YOUR OFFICIAL **RASPBERRY PI** MAGAZINE

The MagPi

The official Raspberry Pi magazine | Issue 67 | March 2018 | raspberrypi.org/magpi

RETRO COMPUTING

HARDWARE EMULATION & CLASSIC CODING WITH RASPBERRY PI

SMART HEARING AID
Building better audio aids with Raspberry Pi

MAKE A GEIGER COUNTER
Prep for fallout with a smart radiation tool

WATCHDOG ROBOT
The security guard bot on patrol

PI SECURITY
Browse safely and stay private online with Tor and Pi-hole

Also inside:

- AN ARCADE MACHINE IN A LUNCHBOX
- **USE WII NUNCHUKS TO CONTROL YOUR PI**
- CONNECT A PI TO A PC WITH GPIO EXPANDER
- **RASPBERRY PI ON DISPLAY AT THE V&A MUSEUM**

Issue 67 • Mar 2018 • £5.99
9 772051 998001 03
Raspberry Pi PRESS

CanaKit™

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!

WiFi™ Bluetooth®

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO

WELCOME TO THE OFFICIAL MAGAZINE

We all had to start somewhere. I started with a lovely ZX Spectrum, then an adorable C64, and finally an amazing Amiga.

These computers weren't retro when I got them. They were cutting-edge, well-engineered masterpieces packing state-of-the-art technology. Just like the Raspberry Pi today.

But a Raspberry Pi can become a ZX Spectrum, or a C64, or an Amiga, or an Acorn BBC Micro or Apple II. You can even emulate rarer machines like the MSX and Oric-1.

But why bother when you've got all that processing power at your fingertips? After all, there's no way a Spectrum could run a modern masterpiece like Minecraft.

Nostalgia plays a part. But there's still a lot to learn from classic computers. The nascent computer hardware of the 1980s is, in many ways, easier to understand that the multilayered and highly optimised technology stack used in modern computers.

Our Retro Computing feature (page 16) has everything you need to get started, including how to put a Raspberry Pi inside an old computer (in our example an old Sinclair Spectrum, but you can apply the techniques to other machines).

There's something wonderfully glorious about revisiting classic computers with a Raspberry Pi. It's the best of the old and the new working together in perfect harmony.

Lucy Hattersley
Editor

THIS MONTH:

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

The MagPi

Available on the App Store

GET IT ON Google Play

CC BY NC SA

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

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Mike Kay and Lee Allen**
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
Bill Ballard, Brian Beuken, Mike Cook, David Crookes, Gareth Halfacree, Maanas Manoj, Sean McManus, Ben Nuttall, KG Orphanides, Matt Richardson, Richard Smedley, Clive Webster

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

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

Contents

Issue 67 March 2018

raspberrypi.org/magpi

TUTORIALS

IN THE NEWS

COVER FEATURE

Contents

MOZILLA PROJECT THINGS

New 'World Wide Web of things' open standard

Firefox maker Mozilla has launched a new, open standard for IoT and smart homes called Project Things. Mozilla announced that Project Things "makes it easy for anyone with a Raspberry Pi to build their own Things Gateway [to control any kind of smart device] directly from the web."

We asked Ben Francis, Mozilla software engineer, for more detail. "The Things Gateway is designed to bridge existing smart home protocols and devices to the 'Web of Things' using 'adapters'," he says, to add extra wireless protocols to the Pi.

Ben confirms, "The latest version has built-in adapters for the Zigbee and Z-Wave protocols [as well as] adaptor add-ons for devices from brands like Philips and TP-Link... We're calling on the community to help us build a whole directory of add-ons."

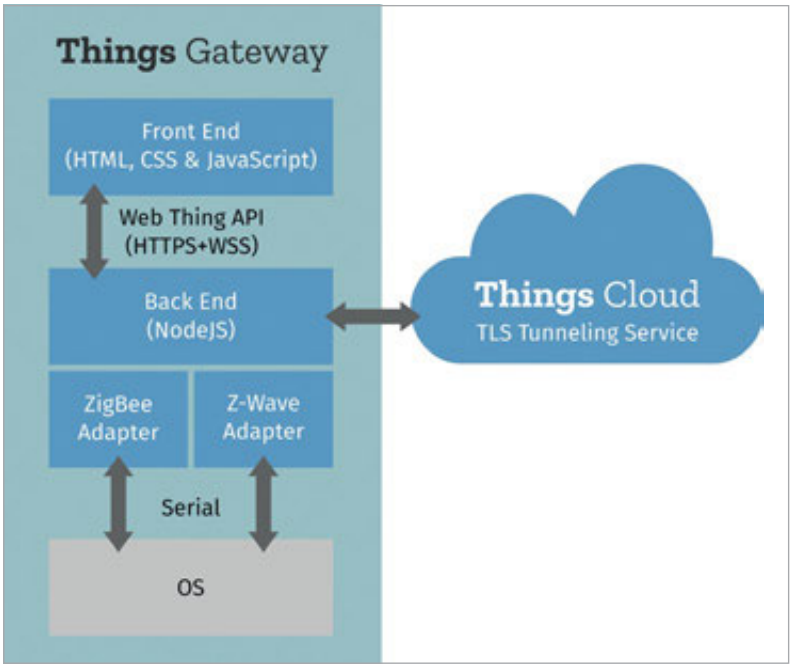
Above Use the simple web interface to control all your smart devices

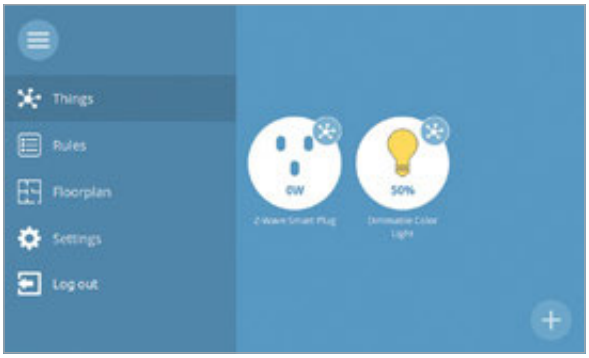
Many smart home devices use Zigbee or Z-Wave to minimise power draw. However, as few smartphones have Zigbee or Z-Wave hardware, smart home apps have to relay information via a physical hub unit. These hubs tend to be proprietary, closed 'black boxes' – even worse, you might need one for your lights, another for your heating system, and yet another for your multi-room speakers.

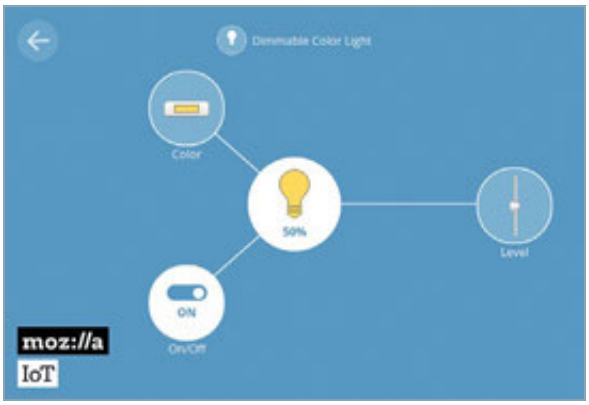
Ben says that Project Things aims to "standardise web technologies as a common layer", thus consolidating the need for hubs into one open-source, customisable and upgradable, Raspberry Pi-powered unit.

Beyond the black box

Any smart device should be compatible with a Things Gateway, as Ben clarifies: "Our goal is not to create a proprietary Mozilla IoT platform, but rather help to build a World Wide Web of things, much like the World Wide Web of pages you can browse with Firefox."







