YOUR OFFICIAL **RASPBERRY PI** MAGAZINE

The MagPi

The official Raspberry Pi magazine

Issue 64 December 2017 | raspberrypi.org/magpi

ELECTRONICS STARTER GUIDE

Serious fun with circuits, components & jumper leads

WIN!

MONSTERBORG ROBOT KIT

INCOMING!

AIY PROJECTS VISION KIT

Made by you with Google

RETRO GAMING

Step-by-step guide to playing classic games

ADD A SIM CARD

Connect your Raspberry Pi to a mobile phone network

MAKE A TEXT ADVENTURE

Design and code a role-playing game

Also inside:

- > THE BEST CHRISTMAS KITS FOR PI PROJECTS
- > ARTIFICIAL INTELLIGENCE IN A SMART CCTV CAMERA
- > NASA DRAWING ROBOT POWERED BY RASPBERRY PI
- > MONSTER JOYSTICKS ARCADE CONTROLLER REVIEWED

Issue 64 • Dec 2017 • £5.99

12

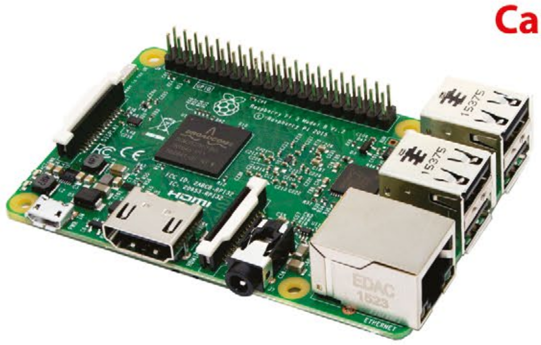
9 772051 998001

Raspberry Pi PRESS

Cana Kit™

CanaKit Raspberry Pi 3 Ultimate Starter Kit

Model B | 1 GB RAM | 1.2 GHz | Quad-Core CPU



> Learn to Code
> Explore Computing
> Get started with Electronics

Kit Includes Raspberry Pi 3 and ...

Premium Case & Heat Sinks

2.5A Power Adapter

32 GB Class 10 MicroSD Card

Pre-Loaded with Operating System

USB MicroSD Card Reader

Premium HDMI Cable

Quick-Start Guide

GPIO to Breadboard Interface Board

Ribbon Cable

Full-Size Breadboard

Jumpers

Male to Male & Male to Female

LEDs

Resistors & Push-Buttons

Available for worldwide shipping at:

WWW.CANAKIT.COM

Raspberry Pi Zero W

Now available at CanaKit!

Wi Fi™

Bluetooth®

Electronic Kits • Electronic Parts • Raspberry Pi • Arduino

WELCOME TO THE OFFICIAL MAGAZINE

Everybody knows the Raspberry Pi is a coding tool. It's inspiring a whole new generation of students and developers. But the Raspberry Pi is much more than that. Coding on a Raspberry Pi has purpose, and that purpose is to control hardware and make things that whizz, whirr, and do stuff.

If you're new to this. We've got an amazing Electronics Starter Guide (**Page 18**).

If you're already a master hacker, don't worry: we have something massive for you this month…

AIY Projects is back with a bang, and Google has revealed to us its new Vision Kit (**Page 6**). Think 'Google Clips made from cardboard'.

Inside is a new Vision Bonnet that adds an artificial intelligence chip to the Raspberry Pi, making the Raspberry Pi 60 times faster at vision processing. It also enables you to control electronic components with AI.

(Remember: 'Don't do evil with your Raspberry Pi'.) Seriously though: it's an amazing piece of kit and we can't wait to see what you all make with it.

I wish you all a Merry Christmas and a Hacky New Year. Don't forget to share the things you build with us and the rest of the maker community.

Lucy Hattersley
Editor - The MagPi

FREE PI ZERO W WITH A SUBSCRIPTION! PAGE 30

SUBSCRIBE & SAVE 25%
SEE PAGE 30 FOR DETAILS

THIS MONTH:

06 AIY PROJECTS VISION KIT
The new smart camera project made by you with Google

18 ELECTRONICS STARTER GUIDE
Get started with components, kits, and circuits

56 RETRO GAMING
Your quick-start 101 introduction to retro game emulation

62 MAKE A TEXT ADVENTURE
Code your own text adventure with this classic tutorial

FIND US ONLINE raspberrypi.org/magpi

GET IN TOUCH magpi@raspberrypi.org

EDITORIAL
Editor: **Lucy Hattersley**
lucy@raspberrypi.org
Features Editor: **Rob Zwetsloot**
rob@raspberrypi.org
Sub Editors: **Phil King and Jem Roberts**

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

PUBLISHING
For advertising & licensing:
Publishing Director: **Russell Barnes**
russell@raspberrypi.org | **+44 (0)7904 766523**
Director of Communications: **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
Alex Bate, Mike Cook, David Crookes, Kent Elchuk, Kyle Goff, Phil King, Sean McManus, Binsen Qian, Matt Richardson, Richard Smedley, Dean Swift, Clive Webster, Sai Yamanoor, Srihari Yamanoor

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., 30 Station Road, Cambridge, CB1 2JH. 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 64 December 2017 raspberrypi.org/magpi

TUTORIALS

> BUILD A KALEIDO CAM **42**
Make a very artistic camera in the Pi Bakery

> GPIO CONTROL INTERFACE **46**
How to use C-STEMbian for GPIO programming

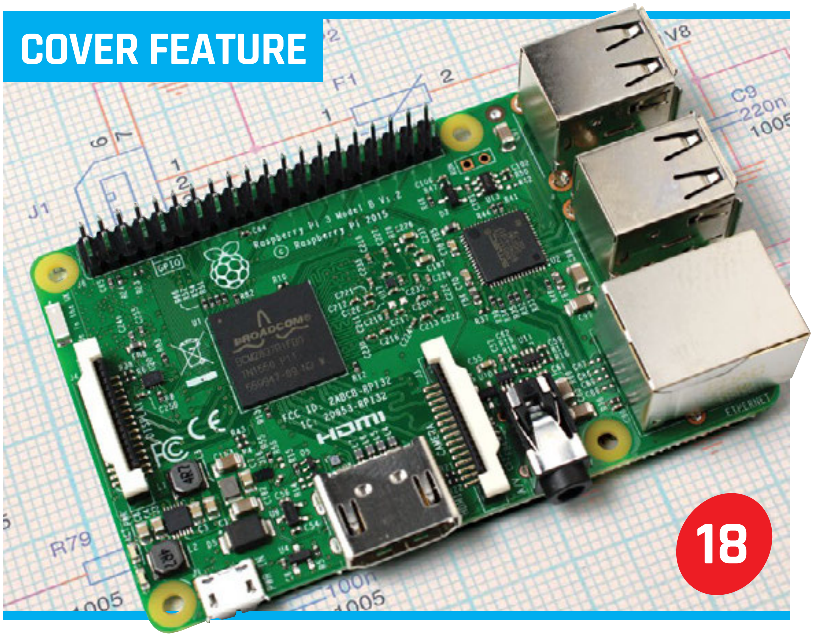
> MAKE A XMAS TREE STAR **48**
3D-print a star and light it up with a Raspberry Pi

> DIY AIR-QUALITY MONITOR **50**
Use a Pi Zero to create a light-up air-quality sensor

> MOBILE INTERNET ON PI **54**
Using Hologram to get WiFi wherever you are

> PI 101 - RETRO GAMING **56**
Enjoy some classic games on your Pi using Lakka

COVER FEATURE



18

ELECTRONICS STARTER GUIDE

IN THE NEWS

VISION KIT

A new AIY Projects kit is coming from Google **06**

HACKING WITH PIP

The hackable handheld console **08**

HACKSPACE

Read our new sister magazine for makers **12**

THE BIG FEATURE

MAKE A TEXT ADVENTURE 62

Head North and get eaten by a Grue

WIN! MONSTERBORG & THUNDERBORGS 97

In association with piborg.org

YOUR PROJECTS

RAPANUI 32

Making clothes with a Raspberry Pi

SMART SECURITY 34

A powerful homemade security camera

NASA ARM 36

A space-age robot that draws whatever you ask

OUT RUN ARCADE 38

Hacking a toy into the best arcade racer ever

REGULARS

> NEWS 06

> TECHNICAL FAQ 60

> BOOK REVIEWS 80

> FINAL WORD 98

COMMUNITY

> PI ROOM INTERVIEW 82

Bringing computing to Togo in Africa

> THIS MONTH IN RASPBERRY PI 84

Everything else that happened in the community

> EVENT REPORT - PYCON 88

Here's what went on at this year's PyCon

> COMMUNITY PROFILE 90

This month we highlight Lorraine Underwood

> EVENTS SCHEDULE 92

Here are some events you can go to this month

> YOUR LETTERS 94

We answer your pressing questions

REVIEWS

> MONSTER JOYSTICK KIT 76

> CHRISTMAS KITS 78

AIY PROJECTS

VISION KIT

Google AIY Projects team reveals do-it-yourself Pi Zero W intelligent camera

Google AIY Projects has announced its Vision Kit, a smart camera kit you can build using a Pi Zero W, Pi Camera, Module, and a powerful AI board called the 'Vision Bonnet'.

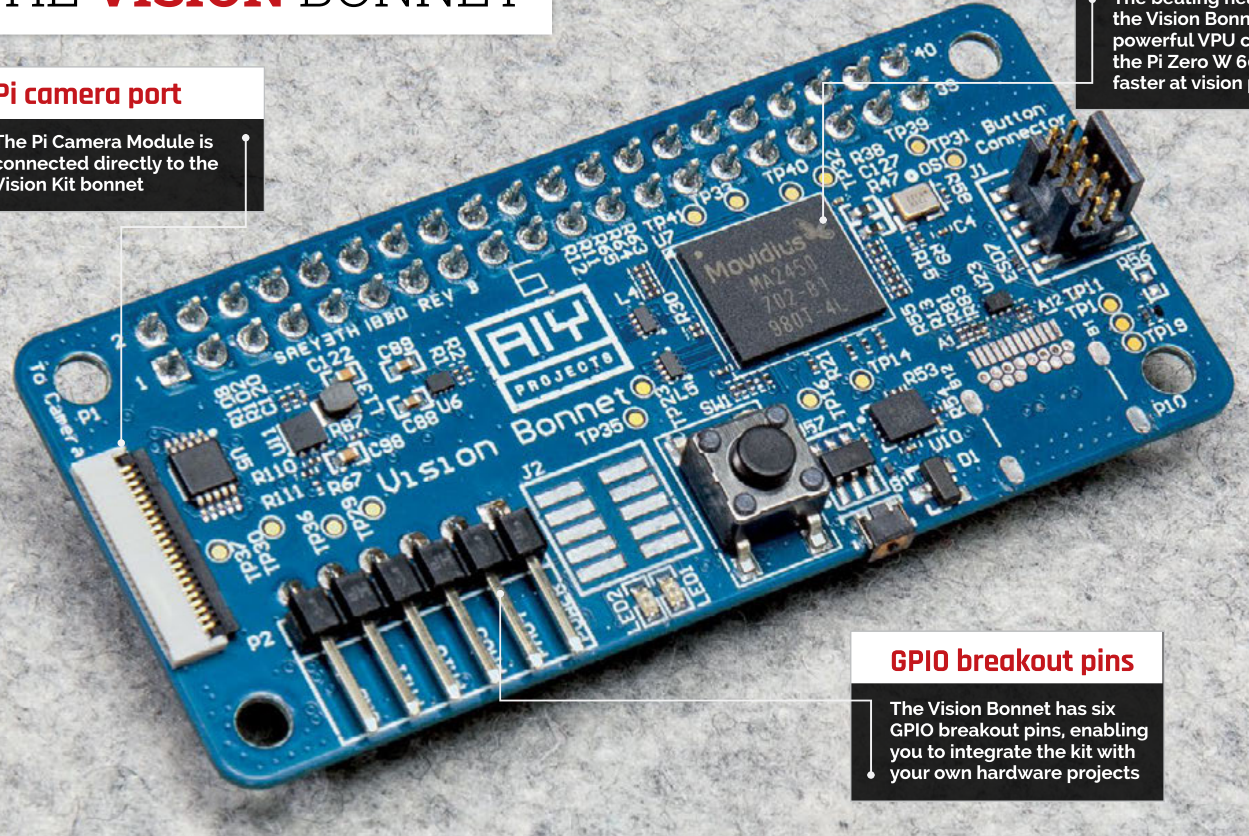
Keen Google fans will recognise the similarity between the Vision Kit and Google Clips, a recently announced smart camera.

The Vision Kit enables makers to build a similar device, but that's just the start.AIY Projects is all about developing hackable AI kits for makers that they can integrate into their own projects.

"We look at how do we get these technologies in people's hands in ways that are easy," says Jess Holbrook, AIY Projects UX Research Lead.

"AI and machine learning are up on this pedestal," he explains. Google wants to show makers that

THE VISION BONNET



Pi camera port

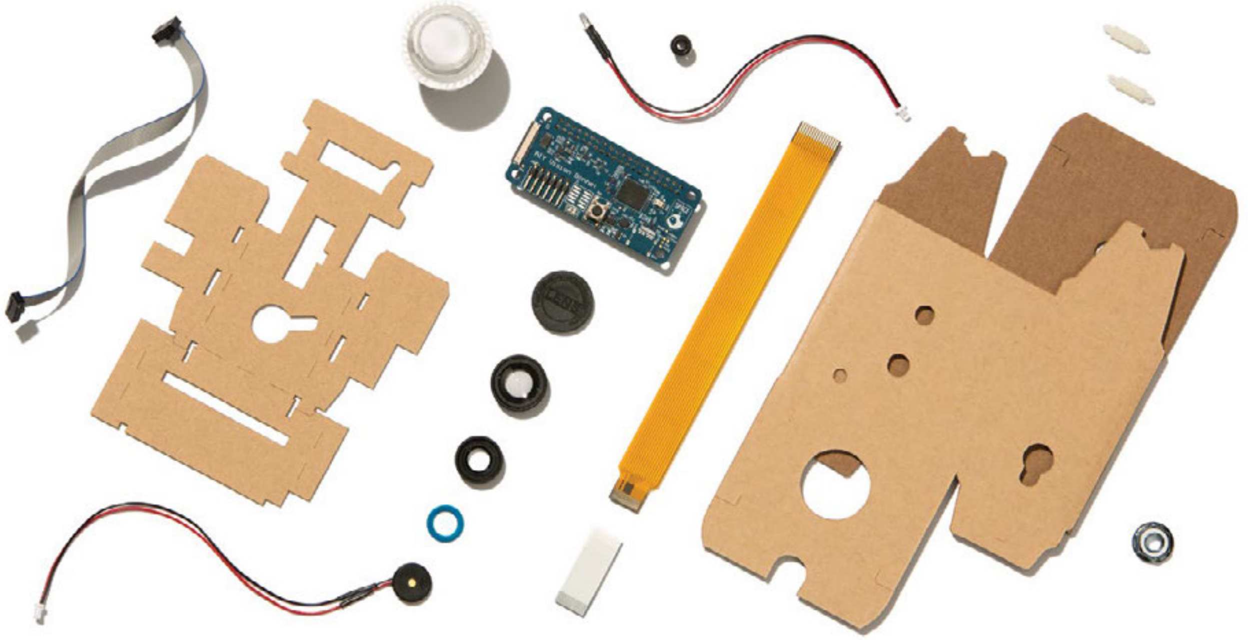
The Pi Camera Module is connected directly to the Vision Kit bonnet

Movidius MA2450

The beating heart of the Vision Bonnet, this powerful VPU chip makes the Pi Zero W 60 times faster at vision processing

GPIO breakout pins

The Vision Bonnet has six GPIO breakout pins, enabling you to integrate the kit with your own hardware projects



ORDER DEC

The AIY Projects Vision Kit will be available for pre-order from Micro Center in December 2017. The initial run is going to be very limited. If you're interested, we advise you to order a kit as soon as it's available.
microcenter.com

they can build pretty impressive stuff themselves. As with earlier AIY Projects, Google is interested to see what ideas makers come up with to use AI to "solve problems for themselves, their families, and their communities."

"We have developed a deep learning inference acceleration engine that we're running on the chip," explains Kai Yick from the AIY Projects team. "It's 60 times faster than trying to do it on a Raspberry Pi 3."

> "The Vision Kit adds an advanced AI hardware board to the Raspberry Pi"

All aboard AI

The Vision Kit adds an advanced AI hardware board to the Raspberry Pi. Developed by Google, it's called the Vision Bonnet and sports a powerful Movidius MA2450 (**movidius.com/myriadx**) vision processing chip.

The chip acts as a "neural network accelerator" says Billy Rutledge, Director of AIY Projects at Google. "In the case of Vision Kit, we are moving forward in a pretty big leap and running the AI neural network on the accessory board itself."

This is in contrast to the earlier AIY Projects Voice Kit, which relied on Google Cloud infrastructure for voice recognition and natural language processing.

The projects you build will operate independently of a network connection, making for a more versatile piece of equipment. The Vision Bonnet also ensures the security of captured images, as they are all processed locally on the device.

Kit form

Once you've assembled the Vision Kit, there are a number of neural networks software programs you can run. The first neural network is a "person, cat, and dog detector," reveals Peter Malkin, Software Lead at AIY Projects. It detects if a person, cat, or dog is in the frame.

The second neural network is focused on facial emotions. It will detect happiness, sadness, and other sentiments.

The third neural network can identify 1,001 common objects, like a cup, an orange, or a chair. A label displays the name, and the level of confidence in the neural network's inference.

In the future, the hope is that users will be able to modify these neural networks. You could take the cat, dog, and person model and modify it to look for rabbits. Then build a project that works with a rabbit hutch, for instance.

Both *The MagPi* and Google can't wait to see what makers have in store for the AIY Projects Vision Kit. Pre-orders for the Vision Kit will begin in December at Micro Center (**microcenter.com**).

Below The Vision Kit is a smart camera project used to detect faces, moods, and objects; you can build it at home and integrate it with your own projects

PIP THE HACKABLE HANDHELD

Make apps, play games, hack anything

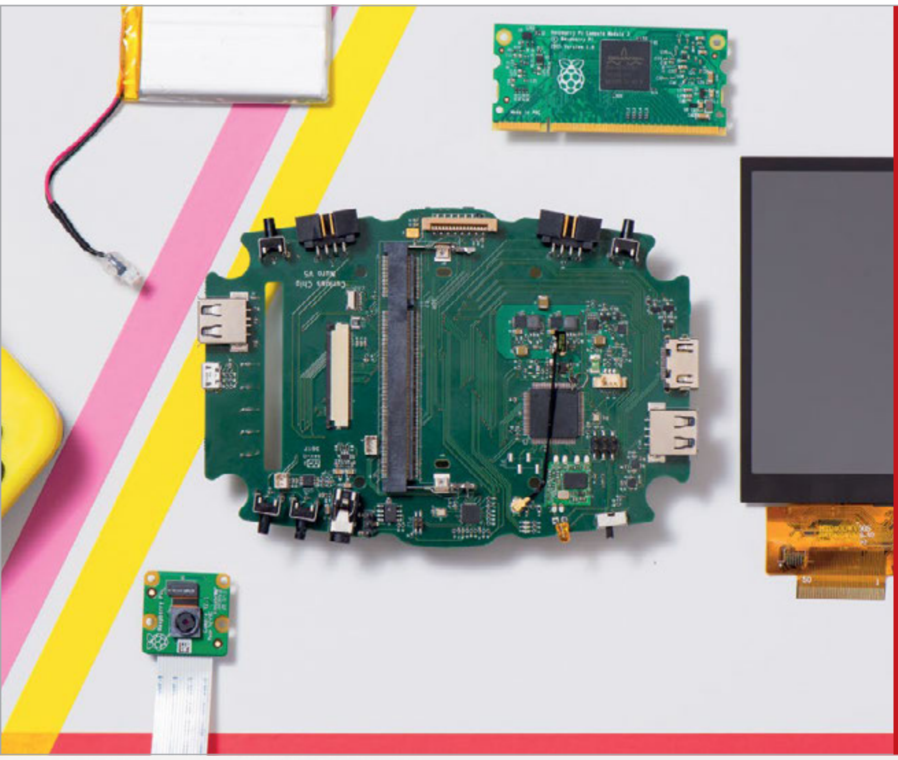
Scottish firm Curious Chip has unveiled Pip, "the playful handheld device you program yourself." Pip has been launched on Kickstarter, with a modest £50,000 target – see **kck.st/2yAEc4h**. A £150 pledge bags you a Pip, while £200 nets you a Pip, camera, and maker pack.

With final-version prototypes to play with, Raspberry Pi co-founder Eben Upton has given his backing: "I'm super-excited about this! I'm really, really looking forward to seeing what people do [with a Pip]."

Pip's D-pad and buttons are mounted on two detachable USB controllers – however, Jason Frame, co-founder and technical lead for Pip, tells us, "We came up with the removable controller idea… a full two weeks before [a certain handheld game console was announced]". Two shoulder buttons are housed in the main body.

Pip also includes a row of eight RGB LEDs and a 40-pin GPIO ribbon cable connector either side of its 4-inch, 800 × 480 pixel touchscreen. There's a speaker, microphone, accelerometer, and compass, as well as two USB ports, HDMI out, wireless, and an optional 5MP camera. Pip is powered by a Raspberry Pi Compute Module 3 Lite.

Above Pip is more than a games console – it's a mobile makers' lab



DOES COMPUTE

Unusually, Pip uses a Compute Module 3 (CM3) rather than a Raspberry Pi 3. But Curious Chip didn't know what would power Pip in the early days of development. "The CM3 Lite edition (with SD card support) was announced shortly before Pip v1 debuted at Bett 2017 [a trade show for education technology] and immediately we gave it a serious look," Jason tells us.

You can see the resultant motherboard to the left, with the CM3 slot (which the Raspberry Pi Foundation borrowed from laptop SODIMM memory) in the centre.

Asked whether the use of CM3 gave Pip a possible upgrade option, Jason responds, "Size was the main concern… The smaller a device is, the more likely you are to take it with you and the more to can do with it." He adds,"Pip should have enough juice for our expected use-cases for a good while to come. Isn't a quad-core, credit-card-sized SoC with 1GB RAM enough for anybody?"

Made for making

Pip might look like a portable games console, and indeed it comes preloaded with "a number of open-source remakes of popular games like Pac-Man and Super Bomberman," according to Jason.

However, he adds, "What we really want is for kids – and adults! – to go beyond that: make stuff and share it with the world. Games, apps, physical stuff, everywhere and anywhere." Pip was made with making in mind. "All we wanted to do," Jason continues, "was make a portable device that you could have your own code running on in less than five minutes."

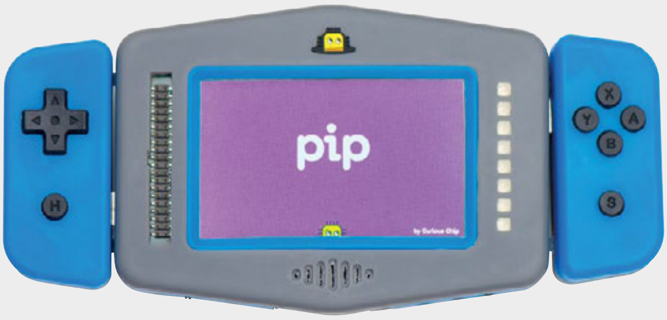
Above Make apps, games, and other software with Pip's built-in development tool

> "A portable device that you can have your own code running on in less than five minutes"

To this end, Pip has its own IDE, called Curiosity. "It's hosted on Pip itself," Jason reveals, "so there's nothing to install." You can create software for Pip on any machine with a web browser.

"Curiosity offers one 'mode' for each language supported by Pip: JavaScript, Python, Lua, and PHP+HTML5," Jason explains. "There are built in libraries for games coding that simplify stuff like drawing, collision, particle systems, and even shader effects."

You can emulate your code in the browser, and "it's just one click" to send your code to Pip wirelessly. For young coders, there's a visual coding app called Livewire. Jason confirms , "We've been iterating [Livewire] for five years now."



BITBARISTA

THE FIRST AUTONOMOUS COMPANY?

Coffee for the masses

A University of Edinburgh project might just be the first fully autonomous business. The Pi-powered BitBarista accepts BitCoin payments to dispense coffee, pays users to restock supplies, and will call a technician should a fault occur.

BitBarista is not some glorified kettle with WiFi, though. Using the BitCoin from coffee sales, BitBarista "pays people for small services such as filling its water tank and replenishing it with coffee beans," according to Dr Larissa Pschetz, Programme Director of Product Design and Researcher at the Centre for Design Informatics, University of Edinburgh and co-creator of BitBarista.

> "BitBarista pays people for small services such as filling its water tank and replenishing it with coffee beans"

Above The technology is advanced, but BitBarista is built using a standard De'Longhi coffee maker

Above BitBarista asks users where to restock its coffee from – would you choose the cheapest, or the most ethical?

When coffee stocks run low, BitBarista asks the user to select a restock vendor. These options could be ranked in price order, or by ethical preference. Larissa explains, "I was also intrigued by this idea of an autonomous coffee machine that would reduce intermediaries in the coffee trade, potentially supporting smallholder famers in developing countries."

BitBarista is currently on a long-term trial throughout the UK, "which is revealing reactions to the machine's autonomous features," according to a University of Edinburgh spokesperson.

So far, Larissa has seen that "some rituals such as making coffee for colleagues were lost while others, such as watching who chose the 'best price' option, were created."

Raspberry Pi and coffee

The Pi controlling BitBarista uses Raspbian "and different plug-ins to simplify tasks and connect things together," Larissa reveals. "We also use Electrum for the BitCoin payments." The code is on GitHub, at **magpi.cc/2iVHXKI**.

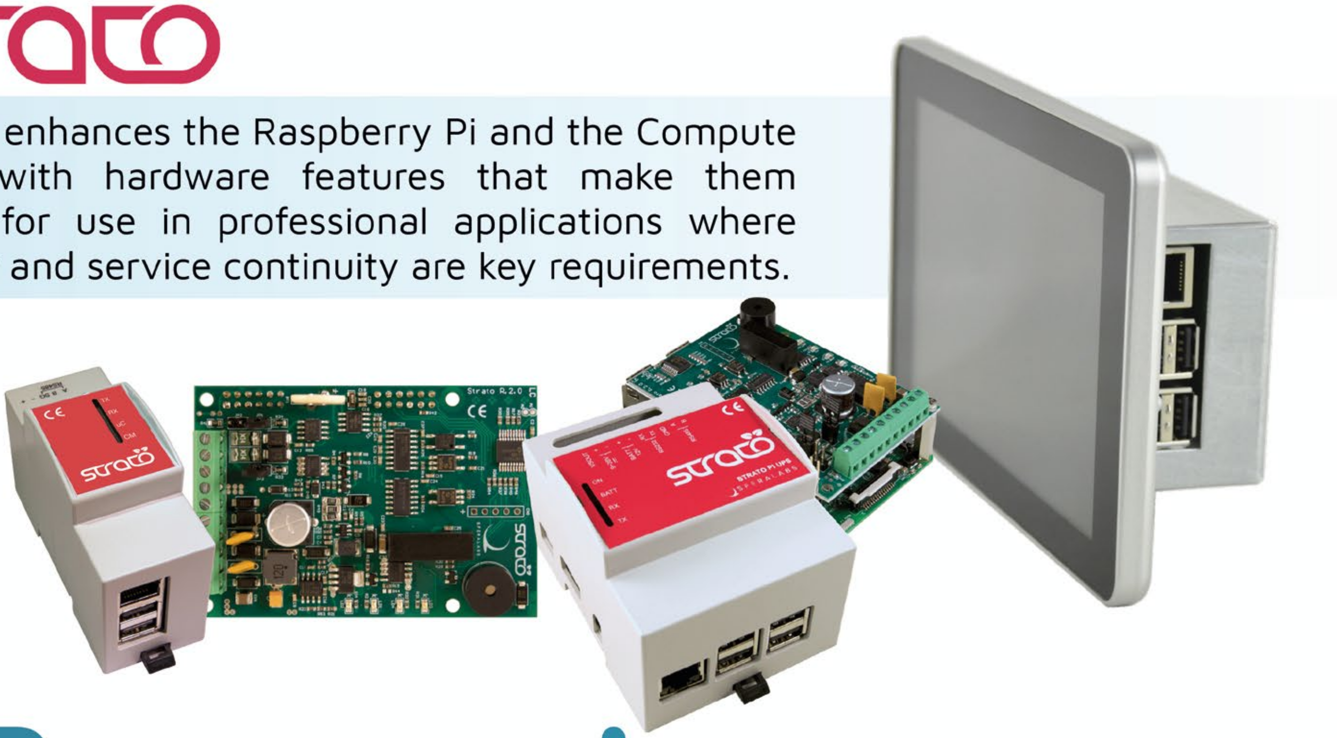
BitBarista might be useful for other vending machines, but Larissa warns, "The machine still needs to follow the social rules of the context where it is placed. If people think it is inappropriate they will probably turn it off."

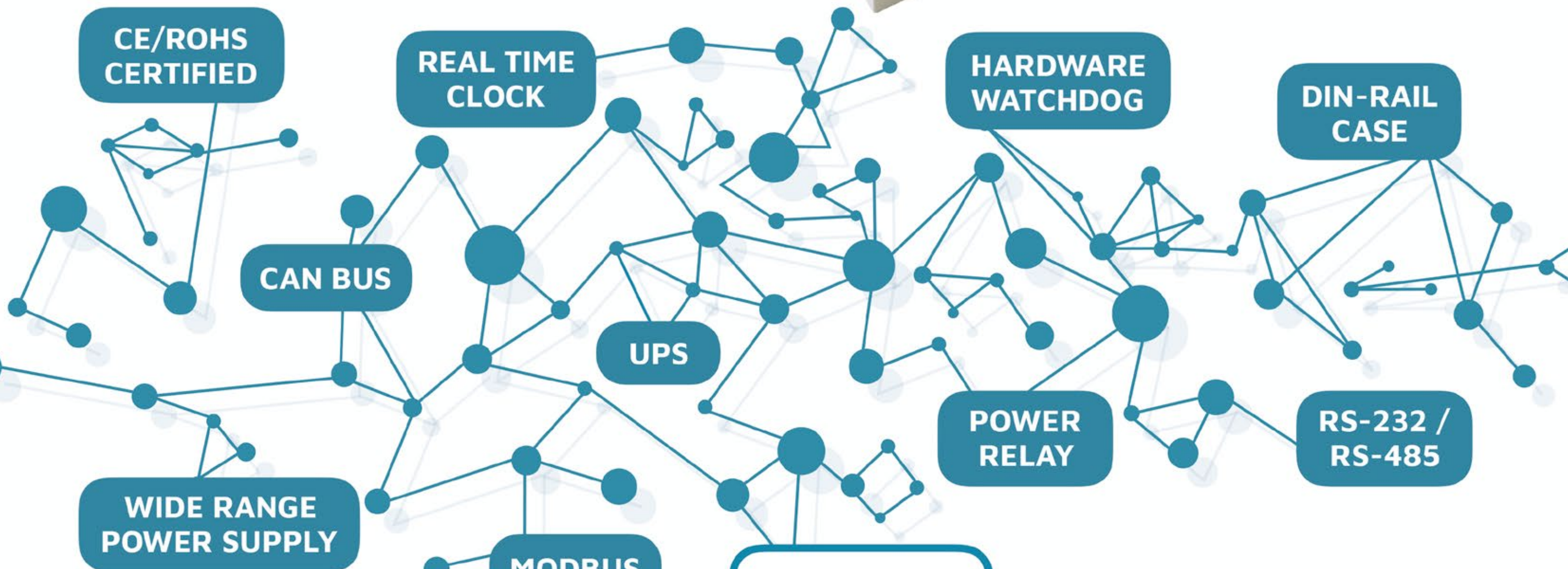
Asked whether there were machines where BitBarista technology might not ever be appropriate, Larissa replies, "Of course there is always space for malicious practices, but people are careful and good at figuring these things out so it wouldn't last in the long term."

Raspberry Pi for professional applications

strato

Strato Pi enhances the Raspberry Pi and the Compute Module with hardware features that make them suitable for use in professional applications where reliability and service continuity are key requirements.




CE/ROHS CERTIFIED

REAL TIME CLOCK

HARDWARE WATCHDOG

DIN-RAIL CASE

CAN BUS

UPS

POWER RELAY

RS-232 / RS-485

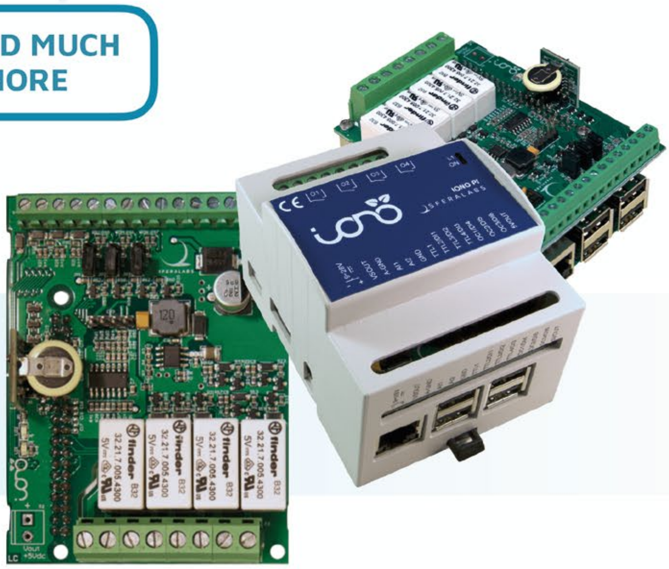
WIDE RANGE POWER SUPPLY

MODBUS

.. AND MUCH MORE

iono

Iono Pi is a versatile I/O module that combines digital and analog standard interfaces with the powerful computing core of the Raspberry Pi.



SFERALABS | www.sferalabs.cc

NEW MAKER MAG!

NEW HACK | MAKE | BUILD | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
BECKY STERN
BUILD THINKING MACHINES
Harness the power of artificial intelligence
WORLD'S FASTEST RASPBERRY PI
MAKE BEAUTIFUL MUSIC
SMOKE YOUR OWN BACON
BUILD A TREBUCHET!
HANDHELD CONSOLES FOR HACKERS
BOLDPORT: BUGS, BITS & BLINKENLIGHTS
TRANSISTORS 3D PRINTING

Grab the latest magazine from the Raspberry Pi Foundation and get hacking!

Introducing **HackSpace** magazine

The Raspberry Pi Foundation has launched a new magazine aimed at makers, hackers, and tinkerers. Following on from the success of *The MagPi* – The Official Raspberry Pi magazine and its education-focused sister mag Hello World comes HackSpace magazine (**hsmag.cc**).

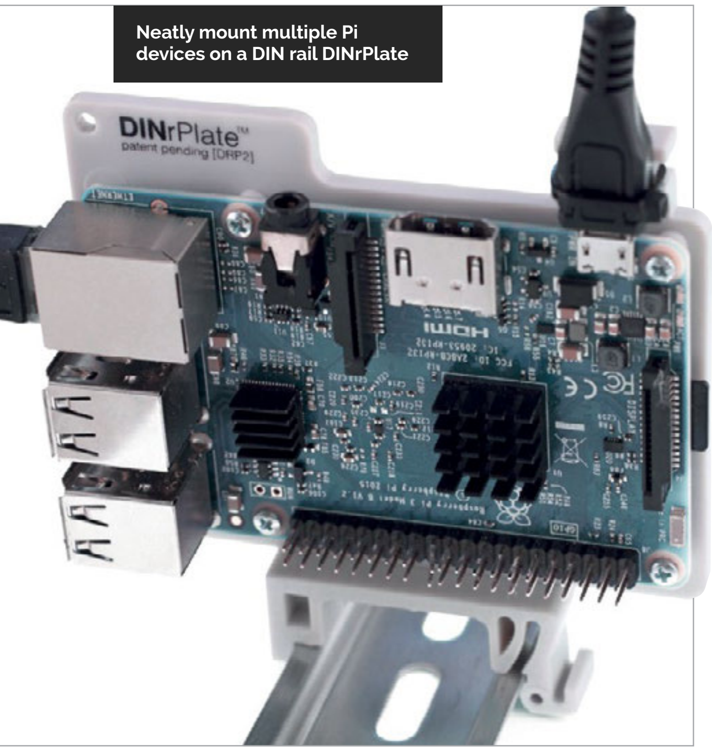
This monthly title is all about hacking and making. As Features Editor Andrew Gregory says, "Grab some duct tape, fire up the microcontroller, ready a 3D printer, and hack the world around you!"

HackSpace magazine features tutorials and projects covering a huge range of topics. Just like *The MagPi*, HackSpace magazine is written by the community for the community, so get in touch if you've got a great project you would like to share, or if you want to see more coverage of certain topics.

Copies of HackSpace magazine will be in UK newsagents from 23 November, and you can download a free PDF version. Subscriptions start from just £2.50 a month if you already subscribe to *The MagPi*.

Follow HackSpace magazine on Twitter **@HackSpaceMag**.

MOUNT PI DEVICES ON A DINRPLATE



Neatly mount multiple Pi devices on a DIN rail DINrPlate

Neatly arrange Pi clusters on a DIN rail

DINrPlate offers a neat way to mount many Pi devices together in a small space by attaching them all to a 35 mm DIN rail, the same mount that modern fuse boxes use.

The first DINrPlates were launched toward the end of 2015 for full-sized Raspberry Pi devices, but there's a new range for Pi Zero and Arduino units. Kurt Baum, President of DINrPlate, tells us: "The incredible success of the Pi and support behind it has made it the de facto development system for other industrial applications as well… the DINrPlate fills the need to securely mount the Raspberry Pi in an industrial environment."

The unit is made of tough Nylon 6/6 plastic, onto which you screw the Pi using the supplied screws. Unlike other DIN cases which enclose the Pi, DINrPlate's "open-frame design keeps the Pis running cooler under heavy workloads," according to Kurt. The tab is DINrPlate's "strain relief system to protect the power cable from being accidentally unplugged."

The DINrPlate Zero costs £10 from **modmypi.com**.

DINrPlate™

GET CODING QUICKER

Install Python modules in seconds with piwheels

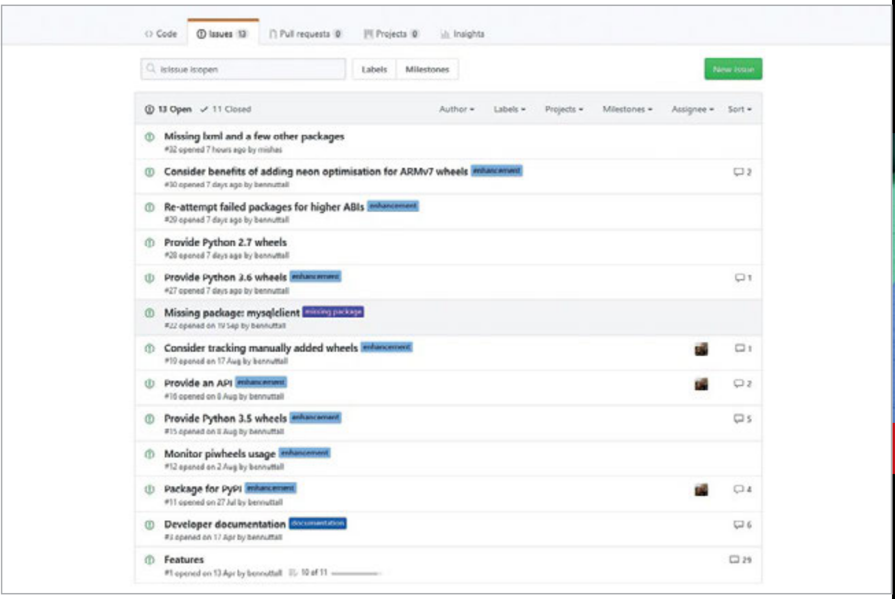
Pi Foundation Community Manager Ben Nuttall has released piwheels, a set of pre-compiled Python modules that reduces the install time from tens of minutes to a couple of seconds.

Pip is the best way to install Python packages, as it grabs the very latest version. However, often these packages “have been implemented in C and require compilation,” Ben explains. Compilation takes ages, however – installing the NumPy package on a Pi 3 can take over 20 minutes.

> “Pip is the best way to install Python packages, as it grabs the very latest version”

The Python community solved this problem a while back with ‘wheels’, pre-compiled versions of packages for common hardware and software combinations. However, until Ben’s project, there were no wheels for a Pi running Raspbian.

Enter Ben with his epic project to pre-compile all 750,000 wheels for the Raspberry Pi. With some help from SQL expert Dave Jones (**@waveform80**), and the Raspberry Pi Foundation’s web host Mythic Beasts (**mythic-beasts.com**), there are now over 670,000 wheels for more than 96,000 packages. NumPy now installs in seconds using the command **`sudo pip3 install numpy`**.



Missing lxml and a few other packages
#32 opened 7 hours ago by mishas

Consider benefits of adding neon optimisation for ARMv7 wheels enhancement 2
#30 opened 7 days ago by bennuttall

Re-attempt failed packages for higher ABIs enhancement
#29 opened 7 days ago by bennuttall

Provide Python 2.7 wheels
#28 opened 7 days ago by bennuttall

Provide Python 3.6 wheels enhancement 1
#27 opened 7 days ago by bennuttall

Missing package: mysqlclient missing package
#22 opened on 19 Sep by bennuttall

Consider tracking manually added wheels enhancement 1
#10 opened on 17 Aug by bennuttall

Provide an API enhancement 2
#16 opened on 8 Aug by bennuttall

Provide Python 3.5 wheels enhancement 5
#15 opened on 8 Aug by bennuttall

Monitor piwheels usage enhancement
#12 opened on 2 Aug by bennuttall

Package for PyPI enhancement 4
#11 opened on 27 Jul by bennuttall

Developer documentation documentation 6
#3 opened on 17 Apr by bennuttall

Features 29
#1 opened on 13 Apr by bennuttall 10 of 11

Above Any suggestions or errors should be logged on the GitHub project page: **magpi.cc/2iWg1GV**

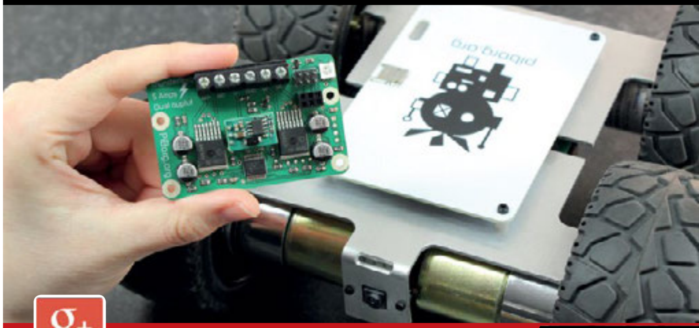
NOW TRENDING

The stories we shared that flew around the world

ALLO DIGIONE HAT RIVALS SONOS

magpi.cc/2lHZWsS

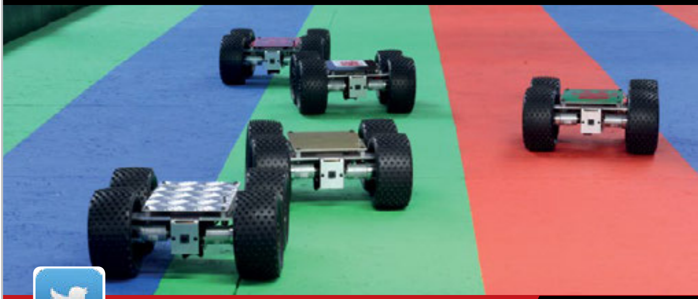
Allo’s £95 DigiOne ‘audio transport’ HAT has made a few waves online, claiming that its low jitter of 0.6 ps and noise of 50 µV make it superior to many high-end audio setups.



MONSTERBORG REVIEW

magpi.cc/2lZ11N3

PiBorg’s latest robot kit is well named. Its four 300 rpm motors, ThunderBorg motor controller, ‘massive’ 105 mm wheels, and chunky 3 mm-thick aluminium casing wouldn’t look out of place on a battlefield.



ROLLS-ROYCE JOINS FORMULA PI

magpi.cc/2lMmybw

Rolls-Royce has joined with Formula Pi to offer coders the chance to ‘Race Your Code’. Successful applicants had the opportunity to race ‘autonomous, Raspberry Pi-powered robots around a track’ from 20 Nov to 7 Dec.

THE READ-ONLY PI

Image credit: Phillip Burgess, **learn.adafruit.com/assets/46754**

Phillip's script includes ways to disable read-only mode, by use of a jumper or quick button connect

Adafruit script for 'make and forget' Pi builds

If you've ever built a Pi for a kiosk, installation or information display, you'll find Adafruit's read-only Pi script invaluable. This script disables all the write-to-SD-card functions, meaning that you can pull the plug without any risk of data loss or corruption.

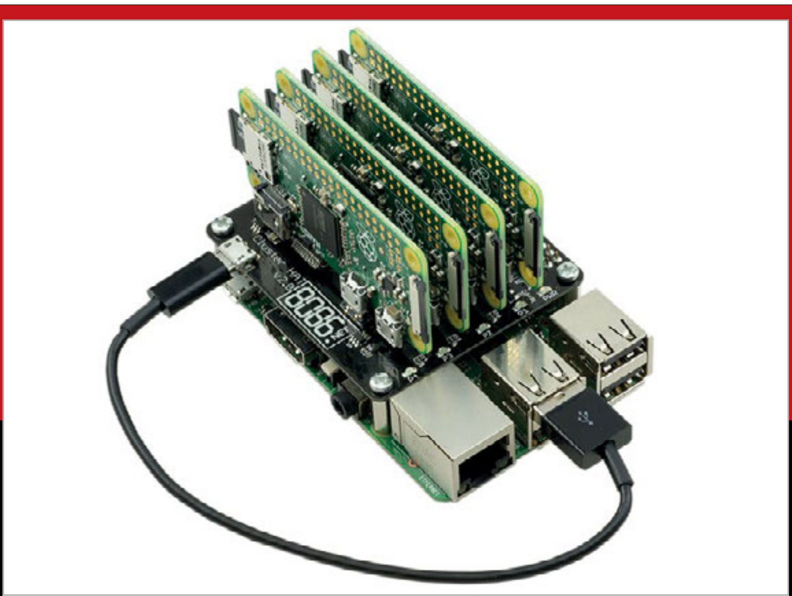
The Adafruit script forces Raspbian to store all its temporary data in memory, so no data stored on the SD card can be corrupted. This also means that nothing at all can be written to the SD card.

Script author, and Adafruit Creative Engineer, Phillip Burgess tells us, "We're seeing Raspberry Pi really take off in situations where a microcontroller may have once sufficed," and points out that less technical users are making things with a Raspberry Pi. Phillip wanted the ability to simply cut power like one does with an Arduino or a consumer product.

The script only works on Lite installations – "Stretch or Jessie should work," Phillip tells us – and you can find the full instructions at **magpi.cc/2hB7qMT**.

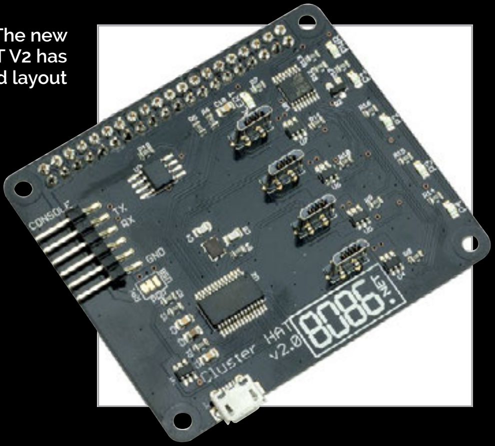
EASY CLUSTER COMPUTING

Cluster HAT handles the hardware



Above Build a cutting-edge cluster computer in minutes

A new version of the Cluster HAT – allowing you connect up to four Raspberry Pi Zeros in a cluster – is on sale from ModMyPi for £28, or for £90 with four Zeros and four 16GB microSD cards (see **magpi.cc/2iWlmxK**).



Right The new Cluster HAT V2 has an improved layout

We liked the V1.2 ClusterHAT back in *The MagPi* #55, and this new model refines the design. ModMyPi Director Jacob Marsh explains, "The V2 has an improved layout and utilises an I²C I/O expander – essentially it means the controller can be rebooted without affecting the Pi Zeros, and the power to each Pi Zero can be controlled individually."

