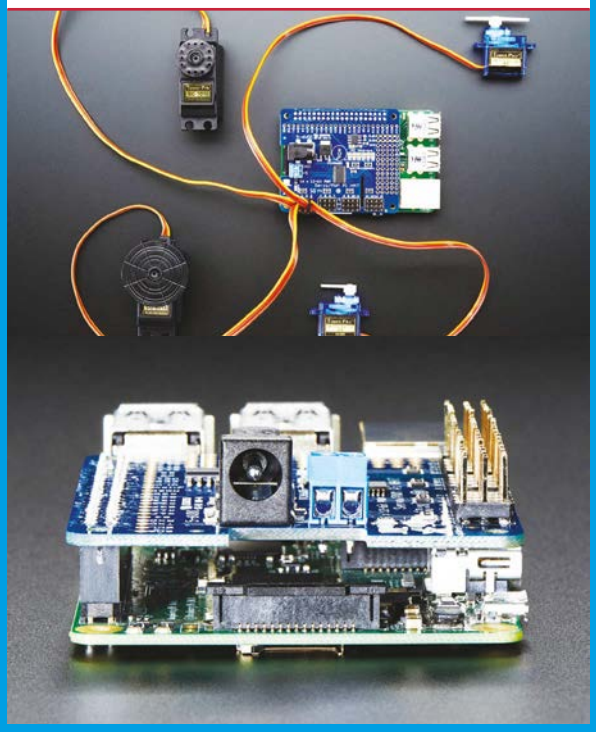
The Unicorn HAT is one of the coolest HATs around. It features 64 RGB LEDs arranged in an 8×8 pattern to form a vibrant display. The LED lights are phenomenally bright, and the Python API can be used to create all kinds of crazy visual displays.

There are some great projects around. You can create pixel works of art from your web browser, make rainbows appear, build animations, and even create basic retro games like Pong. It's one of the most fun HATs around, and a great way to make programming instantly vibrant and interesting.

£24 magpi.cc/1PCpRZ5

ADAFRUIT SERVO HAT

Take charge of your motors with this cracking little servo control HAT



The Raspberry Pi is great for a lot of things fresh from the box, but one thing it needs a hand with is controlling servos (RC servomechanisms).

Small robotics projects often require servos, and they need specific repetitive timing pulses to get into position.

While it is possible to switch the GPIO pins quickly on and off to send the pulses, it's tricky to get the timing right. Robotics projects also benefit from having a separate device to manage the motors, because of the additional power required by servo motors – typically between 5V and 6V each.

So if you're looking to get serious about robotics, you'll want a servo control board like this Adafruit Servo HAT. Because it's a HAT, it's really easy to set up and you can control up to 16 servo motors simultaneously. (If you're building something truly ridiculous, you can even stack up to 62 Servo HATs on top of each other to control up to 992 servos.)

The real advantage is that Adafruit has built a Python library that you can use to control your robot, allowing you to focus on the design and management of the robot rather than the intricate details of programming each servo.

It also solves the power issue by having a second power supply that you connect through a 2.1mm DC jack on the HAT. This supply is used to provide power to the motors, while the board's power management chip gets its power from the Raspberry Pi.

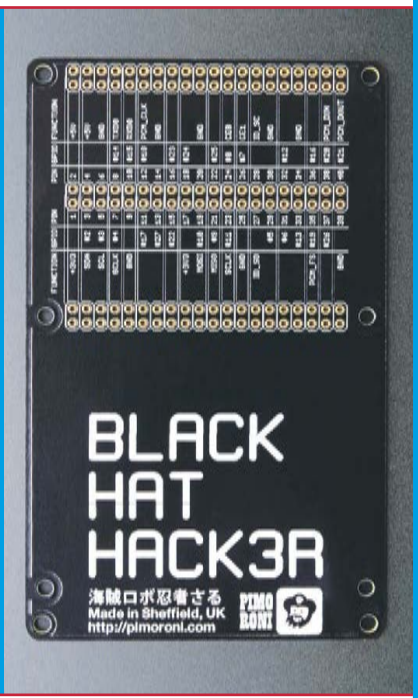
£16 magpi.cc/10AMMAy

BLACK HAT HACK3R

Access your GPIO pins while running a HAT

All of the HATs here offer a range of different functions, but most of them take over the GPIO pins on your Raspberry Pi and don't let you access them any more.

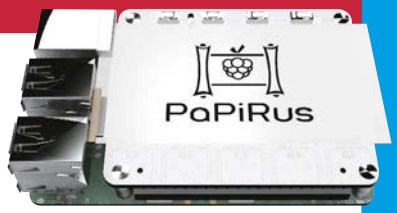
In most cases this is fine, because you'll be busy using the features of the HAT, but if you're looking to create a complex project that combines the features of a HAT with other circuitry, or even combine multiple HATs, then this special device is what you need. Black HAT Hack3r lets you access all of the pins on your Raspberry Pi while also running a HAT.



£10 magpi.cc/10AN36w

PAPIRUS

Add a low power e-ink display to your Raspberry Pi

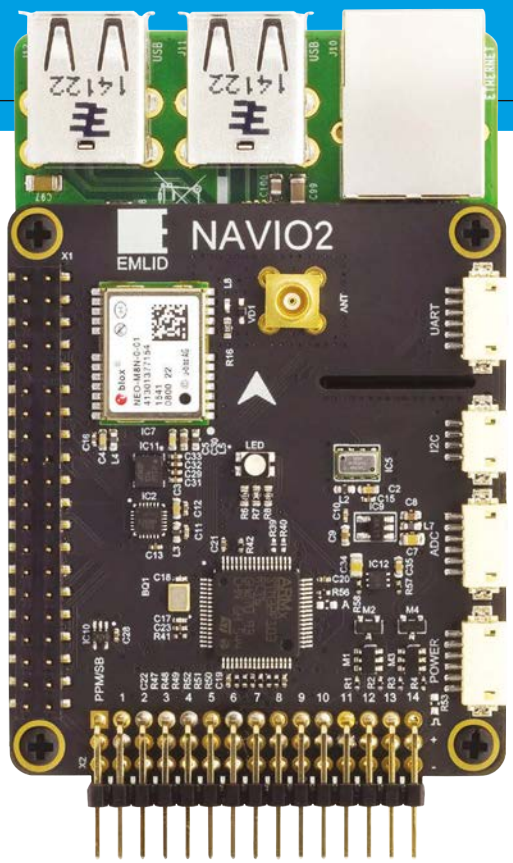


PaPiRus is a brand new HAT that offers a unique and intriguing function. The HAT contains a small electronic-ink screen, the kind you find on e-book readers such as the Amazon Kindle.

The screen is small (three options are available: 1.44″, 2.0″, and 2.7″) so you won't be reading books on it, but it's incredibly low-powered and works in direct sunlight.

"E-paper displays and single board computers are a match made in heaven," says Aaron Shaw, the creator of PaPiRus, "because together they use a very small amount of power but still bring extremely capable displays to your project."

£30-£65 magpi.cc/10A0bXV



NAVIO2

Take your Pi to the sky with this flight control board

Drones are one of the most interesting gadgets to build, but they're surprisingly tricky to get up into the air safely.

On the one hand, the Raspberry Pi is the ideal platform for drone building. It's small, weighs very little, draws low power, and it's easy to add WiFi and LTE dongles for communication, along with cameras and mics for recording.

On the other hand, the Raspberry Pi isn't a natural pilot. It needs a lot of additional hardware to know where it is and manage the motors, and even with the right hardware you need flight control software precisely tuned to keep your drone in the air.

Navio2 is a Linux autopilot to help you keep your drone in the air, or on the ground: it also works for cars, boats, and subs. Most people use it with flying devices, and community projects include blimps, airplanes, and lots and lots of quadcopters.

The HAT features an APM (ArduPilot Mega) flight stack. This stack provides different flight modes, such as manual, stabilize, follow me, and auto. The HAT also has gyroscope, accelerometer, magnetometer, and barometer sensors, plus GPS.

You still have plenty of freedom to build your drone: DF13 ports with ADC, I²C, and UART interfaces are available for connecting sensors and radios. Open-source code and tutorials are available in C and Python.

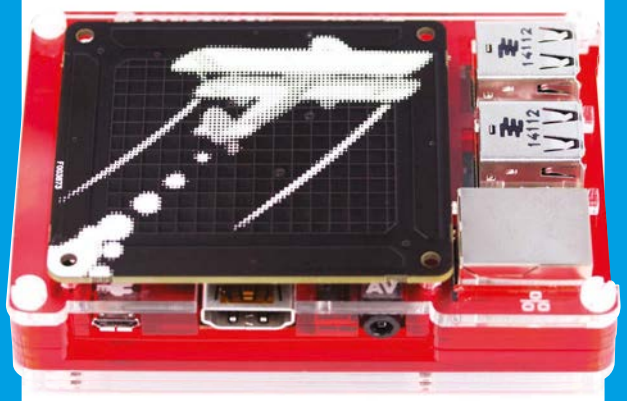
Neat touches like a triple-redundant power supply give the Navio a professional edge, and while UAV building remains a challenge, this HAT makes it more achievable.

$168 (£117) emlid.com



SKYWRITER HAT

Control your Raspberry Pi with a wave of your hand, using the Skywriter



The Skywriter HAT brings mid-air gesture control to your Raspberry Pi, so you can control your Pi (and other devices) by waving your hands around in the air. It's the closest you're likely to come to using The Force for real.

Skywriter HAT creates an electrical near-field just above the Raspberry Pi that detects hand movement. It isn't just a wave detection kit, though: the Skywriter HAT reads 3D positional data (x, y, z) and determines the exact position of your hand above it. This enables it to detect gestures like swipes and taps in mid-air.

Many people think that wave gestures are a future interface for computing, and it's a good idea to get a head start in this area of technology. But the Skywriter HAT is also great fun to use. For instance, you could try setting up the Pi to say "These are not the droids you're looking for" with a wave of your hand, or create musical instruments that respond to air gestures. One user has even combined the Black HAT Hack3r with a Skywriter HAT and Unicorn HAT to create light shows that respond to hand gestures.

This is a great example of a HAT. It's easy to set up, great fun to program, and gives the Raspberry Pi features you wouldn't think possible.