Above Project Things uses open-source standards to bring together competing smart home standards

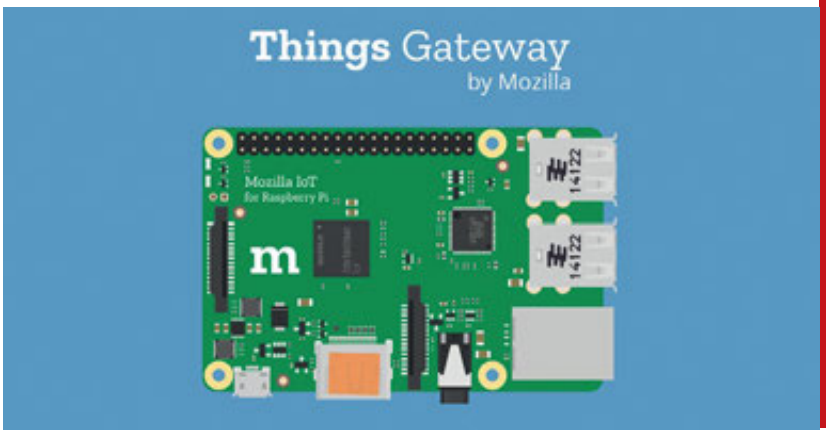
Ben expounds that by "using the Web Thing Description format and Web Thing API (**iot.mozilla.org/wot**) we're contributing towards standardising through the World Wide Web Consortium (W3C)."

Ben adds that "working towards royalty free IoT standards, with multiple competing implementations, should help to drive down the certification and licensing costs, [both for hardware devices and] the software needed to connect them together."

> "To build a World Wide Web of things, much like the World Wide Web of pages"

Interestingly, given Mozilla's web-based background, Ben confirms that "the rules engine built into the Things Gateway runs locally on the home network." Further, "Mozilla provides a secure tunnelling service to make it safe and easy to access the gateway over the internet."

The Project Things GitHub page is live (**github.com/mozilla-iot**) and you can start making your Things Gateway at **iot.mozilla.org**. Mozilla recommends a Raspberry Pi 3, and a few USB dongles to add support for Zigbee and Z-Wave.



THE FUTURE OF PROJECT THINGS

Project Things was only announced in June 2017, so to have working code already just seven months later is pretty impressive. But Mozilla isn't stopping there.

Mozilla software engineer Ben Francis explains, "The Things Gateway is one part of a framework of software and services from Mozilla to help create greater interoperability, privacy, and security on the Internet of Things."

Ben even gave some details of the kinds of thing we can expect as part of the Project Things framework: "We also plan to provide a collection of IoT cloud services (starting with the secure cloud tunnelling service) and a device framework which provides reusable software libraries for platforms like Arduino and Android Things."

The aim is to allow developers to create "native web things which connect to the Web of Things directly."

With open-source software, combined with cheap and well-supported hardware such as the Raspberry Pi and Arduino boards, Project Things could finally bring some harmony to the confusion of conflicting smart home standards.

COOLEST PROJECTS COMES TO THE UK

Above Coolest Projects is an international event for young digital makers aged 18 and under

Below There's plenty to do at Coolest Projects, besides demonstrate your own build

Images courtesy of: Kenneth O'Halloran, Coolest Projects 2017

CoderDojo event expands

An international show for young digital makers to demonstrate their projects, CoderDojo's Coolest Projects is coming to the UK for the first time this year.

Coolest Projects International will take place at the Royal Dublin Society on Saturday 26 May (**coolestprojects.org**), while Coolest Projects UK is on Saturday 28 April, at Here East in the Queen Elizabeth Olympic Park, London (**hereeast.com**).

Anyone under the age of 18 and currently in primary, secondary or further education can enter, either as an individual or part of a team of up to five members. Projects are entered into five broad categories: Scratch, Websites, Games + Web Games, Mobile Apps, and Hardware. The full criteria for each are at **magpi.cc/2CPqaK5**.

Rosa Langhammer, CoderDojo General Manager, Outreach & Engagement, explains: "Coolest Projects is about bringing an idea and sharing it with your peers, no matter how big or small or even if your project isn't finished." The entry categories are so broad because CoderDojo wants "Coolest Projects to be as inclusive as possible."

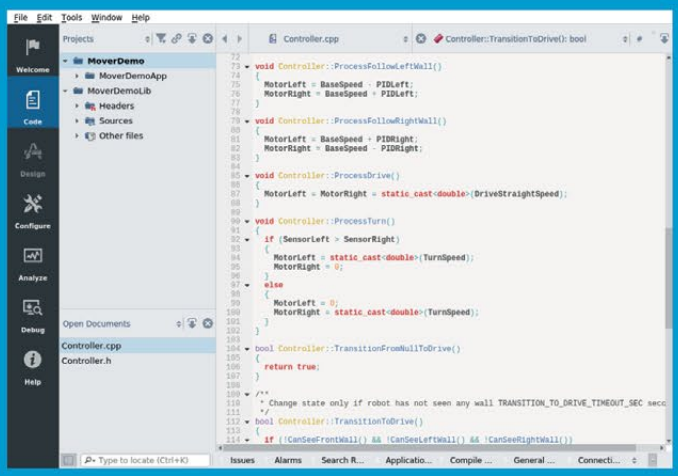
Inspiration across the nations

Coolest Projects started "in 2012 with 19 projects," Rosa confirms. Now Coolest Projects is "an international showcase with 750 young people participating last year from 16 countries!"

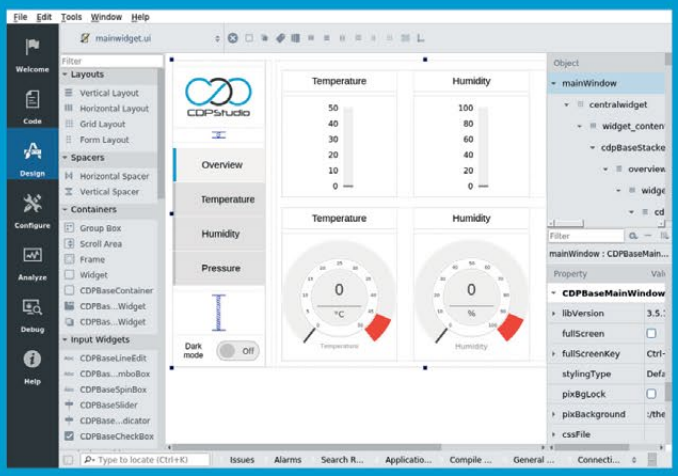
Each participant will have their own area to set up their project, and will "have some free time throughout the day to explore other projects, as well as some of the cool demos and speakers that will be joining us for the day," says Rosa.

"It's really important to have hands-on activities scattered throughout," Rosa continues, "so parents, young people, and the general public can get hands on with science, technology, and the arts."

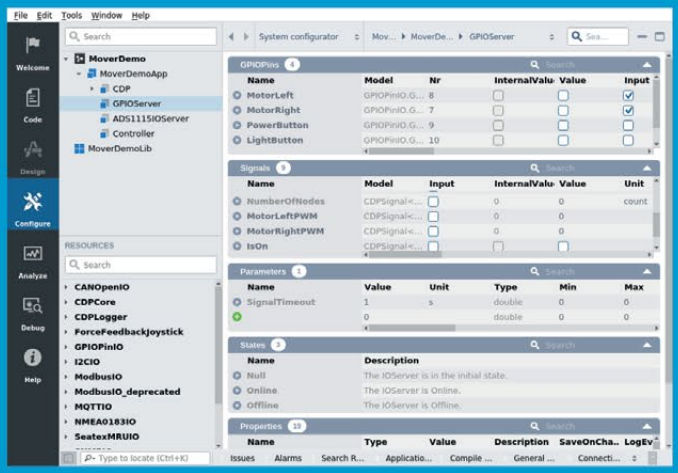
Rosa's favourite Coolest Project came from 12-year-old Amy, who created a smart beehive "with a mission to save bees!" Amy uploaded data from her hive to **hivetool.net**, helping "international scientists to understand why bees are dying."