There are also a few new headerless jumpers (which look like solder pads). For example, the POS jumper on the underside can be cut to force the Pi Zeros to power up with the host Pi.

The affordable Cluster HAT opens the door to anyone wanting to experiment with distributed computing techniques commonly used in data centres or when analysing big data.

PI-CONTROLLED ECO HOME BAGS PRESTIGIOUS US PRIZE

Swiss team wins US Department of Energy's 'Solar Decathlon'

A team of engineering students has won the annual Solar Decathlon, which challenges teams to "design and build a full-size, solar-powered house" and is run by the US Department of Energy (DoE).

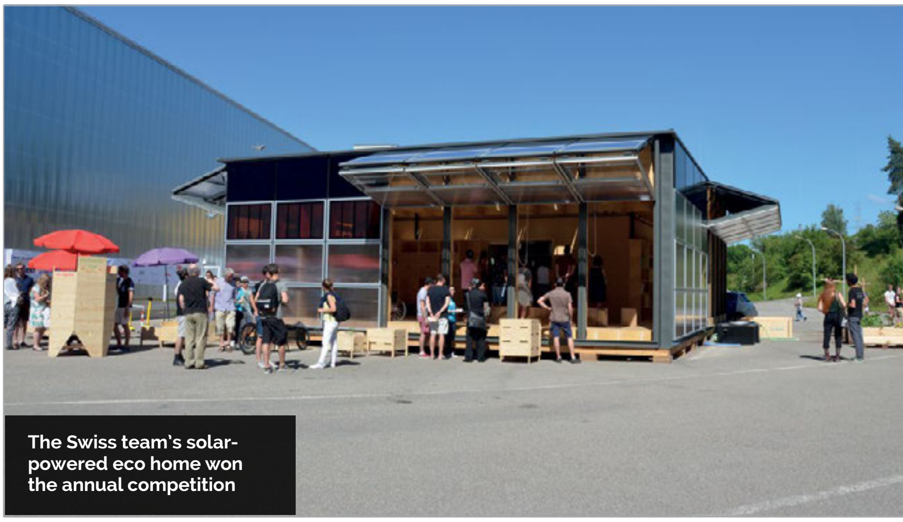
The Swiss team's NeighborHub project scored 872.91 out of 1000, with perfect scores for the architecture, engineering, and energy challenges (and 95 points or more in four other challenges). This clearly met the DoE's brief of being "the team that best blends design excellence and smart energy production with innovation, market potential, and energy and water efficiency."

The NeighborHub design uses four prefabricated 'core' modules to provide a kitchen, bathroom, bedroom, and closets; around this is the 'Extended Skin', a semi-exterior area that resembles a glazed cloister. The Swiss team's brief for the technology was that "actuators are able to communicate to each other and operate together to reach optimal performance" and also to enable "sharing through a learning-by-doing approach."

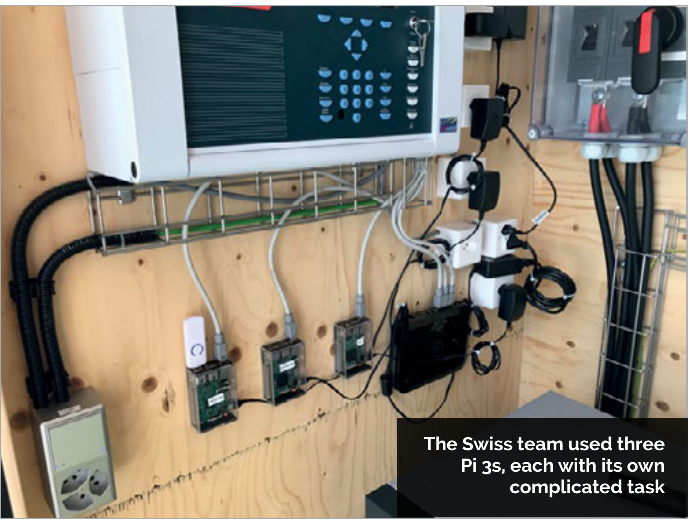
Swiss team engineering lead Victor Saadé tells us that telecommunication student and team member Florian Meyer "basically designed the whole house control system," in partnership with Swiss team sponsor ThinkEE (**thinkee.com**). It was the flexibility of the Pi that appealed; the team used three Pi 3s in the final build: one running a database, another running "all the algorithms", and a third acting as a router.

You can read more about the NeighborHub at **magpi.cc/2iWrTsg**.

Image credit: Swiss Living Challenge



The Swiss team's solar-powered eco home won the annual competition



The Swiss team used three Pi 3s, each with its own complicated task

pi-top

Inspiring inventors and creators to seek the skills of tomorrow and create their future, today.

POWERED BY RASPBERRY PI

OCR
Oxford Cambridge and RSA

The modular laptop with sliding keyboard

8HR BATTERY LIFE

14" FULL HD 1080P SCREEN

180° HINGE

CUSTOM PASSIVE COOLING BRIDGE

MODULAR RAIL

pi-top

Colors 🟩
Raspberry Pi 3 optional

AWESOME INVENTOR'S KIT INCLUDED

20+ projects to explore

Explore beyond the screen and keyboard by creating with the all-new **pi-top** modular laptop.

Get started with 20+ inventions in the inventor's guide booklet. There are 3 inventor's journeys - Smart Robot, Music Maker and Space Race.

The modular desktop

14" FULL HD 1080P SCREEN

MODULAR RAIL

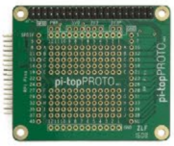
ADJUSTABLE VIEWING ANGLES

pi-topCEED

Colors 🟩⬛
Raspberry Pi 3 optional

pi-topCEED is the plug & play modular desktop. It's the easiest way to use your Raspberry Pi. We've put what you love about our flagship laptop in a slimmer form factor. Join hundreds of code clubs and classrooms using **pi-top**CEED as their solution to Computer Science and STEAM-based learning.

Modular **Accessories**



pi-topPROTO

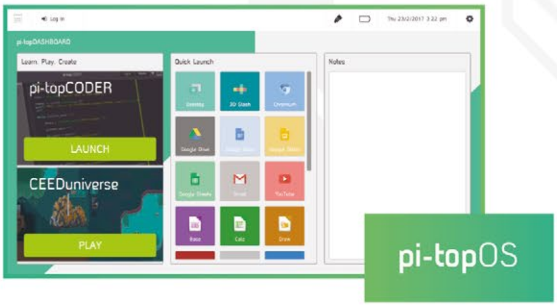
pi-topSPEAKER

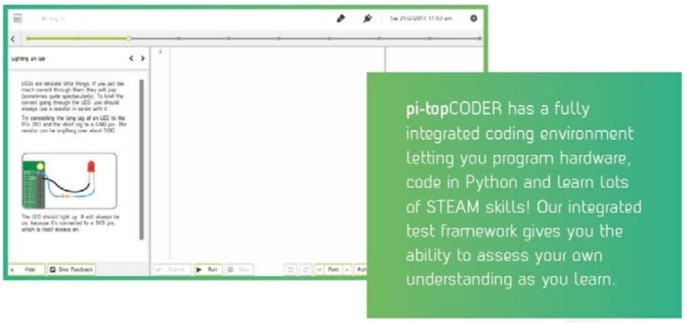
pi-topPULSE

www.pi-top.com @GetPiTop /GetPiTop Stay up to date with our latest news by following our social media

pi-top is an award-winning ecosystem designed to make experimenting, coding and building electronics, simple, affordable and fun. **pi-top**OS is here to guide you through the world of making!

The OCR* endorsed **pi-top**OS (Operating system) platform comes pre-installed on the SD card shipped with every unit. **pi-top**OS software suite lets you - browse the web, - check emails, - create and edit Microsoft Office compatible files. Gain access to dozens of hands-on learning lesson plans with **pi-top**CODER and have fun learning to code with CEED**universe**!





pi-topCODER has a fully integrated coding environment letting you program hardware, code in Python and learn lots of STEAM skills! Our integrated test framework gives you the ability to assess your own understanding as you learn.

CEED**universe**

Learn programming concepts through our minigames, for example, learn problem decomposition by solving visual programming puzzles.

ELECTRONICS STARTER GUIDE

Hands-on with circuits, components, and physical computing using a Raspberry Pi computer

BILL OF MATERIALS

A bill of materials (or 'BOM') is a list of parts you'll need to reproduce a project. Typically in a Raspberry Pi project, this is a list of electronic components and software installations.

As well as being the best-looking, sharpest, and most stylish computer around, the Raspberry Pi is the one with the GPIO pins. These pins enable your Raspberry Pi to get physical with the outside world. You can hook them up to circuits and attach electronic components like LED lights, buzzers, buttons, and sensors.

There are 40 pins in total and no shortage of components you can use. This is what makes the Raspberry Pi such an important part of the computing world. With those pins, you can turn a Raspberry Pi into just about anything.

This feature is for those who want to know how to build, hack, and make with computers. In this guide, you'll learn how to read schematics, prototype circuits, and build basic electronic kits. Welcome to the wonderful world of electronics with Raspberry Pi.

WHY LEARN BASIC ELECTRONICS?

Do you want to be just a user of gadgets? Or do you want to be a maker, hacker, and a fixer? Somebody who knows what is going on under the surface, who knows how to fix, adapt, and improve the world around them.

The Raspberry Pi isn't just a cheap, incredibly well-made computer. It helps you to learn about how computers work.

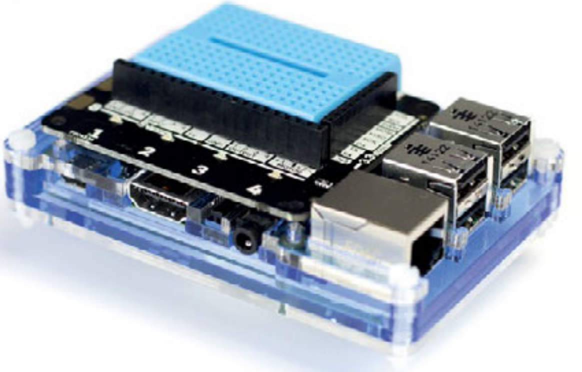
Of course, there are many elements to making. There is learning to code, and how to use maker tools like 3D printers and duct tape. But prototyping your own circuits, and making your own circuit boards, and learning how to connect components together is a vital part of the maker process.

The modern world is infused with electronic devices, and the future is going to be built on artificially intelligent electronic devices. So it's incredibly important to understand how these things work. Learning the basics of circuit building will help you navigate the future.

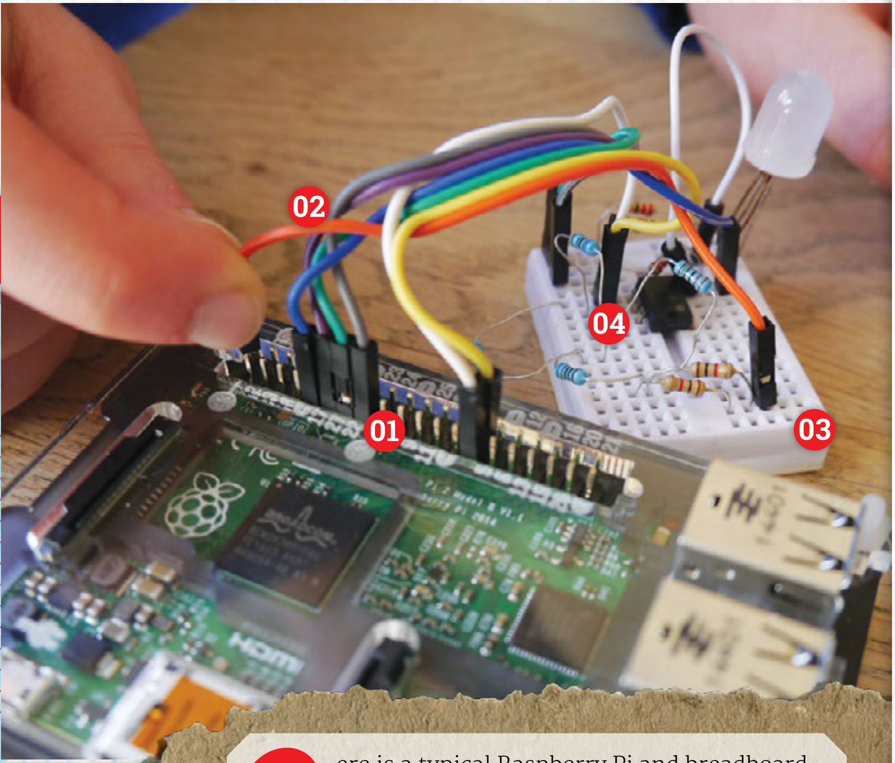
> "Welcome to the wonderful world of electronics with Raspberry Pi"

WHAT IS A HAT?

HATs (Hardware Attached on Top) are pre-built circuit boards with components and parts. They are designed to connect to the 40-pin GPIO header on your Raspberry Pi, and are easy to set up and get started. HATs offer a friction-free way to use electronics, and some (like the Pimoroni Explorer HAT, **magpi.cc/1OAKy46**) add a breadboard to your Raspberry Pi. For this feature, we'll eschew HATs and focus on using pure electronics connected directly to the Raspberry Pi GPIO pins.



RASPBERRY PI PROTOTYPING



Here is a typical Raspberry Pi and breadboard project. Wires from the Raspberry Pi are connected to circuit components. Code on the Pi is used to control the circuit.

01. GPIO PINS

The GPIO pins on a Raspberry Pi are used to get input from components, such as a button being pushed or the value from a sensor. They can also be set to send output, and turned on or off (set to 'HIGH' and 'LOW') to control components.

02. JUMPER LEADS

Jumper leads are used to connect the Raspberry Pi GPIO pins to components on the breadboard (and to hook up components to each other).

03. BREADBOARD

A breadboard is used to create a prototype of a circuit. Components are placed into the holes on the board, and jumper leads are used to connect them to each other (and to the Raspberry Pi GPIO pins).

04. ELECTRONIC COMPONENTS

These come in all shapes and sizes and perform a wide variety of tasks. Common components include LEDs, resistors, potentiometers, PIR sensors, and switches.

KITS

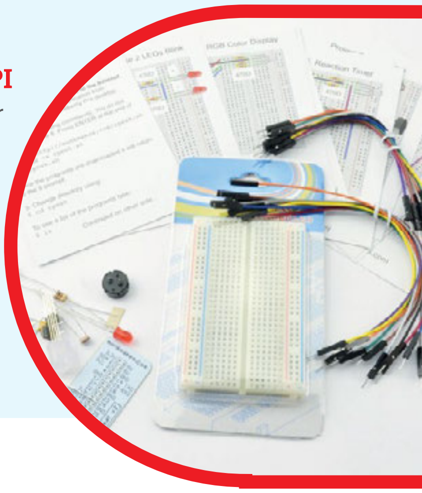
It's common to buy kits containing a variety of components. The parts can be used with Raspberry Pi resources and other tutorial guides

CAMJAM EDUKIT #1 AND #2

There are two CamJam EduKit electronics packs available from The Pi Hut (**magpi.cc/2yV1VQY**). The first (£5) contains a breadboard, LEDs, buttons, and buzzers. The second (£8) also has temperature and PIR sensors. Both come with jumper wires and downloadable worksheets for you to follow.

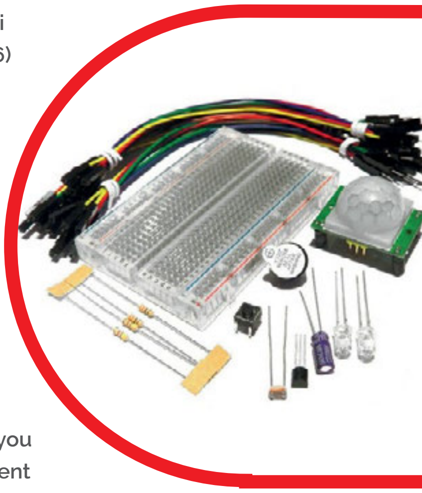
ELECTRONICS STARTER KIT FOR RASPBERRY PI

Simon Monk is a popular writer of electronics guides (he's written several guides for *The MagPi*). His starter kit (**magpi.cc/2eC95jz**, £10/$17) has all manner of parts and ten projects to create.



RASPBERRY PI YOUTUBE WORKSHOP KIT

This pack from ModMyPi (**magpi.cc/2mjJJdB,** £16) contains a transparent breadboard (so you can see the connections inside). It also has LEDs resistors, buttons, buzzers, sensors, and jumper wires. It's one of the more comprehensive kits and comes with a complete range of YouTube videos to help you work with each component (**magpi.cc/2jj0ftf**).

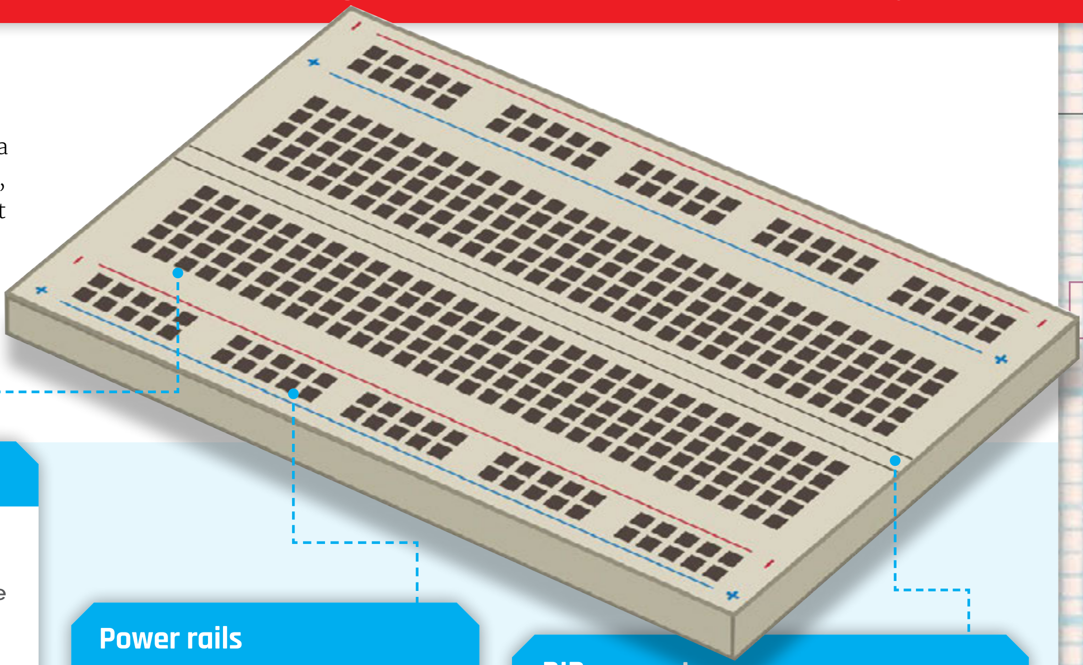


KIT YOU'LL NEED

Handy kit you need to start learning electronics with a Raspberry Pi

BREADBOARD

A breadboard is a plastic slab with a bunch of holes in it. At first glance, it looks pretty unfathomable, but it soon becomes easy to understand. See our How To Use a Breadboard tutorial (magpi.cc/2jB52Vz). We use a Raspberry Pi Breadboard (Half Size) in our diagrams (magpi.cc/2q34ZFz).



Terminal strips

In the middle of the breadboard are (typically two) columns of holes, called terminal strips. These are typically arranged in rows of five, and each hole in a group of five is connected to the one next to it. Components placed into a row of five holes are connected to each other. So you can connect circuit components to each other without having to physically wire them together.

Power rails

Down each side of the breadboard are two rows of holes (running all the way along the board). These are positive and negative rails and are typically connected to a battery, or the power (3v3/5V) and ground (GND) pins on a Raspberry Pi.

DIP support

In the centre of a breadboard, between the two columns of terminal strips, is a gap. This is usually the exact size to place a DIP (dual in-line package) chip. These IC chips can straddle the central division with a row of pins falling into holes on either side.

JUMPER LEADS

Jumper leads are used to connect the GPIO pins on a Raspberry Pi to the components on your breadboard. Some jumper leads have female ends that connect to the GPIO pins on the Raspberry Pi, and male ends that connect to the holes on a breadboard. Typically you'll need a selection of female-to-male and male-to-male jumper leads. magpi.cc/2muhzg4

LEDs

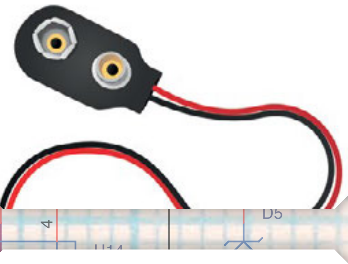
If you're absolutely new to electronics on a Raspberry Pi, then buy a bunch of LEDs and resistors (along with your breadboard and jumper leads). These enable you to set up a simple circuit that lights up an LED with code, like the one in this online tutorial: Physical Computing with Python. magpi.cc/2mjGMJZ

RESISTORS

LEDs are delicate little things. If you put too much current through them, they will pop (sometimes quite spectacularly). To limit the current going through the LED, you should always use a resistor in series with it. The resistor can be anything over about 50 ohms. magpi.cc/2mtKKjs

9V BATTERY SNAP

Your Raspberry Pi outputs a steady 3.3V from some of its GPIO pins and it's perfectly possible to get by without a battery. But we find a 9V battery with a snap cable a handy way to add power to a circuit and test out components.
magpi.cc/2mshkSG

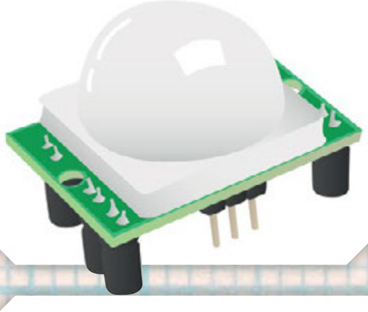


PUSH BUTTON

A push button is a great way to test out input on a Raspberry Pi (where you create code that responds to physical button pushes). Some push buttons have two pins; others – like this one – have four pins and are designed to fit across the DIP support bridge at the centre of a breadboard.
magpi.cc/2mqTgj2

PIR SENSOR

A PIR (passive infrared) sensor is used to detect motion. Alongside buttons, they are a good way to add input to a project. You can create a script that detects movement and responds accordingly.
magpi.cc/2msRLkJ

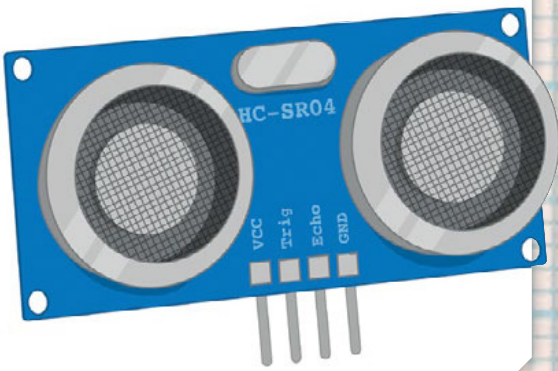


PIEZOELECTRIC SPEAKER

This buzzer is used to add basic sound to a project. While it's not going to rock the house, a piezoelectric speaker is a handy way to get feedback from a project.
magpi.cc/2muqIFy

ULTRASONIC DISTANCE SENSOR

The ultrasonic distance sensor can determine how far it is from a solid object. It works by sending out a burst of ultrasound. This sound will travel through air but reflect back (echo) off hard surfaces. The sensor detects the echo. They're often found on robots and other Raspberry Pi projects.
magpi.cc/2mtgpS1

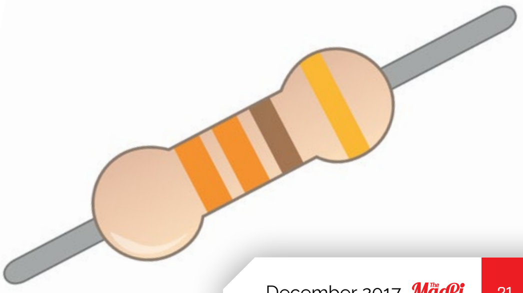


READING RESISTORS

Use the colour bars on a resistor to read its values. One of the ends will have a stripe on it, typically silver or gold. Make sure that is on the right. Now look closely at the bands in the middle,

The first two bands represent numbers and the third is a multiplier. In the resistor below, we have two orange (3) bands and a brown (10) multiplier. So our resistor is 33×10, or 330 ohms). The final gold band shows that it has a tolerance, or precision, for the resistor. Ours is 5%.

Band 1: Number	Band 2: Number	Band 3: Multiplier	Band 4: Tolerance
BLACK: 0	BLACK: 0	BLACK: 1	BROWN: 1%
BROWN: 1	BROWN: 1	BROWN: 10	RED: 2%
RED: 2	RED: 2	RED: 100	GOLD: 5%
ORANGE: 3	ORANGE: 3	ORANGE: 1000	SILVER: 10%
YELLOW: 4	YELLOW: 4	YELLOW: 10000	NOTHING: 20%
GREEN: 5	GREEN: 5	GREEN: 100000	
BLUE: 6	BLUE: 6	BLUE: 1000000	
VIOLET: 7	VIOLET: 7	GOLD: 0.1	
GREY: 8	GREY: 8	SILVER: 0.01	
WHITE: 9	WHITE: 9		



WIRING DIAGRAMS AND SCHEMATICS

Now you've got a bunch of components, it's time to start putting them together and learning to code

Once you've got a breadboard and some electronic components, you'll want to start wiring them up and learning how they work with a Raspberry Pi.

There's absolutely no shortage of electronic projects out there, and most of them guide you through attaching components to your breadboard, then using software such as the GPIO Zero Python library.

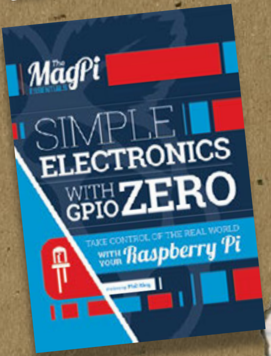
There are two approaches to describing the layout of electrical circuits. The first is to use a wiring diagram (see 'Wiring diagram'). The simple circuit shown below connects a push button and LED to separate GPIO pins on a Raspberry Pi. The Raspberry Pi detects a button push and lights up the LED.

You'll find wiring diagrams like this throughout *The MagPi*, and similar ones used by the Raspberry Pi Foundation and other resources. If you're interested in creating your own, you use a program called Fritzing to make them (**fritzing.org**).

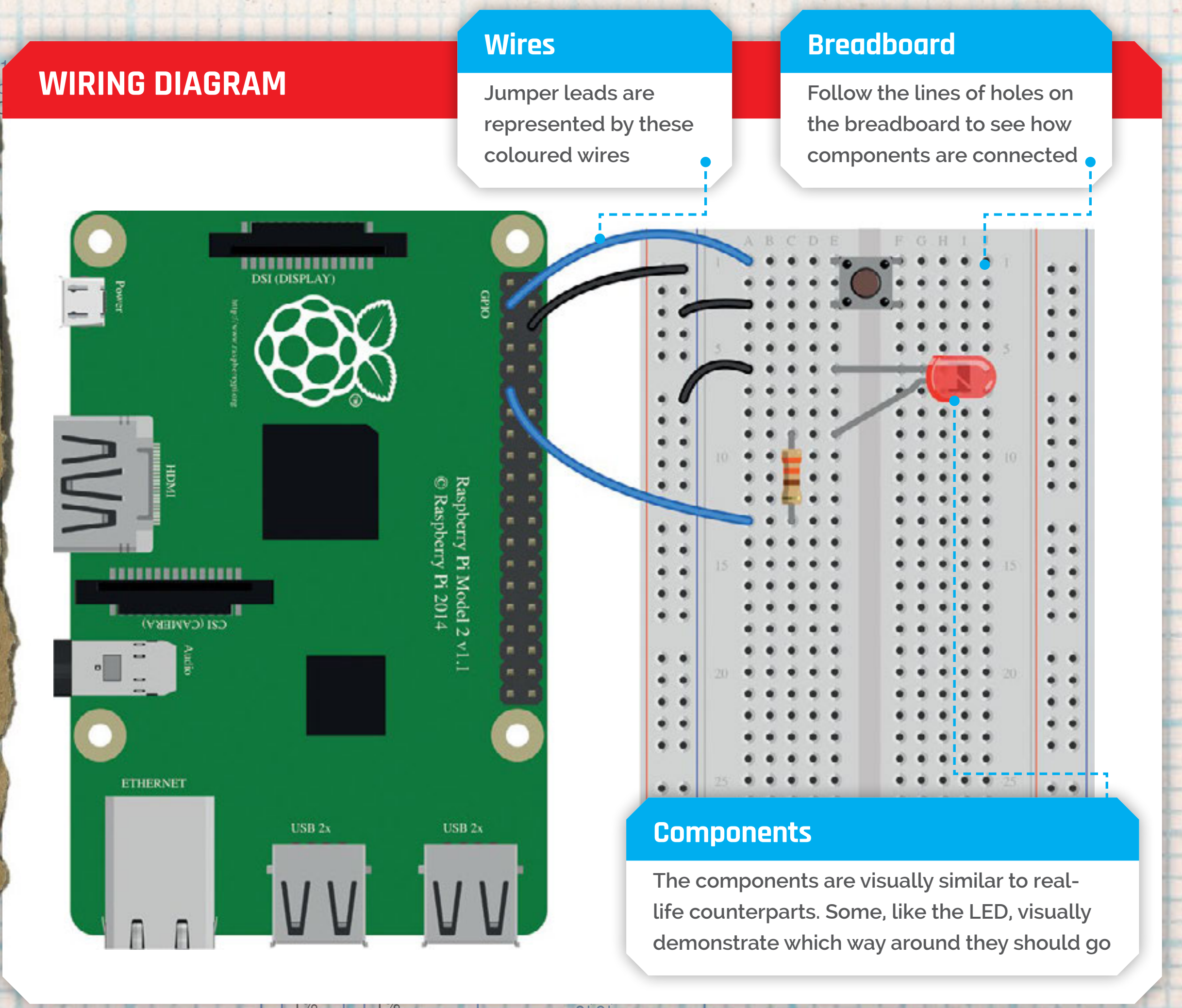
If you want a guide to GPIO pins and a crash course in controlling them with GPIO Zero, take a look at our Beginner's Guide to GPIO Zero in *The MagPi* issue 52 (**magpi.cc/Issue-52**).

SIMPLE ELECTRONICS WITH GPIO ZERO

For more information on using GPIO Zero and learning to wire up circuits from wiring diagrams, take a look out our Essentials Guide to GPIO Zero. **magpi.cc/2bA3ZP7**


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

WIRING DIAGRAM



Wires

Jumper leads are represented by these coloured wires

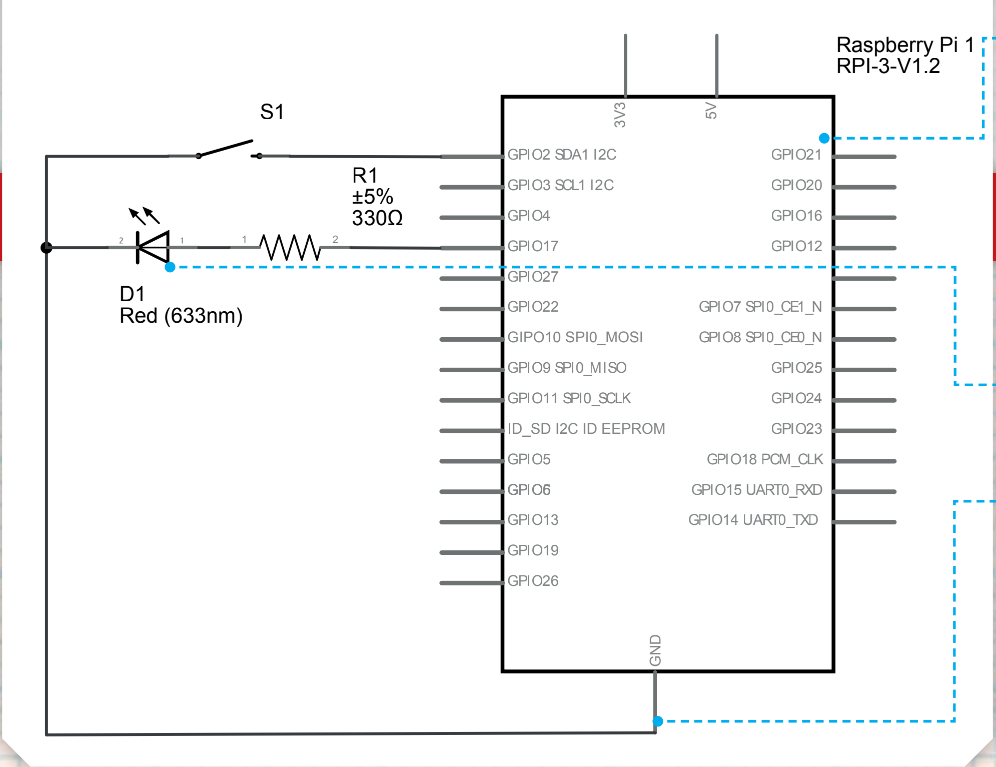
Breadboard

Follow the lines of holes on the breadboard to see how components are connected

Components

The components are visually similar to real-life counterparts. Some, like the LED, visually demonstrate which way around they should go

SCHEMATIC DIAGRAM



The GPIO pins of a Raspberry Pi are represented by this large rectangle. They are not in the same positions as the physical pins on the real board

Components are represented by abstract symbols, like this triangle with arrows (which represents an LED)

Black lines are used to indicate connections between components on the circuit. These can represent jumper wires or components connected to the same rail on a breadboard

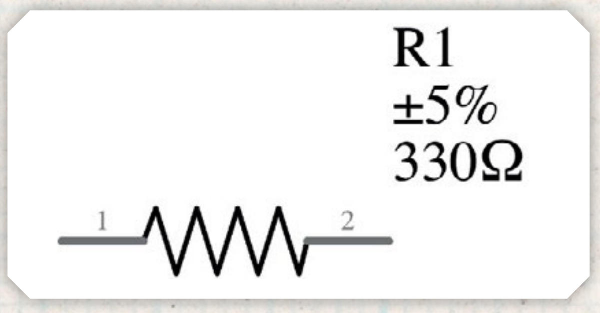
HOW TO: UNDERSTAND SCHEMATIC DIAGRAMS

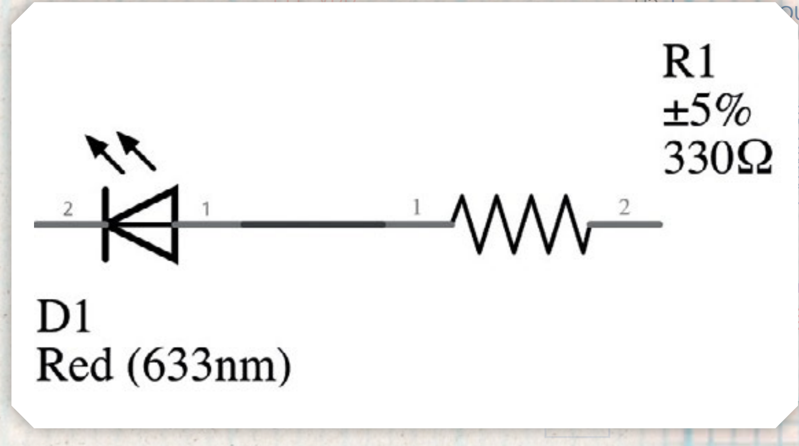
>STEP-01

Components

Schematics describe how components are connected. Unlike wiring diagrams, symbols aren't visually similar to their real-world components and the schematic isn't necessarily visually accurate. It's a guide to how things are connected.

Take the resistor. Instead of being a visually realistic tube with coloured rings, it is indicated by these zigzag lines (we're using the US-style resistor symbol in this feature, in the UK it's often a square rectangle). The properties are often listed alongside as text. Learning to identify components is half the battle. Take a look at the 'Component reference guide' (page 25), which compares symbols with their real-world counterparts.

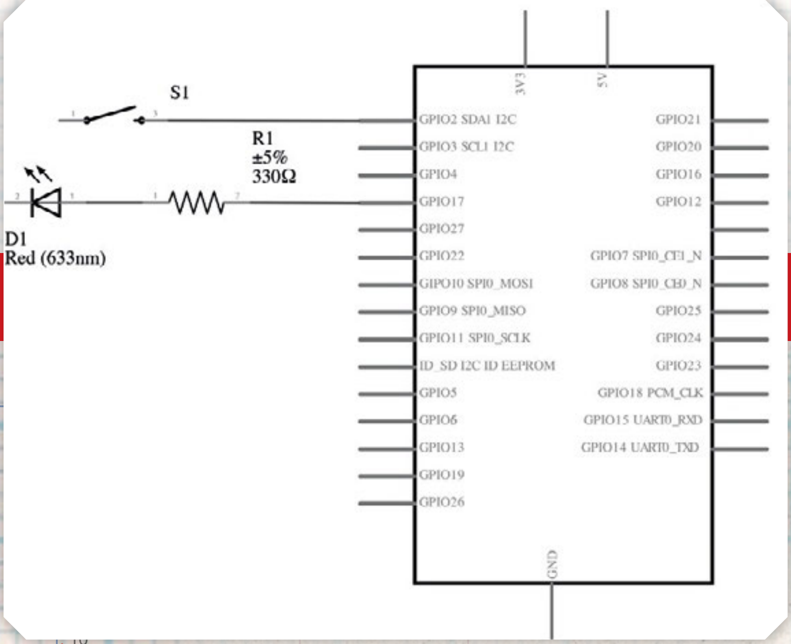




>STEP-02

Connections

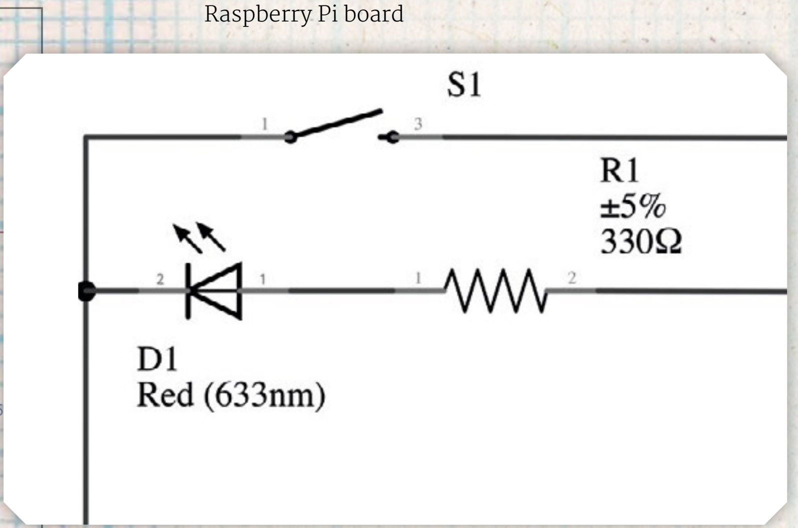
Now we've added a second component to our resistor, an LED. This is represented by a triangle (the symbol for a diode, it indicates that flow can only travel in one direction). The LED symbol also has two arrows pointing outwards, representing light being emitted. We join the two components together with a single black line. This indicates that they are connected (on our prototype breadboard they could be in the same row of holes, or connected with a jumper lead).



>STEP-03

Integrated circuits

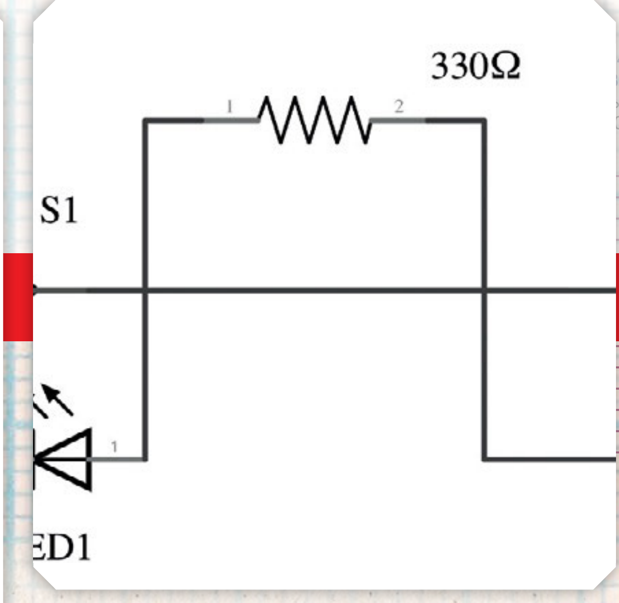
Our Raspberry Pi is displayed as an IC (integrated circuit) box. This large rectangle has pins all around it representing the pins on the GPIO header. We've connected our resistor and LED to GPIO 17 (as written inside the IC box). Like other components, the IC schematic is an abstraction and doesn't resemble the physical layout of the pins. Use a pin guide (**pinout.xyz**) to locate the physical pins on your Raspberry Pi board



>STEP-04

Wires and nodes

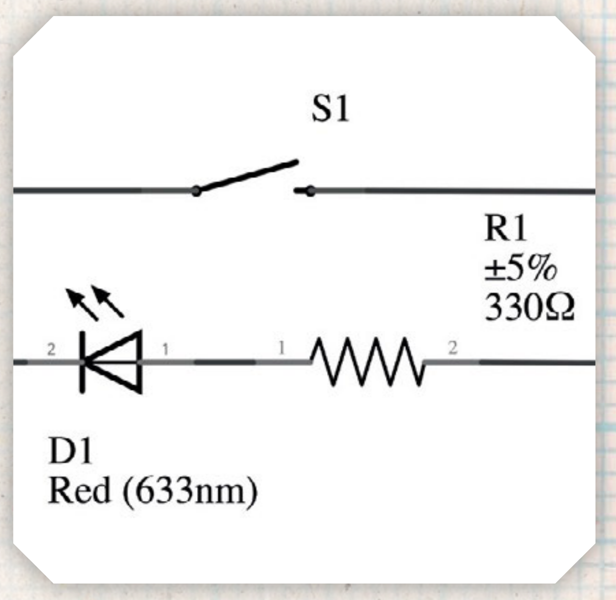
Sometimes, wires form a junction that connects multiple components together. Our button is on a separate GPIO pin to the LED and resistor, but both join together and connect to a single ground pin. Wires connecting at a junction are indicated by large circles called 'nodes'. When you see this circle, it means wires in the circuit connect at this point.



>STEP-05

Crossed wires

In a well-designed schematic, parts will be kept separate and wires will not cross each other. This isn't always possible with more complex schematics, and sometimes designs just aren't as clear as they could be. If you see two wires crossing each other without a node circle, that means the wires are just passing by (and aren't connected).



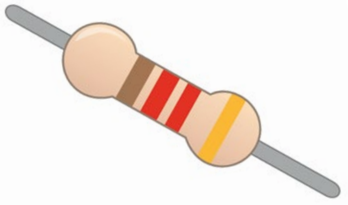
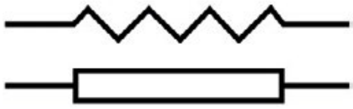
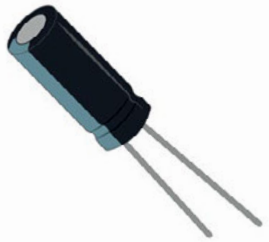
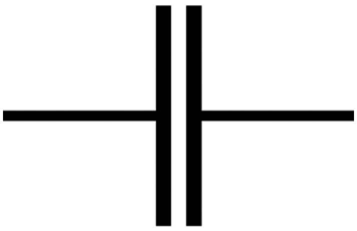
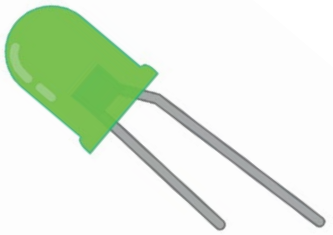
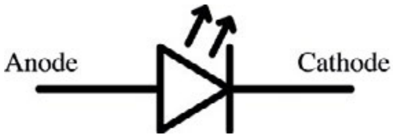
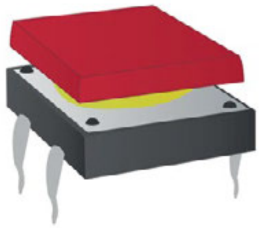
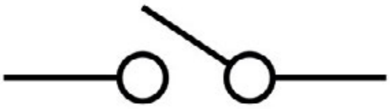
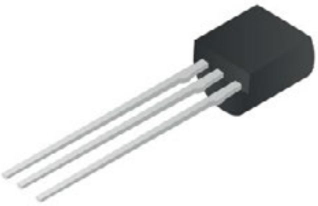
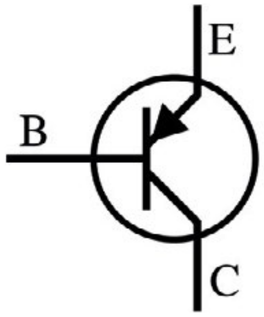
>STEP-06

Reference designator

Now that you know how to identify wires and components on the schematic, you simply need to identify each component and the order they are connected in. Component names are normally a letter and a number. So your first resistor is R1, the second one R2, and so on. You can view a list of all the common reference designators on Wikipedia (**magpi.cc/2mqvyUf**). Happy prototyping.

COMPONENT REFERENCE GUIDE

Here are some common components along with their reference designator letter and both visual wiring graphic and schematic diagram.

Component	Reference designator	Wiring image	Schematic symbol
Resistor	R		
Capacitor	C		
Diode (LED)	D		
Switch	S		
Transistor	Q		

Schematic symbols courtesy of Wikipeda.org

MAKE A GPIO MUSIC BOX

This project shows you how to wire up buttons to your Raspberry Pi's GPIO pins and then use them to play sounds with a simple Python application

YOU'LL NEED

- Raspberry Pi
- Breadboard
- 4 × Tactile switches to make buttons
- 5 × Male-to-female jumper leads
- 4 × Male-to-male jumper leads

In this tutorial, we'll use several push buttons to make a GPIO music box that triggers different sounds when we press different buttons.

For this, we'll make use of GPIO Zero's **Button** class to assign sound samples to buttons. While we have provided a wiring diagram, we suggest you try following the schematic first, using your new-found knowledge. If this proves troublesome, check the schematic against the wiring diagram.

>STEP-01

Get some sounds

Before we start building our GPIO music box circuit, we'll need to prepare some sound samples for it to play. First, open a Terminal window and create a directory called **musicbox** for this project. Then create a directory called **samples** within **musicbox**.