£16 magpi.cc/10APeHb

magpi.cc/1RPbTpG

£7 / $10



Maker Says

"Touch-sensitive screen gives you finger-tip control"

SB Components

LCD CONTROL CASE

A simple case for mini-LCD Raspberry Pi screens – how does it hold up under proper use?

Last issue we reviewed the SPI-Box from SB Components, a custom case for the Raspberry Pi 2 (and B+) that allows for a motion sensor and a Pi Camera Module to poke out of the case. With the bundled software, this means you have a functional and cheap security camera, with plenty of software bells and whistles to make it genuinely useful. Using a similar design ethos, SB Components has also created a special case for using a 3.2″ LCD touchscreen with the Pi in a proper enclosure. It's very inexpensive as well.

The case comes in three parts: a base that's near enough identical to the SPI-Box's design, along with an interstitial layer to protect some of the Pi components from the screen, and finally a top part that fits snugly over the Pi and screen. Like the SPI-Box, the base has the Pi snapped very firmly in place; possibly a little too firmly, as removing it again can be a little bit of an undertaking.

Putting an LCD screen over the interstitial layer and then enclosing it with the top is very easy – a lot of Pi mini TFT screens only use the first 26 GPIO pins, a legacy from the original Raspberry Pi design. The interstitial layer only opens up these pins, meaning there's no confusion. The entire screen is then accessible through the case, and even the usual array of microswitches (like on the Adafruit screen) can be accessed via little pinprick holes. Think a reset switch on an old piece of electronics – just try to avoid using something sharp.

With the case alone, you have an excellent addition to a Raspberry Pi you're using with a touch LCD screen. It's sturdy, very easy to construct, and microswitch access is quite neat as well – normally those sort of switches are forgotten in these cases. Crucially, there's also more than enough room to use the full screen. In addition, it has mounts to fix to the wall, much like the SPI-Box does. As it technically only needs power and maybe a wireless dongle, it can be a great portable Pi or a cheap home automation controller on the wall – the only thing really hampering that is the depth of the full unit, although that's not the case's fault.

If you like the idea of the screen and case but are baulking at the idea of buying a screen, assembling it, and then making the whole setup work on your Raspberry Pi, then SB Components also offers a bundle kit (**magpi.cc/1OAFFYY**). At only £29.99 ($43) on top of the Raspberry Pi, you basically get the kit to create a portable Pi for just over £50 in total.

Related

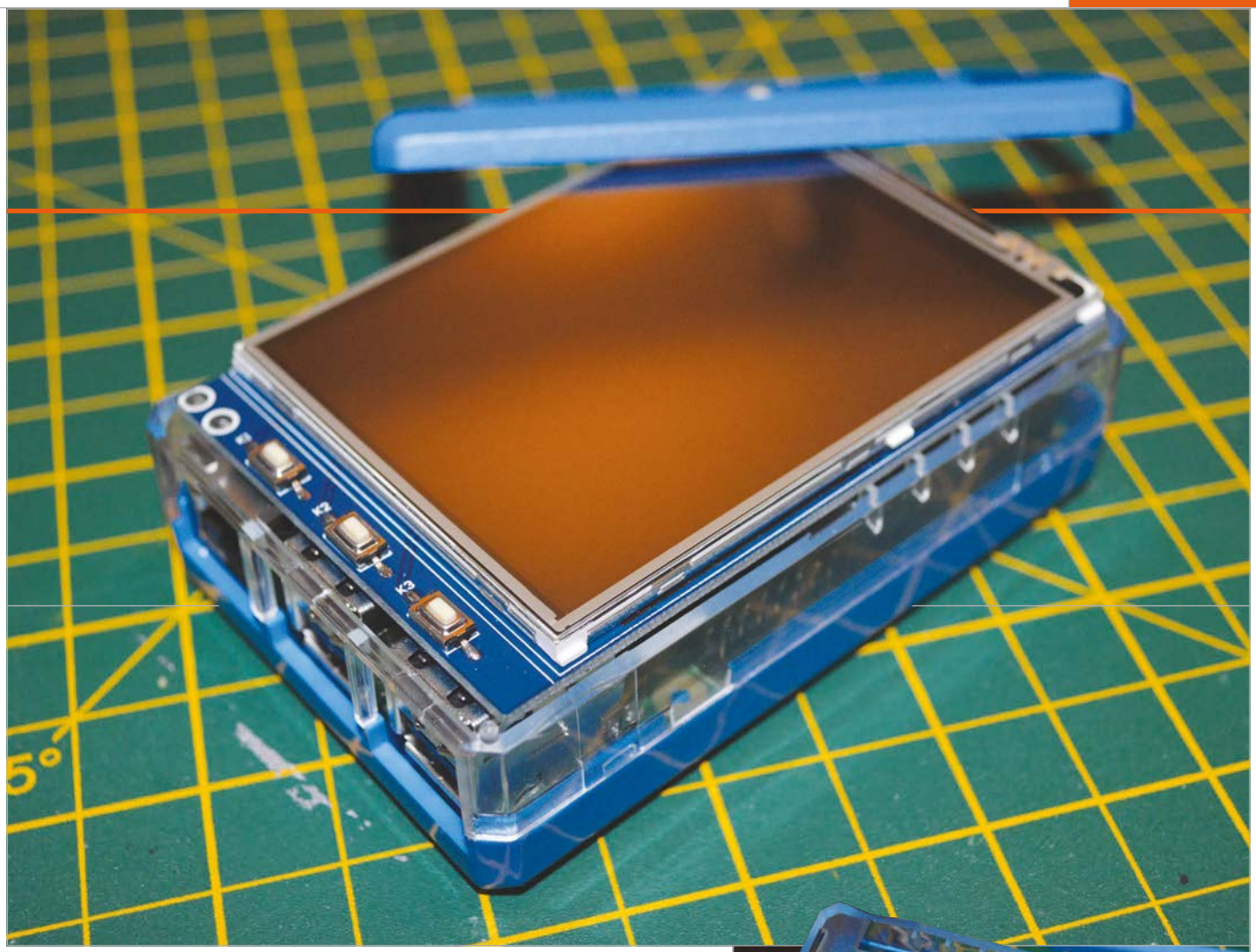
PIBOW PITFT+

The updated PiTFT Pibow allows for the slightly larger second version of Adafruit's PiTFT screen on the Pi Model 2 and B+.



£13.50 / $19.95

magpi.cc/10AI31M



The bundle comes with a compatible mini touchscreen that works with the Pi and the case, a power supply, a WiFi dongle, and a pre-prepared SD card. As it's the exact same case, construction is the same; the real treat, however, is the SD card.

> "The bundle comes with a compatible mini-touchscreen that works with the Pi"

In the past, preparing Raspbian to work with these touchscreens was not a quick process. It's slightly easier now, thanks to more readily available images. Even so, it's nice to have one that just works out of the box. It still runs modern Raspbian, which includes the interface update from just over a year ago. It's a little tricky to use on this size of touchscreen, but it's not completely unusable. At the very least, it leaves it open for you to create your own custom interface and it can also access the SPI-Box, which is a nice little addition.

Both versions of the LCD case are great value for money, especially as they fill a slightly niche market for people using or wanting to use this type of screen. It's easy to modify the microswitch holes to be slightly more usable as well, so you can improve it with the right know-how. Whether you want the beginnings of a portable Pi or a touchscreen remote, you can't go far wrong with this.

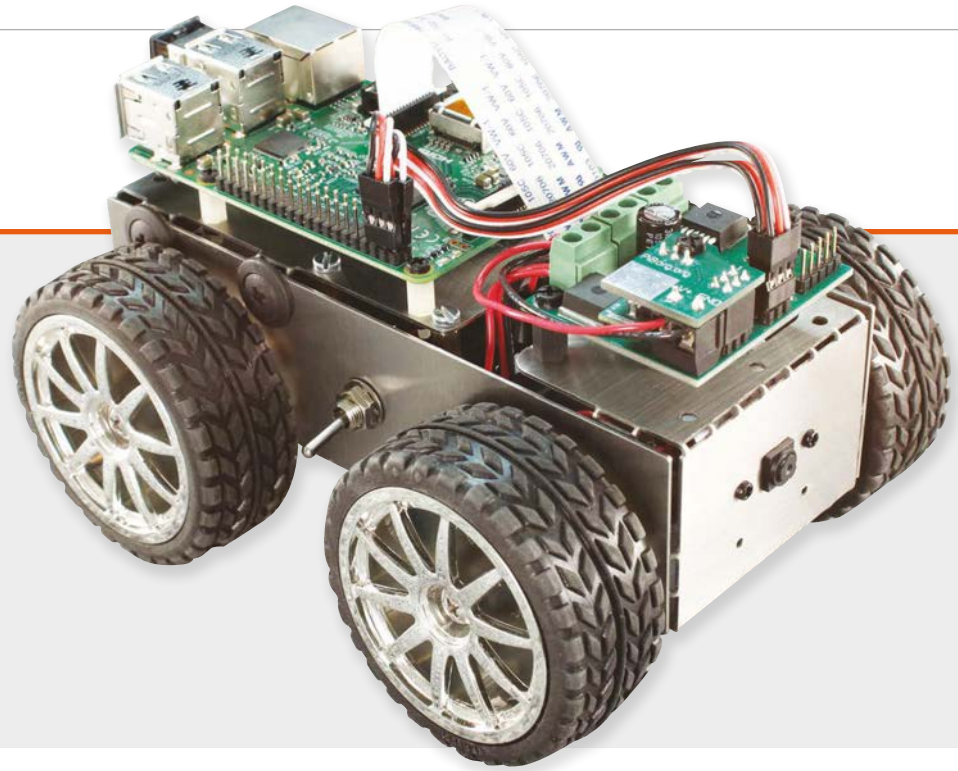


Above Much of the Pi is accessible with the screen off, although it concentrates on the first 26 pins of the GPIO

Last word

A sturdy and well-presented box that is as good as any other for using one of the small touchscreen displays, opening up a world of possibilities for other projects and uses.

★★★★☆



Maker Says

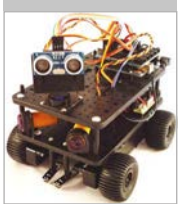
"An armour-plated 4×4 robot for only £99!"
PiBorg

4BORG

An affordable robot kit from the robotic masters over at PiBorg, this is both cool and educational

Related

ULTIMATE 4TRONIX INITIO 4WD ROBOT KIT
A similarly expensive robot, although this one comes with a whole host of sensors you can use, albeit without a sturdy metallic chassis.