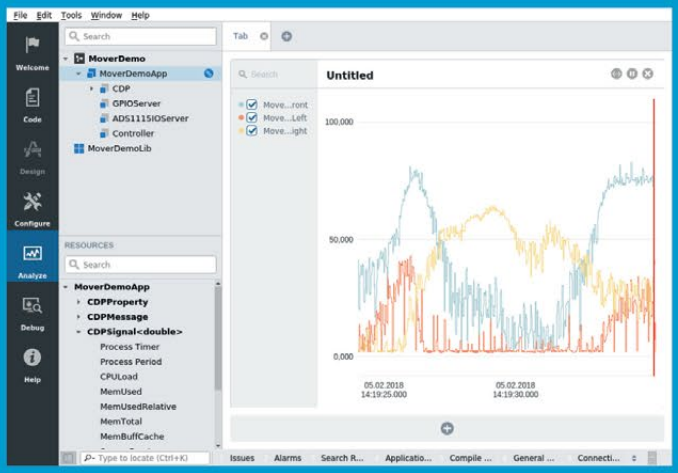
Code



Design



Configure



Analyze

Now free for home projects

A professional control system development tool

CDP Studio is a development platform for industrial control systems, now coming with a free version for non-commercial use. The system can run on a Raspberry Pi, supports C++, open source libraries and has a large feature toolbox including GPIO, I2C and MQTT. Its built in GUI design tool and features lets you code less and do more.

Free download on www.cdpstudio.com

CDP Technologies AS
Nedre Strandgate 29
P.O. Box 144
NO-6001 Ålesund, Norway

Tel: +47 990 80 900
info@cdptech.com
www.cdpstudio.com

CDPStudio

PI-MADE MUSIC ALBUM

NOW That's What I Call Chiptune!

Demoscene musician Yerzmyey has created an entire music album on a humble Raspberry Pi 2, called RPI Zwei.

Yerzmyey's music is as full and complex as any track you'd hear on the radio or Spotify. Yerzmyey tells us that "The most complex song on the album has 26 independent channels of digi-music" and even says the Pi "could surely manage more."

As Yerzmyey explains, the Pi doesn't have a DAC (a digital-to-analogue converter) so instead you have to record music on the processor itself. This limits the audio to 11-bit, 40 MHz, but that doesn't seem to have held Yerzmyey back.

Having first started creating 'demoscene' music on a ZX Spectrum in the 1980s, Yerzmyey used MilkyTracker (**magpi.cc/2C2HqhH**) to "program all [the] notes one-by-one... Otherwise I use loops only when it comes to drum 'n' bass" and instruments such as guitar. Layering notes and loops, Yerzmyey "composed the music on the Pi and then recorded it all directly from the 'jack' output."

You can listen to RPI Zwei here: **magpi.cc/2DC1Ym5**.

Yerzmyey blowing out beats in Tokyo in 2014

NEW PI-FRIENDLY 10.1" TOUCHSCREEN

IPS tech for iPad visuals

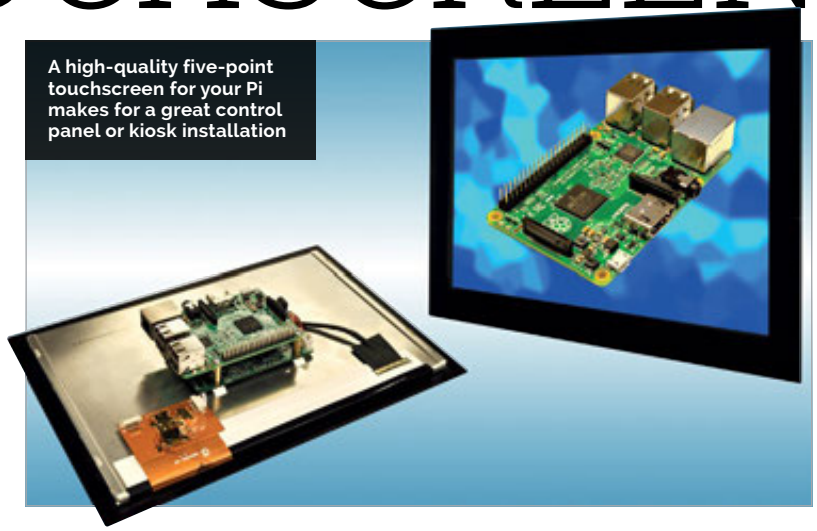
Reading-based electronics specialist supplier Inelco Hunter has launched a new touchscreen, designed for Raspberry Pi. The screen has a 1366×768 resolution and uses IPS technology.

The use of IPS technology over the cheaper TN offers "a wider view angle and the full 16.7 million colours," says Inelco Hunter managing director David Bushnell.

David tells us that demand is for wide viewing angles and higher definition; IPS technology also offers "a better contrast ratio" and so delivers stronger, punchier colours.

The 10.1-inch screen "is designed to work with all models of the Raspberry Pi," David confirms, with standoff pillars and screws provided.

The screen "supports Raspbian and NOOBs," says David. "We are working on drivers for the I²C interface" if you'd rather not use the currently supported USB connection. "Once Raspbian is installed, the five-point touch will work automatically," David explains.



A high-quality five-point touchscreen for your Pi makes for a great control panel or kiosk installation

The new screen is on sale now, from **magpi.cc/2C35bX4**, but pricing is based on order quantity. You can contact Inelco Hunter directly at **marketing@inelcohunter.co.uk**.

WORLD'S SMALLEST TV STATION

LimeSDR dongle turns a Raspberry Pi into a TV broadcast mast

Software-defined radio (SDR) enthusiast Evariste Okcestbon has made possibly the world's smallest TV station, using a LimeSDR Mini dongle, a Raspberry Pi Zero, and a mini camera module to broadcast a TV signal to a Pi 3 with USB TV tuner.

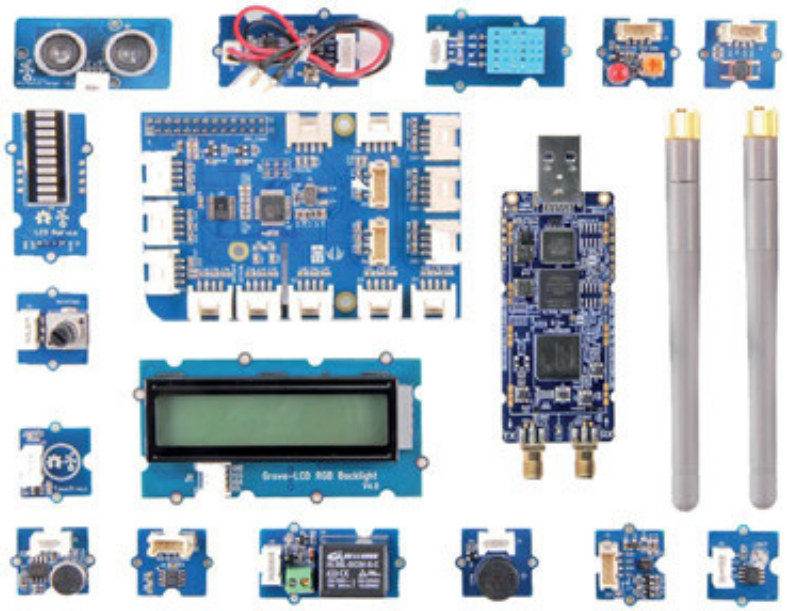
> "We wanted people to be able to simply pick one up and play"

Evariste showcases his build on YouTube (**youtu.be/jo6pCIeTIzs**), and then explains his setup, which gets quite technical.

Ebrahim Bushehri, CEO of Lime Microsystems (maker of the LimeSDR Mini dongle), tells us, "We were delighted to see Evariste's system." Ebrahim adds that he's also seen "systems to talk to the ISS Space Station, LTE [4G] base stations, [and] IoT platforms."

LimeSDR recently released an SDR Starter Kit to make it easy for anyone to build similar projects. The Kit uses Grove connectors for the bundled 15 sensors and inputs, and is programmable in Scratch. Ebrahim explains, "We wanted people to be able to simply pick one up and play."

The Starter Kit costs $249 from **magpi.cc/2BZp6WF**.