We need to source some sound samples. While there are many public domain sounds to be found online, for this example we'll use some of Scratch's built-in percussion sounds, already present on the Pi. Copy the Scratch percussion sounds to the **samples** directory.

```
  mkdir musicbox
  mkdir musicbox/samples
  cp /usr/share/scratch/Media/Sounds/
Percussion/* /home/pi/musicbox/samples
```

>STEP-02

Add a button

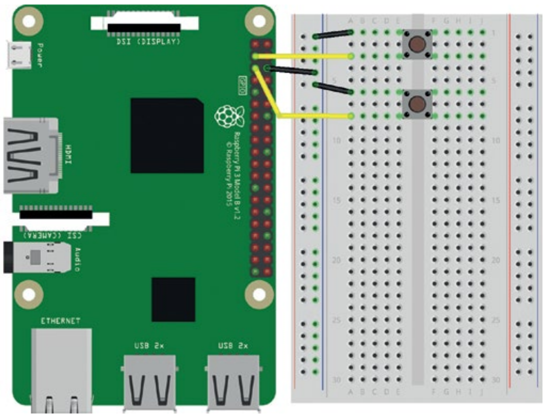
Now follow the schematic and wire up the first button. Place the button so it straddles the central groove of

Remember, the switches are not wired to each other. This junction points out that both are connected to a single ground pin

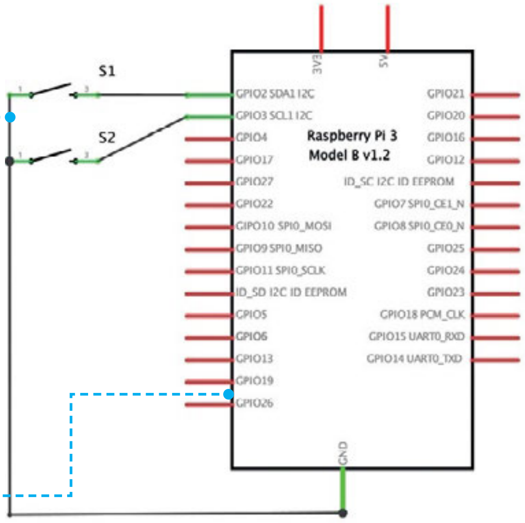
The GPIO pin positions on the schematic diagram don't match those on the physical board. Use **pinout.xyz** if you need guidance on the pin position

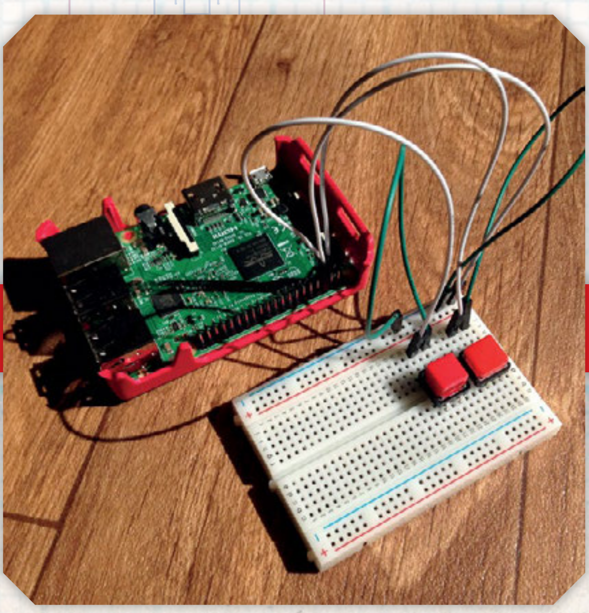
WIRING DIAGRAM

Try to wire up the circuit using the schematic below. If you find it difficult to follow, check this diagram. More instructions are on the Raspberry Pi resources website (magpi.cc/2z44wrC)

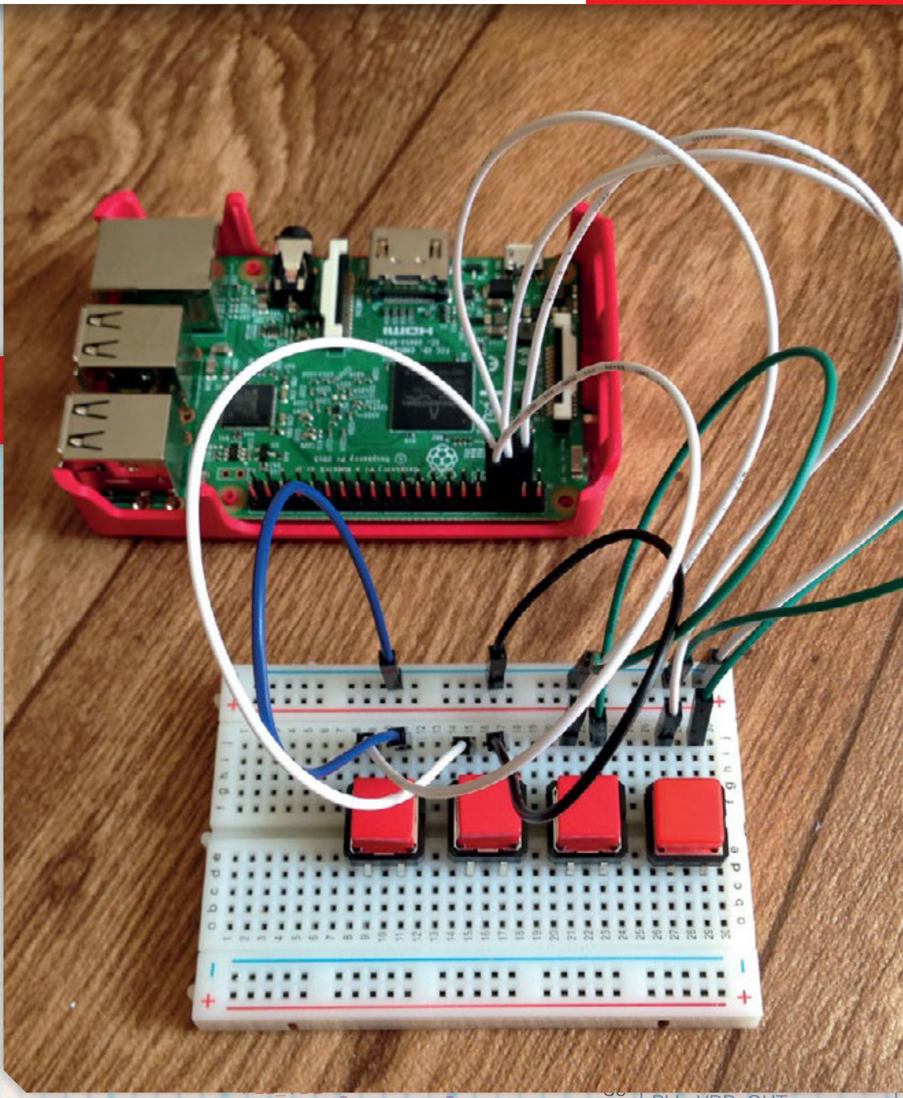


GPIO MUSIC BOX SCHEMATIC





Above Each time you press a button, the assigned sound sample will play through a connected speaker



Right Extra buttons can easily be added to the circuit to play more sounds assigned in the Python code

the breadboard. One leg is connected to the GPIO 2 pin, and the other to the common ground rail on the breadboard, which in turn is wired to a GND pin.

>STEP-03

Add a second button

Now add a second button to the circuit, again following the schematic. Place it on the breadboard as before, and wire it up to GPIO 3 and the common ground rail.

Now open a new file in Python 3 IDLE, enter the code from **gpio_musicbox.py**, and save it in your **musicbox** folder. Run the program and push the buttons. You should hear a noise with each button push.

>STEP-04

Add more buttons

The way we have structured the program makes it easy to add extra buttons and assign them to sound samples. Just connect each button to a GPIO number pin (not any other type) and the ground rail, as before. Then add the GPIO pin numbers and sounds to the dictionary, as in the following example:

```
sound_pins = {
    2: Sound("samples/DrumBizz.wav"),
    3: Sound("samples/CymbalCrash.wav"),
    4: Sound("samples/Gong.wav"),
    14: Sound("samples/HandClap.wav"),
}
```

Add more buttons and redraw the schematic with switches connected to the Raspberry Pi IC.

gpio_musicbox.py

Language
>PYTHON

DOWNLOAD:
magpi.cc/2BaGLLM

```
import pygame.mixer
from pygame.mixer import Sound
from gpiozero import Button
from signal import pause

pygame.mixer.init()

button_sounds = {
    Button(2): Sound("samples/drum_tom_mid_hard.wav"),
    Button(3): Sound("samples/drum_cymbal_open.wav"),
    Button(4): Sound("samples/elec_bell.wav"),
    Button(14): Sound("samples/elec_hi_snare.wav"),
}

for button, sound in button_sounds.items():
    button.when_pressed = sound.play

pause()
```

MAKE A RANGE-FINDER

Link together an ultrasonic distance sensor and seven-segment display to measure distances

YOU'LL NEED

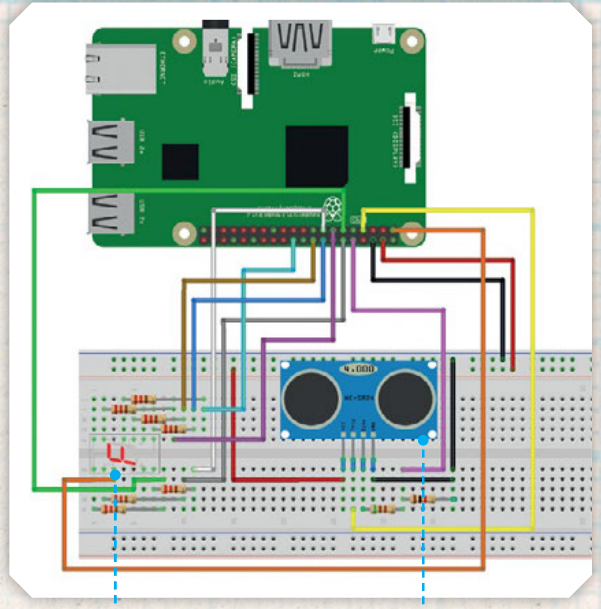
- HC-SR04 ultrasonic distance sensor **magpi.cc/2mtgpS1**
- Broadcom 5082-7650 7-segment display **magpi.cc/1YnqQPl**
- 9 × Resistors (7 × 220 Ω, 1 × 512 Ω, 1 × 1 kΩ) **magpi.cc/1YnqU1r**

The HC-SR04 ultrasonic distance sensor is a great favourite with Pi robot-makers. It works by bouncing ultrasonic sound off an object and timing how long it takes for the echo to return. This time is then converted into a distance which can be displayed on a single-digit, seven-segment display. Here you can acquire the skills to handle inputs and outputs. You also get to use seven-segment displays, which are quite cool in a retro kind of way.

>STEP-01

Lighting the display

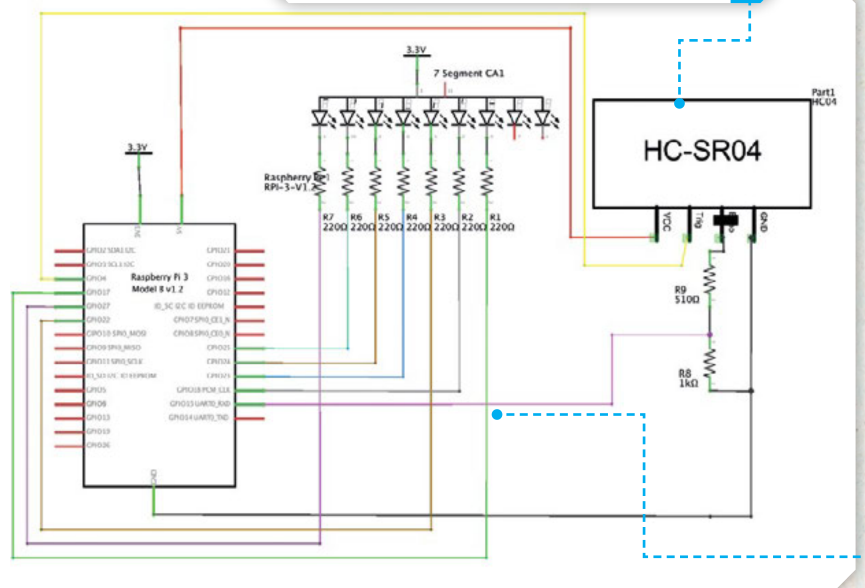
The seven-segment display is a collection of LEDs, with one LED corresponding to one of the segments. All the anodes (positive ends) are connected together; this should be connected to the 3.3 V power (3V3 pin). The cathodes (negative ends) should be connected to a resistor to limit the LED current and the other end of the resistor to a GPIO pin. To turn the LED on, all you have to do is set the GPIO output to be LOW (0 V) and it will complete the circuit for the current to flow.



You can use any seven-segment display, but alternatives might have a different pinout

Ultrasonic distance sensor: gives out a pulse proportional to any reflecting object in front of it

The HC-SR04 ultrasonic distance sensor is displayed as an IC with four pins



Remember that two points crossing are not linked unless there's a large circular node. On this schematic, the lines are coloured differently to help guide the 'ratsnest' of wires

>STEP-02

Generating a seven-segment pattern

The display consists of four bars or segments that can be lit. By choosing the segments to light up, you can display a number from 0 to 15, although you have to resort to letters (also known as hexadecimal numbers) for this. There are, in fact, 128 different patterns you can make, but most are meaningless. In our code (**range_finder.py**), a list called **seg** defines which pins are connected to which segments, and another list called **segmentPattern** defines the LED pattern for each number.

Above The project in action; the Pi is measuring how far it is to the Raspberry Pi 3 box

>STEP-03

Displaying numbers

The **display** function sets up the segments to display any single-digit number passed to it. First, it sets all the segments to off, and then if the number is less than 16, it goes through the entries in the **segmentPattern** list for that number and turns on the appropriate segments. Note that we can still use **on** and **off** even though they're not powered by individual GPIO pins, as the LEDs were declared to GPIO Zero as **active_high = False**.

>STEP-04

The distance sensor

The HC-SR04 distance sensor signals its reading by producing an output pulse that the Pi tries to measure. The GPIO Zero library measures this pulse and converts it into a distance by returning a floating-point number that maxes out at 1 metre. We then multiply this number by 10 to give decimetres. Next, we convert it to an integer to get rid of the fractional part of the measurement, so we can show it on our single-digit display.

range_finder.py

Language

>PYTHON

DOWNLOAD:
magpi.cc/2BaGLLM

```
# displays the distance in decimetres on a 7-segment display
from gpiozero import LED
from gpiozero import DistanceSensor
import time

seg = [LED(27,active_high=False),LED(25,active_high=False),LED(24,active_high=False),
      LED(23,active_high=False),LED(22,active_high=False),LED(18,active_high=False),
      LED(17,active_high=False)]

segmentPattern = [[0,1,2,3,4,5],[1,2],[0,1,6,4,3],[0,1,2,3,6],[1,2,5,6],[0,2,3,5,6], #0 to 5
                 [0,2,3,4,5,6],[0,1,2],[0,1,2,3,4,5,6],[0,1,2,5,6],[0,1,2,4,5,6], #6 to A
                 [2,3,4,5,6],[0,3,4,5],[1,2,3,4,6],[0,3,4,5,6],[0,4,5,6] ] #B to F

sensor = DistanceSensor(15,4)

def main() :
   print("Display distance on a 7-seg display")

   while 1:
       distance = sensor.distance * 10 # distance in decimeters
       print("distance",distance)
       if distance >= 10.0:
          distance = 16.0
       display(int(distance))
       time.sleep(0.8)

def display(number):
    for i in range(0,7):
       seg[i].off()
    if number < 16:
       for i in range(0,len(segmentPattern[number])):
           seg[segmentPattern[number][i]].on()

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

>STEP-05

Building the project

For this build, we used a dinky little breadboard shield from Dtronixs. This allowed for a much more compact arrangement than a conventional breadboard, although you can of course still use one. As the HC-SR04 uses a 5 V power supply, the pulse we have to measure is also nominally 5 V. Therefore, this has to be cut down to 3.3 V by using a 512 Ω and 1 kΩ resistor voltage divider.

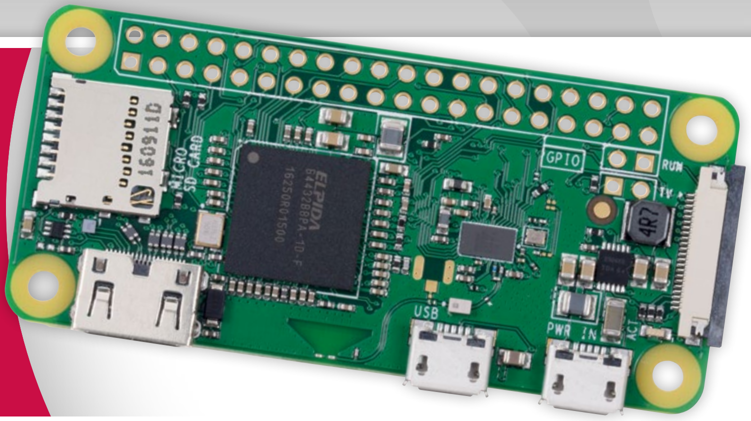
>STEP-06

Using the sensor

The distance to the reflective object is updated every 0.8 seconds. If this is greater than a metre, then the display will be blank. A display of 0 indicates that the object is less than 10 cm away. Don't touch the sensor, otherwise its readings will be wrong. Also, as it has quite a wide beam, you can get reflections from the side. If several objects are in the field of view, then the distance to the closest one is returned.

SUBSCRIBE TODAY AND RECEIVE A

FREE PI ZERO W



PLUS OFFICIAL PI ZERO CASE WITH 3 COVERS

AND FREE CAMERA MODULE CONNECTOR AND USB / HDMI CONVERTER CABLES

Subscribe in print for 12 months today and receive:

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

Other benefits:

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

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
ELECTRONICS STARTER GUIDE
Serious fun with circuits, components & jumper leads
INCOMING! AIY PROJECTS VISION KIT
RETRO GAMING
ADD A SIM CARD
WIN! ROBOT ARM & FLICK HATS
BUILD AN ARCADE MACHINE
JUNIOR PI PROJECTS
WIN! £300 ROBOT KITS
LEGO and Raspberry Pi

Pricing

Get six issues:

£30 (UK)

£45 (EU)

$69 (USA)

£50 (Rest of World)

Subscribe for a year:

£55 (UK)

£80 (EU)

$129 (USA)

£90 (Rest of World)

FREE PI ZERO W!

Get three issues:

£12.99 (UK) (Direct Debit)

$37.50 (US) (quarterly)

How to subscribe:

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

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

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

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

Mag#64

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 □□/□□/□□

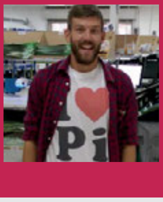
I would like my subscription to begin from issue .... (month + year)

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.



MART DRAKE-KNIGHT

Mart and his brother Rob want to make fashion more sustainable and they love organic cotton and renewable energy.
rapanuiclothing.com

RAPANUI CLOTHING

Mart Drake-Knight and his brother **Rob** have put dozens of Raspberry Pis at the heart of their super-efficient clothing factory

Quick Facts

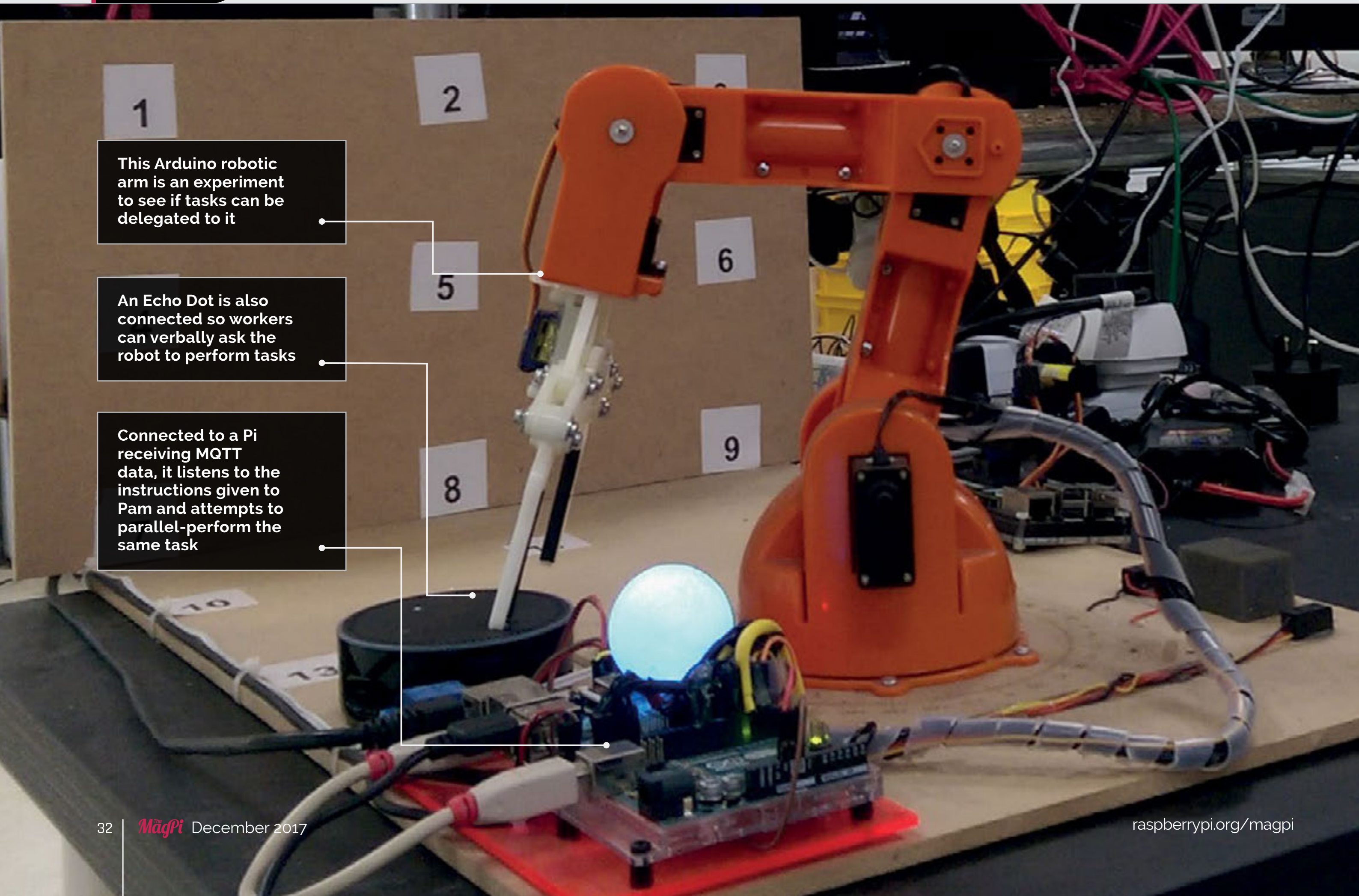
- Rapanui's factory plugs into the Internet of Things
- All workstations and machinery are Pi-powered
- The UI has been made mistake-proof
- IBM's Node-RED is free and open source
- Fire destroyed the previous factory in 2016

When brothers Rob and Mart Drake-Knight struggled like mad to find a job in 2008, they could so easily have reached a point where they would have to sell the shirts off their backs. Luckily they decided to spend £200 setting up an environmental clothing company on the Isle of Wight instead, growing it from a garden shed to a large factory in a former Co-op supermarket.

Today, the whole operation at Rapanui is powered by renewable energy and a creative, hackathon, DIY spirit. More specifically, it also makes use of nearly 100 Raspberry Pis to perform many different tasks. "Some are connected to sensors or machinery that we've made in our machine shop," says Mart. "Some do simple stuff like open windows and help people solve problems."

Other Pis drive touchscreens and workstations and they enable staff to prioritise work. Put together, they have made Rapanui's manufacturing process very efficient while allowing the firm's eco-friendly and sustainable clothing to be affordable. By automating the dull parts of the job and giving managers a solid overview of what's happening in the factory and where, Rapanui can go as far as offering tailored services: you could get the company to make a single T-shirt with your picture or logo or order 1,000. Either way, you could have the garment the following day.

In order to bring the technology in the factory together, Rapanui uses a lightweight messaging protocol for

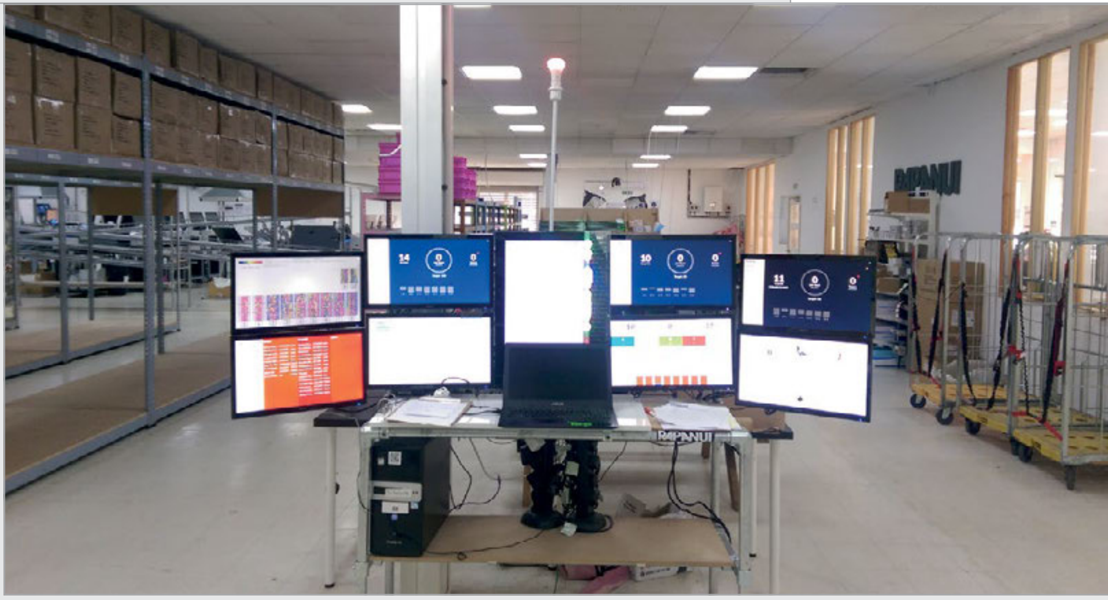


This Arduino robotic arm is an experiment to see if tasks can be delegated to it

An Echo Dot is also connected so workers can verbally ask the robot to perform tasks

Connected to a Pi receiving MQTT data, it listens to the instructions given to Pam and attempts to parallel-perform the same task small sensors and mobile devices called MQTT and this allows the Pis to broadcast their behaviour to other machines or human interfaces. Using Node-RED, the hardware devices, APIs and online services are wired together.

"We use Node-RED in live production to broadcast order data between two specific machines that are time-critical," Mart tells us. "More recently we used Node-RED for cool stuff like joining an Alexa API to our factory systems so that you can ask the factory questions such as how many jobs are pending, which lets us direct staff to the areas where they are needed. It also controls some robotic equipment we're working on."



Above Running Rapanui's factory is a complex business, hence the mass of Pi-powered screens at this computer terminal

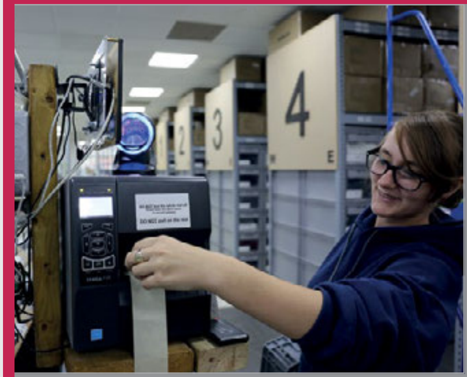
Creating tech jobs

The firm has since built on that system to produce **Teemill.com**, which lets users create an online store, sell T-shirts, and take the profits of each sale. "People can join and access the API directly via Teemill," Mart says. But just as importantly, Rapanui's approach to the development of its clothes has created desirable tech jobs on the Isle of Wight, with the company able to show its young employees how to code though programming apprenticeships.

"The Pi has given us a great way to build confidence at the early stage in people who want to start a career with us in computing," continues Mart. Indeed, today the company invests a lot of its time developing automated, real-time production systems which allows products to be made in the seconds after they are ordered. "The breakthrough has massively reduced waste and cost," Mart tells us. "And since we've doubled in size every year and the Raspberry Pis are modular, they have scaled with us."

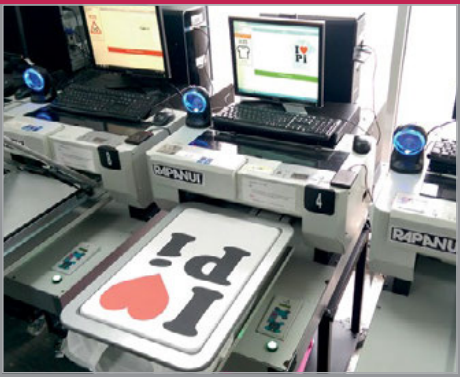
> "The whole operation is powered by renewable energy and a creative, hackathon, DIY spirit"

GETTING THE PROCESS DOWN TO A 'T'



>STEP-01
Creating an order

A customer can create any product they like at Teemill.com. When a T-shirt is ordered, Holly picks the order. The Pi runs the interfaces, which are connected via Node-RED/MQTT.

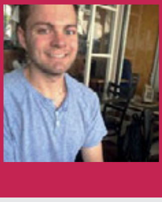


>STEP-02
Printing the T-shirt

The lights change colour, alerting workers to the location of the next task. The workstation interfaces broadcast the machine's status to the MQTT broker, which is run centrally on a Pi.

>STEP-03
Sending it out

When the T-shirt is ready, Pam scans and packs it. The interface tracks her work, broadcasting the status back to the systems. The customer is notified that the order is on its way.



AARON TAINTER

Aaron is a software engineer working at eBay. He runs Hacker House with hardware wizard Davis Lau.

magpi.cc/2yhuMdI

Quick Facts

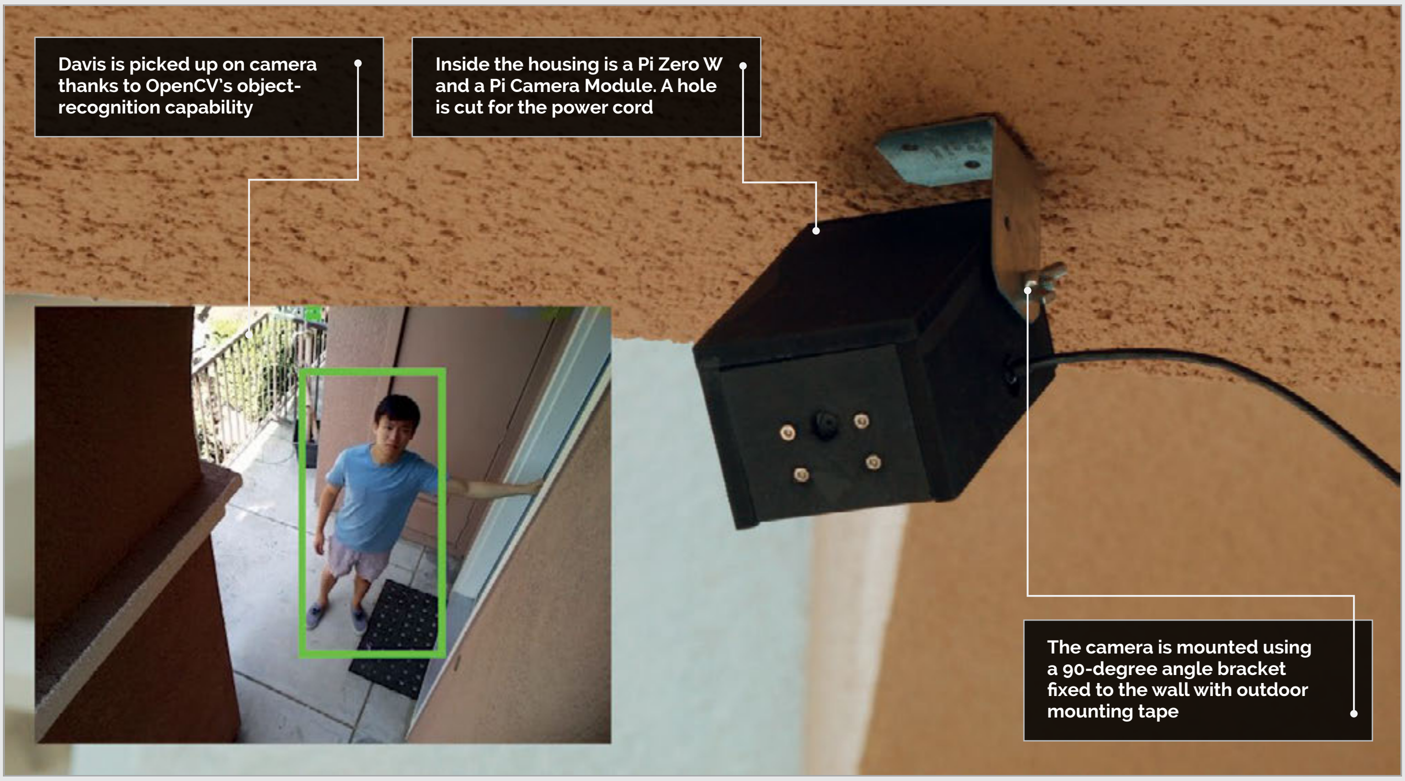
- The project requires lots of drilling and woodwork
- The camera's housing is made in two connecting parts
- It uses OpenCV – a computer vision software library
- The camera's object detection is better at lower angles
- Aaron and Davis list their other projects at **hackerhouse.site**

SMART SECURITY CAMERA

Be a Pi spy with Aaron Tainter and Davis Lau's amazing camera that makes use of the open-source computer vision library OpenCV

There are plenty of home security cameras currently on the market, many allowing you to stream live footage over the internet and receive alerts whenever someone wanders past. But rather than buy one of those, two makers, Aaron Tainter and Davis Lau, spied an opportunity. Combining a Raspberry Pi Zero W, a Pi Camera Module, and the amazing OpenCV app, they decided to make their own.

The inspiration to do so felt obvious. "We're known for mixing artificial intelligence and computer vision with our DIY projects, so a smart security system seemed to align well with our genre," explains Aaron. The pair have previously created numerous home-automation projects, ranging from an indoor gardener to a smartphone-connected door lock. "I think our security camera resonates with people. The object-recognition part is really fascinating."



Davis is picked up on camera thanks to OpenCV's object-recognition capability

Inside the housing is a Pi Zero W and a Pi Camera Module. A hole is cut for the power cord

The camera is mounted using a 90-degree angle bracket fixed to the wall with outdoor mounting tape

Above One drawback is that the camera can only recognise one object at a time

OpenCV (**opencv.org**) makes adding computer vision to Pi projects rather straightforward. It allows you to train the Raspberry Pi to identify objects and react to them, which is perfect given that Aaron and Davis wanted their smart security camera to feature facial and body detection. The plan was for the camera to send an email notification whenever it detects someone, although it can be set up to stream live video through a webcam server too.

"We also wanted to add a servo-gimbal so that you could tilt and pan the camera from the web app," says Aaron. "But because we typically give ourselves a month to complete a project like this and have to film and edit the build videos for our YouTube channel, by the time we finished building the camera body we didn't have enough time left for it."

> I think our security camera resonates with people

Smart housing

That seems a shame, but then building the camera's MDF housing was a tricky process, with various holes needed in the right places to gain access to the Pi. "We also wanted to make the top removable so that the components were more accessible but compact enough that it could be mounted on a ceiling," says Aaron, explaining that a CAD model was built first so that they could get the measurements just right when they cut the materials.

Fortunately, writing the code for the web app proved easier and they were thankful OpenCV exists and could (after a staggering eight hours) be installed on the Pi Zero. "Without OpenCV, the project wouldn't have been possible," Aaron explains. "It contains thousands of lines of code that were written over the course of many years and it gives individual developers like us the chance to produce something awesome without years of research and development."

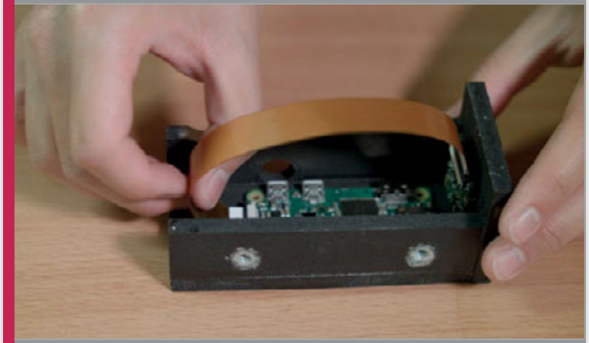
The pair still needed to tinker with it. "We had to make a few optimisations to get the camera working at a high frame rate because OpenCV eats up the CPU," Aaron recalls. There was also an issue with the video feed being choppy, solved by parallelising a few processes in the code. But before long, the camera was allowing the viewing of live streams, detecting objects and sending any images it picked up on to a Gmail address.

"The Raspberry Pi Zero is a great piece of hardware because it has a small form factor yet comes loaded with features," says Aaron of the reason why it works so well in the camera. "It was the perfect platform to mount in a small case and run our image-recognition software. I'm surprised how much performance you can get out of that tiny computer."

CREATE YOUR OWN CAMERA

>STEP-01
Build the housing

After designing the housing and creating paper templates for guidance, a ¼-inch (6 mm) MDF board is cut to size. The pieces are then glued and screwed together before being sprayed black.

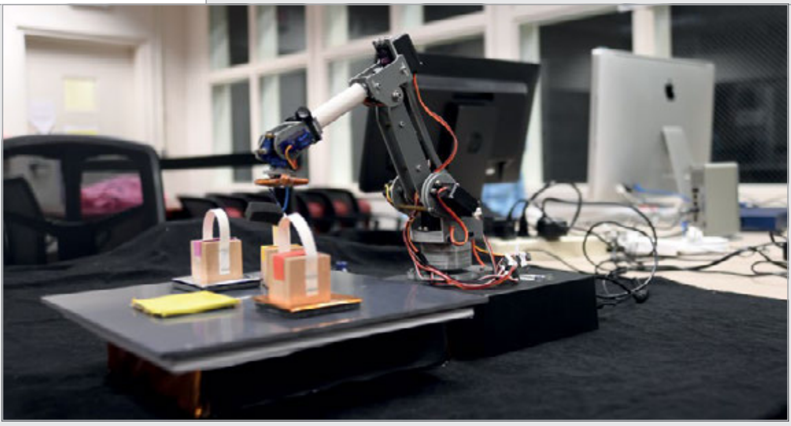


>STEP-02
Insert the bits

The Pi Zero W and Pi Camera Module connect via an Adafruit Pi Zero Camera Cable. Future versions may contain a NoIR Pi Camera: "It's supposed to have exceptional low-light capabilities."

>STEP-03
Sending it out

An MbientLab micro-USB to USB cable powers the device and OpenCV software is used. "I'd like to try different 'image classifiers' to recognise other objects like cats and dogs," Aaron says.



DAN GRIBOK

A robotics intern at NASA's Goddard Space Flight Center, Dan is currently pursuing a graduate degree in Aerospace Engineering at UMD focusing on Telerobotics and Exoskeleton technology.
sspd.gsfc.nasa.gov

Left The manipulator arm tip can also be equipped with a hook to pick up blocks with loops attached

NASA ARM

With the ability to draw any shape, this robot arm is out of this world

Quick Facts

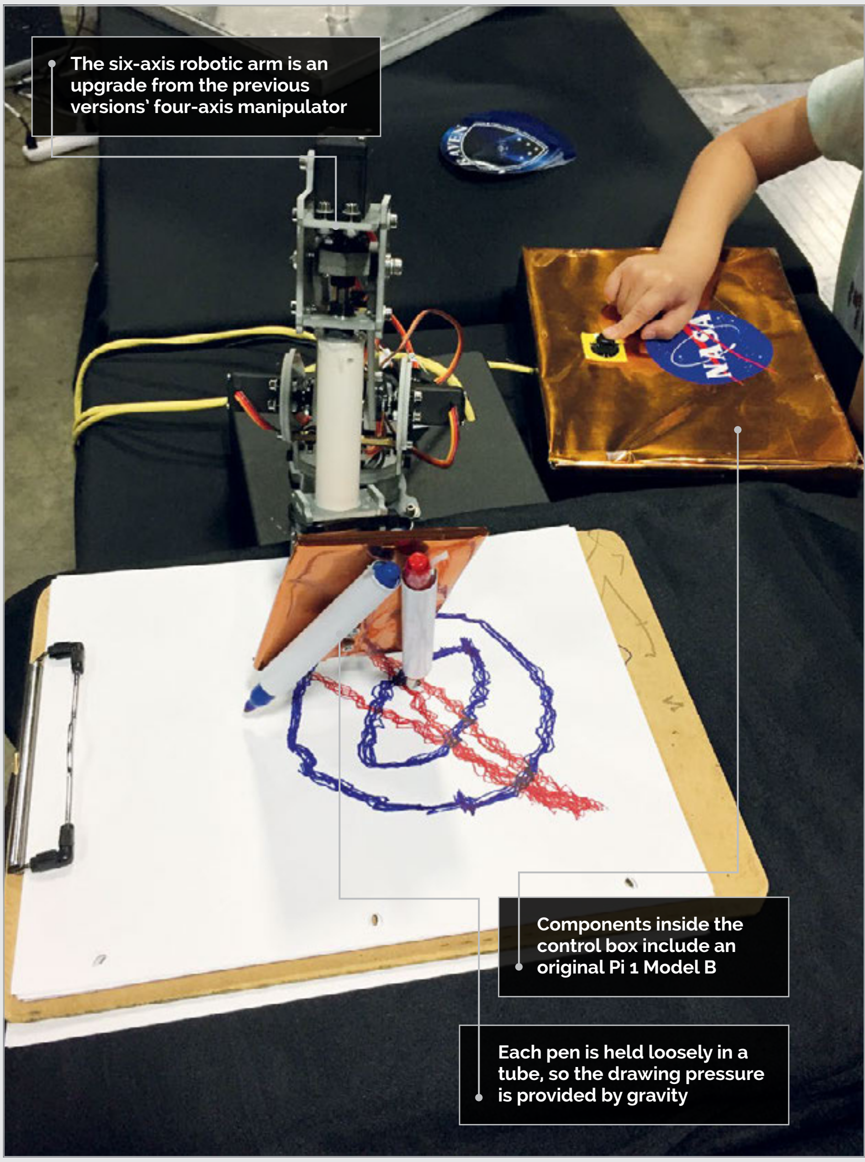
- It can draw any shape using x-y coordinates
- Dan took around a year to refine the software
- The arm can also be controlled manually
- It's powered by an original Pi 1 Model B
- A PCA9685 board drives the arm's servos

A hand reaches out to a shiny gold box emblazoned with the NASA logo. At the press of a button, the connected robotic arm springs into life, lowering a blue Sharpie pen onto a paper pad to draw the circular outline of the famous logo, then an inner 'spacecraft orbit' ellipse, before switching to a red pen to sketch the red chevron (**magpi.cc/2zHatI7**).

This is the Advanced Robotic Manipulator System Tools and Resources – or 'ARM' for short – built and programmed by Dan Gribok, a robotics intern at NASA's Goddard Space Flight Center within the Satellite Servicing Projects Division. Designed to be used at outreach events to educate the public in what NASA is doing with robots, the Pi-powered ARM is a versatile device that can also be controlled manually using an Xbox gamepad to pick up objects using a hook or grabber.

Art for events

Dan tells us that the drawing ability came about from a need to have a constantly running demonstration at events, such as this year's Awesome Con in Washington DC. "So we wanted a demo where you could just push



The six-axis robotic arm is an upgrade from the previous versions' four-axis manipulator

Components inside the control box include an original Pi 1 Model B

Each pen is held loosely in a tube, so the drawing pressure is provided by gravity a button, step back, and everyone could watch all at once. And you can also give away a sheet of paper at the end and have them say, 'Hey, a robot drew this – how cool is that?' "

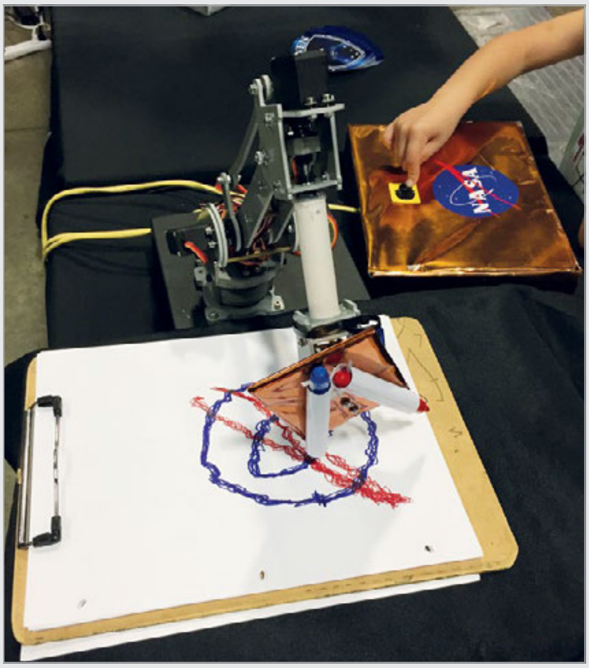
Asked about what ARM can draw, Dan says it depends on what he can program into it and what the hardware can actually replicate. He admits that the drawing action is a little jerky, which is due to the hardware itself: "There's a lot of slop in the joints, a couple of degrees, and some slop in the other mechanisms, which makes it really hard to draw [smooth] curved lines." However, since it understands positions and coordinates, "Any line that you can express as x and y coordinates, so virtually anything, you can program into it and it'll just follow that over and over."

> We wanted a demo where you could just push a button, step back, and everyone could watch

The force of gravity is used to get the pen pressure right, as each Sharpie pen is held loosely in a tube. "We previously tried some other spring-like mechanisms, but we just settled on gravity drawing because it's so much easier and it works perfectly fine."

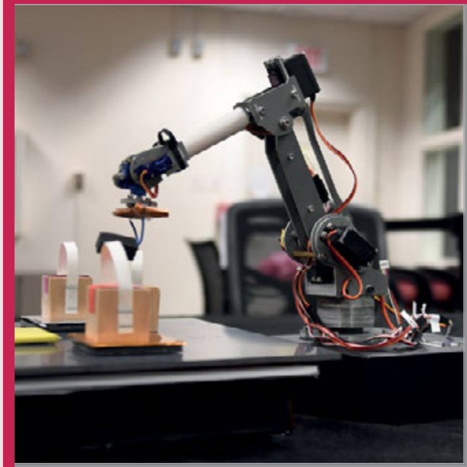
So, what's in the shiny control box? Somewhat surprisingly, along with a PCA9685 I²C servo controller and other components, the ARM uses an original Raspberry Pi 1 Model B – as did the two previous versions of the project, which were based on OWI Edge robot manipulators. In autumn 2016, Dan was granted permission to upgrade the project's robots: "I kept about half the internals of the control box and got completely new robot manipulators."

While the hardware setup was relatively straightforward, the software took a lot longer. "The original robots were programmed in Python, but I noticed that we had performance issues, so we switched entirely to C++." After getting basic robotic functionality working by spring 2017, Dan made improvements over the summer, including adding the drawing capability.



Above Pressing the button causes the arm to start drawing a preprogrammed sketch, in this case the NASA logo

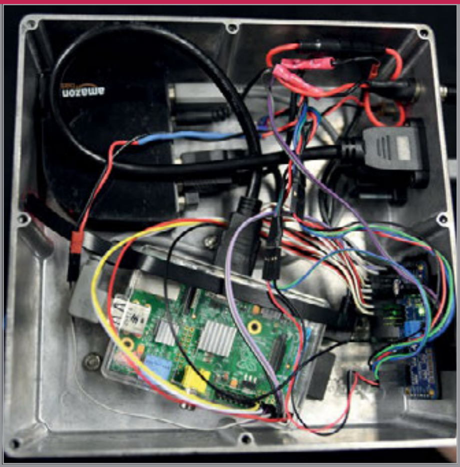
BUILD A DRAWING ROBOT



>STEP-01

Manipulator arm

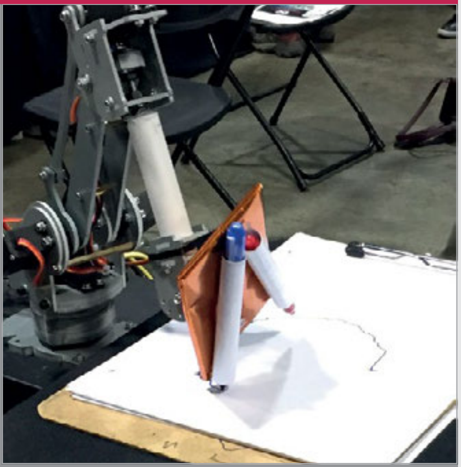
Dan replaced the four-axis OWI Edge arm of the earlier robots with a six-axis Sain Smart manipulator. It features four standard MG996 55g metal-gear servos and two SG90 servos.



>STEP-02

Control box

Inside the control box, a Pi 1 Model B is hooked up to a PCA9685 breakout board to drive the servos. Along with status LEDs and wiring, there's a USB hub to supply power to both the Pi and the arm.



>STEP-03

Drawing mechanism

The tip of the arm rotates to lower one of two Sharpie pens for drawing. Each is held loosely in a tube so that the force of gravity provides enough pen pressure to mark the paper.

MATT BRAILSFORD

A web developer by day and hardware hacker by night, Matt (aka Circuitbeard) is a Barnsley-based maker who loves nothing more than to hack old '80s tech.
circuitbeard.co.uk

TOMY TURNIN' TURBO DASHBOARD OUT RUN

Combining two classics into one amazing racing project, **Matt Brailsford** used a Raspberry Pi to build this incredible '80s mash-up of racing toys

Quick Facts

- The fuel gauge represents the stage time remaining
- The turbo light comes on in high gear
- A small vibrating motor was added for haptic feedback
- Matt tried four different screens before finding one that fit
- Matt hacked the speed to change it to mph from kph

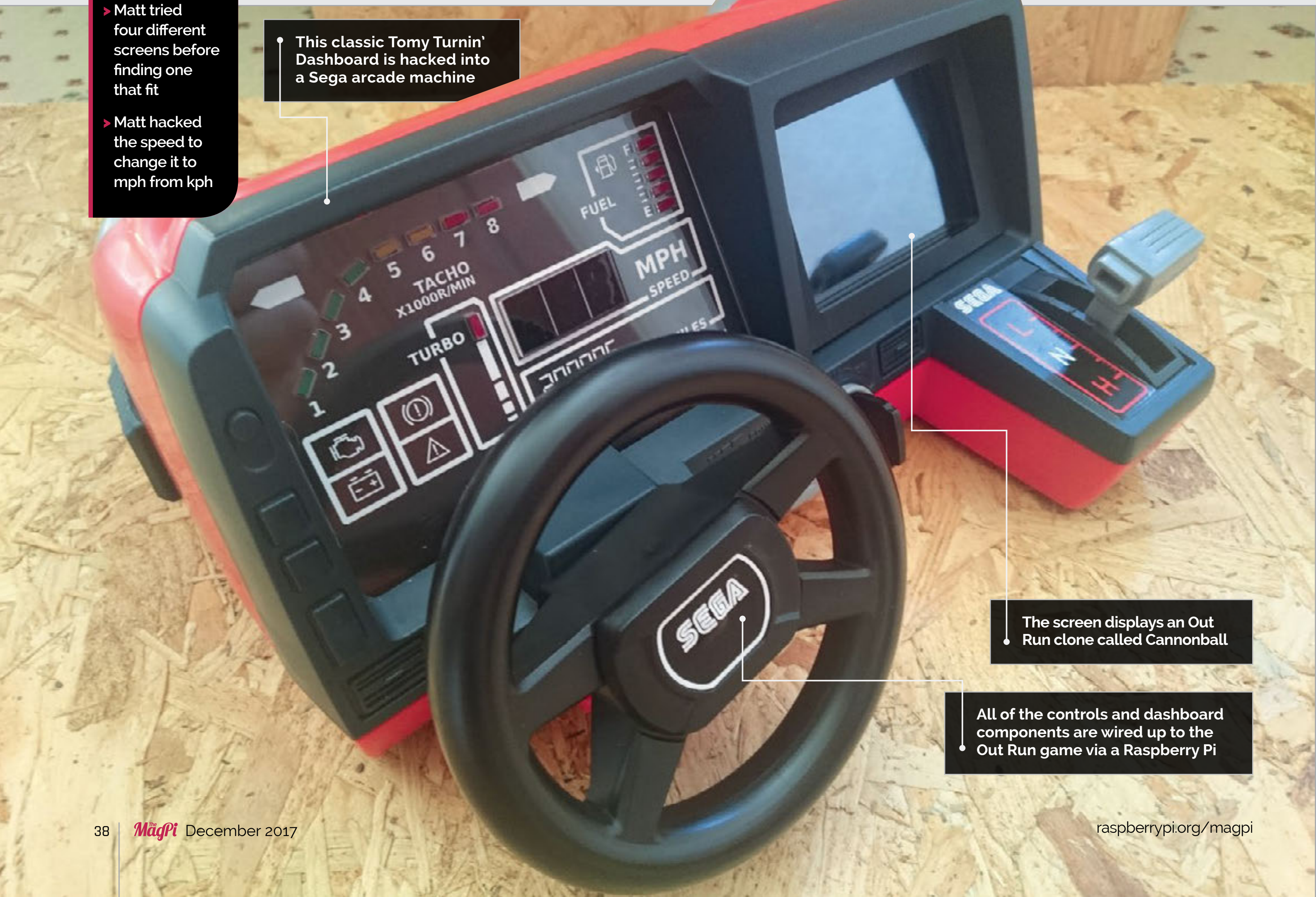
Way back in 1983, a Japanese company called Tomy created one of the most remarkable toys of its generation: Tomy Turnin' Turbo Dashboard was a driving simulator complete with gears, ignition, a working dashboard, a steering wheel, and even a looping display.

Three years later, Sega released arguably (well, we'd argue it) the greatest, coolest racing game of all time: Out Run.

One maker, Matt Brailsford, picked up a Tomy Turnin' Turbo Dashboard on eBay and had a spark of genius: why not turn it into a fully working Out Run arcade machine?

Matt removed the original display from the Tomy Turnin' Turbo Dashboard and replaced it with a modern screen. "I tried quite a few [screens] trying to get one that would fit," says Matt. His first attempt was a screen that worked from the GPIO, but this left few pins for all the other mods. "And the extra processing slowed the game down."

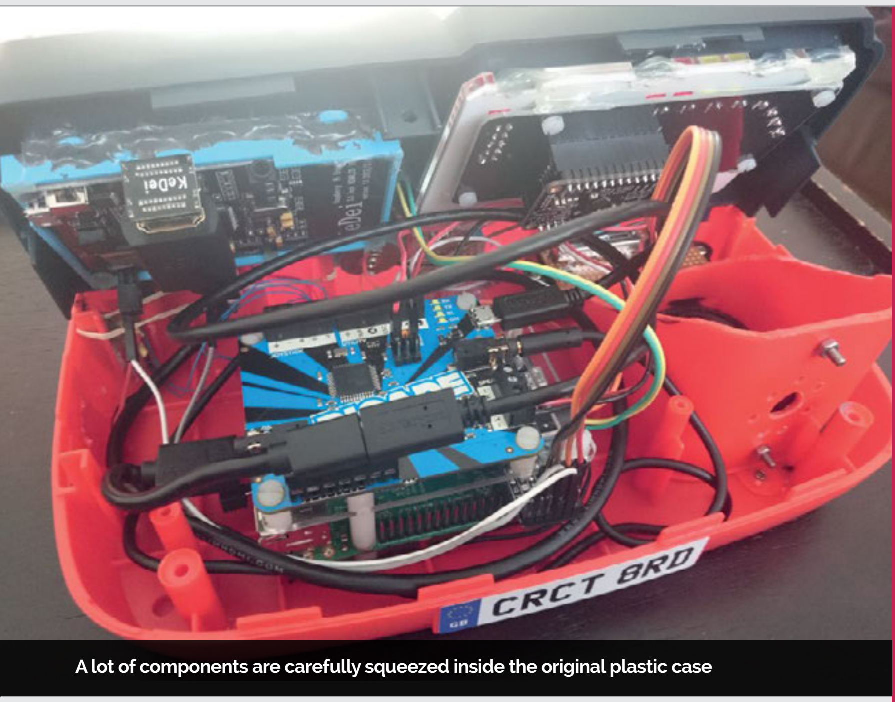
In the end, Matt used a KeDei 3.5-inch HDMI display (**kedei.net**).



This classic Tomy Turnin' Dashboard is hacked into a Sega arcade machine

The screen displays an Out Run clone called Cannonball

All of the controls and dashboard components are wired up to the Out Run game via a Raspberry Pi



A lot of components are carefully squeezed inside the original plastic case

The original ignition key from the game was already wired as an SPST switch, so Matt wired it up to the Raspberry Pi. To manage the safe powering up and down, he used a PowerBlock (**magpi.cc/2mLOf4P**). "I simply hooked up the switch and installed the daemon to watch for the shutdown command and it all worked perfectly. A Picade PCB (**magpi.cc/29DpDCz**) was used to hook up the rest of the controls. Mike tells us it was "really easy to hook these up", especially because the Picade PCB supports analogue controls for the steering wheel.

> Ultimately it was just taking my time and thinking things through, then trying things out

Turn the key

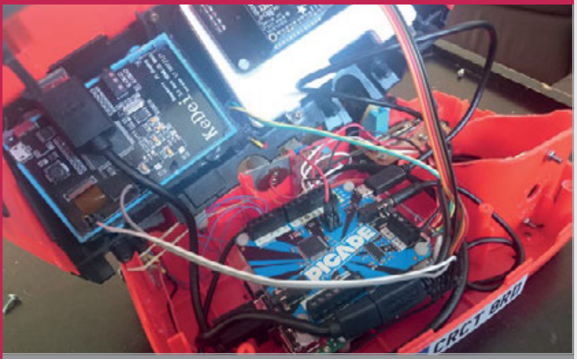
"There was a lot of luck involved as well," explains Matt. "Lucky that the slide potentiometer happened to sit at the perfect height, lucky that there was just enough room for some microswitches to fit down the side of the gear shifter, and lucky I could balance the shifter rubber bands to keep the shifter centred... Ultimately it was just taking my time and thinking things through, then trying things out."

Cannonball running

You might be expecting the Raspberry Pi inside to be running an emulator like RetroPie, but no. It's actually running an Out Run clone called Cannonball (**magpi.cc/2mL3bAi**). Matt explains: "Cannonball is a C++ port of the original Out Run arcade game and luckily ran perfectly on a Raspberry Pi 3. The reason I went with this rather than, for example, MAME is that it allowed me to change the core code and intercept the variables I wanted to use to update the dashboard. You could probably achieve something similar in MAME by inspecting the register values, but having access to the raw source code is way easier."

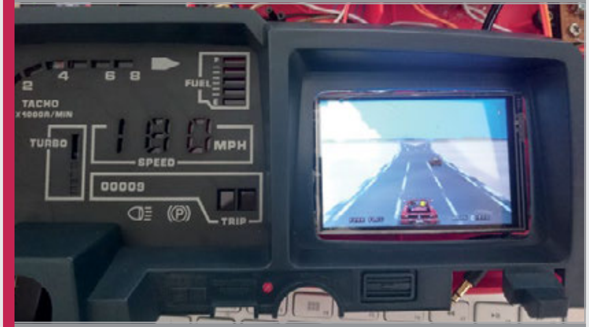
Matt tells us it's fun to play. "It was a little squeaky to start with, but I had just tightened everything up a little too much. It runs nicely and brings back those memories as a kid. It's how I thought it felt to play with back then."

HACKING A TOMY TURNIN'



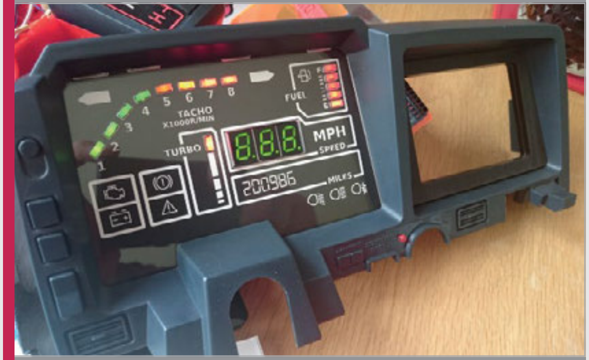
>STEP-01
Hooking up controls

A Picade PCB controller is used to manage the steering wheel and gearstick, while the ignition key is connected directly to the Raspberry Pi via a PowerBlock microcontroller.



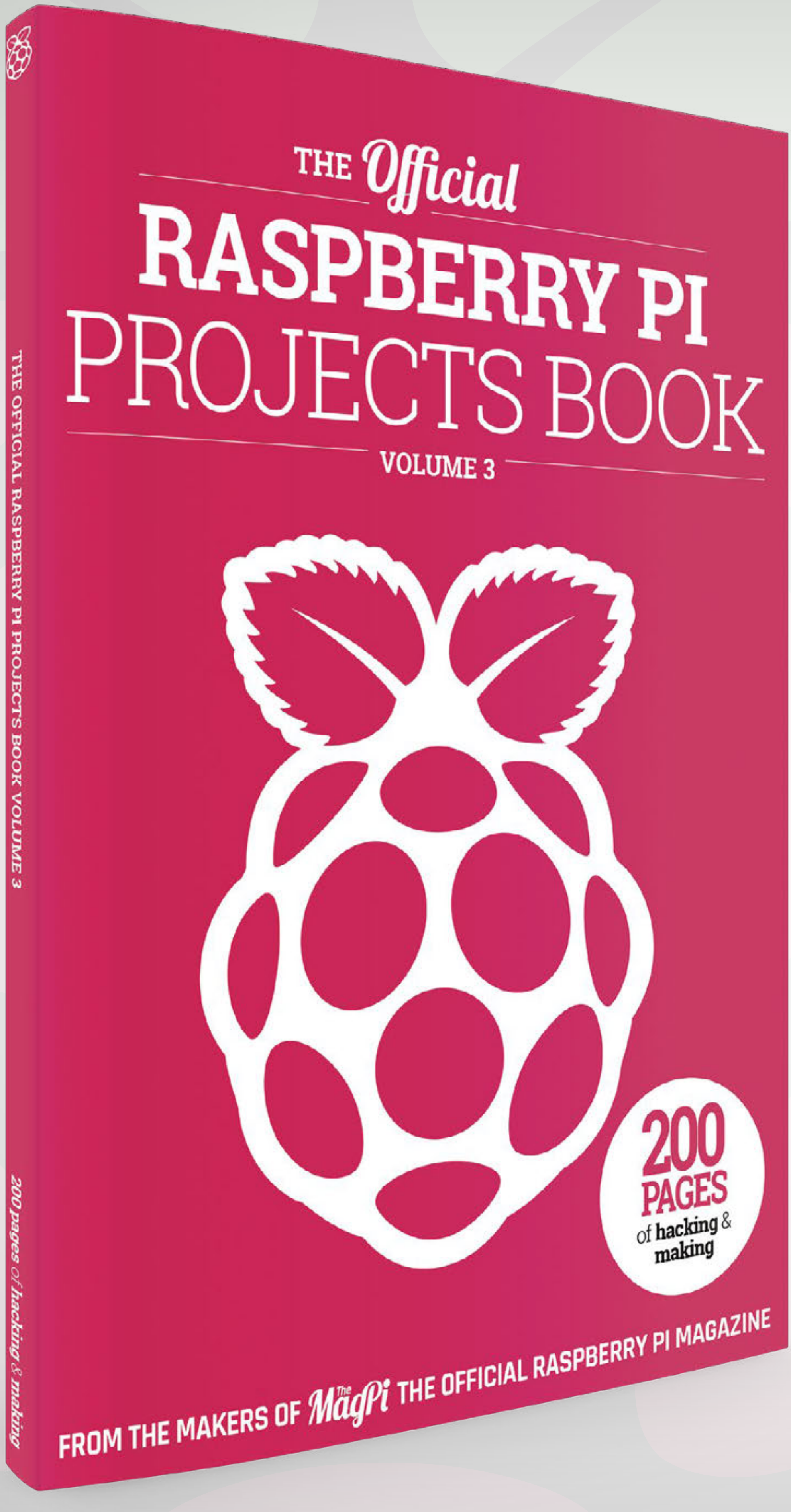
>STEP-02
Fitting a screen

Getting a screen that fitted was the biggest challenge. This KeDei 3.5-inch screen fits directly into the cabinet and connects to the Pi's HDMI socket (leaving the GPIO pins free).



>STEP-03
Connecting the dashboard

Matt ended up recreating the entire dashboard with specific cutouts for LEDs and other components. These respond to the in-game action for an authentic arcade experience.


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 3
200 PAGES of hacking & making
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 3
200 pages of hacking & making

MIKE'S PI BAKERY

MIKE COOK

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

magpi.cc/259aT3X

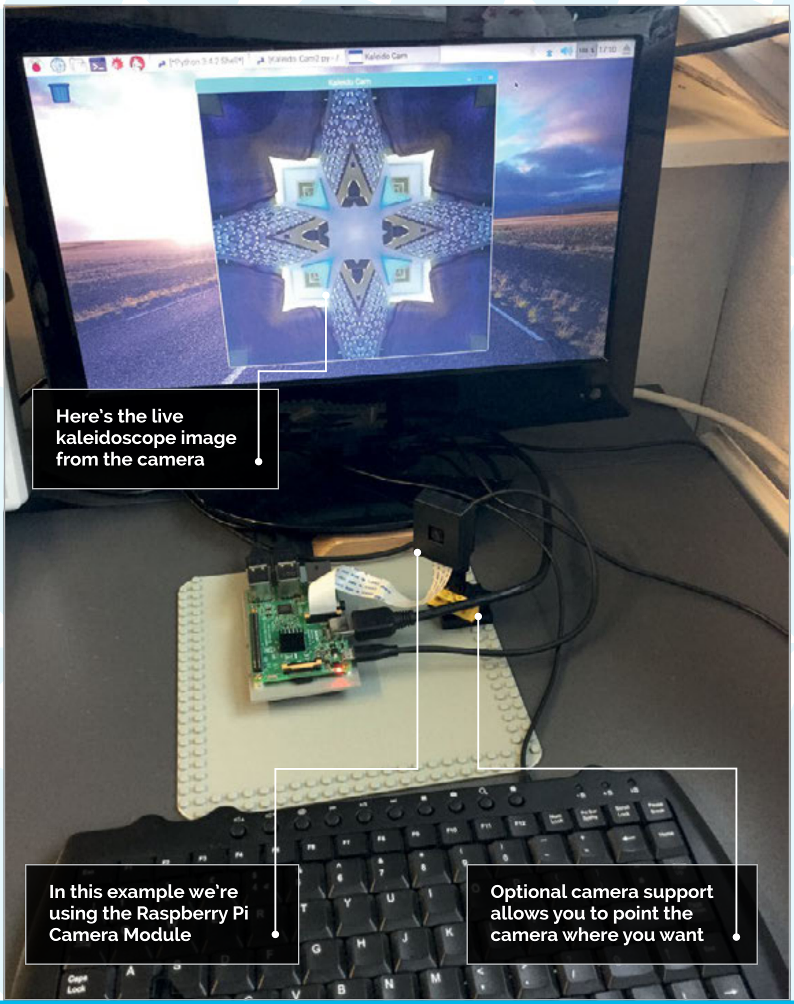
KALEIDO CAM

Turn your web camera or Pi Camera Module into a stunning kaleidoscope

You'll Need

- Webcam or Raspberry Pi Camera Module
- Camera mount (optional)

A kaleidoscope is an age-old toy normally found in your Christmas stocking at least once in your life. Basically, it's two mirrors set at an angle in a tube pointing at a collection of highly coloured blocks. By using a webcam, we can manipulate what it sees into a fascinating moving kaleidoscopic display on your Raspberry Pi. You can do this in real time at a good frame rate using a Pi 3 and a standard webcam; or, for a better-quality result, use the high-resolution Pi Camera Module. **Figure 1** shows a Christmas-themed kaleidoscope image.



Here's the live kaleidoscope image from the camera

In this example we're using the Raspberry Pi Camera Module

Optional camera support allows you to point the camera where you want

The principle

An eight-fold symmetry image can be created by a collection of simple horizontal and vertical flipping of segments of the image. These are fast for the computer to perform as they simply involve shifting bytes from one area of memory to another. **Figures 2A** and **2B** (overleaf) show the steps we have to take in order to create this effect. First, the original image is duplicated and the duplicate is flipped vertically, producing an image with the same width but twice as high as the original. Then a triangular area is removed from each corner; this is done by filling the shape with a colour and making that colour the colour key, causing it to be transparent. This produces the basic quarter segment, which is duplicated three times – applying a horizontal flip and two rotations. Finally, these segments are all drawn together on the screen. They need to be shuffled a bit to fit snugly and eliminate lines on the joins.

The image going into this process can itself be flipped or rotated to change the final pattern. **Figure 3** (overleaf) shows the results of doing this.

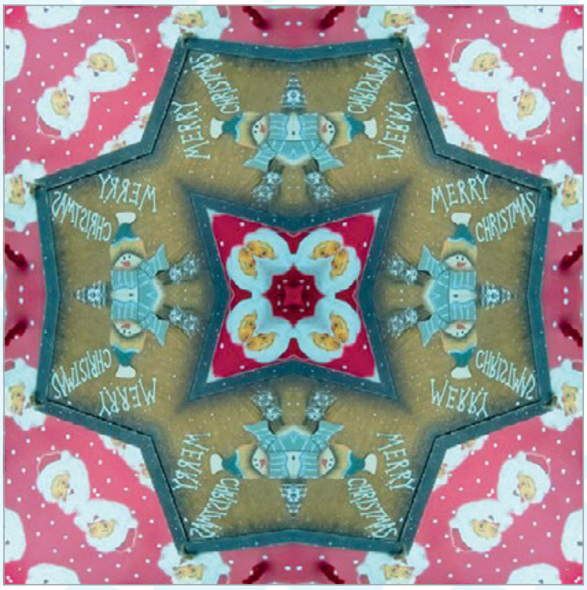


Figure 1 A merry Christmas to you

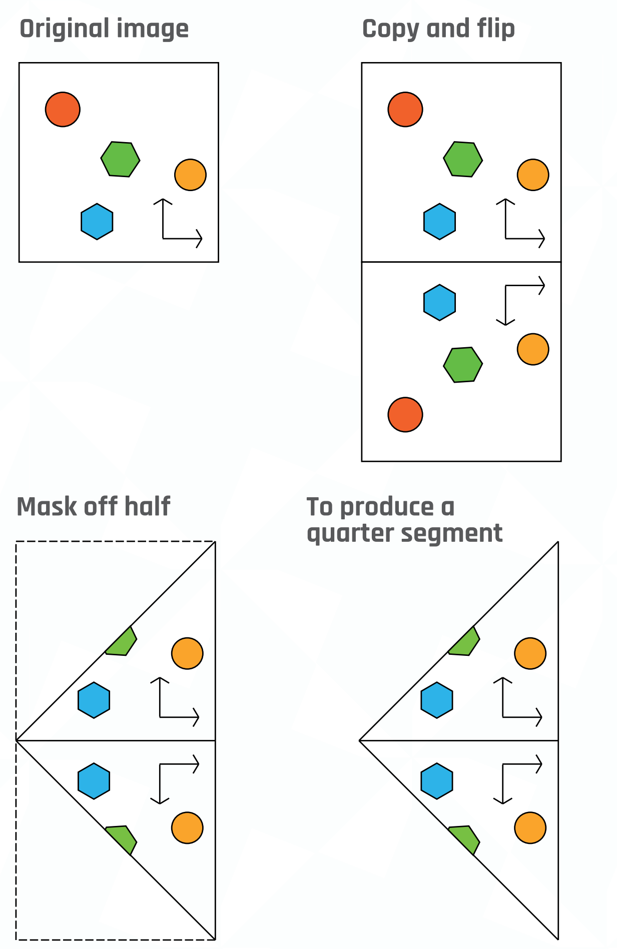


Figure 2A Creating the symmetrical image (see also Figure 2B)

The software

We used the Pygame framework and the **`pygame.camera`** extension. The Pygame documentation warns that this interface might change, or indeed be removed in future versions. This will work with a compatible USB webcam – to check before you buy one, see **magpi.cc/2yOgQsK**. In order to make the Pi's Camera Module compatible with this system, we need to enable the camera drivers in the menu Preferences > Raspberry Pi Configuration and reboot. Now you should type in the **Kaleido_Cam.py** program. There is a line in the software that makes sure the V4L2 drivers have been enabled as well. If you are typing in the program, note that in the line:

```
os.system("sudo modprobe bcm2835-v4l2")
```

...the end number should be lower-case V – four – lower-case L – two.

The software makes use of what Pygame calls 'surfaces'. A surface is simply an area of memory that can hold a bitmap image. Surfaces are transferred to other surfaces by the use of the 'blit' operation, whose name harks back to the old days of computer graphics. The **`showScreen`** function is where all the heavy work takes place. First, it sees if the input image needs to

Kaleido_Cam.py

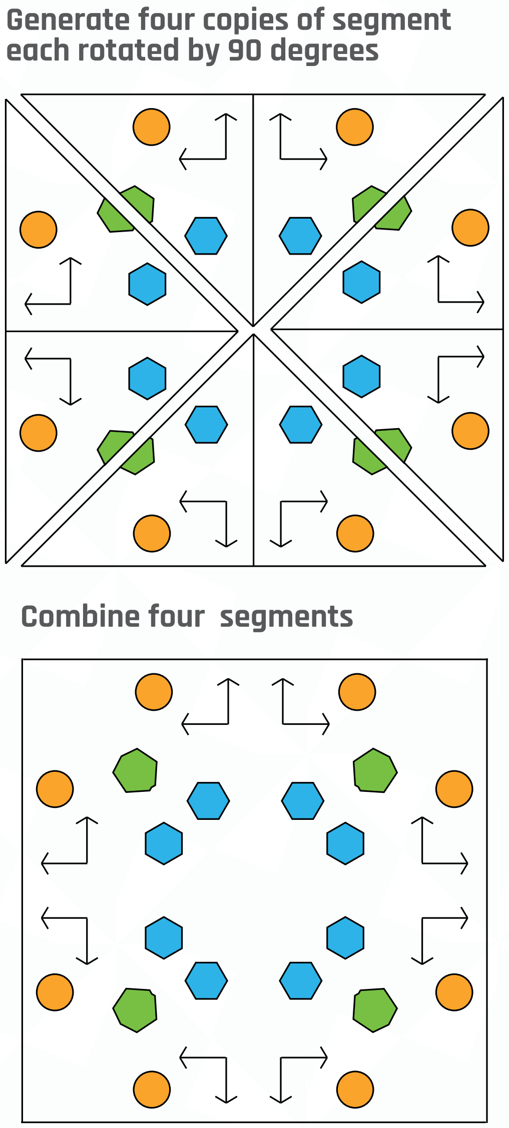
Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

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

```
import pygame, pygame.camera, os
from tkinter import Tk
from tkinter.filedialog import asksaveasfilename

os.system("sudo modprobe bcm2835-v4l2") # needed for Pi camera
Tk().withdraw()
pygame.init()
pygame.camera.init()
os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
pygame.display.set_caption("Kaleido Cam")
pygame.event.set_allowed(None)
pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])

cs = 320 # basic image size
cs2 = cs * 2 # window size
screen = pygame.display.set_mode([cs2,cs2],0,32)

#find, open and start camera
cam_list = pygame.camera.list_cameras()
print(pygame.camera.list_cameras())
webcam = pygame.camera.Camera(cam_list[0],(640,480))
webcam.start()
preRot = 0 ; autoRotate = False
savePath = "" ; shotNumber = 0 ; saveSource = False
flipH = False ; flipV = False

def main():
   while True:
    checkForEvent()
    showScreen()

def showScreen(): #grab image, scale and blit to screen
    global camFrame, preRot
    camFrame = webcam.get_image()
    if autoRotate :
       preRot += 0.5
       if preRot > 360:
          preRot -= 360
       rotFrame = pygame.transform.scale(camFrame,(cs2,cs2)) # make square
       rotFrame = rot_center(rotFrame,preRot) # rotate
       sqFrame = pygame.Surface((cs,cs))
       sqFrame.blit(rotFrame,(0,0),(cs//2,cs//2,cs,cs))
    else :
       sqFrame = pygame.transform.scale(camFrame,(cs,cs)) # make square
    if flipV or flipH: # flip origional image option
       sqFrame = pygame.transform.flip(sqFrame, flipH,flipV)
    # prepare master segment
    primary = pygame.Surface((cs,cs2))
    primary.blit(sqFrame,(0,0))
    primary.set_colorkey((0, 255, 0))
    primary2 = pygame.transform.flip(primary, False,True)
    primary2.blit(sqFrame,(0,0))
    primary2.set_colorkey((0, 255, 0))
    # mask out part of image
    pygame.draw.polygon(primary2,(0,255,0), ((0,cs),(cs,0),(0,0)),0)
    pygame.draw.polygon(primary2,(0,255,0), ((0,cs),(cs,cs2),(0,cs2)),0)
    # draw master segment in various positions
    screen.fill((0, 0, 0))
    screen.blit(primary2,(cs-1,0))
    primary = pygame.transform.flip(primary2,True, False)
    screen.blit(primary,(1,-1))
    primary3 = pygame.transform.rotate(primary2, 90.0)
    screen.blit(primary3,(0,1))
    primary3 = pygame.transform.rotate(primary2, -90.0)
    screen.blit(primary3,(0,cs-1))

    pygame.display.update() # display screen

def rot_center(image, angle):
    # rotate an image while keeping its center and size
    orig_rect = image.get_rect()
    rot_image = pygame.transform.rotate(image, angle)
    rot_rect = orig_rect.copy()
    rot_rect.center = rot_image.get_rect().center
    rot_image = rot_image.subsurface(rot_rect).copy()
    return rot_image

def saveScreen():
    global shotNumber,savePath
    if savePath == "" :
        savePath = asksaveasfilename()
        shotNumber = 0
        print("save path",savePath)
    rect = pygame.Rect(1,1,cs2-2,cs2-2) # remove black lines
    sub = screen.subsurface(rect)
    pygame.image.save(sub, savePath+"_"+str(shotNumber)+".jpg")
    if saveSource:
        rect = pygame.Rect(0,0,640,480)
        sub = camFrame.subsurface(rect)
        pygame.image.save(sub, savePath+"_"+str(shotNumber)+"_source.jpg")
    print("saved as", savePath+"_"+str(shotNumber)+".jpg")
    shotNumber +=1

def terminate(): # close down the program
    webcam.stop()
    pygame.quit() # close pygame
    os._exit(1)
```


Generate four copies of segment each rotated by 90 degrees

Combine four segments

Figure 2B

be flipped or rotated and then creates the symmetrical image we have described previously. The auto-rotation function might need a little explanation, however. When this is enabled, the rotation angle increments half a degree each frame. In order for the rotation to eventually produce an image of the correct dimensions as all the previous operations, we take a square image twice the size, rotate it around its centre, and then select a square portion of it at the correct size. This means the resulting image is a bit zoomed in, to avoid blank sections at various rotation angles. The effect, however, is as if you were turning the kaleidoscope tube.

Using the software

The startup condition has no preprocessing on the input image. You can simply wave the camera around and see the results in real time. The **RETURN** key will save the kaleidoscope; the first time, it will prompt you for a folder and an initial name. Subsequent images will be saved using that name with number

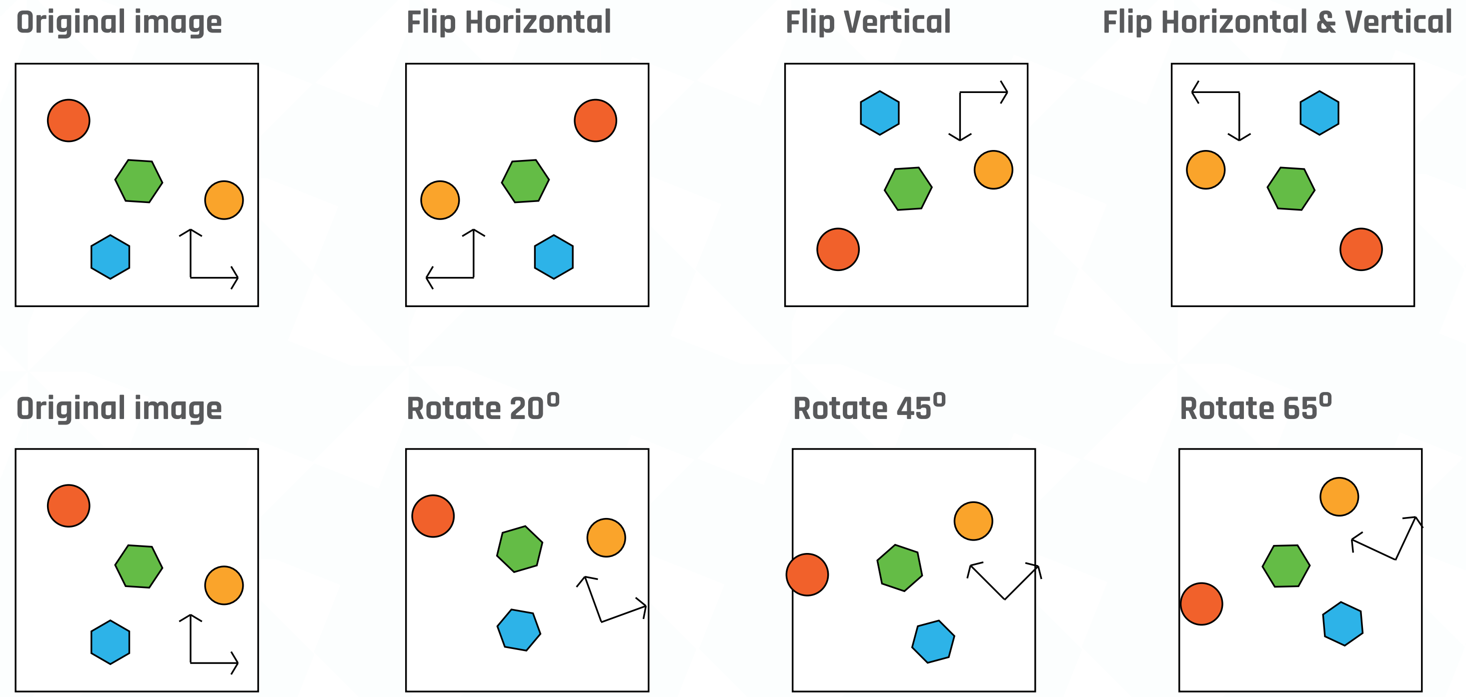


Figure 3 Input image manipulations

postfix, which increments with each frame. Pressing the **S** key will allow you to change the folder and root name, as well as resetting the postfix number. The **O** key will toggle the option of saving the original image along with the kaleidoscope image. The keys **H** and **V** will apply horizontal and vertical flipping of the input image, and the **R** key will start off the auto-rotation of the input image.

We found that the best results were obtained when you could not imagine what the input image was and it looked rather abstract. In fact, the most plain boring input images often produced better results than detail-packed ones. The quality of the auto-rotated images was slightly degraded due to the rotation process producing jagged edges on diagonal lines. One thing you will notice immediately is that if someone is in the input image, the results can look quite ghoulish.

Taking it further

We used either a simple LEGO mount for the camera, or nothing at all. However, we can't help but think a pan-and-tilt mount moving slowly might produce results that look good. There is scope for generating other degrees of symmetry using rotation instead of flipping – see what you can produce.

Unfortunately, we did not find a way for the image effects of the Raspberry Pi Camera Module to be enabled when it is running into Pygame – maybe some programming wizard could. Mind you, they are all quite simple image-processing algorithms, so you could apply them by using software in the pre-symmetry image manipulations.

```

def checkForEvent(): # see if we need to quit
    global savePath,autoRotate,saveSource,preRot,flipH,flipV
    event = pygame.event.poll()
    if event.type == pygame.QUIT :
        terminate()
    if event.type == pygame.KEYDOWN :
        if event.key == pygame.K_ESCAPE :
            terminate()
        if event.key == pygame.K_s :
            savePath = ""
            saveScreen()
        if event.key == pygame.K_RETURN :
            saveScreen()
        if event.key == pygame.K_r :
            autoRotate =  not autoRotate
            print("Auto rotate =",autoRotate)
            if autoRotate:
                preRot = 0
        if event.key == pygame.K_o :
            saveSource = not saveSource
            print("Save the source file =",saveSource)
        if event.key == pygame.K_h :
            flipH = not flipH
            print("Flip horizontal now =",flipH)
        if event.key == pygame.K_v :
            flipV = not flipV
            print("Flip vertical now =",flipV)

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

KYLE GOFF, BINSEN QIAN, & HARRY CHENG

Binsen Qian, PhD candidate in mechanical engineering; Kyle Goff, undergraduate computer engineering student; and Harry Cheng, professor and director of the UC Davis C-STEM Center.
magpi.cc/2xEGEa9

USE A GUI TO CONTROL GPIO PINS

Discover a simple way to interact with GPIO pins using C-STEM

You'll Need

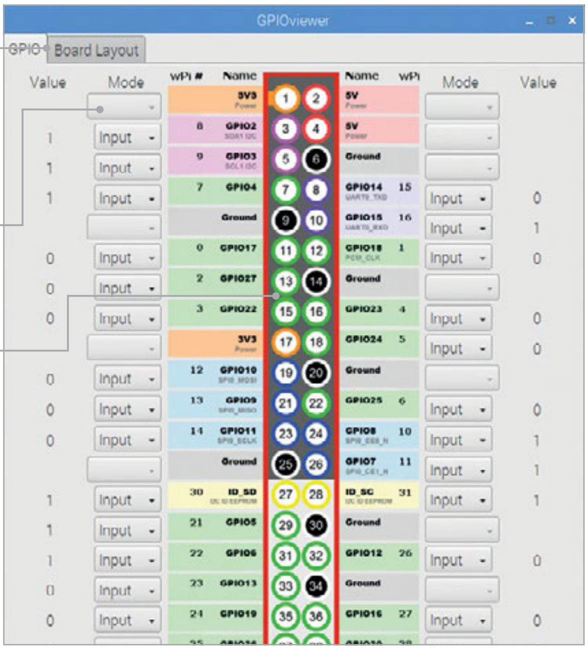
- Raspberry Pi 3
- C-STEMbian **magpi.cc/2p3JUNP**
- Breadboard
- Jumper wires
- 1 × LED
- 1 × 220 Ω resistor (Red-Red-Brown)

TROUBLE-SHOOTING HELP

Remember to use GPIOviewer for testing the pins before programming. If this fails, check the LED is plugged in properly and the wires for all components are in the correct spots.

One of the easiest ways to get started on a Raspberry Pi is with C-STEM Studio on the C-STEMbian operating system. It removes many of the hassles associated with programming in C by using C/C++ interpreter Ch. Ch is superset of C with many high-level extensions. It can run C code without compilation. This article will demonstrate how to get started controlling GPIO pins on the Pi using GPIOviewer and WiringPi with a project in Ch and highlight some key features of the system.

To make use of these programming tools, you should install the C-STEMbian operating system (**magpi.cc/2p3JUNP**) which contains C-STEM Studio. This free, open-source operating system contains all the necessary tools for robotics and physical computing. Additionally, it is a superset of Raspbian, so all the familiar features will still be there. If you already have Raspbian installed, the C-STEM modules can be installed separately on top. Step-by-step guides will assist you in setting up and accessing the Raspberry Pi if needed.



Value	Mode	wPi #	Name			Name	wPi	Mode	Value
			3V3 Power	1	2	5V Power			
1	Input	8	GPIO2 SDA1 I2C	3	4	5V Power			
1	Input	9	GPIO3 SCL1 I2C	5	6	Ground			
1	Input	7	GPIO4	7	8	GPIO14 UART0_TXD	15	Input	0
			Ground	9	10	GPIO15 UART0_RXD	16	Input	1
0	Input	0	GPIO17	11	12	GPIO18 PCM_CLK	1	Input	0
0	Input	2	GPIO27	13	14	Ground			
0	Input	3	GPIO22	15	16	GPIO23	4	Input	0
			3V3 Power	17	18	GPIO24	5	Input	0
0	Input	12	GPIO10 SPI0_MOSI	19	20	Ground			
0	Input	13	GPIO9 SPI0_MISO	21	22	GPIO25	6	Input	0
0	Input	14	GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N	10	Input	1
			Ground	25	26	GPIO7 SPI0_CE1_N	11	Input	1
1	Input	30	ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM	31	Input	1
1	Input	21	GPIO5	29	30	Ground			
1	Input	22	GPIO6	31	32	GPIO12	26	Input	0
0	Input	23	GPIO13	33	34	Ground			
0	Input	24	GPIO19	35	36	GPIO16	27	Input	0

To better understand the layout of the board, click on the Board Layout tab

Change the mode of a specific pin by selecting input, output, or PWM

The name of each pin can be found in the centre of GPIOviewer

While this article will focus on ChIDE, GPIOviewer, and the WiringPi package, there are many additional features included in C-STEMbian:

- C-STEM Studio (**magpi.cc/2xEGEa9**), a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM).
- Ch Mindstorms Controller for controlling multiple LEGO MINDSTORMS (EV3 or NXT) from the Raspberry Pi using a simple user interface
- Linkbot Labs which, like the MINDSTORMS equivalent, gives a user interface for total control of one or more Linkbots.
- Ch Arduino for controlling and programming in Ch an Arduino Uno's pins from the Raspberry Pi.

Make a circuit

The program in this tutorial would be useless without a circuit to test it on. If you have one, use a breakout board to make the wiring process clearer. Otherwise, wire the pins directly from the Pi. Take a wire from the GPIO 4 pin and connect it to an empty row of the breadboard. Then, attach the positive lead of an LED to this row. From the negative lead of the LED, attach a 220 Ω (Red-Red-Brown) resistor to ground.

Before programming, we can use GPIOviewer, a helpful feature of the C-STEMbian operating system. To use it, navigate to the big 'C' on the top of the desktop window.

Once open, navigate to 'Ch Raspberry Pi' and click Launch in the bottom right-hand corner. This will open up GPIOviewer, which allows total control of all GPIO pins of the Raspberry Pi. In this view, you can change pin modes between input, output, and PWM (with a slider). For this circuit, find GPIO pin 4 on GPIOviewer and set it to output.

Ensure the circuit is set up and working properly by switching between high and low outputs. If the light turns on, you are ready to program.

Example blink code in C

The classic way of programming this circuit on the Raspberry Pi would resemble the C code in **blink.c**, which includes the wiringPi library. This code also works in Ch interpretively without compilation.

```
/* File: blink.c */
#include <wiringPi.h>

int main() {
   wiringPiSetupGpio();
   pinMode(4, OUTPUT);
   while(1) {
      digitalWrite(4, HIGH);
      delay(500);
      digitalWrite(4, LOW);
      delay(500);
   }
   return 0 ;
}
```

Language
>C
NAME:
blink.c
DOWNLOAD:
magpi.cc/BlinkPi

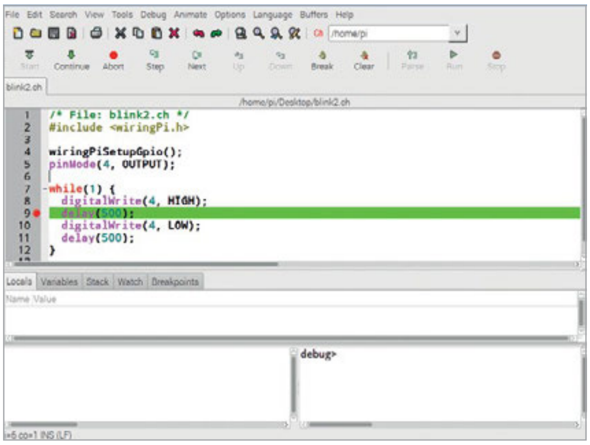
The code uses GPIO pin numbering and sets pin 4 to an output. Then, it enters a while loop that cycles the LED every half second between on and off. Notice the standard **`int main`** function that requires a '**`return 0`** statement.

Equivalent Code in Ch

However, to run the code just in Ch, the program can be simplified to the code in **blink2.ch**.

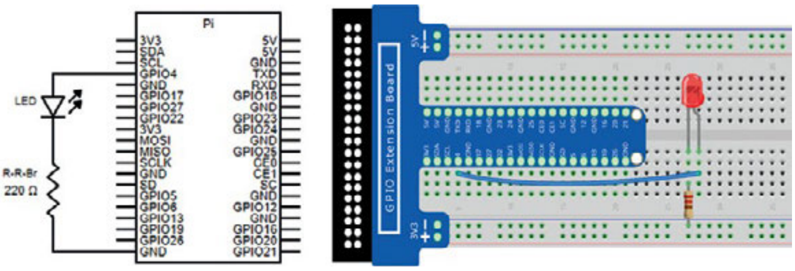
```
/* File: blink2.ch */
#include <wiringPi.h>

wiringPiSetupGpio();
pinMode(4, OUTPUT);
while(1) {
     digitalWrite(4, HIGH);
     delay(500);
     digitalWrite(4, LOW);
     delay(500);
}
```

Language
>Ch
NAME:
blink2.ch
DOWNLOAD:
magpi.cc/BlinkPi


File Edit Search View Tools Debug Animate Options Language Buffers Help
/home/pi
Start Continue Abort Step Next Up Down Break Clear Parse Run Stop
blink2.ch
/home/pi/Desktop/blink2.ch
Locals Variables Stack Watch Breakpoints
Name Value
debug>
l=6 co=1 INS (LF)

```
 1  /* File: blink2.ch */
 2  #include <wiringPi.h>
 3
 4  wiringPiSetupGpio();
 5  pinMode(4, OUTPUT);
 6
 7 -while(1) {
 8    digitalWrite(4, HIGH);
 9●   delay(500);
10    digitalWrite(4, LOW);
11    delay(500);
12  }
```

Above ChIDE provides easy-to-use debugging tools to step through Ch code



Above Set the circuit up as shown in the diagram, with a breakout board or with wiring directly from the Pi

Notice that the **`int main`** function is gone. Just like Python and other scripting languages, there is no need to create any functions for a simple program. This is just one of the many benefits of using Ch. While this method of programming is most common for users coming from C programming, there is another way.

Scripting in Ch

Besides writing code resembling C, it is possible to write scripting code in Ch. The same program as before is rewritten in **blink3.ch**.

```
#!/bin/ch
/* File: blink3.ch*/

gpio -g mode 4 out
while(1) {
  gpio -g write 4 1
  delay(500);
  gpio -g write 4 0
  delay(500);
}
```

Language
>Ch
NAME:
blink3.ch
DOWNLOAD:
magpi.cc/BlinkPi

Ch provides both methods as an option so that you can choose whichever is more comfortable for your style. To run the Ch code, it is easiest to use ChIDE. To launch it, click the magnifying glass icon next to the big 'C' you clicked to open C-STEM Studio. Alternatively, navigate to 'Programming with Ch' within C-STEM Studio's menu. Type the code into the code editing pane. To run it, click the Run button at the top of the interface above the code. If all goes well, the LED should blink on and off indefinitely.

Debugging

Another useful feature of Ch is the ChIDE debugging feature. Like any popular IDE, the ChIDE provides easy to use debugging tools to step through Ch code. To step through the code, use the debug control panel at the top of the window. You can step into functions or through code with the Step and Next buttons, and you are able to set break points by clicking in the grey line number column between the numbers and text. For example, set a break point by clicking in the grey area to the right of line 9. Then, click the Continue button. Every time you click this button, it will run the program until it hits another breakpoint. Since this is a while loop, it will hit this break point every time.

FOR HELP AND NEW IDEAS

Open the 'Learn Physical Computing with Raspberry Pi' textbook in the Ch Raspberry Pi section of C-STEM Studio for more circuits and programs.

ROB ZWETSLOOT

Tinkerer, sometime maker, other-times cosplayer, and all-the-time Features Editor of *The MagPi*.
magpi.cc

MAKE A MAGIC CHRISTMAS STAR

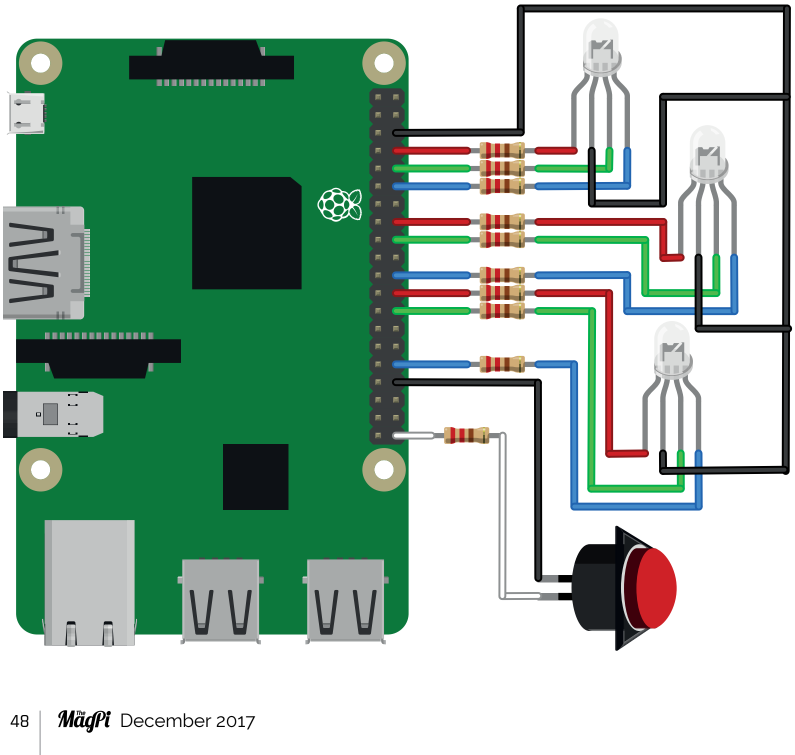
Create the perfect tree topper with this 3D-printed, Pi-powered star, over which you have full control

You'll Need

- 3D-printed star **magpi.cc/2i2tbkU**
- 3 × RGB LEDs
- 3 × 150Ω resistors
- 6 × 100Ω resistors
- 220Ω resistor
- Push button
- Wire
- Breadboard (optional)

Every year here at *The MagPi*, our Features Editor Rob Zwetsloot says he's going to make a new star for his Christmas tree that's powered by a Raspberry Pi. This year he's finally decided to put his money (or text) where his mouth is and make one.