£84 / $119
magpi.cc/1JjNmG3

We're very excited about this one. For a long time, any other Raspberry Pi robot has seemed to pale in comparison with anything put together by the PiBorg team. It's never really been a fair comparison, of course; a lot of Pi robots are made for those on a budget, and so focus on making a first foray into robotics inexpensive. Not all the PiBorg robots are kits to begin with, so when the monstrous DoodleBorg is racing around in videos, it's a wonderful spectacle, but not many people are actually going to be able to build it.

The main kit that PiBorg usually sells is the DiddyBorg, a six-wheeled robot retailing at around £200 that's a very serious bit of kit compared with the budget ones. It's still expensive in comparison, which is seemingly where the 4Borg comes in. Introduced as 'DiddyBorg's smaller brother', the 4Borg is a four-wheeled robot that strips the PiBorg design ethos down to its essentials: four motors, four wheels, a motorboard to control them, and a metal chassis to hold it all together. It's powered by Raspberry Pi and even comes with a rechargeable battery pack like the kind you'd find in an RC car.

Simple but sturdy

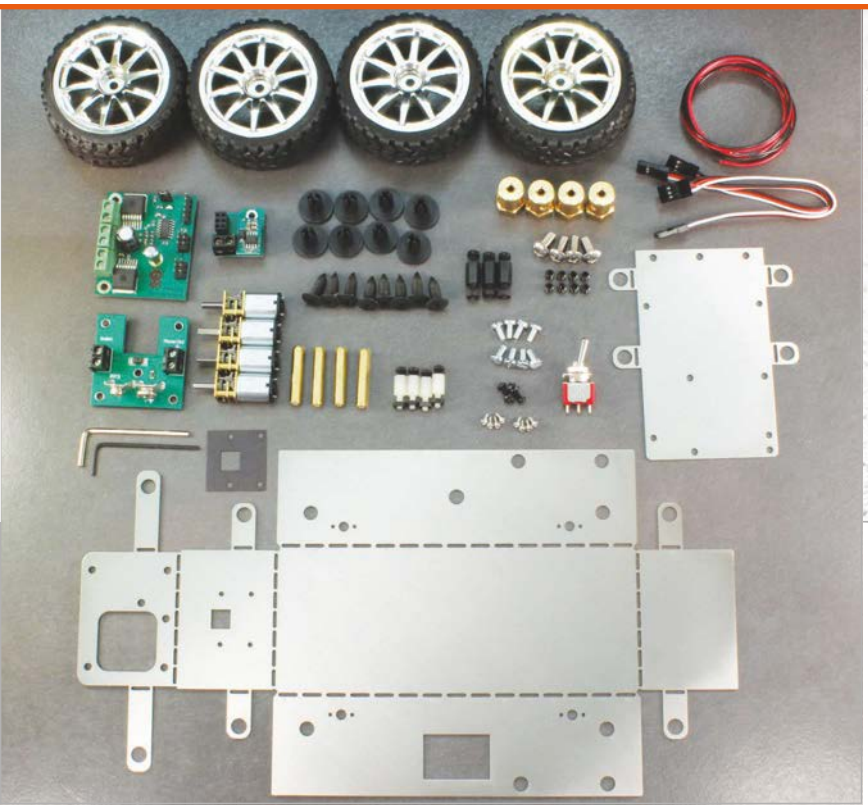
On the surface, then, it does seem like a simple robot – no sensors are included, although there's a mount for a Pi Camera Module in the front of the chassis. Automation for the moment is therefore limited to timed motor bursts, although with a bit of work you should be able to add in some motion and face detection using a camera.

Building it is a joy. The best and possibly most clichéd way to describe it is like a toy for grown-ups. The chassis comes as a thin sheet of pre-cut metal that you need to bend into place to fit together, like a cardboard model. It's extremely sturdy and all the supplied parts feel like they're good-quality gear (because they are). You can spend a few pounds more when buying it to have the motors and an on/off switch pre-soldered; they're definitely not difficult soldering jobs, but if you're still a complete soldering novice it's a small price to pay for confidence in your materials.

As you can see from the images, the Raspberry Pi and motorboard are mounted on top, leaving the insides of the chassis fairly hollow, save for the ends of the motor and a small slot for the battery. Due to

piborg.org/4borg

£99 / $140





Above There's plenty of clearance to add line sensors onto the bottom of the chassis

Left It's really not that many parts – it should take around an hour or so to build

this and its metallic body, you could definitely upgrade the robot down the line by mounting line sensors, or affixing custom brackets for an ultrasonic sensor. Adding them via code shouldn't be too difficult with a bit of research.

Speaking of code, once complete, you can begin using it pretty much straight away after construction. Code examples are available in the resources for the 4Borg online and just require a little setup on the Raspberry Pi before getting them to work; the examples are downloaded and installed during this setup phase, so you don't need to add them manually.

The robot is programmed in Python, making it very compatible with the Raspberry Pi. Instead of using GPIO Zero, it mainly uses its own library called PicoBorgRev. This is for the PicoBorg Reverse, the motorboard controlling the robot. The code examples are quite elaborate, showing off how to perform an automated drive or how to control it via a connected joypad – specifically, a PlayStation 3 controller. You can get a good feel for how you can manipulate the code by studying it. There are a few basic commands, such as driveLeft and driveRight, that control the power going to the left or right motors respectively; you can only control each side rather than each motor with the 4Borg setup. It's a touch more complicated than some other cheaper kits, but the 4Borg is a touch more advanced than those kits anyway.

> "For £99 you're getting a proper PiBorg robot, with the inherent quality that goes with that"

Honestly, the 4Borg is everything we expected. For £99 you're getting a proper PiBorg robot, with the inherent quality that goes with that. It may start off a little simple, but there's plenty of room to grow with this as the platform, or even go bigger and better with another chassis and kit. With a new *Robot Wars* around the corner, maybe this could be your entrance to a wider world of robots...

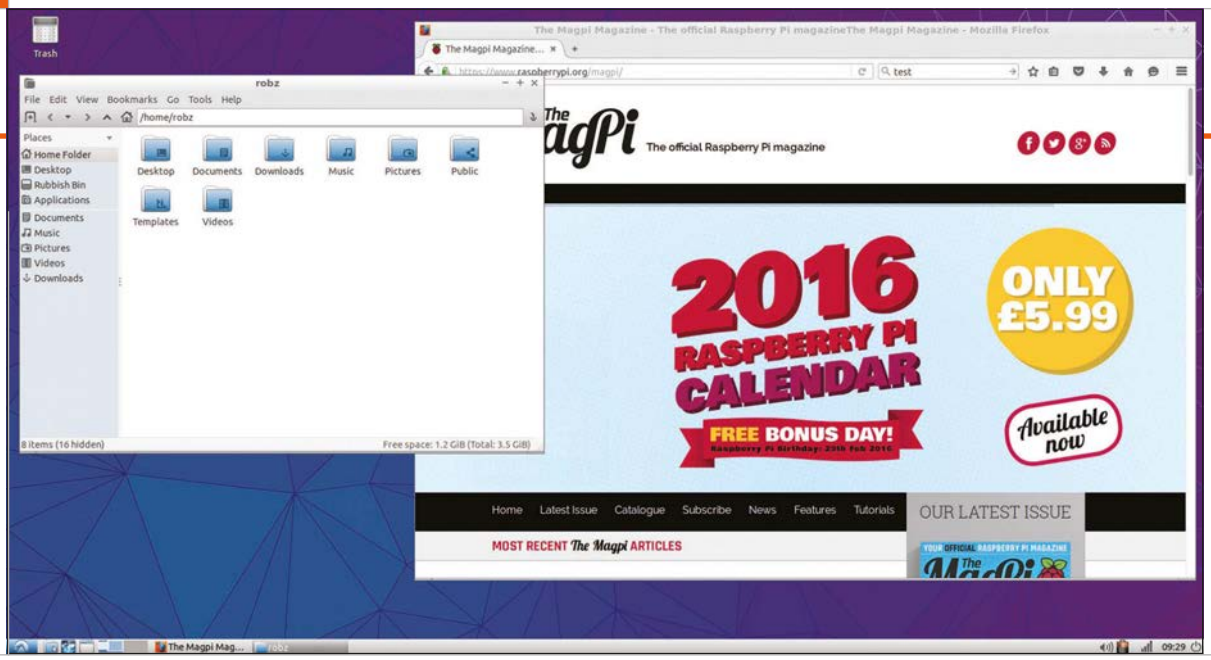
Last word

A quality piece of kit, just what you'd expect from PiBorg, and a fantastic jumping off point for people who really want to get the most from their robotics projects.

★★★★★

Maker Says

> Make as many Ubuntu flavours for the Pi as you can think of
>
> **Ubuntu Pi Flavour Makers**



UBUNTU PI FLAVOUR MAKER

More flavours of Ubuntu for your Raspberry Pi. Does the Pi provide enough power for them, though?

If you've been following Raspberry Pi long enough, you may know that early on in its life there was a call for Ubuntu to be used as an operating system for it. As one of, if not the most popular, version of Linux for home users, this distro was wanted by many seeking something familiar for their new tiny computer. It never came to be. However, as Ubuntu is based on Debian, Raspbian was basically the next best thing.

When the Raspberry Pi 2 came out, this changed with the release of Ubuntu MATE for the Pi, and this opened the door to more 'versions' of Ubuntu – or, at least, different desktop environments running on Ubuntu. Ubuntu Pi Flavour Maker is one of the results of this.

At its core, Flavour Maker is a series of scripts that allow you to build a slightly custom version of Ubuntu for the Raspberry Pi. It uses the all-important armhf build of Ubuntu that's optimised for the type of chip running on the Pi 2, giving you options for different software and desktops, and even the choice between server and desktop versions depending on your needs.

That may sound a little complicated for some more novice users, but don't worry – the kind folk who make the software have created a series of images you can install to an SD card, much like any Raspberry Pi operating system. On tap are two versions of Ubuntu Server: one a very minimal install for maximum speed, and a regular install if you need more software. There are also three 'spins' that use different desktops: LXDE, Xfce, and MATE. LXDE and Xfce are common lightweight desktop interfaces used on low-powered PCs; Raspbian has used LXDE, or at least a version of it, since it came out.