Above Evariste's build might be a bit advanced, but the LimeSDR Starter Kit should get you going without any need for soldering or text-based coding

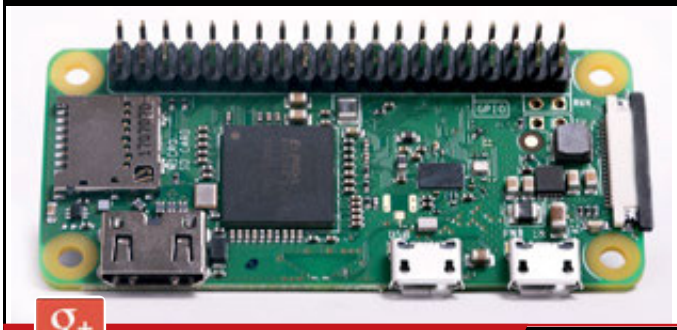
NOW TRENDING

The stories we shared that flew around the world

THE 3000-CORE RASPBERRY PI CLUSTER

magpi.cc/2F2MUv1

BitScope partnered with the Los Alamos National Laboratory to build a single box containing 750 Raspberry Pi 3s in a cluster configuration – that's 3000 processing cores!



NEW RASPBERRY PI ZERO WH

magpi.cc/2C1MeEf

For those who don't have the confidence, kit, or time to solder a GPIO header to a Raspberry Pi Zero W, the organisation has launched the Pi Zero WH – a Pi Zero W with pre-soldered header.

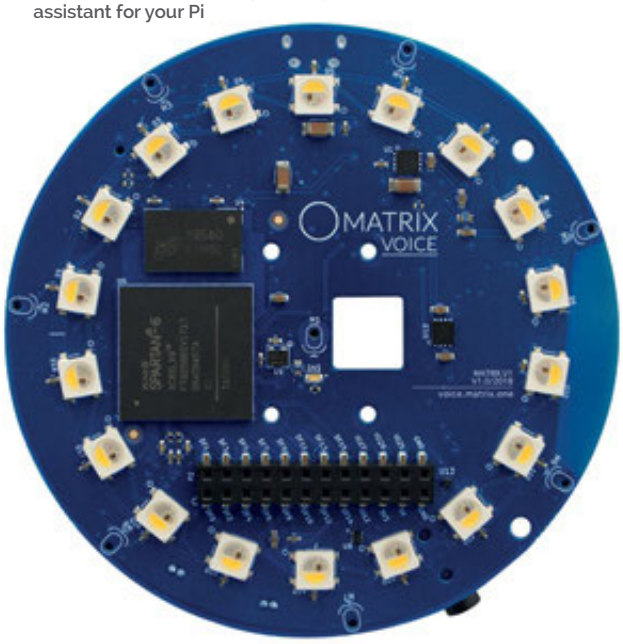
WEBKIOSK OS 7 RELEASED

magpi.cc/2COWN1T

A Raspberry Pi makes for an ideal kiosk PC, but you'll need a robust, secure, and dedicated OS to go with it – enter Raspberry WebKiosk 7, now updated to Raspbian Stretch.

NEW EIGHT-MIC MATRIX VOICE BOARD

Below With an eight-microphone array and on-board audio processing, the Matrix Voice is a complete digital assistant for your Pi



Ideal for DIY digital assistant builds

Matrix Voice's latest IoT add-on, the Matrix Voice, has been released in the UK – via Newark element14, or direct from **matrix.one**. The standard circular board (with a female header that slots onto the Pi's GPIO pins) costs around £40, while the ESP32 version costs roughly £45.

Loaded with eight microphones, stereo output (headphone and 3W speaker amplifier), plus a ring of 18 RGBW LEDs, WiFi, Bluetooth 4.0 LE, and a dual-core Tensilica Xtensa processor, the Matrix Voice is compatible with all digital voice assistant standards, even "on the edge services like Pocketsphynx and Snowboy," according to Matrix Labs CEO and co-founder Rodolfo Saccoman.

"Developing voice-enabled IoT products should not only be reserved for those with multimillion-dollar hardware budgets," Rodolfo tells us. "The open-source Matrix Voice empowers makers and industrial engineers."

Rodolfo confirms that the Matrix Voice has been tested with the Raspberry Pi 3, Zero, and Zero W, and that "we provide beamforming and direction of arrival methods" with the supporting software, which is compatible with Raspbian and Android Things.

EBEN UPTON ON RADIO 4

The Life Scientific interviews Raspberry Pi co-founder

Raspberry Pi co-founder Eben Upton appeared on a recent episode of BBC Radio 4's *The Life Scientific*, to discuss the amazing success of the Raspberry Pi and how it came to be.

Host Jim Al-Khalili asked Eben how the name 'Raspberry Pi' came about – it follows the grand tradition of naming computers after fruit, with Eben even listing "Acorn – very technically a fruit" among more obvious names such as Apple.

"Actually," Eben continued, "it's such a long tradition that there aren't that many [fruits] left." The other part of the name comes, Eben says, from Python: "Python has always been at the heart of everything we've done because it's a fantastic introductory programming language."

Eben also detailed the serendipitous role that BBC technology correspondent Rory Cellan-Jones played in the creation of the Raspberry Pi as we know and love it today.

Above Eben talks about the birth and success of the Raspberry Pi on Radio 4 show The Life Scientific

You can listen to the whole program on the BBC Radio 4 website: **magpi.cc/2CPZcSA**.

PYTHON DAY ON THE ISS

Astro Pi Mission Zero Executes 1771 scripts in a day

A recent image of Italian astronaut Paolo Nespoli using one of the two Astro Pi modules

With over 2500 entries to the new Astro Pi Mission Zero challenge, over 1771 Python scripts made it to the International Space Station, hurtling through space 250 miles above us. Each script was executed in series on Thursday 1 February.

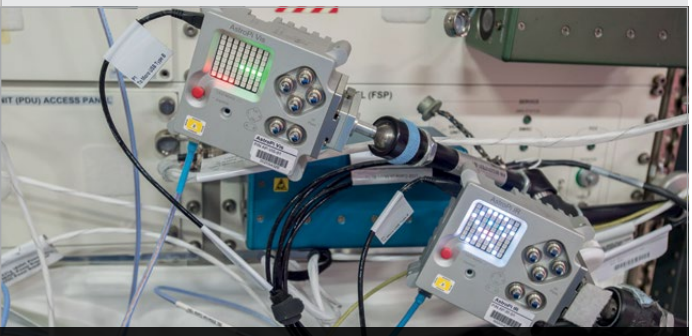
The Mission Zero challenge was to write a Python script that showed the astronauts aboard a message on the Astro Pi LED matrix while sensing and displaying the temperature of their ISS cabin. Each script that passed the rigorous scrutiny procedure (you can't just run any old code in space) was guaranteed to run on an Astro Pi aboard the ISS for 30 seconds.

MCP is in control

Thankfully, each script didn't need to be launched manually. As David Honess, Raspberry Pi Foundation education resource engineer, explains, "We have a special program called the MCP (master control program); it's a homage to the 1982 Disney film *Tron*. This is responsible for running all the programs in sequence and storing the times, ISS locations, and the process exit code."

The location of the ISS when executing each script is important, as each successful applicant will receive a certificate showing the location of the ISS when their code executed.

If you want to spot the ISS (it orbits Earth 15 times a day), David recommends pointing your web browser to Heavens Above: **heavens-above.com**.