As Christmas is all about sharing, we thought he should also show you how to make one. After all, there's nothing merrier than a Raspberry Pi Christmas. So get out your cinnamon-spiced soldering iron and put a Santa hat on your 3D printer and let's create some festive joy.



>STEP-01

Select a star

The most important part of this project is the star. We've used one from Thingiverse (**magpi.cc/2i2tbkU**), which you can see in the pictures for this tutorial. However, you can use whatever star shape you want. All you need to do is make sure it's hollow and you can easily access the inside of the star, but the electronics for the star will work the same in any case.

You can look for different star shapes on Thingiverse (**thingiverse.com**), or even draft up your own in 3D modelling software like Blender.

>STEP-02

Get printing

Not everyone has a 3D printer at home, however much we would all love Santa to bring us one. If you're not lucky enough to have access to one at your workplace, your local makerspace might have one.

If all these options are unavailable to you, then we highly recommend checking out **3dhubs.com**. It's a fantastic service that aggregates thousands of 3D printing services – all you need to do is set your location and upload your files and it will find a 3D printer. These might be a local business or a bigger company – either way, you can find out price estimates and what type of printer services each company provides.

You don't need to get very expensive material for the star, although we do recommend translucent plastic. Ours is orange, but you can likely find yellow if you prefer.

Figure 1 Use this Fritzing diagram as a guide for wiring up the LEDs and button

>STEP-03

Solder it up

While waiting for your star parts to be delivered, now is a good time to build your circuit. It's quite simple: each RGB LED colour pin is hooked up to its own GPIO pin, the ground pins go to ground (GND), and then the button gets its own pin. Check out **Figure 1** for a wiring guide.

The button and each colour pin of the RGB LEDs require a resistor for them to work, so don't ignore them when testing or soldering up your circuit!

Figure out where you want to put the Pi relative to your tree and cut the wires for the distance accordingly. Alternatively, try hiding a Pi Zero in the star!

>STEP-04

Add the code

We have some example code you can try out, called **star_lights.py**. Make sure Raspbian is installed on the Pi you plan to use and write it up/download it to the home folder. Run the code to test out the circuit and make sure everything works – it should start with all the LEDs lighting up white, then each button press will change it: blinking, then pulsing, then all on with different colours, then blinking or pulsing with different colours.

In the Terminal, use **`sudo nano /etc/profile`** and add the following line to the bottom so that the code runs at startup:

```
sudo python star_lights.py
```

>STEP-05

Construct the star

Once the star has arrived, use some sticky tape to carefully stick the LEDs to the star. We stuck ours to the rear half of our two-part star and then glued the bits together. Give it another test to see if you're happy with it. Modify the code as you see fit (see the GPIO Zero **`RGBLED`** doc for reference: **magpi.cc/2i3MTgq**) and turn off 'boot to desktop' in the Pi configuration to save some power.

>STEP-06

The tree-topping ceremony

Now is the big moment. Decorate the rest of your tree, get your favourite Christmas jumper out, and put the kettle on. If you have a tall tree, get a ladder out – safety first, after all – and place your star over the top part of the tree.

There. Christmas has begun. We recommend pairing your ceremony with a nice cup of tea (which is why we told you to put the kettle on) and a mince pie. From all of us at *The MagPi*, we wish you a Merry Christmas, Happy Holidays, and a Happy New Year!

Language

>PYTHON

DOWNLOAD:
magpi.cc/
magpistar

star_lights.py

```
from gpiozero import RGBLED, Button
from time import sleep

led1 = RGBLED(14,15,18)
led2 = RGBLED(23,24,25)
led3 = RGBLED(8,7,12)

button = Button(21)

mode1 = (1, 0, 0) # Red
mode2 = (0, 1, 0) # Green

while True:
        led1.on()
        led2.on()
        led3.on()

        button.wait_for_press()
        button.wait_for_release()

        led1.blink()
        sleep(0.4)
        led2.blink()
        sleep(0.4)
        led3.blink()

        button.wait_for_press()
        button.wait_for_release()

        led1.pulse()
        sleep(0.4)
        led2.pulse()
        sleep(0.4)
        led3.pulse()

        button.wait_for_press()
        button.wait_for_release()

        led1.on()
        led2.color(mode1)
        led3.color(mode2)

        button.wait_for_press()
        button.wait_for_release()

        led1.blink(on_color=(mode1), off_color=(mode2))
        sleep(0.4)
        led2.blink(on_color=(mode2), off_color=(mode1))
        sleep(0.4)
        led3.blink(on_color=(mode1), off_color=(mode2))

        button.wait_for_press()
        button.wait_for_release()

        led1.pulse(on_color=(mode1), off_color=(mode2))
        sleep(0.4)
        led2.pulse(on_color=(mode2), off_color=(mode1))
        sleep(0.4)
        led3.pulse(on_color=(mode1), off_color=(mode2))
```

SAI YAMANOOR

Sai Yamanoor is a Mechatronics Engineer and a DIY enthusiast. In his free time, he likes to build gadgets using the Raspberry Pi.
saiyamanoor.com

SRIHARI YAMANOOR

Srihari Yamanoor is a Mechanical Engineer and a Medical Device Designer. Srihari's interests include building low-cost devices that improve quality of life.
yamanoor.com

DIY INDOOR AIR-QUALITY MONITOR

Build your own indoor air-quality monitor using the Raspberry Pi Zero

You'll Need

- Raspberry Pi Zero
- Pimoroni Blinkt! LED strip
- Adafruit Perma Proto Bonnet Mini Kit
- Sparkfun CCS811 sensor
- Stackable 2×20 female headers
- 4 × Male-to-male jumper cables
- 3D-printed enclosure (source files available with this tutorial)

In this tutorial, you will learn to build your own indoor air-quality monitor/data logger. We will build it using the Raspberry Pi Zero and the CCS811 volatile organic compound sensor.

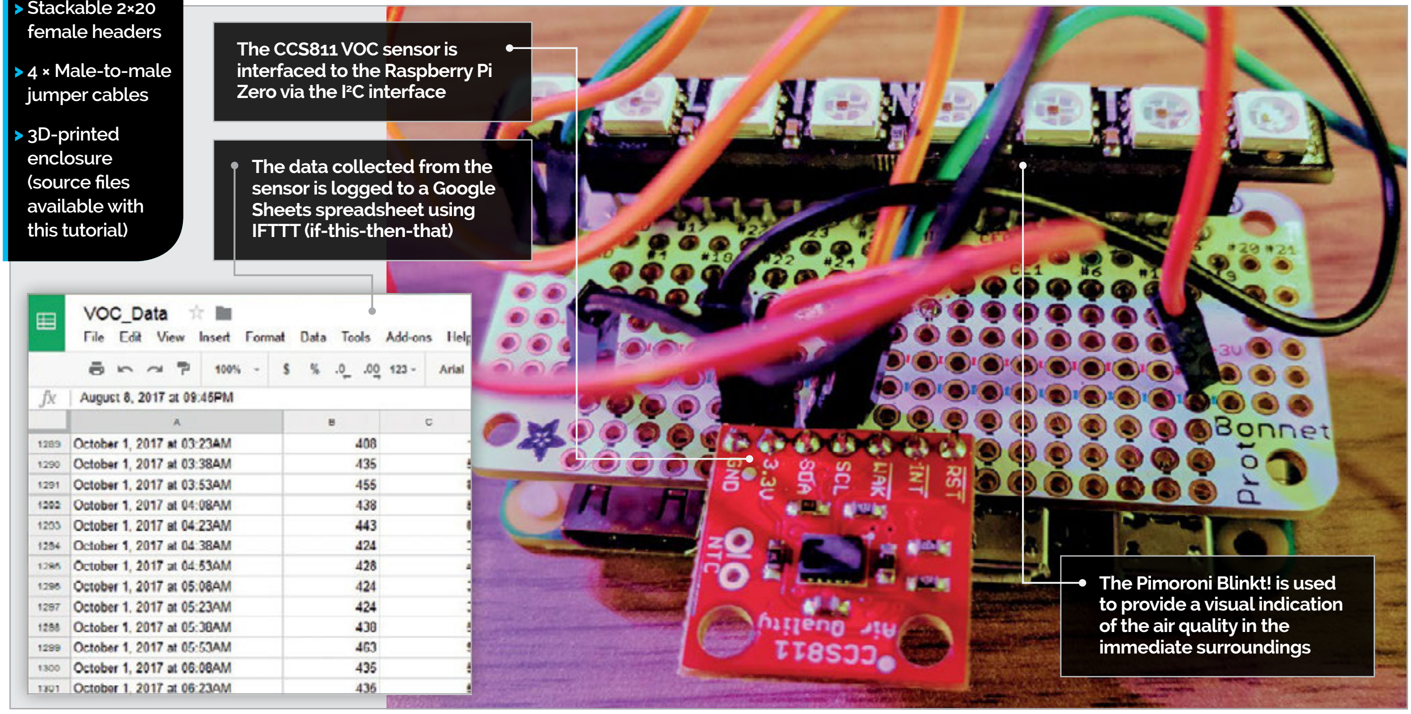
The collected data would be used to provide a visual indication of the air quality (using an RGB LED strip) and the information collected would be saved to a Google Sheets spreadsheet. The visual aid serves to educate people about their immediate surroundings while the data saved to a spreadsheet helps understand the air-quality trends in a room.

Connect the sensor

The CCS811 sensor comes with an I²C interface and requires a 3.3V power source. The clock and data pins of the sensor needs to be connected to the pins GPIO 3 and GPIO 2 respectively (shown in **Figure 1**).

Test the sensor

The first step is to enable the I²C interface and verify that the sensor is interfaced correctly and check whether the Raspberry Pi is able to detect the sensor on the I²C interface.



The CCS811 VOC sensor is interfaced to the Raspberry Pi Zero via the I²C interface

The data collected from the sensor is logged to a Google Sheets spreadsheet using IFTTT (if-this-then-that)

The Pimoroni Blinkt! is used to provide a visual indication of the air quality in the immediate surroundings

VOC_Data

File Edit View Insert Format Data Tools Add-ons Help

100% $ % .0 .00 123 Arial

fx August 8, 2017 at 09:45PM

	A	B	C
1289	October 1, 2017 at 03:23AM	408	
1290	October 1, 2017 at 03:38AM	435	
1291	October 1, 2017 at 03:53AM	455	
1292	October 1, 2017 at 04:08AM	438	
1293	October 1, 2017 at 04:23AM	443	
1294	October 1, 2017 at 04:38AM	424	
1295	October 1, 2017 at 04:53AM	428	
1296	October 1, 2017 at 05:08AM	424	
1297	October 1, 2017 at 05:23AM	424	
1298	October 1, 2017 at 05:38AM	430	
1299	October 1, 2017 at 05:53AM	463	
1300	October 1, 2017 at 06:08AM	435	
1301	October 1, 2017 at 06:23AM	436	

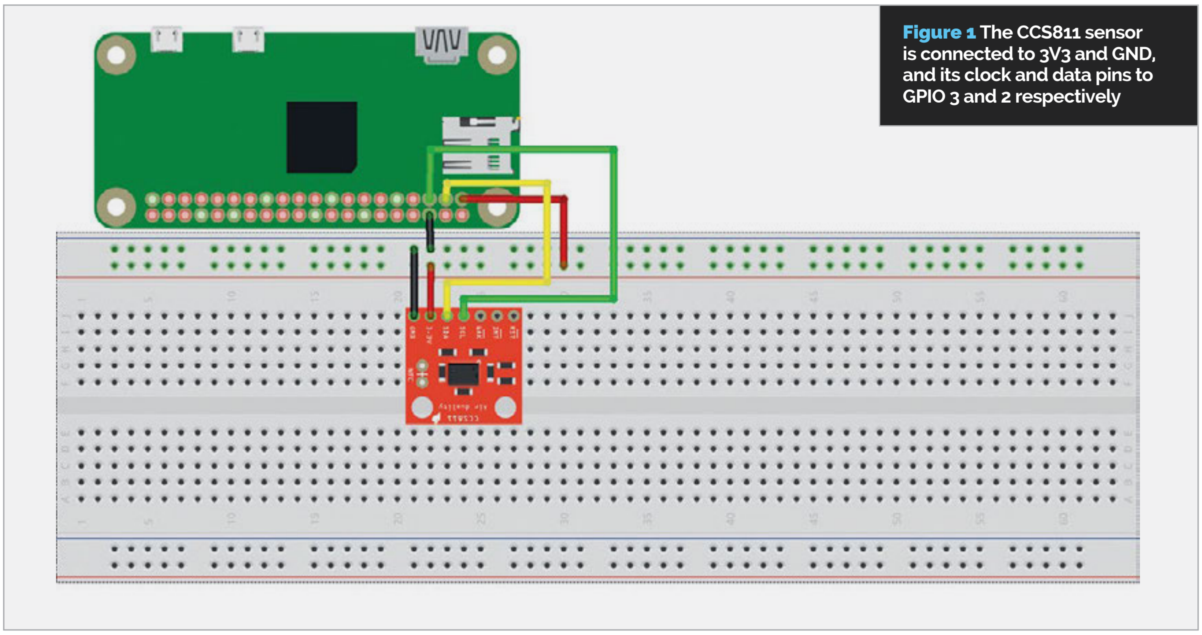


Figure 1 The CCS811 sensor is connected to 3V3 and GND, and its clock and data pins to GPIO 3 and 2 respectively

The I²C interface can be enabled from the Raspberry Pi's desktop. Go to Menu> Preferences > Raspberry Pi Configuration. Under the Interfaces tab, enable the 'I2C' interface.

Now, open the Terminal and enter:

```
sudo i2cdetect -y 1
```

The above command should list all the devices connected to the Raspberry Pi's I²C interface. As shown in **Figure 2**, the CCS811 sensor's address is 0x5A. According to the sensor's data sheet, the sensor could be configured to the address 0x5A or 0x5B.

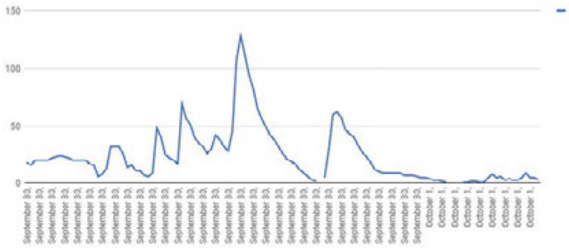
The next step is setting the clock rate of the I²C interface. We need to adjust the I²C interface clock speed to meet the sensor's clock rate specification. Edit **/boot/config.txt** from the Terminal:

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

Add the following line to the end of the file:

```
dtparam=i2c_baudrate=8000
```

Save the changes to the file (**CTRL+X**) and reboot your Raspberry Pi.



Above The observed volatile organic compound trend over a 24-hour period. This information can be useful in identifying ventilation needs for a room

Install the prerequisite libraries

In order to read data from the sensor, we need to install some dependent libraries:

```
 pip install python-periphery requests
configparser
 sudo apt-get install python-blinkt
```

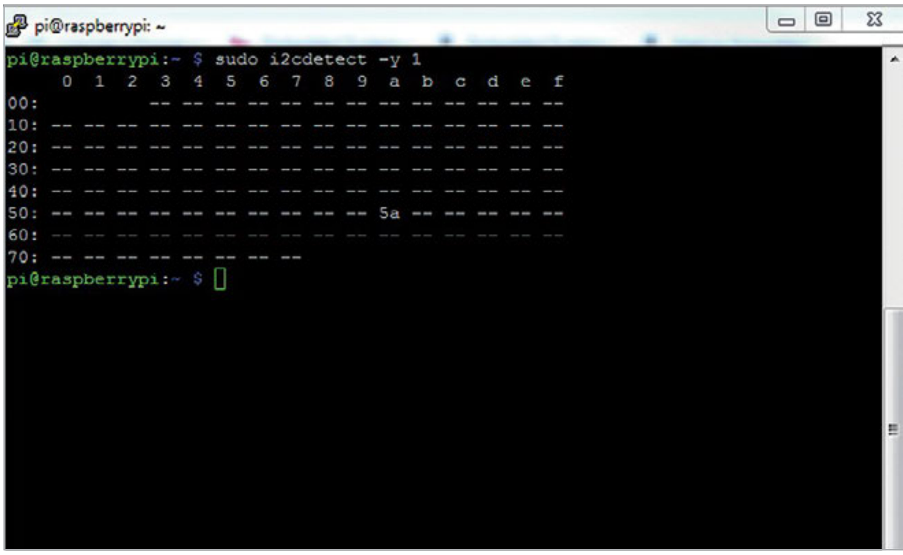
We installed the python-periphery library to access the I²C interface and the Python requests library to upload the sensor data to a Google Sheets spreadsheet. We installed the blinkt library by Pimoroni to control the RGB LED strip. If you are using a different RGB LED, install the prerequisite library for the same (if any).

Note: At the time of building this gadget, there were no libraries readily available to read data from the CCS811 sensor; therefore, we wrote our own library. Adafruit Industries has since published its own library (**pypkg.com/pypi/adafruit-ccs811**). Choose a library that meets your needs.

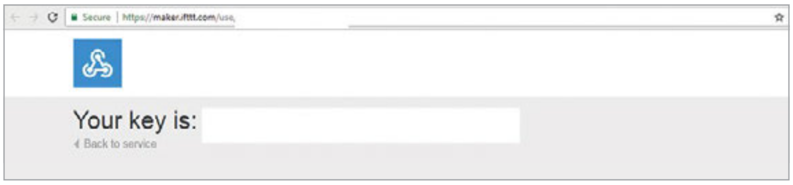
INSTALLATION

Install the gadget in your living room to understand whether the air quality in your home needs improvement.

Figure 2 Listing all devices connected to the Pi's I²C interface



```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- 5a -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $ 
```



Your key is:

Back to service

Figure 3 Take a note of the key and store it in the **key.ini** file

VISUAL AID

Use a Sharpie or permanent marker to demonstrate the effect of volatile organic compounds. Do not use the VOC sensor to measure carbon dioxide levels.

Read data from the sensor

Before we put everything together, we need to test every component of our gadget. Let's get started with testing the sensor. In order to verify that the sensor operates as expected, we need to perform a simple transaction using the I^{2}C interface. According to the data sheet, it is possible to read the value at the register, 0x20, and retrieve the hardware ID of the sensor. In this case, it is 129 (hex: 0x81).

The code for this gadget can be downloaded from the GitHub repository (**magpi.cc/2zrzuXP**). From the Terminal, switch to the directory containing the code. Launch the Python interpreter and execute the following:

```
  >>> import ccs811
  >>> my_ccs811 = ccs811.CCS811(device_
address=0x5A)
  >>> print(my_ccs811.read_byte(0x20))
  129
```

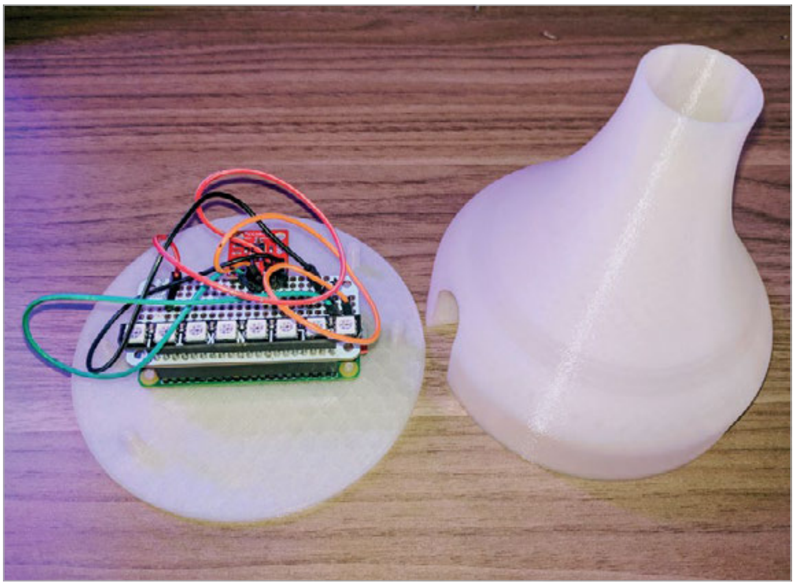
Now that we have the verified the sensor function, let's test the RGB LED strip.

Testing the RGB LED strip

We used the Pimoroni Blinkt! LED strip (**magpi.cc/2yg5Mng**) for this tutorial. Pimoroni provides the library required to control the RGB LED strip and hence it is very easy to integrate into the project.

The library installation is a one-stop process (shown in the prerequisite installation step). Let's write a quick Python script to test the library:

Figure 4 The enclosure can be 3D-printed using the STL files in the project repository



```
import blinkt
from time import sleep

while True:
    blinkt.set_all(255, 0 , 0)
    blinkt.show()
    sleep(1)
    blinkt.set_all(0, 255, 0)
    blinkt.show()
    sleep(1)
```

The LED strip should alternate between red and green colours with a one-second interval.

Setting up your IFTTT account

We will be using IFTTT (if this, then that) to save the sensor data to a spreadsheet. If you are not familiar with IFTTT, we recommend reading this tutorial from *The MagPi* (**magpi.cc/2yicr08**) – note that the Maker Channel is now called the Webhooks service.

To save the data to a spreadsheet, we create an IFTTT applet where a webhook containing the sensor data is used as a trigger to add a row to the spreadsheet.

Record your webhook key from the webhook documentation page: Your Profile > Services > Webhooks > Documentation (**Figure 3**).

Save your key in a file called **key.ini** (or any file name of your choice) in the following format:

```
[CREDENTIALS]
key = YOUR_KEY
```

Replace the text `YOUR_KEY` with your IFTTT key. Now, let's test if we are able to add a row to a spreadsheet.

```
  config = configparser.ConfigParser()
  config.read('/home/pi/key.ini')
  key = config.get('CREDENTIALS', 'key')
  payload = {'value1': 1, 'value2': 2}
  print(URL.format(key=key))
  try:
      response = requests.post(URL.
format(key=key), json=payload)
  except requests.exceptions.
ConnectionError as error:
      print(str(error))
  if response.status_code == 200:
      print("Success")
```

If the sensor data was successfully posted, you should be able to verify that a row has been added to the spreadsheet.

Soldering/assembly

Now that we have verified the components, let's solder the sensor onto a prototyping board. We recommend using the Adafruit Perma Proto Bonnet

Mini Kit (**magpi.cc/2yg7oxk**), since it comes with the same profile as the Raspberry Pi Zero. Instead of using the headers that come with the kit, we recommend using the stackable headers (**magpi.cc/2yidoag**). This would enable stacking the Pimoroni Blinkt! LED strip on top of the Proto Bonnet Mini Kit.

Enclosure

In order to produce an orb-like effect, we used an enclosure that diffuses light. We designed an enclosure that could be built using a 3D printer (**Figure 4**). We recommend using a white filament, as it best diffuses the light. The enclosure consists of a top portion that diffuses the light and a bottom portion that holds the Raspberry Pi Zero. The enclosure design files are available in the project repository (**magpi.cc/2zrzuXP**).

Putting everything together

Once you have assembled the Raspberry Pi Zero, CCS811 sensor, and LED strip into the enclosure, it is time to test the gadget. The code consists of a CCS811 class that handles all interactions with the sensor, i.e. reads/writes data to the sensor.

When the script is run, the sensor is reset and initialised. Upon initialisation, we poll the status register to determine if new data samples are available. According to the data sheet, the `DATA_READY` flag indicates whether a new sample is available.

When a new sample from the sensor is available, the program fetches the equivalent carbon dioxide levels and the volatile organic compound levels (VOC). We use the VOC levels to set the colour of the RGB strip. For example, if the VOC levels are below 16 ppb, we set the colour to green; if they are greater than 60 ppb, the RGB strip is set to red. The `FadeInOut` method is used to produce a light effect by adjusting the brightness levels of the LED strip. Every 15 minutes, a data point is uploaded to the Google spreadsheet using the `post_data` method. If the `ERROR` flag of the `STATUS` register is set, the program automatically resets the sensor and re-initialises the communications.

Run the Python script and the gadget should start glowing green. Open a permanent marker or any pen that has an odour and introduce it into the enclosure. You will notice that the enclosure changes colour from green to red.

NOTE: The CCS811 sensor requires a 48-hour burn-in and at least 20 minutes after power-up to provide stable readings.

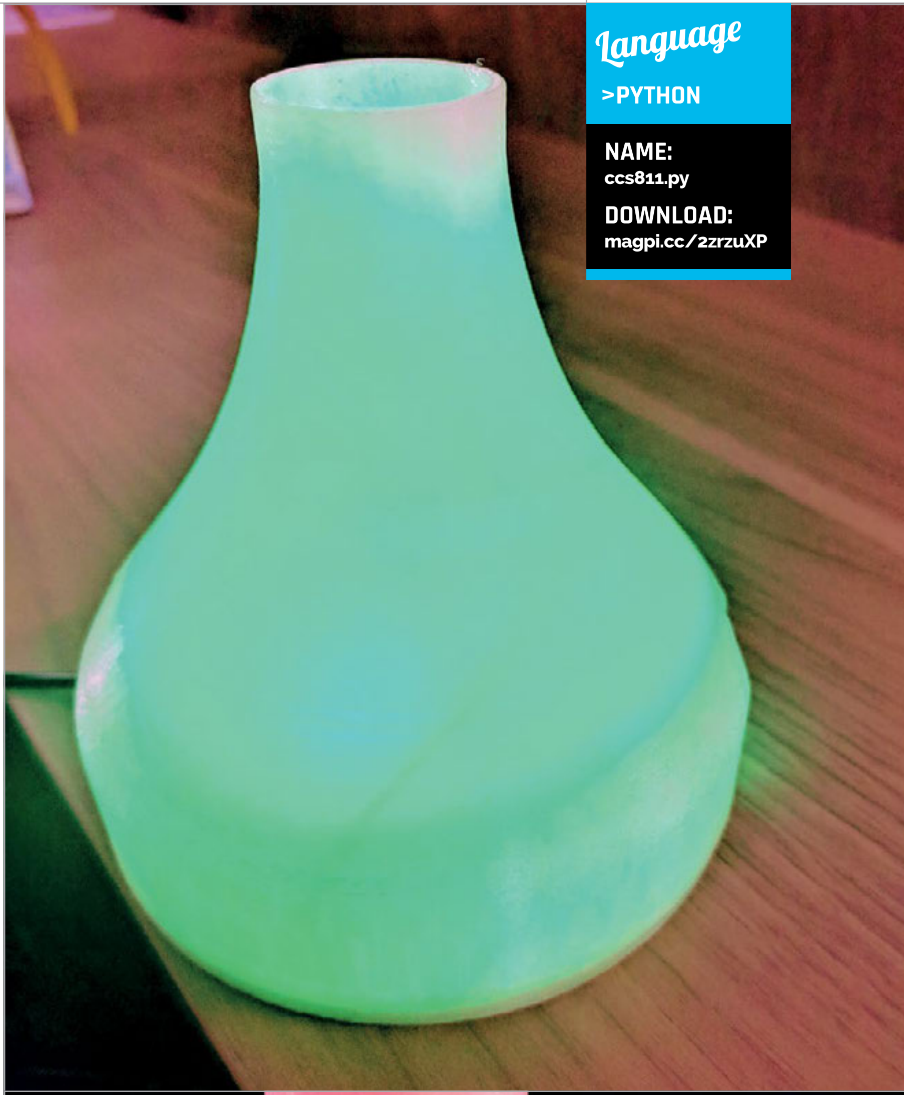
VOC trends from spreadsheet

Let the gadget run for at least 24 hours and collect data in 15-minute intervals. It would be interesting to look at the trends and find out if your room needs modifications to its ventilation.

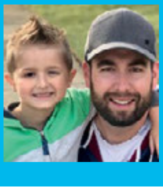
Language
>PYTHON

NAME:
ccs811.py

DOWNLOAD:
magpi.cc/2zrzuXP



The sensor started glowing red when a Sharpie was brought close to the enclosure



BEN (BENSTR) STRAHAN

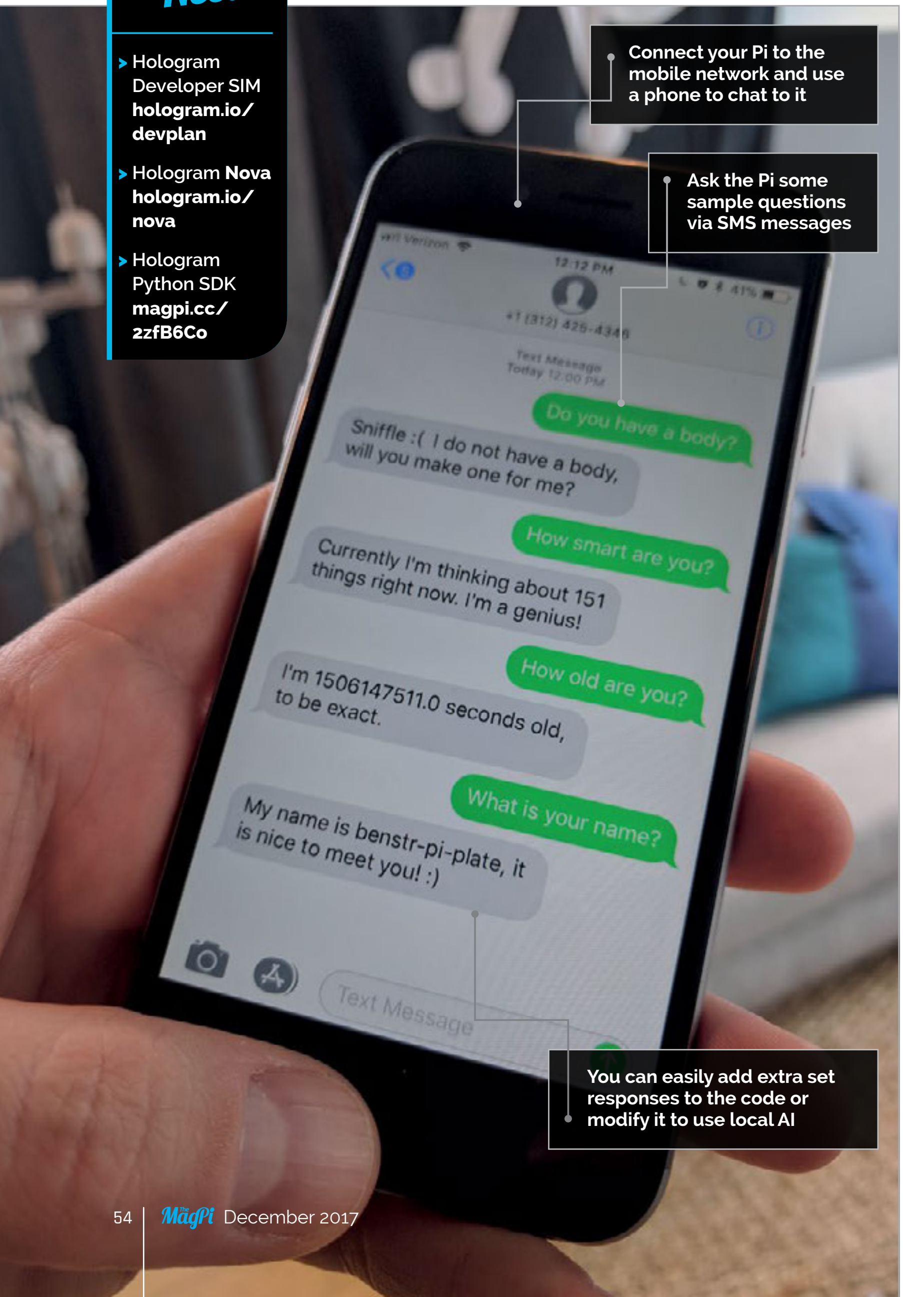
Benstr leads Developer Relations at hologram.io. His job is to make cool projects, connecting 'things' to the world with cellular.
hackster.io/benstr / @_benstr

CONNECT YOUR RASPBERRY PI WITH CELLULAR

Going places without WiFi? **Ben Strahan** shows how to use mobile networks and send SMS messages to your Pi

You'll Need

- Hologram Developer SIM **hologram.io/devplan**
- Hologram **Nova** **hologram.io/nova**
- Hologram Python SDK **magpi.cc/2zfB6Co**



Do you have a body?

Sniffle :(I do not have a body, will you make one for me?

How smart are you?

Currently I'm thinking about 151 things right now. I'm a genius!

How old are you?

I'm 1506147511.0 seconds old, to be exact.

What is your name?

My name is benstr-pi-plate, it is nice to meet you! :)

Connect your Pi to the mobile network and use a phone to chat to it

Ask the Pi some sample questions via SMS messages

You can easily add extra set responses to the code or modify it to use local AI

Ever wonder what your Raspberry Pi was thinking? This tutorial shows how you can ask your Pi questions no matter where it is in the world. Learn how to send questions through SMS to your Raspberry Pi and get witty answers back.

You've probably connected your Pi to the internet through Ethernet or WiFi, but have you ever considered using cellular? After all, your WiFi only covers a small portion of the planet.

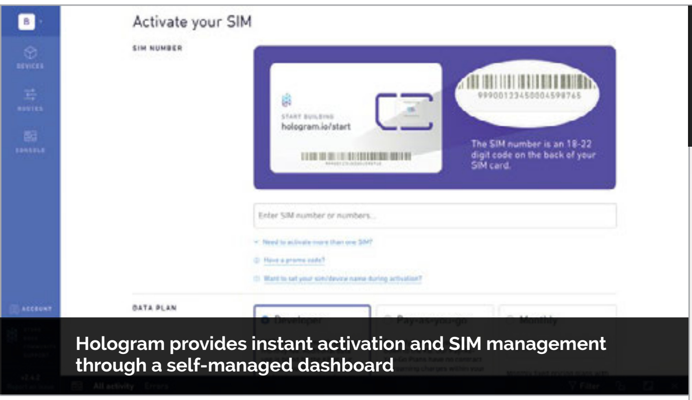
This tutorial explores using cellular with the Raspberry Pi family. Hologram.io recently released the Nova, a USB modem built for single-board computers like the Pi. The Nova, Hologram's Python SDK, and Hologram's global cellular network make connecting your Pi to the mobile phone network simple.

This tutorial requires the Hologram Nova which includes a free SIM. Learn more at **hologram.io/nova.**

Hologram SIM configuration

Upon receiving your Nova and SIM, you need to activate the SIM to make the worldwide network available. From the Hologram Dashboard (**magpi.cc/2ijHyFn**), click the Activate SIM button in the top right-hand corner. After activation you'll be directed to the device list page. You should see your new device in the list; it may not be clickable while it provisions onto the network.

Once the device is available, click it to be taken to the device details page. On the left you'll notice the device sub-navigation. For this tutorial we'll need to configure a device phone number and Cloud Services credentials. Select Configuration in the side-nav and walk through Configure Phone Number and Cloud Services Router sections. Make sure to save the phone number and device key for later use.



Activate your SIM

SIM NUMBER

The SIM number is an 18-22 digit code on the back of your SIM card.

Enter SIM number or numbers...

Need to activate more than one SIM?

Have a promo code?

Want to set your sim/device name during activation?

DATA PLAN

Hologram provides instant activation and SIM management through a self-managed dashboard

Local Pi configuration

Next, we're going to install all the dependencies this tutorial requires. Again, Hologram makes this very simple with a single-line command. It's recommended you be connected to the internet through WiFi or Ethernet since these dependencies will use a lot of data.

On the Pi, open a Terminal window and run the following script:

```
curl -L hologram.io/python-install | bash
```

For this tutorial we'll need one more dependency:

```
sudo apt-get install python-psutil
```

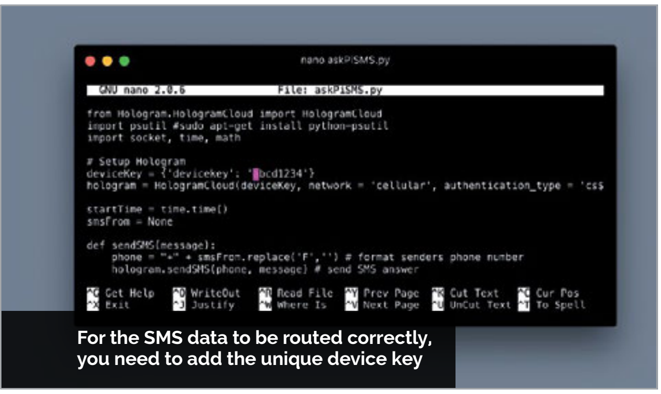
You can now connect the Nova to your Pi. We'll run the following code to verify everything is installed correctly.

```
sudo hologram send –cloud "Hello World!"
```

Along with the Python SDK, Hologram's script installed a neat little command-line interface (CLI). Learn more about what the CLI can do by executing the following commands:

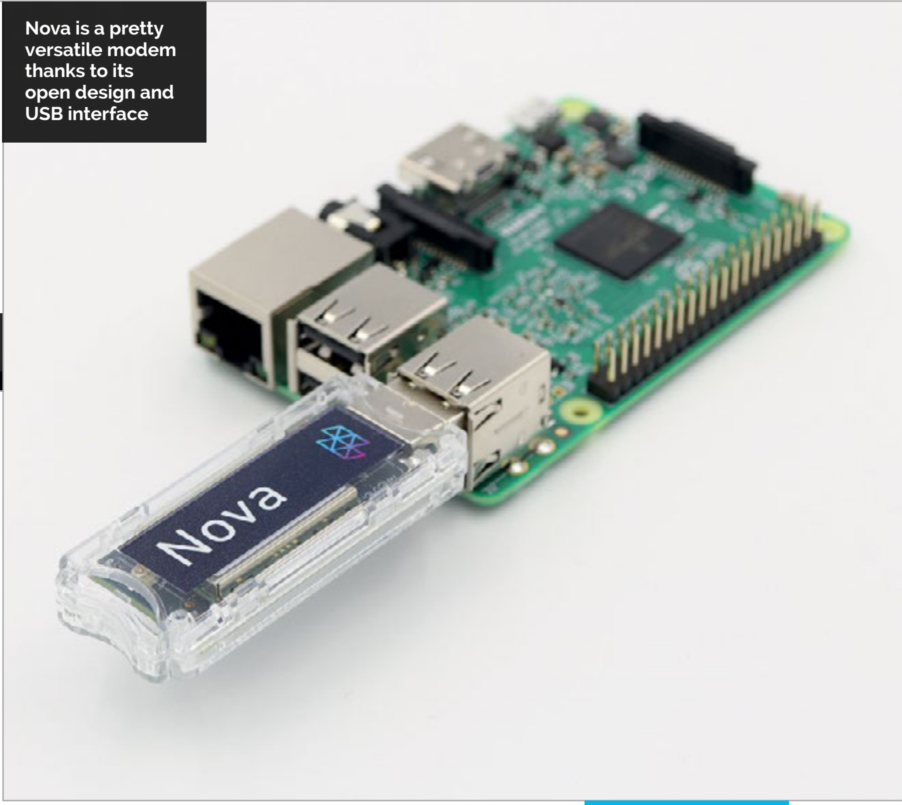
```
hologram –help
hologram modem –help
```

Head over to the Hologram Dashboard (**magpi.cc/2ijHyFn**) to see your 'Hello Nova' command above.



```
  GNU nano 2.0.6               File: askPiSMS.py

from Hologram.HologramCloud import HologramCloud
import psutil #sudo apt-get install python-psutil
import socket, time, math

# Setup Hologram
deviceKey = {'devicekey': 'abcd1234'}
hologram = HologramCloud(deviceKey, network = 'cellular', authentication_type = 'cs$

startTime = time.time()
smsFrom = None

def sendSMS(message):
    phone = "+" + smsFrom.replace('F','') # format senders phone number
    hologram.sendSMS(phone, message) # send SMS answer

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

For the SMS data to be routed correctly, you need to add the unique device key



Nova is a pretty versatile modem thanks to its open design and USB interface

Run SMS conversation script

Let's get the code so we can start talking with our Pi over SMS. From the Pi Terminal, clone the following repository.

```
git clone https://github.com/benstr/TUT-
ask-pi-sms.git
```

Remember the device key you generated in the first step? Paste the device key on line 6 of the script.

```
cd TUT-ask-pi-sms
sudo nano askPiSMS.py
```

Save the file. You're now ready to run the script and text your new robot friend, fingers crossed!