Booting up

How do they actually run, then? Much like the normal Ubuntu, there are a few setup steps on your first boot, although the main installation has already occurred so it's a little truncated. Settings such as locale and keyboard are sorted before booting you into the desktop, although this does take a few minutes to actually occur. Each successive boot is much faster, though, albeit nowhere near the speed of booting into Raspbian.

The OS is presented in a very different way from Raspbian. Whereas much of Raspbian is optimised for the Raspberry Pi, with a lot of educational tools in there as well, the Ubuntu spins

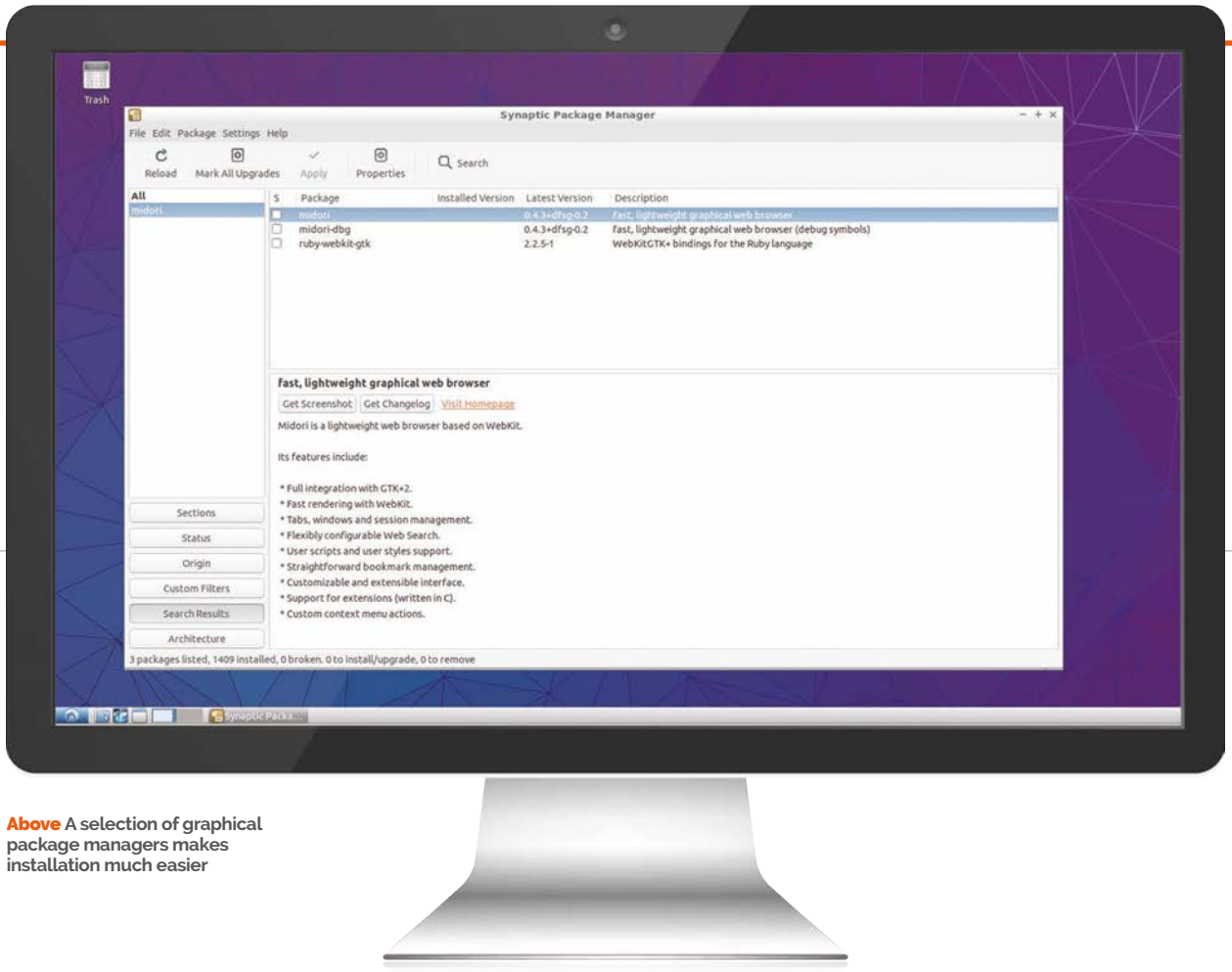
Related

UBUNTU MATE

When the Raspberry Pi 2 came out, Ubuntu was finally able to come to the Pi, starting with a version of Ubuntu MATE.



FREE
ubuntu-mate.org



Above A selection of graphical package managers makes installation much easier

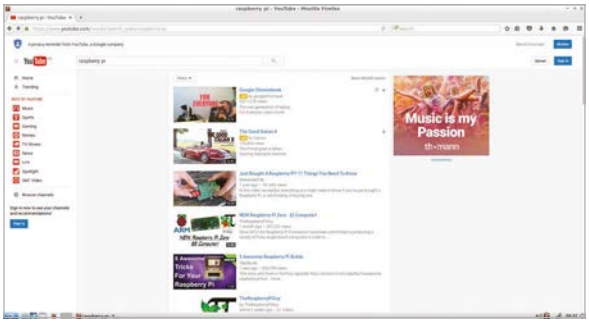
are more a traditional desktop – no programming tools installed as standard, and using normal programs as well.

A novelty compared to Raspbian is the addition of a graphical package manager. Both Software Centres and classic Synaptic are available, giving users a way to search for software they may want, which Raspbian currently doesn't let you do easily. As this is based on Ubuntu armhf, the selection isn't as broad as in the full version of Ubuntu, but it's still very good. A lot of major software is available, with the only real limit being the power and performance of the Raspberry Pi 2 running Ubuntu.

Due to its use of more standard software, Ubuntu can lag a bit on the Raspberry Pi 2. Firefox is the default browser, and while a lightweight alternative like Midori is available, it's not always fast and responsive. It feels like how the browser used to be on the original Raspberry Pi – functional, but a little frustrating to use after a while. Video on YouTube seems right out as well, if you want to watch anything larger than 360p.

It's not all slow, though. With the right selection of packages and the right use case, it can be just fine. Ubuntu is a slightly more resource-heavy base than Debian as it is, so for anything requiring the extra power, it was never going to be the operating system or distro of choice. The Xfce version does run the best out of the three, but Raspbian is still better in terms of performance.

As a desktop system, though, it's better. With a few modifications to the default apps, it's perfectly usable as one, and looks a bit more the part than Raspbian as well. For projects and more niche uses, however, it looks like Raspbian is still king.



Above YouTube is hit-and-miss, especially on a more resource-heavy browser like Firefox

Last word

It was never going to be as quick as Raspbian, but it's a fantastic effort that adds some more variety to the operating systems available for the Raspberry Pi.

★★★☆☆

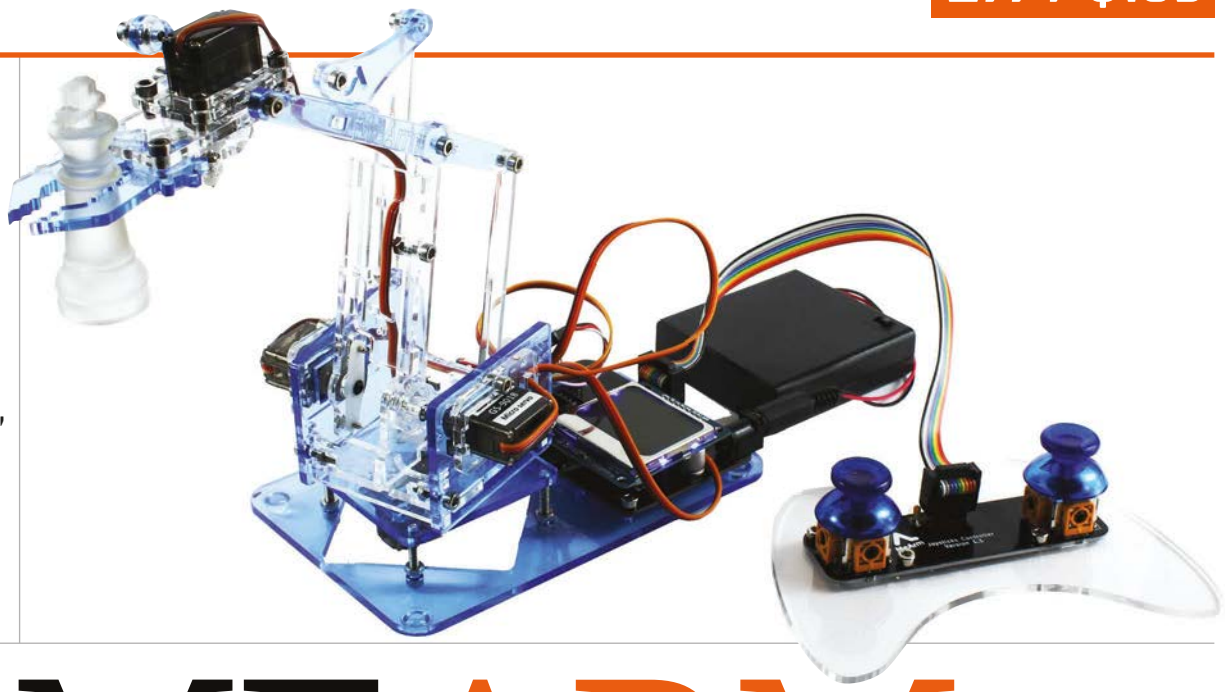
mearm.com

£77 / $109

Maker Says

"The world's most accessible learning tool and smallest, coolest robot arm"

MeArm



MEARM

A robot arm kit designed as an educational tool – how can people learn from MeArm?

As a kid, it was weird coming to the realisation that the only real robots were robot arms. This was only really for a period in the Nineties, though, and probably not true; from the perspective of a child, however, that was how it seemed. Proper automatons, both expensive bipeds and cheaper wheely-trundly types, are now far more common than they were. The MeArm, then, is a bit of a nice break from the now-normal robot types.

It's designed very much as an educational tool, teaching people about electronics, robotics, and some programming as well if they're interested. Construction is the big first step in the process and it's definitely not a quick one; while it's not an overly complex build, there are a lot of parts to it and a bit of calibrating to do before you even start. The process is generally simple, though; as long as you stay aware of the screws you need and double-check which way some of the pieces are orientated, it's straightforward enough. It's a bit like building Meccano, and it can be quite fun.