The two Astro Pi modules – you can emulate them using the Sense HAT emulator, developed in partnership with Trinket (trinket.io/mission-zero)

pi-top

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

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.

pi-topPROTO

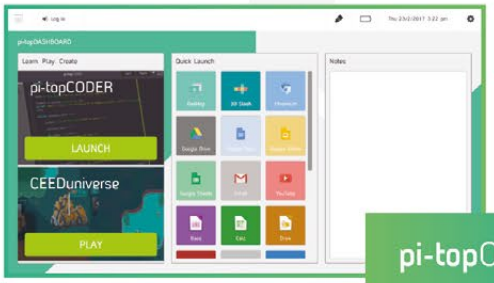
pi-topSPEAKER

pi-topPULSE

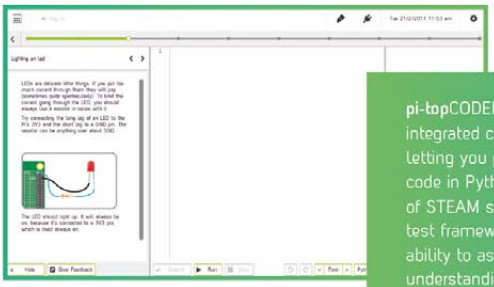
Modular **Accessories**

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-topOS



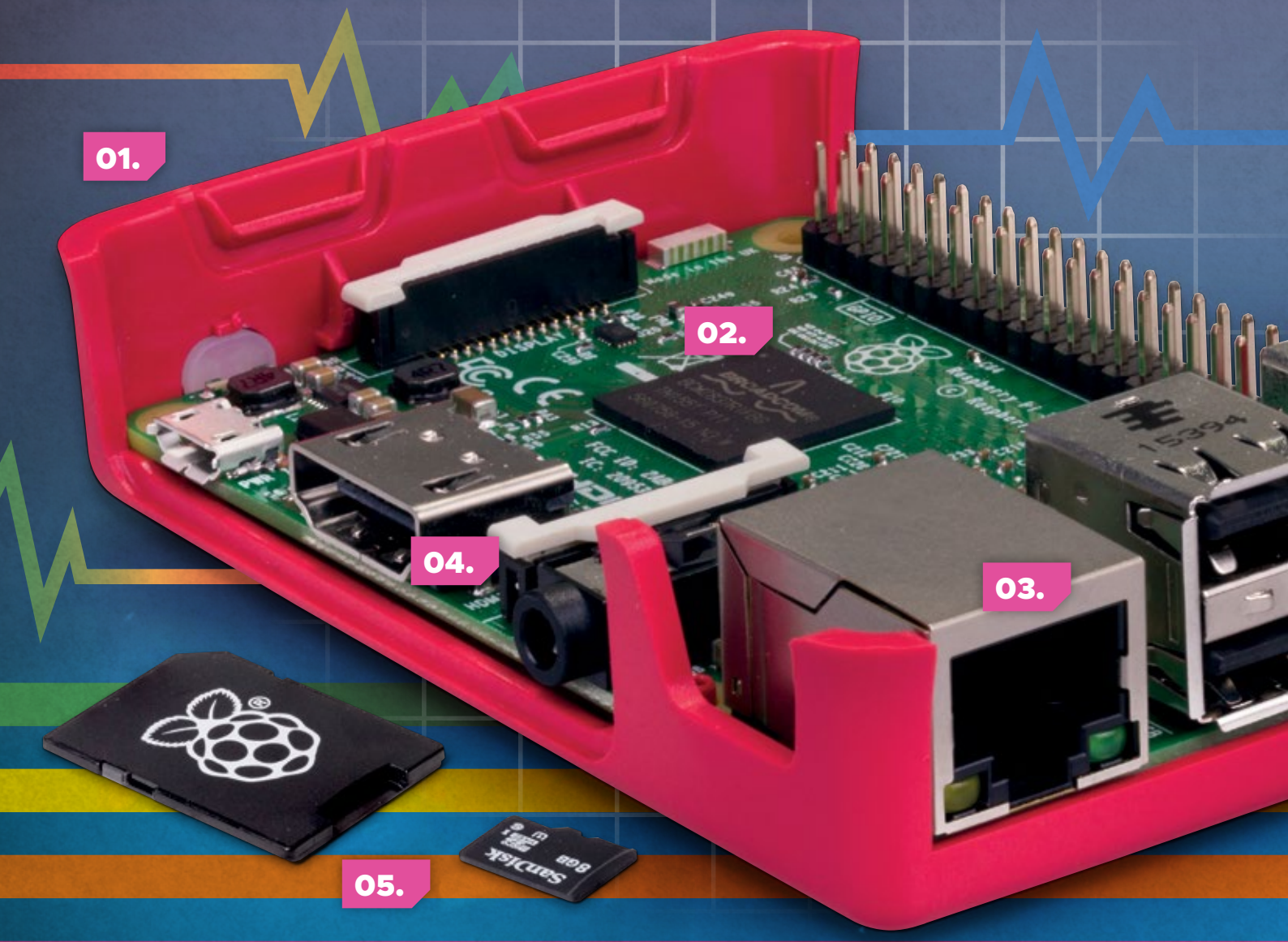
pi-topCODER has a fully integrated coding environment letting you program hardware, code in Python and learn lots of STEAM skills! Our integrated test framework gives you the ability to assess your own understanding as you learn.

CEED**universe**

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

RETRO COMPUTING

HARDWARE EMULATION & CLASSIC CODING WITH RASPBERRY PI



Everyone we know has their own favourite computer. Many of *The MagPi* team still love the Sinclair Spectrum; others remain loyal to the Commodore 64.

Eben Upton's eyes light up when he talks about his Commodore Amiga.

It was this experience of classic computing that led to the creation of the Raspberry Pi. This marvellous computer was designed to teach young people to program. Back in the 1980s, kids had to program computers in order to use them. Consequently, these kids then grew up with an innate understanding of how computers work.

The Raspberry Pi is an amazing computer that harks back to this golden age, and it's capable of emulating the very machines that inspired it. You can turn a Raspberry Pi into a Sinclair Spectrum, or a Commodore Amiga, and discover (or rediscover) all the classic programs, software, and coding techniques that kids learnt back in the 1980s.

You can use a Raspberry Pi in its original case. Or even put a Raspberry Pi inside the case of a classic computer. Then you get all the love of the original machine, with lots of modern features (such as HDMI support and networking).

In this feature we'll do all of that and much more. Get ready to go back in time to the golden age of computers.

01. CASE

You can place the Raspberry Pi inside its official case, or fit one inside a retro-themed case (or even an actual retro computer).

02. POWERFUL PROCESSOR

The modern ARM processor inside a Raspberry Pi is vastly more powerful than the chips found in older computers. A Raspberry Pi can easily emulate computers like a Commodore 64 or Spectrum.

03. MODERN CONNECTIVITY

A big advantage the Raspberry Pi has over older computers is Ethernet and wireless networking capability. This makes it much easier to upload files to your retro computer, and some hobbyists have even created stylised web browsers for classic computers.

04. VIDEO OUT

Most Raspberry Pi models connect to an HDMI television. You can also use the 3.5 mm audio/video jack on a Raspberry Pi 2 or 3 (or A+ or B+) to connect AV cables to a classic CRT (cathode ray tube) television.

05. MICRO SD CARD

If you plan to download lots of emulated software, then you might need a larger microSD card. But 8GB should be plenty to set up and test a retro computer.

KIT YOU'LL NEED

RASPBERRY PI

We'd recommend using a Raspberry Pi 3 to power your emulated computer. But you can use other models, like the Pi Zero W.

HDMI CABLE

We recommend starting out by hooking your Raspberry Pi to an HDMI display using a regular cable. This will be easier to set up than going straight for a classic CRT option.

MOUSE & KEYBOARD

While many older computers didn't use a mouse, you'll still need a USB mouse and keyboard to install and set up the emulation software.

HARDWARE EMULATION

RETROPIE IS YOUR ONE-STOP SHOP IF YOU WANT TO EMULATE CLASSIC COMPUTERS AND CONSOLES, FROM THE ZX SPECTRUM TO EARLY IBM PCS

YOU'LL NEED

- RetroPie **magpi.cc/2Eq4H2o**
- 7-Zip **magpi.cc/2EsAYFU**
- Etcher **etcher.io**
- USB stick
- USB game controller (optional)

While you can emulate each classic computer system you fancy separately, either using a dedicated bootable image or a desktop application, we've picked RetroPie for our main emulation tutorial.