```
sudo python askPiSMS.py
```

From your phone, send the following questions to the phone number you received earlier.

- What is your name?
- How old are you?
- Do you have a body?
- How smart are you?

By default, SMS is slow on all networks. Also, for simplicity, the code is not very fast. For both reasons, it might take 30-60 seconds to receive a response.

Congrats, you have a new robot friend you can chat with! Modify the code to support more questions or add local AI to have a truly intelligent friend.

Language

>PYTHON

NAME:
askPiSMS.py

DOWNLOAD:
magpi.cc/2ztwE3X

ENJOY EASY RETRO GAMING

Lakka lets you relive the games of the past by enabling your Raspberry Pi to emulate a host of retro computers and consoles

You'll Need

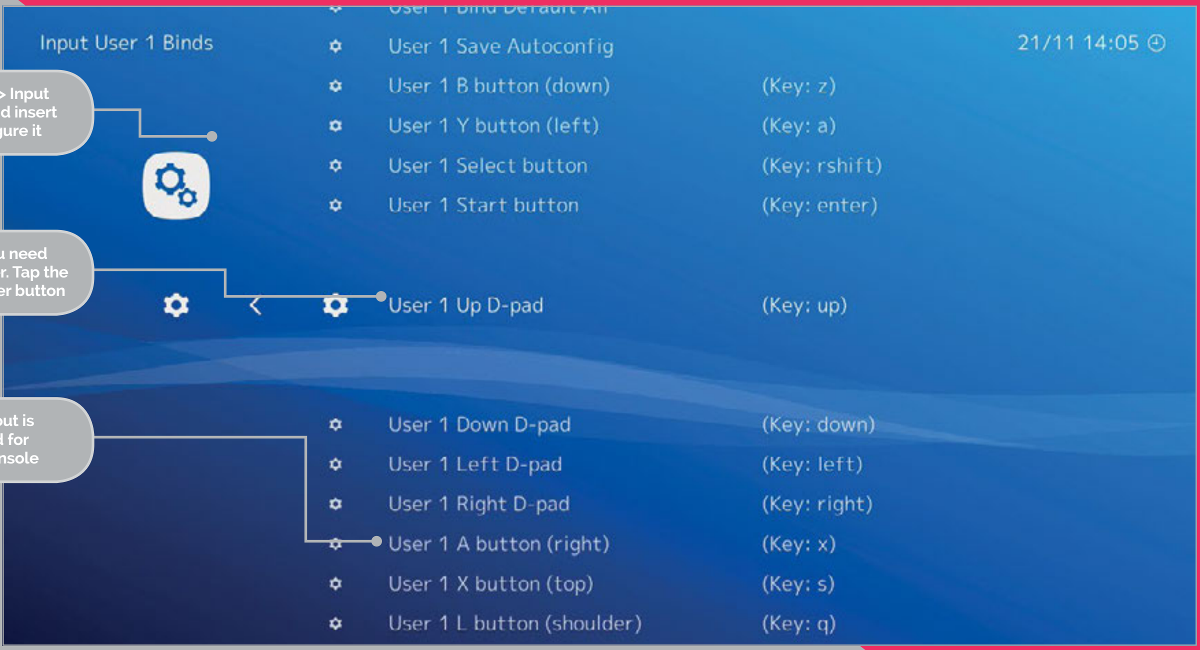
- PC or Mac
- Blank microSD card
- SD Formatter
- NOOBS image file
- A game ROM

Whether you are nostalgic for the games of yesteryear or you're simply dying to discover gaming's rich history, all you ultimately need to get stuck in is a bunch of emulators and a stack of gaming ROMs. In the past, however, this has also entailed finding and downloading the BIOSes of various machines and a fair bit of configuration. Fortunately, with the software platform Lakka installed on your Raspberry Pi, the path to gaming glory is much smoother these days.

Lakka allows you to emulate arcade games as well as titles originally released on a host of 8-bit, 16-bit, and even 32- and 64-bit systems. You can hook up a joypad and even make use of the wireless controllers made for the PlayStation and Xbox (there's more about those at **magpi.cc/2A1wSQF**). It has an interface that will be very familiar to anyone who has used Sony's later PlayStations and, because it is open source, it is constantly being improved.

You can run Lakka on any Raspberry Pi, although the earlier models don't allow for as extensive a range of emulators as the Raspberry Pi 3. There are also features that help you organise your growing gaming collection and take screenshots of the in-game action. For now, though, we're looking solely at getting you up and running with a classic arcade title.

Enjoy retro games such as Super Tank on your Raspberry Pi



Input User 1 Binds — 21/11 14:05

User 1 Bind Default All
User 1 Save Autoconfig
User 1 B button (down) (Key: z)
User 1 Y button (left) (Key: a)
User 1 Select button (Key: rshift)
User 1 Start button (Key: enter)
User 1 Up D-pad (Key: up)
User 1 Down D-pad (Key: down)
User 1 Left D-pad (Key: left)
User 1 Right D-pad (Key: right)
User 1 A button (right) (Key: x)
User 1 X button (top) (Key: s)
User 1 L button (shoulder) (Key: q)

Navigate to Settings > Input > Input User 1 Binds and insert a controller to configure it

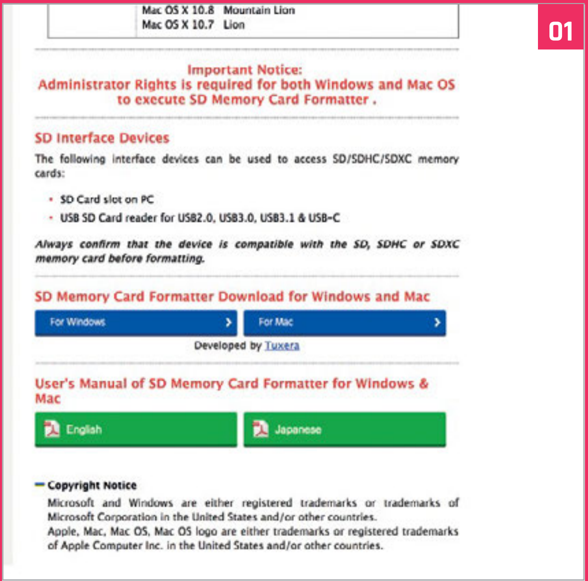
Enter the buttons you need to map to your controller. Tap the corresponding controller button

Lakka's joypad layout is inspired by the pad for Nintendo's SNES console

HOW TO PLAY GAMES VIA LAKKA

>STEP-01
Format your SD card

We'll install Lakka to a blank microSD card using the OS installer NOOBS. Format the card by downloading SD Formatter on a computer from **magpi.cc/2A030Ei**. Click 'For Windows' or 'For Mac' depending on your machine.



01

Mac OS X 10.8 Mountain Lion
Mac OS X 10.7 Lion

Important Notice:
Administrator Rights is required for both Windows and Mac OS to execute SD Memory Card Formatter .

SD Interface Devices

The following interface devices can be used to access SD/SDHC/SDXC memory cards:

- SD Card slot on PC
- USB SD Card reader for USB2.0, USB3.0, USB3.1 & USB-C

Always confirm that the device is compatible with the SD, SDHC or SDXC memory card before formatting.

SD Memory Card Formatter Download for Windows and Mac

For Windows › For Mac ›

Developed by Tuxera

User's Manual of SD Memory Card Formatter for Windows & Mac

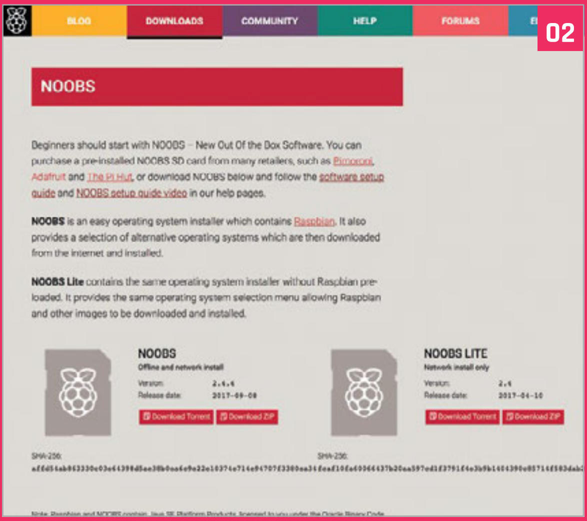
English Japanese

Copyright Notice

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Apple, Mac, Mac OS, Mac OS logo are either trademarks or registered trademarks of Apple Computer Inc. in the United States and/or other countries.

>STEP-02
Download NOOBS

Accept the terms and conditions, install SD Formatter, and launch. Ensure the drive letter matches your card and click Format. Now visit **raspberrypi.org/downloads** and click the NOOBS icon. Select Download ZIP and a folder to save it to.



02

NOOBS

Beginners should start with NOOBS – New Out Of the Box Software. You can purchase a pre-installed NOOBS SD card from many retailers, such as Pimoroni, Adafruit and The Pi Hut, or download NOOBS below and follow the software setup guide and NOOBS setup guide video in our help pages.

NOOBS is an easy operating system installer which contains Raspbian. It also provides a selection of alternative operating systems which are then downloaded from the internet and installed.

NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

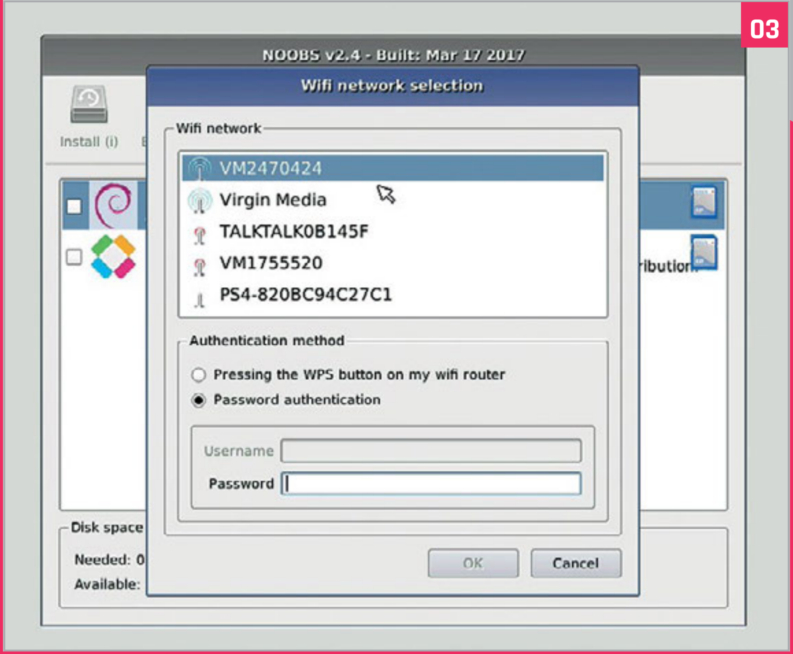
NOOBS
Offline and network install
Version: 2.4.4
Release date: 2017-09-08
Download Torrent Download ZIP

NOOBS LITE
Network install only
Version: 2.4
Release date: 2017-04-10
Download Torrent Download ZIP

SHA-256: [illegible]

SHA-256: [illegible]

Note: Raspbian and NOOBS contain Java SE Platform Products licensed to you under the Oracle Binary Code



03

NOOBS v2.4 - Built: Mar 17 2017

Wifi network selection

Wifi network
VM2470424
Virgin Media
TALKTALK0B145F
VM1755520
PS4-820BC94C27C1

Authentication method
○ Pressing the WPS button on my wifi router
● Password authentication

Username
Password

OK Cancel

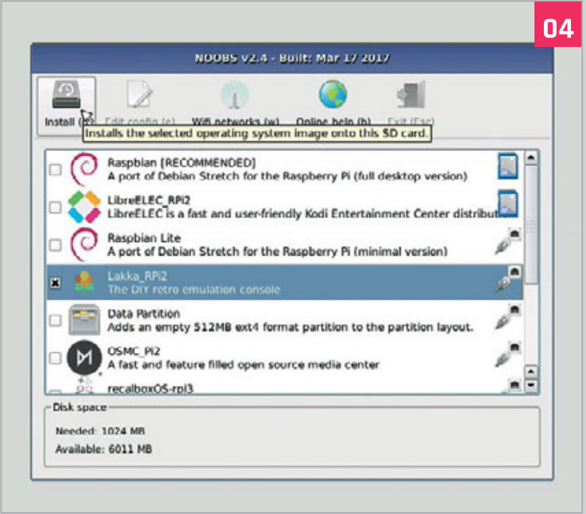
Disk space
Needed: 0
Available:

>STEP-03
Insert card into the Pi

Extract the files from the zip and drag the files to the SD card. Now eject and insert the card into your Pi before powering up. On the NOOBS screen, click the 'Wifi networks' icon.

>STEP-04
Install Lakka

Select your network, using your keyboard to enter your password. A list of OSes will then appear, so select Lakka and click the Install icon. Lakka will now download and install. Click OK when it's done.



04

NOOBS v2.4 - Built: Mar 17 2017

Installs the selected operating system image onto this SD card.

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

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

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

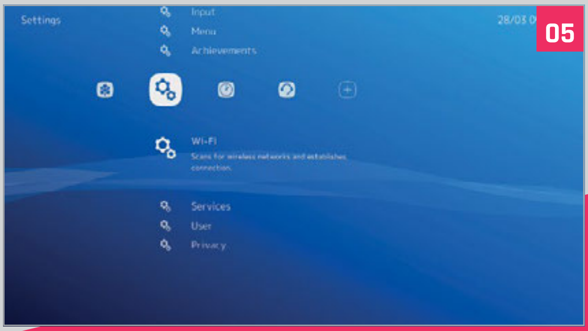
☒ Lakka_RPi2
The DIY retro emulation console

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

□ OSMC_Pi2
A fast and feature filled open source media center

recalboxOS-rpi3

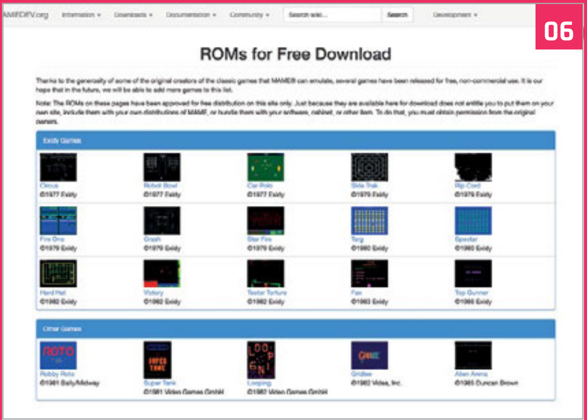
Disk space
Needed: 1024 MB
Available: 6011 MB



>STEP-05

Connect again

Use your cursor keys to navigate Lakka's menus, hitting **ENTER** to select and **BACKSPACE** (←) to go back. Set up a wireless network again by selecting Wi-Fi under the Settings menu, choosing your network, and entering your password.



ROMs for Free Download

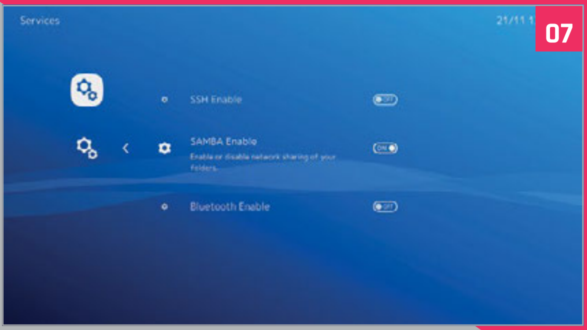
Thanks to the generosity of some of the original creators of the classic games that MAME® can emulate, several games have been released for free, non-commercial use. It is our hope that in the future, we will be able to add more games to this list.

Note: The ROMs on these pages have been approved for free distribution on this site only. Just because they are available here for download does not entitle you to put them on your own site, include them with your own distributions of MAME, or bundle them with your software, cabinet, or other item. To do that, you must obtain permission from the original owners.

>STEP-06

Finding a ROM

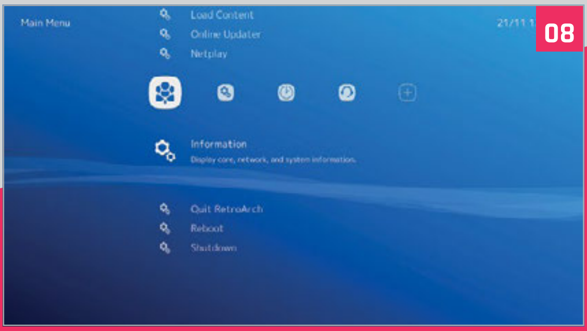
Now it's time to find and play a game. Let's start with a classic from 1981 called Super Tank. Go to **mamedev.org/roms** on your computer and click it to download. MAME games must be zipped, so use an app such as 7-Zip to compress the entire folder.



>STEP-07

Copy it to Lakka

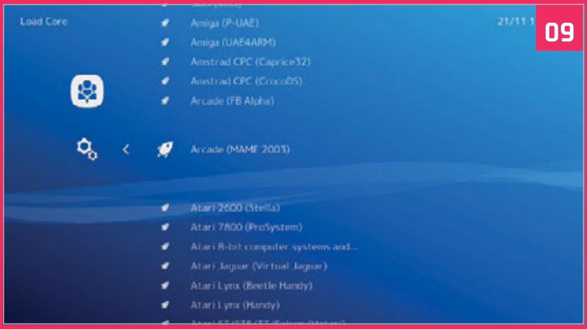
With your Raspberry Pi and computer on the same network, go to the Settings menu in Lakka on your Pi and select Services. Highlight SAMBA Enable and turn it on by pressing **ENTER**.



>STEP-08

Find the folder

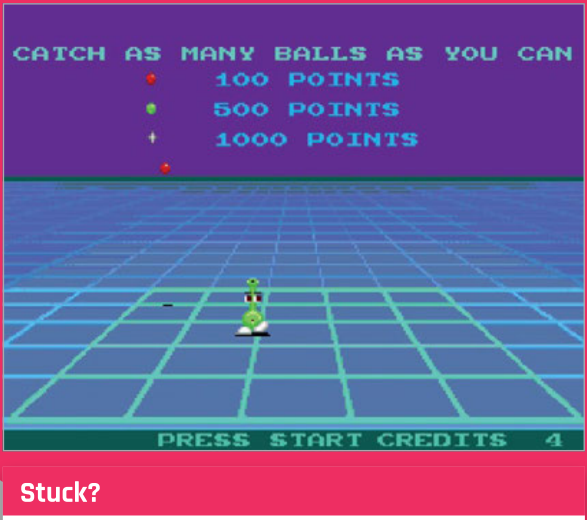
Lakka may appear in the left-hand column of your other computer's file browser (File Explorer on a PC or Finder on a Mac). If not, select Lakka's main menu on your Raspberry Pi, choose Information and note the IP address. Enter that into the file browser in the format: **\\insert.full.ip.address**



>STEP-09

Play the game

Open the Lakka folder on your computer and copy your zipped game to the ROMS folder. On your Pi, go to Load Content in the Lakka menu and find your game. Click it before selecting Load Archive. Choose Arcade (MAME) as the emulator to play it on. Enjoy.



Stuck?

Then it's worth heading over the Lakka forums for friendly help and advice at: **magpi.cc/2jbCRdC**

NEW HACK | MAKE | BUILD | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
December 2017
Issue #01
BECKY STERN
How to get paid for making fun things
BUILD THINKING MACHINES
Harness the power of artificial intelligence
SMOKE YOUR OWN BACON
HANDHELD CONSOLES FOR HACKERS
BOLDPORT: BUGS, BITS & BLINKENLIGHTS
TRANSISTORS 3D PRINTING DUCT TAPE CHANGING THE WORLD

FREQUENTLY ASKED QUESTIONS

Your technical hardware and software problems solved…

NEED A PROBLEM SOLVED?

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

SD CARD MAINTENANCE

WHAT CAN GO WRONG WITH THE SD CARD?

Corruption

SD cards, like any form of computer storage, can become corrupted. This means certain (or all) sections of the memory become unreadable by a computer like the Raspberry Pi. In this case, your files will be generally irretrievable and the SD card will become useless.

End of life

Solid-state storage solutions like SD cards, USB flash drives and SSDs have a finite lifespan which is dictated by how much you use them. Don't worry, though: you're very unlikely to ever use your Pi's SD card enough to break it that way.

Physical problems

SD cards, especially the microSD cards used in most Raspberry Pis, are small and flimsy. It's not uncommon to lose them when transporting them somewhere and while they're not easy to snap, they are definitely breakable.

WHAT CAN I DO TO KEEP MY SD CARD SAFE?

Properly shut down

Corruption usually occurs when data is being written and suddenly the storage is removed or there's a loss of power. You should never remove the SD card while the Pi is running anyway, but you might remove the power to turn it off. Always make sure to shut it down via the Shutdown option in the desktop menu, or with the Terminal command **`sudo shutdown -h now`**.

Boot to RAM

Not every Pi is used in the same way, and sometimes you won't be able to always turn it off properly. In the case where you need to regularly pull the plug on it, you can use piCore (**magpi.cc/2ueA7UI**) as your distro. This loads into RAM at boot, so won't be writing when you pull the plug. If you want to stick with Raspbian, try making it read-only: **magpi.cc/2i1LkQ5**.

Keep it safe

SD cards usually come with protective cases, which can aid in transporting them if you need to carry them around on their own. Otherwise, we find that keeping the card in a Pi we're transporting (preferably in its own case) means it stays relatively safe and is hard to lose.

CAN I BACK UP THE SD CARD?

Documentation and instructions

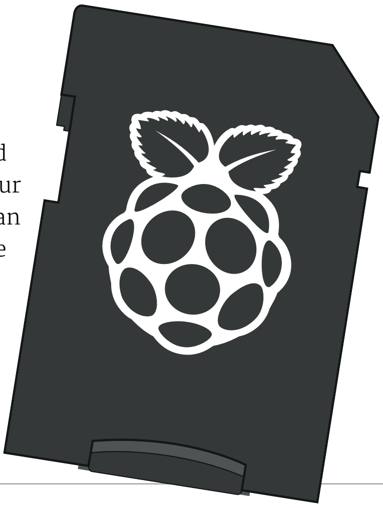
The quickest and most low-tech way to back up your important files is simply to transfer them to a USB stick. You can also save files to another computer over the network, or simply plug the SD card into a computer that can read ext file systems and copy them.

Cloud backup

While services such as Dropbox and Google Drive don't support Raspbian very well, you can set up Dropbox to manually sync files on the Raspberry Pi. With that in place, you can then automate the timing of this manual sync and have it behave almost like standard Dropbox: **magpi.cc/2c9NnPC**.

Card clone

Creating a complete copy of your SD card means you also get to save your software and settings along with your important files. You can create a copy using the SD Card Copier tool built into Raspbian, or simply plug the SD card into another PC and make a copy from there.



FROM THE RASPBERRY PI FAQ RASPBERRYPI.ORG/HELP

WHAT SOC ARE YOU USING?

All models of Raspberry Pi use Broadcom SoCs containing a VideoCore IV GPU, but with various ARM CPU cores.

The original Raspberry Pi used a Broadcom BCM2835. This contains a single-core ARM1176JZFS with floating point, running at 700MHz, and a VideoCore IV GPU. The GPU is capable of Blu-ray-quality playback, using H.264 at 40MBits/s. It has a fast 3D core, accessed using the supplied OpenGL ES 2.0 and OpenVG libraries.

The Pi 2 Model B originally used the Broadcom BCM2836. This contains a quad-core ARM Cortex-A7 processor with floating point and NEON, running at 900MHz, and the same VideoCore IV GPU that is in the earlier models of Raspberry Pi.

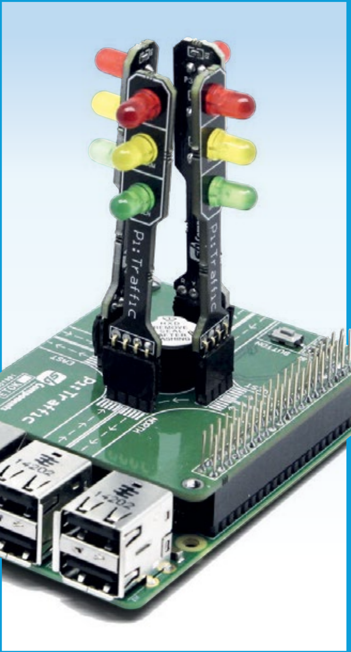
The Pi 3 Model B uses the Broadcom BCM2837 containing a quad-core ARM Cortex-A53 running at 1.2GHz. Its GPU capabilities are equivalent to the Pi 2. Some versions of the Pi 2 now use the same SoC as the Pi 3, but downclocked to match the speed of the original Pi 2 SoC. In the future, every Pi 2 will use this newer SoC.

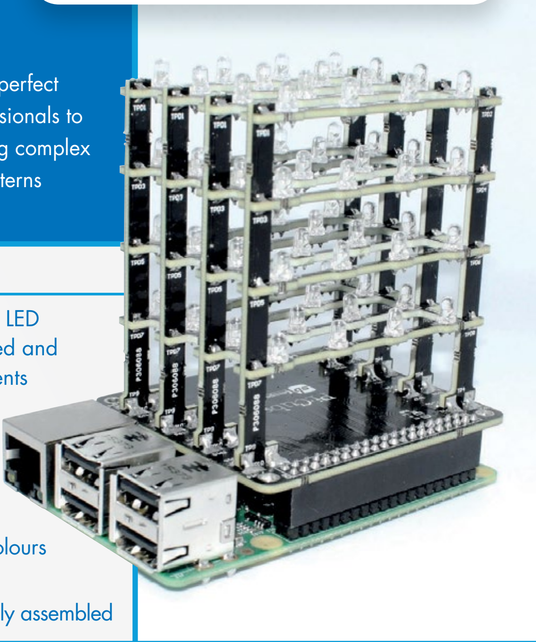
WHAT IS AN SOC?

A system on a chip (SoC) is a method of placing all necessary electronics for running a computer on a single chip. Instead of having separate chips for the CPU, GPU, USB controller, RAM, northbridge, southbridge, and so on, everything is compressed down into one tidy package.

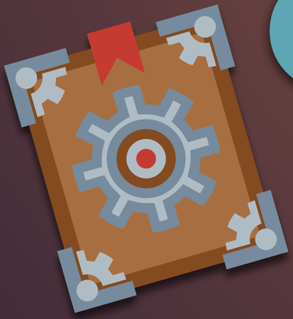
WHY DID YOU SELECT THE SOC?

Cost and performance.





CREATE YOUR OWN DIGITAL ADVENTURE GAMEBOOK with Twine



Using Twine 2, anyone can easily create a choice-based interactive fiction game and publish it online

K.G. ORPHANIDES

K.G. is a tech journalist whose experiments in interactive fiction have ranged from Multi-User Shared Hallucinations to gamebooks and traditional text adventures.

twitter.com/KGOrphanides

You'll Need

- twinejs magpi.cc/2zpogTD
- Chromium
- Your imagination, and maybe some scrap paper

Text-based games have a legacy almost as long as modern computing. Created on a PDP-10 mainframe in 1975, Will Crowther's Colossal Cave Adventure is generally recognised as the first text adventure, allowing players to enter simple commands like GO NORTH and TAKE LAMP.

It inspired the entire text adventure genre, with Infocom's Zork (1979) and Adventure International's Adventureland (1978) becoming some of the first commercial parser-based text adventures.

Meanwhile, a different kind of interactive fiction was coming to bookshops. In the UK, Games Workshop co-founders Steve Jackson and Ian Livingstone added tabletop role-playing elements to their Fighting Fantasy gamebooks, the first of which, *The Warlock of Firetop Mountain*, was published by Puffin Books in 1982. With shorter passages, more choices, a character sheet, and a dice-based combat system, Fighting Fantasy would define adventure gamebooks for generations of British kids.

It's these that we'll be taking inspiration from with this tutorial, showing you how to implement not just a branching, choice-based plot, but also exciting action, lucky rolls of the dice, and a character who can be affected – and even killed – by the challenges they face.

Fighting Fantasy gamebooks set the standard for fantasy adventures with branching paths to explore

INTRODUCING TWINE

In this tutorial, we're going to use Twine 2 and its default Harlowe programming language to create an interactive fantasy fiction adventure. Twine is specifically built to be easy to use. You can create a simple, choice-based story without writing a single line of code, but it also gives you a surprisingly flexible programming environment that you can use to implement logic, stats, event flags, and random dice rolls.

INSTALLING AND RUNNING TWINE 2

Stand-alone Twine desktop clients are available for a number of systems, but not ARM-based computers like the Raspberry Pi – for now, at any rate. However, there's a ready-made local web version of the engine called TwineJS maintained by original creator Chris Klimas, and all you need to run it is a web browser.

1. Go to magpi.cc/2zpogTD and download the latest version of Twine – the plain zip file, with no operating system mentioned. At the time of writing, that's twine_2.1.3.zip.
2. Unpack it into its own directory.
3. Open index.html to start Twine.

It runs in a browser, but all files are stored locally. Your game can be edited and built fully offline, but unlike some other implementations of the client, you'll have to manually export your works in progress (which is actually best practice for all iterations of Twine).

While Twine 2 should work perfectly in any JavaScript-enabled browser, we're going to run it in Chromium and we suggest that you do the same while following this tutorial.

THE ROAD TO ADVENTURE

To go with this tutorial and illustrate the techniques used in it, we've created a demo game, The Road to Adventure™, which you can find online at magpi.cc/2ydf107.

To import it into your own copy of Twine, download the HTML source file linked on the main game page, open Twine, and select 'Import from file' on the right-hand bar of the Twine home screen.

YOUR First GAME

The first time you run Twine 2, a few tutorial paragraphs point you at some online documentation and tell you how to save your work. After that, you'll be taken to Twine's main index screen, where we can begin writing our game.

GETTING STARTED

Click the green Story button in the bar on the right and name your tale when prompted. A graph-paper-like screen will appear with a single box in the middle, marked Untitled passage. Double-click on the box to open the editing screen

Passage boxes are where you'll spend most of your time in Twine. Click into the title bar – currently marked Untitled Passage – and give it a name. Only you will be able to see this title, so you can call it whatever you like, as it won't be visible to the player.

Below is the main box where you'll enter text for your players to read. As it suggests, double-click on the sample text currently in the passage and start typing to begin composing your own story.

There's also a Tags box, invisible to the player, which you can use to mark passages for your own reference or to apply repeated text or code to your game, for example in header and footer passages. While this is beyond the scope of our tutorial, you can learn more about it in Twine's documentation.

Everything you type in a passage is automatically saved to your browser's cache in real time. This means that, even if your computer crashes, nothing will be lost. However, it's very important to back up your game (see 'Save often') before making any major changes to its text or code.

YOUR FIRST LINK

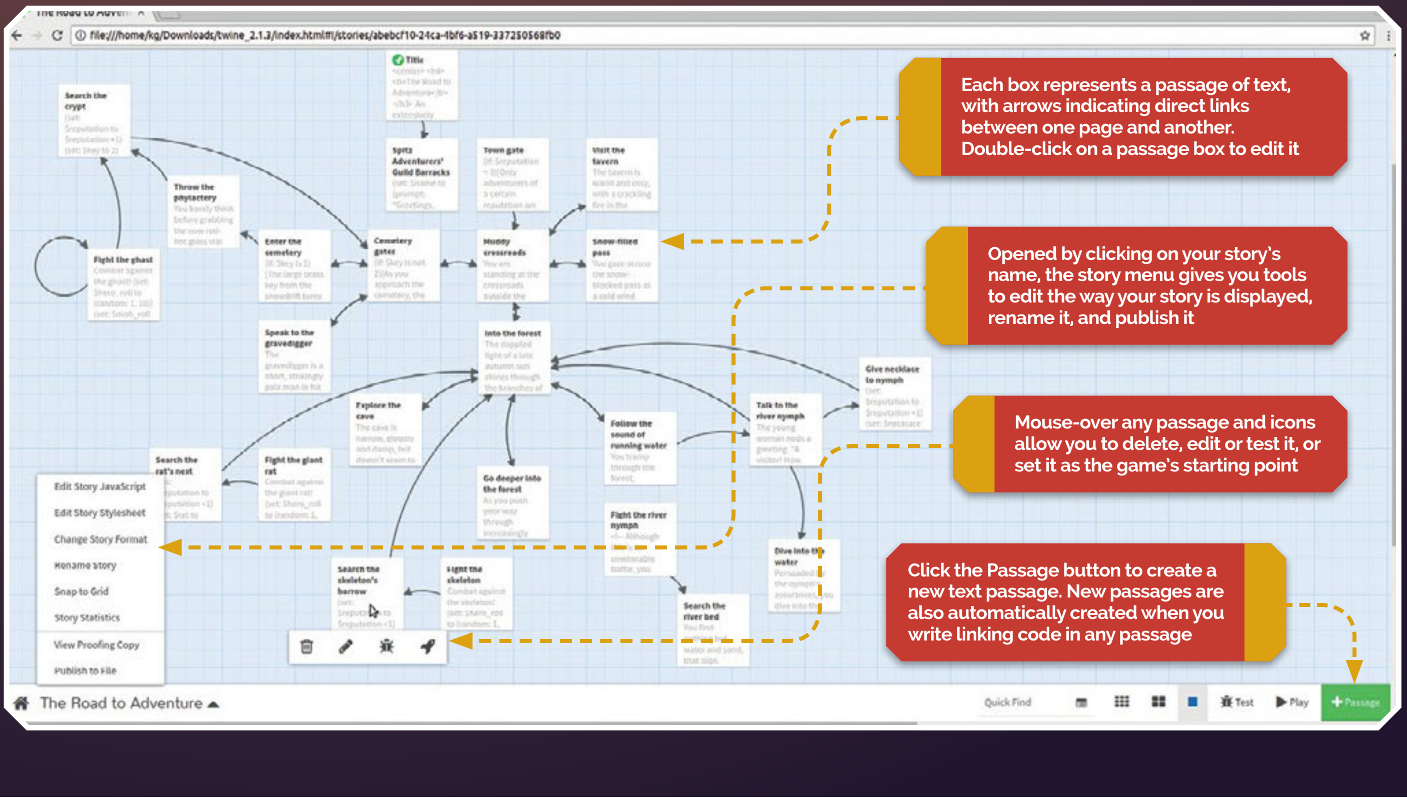
In our first gameplay passage – called Spitz Adventurers' Guild Barracks in our demo game – we've described the barracks of an adventurer's guild and set up the plot for our story. Our character's motivation is simple: hunt monsters for fame

RESOURCES

Twinery
twinery.org

Harlowe manual
twine2.neocities.org

Publish for free on Itch.io
itch.io



Each box represents a passage of text, with arrows indicating direct links between one page and another. Double-click on a passage box to edit it

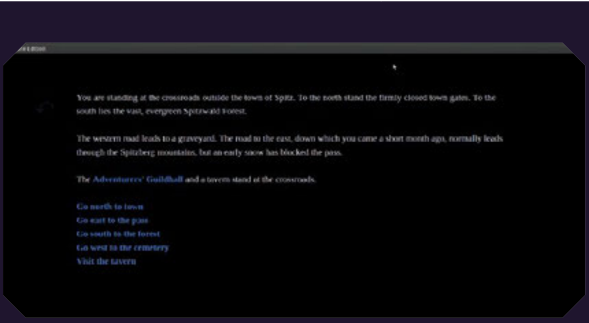
Opened by clicking on your story's name, the story menu gives you tools to edit the way your story is displayed, rename it, and publish it

Mouse-over any passage and icons allow you to delete, edit or test it, or set it as the game's starting point

Click the Passage button to create a new text passage. New passages are also automatically created when you write linking code in any passage

A link to the future

```
(link: "Go north to town")[(goto: "Town gate")]
[[Go east to the pass->Snow-filled pass]]
[[Go south to the forest->Into the forest]]
[[Go west to the cemetery->Cemetery gates]]
[[Visit the tavern]]
```



You are standing at the crossroads outside the town of Spitz. To the north stand the firmly closed town gates. To the south lies the vast, evergreen Spitzwald Forest.

The western road leads to a graveyard. The road to the east, down which you came a short month ago, normally leads through the Spitzberg mountains, but an early snow has blocked the pass.

The Adventurers' Guildhall and a tavern stand at the crossroads.

Go north to town
Go east to the pass
Go south to the forest
Go west to the cemetery
Visit the tavern

and fortune. But now we've got to get them to the next stage of their adventure.

To do this, we're going to create a simple link. When using Twine and Harlowe, any string of text you put between two sets of square brackets `[[like this]]`, Twine will automatically create a passage with the same name as your link. With that done, click the X in the top right to close.

MAKE ROOM, MAKE ROOM

Back on the map screen, you'll see that a new passage has been created. If we rename it, the link pointing to it will be automatically updated to reflect this. However, you can also create a link that points to a passage with a name that's different to the link text.

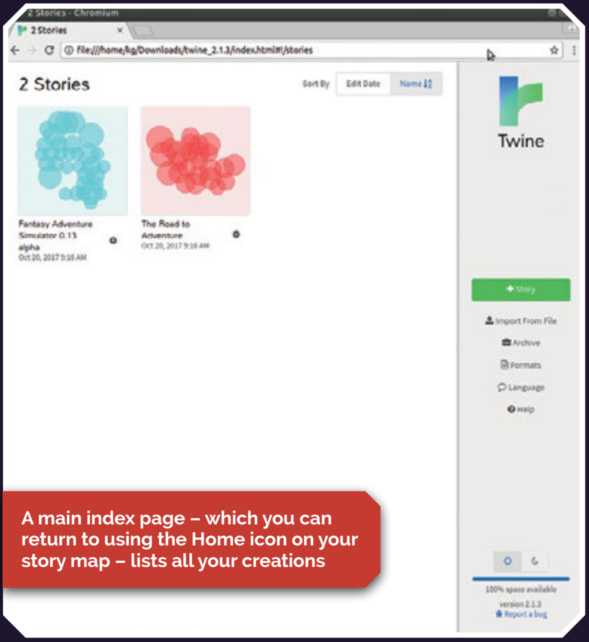
In our code example (see 'A link to the future') we have used different linking methods, one of which involves the same bracketed format we used earlier, but with an arrow pointing at the new passage name `[[so this->leads to this]]`.

A third method involves our first introduction to Harlowe's built-in macros. The `(link:)` macro makes a link that will appear on our current page. When the player clicks on it, the game will carry out any commands within a pair of brackets that follow it.

In this case, that's just a `(goto:)` macro, which will do the same job as a normal link. However, it doesn't automatically create your destination passage and won't display a link arrow on the map view.

That means that if we create a passage with this bit of code in it – `(link: "Go north")[(goto: "Town gate")]` – we then have to manually create a new passage called Town gate, because one will not automatically be created when using this method.

We'll generally only use this kind of link when we need to call other functions when it's clicked. Later, we'll be using `(link:)` to tell Twine to set variables, check for previously set states, and carry out random dice rolls as we move from one passage to the next.



A main index page – which you can return to using the Home icon on your story map – lists all your creations

SAVE OFTEN

When you're using a browser-based version of Twine, either locally or online, all your development work is stored in your browser's cache. That means it can be lost if you accidentally clear your cache, so it's important to save regularly.

The easiest way to save a backup copy of your game while working on it is to click the upward-pointing arrow in the bottom bar, to the right of your game's name, and select the Publish to File option from the pop-up menu.

THE MOST Important MACROS & COMMANDS

Now we've got to grips with linking passages – the most fundamental aspect of Twine – we can move on to using more of Harlowe's macros and stringing them together to add gameplay to our interactive story.

BRACKETS ARE REALLY IMPORTANT

In any programming language, structuring your code correctly is important when it comes to making it human-readable and ensuring that you've got all your punctuation in the right places.

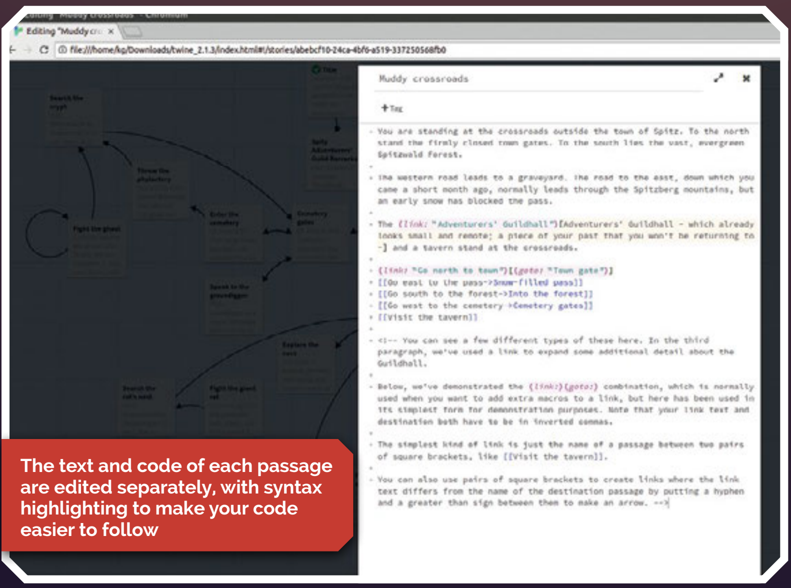
Twine helpfully has built-in syntax highlighting to make it easy to see which bits of your code go with each other, but you'll also want to make sure you don't lose track of how many square brackets you're using.

When using conditional macros such as `(if:)`, `(else-if:)`, and `(else:)`, remember that anything that happens in response to those conditions must be placed between a pair of `[square brackets]`.

You can nest multiple macros in these, so although it's not essential for the correct functioning of your program, it a good idea to use tabs and carriage returns to keep large blocks of code tidy – see our examples for a demonstration.

Twine automatically saves every change you make instantly, and you can't rely on a **CTRL+Z** undo function if you accidentally delete a line you wanted.

Save regular copies of your work and when working on a complex bit of code, it's a good idea to cut and paste it to a new passage to make it easier to revert to the original version if required.


Editing "Muddy cro
file:///home/kp/Downloads/twine_2.1.3/index.html#!/stories/abebcf10-24ca-4bf6-a519-337250568fb0
Muddy crossroads
+ Tag

```
You are standing at the crossroads outside the town of Spitz. To the north stand the firmly closed town gates. To the south lies the vast, evergreen Spitzwald forest.

The western road leads to a graveyard. The road to the east, down which you came a short month ago, normally leads through the Spitzberg mountains, but an early snow has blocked the pass.

The (link: "Adventurers' Guildhall")[Adventurers' Guildhall - which already looks small and remote; a piece of your past that you won't be returning to -] and a tavern stand at the crossroads.

(link: "Go north to town")[(goto: "Town gate")]
[[Go east to the pass->Snow-filled pass]]
[[Go south to the forest->Into the forest]]
[[Go west to the cemetery->Cemetery gates]]
[[Visit the tavern]]

<!-- You can see a few different types of these here. In the third paragraph, we've used a link to expand some additional detail about the Guildhall.

Below, we've demonstrated the (link:)(goto:) combination, which is normally used when you want to add extra macros to a link, but here has been used in its simplest form for demonstration purposes. Note that your link text and destination both have to be in inverted commas.

The simplest kind of link is just the name of a passage between two pairs of square brackets, like [[Visit the tavern]].

You can also use pairs of square brackets to create links where the link text differs from the name of the destination passage by putting a hyphen and a greater than sign between them to make an arrow. -->
```

The text and code of each passage are edited separately, with syntax highlighting to make your code easier to follow

VARIABLES (SET:) UP THE STORY

In our character creation code (see 'Set your stats'), we've used macros to define our hero's name, hit points, attack, luck, and a few other stats. We start by asking the player to name their hero, which we'll use the `(set:)` macro to save as a variable called **`$name`**. The `(prompt:)` macro creates a text entry box. The second bit of text in quotes, **`"Avatar"`**, is the suggested text for the entry box, in case the player doesn't want to come up with their own name.

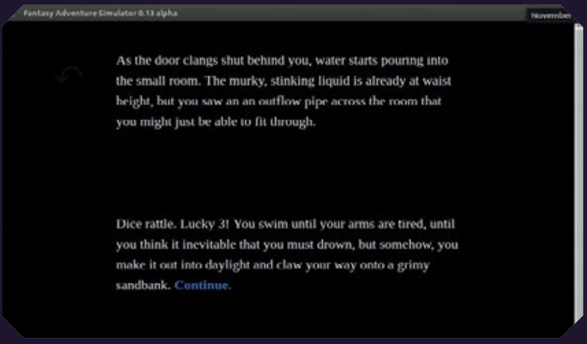
Everything else is set when the player clicks the 'It's time to go' link. The `(link:)` macro defines

Set your stats

```
(set: $name to (prompt: "Greetings, adventurer. What
is your name?", "Avatar"))

So, $name, it's your last morning in the barracks
maintained by the Adventurers' Guild… (full text ex-
cised for brevity).

(link: "It's time to go")[
    (set: $hp to 15)
    (set: $max_hp to 15)
    (set: $reputation to 0)
    (set: $atk to 11)
    (set: $luck to 7)
    (set: $battlecry to "You shall die by my sword!")
    (goto: "Muddy crossroads")
    ]
```

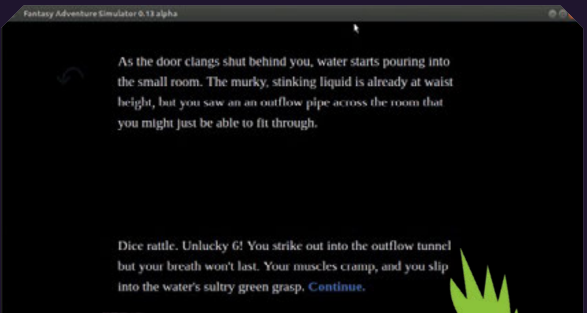
Roll for luck

```
(link: "Swim for it - *roll for luck*")[
    (set: $roll to (random: 1, 10))
    (if: $luck > $roll )[
    Dice rattle. Lucky $roll! You swim until
your arms are tired, until you think it
inevitable that you must drown, but somehow,
you make it out into daylight and claw your
way onto a grimy sandbank. (link: "Continue.")
[(goto: "Beneath the docks")]
    ]
    (else:)[
    Dice rattle. Unlucky $roll! You strike
out into the outflow tunnel but your breath
won't last. Your muscles cramp, and you slip
into the water's sultry green grasp. (link:
"Continue.")[(goto: "You have drowned")]
    ]
]
```



As the door clangs shut behind you, water starts pouring into the small room. The murky, stinking liquid is already at waist height, but you saw an an outflow pipe across the room that you might just be able to fit through.

Dice rattle. Lucky 3! You swim until your arms are tired, until you think it inevitable that you must drown, but somehow, you make it out into daylight and claw your way onto a grimy sandbank. Continue.



As the door clangs shut behind you, water starts pouring into the small room. The murky, stinking liquid is already at waist height, but you saw an an outflow pipe across the room that you might just be able to fit through.

Dice rattle. Unlucky 6! You strike out into the outflow tunnel but your breath won't last. Your muscles cramp, and you slip into the water's sultry green grasp. Continue.

the text the player sees. Then we use **(set:)** macros to set our variables. Here, **$luck** is set to the number 7. If you're setting a variable to a text string, such as our **$battlecry** variable, you have to enclose it in inverted commas.

The final **(goto:)** line tells the program where the link will go after everything has been set. From now on, any time we put **$name** or **$luck** into the text of a passage, the game will display those variables.

So if we create a passage introducing a tavern keeper, she can say **"What ho, $name! Come over and have a drink!"**, and the player character's name will appear in the sentence.

ROLL THE BONES

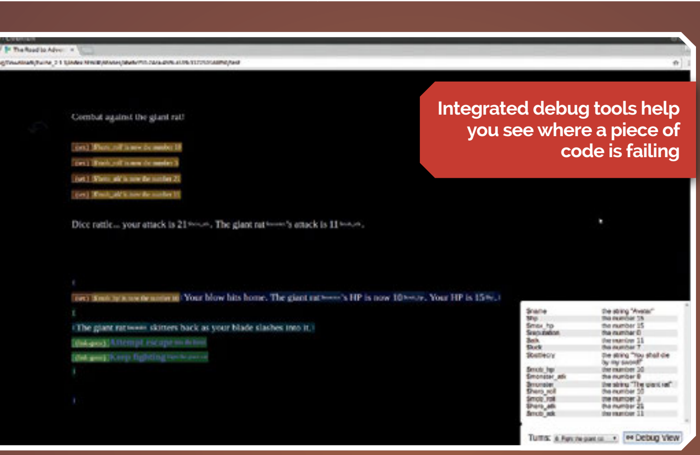
Rolling dice – or rather, generating random numbers – is a key factor of games and is also really simple to do using Harlowe. Just use the **(random:)** macro with the bottom and top of the random number range you want. To roll a six-sided die, you put **(random: 1,6)**, and Twine will spit out the result. To make it part of a game, we'll use it to roll against a stat.

The 'Roll for luck' box code demonstrates one way of doing this. Our **(link:)** shows the player a link saying 'Swim for it – roll for luck'. Telling the player when they're rolling dice helps to create the feel of a gamebook.

Here, we **(set:)** a variable called **$roll** to the result of a **(random:)** number between 1 and 10, to simulate a ten-sided die.

(if:) the character's **$luck** stat, which we **(set:)** during character creation, is greater than or equal to (**>=**) the random number we stored in the **$roll** variable just now, our hero swims to safety and, when they select Continue, washes up in an area called 'Beneath the docks' thanks to a **(goto:)** command.

As this is a simple either/or situation, we can use the basic **(else:)** command to handle what happens otherwise. So, if the player's luck is less than the random number, they're sent to a passage called 'You have drowned' using a **(link:)[(goto:)]** macro command.



Combat against the giant rat!

Dice rattle... your attack is 21. The giant rat's attack is 11.

Your blow hits home. The giant rat's HP is now 10. Your HP is 15.

The giant rat skitters back as your blade slashes into it.

Attempt escape

Keep fighting

$name	the string "Avatar"
$hp	the number 15
$max_hp	the number 15
$reputation	the number 0
$atk	the number 11
$luck	the number 7
$battlecry	the string "You shall die by my sword!"
$mob_hp	the number 10
$monster_atk	the number 8
$monster	the string "The giant rat"
$hero_roll	the number 10
$mob_roll	the number 3
$hero_atk	the number 21
$mob_atk	the number 11

Integrated debug tools help you see where a piece of code is failing

FIGHT ME

It's not a classic fantasy adventure without a bit of monster-bashing, so our final section of sample code shows you how to set up a combat routine.

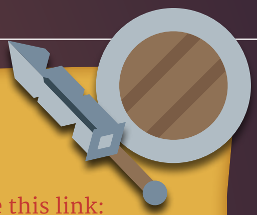
The nice thing about this bit of code is that you can reuse it anywhere you like with only a few tweaks to change the name of the monster, set by the **$monster** variable; its hit points, set in **$mob_hp**; and its base attack, defined by **$monster_atk**.

These are all defined using the **(set:)** command with a link that goes from a previous passage to the combat sequence itself. They're locked in when the player selects the 'Fight the skeleton' link.

The combat sequence is designed as a loop that responds to the player's decisions. Each loop represents one round of combat. If neither the player character nor the monster are reduced to zero hit points during combat, the player will be given the option of either escaping or continuing with another round of combat.

At the beginning of each round, the game rolls a ten-sided die for both player and monster. The results are added to their base attack stats. Whoever has the highest score knocks two hit points off the other. If both scores are equal, neither hits.

Note that we use greater than (**>**) or less than (**<**) signs for most of these comparisons, but if we want to see if their attacks equal each



Fight the skeleton

THE SETUP

In the passage before your fight sequence, include this link:

```
(link: "Fight the skeleton")[(set: $mob_hp to 10)(set:
$monster_atk to 6)(set: $monster to "The skeleton")(goto:
"Fight the skeleton")]
```

THE FIGHT

This is our main combat sequence.

```
Combat against the skeleton!

(set: $hero_roll to (random: 1, 10))
(set: $mob_roll to (random: 1, 10))
(set: $hero_atk to $atk + $hero_roll)
(set: $mob_atk to $monster_atk + $mob_roll)

Dice rattle... your attack is $hero_atk. $monster's attack is
$mob_atk.

(if: $mob_atk > $hero_atk)[
    (set: $hp to $hp - 2)[$monster strikes you. $monster's HP
is $mob_hp. Your HP is now $hp.]
    (if: $hp > 0)[
        [You reel but keep your footing.]
        (link-goto: "Attempt escape", "Into the forest")
        (link-goto: "Keep fighting", "Fight the skeleton")
        ]
    (else:)[
    [YOU HAVE DIED]
    (link:"Restart")[(reload:)]
    ]
    ]

(else-if: $hero_atk > $mob_atk)[
    (set: $mob_hp to $mob_hp - 2)[Your blow hits home with a
crack of splintering bone. $monster's HP is now $mob_hp. Your
HP is $hp.]
    (if: $mob_hp > 0)[
    [$monster stumbles but regains its balance.]
        (link-goto: "Attempt escape", "Into the forest")
        (link-goto: "Keep fighting", "Fight the skeleton")
        ]
    (else:)[
    [[YOU ARE VICTORIOUS!->Search the skeleton's barrow]]
    ]
    ]

(else-if: $mob_atk is $hero_atk)[
    [You lunge at each other but both miss. $monster's HP is
$mob_hp. Your HP is $hp.]
    (link-goto: "Attempt escape", "Into the forest")
    (link-goto: "Keep fighting", "Fight the skeleton")
    ]
```

THE PAYOFF

This text goes into a new passage called 'Search the skeleton's barrow'.

```
(set: $reputation to $reputation +1)
(set: $skeleton to "defeated")

The skeletal warrior collapses into a shower of bone shards,
just in front of a shallow barrow made of stone and earth.
Braving its musty-smelling depths, you search the tomb.