The whole kit comes with everything you need to run it, including its Arduino-based 'brains'. As you may expect, this means the MeArm is programmed in Arduino; an example sketch is available that allows you to control the MeArm using a slimline, acrylic controller with analogue sticks. The code has plenty of unused sections that allow you to try to add outputs to an included display – salvaged from old Nokia phones, no less. With enough playing and poking, you could feasibly get the robot arm to move on its own as well, albeit only with timed movements as it doesn't have any extra sensors out of the box.

Generally, though, it's a lot of fun. With a build that will keep any techie kids entertained and the instant satisfaction of being able to use the arm with the included controller, there's a lot of flexibility and longevity with an arm like this. Don't be put off it if you're an adult, either: there's definitely plenty that grown-ups can learn from and enjoy too.

Last word

It's a little expensive, but an amazing package that can be programmed with your own initiative and ideas. Perfect for older kids and adults alike.

★★★★☆

Related

BUILD YOUR OWN – ROBOT ARM

A cheaper, USB-powered robot arm that offers far fewer modification options than the MeArm, but does have a similar controller.



£40 / $68

magpi.cc/1JjNJk4

Raspberry Pi SWAG STORE



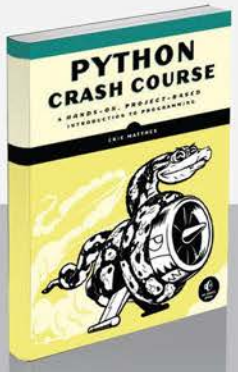
swag.raspberrypi.org

RASPBERRY PI BESTSELLERS NO STARCH

Find out why New Starch's recent Python books are flying off the shelves [Spoiler alert: they're all great fun!]...

PYTHON CRASH COURSE

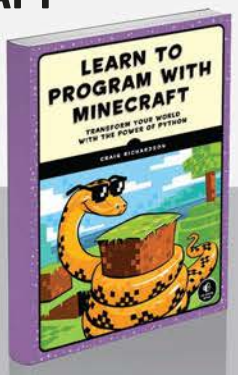
Author: Eric Matthes
Publisher: No Starch
Price: £26.50
ISBN: 978-1593276034
magpi.cc/1NfOBAS



Practical and involving introduction to Python programming, with great projects covering not just games, but also web applications and data visualisation, using Django and Python's excellent libraries. Five out of five, recommended.

LEARN TO PROGRAM WITH MINECRAFT

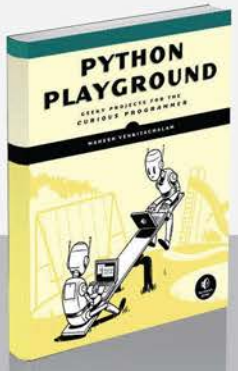
Author: Craig Richardson
Publisher: No Starch
Price: £25.78
ISBN: 978-1593276706
magpi.cc/1NfOHsk



If your offspring are crazy about *Minecraft*, Python will speed up their resource mining, while opening up a whole new world within the game – and beyond. Fun and learning are seamlessly integrated in this book.

PYTHON PLAYGROUND

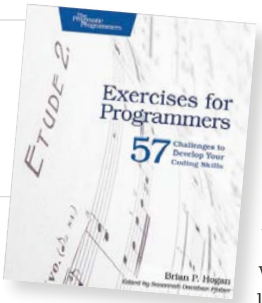
Author: Mahesh Venkitachalam
Publisher: No Starch
Price: £19.99
ISBN: 978-1593276041
magpi.cc/1NfOLIK



A fun, project-based book of Python, with plenty of art and graphics [Spirograph!], for all but absolute beginners. Concludes with a Raspberry Pi weather monitor, using the Bottle web framework.

EXERCISES FOR PROGRAMMERS

Author: Brian P Hogan
Publisher: Pragmatic
Price: £15.99
ISBN: 978-1680501223
magpi.cc/1NfQ3Dz



Most of the coding books we review are to be used with a single language, but here's one that will help you with any language that you use. This book is a collection of exercises in the form of problems to solve, from the simple – character counting and sharing out pizza slices – to full programs.

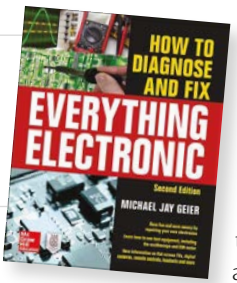
Along the way, solving the problems takes in decision making, functions, data structures, working with files, and dealing with external services through APIs. One problem in this latter section uses the Open Notify API to see how many astronauts are currently in space and who they are, so you can check Tim Peake is still on the ISS with the Astro Pi boards!

Practising coding exercises keeps your skills sharp: the parallel is with the exercises performed by musicians, or katas worked through by karate experts. More than that, most tutorial books for individual languages have to guide you through every step of learning – but running through the exercises here will enable you to learn to solve problems for yourself in a language, giving you a deeper understanding and appreciation. Extra challenges for each exercise push you further. Unhesitatingly recommended.

Score ★★★★★

HOW TO DIAGNOSE & FIX EVERYTHING ELECTRONIC

Author: Michael Jay Geier
Publisher: TAB Electronics
Price: £18.99
ISBN: 978-0071848299
magpi.cc/1NfQDRx



Inventor, composer, and 'renaissance man' Michael Jay Geier was operating an electronics repair service for neighbours by the age of eight. He brings passion, wisdom, insight and a fair degree of common sense to all practical aspects of repairing everyday electronic gear. Not just a good end in itself – particularly from an environmental perspective – but a great boost to your maker skills, for diagnosing the mysterious fault in your latest Raspberry Pi add-on.

Topics covered by the book include safety, philosophy, test equipment, and components, along with all the techniques, tips and tricks that you'll need. Circuit diagrams are covered, as well as identifying components and features when looking inside a product. There are plenty of tips on finding faults and replacing components, preceded by tips on how to get inside today's 'sealed' boxes, and followed by useful help on reassembly.

The large final chapter on problems particular to different types of products is where this book could really earn back its purchase cost – from fixing VCRs so that you can watch treasured family tapes again, to repairing video projectors and flat-screen televisions, which are often thrown out due to a tiny component failure.

Score ★★★★½

BUILDING A HOME SECURITY SYSTEM WITH RASPBERRY PI

Author: Matthew Poole
Publisher: Packt
Price: £22.99
ISBN: 978-1782175278
magpi.cc/1NfQVli



Building a home security system with the Raspberry Pi teaches many skills. For example, Poole starts with connecting things to the GPIO – a push switch, an LED, and a handful of components – to start you off on using sensors; by the time you've got to the end of the chapter, you'll have an understanding of shell scripts, the command-line interface to the GPIO, and working with circuits and electronics. In other words, this is a great 'dive straight in' introduction for beginners, thanks to clear explanations, as well as a useful guide for anyone looking at building home security systems.

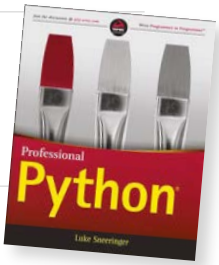
Next, 'Extending Your Pi to Connect More Things' moves beyond the eight-pin limit of the GPIO with an I²C-based port expander, and looks at serial-to-parallel and parallel-to-serial conversions. By now you'll have a shopping list of components, but they're all available from the usual Pi-friendly mail order companies, which also sell the ready-made alternatives discussed in the chapter.

From adding sensors (wired and wireless), through driving inductive loads, to building a web-based control panel, this is a comprehensive guide to all of the paraphernalia of home security systems, as well as a relatively easy way to learn interconnected Pi skills in software, hardware, and web connectivity.

Score ★★★★★

PROFESSIONAL PYTHON

Author: Luke Sneeringer
Publisher: Wrox
Price: £33.99
ISBN: 978-1119070856
magpi.cc/1NfRBxa



There are other intermediate Python books which show you how to use Python's special, powerful features, but Sneeringer's book doesn't duplicate titles which tell you everything there is to know; instead, it takes key powerful features, grouped by sensible examples, then tells you what you need to know about them and shows you how to use them.

The first section, on Functions, covers the essential topics of generators, decorators, and context managers, which you'll soon find you cannot live without. The Classes section includes magic methods, metaclasses, and abstract base classes, the latter being useful for where duck typing is too hazardous and you want to assign identity. The Data section covers strings and Unicode, then moves onto regular expressions (and when not to use them) – a useful primer.

The last section groups together many miscellaneous topics of use, starting with differences between Python 2 and Python 3, and how to write code that will run in both. Unit testing gets a useful run-through, then Asyncio, and rounding off with important notes on style. All in all, a good second Python book if you're beginning to spend more time with the language.

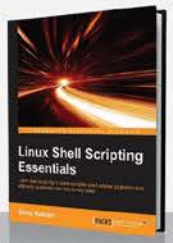
Score ★★★★☆

ESSENTIAL READING: SHELL SCRIPTING ESSENTIALS

While you shouldn't neglect the classics, there's a whole new generation of shell scripting books to investigate...

Linux Shell Scripting Essentials

Author: Sinny Kumari
Publisher: Packt
Price: £31.99
ISBN: 978-1785284441
magpi.cc/1NfSjuk



Powerful and comprehensive – yet reasonably concise – beginner-friendly Bash scripting tutorial. Full of shell wisdom.

Pro Bash Programming

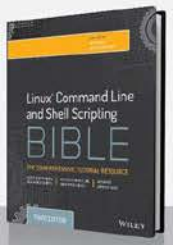
Author: Chris Johnson
Publisher: APress
Price: £29.50
ISBN: 978-1484201220
magpi.cc/1NfTsC7



Despite a few oddities, there's much to learn from Johnson's passionately 'Bash for absolutely everything' approach to scripting.

Linux Command Line and Shell Scripting Bible

Authors: Richard Blum & Christine Bresnahan
Publisher: Wiley
Price: £33.99
ISBN: 978-1118983843
magpi.cc/1NfSvtu



Comprehensive guide full of useful insight into the what, why, how, and when of scripting.

Mastering Linux Shell Scripting

Author: Andrew Mallett
Publisher: Packt
Price: £22.99
ISBN: 978-1784396978
magpi.cc/1NfSFkM



Very practical tutorial, taking in Apache vhost creation, an Awk tutorial with useful logfile analysis, and some Perl and Python.