It's not perfect, and in some cases you'll get better results by installing emulators individually, but it's the coolest and easiest way to get a feel for the world of emulation.

RetroPie comes with a huge selection of pre-installed and optional emulators for almost every computer you can think of, with the conspicuous exception of the BBC Micro, although AdvMAME provides some functionality in this department.

A Raspberry Pi Zero is powerful enough to smoothly emulate 8-bit computers, but if you want to experiment with more demanding software and emulators such as DOSBox, we recommend using a Raspberry Pi 3.

>STEP-01

Download and image RetroPie

While you can install RetroPie on top of Raspbian, the ready-made RetroPie distro is easier to use and more resource-efficient. Download the appropriate version for your Pi from **magpi.cc/2Eq4H2o**, extract it, and use Etcher to copy it to a microSD card.

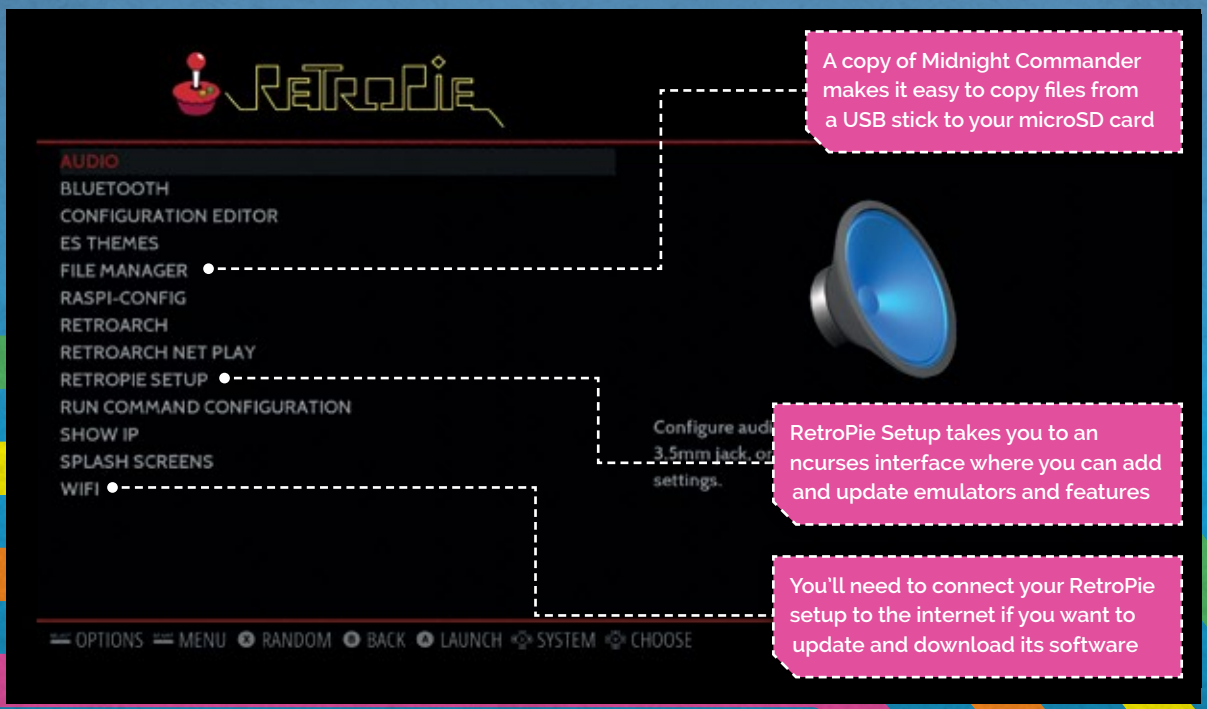
RetroPie is a chunky distro, so you'll need at least a 4GB card if you want to install a few different emulators, and we recommend opting for a 32GB SD card to leave plenty of room for your software collection.

>STEP-02

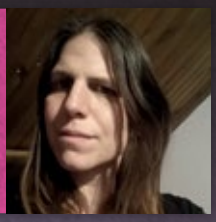
Set up RetroPie

The first time you boot its SD card, RetroPie will take a few minutes to complete its installation procedure before handing over control to you.

You'll be asked to configure your controllers. You can use a



K.G. ORPHANIDES
K.G. is a software preservationist and occasional developer with an abiding love of vintage computers.
twitter.com/kgorphanides



Left A customisable configurator lets you assign input device buttons as needed

keyboard for this, but RetroPie is easiest to navigate using a joystick or gamepad. Press and hold any key on your input device of choice to set it up. If you run out of buttons, just press and hold a key to skip each subsequent setting. At the bottom, press the button you set as A to complete the process.

If you don't have a physical network connection, your next step should be enabling WiFi. You can scroll between different emulators in RetroPie's main settings menu using the left and right buttons.

To configure WiFi, move to RetroPie menu shortcut (your emulator should open on this screen) and press button A. Select the WiFi option at the bottom of the menu.

>STEP-03
Add extra emulators

RetroPie comes preconfigured with a number of popular emulators, and a few old computers are supported out of the box, including the ZX Spectrum (via lr-fuse) and Amstrad CPC (with Caprice32).

However, its default set is mostly focused on old-school consoles. As we're more interested in classic computers than in consoles, you'll want to add a few extras.

On the RetroPie menu, go to RetroPie Setup, then Manage packages > Optional packages, where you'll find the original version of FUSE (required for our Spectrum tutorial on page 24), DOSBox, Hatari, Vice, and dozens more emulators dedicated to everything from arcade machines to point-and-click adventure games.

Remember to check the operating instructions for each emulator at **retropie.org.uk/docs**.

>STEP-04
Find some software

Most of RetroPie's emulators come fully loaded with the ROM (read-only memory) images of the computer operating systems you'll be using, but unless you want to write all your own programs from scratch – which is certainly an option – you'll have to find some software.

There are plenty of websites dedicated to sharing new and old software for vintage computers. Specialist sites such as **worldofspectrum.org** and **c64.com** are packed with programs to download for your emulator.

While a lot of vintage software has been placed in the public domain or made open-source by its creators, 'abandonware' has no legal meaning and downloading no-longer-supported software counts as piracy in many countries, including the UK.

>STEP-05
Boot to the OS

RetroPie's default behaviour is to boot directly into the software you're trying to run on a given emulator, but this isn't much help if you want to access its original operating system.

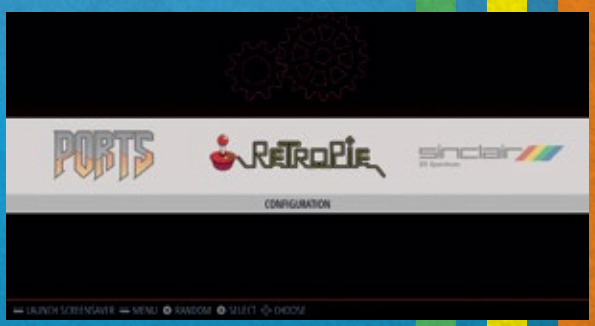
Some of its emulators, like FUSE, are installed with a blank file that gives you access to its emulated Spectrum interface. For others, such as the Vice C64 emulator, the easiest option is to open the Midnight Commander file manager from the settings menu, navigate to the relevant **roms** subdirectory and create a blank file with the appropriate extension.

For Vice, go to the **~/RetroPie/roms/C64** directory and type **touch empty.prg**. Load from the graphical interface and you'll be taken straight to the C64's BASIC interface.

>STEP-06
Copy over your files

The most obvious way to copy over your files is to put them on a USB stick, plug it into your Raspberry Pi, and use Midnight Commander, as described above, to transfer everything into an appropriate directory.

However, if connected to your network, RetroPie also uses Samba to share a directory that you can copy files to from other computers on the network. You can also enable the browser-based retropie-manager, via the experimental package list under 'Manage packages'.