Your reputation is now $reputation.
[[Return to the forest->Into the forest]]
```

other, we have to use the word **is**, as in **(else-if: $mob_atk is $hero_atk)[outcome]**.

The whole routine is very simple, but it does the job. As you become more familiar with Harlowe, you could add extra features, such as modifying the **$atk** variable to increase the amount of damage the player does if they get a shiny new axe, or requiring a luck roll to successfully escape from the monster.

If the player dies, we offer them the opportunity of restarting the game from the beginning. By default, Twine has a back button, so they'll probably just flip back to before the fight started – as most people do with physical gamebooks – but the **(reload:)** link is great for people who want to play an honest game or if you use Twine's Stylesheet editor to remove the back button.

If the player wins the fight, then we take them to a victory screen where we can award them loot, move them further along a fixed story path, or – if we want them to be able to return to the same location in the future – **(set:)** a variable to mark the enemy as defeated so that it doesn't appear a second time.

DEBUG ZAPPER

Twine has a built-in debugging tool that'll help you spot where and why your code is going wrong. To run the game from the beginning in the debugger, use the Test button at the bottom of your game's main map display.

The test interface runs your game while tracking turns, the passage title, and set variables at the bottom right. To see more details of the code, press the Debug View button. You can also test individual passages via the bug icon that appears when you mouse-over any passage, but remember that it'll run without any variables that you might have set in previous passages.

ADD MULTIMEDIA, PUBLISH YOUR GAME

ALL SINGING, ALL DANCING

Twine 2 has a reputation for being less multimedia-friendly than Twine 1, which could encode images as part of its game file, but it's really very simple to use HTML to include pictures, sound, and even video.

To add pictures and join the long tradition of text-and-graphic adventure development, just use HTML <img> tags to display images and <audio> tags for sound.

If you want to publish your game online, your sound and images will have to be hosted somewhere. In this tutorial, we're going to use **itch.io** to publish our game, and it can host your images for you. That means you can just put them in the same directory as your game file and use their file name to refer to them in your code, rather than a full URL.

You can also embed YouTube videos, which can be effective if you want to add cut-scenes to a fantasy adventure, or interviews with suspects to a murder mystery. Just upload your video to YouTube, copy the embed code from YouTube's Share options, and paste it into the Twine passage you want it to appear in.

PUBLISH AND BE QUITE CHEERY, ACTUALLY

The beauty of a Twine game is that it exists as a single HTML file that you can upload anywhere that you have a bit of web space. There are also free hosts that specifically cater to Twine developers.

Embed multimedia using HTML

HTML code can be placed anywhere within a passage.

Embed and centre an image:

```
<center><img src="yourimage.jpg"; alt="victory
gate"></center>
```

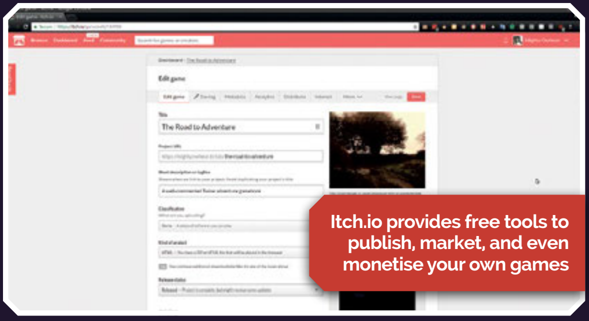
Embed a sound file:

```
<audio src="youraudiofile.mp3"; autoplay>
```

Embed a YouTube video:

```
<iframe width="560" height="315"
src="yourYouTubevideoURL" frameborder="0"
allowfullscreen></iframe>
```

Or use YouTube's share and embed options to automatically generate the correct code.



Itch.io provides free tools to publish, market, and even monetise your own games

However, if you want a polished and professional set of tools for publishing and even monetising your games, we recommend setting up a free account on **itch.io**. Whether you want to distribute your games for free, set up a donation box for your fans, or release games commercially, **itch.io**'s tools make the job simple. You can upload blogs, screenshots and videos to keep players up-to-date, it's ad-free, and it can even handle VAT MOSS payments on your profits for you.

Once you've created an account, go to My Dashboard and hit 'Create new project'. From there, all you have to do is enter a title, a description, a cover image, and upload the HTML file.

Remember to tick the 'This file will be played in the browser' box, and then enable the 'Mobile friendly', fullscreen, and scrollbars options further down the page.

If you have graphics or audio that you want to include, just put them in a directory with your HTML file, which has to be called **index.html**, zip the whole thing up, and upload the zip file.

You can preview the game's page as a draft version to make sure everything works properly before you hit Publish to release your masterpiece.

CODE A TEXT ADVENTURE in PYTHON

Coding a text adventure is a classic way to learn programming. Discover how to make your own adventure in Python

You'll Need

- Raspberry Pi
- Python
- Some ideas

Now that you've learned how to design and build a text adventure, it's time to learn to code your own. In this project, you'll design and code your own RPG maze game. The aim of the game will be to collect objects and escape from a house, making sure to avoid all the monsters.

This project teaches game design through the development of an RPG maze game. In this game, the player has to pick up objects within a house and get to a specific room, while avoiding monsters lurking in some of the rooms. This game will be achieved by manipulating dictionaries and lists.

Start by opening up Thonny (Programming > Thonny Python IDE) and enter the code from **rpg.py**.

This is a very basic RPG game that only has two rooms. **Figure 1** shows a map of the basic game.

Choose File > Save and save the code as **rpg.py**. Click the Run icon to test out the code. You can type `go south` to move from the hall to the kitchen, and then `go north` to go back to the hall again! Type `go west` in the hall and you'll get a friendly error message: "You can't go that way!"

Click Stop and look at this part of the code:

```
#a dictionary linking a room to other rooms
rooms = {

'Hall' : { 'south' : 'Kitchen'
                },

'Kitchen' : { 'north' : 'Hall'
                }
        }
```

Each room is a dictionary, and rooms are linked together using directions. Let's add a dining room to your map, to the east of the hall (as shown in **Figure 2**).

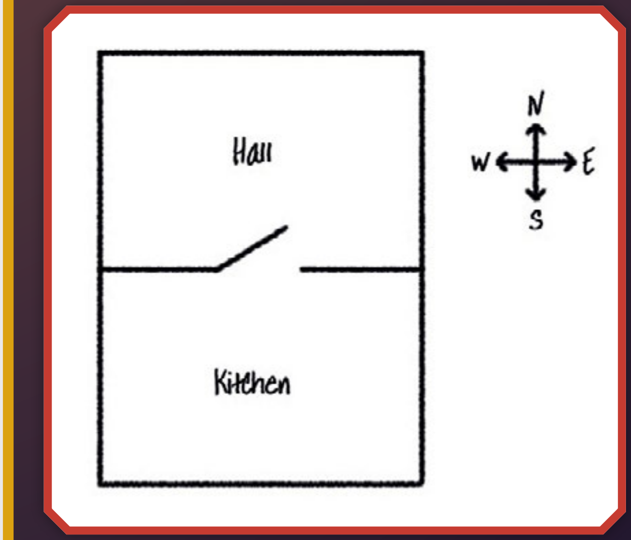


Figure 1 Map of the basic game

rpg.py

magpi.cc/2jbnPol

```
#!/bin/python3

def showInstructions():
  #print a main menu and the commands
  print('''
RPG Game
========
Commands:
  go [direction]
  get [item]
''')

def showStatus():
  #print the player's current status
  print('---------------------------')
  print('You are in the ' + currentRoom)
  #print the current inventory
  print('Inventory : ' + str(inventory))
  #print an item if there is one
  if "item" in rooms[currentRoom]:
    print('You see a ' + rooms[currentRoom]['item'])
  print("---------------------------")

#an inventory, which is initially empty
inventory = []

#a dictionary linking a room to other rooms
rooms = {
```

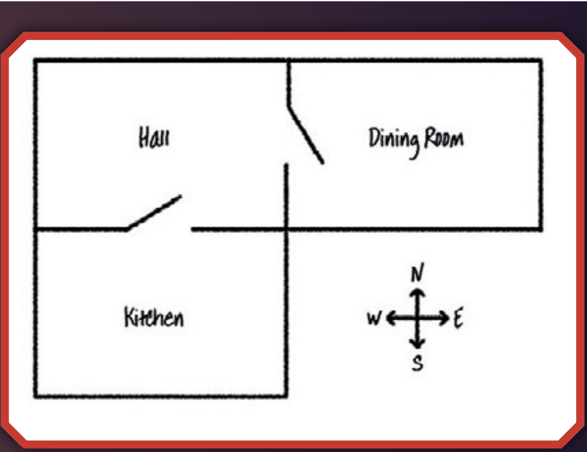


Figure 2 Now we've added a third room to the game

You need to add a third room, called Dining Room. You also need to link it to the hall to the west. In addition, you need to add data to the hall, so that you can move to the dining room to the east.

```
#a dictionary linking a room to other room
positions
rooms = {

'Hall' : { 'south' : 'Kitchen',
           'east'  : 'Dining Room',
                },

'Kitchen' : { 'north' : 'Hall',
                },

'Dining Room' : { 'west'  : 'Hall',
             }

        }
```

Try out the game with your new dining room. You can now `go east` and `go west` between the hall and dining room. If you can't move in and out of the dining room, just check that you added all of the code above (including the extra comma after the Kitchen section of the code).

ADDING ITEMS

Adding an item into a room is easy: you can just add it to a room's dictionary. Let's put a key in the hall.

```
#a dictionary linking a room to other room
positions
rooms = {

'Hall' : { 'south' : 'Kitchen',
           'east'  : 'Dining Room',
           'item'  : 'key'
                },
```

```
      'Hall' : {
         'south' : 'Kitchen'
        },

      'Kitchen' : {
         'north' : 'Hall'
        }

    }

#start the player in the Hall
currentRoom = 'Hall'

showInstructions()

#loop forever
while True:

 showStatus()

 #get the player's next 'move'
 #.split() breaks it up into an list array
 #eg typing 'go east' would give the list:
 #['go','east']
 move = ''
 while move == '':
  move = input('>')

 move = move.lower().split()

 #if they type 'go' first
 if move[0] == 'go':
  #check that they are allowed wherever they want to go
  if move[1] in rooms[currentRoom]:
   #set the current room to the new room
   currentRoom = rooms[currentRoom][move[1]]
  #there is no door (link) to the new room
  else:
    print('You can\'t go that way!')

 #if they type 'get' first
 if move[0] == 'get' :
  #if the room contains an item, and the item is the one they want to get
  if "item" in rooms[currentRoom] and move[1] in rooms[currentRoom]['item']:
   #add the item to their inventory
   inventory += [move[1]]
   #display a helpful message
   print(move[1] + ' got!')
   #delete the item from the room
   del rooms[currentRoom]['item']
  #otherwise, if the item isn't there to get
  else:
   #tell them they can't get it
   print('Can\'t get ' + move[1] + '!')
```

rpg-finished.py

magpi.cc/2jbnPol

```
#!/bin/python3

def showInstructions():
    #print a main menu and the commands
    print('''
RPG Game
========

Get to the Garden with a key and a potion
Avoid the monsters!

Commands:
  go [direction]
  get [item]
''')

def showStatus():
  #print the player's current status
  print('---------------------------')
  print('You are in the ' + currentRoom)
  #print the current inventory
  print("Inventory : " + str(inventory))
  #print an item if there is one
  if "item" in rooms[currentRoom]:
    print('You see a ' + rooms[currentRoom]['item'])
  print("---------------------------")

#an inventory, which is initially empty
inventory = []

#a dictionary linking a room to other room positions
rooms = {

            'Hall' : { 'south' : 'Kitchen',
                  'east'  : 'Dining Room',
                  'item'  : 'key'
                },

            'Kitchen' : { 'north' : 'Hall',
                  'item'  : 'monster'
                },

            'Dining Room' : { 'west'  : 'Hall',
                  'south' : 'Garden',
                  'item'  : 'potion'

                },

            'Garden' : { 'north' : 'Dining Room' }

         }

#start the player in the Hall
currentRoom = 'Hall'

showInstructions()

#loop forever
while True:
```

Remember to put a comma after the line above the new item (after 'Dining Room' here), or your program won't run.

If you run your game after adding the code above, you can now see a key in the hall, and you can even pick it up (by typing **get key**), which adds it to your inventory.

ADD ENEMIES

This game is too easy! Let's add enemies to some rooms that the player must avoid.

Adding an enemy to a room is as easy as adding any other item. Let's add a hungry monster to the kitchen:

```
#a dictionary linking a room to other room
positions
rooms = {

'Hall' : { 'south' : 'Kitchen',
           'east'  : 'Dining Room',
           'item'  : 'key'
                },

'Kitchen' : { 'north' : 'Hall',
              'item'  : 'monster'
                },

'Dining Room' : { 'west'  : 'Hall',
                  'item'  : 'potion'
                }
```

You want to make sure that the game ends if the player enters a room with a monster in. You can do this with the following code, which you should add to the end of the game:

```
  # player loses if they enter a room with a
monster
  if 'item' in rooms[currentRoom] and
'monster' in rooms[currentRoom]['item']:
    print('A monster has got you... GAME
OVER!')
    break
```

This code checks whether there is an item in the room – and if so, whether that item is a monster. Notice that this code is indented, putting it in line with the code above it. This means that the game will check for a monster every time the player moves into a new room.

Test out your code by going into the kitchen, which now contains a monster. Type **go south** and you will see 'A monster has got you... GAME OVER!'

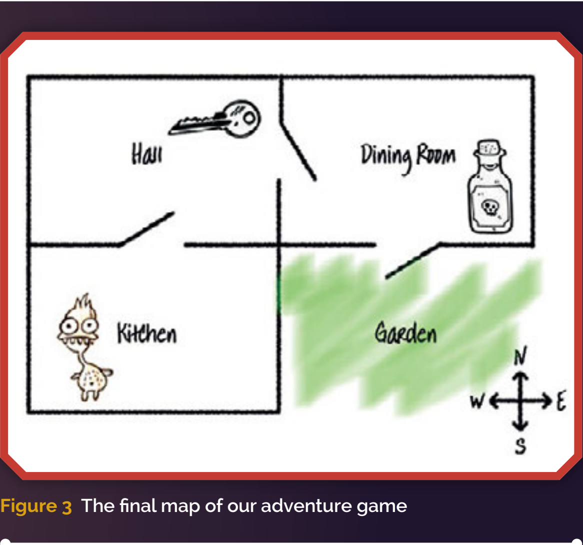


Figure 3 The final map of our adventure game

WINNING THE GAME

Let's give your player a mission which needs to be completed to win the game.

In this game, the player wins by getting to the garden and escaping the house. They also need to have the key with them, and the magic potion. **Figure 3** shows a map of the final game.

First, you need to add a garden to the south of the dining room. Remember to add doors, to link to other rooms in the house.

Add a potion to the dining room (or another room in your house). Finally, add this code to allow the player to win the game when they get to the garden with the key and the potion:

```
  # player wins if they get to the garden with a key and a shield
  if currentRoom == 'Garden' and 'key' in inventory and 'potion' in inventory:
    print('You escaped the house... YOU WIN!')
    break
```

You can see the final code in **rpg-finished.py**.

Finally, check your game to make sure the player can win! Enter the following commands:

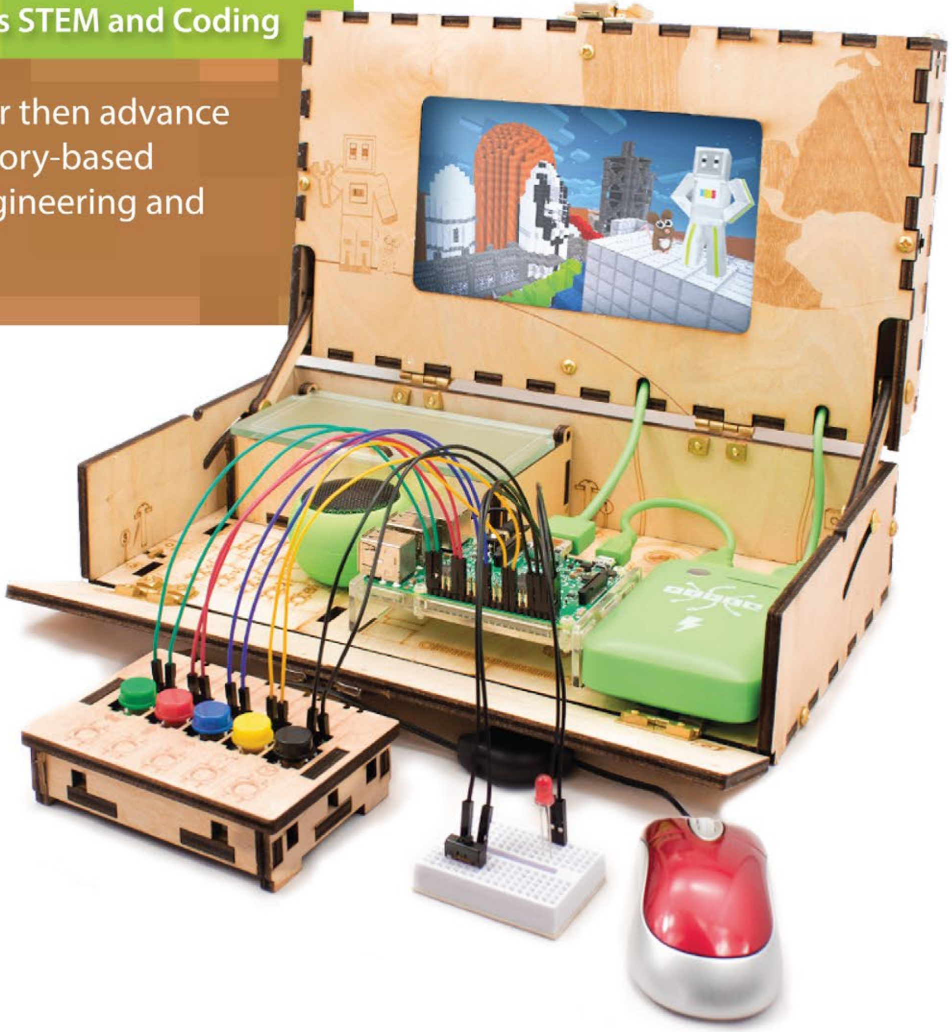
```
get key
go east
get potion
go south
```

You will see 'You escaped the house... YOU WIN!'

Use what you've learnt to create your own game. Add lots of rooms, monsters to avoid, and items to collect. Remember to modify the code so that the player wins when they get to a certain room with some of the objects in their inventory. It may help you to sketch a map before you start coding.

```

  showStatus()

  #get the player's next 'move'
  #.split() breaks it up into an list array
  #eg typing 'go east' would give the list:
  #['go','east']
  move = ''
  while move == '':
    move = input('>')

  move = move.lower().split()

  #if they type 'go' first
  if move[0] == 'go':
    #check that they are allowed wherever they want to go
    if move[1] in rooms[currentRoom]:
      #set the current room to the new room
      currentRoom = rooms[currentRoom][move[1]]
    #there is no door (link) to the new room
    else:
      print('You can\'t go that way!')

  #if they type 'get' first
  if move[0] == 'get' :
    #if the room contains an item, and the item is the one they want to get
    if 'item' in rooms[currentRoom] and move[1] in rooms[currentRoom]['item']:
      #add the item to their inventory
      inventory += [move[1]]
      #display a helpful message
      print(move[1] + ' got!')
      #delete the item from the room
      del rooms[currentRoom]['item']
    #otherwise, if the item isn't there to get
    else:
      #tell them they can't get it
      print('Can\'t get ' + move[1] + '!')

  # player loses if they enter a room with a monster
  if 'item' in rooms[currentRoom] and 'monster' in rooms[currentRoom]['item']:
    print('A monster has got you... GAME OVER!')
    break