Text Processing with Ruby

Author: Rob Miller
Publisher: Pragmatic
Price: £23.99
ISBN: 978-1680500707
magpi.cc/1NfSRAp



Sometimes the ubiquitous Bash is not enough. Using Ruby, Miller goes from shell one-liners to sophisticated text manipulation.

THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi

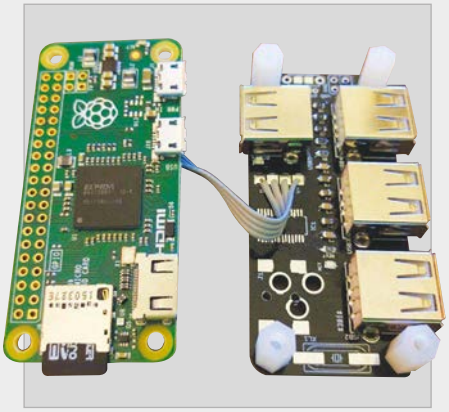
MORE RASPBERRY PI ZERO PROJECTS

Every day, people are surprising us with what they can do with a Raspberry Pi Zero. We knew it was going to be used in some fantastic projects, but we honestly didn't expect this many great things so soon. Here's some of the awesome things we've seen this month...

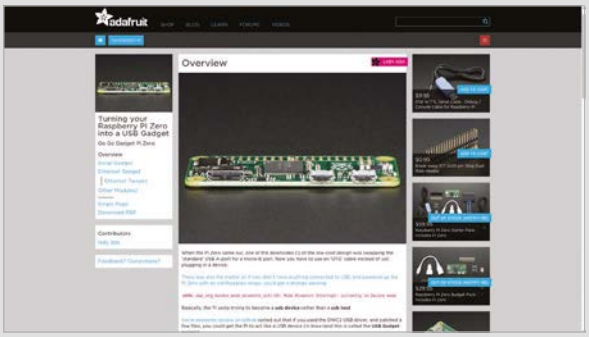
PI ZERO-POWERED RECIPE KIOSK

It's something we still haven't quite got a universal solution to, even in 2016, but the electronic recipe book is something a lot of people seem to want. Reddit user zarderxio has taken an old car DVD player and attached the Pi Zero's video out. It's mounted inside the case and, with the help of a WiFi adaptor and a wireless keyboard and mouse, is now a fully fledged kitchen recipe computer. Find out exactly what he did on his image album: **magpi.cc/1niNuvO**

USB ZERO HUB



To keep the Zero small, some sacrifices had to be made in the ports. Notably, the micro USB for data is quite a change, especially compared to the four USBs on a B+ or Pi 2. Richard Hawthorn has managed to create a mod that adds four USB A ports to a Pi Zero, with only a slight increase to its height. Find out how to make your own on his site: **magpi.cc/1niNgF8**

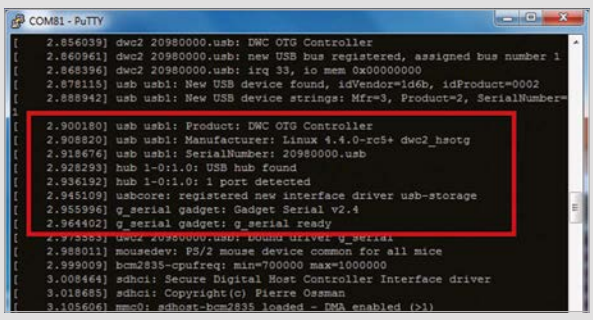


PI ZERO USB GADGET

When we spoke to Mike Stimson for issue 40, one of the things he pointed out was that with a bit of hacking, you could use the Raspberry Pi Zero as a USB device. His example was something like a USB storage device. Apparently, other people figured this out after launch and Adafruit had quickly put up a tutorial (**magpi.cc/20bTODh**) on how to turn your Pi Zero into a USB gadget.

Although there's a lot more you can do with the Zero in terms of making it a gadget, the Arduino tutorial concentrates on serial and Ethernet varieties. Serial gadgets are the simplest, and this method allows another computer to both power and talk to the Raspberry Pi via the USB port. This doesn't have many uses, but it's still seriously cool.

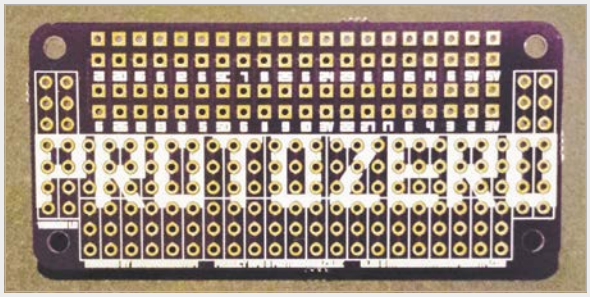
The Ethernet gadget is a little different, allowing you to access the Pi via networking protocols. You can even share a computer's internet to the Pi connected to it as an Ethernet gadget. It's a little more complicated and also has limited use cases, but it's a nice experiment.



Above You're essentially tricking the Pi Zero into acting as different USB devices

CROWDFUND THIS!

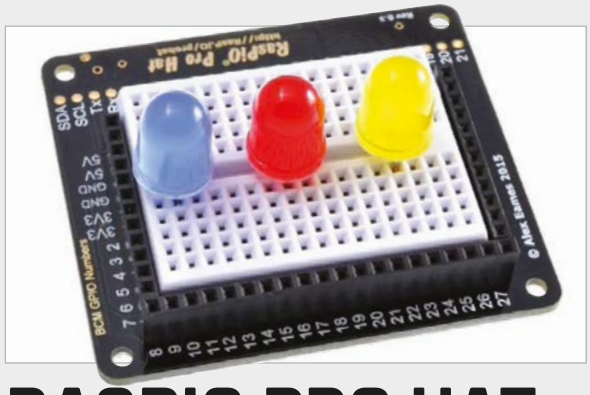
The best crowdfunding hits this month that you should check out...



PROTOZERO

kck.st/1NBfQcO

The ProtoZero is a prototyping board for the Raspberry Pi Zero. A Kickstarter from Average Man vs Pi, it's actually almost over at the time of writing, due to a short campaign. The good news is that it's already hit eight times its funding goal, so you'll be able to get one when it comes out. It hooks up to the Pi Zero GPIO and uses a similar configuration as a breadboard to allow you to prototype circuits and HATs – you can even solder all your components onto it and treat it as a HAT. At £5 they're pretty cheap, so good for making quick and cheap HATs for your £4 computer.



RASPIO PRO HAT

kck.st/1WdN3hM

A breadboard that connects to the Raspberry Pi's GPIO ports and labels them in a more useful fashion (i.e. by GPIO number and not by pin number), the Pro HAT is the new HAT from Raspi.TV and RasPiO. It works right out of the box, mounted on top of the Pi's GPIO pins, which means it can work instantly with the GPIO Zero Python library as well. At the time of writing, the goal has been hit, but there'll still be time to pledge for one before the deadline. It should be a fantastic little educational kit for people with a Raspberry Pi.



LIV PI STARTER

igg.me/at/rpi/x

A home air quality kit for Raspberry Pi, it will enable you to keep an eye on how good the air is inside your home. Not only that, but it monitors temperature, humidity, and air pressure and lets you know how it works. It comes with a nice acrylic box to keep it all in as well! For a little bit extra, it will also keep an eye on CO_2 levels for you. Go have a look and see if it's the kind of thing you want in your home!

BEST OF THE REST

Here's some other great projects we saw this month

SUPERGAMEGIRL INSTRUCTIONS

magpi.cc/1NiHXtJ



A few issues ago, we talked about the SuperGameGirl – it's a fun little project that uses a 3D printed, modified Game Boy case and powers it with a Raspberry Pi to create a portable emulation machine. Its inventor has now put out a full set of instructions on how to do it yourself, after spending some time making sure it could be followed. Give it a look!

SELF-DRIVING (RC) CAR

magpi.cc/1niPfcz



People have made automatons in terms of robots with the Raspberry Pi, but never a fully fledged self-driving car... albeit one on the small side. The technique is roughly the same, though: ultrasonic distance sensors, a camera, and some code to control the car. Read up on how Zheng Wang made one on his blog via the link.



RASPBERRY JAM EVENT CALENDAR

Find out what community-organised, Raspberry Pi-themed events are happening near you...

PUT YOUR EVENT ON THE MAP

Want to add your get-together? List it here:
raspberrypi.org/jam/add

1 HUDDERSFIELD RASPBERRY JAM

When: Saturday 6 February
Where: Huddersfield Library, Huddersfield, UK
magpi.cc/1Q9mZlx
The first Huddersfield Jam of 2016, where people can come along and learn about the Raspberry Pi.

2 JERSEY RASPBERRY JAM

When: Saturday 6 February
Where: Jersey Public Library, St Helier, Jersey
jerseyraspberryjam.com
A place for people to learn how to use a Raspberry Pi, or show off something cool they've made.

3 KYIV RASPBERRY PI / ARDUINO MEETUP

When: Thursday 11 February
Where: 35 Yamaska St, Kiev, Ukraine
magpi.cc/1SYFvjp
A Ukrainian Raspberry Jam in Kiev, for people who want to find out more about Pi outside the UK.

4 CHELMSFORD RASPBERRY JAM #2

When: Saturday 13 February
Where: Chelmsford Library, Chelmsford, UK
magpi.cc/1Q9pK6z
With activities for all ages and abilities, this Jam is great for schoolkids and beginners.

5 CORNWALL TECH JAM

When: Saturday 13 February
Where: Pool Innovation Centre, Redruth, UK
magpi.cc/1SYIsRo
A Jam for anyone interested in computing a technology, run by volunteers from education and IT sectors.

6 COVENTRY AND WARWICKSHIRE RASPBERRY JAM

When: Saturday 13 February
Where: Coventry MakerSpace, Coventry, UK
magpi.cc/1Q9qc4E
A three-hour Jam focused on sharing and exchanging projects to spark people's tech creativity.



7 TORBAY TECH JAM

When: Saturday 13 February
Where: Paignton Library and Information Centre, Paignton, UK
torbaytechjam.org.uk
A fun, informal, and family-friendly event that hopes to inspire people to take an interest in coding and STEM.

8 RASPBERRY JAM IPSWICH

When: Saturday 27 February
Where: Waterfront Building, University of Ipswich, Ipswich, UK
ipswichraspberryjam.co.uk
A volunteer-run event for people interested in learning about the Pi, showcasing what can be done.