Above RetroPie provides a polished, joystick-friendly front end to access your emulators and software

AMAZING EMULATORS

EMULATORS ARE AVAILABLE FOR EVERY RETRO COMPUTER YOU CAN THINK OF, BUT NOT ALL OF THEM PLAY NICELY WITH THE RASPBERRY PI. HERE ARE OUR FAVOURITES, WITH A BIT OF EXTRA INFO ON HOW TO USE THEM AND WHAT TO DO WITH THEM

SPECTRUM: FUSE

> **magpi.cc/2ErcA7S**

The Free Unix Spectrum Emulator (FUSE) is one of the longest-running and best-supported emulator projects around. It's in the Raspbian and Ubuntu MATE repositories, so you can install it with **`sudo apt install fuse-emulator-sdl`**.

Before you get cracking, you'll want some Spectrum operating system ROM files (**`sudo apt install spectrum-roms`**) and utilities (**`sudo apt install fuse-emulator-utils`**). Press **F1** to access the menu (and its full screen in the options menu, needed if you run it from a GUI).

FUSE opens with a perfect emulation of the ZX Spectrum BASIC, so you can start programming straight away.

APPLE II: LINAPPLE-PIE

> **magpi.cc/2ErIWiC**

The Apple II was always more popular in the USA than in the UK, but as an early step on the hardware development path that led to Apple becoming the household name it is today, it resulted in the creation of some of most influential software ever made, particularly games.

LinApple-Pie provides an authentic recreation of the Apple II and its implementation of BASIC, and is available for RetroPie and via GitHub to download and compile under any Linux-based Raspberry Pi OS.

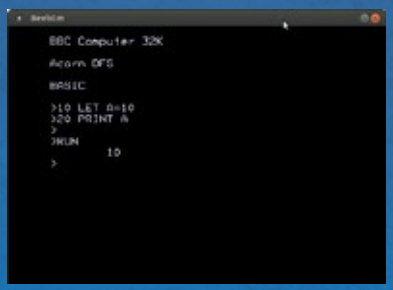
ATARI ST: HATARI

> **hatari.tuxfamily.org**

The Atari ST spent much of the 1980s and 1990s as the arch-rival of the Commodore Amiga for home computing, the two 16-bit computers having superseded the primitive-by-comparison 8-bit machines. In addition, the ST's integrated MIDI ports and generous memory (anywhere between 512kB and 4MB) saw it appearing in music production studios across the UK.

The Hatari emulator supports USB MIDI adapters, which means you can connect MIDI input devices like keyboards and output hardware like Roland's SC-55 with software such as Cubase and Notator.

You can install Hatari from Raspbian's standard repositories (**`sudo apt install hatari`**), but you'll need ROM images of Atari's TOS (The Operating System), which you can learn about at **magpi.cc/2Em8Fcv** and download from **magpi.cc/2EstrqE**.



BEEBEM

> **magpi.cc/2EoKAS3**

The BBC Micro isn't as well supplied with emulators as some of its more international rivals, but although BeebEm hasn't been updated in over a decade, it's solid and stable. However, compiling it involves a couple of extra steps.

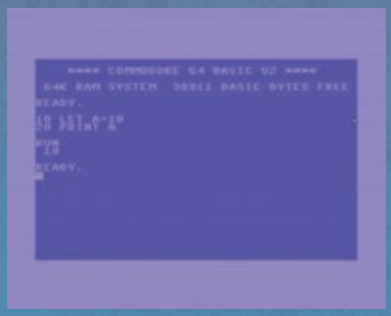
Download **beebem-0.0.13.tar.gz** and the 64-bit, keys, and menu_crash patches. Now:

```
sudo apt-get install
libgtk2.0-dev libsdl1.2-dev
tar -zxvf beebem-0.0.13.tar.gz
```

Before you **cd** into the directory…

```
patch -p0 < beebem-
0.0.13_64bit.patch
patch -p0 < beebem-0.0.13-
keys.patch
patch -p0 < beebem-0.0.13_
menu_crash.patch

cd beebem-0.0.13
./configure
sudo make install
```

The emulator comes fully loaded with operating system ROMs, and you can access its settings by pressing **F12**.



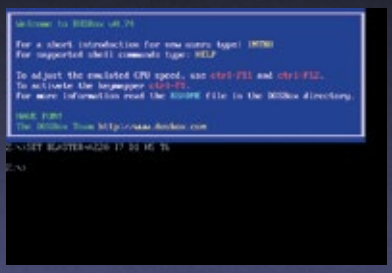
C64: VICE

> **magpi.cc/2Erd120**

VICE, the Versatile Commodore Emulator, is a fantastically authentic emulator that not only provides a spot-on reproduction of the C64 user experience, but also the rest of Commodore's 8-bit computer range, including the VIC-20 and Plus4.

To install it under Raspbian, enter: **sudo apt install vice**.

Once running, you can access its menus by pressing **F12**. A huge range of clearly labelled settings let you do everything from saving programs to connecting your emulated computer to the internet.



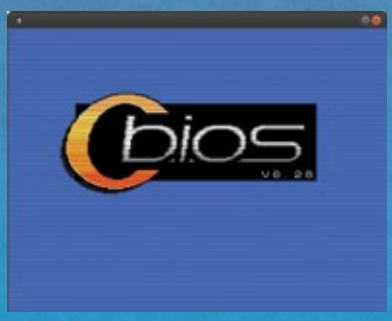
IBM-PC: DOSBOX

> **dosbox.com**

Emulating DOS software on a Raspberry Pi can be a challenge, simply because of the huge range in required specifications between MS-DOS's first release in 1981 to its final iteration in 2000.

DOSBox is available in the Raspbian repositories, so you can **sudo apt install dosbox** and consult the excellent documentation (**magpi.cc/2EmbSJ3**) to get your vintage software mounted and installed.

DOS has the widest imaginable range of software, from DTP programs and office suites to fractal generators and games. However, given the Raspberry Pi's limited processor power, you're best off sticking to software released before the mid-1990s.



MSX: OPENMSX

> **openmsx.org**

The MSX, with its built-in Microsoft eXtended BASIC, had near-arcade quality games and some of the first implementations of the MIDI electronic music standard.

While there are a number of good MSX emulators out there, OpenMSX is regularly updated, extremely faithful, and can be found in the standard Raspbian repository, so you can install it with **sudo apt install openmsx**.

It even supports MIDI via USB adapters and soft synths, so you can use the extensive range of MSX music utilities archived online.

OpenMSX comes with an open-source operating system ROM, but this doesn't include MSX BASIC, so you'll have to hunt down original ROM images for your MSX before you can write your own software.

AMIGA: AMIBERRY

> **magpi.cc/2ErK2ee**

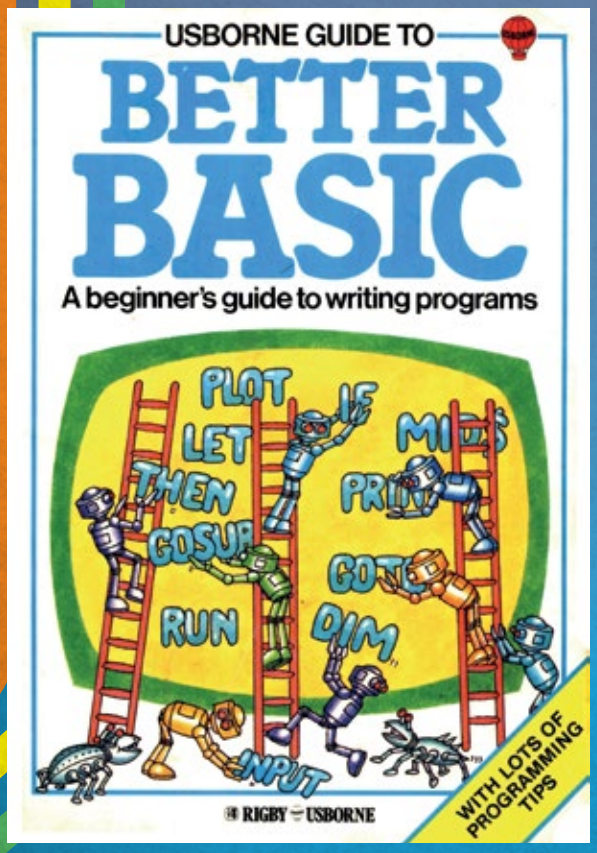
During the late 1980s, the Amiga was known for its huge range of games and audiovisual demos that pushed the limits of what Commodore's systems could achieve.