  # player wins if they get to the garden with a key and a shield
  if currentRoom == 'Garden' and 'key' in inventory and 'potion' in inventory:
    print('You escaped the house... YOU WIN!')
    break
```

PIPER
BUILD A COMPUTER
FOLLOW MISSIONS
LEARN PROGRAMMING



Maker Says

"Level up your Raspberry Pi"

Monster Joysticks

DELUXE ARCADE CONTROLLER KIT

Rob Zwetsloot builds a mini arcade machine with this all-in-one controller kit from Monster Joysticks

In the last issue of *The MagPi* we took you through a comprehensive arcade machine build, including a complete wooden build of the cabinet itself. While it's certainly impressive, not everyone has the space, time, or money for one. This is where awesome little kits like this one from Monster Joysticks come in.

You've probably seen this type of kit before – it's an all-in-one arcade joystick and case for your Raspberry Pi that turns it into a small and portable arcade machine. Just hook it up to the nearest television and you're ready for some Elevator Action. It's like the plug-and-play mini Mega Drive you got for Christmas a few years ago, or the SNES Classic Mini you missed out on this year due to limited stock. Thanks, Nintendo.

Unlike the stocking filler plug-and-play consoles, this kit requires you to build your gaming system and supply the Raspberry Pi that powers it. Construction is very simple, though: there are six acrylic panels for each side of the box and only eight screws required to fasten them all together.

Quality components

The kit comes with nine genuine Sanwa arcade buttons and a Sanwa joystick, which just simply click into the acrylic panels as you build them.

To wire up the buttons and joystick, a little add-on board is provided with colour-coded wires. They can be a little tricky to properly attach to the connections as the connectors themselves are a bit tight, but you don't have to worry too much about wires getting tangled up. You may also need to push down the top panel a bit due to resistance of all the wires, but otherwise it all fits fairly neatly inside. You can find the full build instructions on the Monster Joysticks website: **magpi.cc/2i3iQp8**.

The build took us just shy of three episodes of *The Simpsons*, so make sure you set aside about an hour for the job. Our only real complaint about the build is that, while all the ports and even SD card slot are readily accessible, the Raspberry Pi can only be removed by taking the case apart. It will only take a couple of minutes to remove it, but we'd have preferred it to be a little easier.

Related

PICADE CONSOLE

The product that launched Pimoroni is very similar to the Monster Joysticks version, although it has a slightly nicer case.

£90 / $120

magpi.cc/2i1H82R

monsterjoysticks.com

£100 / $132

The final part of the build involves attaching little rubber feet to the bottom – very welcome, as the case had been slipping a bit on the glass table it had been built on.

The stick feels solid and has a decent weight to it thanks to the included components, so you feel pretty safe giving the buttons and joystick a proper workout. The included Sanwa components are quite important as not only are they high-quality and can survive a bit of classic Street Fighter button mashing/frame-perfect combo-timing, they're also quite customisable. For instance, if you don't fancy the button colour scheme, you can always swap them out. The joystick itself can also be customised: the version that comes with the kit has square four-way gates, but they can be upgraded to an octagonal eight-way gate, or any other gate style if you prefer.

> The stick feels solid and has a decent weight to it thanks to the included components

Quick configuration

Software customisation for RetroPie is also very simple. With a custom add-on board to connect the controls to the Pi over GPIO, we initially feared we'd have to download custom scripts for the job. Not so, though, and while you do need to go into the RetroPie configuration menu and install an extra driver (snesdev), it's all quick and included in the RetroPie archive. Once that's done, you can configure the stick controls, as well as any extra controllers you've plugged into the USB ports. Co-op Contra, anyone?

This kit is a great, solid package and it looks good as well. We recommend investing in some nice, long HDMI and USB cables to power the box and don't be afraid to put some sticks or a little custom decal onto the case as well. With Christmas coming up, it may just be the perfect gift for someone, especially if they missed out on the aforementioned SNES Mini.

Last word

A great little kit. It's a fun build but also a good quality product to use. We'd prefer the Raspberry Pi to be a bit more accessible, but otherwise the high customisability is a big plus.

★★★★☆

KITS FOR CHRISTMAS

We look at some Raspberry Pi kits that are perfect for the holiday season

3D XMAS TREE

Rob Zwetsloot starts Christmas decorating with this fun add-on for the Raspberry Pi

magpi.cc/2iZCkel

From £12 / $16

Maker Says

> The perfect stocking filler for any geek with an interest in electronics and the Pi
>
> **The Pi Hut**

If you watched our video on the Pimoroni Unicorn HAT HD (**youtu.be/b4ItGer9q1o**) a little while back, or keep an eye on our Twitter feed, you'll know that our features editor Rob was using the Unicorn HAT HD to count down to a forthcoming holiday. Now that's been and gone, it's time to replace it with something else and what better than a little festive add-on for the Christmas season?

The 3D Xmas Tree acts very much like a HAT for the Raspberry Pi, sitting neatly on top of the GPIO pins. However, it doesn't cover the top of the Pi and, due to the '3D' nature of the design, it also hangs off the edge. Still though, it's neat to just slot on.

The £12 price is for the unsoldered version which you need to assemble yourself, but there is also a £15 ($20) version that comes pre-soldered. We're always up for a bit of soldering, so were happy to get a soldering iron out when the unsoldered version arrived. However, due to the nature of the tree, it's a little bit of a chore. 25 resistors and 25 LEDs need to be individually soldered to the board along with a 40-pin GPIO header and, while it's perfectly fine and pretty easy, the repetition is noticeable. Still, we constructed the tree in about 45 minutes or so.

Programming the tree is a doddle and makes good use of the `LED` and `LEDBoard` classes in the GPIO Zero Python library on the Raspberry Pi rather than using its own dedicated software. The example code on The Pi Hut product page (**magpi.cc/2iZCkeI**) is enough to get you started and you can easily modify it to make your own little light routine with some quick tinkering.

We like this kit as it looks great once finished and it's pretty customisable as well: don't want all the main LEDs to be red? Remove them and add some multicoloured alternatives! For now, though, we're just going to have the standard red LEDs flash at our desk while we work. Maybe next year.

Last word

A great little Pi-themed decoration that can fit almost anywhere you want it to. We suggest spending a bit more on the pre-soldered version, though.

★★★★☆

STOCKING FILLERS

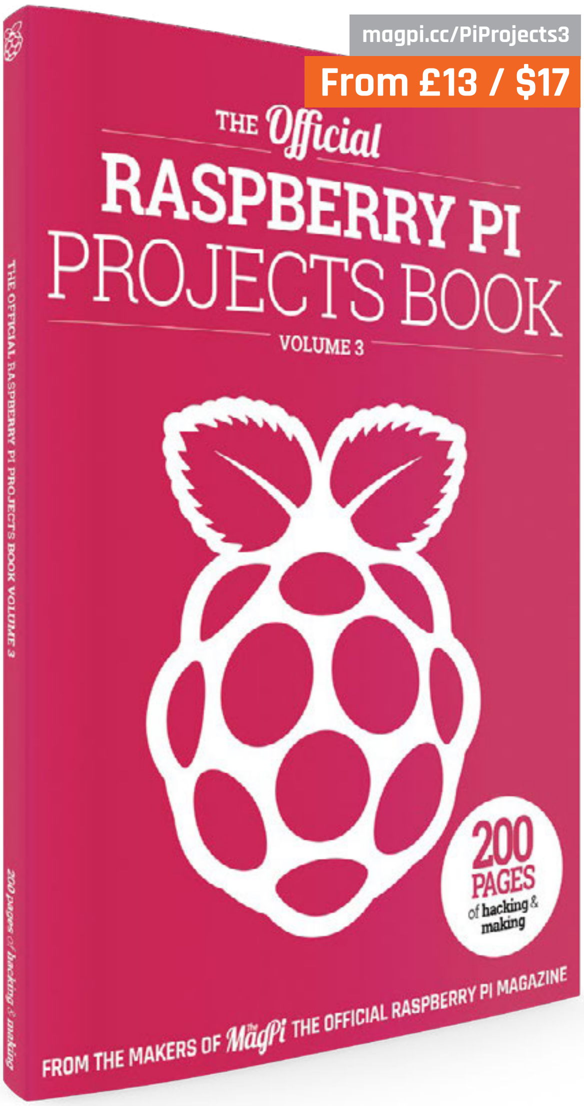
Here’s a little gift guide for the young and not-so-young makers in your family

THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 3

The latest edition of our *Projects Book* is out and comes with 200 pages packed with awesome projects, amazing guides, and very helpful product reviews. We think it’s a perfect little book for newcomers to Raspberry Pi or people wanting to learn a bit more about making. At the very least, it should keep the younger computer whizzes occupied until a more reasonable present-opening hour.

magpi.cc/PiProjects3

From £13 / $17


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 3
200 PAGES of hacking & making
FROM THE MAKERS OF MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

magpi.cc/2j1MnQv

From £6 / $8

BEARABLES

While not Raspberry Pi-powered, these adorable wearables from Pimoroni can be a first step into electronics for someone more on the crafting side of the maker spectrum. They can be easily sewn onto fabric and can even use conductive thread to attach sensors as well. The base can then be customised to suit someone’s individual style – a fun little project during the afternoon. You can also hack them if you know how…

magpi.cc/2zQA2Xt

From £35 / $45

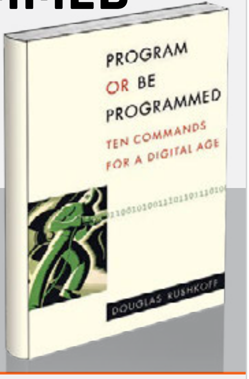
PI ZERO W STARTER KIT

A perfect little kit for someone that asked for a Raspberry Pi for Christmas. This kit includes a Raspberry Pi Zero W along with a case and a Blinkt! LED strip, so you can get started doing cool stuff with your new Pi Zero straight away! It’s also a pretty small case, so it’s easy to hide for a big surprise.

RASPBERRY PI BESTSELLERS O.R. BOOKS

Books which probe and question this internet that we've all begun to take for granted.

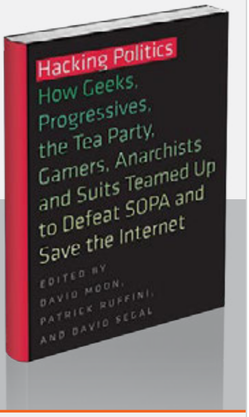
PROGRAM OR BE PROGRAMMED



Author: Douglas Rushkoff
Publisher: OR Books
Price: £10
ISBN: 978-1935928157
magpi.cc/2zqLQiS

Still essential reading, Rushkoff's 2010 call for everyone to learn programming was part of a wave that led to Code Club and the Pi, and the arguments are well worth revisiting.

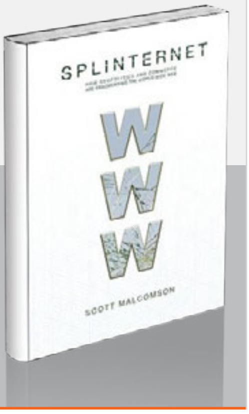
HACKING POLITICS



Authors: David Moon, Patrick Ruffini, David Segal
Publisher: OR Books
Price: £20
ISBN: 978-1939293046
magpi.cc/2zr6q2y

The extraordinary story of the 2012 internet blackout, and the successful campaign to stop SOPA, which would have removed numerous domains from the internet, told in the collective voices of those involved.

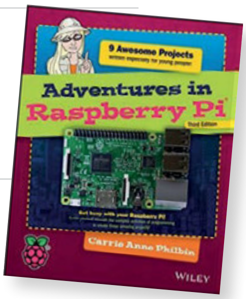
SPLINTERNET



Author: Scott Malcomson
Publisher: OR Books
Price: £13
ISBN: 978-1682190302
magpi.cc/2zqn3v5

"A contested, protean terrain, constantly evolving as different forces intervene to drive it forward." The internet's (military) history, and fragmented present – driven by politics, money, and conflicting idealism (hacktivists versus netizens).

ADVENTURES IN RASPBERRY PI – 3RD EDITION



Author: Carrie Anne Philbin
Publisher: Wiley
Price: £20.99
ISBN: 978-1119269069
magpi.cc/2zt24rO

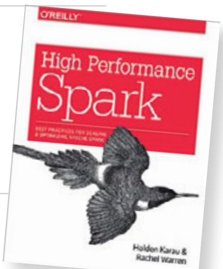
New to Raspberry Pi? Want your children to learn coding and have fun connecting up lights, sensors, or even marshmallows? Or even use the Pi to make a dedicated device? You've come the right place. Philbin is a technology educator with plenty of experience of bringing alive the Pi's possibilities for a young audience.

After setting up the Pi, and learning some command-line basics, readers get to try using the Scratch programming language to animate a monkey sprite, then start building a role-playing game, before diving into Turtle graphics to make shapes with both Scratch and Python. Sonic Pi continues the coding – making music with Ruby code. Throughout these divergent adventures, additional notes, definitions, and tips and tricks combine with the lessons to more broadly educate new programmers.

The last chapters are on hardware: using the GPIO with those sensors and marshmallows, and building a Pi jukebox. Readers who already have the second edition (see *The MagPi* #32) will want to know what's new – a whole extra chapter on experimenting with cameras, and expanding your Pi's capabilities with add-on HATs – but won't need any persuasion to buy extra copies as Christmas presents for young relatives!

Score ★★★★★

HIGH PERFORMANCE SPARK



Authors: Holden Karau & Rachel Warren
Publisher: O'Reilly
Price: £31.99
ISBN: 978-1491943205
magpi.cc/2yg93CX

Apache Spark is a great tool for querying large data sets, but doesn't it need big hardware to run on? In fact, in *The MagPi* #38, Sung-Taek Kim extolled the virtues of a small Raspberry Pi cluster for learning and experimenting with Spark, in an article that highlighted a typical data science stack of Python-based NumPy, SciPy, and Scikit-Learn, as well as Java.

Sung-Taek also mentioned the importance of learning a little Scala, and it is that language that Karau and Warren propose as the best way to work with Spark – although Python is not entirely forgotten here. Language choice justified, the authors give a pithy introduction to the how and why of Spark, and where it fits in the 'Big Data Ecosystem'.

This is a practical book, and the middle section takes the reader through data operations from loading, through optimised queries in Spark SQL, to join types, and transformations. After a chapter on working with key/value data, with a worked example of 'Goldilocks data' showing different approaches and pitfalls, coverage extends to further languages (Julia, R, et al.), testing, machine-learning libraries, and the broader Spark ecosystem. An insightful Spark introduction.

Score ★★★★☆

ALGORITHMS AND NETWORKING FOR COMPUTER GAMES

Authors: Jouni Smed & Harri Hakonen
Publisher: Wiley
Price: £60
ISBN: 978-1119259763
magpi.cc/2yjcB7h

There's a big leap from reimplementing a few classic arcade games in Pygame to designing and programming complex modern games. It's certainly not something that would fit into one book, and Smed and Hakonen are concerned here only with the computer science side of game programming – in particular practical algorithms, and solving network-related problems.

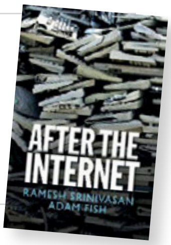
More than 70 algorithms are presented, covering random numbers, noise in data (a realistic world is full of imperfections), procedural generation, tournaments, game trees, path finding, group movement, decision making, and modelling uncertainty – as well as networking problems, including dealing with cheating.

The exercises at the end of each chapter range from simple thought exercises to studying Braben and Bell's name-generation algorithm from Elite (1984).

The gaming landscape has changed massively since the first edition was published in 2006, in particular with the move to online and social gaming, but most of the underlying problems remain the same – so while exercises and examples change, the algorithms remain. You'll need to be comfortable with algorithms and data structures to tackle the book (see *The MagPi* #45 and #62 for our recommendations), but use of pseudocode throughout ensures the book works equally well for C, C++, Java, Python, or even C# programmers.

Score ★★★★☆

AFTER THE INTERNET



Author: Ramesh Srinivasan & Adam Fish
Publisher: Polity
Price: £14.99
ISBN: 978-1509506170
magpi.cc/2ygyvrM

What is the internet? A means of connecting people around the world, empowering them to better things, and spreading enlightenment? A way of monitoring citizens' every activity? An echo chamber of sterile non-debate, and opinions repeated unquestioned? A means of monetising every click you make, as the so-called sharing economy concentrates power and wealth in the hands of a few platforms?

Srinivasan and Fish tackle head on the gloomier prognostications of other commentators and sociologists, whilst tackling some of the utopic myths of the internet. The language is academic – and that of the sociologist, not the CS department – so if talk of post-structuralists, semantics, and semiotics is not your cup of tea, you may take a few pages to settle in. It's worth persisting, though – despite clearing the grounds of myths, the authors discover many signs of hope in the contested space of what makes up the internet.

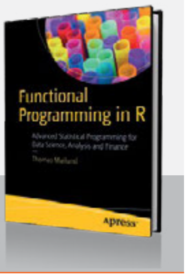
Ignoring both the techno-utopians, and the doom-mongers, gives a clearer view. From grassroots liberation struggles to a platform for whistleblowers, *After the Internet* chronicles successes and failures in struggles for justice and equality, and points towards the internet as an assemblage of transformative movements.

Score ★★★★☆

ESSENTIAL READING: BEYOND FUNCTIONAL

There's more to functional programming than recursive functions, immutable data, and homoiconicity.

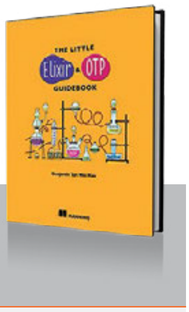
Functional Programming in R



Author: Thomas Mailund
Publisher: Apress
Price: £14.99
ISBN: 978-1484227459
magpi.cc/2yhPN8f

Much more than just a tool for statisticians, R is surprisingly well suited to function composition and immutable data.

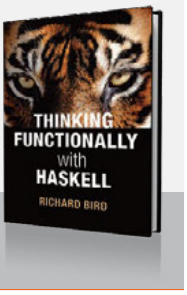
The Little Elixir & OTP Guidebook



Author: Benjamin Tan Wei Hao
Publisher: Manning
Price: £21.99
ISBN: 978-1633430112
magpi.cc/2ygXVpv

Fault tolerance, supervision, and distribution – what more can we say? How about property-based and concurrency testing?

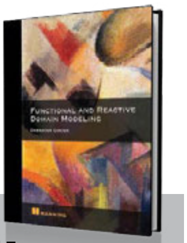
Thinking Functionally with Haskell



Author: Richard Bird
Publisher: Cambridge University Press
Price: £35.99
ISBN: 978-1107452640
magpi.cc/2ygY3Fv

A strongly mathematical way of looking at functional programming – opinionated but excellent stuff.

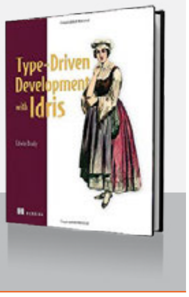
Functional and Reactive Domain Modeling



Author: Debasish Ghosh
Publisher: Manning
Price: £42
ISBN: 978-1617292248
magpi.cc/2ygY240

Domain-driven design with real-world functional programming – Scala and Akka – and the resilience and elasticity of reactive principles. Insightful.

Type-Driven Development with Idris



Author: Edwin Brady
Publisher: Manning
Price: £35
ISBN: 978-1617293023
magpi.cc/2ygPydo

Safer code without depending on unit tests? This practical book will get you to truly appreciate type-driven techniques.

Dominique Laloux

Occupation: Teacher

BRINGING COMPUTING TO TOGO WITH PI

Dominique Laloux has spent five years helping set up educational computer labs in the African nation of Togo. Here's how he's done it and how you can help

One of the key factors of the Raspberry Pi is its low price. This enables less affluent people to get in on the new computing revolution, whether in the UK or in poorer nations like Togo. Brussels computer science teacher Dominique Laloux has been working with and helping the Togolese to bring computing education to schools and has made good use of the Pi in this effort.

Can you tell us about the educational work you do in Togo?

Since 2012, some Togolese friends and I have set up three computer labs in rural areas in Togo, West Africa: two in small villages located in the hilly region of Kuma, the third one in the nearby town of Kpalimé.The first one is set up in the village of Kuma Tokpli, for the benefit of students and teachers of middle and high schools in surrounding villages. The two others are set up in secondary schools. Our first lab was a rather 'conventional' computer lab, set up with 'retired' laptop computers. The two others are Raspberry Pi computer labs: each of their 20+ workstations is a Pi. The original lab will soon be updated to include Raspberry Pis.

Our efforts now tend towards two main objectives: to set up one new Raspberry Pi in one middle/high school every year, and to promote the use of the Raspberry Pi as an alternative to more 'conventional' computers in schools and other educational communities in Togo and anywhere resources are scarce.

How did the idea come about?

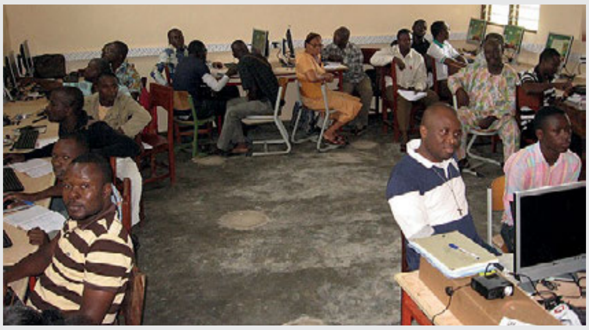
In 2011, while I was visiting Togo in preparation for a project (completely unrelated to ICT), I met a small group of colleagues in the small village of Kuma Tokpli.

During a long, informal conversation, one experienced French and history teacher told me: "I feel like a 21st century illiterate because I cannot use a computer." That statement impressed me very much, and I decided to challenge him by offering to send a few computers along, on the condition that he promised me to learn how to use them.

He probably thought I was not serious about that offer, but a few months later I had collected, refurbished, tested, and packed enough equipment to set up a fully functional computer lab with 25 computers, a network printer, a projector, a whiteboard, etc.

I went back to Kuma a couple of times and, by October 2012 we were ready to install a small local construction with proper electrical wiring, a local area network, and the 25 workstations. We started

Below Teacher training is necessary to make sure students can learn properly



CREATING A PI ROOM

Here are Dominique's tips on setting up a classroom for computing education

- Proper size room (7 × 11 metres typical classroom): must be clean, dry, ventilated, well protected against insects/rodents and, of course, protected against intrusions.
- Electrical wiring: recent and well protected, with differentials, circuit breakers, cable ducting along the walls, no cables on the floor around workstations.
- Local area network, including a switch (100 Mb + 1 Gigabit port).
- Ideally 20 or more Raspberry workstations, plus one for the teacher.
- A local server, which can simply be a faster laptop, working as a file server, a web server, or Kiwix.
- Accessories: network printer, scanners, video projector, etc.
- Whiteboard rather than chalk (to avoid dust) – a flip chart can be sufficient.
- ICT-related posters to illustrate important concepts/decorate.
- A very broad collection of applications including LibreOffice, Firefox, Chromium, Gimp, Audacity, GanttProject, FreePlane, GnuCash, some music composition software, scanning tools, games, utility programs, etc.

training teachers the week after. No school, no village, no community in the entire region had ever had such a ICT facility at their disposal.

2012 is also the year the first Raspberry Pi was introduced and I was lucky to get one of the very first units available in Belgium. I immediately made the link with our small action in Togo.

In 2013, I was offered a full year sabbatical by my employer, in order to be able to focus exclusively on what had become a not-for-profit project – the 'Centre Informatique de Kuma' (Kuma Computer Centre).

During that year, it became obvious that it would be best to have a second computer lab, located in the nearby village of Kuma Adamé, in order to make it possible for students to benefit from an introduction to ICT while avoiding the dangers of walking the path between the two villages

In summer 2014, we opened our second lab in Kuma Adamé – the first Raspberry Pi computer lab in Togo – for the benefit of about 200 young students and their seven teachers.

Why the Raspberry Pi?

Three aspects of the Raspberry Pi make it a very serious alternative to more 'conventional' computers in developing countries or in any place where resources for education are scarce (including in my country!): low cost, low power requirements, low maintenance.

One must consider an often forgotten aspect of computer donations to developing countries: what is initially a kind gesture can easily turn into a 'poisoned gift'. Older, 'retired' computers are inevitably likely to break down after a while. In a developing country where resources are scarce, it can then be difficult and expensive to repair them (spare parts are difficult to find and often expensive; technical skills are not always available…). Such machines are also often relatively more 'power hungry'.

Above Work has to go into making sure the classrooms are ready to be used

What can go wrong with the Raspberry Pi setup? A computer screen going bad? An input device that stops working? A damaged SD card? None of these is really an expensive problem. We've got very nice second-hand 17-inch computer screens for about €5 a piece; nearly new keyboards and mice (discarded by companies) for less than €4 each.

The most expensive failure? One of the computers that fails and must be replaced. That would cost exactly €35. In fact, we have a small reserve of several Raspberry Pi as a backup: in nearly four years, we've never had to use one!

How have students reacted to the classes?

They simply love it! Most of them quickly take ownership of their ICT education. And some are eager to move much further than the simple use of ICT, dreaming about getting engaged in maintenance, programming, and other IT-related higher education. And, in the small groups I have worked with since 2012 in Kuma Tokpli, some of them actually have reached a level where they can consider such a path for their higher education.

HOW YOU CAN HELP

Want to aid Dominique and his friends in their efforts? Head to **initic.be** to find out how you can support them with equipment, money, or your own personal time and labour.

initic INItiation aux TIC

THE MONTH IN RASPBERRY PI

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

YOUR ARCADES

Last month we told you how to build an arcade machine. Here are some you've already created

It the end of issue 63's arcade build feature, we showed off some of the alternative arcades you could try. Unfortunately, we didn't have space to show all of them, so here are some of the best arcade machines from around the community.

CIRCUITBEARD circuitbeard.co.uk

Not content with making one arcade machine, Circuitbeard (Matt Brailsford) has made multiple amazing custom Raspberry Pi arcades. Each of them has had a ridiculous level of work put into it, so we're going to show off three of his incredible creations here.

Rombus-CT magpi.cc/2zWRgSX

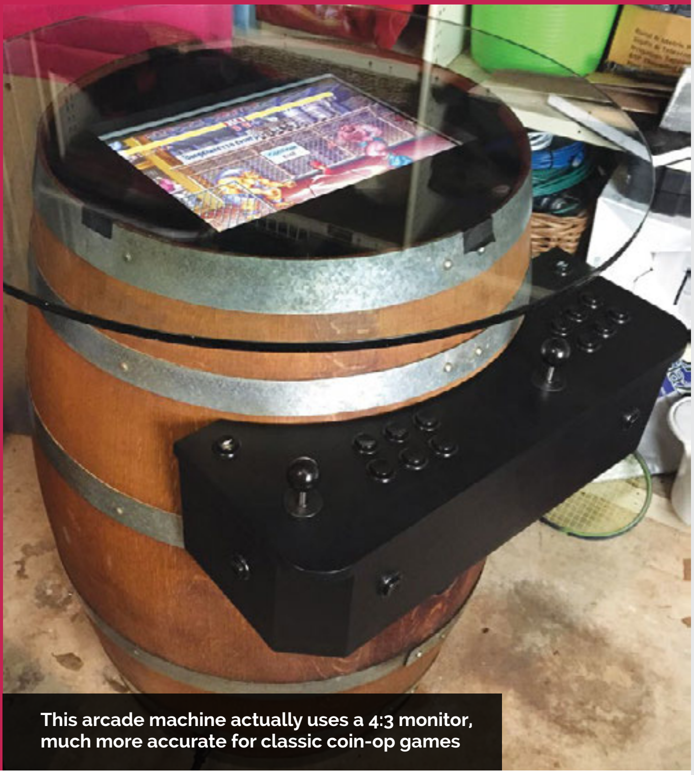
The Rombus-CT is a hand-built two-player cocktail arcade cabinet

Rombus3000 magpi.cc/2zYPh0y

An old Grandstand Scramble desktop arcade machine was repurposed to make this mini Pi arcade

Playing games on it is easy, and Circuitbeard even connected a keyboard to it

Wine Barrel Arcade magpi.cc/2ja1vLF



This arcade machine actually uses a 4:3 monitor, much more accurate for classic coin-op games

MATT SHAW

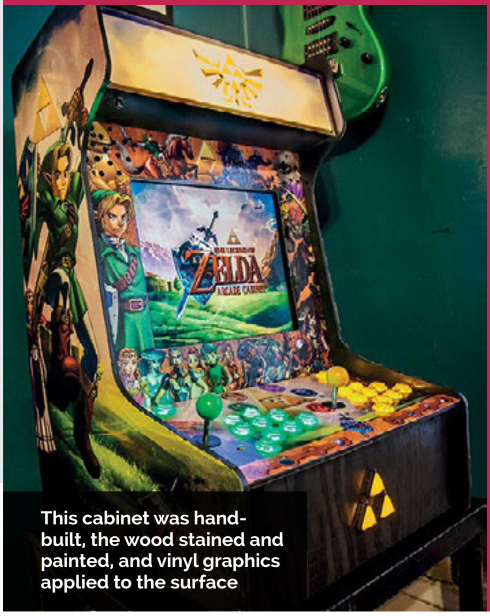
Donkey Kong is one of the all-time classic arcade games, introducing video game superstar Mario to the world, as well as the titular Donkey Kong himself. This excellent build takes an old wine barrel that looks like the kind the big gorilla chucks at you, and Matt has installed a control system and screen so it can be used a bit like a cocktail arcade table.

PHRAZELLE

Arcade cabinets are not just for arcade games! This custom Legend of Zelda build themed around Ocarina of Time is a beautiful thing, its style reminding us more of pinball table than the kind of arcade machines you normally see. Of course, you can still play arcade games on here, but you can also play Link to the Past.

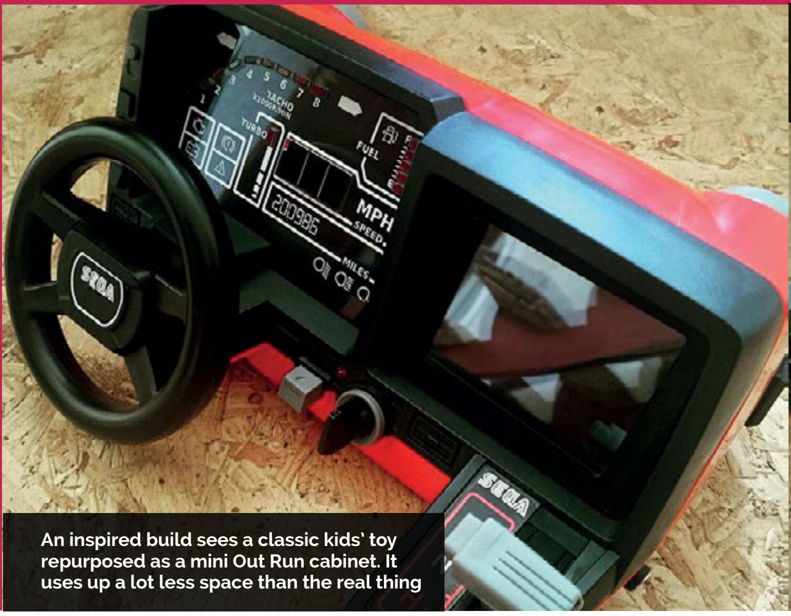
Legend of Zelda Bartop Arcade Cabinet

magpi.cc/1qOD3lx



This cabinet was hand-built, the wood stained and painted, and vinyl graphics applied to the surface

Tomy Turnin' Turbo Dashboard Out Run Arcade magpi.cc/2xckwUo



An inspired build sees a classic kids' toy repurposed as a mini Out Run cabinet. It uses up a lot less space than the real thing

With new decals, it's properly branded to the arcade classic. For more details on the build, see the project showcase in this issue

REGISTERED FOR THE BIG BIRTHDAY WEEKEND?

HERE'S WHAT'S NEXT

The Raspberry Pi Foundation wants to put on a global celebration for its sixth birthday. Here's how you can help

For the past few years, the Raspberry Pi Foundation has held a community event in Cambridge around Raspberry Pi's birthday, where people have come together for a huge party with talks, workshops, and more. Now they want more people to have the chance to join in with the birthday celebrations next year, so they're going to be coordinating Raspberry Jams all over the world to take place over the Raspberry Jam Big Birthday Weekend, 3–4 March 2018. Already registered? Here's what you need to do next.

What do you need to do?

At this point, you should be looking for a venue, and aiming to book the date as soon as possible. Don't leave it too late, or you might have difficulty securing a location. If you need some advice on finding a venue, refer to the appropriate section in the Guidebook (**magpi.cc/2q9DHfQ**).

Once you've done that, you should create a sign-up page for your event (on Eventbrite, for example). Then you can submit your event to the Jam map (**magpi.cc/28Nxeff**), and start to publicise it locally.

We're planning to produce some limited-edition 'tour' T-shirts for the Big Birthday Weekend and we'd like to include as many Jams as possible on the back print. To make sure your event gets included, submit your Jam to the map as soon as possible.

Stay connected

If you're not already a member, you might find it useful to join the Raspberry Jam community on Slack (**slack.com**), where Jam organisers share ideas and help each other. Just email **jam@raspberrypi.org** to request an invitation to join.

NEED HELP STARTING A JAM?

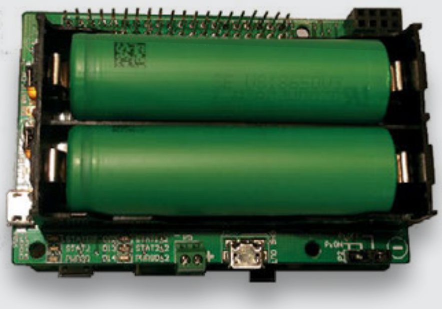
First of all, check out the Raspberry Jam page to read all about Jams, and take a look at our recent blog post explaining the support that we offer: **rpf.io/jam.**

If there's no Jam near you yet, the Raspberry Jam Big Birthday Weekend is the perfect opportunity to start one yourself! If you'd like some help getting your Jam off the ground, we've produced a free Raspberry Jam Guidebook full of advice gathered from the amazing people who run Jams in the UK. Download it from **magpi.cc/2q9DHfQ.**

If you have any further queries, email: **jam@raspberrypi.org.**

KICKSTART THIS!

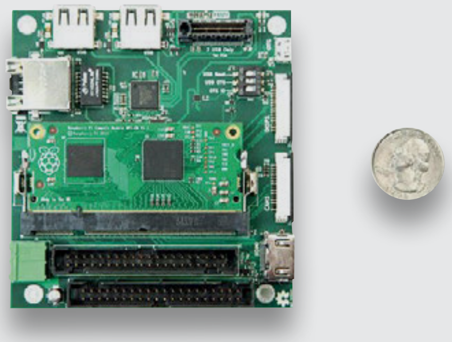
Raspberry Pi projects you can crowdfund this month



DUAL BATTERY HAT

kck.st/2hQ0oz2

Portably power your Raspberry Pi with powerful 18650 lithium-ion batteries thanks to this very special HAT. 18650 batteries are a fair bit larger than AA batteries, but can contain a lot more charge, which is great for when you need your Pi to stay on for longer. It's already hit its goal, so now is a good time to jump on it.



PI/104

magpi.cc/2zYp2Hp

Something a bit more industrial: the Pi/104 is a Raspberry Pi Compute Module dock that allows the CM to conform to the PC/104 standard. The Raspberry Pi is popular in industrial automation setups and the creator project Adam Parker believes the price allows for prosumers to make use of it as well. Give it a look as there's not many products like this around.

BEST OF THE REST

Here are some other great things we saw this month

PI VENDING MACHINE

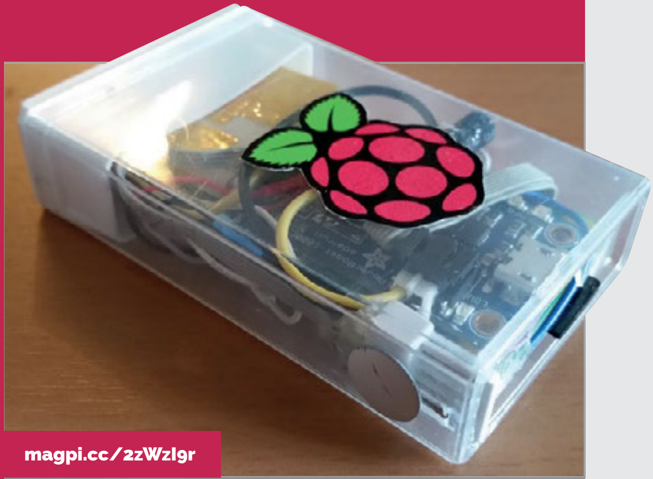
magpi.cc/2zVN6us

We've come a long way since you could get Mars bars and a packet of crisps from a vending machine. We like the idea of heading to the lobby of a makerspace and just grabbing the components you need quickly from one of these. Need a switch? No problem, that's G5.

TOUCHSCREEN BOOMBOX

magpi.cc/2zXDdwt

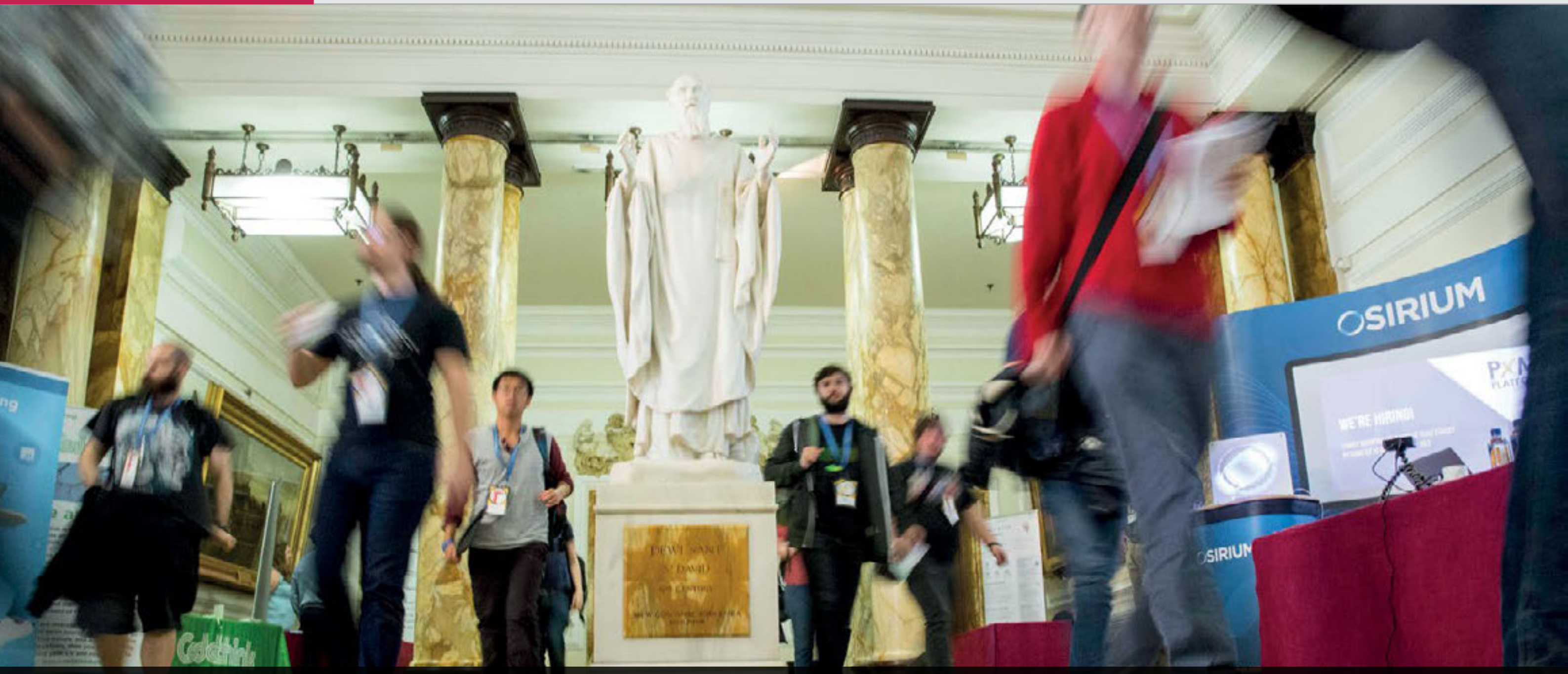
Retro upcycling with a Raspberry Pi is good fun, and we love this proper Eighties stereo being upgraded with a Raspberry Pi. Hopefully it's portable so you can sling it over your shoulder and walk down the street with it.



magpi.cc/2zWzl9r

PI-TAC 2.0

Putting the Pi Zero in small cases is a sub-genre of Pi projects in itself. This one is very cool, squeezing a Pi Zero into a Tic Tac mints box and also including a little display readout so you can get some basic information of how the Pi is running.



PYCON UK 2017

Coders of all levels come together to celebrate Python. **Sean McManus** reports

EVENT NAME:
PyCon UK 2017
DATE:
26–30 October
LOCATION:
Cardiff City Hall

In its tenth anniversary year, PyCon UK took place at Cardiff City Hall in October, bringing together professional Python coders, hobbyists, children, and educators.

The programme was varied and packed with ideas, with up to nine streams running at the same time. Among other things, there was an introduction to object-oriented programming, talks on artificial intelligence, and tips on getting the best performance from Python. BBC micro:bit sessions showed how to code the device in MicroPython, and offered advice for those teaching with it.

Mikey Lear and Luke Spademan test a micro:bit-controlled robot

Photos: Mark Hawkins, composedimages.co.uk

On Thursday, 24 educators attended Picademy, the programme from the Raspberry Pi Foundation that supports them in digital making. Professional coders and hobbyists from the main conference dropped in to help with projects including a man overboard alarm, a spinning pumpkin decorator, and a motion-activated ghoul.

Django Girls ran a one-day workshop to introduce women to programming using the Django web framework. Student Shuping Li took part and said: "The best thing was that it made me feel I was able to use Python despite having zero experience. I went from nothing to making a blog in a day."

Wheely good

Ben Nuttall, Raspberry Pi Community Manager, revealed how the process for installing Python packages has been accelerated on the Raspberry Pi. 'Wheels' are Python packages that have been compiled for a particular architecture. They're smaller files to download, and there's no time spent compiling after you download them. The Python Package Index (PyPI) doesn't support ARM packages, though, so Ben built his own repository, called piwheels. It involved compiling more than 96,000 packages, and a total of over 670,000 version releases, using cloud-based Pis. If you're using Raspbian Stretch, piwheels is already configured as an additional index. For installing a package like NumPy, it could save you over two hours of compilation time on a Raspberry Pi 1.

THE JOHN PINNER AWARDS

This PyCon introduced the John Pinner Awards, named in tribute to the late PyCon UK founder. Ten medals were awarded to recognise those who contribute to the Python community. The first recipients were Cat Lamin, Daniele Procida, Éléonore Mayola, Joshua Lowe, Leona So, Mary Mooney, Sheila Pinner, Tim Golden, Tony Ibbs, and Vincent Knight. Congratulations to them all!

The first winners of the John Pinner Award collect their medals

A jewel in the crown

Saturday was children's day, described by Daniele Procida, a member of the organising committee, as "one of the jewels in the crown of PyCon". There was a Code Club, which included a session led by Joshua Lowe. At 13 years old, he is the creator of EduBlocks, a drag-and-drop interface for programming Python that makes it easier to transition from Scratch. At the same time as the Code Club, a Raspberry Jam took place next door. It included an introduction to physical computing with the Raspberry Pi, Minecraft hacking, and science experiments using the Sense HAT.

> The best thing was that it made me feel I was able to use Python despite having zero experience

At the end of the day, the children presented their projects to the conference in a session held in the main assembly room. Pi projects included an LED caterpillar, a game using the Python turtle module, and a burglar alarm. The micro:bit was used to make a kindness badge, a calculator, and a Morse code transmitter. The children shared some of their coding experiences from the day, too. One said: "We had to type out things. Type in things. Type out things. Then it worked!" Many of the professional coders in the room recognised that pattern.

On Sunday afternoon, some attendees came together to see how Python can be used to help identify people at risk of developing Alzheimer's disease. The conference ended with a day of programming sprints, including Trans*Code, a hackday dedicated to tackling issues faced by transgender people.

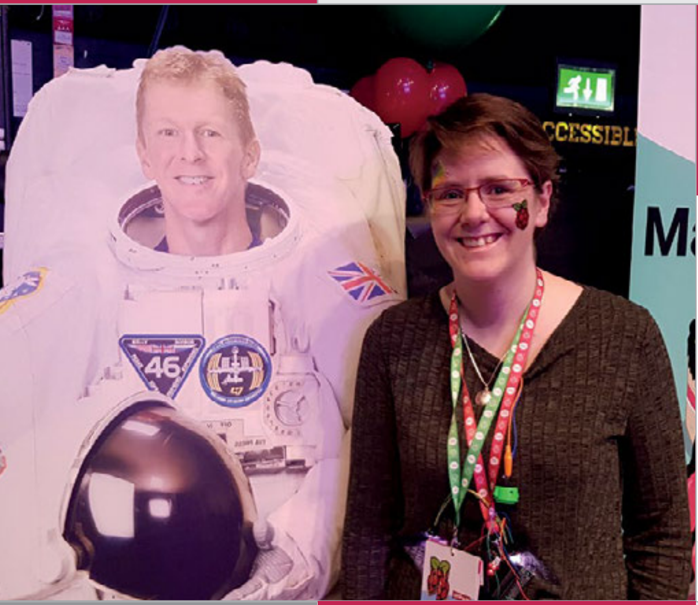
If you missed out this year, you can catch up online. The schedule now includes links to presentations and videos, where available. Find them at **2017.pyconuk.org/schedule**.

Right Joshua Lowe kicks off Code Club by introducing EduBlocks

Bottom Right Fiona Murray and Miriam Winkels make a pumpkin decorator in Picademy

Below With over 700 attendees, the assembly hall was often busy

Photo: Sean McManus, www.sean.co.uk



COMMUNITY PROFILE

LORRAINE UNDERWOOD

Introducing Raspberry Pi Certified Educator, digital maker and all-round tech nerd, **Lorraine Underwood**

Lorraine

Category: Educator

Day job: Computer Teacher and Maker

Website:
lorrainbow.wordpress.com
twitter.com/LMcUnderwood

You just have to glance at educator Lorraine Underwood's Twitter feed to realise she's an avid digital maker. From her popular illuminated temperature staircase to her Nora the Explorer car, she's forever making and sharing her creations. "I find Twitter a really positive and encouraging place to be. I live there!" she admits, going on to share that she tries to use her confidence with tech such as the Raspberry Pi to encourage others who seem a little lost in their first steps into the community of makers.

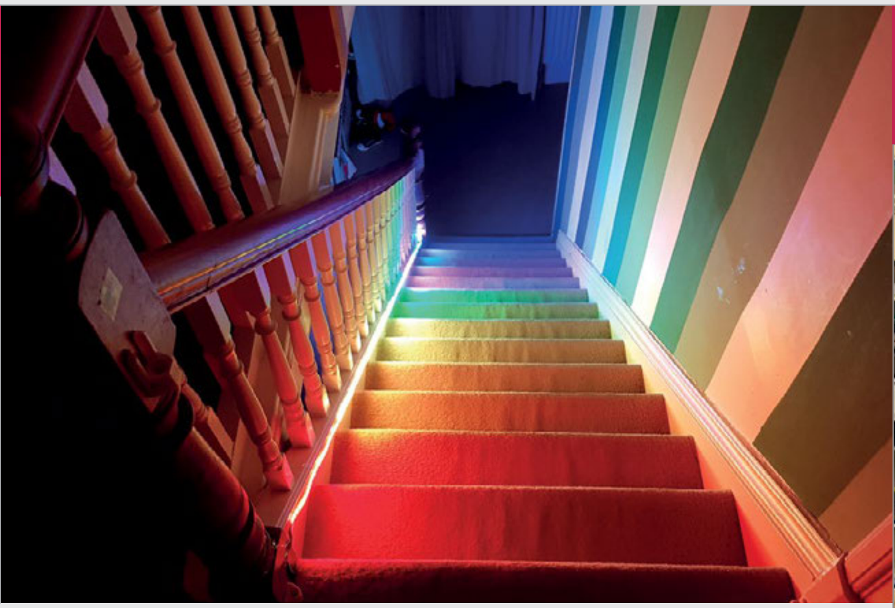
Above "I saw someone had made a virtual-reality Arduino car connected by radio to a headset… but it was with three Arduinos and cost over $300. I thought to myself, 'I could make that with a Pi for so much cheaper' and I did!"

Left Lorraine operates on a Babbage, proving that no bear is safe, even after the final bell of Picademy training

Getting started

With a degree in Computer Science, you'd think Lorraine had all the know-how and confidence to be a digital maker from the start. But as she confesses, her education was 100% computer based, with no physical making aspects involved. It was only when she started working for Lancashire University that she began to dabble in the world of digital making, and then the bug bit her. At the university, Lorraine works as part of a team to help encourage and support primary and secondary teachers in bringing the new computing curriculum to their classroom. And with digital making projects, she's found a hit for enticing dubious educators into the fold. "Teachers are very interested in physical computing and digital making," she says when discussing the conferences and workshops her department hold for teachers based in the North West of the UK. "To run workshops that teachers were interested in, I had to look at what was out there in terms of physical computing and digital making."

Outside of the university, Lorraine continues to work closely within the field of education. She



teaches Computing at a girls' grammar school, runs a small Raspberry Jam in Cumbria, and shares a passion for introducing computing and digital making to girls through Code Club and various community events.

Full of ideas

At home, she's forever tinkering. "I just love to make things! I make a lot of things; some may call me prolific! I just get a cool idea in my head and go with it."

Above "A few people have tweeted their version of my light-up temperature-controlled stairs and that just blows my mind. I love that I've inspired people to try and make things"

Alongside the rainbow stairs, Lorraine is also proud of her virtual-reality car, a project that she went on to showcase at Newcastle Maker Faire. Controlled by a Nintendo Wii Remote, and connected by radio to a VR headset, it stands as a testament to her love of tinkering and working on ideas

> "I just love to make things! I make a lot of things; some may call me prolific!"

Lorraine will be well known by many members of the Raspberry Pi community for making her Pi-powered, temperature-controlled stair lights – featured in *The MagPi* issue #58. With NeoPixels running along the bottom of her bannister, they change their colour from red to blue (with a rainbow of colours in between) depending on the outdoor temperature. The build was documented on her website and she's found the response to be incredible, especially when people share their own versions of the project.

from pre-existing projects to add her own twist, this time replacing three Arduinos with one Raspberry Pi.

In her own words, Lorraine explains of her tutorials, builds, and blog: "I try to keep everything cheap and accessible for your normal person – I try not to solder things or do complicated coding. I think a lot of projects are amazing but are really complicated and out of reach for your average person. I try to make my builds simple and easy to follow."

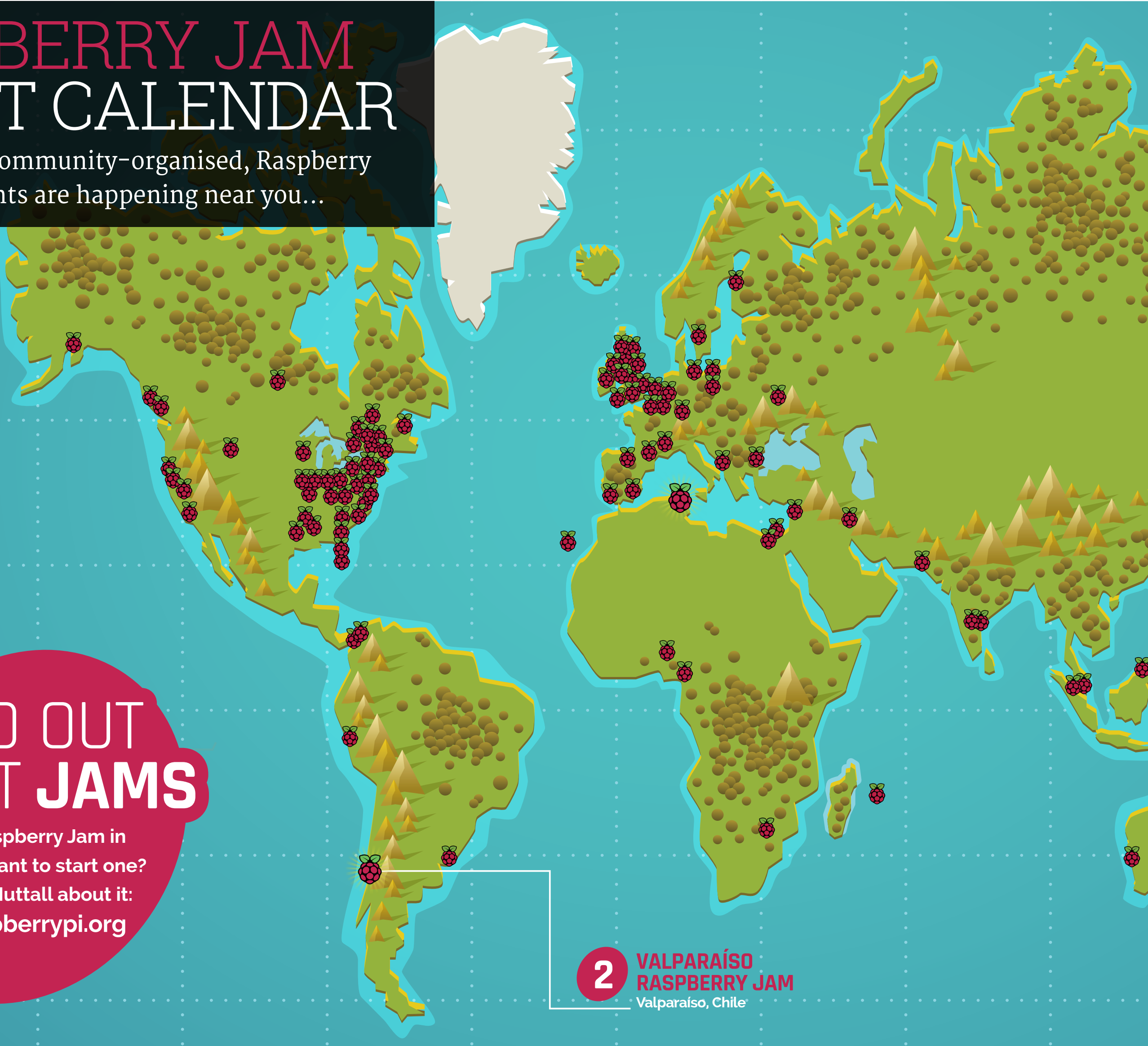
HIGHLIGHT

COMPUTING IN EDUCATION

Lorraine spreads her experience and excitement across many sectors of the educational scope. As Computing at School coordinator for Lancashire University, she supports both primary and secondary educators in delivering the new computing curriculum to the classroom. She runs training, conferences, and events throughout the North West, using her interest in digital making to support and enthuse teachers, enabling them to take back interesting projects to their students. She also teaches at Lancaster Girls' Grammar School, ensuring that she practises what she preaches. During Thursday lunch, she runs a Code Club for students aged 11–14, and also dedicates some of her time to running Kendal Raspberry Jam once a month in Cumbria. "We're a small but growing Jam of mostly children and their parents," she explains. "Our jammers like to play with stop-motion animation, Minecraft, and Scratch."

RASPBERRY JAM EVENT CALENDAR

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



FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:
ben@raspberrypi.org

HIGHLIGHTED EVENTS

1 BOGNOR REGIS RASPBERRY JAM

When: Saturday 9 December
Where: University of Chichester, Bognor Regis, UK
magpi.cc/2iWjjcU

This first Bognor Jam is primarily a show-and-tell event where people bring projects for visitors to try out.

2 VALPARAÍSO RASPBERRY JAM

When: Thursday 14 December
Where: General Cruz #222, Valparaíso, Chile
magpi.cc/2iXM6Oi

A Jam in a Chilean FabLab. There's no schedule of events yet, but keep an eye on the website for more.

3 NUNEATON RASPBERRY JAM

When: Saturday 16 December
Where: Nuneaton Library, Nuneaton, UK
magpi.cc/2iWXWsh

Mod Minecraft Pi with some Christmas-themed Python coding. Make your Minecraft world truly festive!

4 COFFEE, CAKE AND CODING

When: Thursday 7 December
Where: King Edward VI Sheldon Heath Academy, Sheldon, UK
magpi.cc/2zMcJh6

Learn good coding practice in a relaxed and informal setting while enjoying coffee and cake.

REGULAR EVENTS

5 CORNWALL TECH JAM

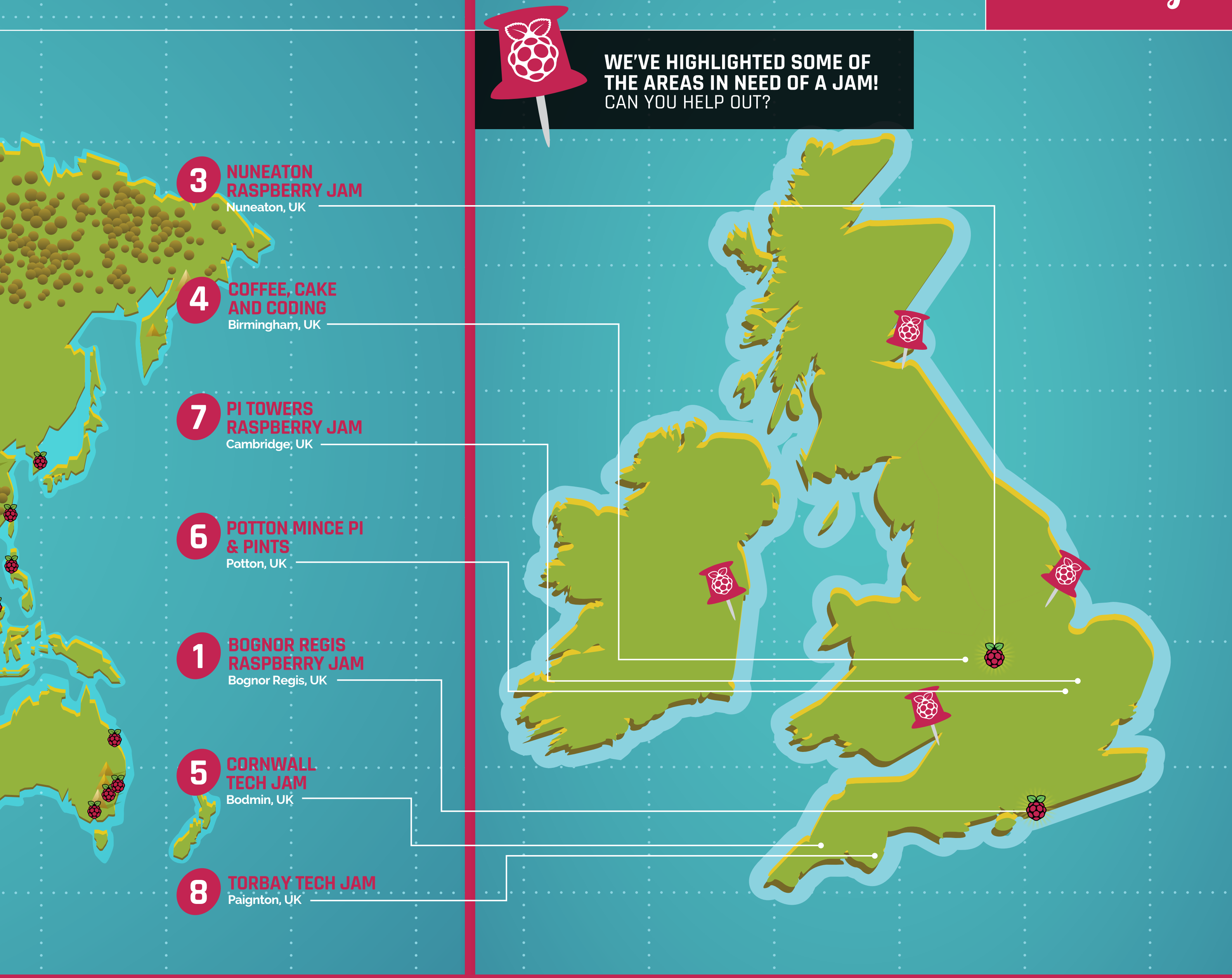
When: Saturday 9 December
Where: Bodmin Library, Bodmin, UK
cornwalltechjam.uk

For anyone interested in technology, any age or ability. Learn about coding in Scratch, Python, Minecraft, and more.

6 POTTON MINCE PI & PINTS

When: Saturday 9 December
Where: The Rising Sun, Potton, UK
magpi.cc/2zJwdmB

A Christmas-themed Pi & Pints event, where you have fun with Raspberry Pi in a relaxed, social atmosphere.


WE'VE HIGHLIGHTED SOME OF THE AREAS IN NEED OF A JAM! CAN YOU HELP OUT?
3 NUNEATON RASPBERRY JAM
Nuneaton, UK
4 COFFEE, CAKE AND CODING
Birmingham, UK
7 PI TOWERS RASPBERRY JAM
Cambridge, UK
6 POTTON MINCE PI & PINTS
Potton, UK
1 BOGNOR REGIS RASPBERRY JAM
Bognor Regis, UK
5 CORNWALL TECH JAM
Bodmin, UK
8 TORBAY TECH JAM
Paignton, UK

7

PI TOWERS RASPBERRY JAM

When: Saturday 9 December
Where: Raspberry Pi Foundation, Cambridge, UK
magpi.cc/2zKYo4L

A family-friendly event where everyone is welcome. Coding and digital making activities will be provided.

8

TORBAY TECH JAM

When: Saturday 9 December
Where: Paignton Library, Paignton, UK
torbaytechjam.org.uk

This Jam is designed to be a fun, informal, and family-friendly event, for all ages and experience levels.

RASPBERRY JAM ADVICE

PLANNING THE EVENT

> "We use the Eventbrite app to check in attendees where possible. However, people tend to arrive all at once and we don't like a queue. If it's not possible to scan the tickets, we just check them as they come in. It means our numbers are a little off, but it keeps people happier."
>
> **Michael Horne**
> CamJam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **magpi.cc/2q9DHfQ.**

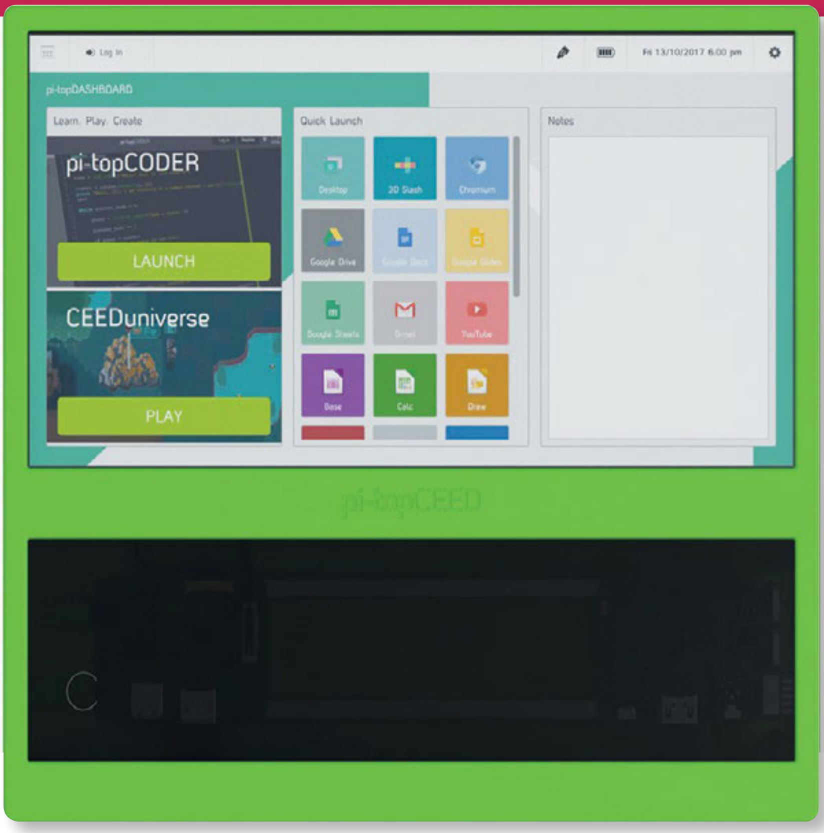
RASPBERRY JAM GUIDEBOOK
STARTING AND RUNNING A RASPBERRY JAM
ADVICE GATHERED FROM THE RASPBERRY PI COMMUNITY

YOUR LETTERS

Monitors of green

Whenever I see pictures get posted from official Raspberry Pi events like a Picademy or something similar, I always see these wonderfully vibrant bright green monitors that are used. Where do those monitors come from? Are they some kind of official Raspberry Pi monitor? I'd love to get one!
Frank Reid

They are very iconic monitors, aren't they? They're not official Raspberry Pi products; in fact, they're actually pi-topCEEDs. These are the cut-down desktop version of the pi-top laptop. They just include a screen, but you can still plug a Raspberry Pi into it – all you then need to add is a mouse and keyboard. They are also readily available for purchase; check out more details on them here: **magpi.cc/2zaTkWj**.



Right The CEED is used by the education and community team at the Raspberry Pi Foundation in many of their events

Missing files

I have a problem downloading the file scratch_gpio_handler.py from issue 10 of *The MagPi*. In the article 'Scratch Controlling GPIO 2 – Birthday Pi', there is a link you need to download files from using wget, but it doesn't work. Perhaps you can help me?
Max

Some of the content in older issues of *The MagPi* points towards an old MagPi website that no longer exists; sorry about that. However, the code from the first 30 issues of the magazine (and the majority of the other issues) can be found in our GitHub repo: **github.com/themagpimag**.

In this instance, the correct files can be found here: **magpi.cc/2jaiYU5**.

For cases like this, you just need to download the files manually from our GitHub repo, then proceed to the next step in the tutorial.

Free PDFs

I wanted to say thank you for allowing this great magazine to be downloaded in PDF form for free. I'm a disabled old guy and have a fixed income. I enjoy working on Raspberry Pi projects because it's one of the few things I can still do. Your magazine has kept me busy and is a joy for me to read every month. I wish you great success and deeply appreciate your generosity.
Dan

We really appreciate your message, Dan. Thank you so much for sending it!

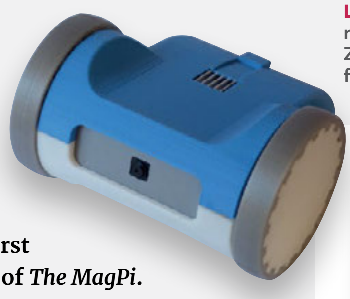
It's always worth reiterating that every issue of *The MagPi* is available as a free PDF download from our website. This includes all our Essentials books and the Official Projects Books as well, so if you're missing back issues then you can always grab a digital copy.

FROM THE FORUM: ROBOT BODIES

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

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**



Left You can now get your ZeroBot STLs from a new place

In one of the more recent *MagPi Essentials* books about GPIO Zero, there is a reference to ZeroBot that was first presented in issue 40 of *The MagPi*. As I seem to have all the stuff needed to build a ZeroBot, I thought I'd do it. However, the link to the thing on Thingiverse is broken. The thing seems to have been taken down. So I can't print the chassis...

Does anyone know where I can find the STLs nowadays? Or is there anyone who can supply them for me?

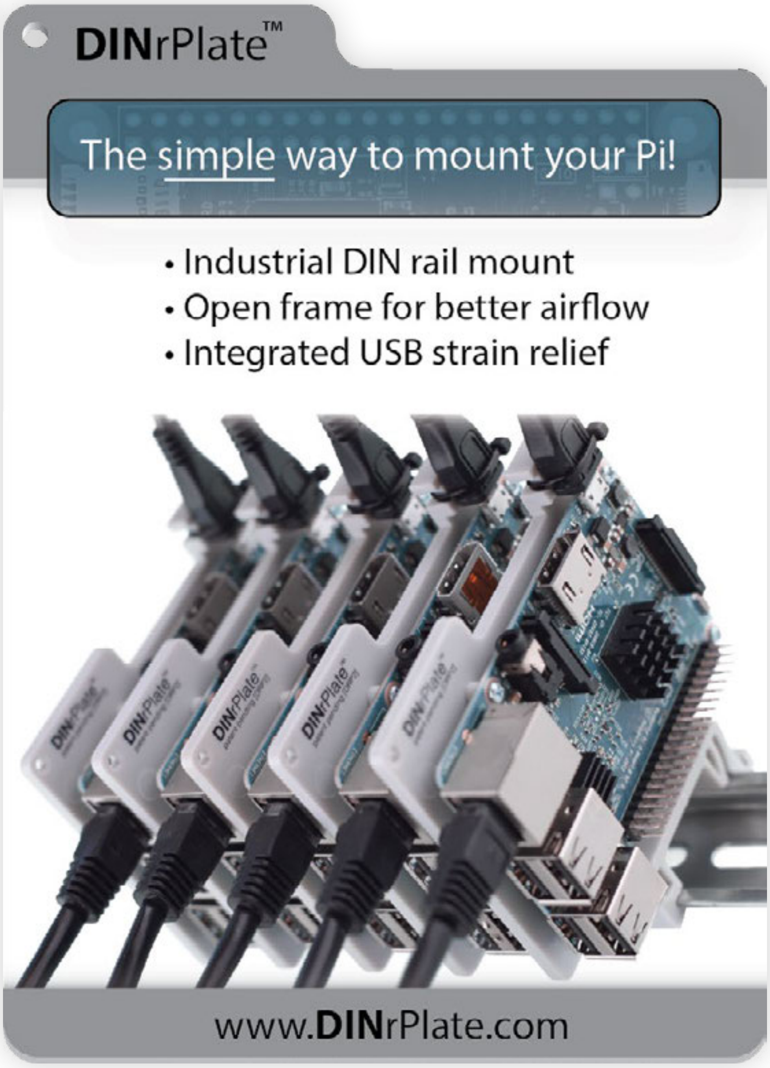
Thanks in advance!

Jonas Lorander

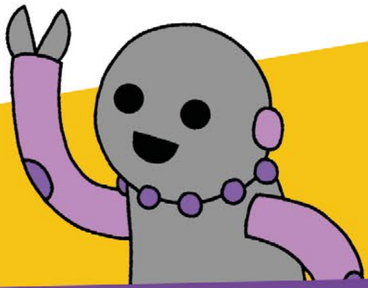
You're not the first person to ask us about this recently, so we thought we'd make a note of it here – the files for the original 3D-printed chassis that Richard Hayler found for that tutorial have mysteriously vanished from the internet. We did locate an alternative upload, but that too has gone.

Since then, someone has been kind enough to upload the files to Thingiverse, and even called it ZeroBot for easy Google searching! You can find it at the link here: **magpi.cc/2zW0mzb**.

Unfortunately, the internet is forever changing and not all of it is archived properly, so things like this may disappear. If you ever find a link that no longer works, please let us know via email at **magpi@raspberrypi.com** or on the forum and we'll try to help out.



code club

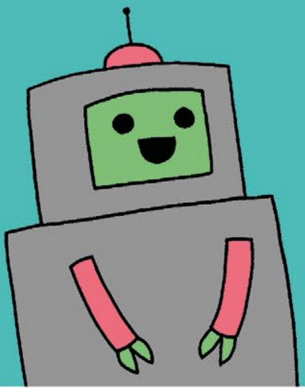


START A CODE CLUB IN YOUR SCHOOL!

Code Club is a network of volunteers and educators who run free coding clubs for young people aged 9-13.

Our aim is to inspire the next generation to get excited about computer science and digital making.

> "We use Code Club's fun educational resources to run a weekly after-school club for Year 7 and Year 8 pupils. The students benefit considerably from the extra challenge!"
>
> **Karen Dadd, Computing Teacher**



- ☑ **Code Club is free**
- ☑ **Code Club provides step-by-step guides for Scratch, Python, HTML, and Sonic Pi**
- ☑ **Code Club helps children develop skills including logical thinking, creativity, and resilience**

We have over 6000 clubs across the UK teaching more than 80,000 young people to code—come and join us!"

Find out more at www.codeclub.org.uk

Code Club is part of the Raspberry Pi Foundation. Registered Charity Number 1129409

WIN!

In association with piborg.org

MONSTERBORG & THUNDERBORGS

piborg “A beast of a kit designed to be taken off road or driven autonomously. It’s also the primary racing robot for Formula Pi.”

Enter now to win a MonsterBorg robot kit

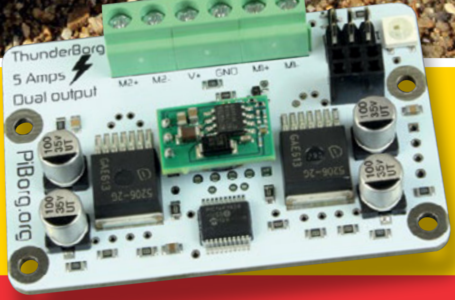
One lucky winner will own this robot

The MonsterBorg can be programmed as a self-driving robot, used for education about self-driving systems, or raced around the garden as an RC off-roader. It also makes a great platform for hacking about with robots by adding extra sensors or servos (**magpi.cc/2ltc8gT**).

For a chance to win a MonsterBorg or one of three ThunderBorgs, you just need to go online and enter our competition.

Three runners-up will win a ThunderBorg

Dual 5A Motor Controller with DC/DC & RGB LED



Enter now at magpi.cc/WinDec17

Terms & Conditions

Competition opens on 29 November and closes on 28 December 2017. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email no more than 30 days after the competition closes. 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. 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. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

MATT RICHARDSON

Matt Richardson is the Executive Director of the Raspberry Pi Foundation North America and author of *Getting Started with Raspberry Pi*. Contact him on Twitter @**MattRichardson**.

EMPOWERING EDUCATORS

Matt Richardson shares his thoughts on bringing digital making to educators

Raspberry Pi computers are meant to get young people to understand how computers work and to empower them with the ability to make things with computers. In order to do that, we make Raspberry Pi devices ultra-affordable and we offer a library of free project-based resources for getting started with them. But as you might already know, we don't stop there.

In 2017 alone, the Raspberry Pi Foundation has trained over 550 educators in digital making with Raspberry Pi computers at 'Picademy' workshops across the United Kingdom and the United States. These two-day, hands-on workshops are free for educators. On the first day, they learn about digital making, computer science, electronic circuits, and physical computing. They get the opportunity to try Raspberry Pi computers with Scratch, Python, LEDs, buttons, switches, the Camera Module, motors, and more.

On the second day, the educators are challenged to collaborate in groups to create a project with tools and concepts that they learned the day before. They are then sent home with a kit of materials so that they can take Raspberry Pi and digital making back into their learning environment right away.

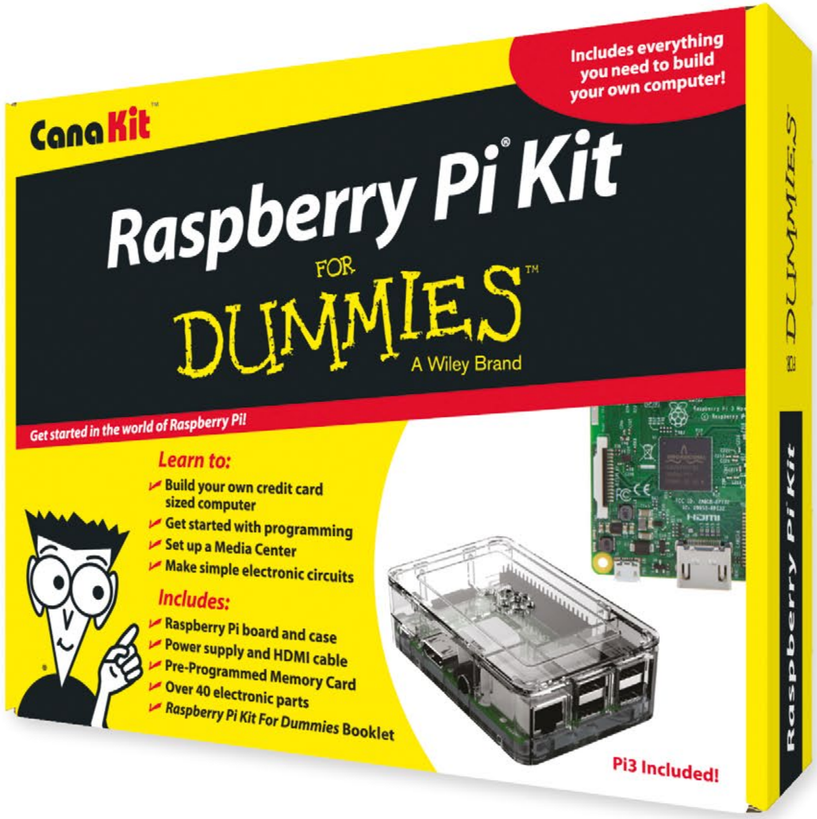
I want to take this opportunity to point out that these learning environments aren't only classrooms. It's important to us that we take a broad view of what 'educator' means. Picademy is not just for classroom teachers: it also includes librarians, museum educators, after-school program leaders, scout leaders, and makerspace educators. There are many opportunities for young people to learn inside and outside of the classroom and we want to make sure we're supporting all possible pathways to learning.

For our charity, training educators is a great way to have the biggest possible impact on young people around world. Educators know what works best for those in their teaching environment. They have the experience to reach youngsters and get them to understand the concepts behind digital making. When we show these educators the power of digital making and Raspberry Pi, they can take these lessons and apply them the way that they see fit for the kids that they work with. And purely from a numbers perspective, if every educator we train goes on to teach only ten young people on average, then our impact is already increased by an order of magnitude.

The Picademy experience doesn't end with the workshop. Those who attend become Raspberry Pi Certified Educators and join a passionate community of digital making enthusiasts. They stay in touch, collaborate, share best practices, organise Raspberry Jams, run Code Clubs, and train other educators back home.

I acknowledge that not every educator has the means to travel to our workshops and spend two full days with us. For those who aren't able to join us in person, we have a series of online courses on the FutureLearn platform. While these courses are geared towards educators, anyone can join them for free. They cover subjects such as physical computing, object-oriented programming, and teaching programming to children in primary school. You can join these courses here: **magpi.cc/2h5Sthf**.

Even if you're not an educator, you can still be involved in our educator training efforts. Every time you buy a Raspberry Pi or if you've subscribed to *The MagPi*, you're supporting our objective to train educators who will go on to have a massive impact on young people. And if you're looking for other ways of supporting, you can help get the word out! If you know an educator in your community who would be perfect for our training, please send them our way: **raspberrypi.org/training**.


Includes everything you need to build your own computer!
CanaKit™
Raspberry Pi® Kit
FOR DUMMIES™
A Wiley Brand
Get started in the world of Raspberry Pi!
Learn to:
Build your own credit card sized computer
Get started with programming
Set up a Media Center
Make simple electronic circuits
Includes:
Raspberry Pi board and case
Power supply and HDMI cable
Pre-Programmed Memory Card
Over 40 electronic parts
Raspberry Pi Kit For Dummies Booklet
Pi3 Included!
Raspberry Pi Kit
DUMMIES