DON'T MISS: JERSEY RASPBERRY JAM

When: Saturday 6 February **Where:** Jersey Public Library, St Helier

An informal gathering, bringing Raspberry Pi users together. There's no agenda, and you can drop in for as long as you want – there'll even be some snacks waiting. There will be several machines set up to demonstrate ideas suitable for all skill and experience levels, and a few hardware projects on display. If you haven't set your Pi up yet, another user will be glad to show you how. For more details, visit **jerseyraspberryjam.com**



WANT TO GET NOTICED?

The MagPi

REACH A MASSIVE AUDIENCE FAST

- World's #1 Pi magazine
- 100,000+ monthly readers
- Available on iOS & Android

The MagPi is the most exciting technology mag on the market today, boasting one of the largest & most engaged audiences in tech.



FOR MORE INFO EMAIL: **russell@raspberrypi.org**

Add a blueberry to your Raspberry Pi

Turn your Raspberry Pi into a Multi-Purpose IoT Controller Board

The Blueberry IoT Controller for the Raspberry Pi is a versatile device designed to interact with the real world by reading sensors, controlling electrical equipment and using popular communication channels.

It is supported by a range of other devices which can turn your Raspberry Pi into the nerve centre of a simple or complex monitoring and control system.



CONTROL

gates, garage doors, air-conditioning, irrigation systems and much more.

Use the power and connectivity of the Pi or the on-board GSM to control your system from anywhere in the world or receive reports and alerts via email or texts (SMS).

Rapid electronics

www.rapidonline.com

Rapid Electronics part of the CONRAD Group

Made in the UK

IQaudIO

Audiophile accessories for the Raspberry Pi



Pi-DAC+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments PCM5122
- Variable output to 2.1v RMS
- Headphone Amplifier / 3.5mm socket
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Access to Raspberry Pi GPIO
- Connect to your own Hi-Fi's line-in/aux
- Industry standard Phono (RCA) sockets
- Supports the Pi-AMP+



Pi-AMP+

- Pi-DAC+ accessory, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TPA3118
- Up to 2x35w of stereo amplification
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- Auto-Mute when using Pi-DAC+ headphones
- Input voltage 12-19v
- Supports speakers from 4-8ohm



Pi-DigiAMP+

- Raspberry Pi HAT, no soldering required
- Full-HD Audio (up to 24bit/192MHz)
- Texas Instruments TAS5756M
- Up to 2x35w of stereo amplification
- Out-of-the-box Raspbian support
- Integrated hardware volume control
- Provides power to the Raspberry Pi
- Software mute on GPIO22
- I/O (i2c, 3v, 5v, 0v, GPIO22/23/24/25)
- Just add speakers for a complete Hi-Fi
- Input voltage 12-19v
- Supports speakers from 4-8ohm

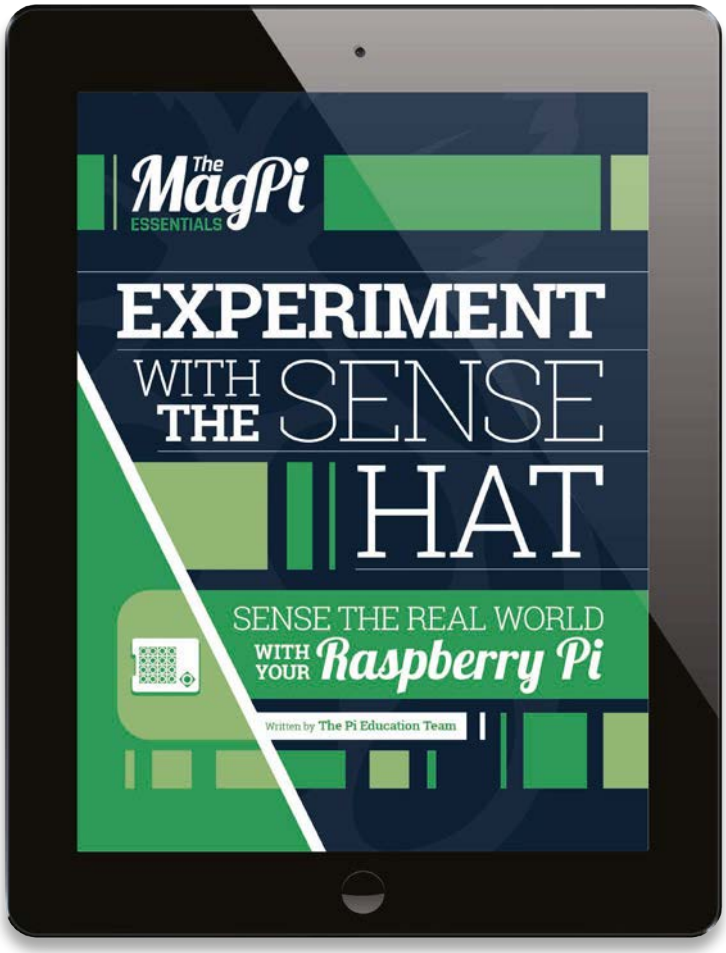
M2P MAX2PLAY · rune audio · VOLUMIO · moOde audio player · PiMusicBox · openelec · piCorePlayer SQUEEZE AUDIO · Archphile

Twitter: @IQ_audio
Email: info@iqaudio.com

WWW.IQAUDIO.COM

IQaudio Limited,
Swindon, Wiltshire.
Company No.: 9461908

YOUR LETTERS



Above We'll be doing more Essentials books on a variety of different subjects

Essential reading

When you first released *The MagPi Essentials* series, I thought you would just be covering the same sort of topics, like the command line. It didn't even occur to me they might all be different, even when the Python one came out. I guess I thought they'd be parallel series to showcase different parts of the command line and Python. With the release of the Sense HAT book, though, I'm happy to see it's a cool mixture of things.

Will you ever go back to stuff like Python or the command line?

Timothy N

The plan is to have a great selection of essential books for a lot of different subjects, so that anyone can pick up one and do something fun or educational. Due to how important and widespread Python is, we'll absolutely be doing more Python Essentials down the line, so watch this space. Or at least watch our online space, where you can more readily see what we're putting out!

Zero Zeros

I'm having a tough time tracking down a Raspberry Pi Zero. I know in your article about the development of the Raspberry Pi Zero that Eben [Upton] said they were doing only so many to start with – are there going to be any more on the way, so more people can get one? I do already have one from the magazine, but I have many project ideas I want to use them with. Thanks.

Alice C

Demand for the Raspberry Pi Zero has been unprecedented – boards are being made as fast as they can, but it's apparently not quite enough to keep up with demand. For the time being, it's best to keep an eye on the various social channels for the Raspberry Pi, such as Twitter and Facebook, so that you can get updates when stock is known to be available.

If your preferred website has a way of sending an email notification when stock returns, try that as well.

Himbeere Pi

I've been looking everywhere I can to try and find a copy of *The MagPi* in Germany, but I've been a bit out of luck. Whereabouts do you sell here? Do you sell in Europe at all?

If you don't sell it in Germany, is there still any way I can get a copy?

Nathan

We're currently not in sale in Germany. In Europe we mainly sell in the UK and Ireland; however, we're looking into being in more countries this year, as there seems to be a demand. If you feel there's a newsagent or shop we should be in, let them know and they might start asking us for copies to sell.

For the moment, though, there are two main ways you can get the magazine. First, we always sell on the Swag Shop (**swag.raspberrypi.org**) when a new issue comes out – it's the normal UK price of £5.99, with shipping charged on top of that.

The cheaper and faster way to do it, though, is via a subscription (**magpi.cc/Subs1**). This way you're guaranteed a copy, as one is always reserved for you when we print them out, and is sent out very quickly.

Starter kits

What's the difference between a Raspberry Pi starter kit and the Raspberry Pi kit that comes with the December issue of *The MagPi*? I've been thinking of doing more with Raspberry Pi, as I'm an educator and want to try and bring it into classrooms in some capacity.
Jennifer Stone

The standalone issue of *The MagPi* #40 wasn't a kit – it was the Raspberry Pi Zero on the front. Subscribers also got a cable bundle allowing them to connect up the Pi Zero to HDMI and USB properly, which is similar to what you might mean by a starter kit; otherwise, the Raspberry Pi on its own is literally just a computer.

There are two types of starter kits you might be referring to. One type of starter kit for the Raspberry Pi contains all the extra accessories you might need to get your Pi running. This can include a power supply, case, HDMI cable, SD card, and maybe even a wireless keyboard and mouse combo.

Otherwise, it may be an electronics starter kit, which will include various components, wires, and possibly a breadboard so that you can start putting together different circuits for the Raspberry Pi. Make sure you're choosing the correct kit, and the right Raspberry Pi, for your own uses.



Above Pimoroni's Pi 2 starter kit has everything you need

FROM THE FORUM: PRINT YOUR OWN

The Raspberry Pi Forum is a hotbed of conversation and problem-solving for the community – join in via **raspberrypi.org/forums**

I've downloaded back issues of *The MagPi* online. I'd like to take them to a local print shop to be printed and bound in one large collection. I don't plan to distribute this; it's just for my own reading. Most print shops require a note saying the author allows this sort of thing. Is this legal? How can I provide proper proof to the print shop?
Yoosi

Absolutely; all *The MagPi*'s content is licensed under Creative Commons. Specifically, we allow for printing for own personal use, so you're in the green. We know it can be tricky to hunt down print versions of some of our issues (especially #40), so go ahead!

As for proof, you can show them that we're licensed under Creative Commons, as our website states (just look under the post for any issue), but we can probably go one step further…

Hi there, lovely printers! Someone may have requested that you print out one of our magazines for themselves, for personal use. This is fine by us. Please use the PDFs they provide you with to make sure it's of the best quality.

There you go – hopefully that should clear up any problems.

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**

TAKE US **ANYWHERE**



SAVE 25% with a Newsstand subscription (limited time offer)

FREE: DOWNLOAD ALL 30 ORIGINAL ISSUES

The MagPi Magazine

Available now

for smartphones & tablets

Available on the App Store

GET IT ON Google Play

Subscribe from

£2.29 rolling subscription or **£26.99** full year subscription

Download it today – it's free!

- Get all 30 legacy issues free
- Instant downloads every month
- Fast rendering performance
- Live links & interactivity

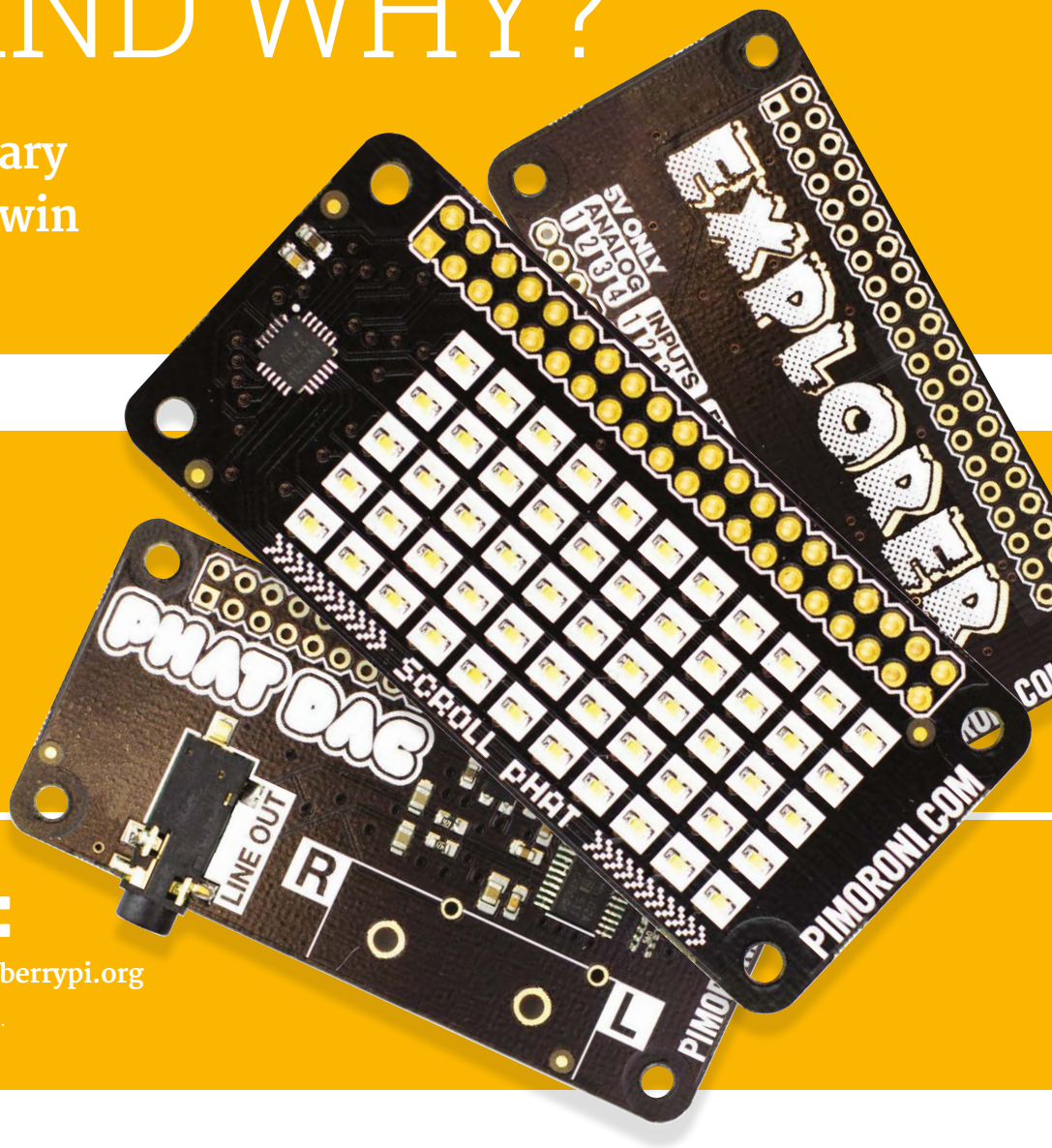
In association with:

PIMORONI

5 OF EACH Pimoroni pHAT MUST BE WON!

WHICH PIMORONI PHAT WOULD YOU LIKE AND WHY?

Tell us by 29 February for your chance to win one of each!



TAKE YOUR PICK!

- **Explorer pHAT**
- **Scroll pHAT**
- **pHAT DAC**

How to enter:

Simply email **competition@raspberrypi.org** with your answer (max 50 words).

Terms & Conditions

Competition closes 29 February 2016. 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.

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.



DON'T FORGET FUN

Playing with Pi can be serious business, according to **Matt Richardson**

We love seeing young people using Raspberry Pi to learn how to code, how computers work, and how to make things with computers. We train teachers and offer free resources for using Raspberry Pi within the classroom. Beyond education, companies all over the world use Raspberry Pis in many different ways, from warehouse sensor networks to internal testing servers.

While Raspberry Pis do some serious work all over the globe, they're also being used everywhere for fun and entertainment. To me, these leisure time uses are just as compelling as all of the practical uses of Raspberry Pi. And even when someone uses Raspberry Pi strictly for fun, we're still achieving our Foundation's mission.

Many of our fans use Raspberry Pi as a media centre. They load up their Pi with software such as OSMC and connect it to their TV to play their video content. As I recently learned by trying OSMC on Raspberry Pi for the first time, it's incredibly customisable and hackable. I wouldn't be surprised to hear if many people who start using OSMC strictly for entertainment end up figuring out interesting ways to customise it. If OSMC can be a gateway to getting more people hacking computers, or getting them interested in the other capabilities of Raspberry Pi, then it's a win for our Foundation.

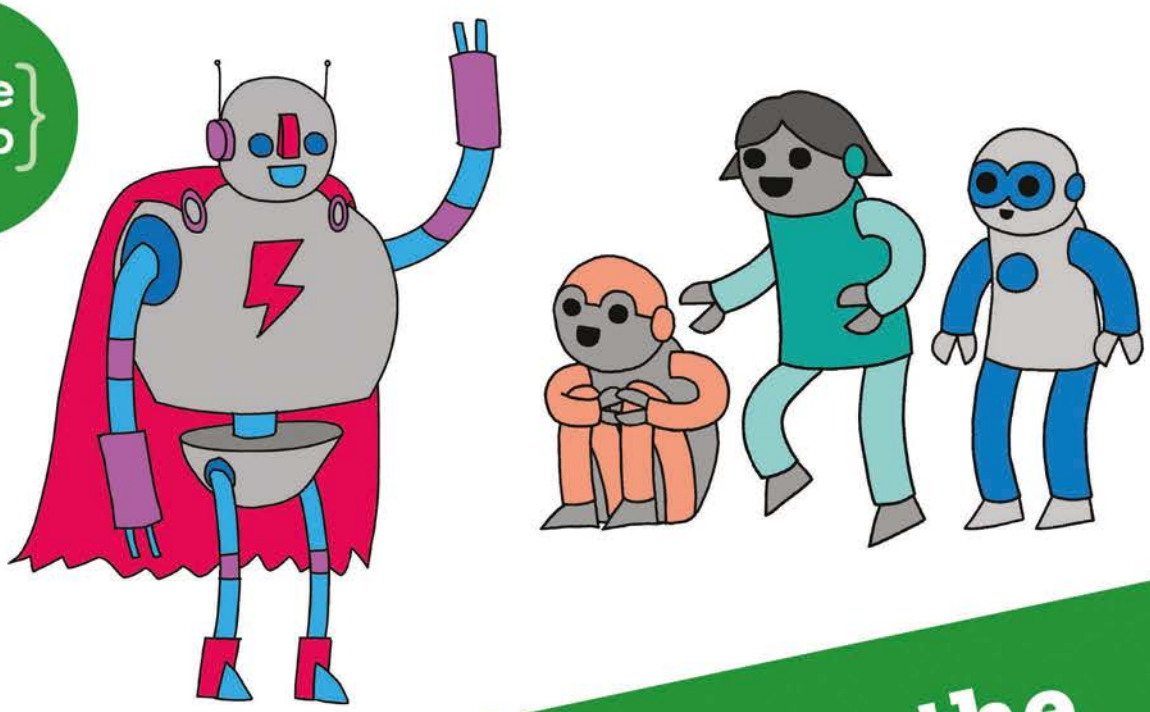
Of course, you can't talk about having fun with a computer without talking about computer games, which have been around for decades. Like any good operating system, Raspbian comes preloaded with a few different games and there are even more free and open-source games available for download.

Minecraft Pi is one of the games you'll find pre-installed in Raspbian. While many kids will enjoy building and exploring *Minecraft* worlds, they can also use the *Minecraft* API in Python in order to use code to build worlds. Among other things, the API enables coders to programmatically build structures. Why would anyone want to do that? If you write the code to build a house with just a click, it means that you can easily build a whole neighbourhood with just a few clicks. So, having fun with *Minecraft* can lead to some serious learning.

You can also have fun using your Raspberry Pi to play pranks on your friends and family. Philip Colligan, the Raspberry Pi Foundation's CEO, was recently telling me a story about how he was working with his son on some experiments with the Sense HAT. They were going through our Magic 8 Ball resource, which displays a random fortune on the LED matrix after you shake the Raspberry Pi. When his son wasn't looking, Phillip reprogrammed the project so that the Sense HAT would display "Self destruct commencing in 10… 9… 8…" According to Philip, he got just the shocked reaction he hoped to get from his son.

I was also delighted by one of the young girls from the Jarvis family, who used her Raspberry Pi to SSH into her father's computer. Remotely controlling her father's computer console, she got it to speak a warning and then force-quit the text editor he was working with. (All of this with proper permissions from mum, of course.) She posted a video of her prank on YouTube, in which she shows exactly how she did it. Her video has over 600,000 views!

It's heartening to see so many young people learn about computers with Raspberry Pi. As a computer, Pi is the real deal; it's not a toy. That's why so many professionals use Pi for practical applications as well. But just because it's not a toy doesn't mean you can't have fun with it!



Can you help inspire the next generation of coders?



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!

So to find out more, join us at **www.codeclub.org.uk**

This advertising space was kindly donated to Code Club by MagPi Magazine

EXPERIMENT WITH THE SENSE HAT

Get started today for **just £2.99 / $3.99**

SENSE THE REAL WORLD WITH YOUR Raspberry Pi

The MagPi ESSENTIALS

From the makers of the official Raspberry Pi magazine

Available on the App Store

GET IT ON Google Play

Find it on The MagPi digital app





magpi.cc/Sense-HAT-book