AmigaOS is still in development, with version 4.1 available to buy for around £30, while older versions of the sort you'll want for emulation are sold by Cloantro (**amigaforever.com**).

Amiberry is optimised to get the best performance out of the Raspberry Pi and is available via RetroPie, as a bootable image, and as source code. If you compile it yourself, follow the First Installation instructions to get all the dependencies it needs in place.

USING A RETRO COMPUTER

DISCOVER ALL THE FUN YOU CAN HAVE WITH AN EMULATED CLASSIC COMPUTER



Above The *Usborne Guide to Better BASIC*, one of the firm's classic computer books now available as a free download

Image Credit: Usborne Publishing

While playing games and writing software are the most obvious uses for your virtual vintage computer, using old-school software provides a unique look and feel to more artistic projects, too.

Back when IBM PCs had nothing more than a buzzing monophonic beeper to communicate with the world, the Atari ST and Commodore Amiga were creating fantastic polyphonic synth tunes.

We've already mentioned the ST's integrated MIDI ports and popularity as a musician's composition, production, and performance tool using software such as Cubase.

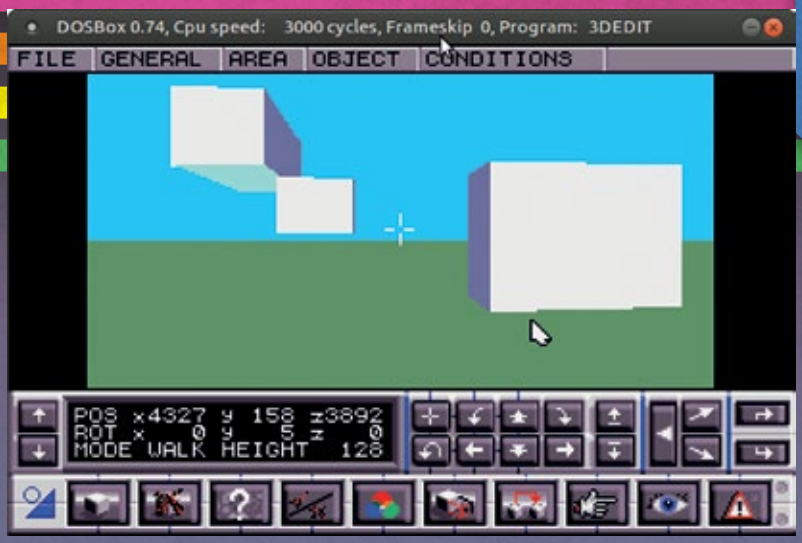
At the same time on the Amiga, MOD files, first created as a native format for the Ultimate SoundTracker audio composition tool, quickly became ubiquitous, and their influence is clearly audible in modern chiptune music. Freeware ProTracker 3.15 is still a popular composition tool.

The Amiga was also the graphical champion of the 16-bit home computer world. Electronic Arts' Deluxe Paint was the art program of choice for game developers and digital artists alike. A powerful bitmap graphics tool, it was eventually ported to MS-DOS, once PC graphics technology caught up, and produced the original pixel art for classic games like The Secret of Monkey Island.

RENDER 3D GRAPHICS

If you fancy turning your hand to ray-traced graphics, POV-Ray – the Persistence of Vision Raytracer – is free and provides old versions for download, all the way back to its 1992 DOS release. You'll have to get to grips with its text-based language for defining the 3D objects you wish to render and your render times on a Raspberry Pi will probably take about as long as they would have on a 386 back in the day.

One of our favourites is 3D Construction Kit II (also known as 3D Virtual Studio) for the Atari, Amiga, MS-DOS, and even some 8-bit systems. Based on the



Above You can use software like 3D Construction Kit II to create 3D worlds and games, and even share them with others

Freescape game engine, it lets you to create and share interactive 3D worlds and games, and feels much like Minecraft's distant ancestor.

HOME OFFICE

You can, of course, run old word processors and office software – WordPerfect 5.1 for DOS provides a surprisingly meditative writing environment. While professional layout and design software from the 1980s and 1990s may be best forgotten, we can wholeheartedly recommend playing with kid-friendly design programs like Springboard's *The Newsroom Pro* and Broderbund's *The Print Shop*. You can even get DOSBox talking to a printer if you want to share your creations with the world.

> "The Amiga was also the graphical champion of the 16-bit home computer world"

The obvious thing to do with an emulated version of a classic is to teach yourself to program the same way thousands of people – including influential developers like Jeff Minter, Peter Molyneux, Anita Sinclair, and Muriel Tramis – got started in the 1980s.

As well as the remarkably informative manuals that came with the computers, a lot of the materials that got developers started back in the day are available for free online.

Usborne still publishes guides to modern programming languages, and has generously made its fantastically illustrated 1980s introductions to computers and programming available for free (scroll down to the bottom of **magpi.cc/2EsGwR6**).

Back in the day, you could learn a lot by typing in the code given away with computer magazines. Plenty of those have been archived online for posterity, including full copies of CVG (**magpi.cc/2EokUoH**), highlights from a range of Sinclair Spectrum mags (**magpi.cc/2ErDnAP**), and Acorn-focused Your Computer (**magpi.cc/2Em9Ybp**).

For later computers, you can find whole development environments, like STOS The Game Creator for the Atari ST and SEUCK (Shoot 'Em Up Construction Kit) for Commodore systems. As well as free homebrew titles, some games made for older and emulated systems even went on to see commercial releases.

TOP FIVE HOMEBREW

A surprising amount of software is still written for old-school computers and emulators, with vibrant homebrew scenes catering to the Spectrum, C64, MSX, and CPC.

SILLY KNIGHT (MS-DOS)

Award-winning, castle-conquering platformer with CGA graphics.

> **magpi.cc/2EbfFJS**

OOZE (ZX SPECTRUM 128K)

Navigate maze-like levels as a blob of gravity-defying goo.

> **magpi.cc/2CgJQti**

TIME OF SILENCE (C64)

This isometric adventure-RPG is a short but lovely exploration of a post-apocalyptic world.

> **magpi.cc/2Emarub**

XIΛƎX (ZEVIMODOKI) (MSX)

Cross hostile terrain and shoot down enemy ships in the vertically scrolling shoot-'em-up.

> **magpi.cc/2E9cVwC**

HORACE AND THE ROBOTS (ZX SPECTRUM)

Escape destructive robots in this strategic arcade blaster with synthesized voice effects

> **magpi.cc/2E7BSZg**

HANDY SOFTWARE

PROTRACKER 3.15

> **magpi.cc/2EsBtji**

MOD FILES

> **modarchive.org**

3D CONSTRUCTION KIT

> **3dconstructionkit.co.uk**

POV-RAY

> **magpi.cc/2EnyFUN**

THE NEWSROOM

> **magpi.cc/2EnnpYC**

THE PRINT SHOP

> **magpi.cc/2EmjxHj**

PUT RASPBERRY PI INSIDE A CLASSIC COMPUTER

TURN A DEFUNCT ZX SPECTRUM CASE AND MEMBRANE KEYBOARD INTO A NEW HOME AND INPUT DEVICE FOR YOUR RASPBERRY PI

YOU'LL NEED

- A ZX Spectrum case with keyboard membrane
- Soldering iron and solder
- Ribbon cable or wire
- Sticky pads and electrical tape to mount hardware
- 2 × Pieces of stripboard large enough to accommodate components and keyboard connectors
- 8 × Diodes (e.g. 1N4148)
- 2 × Molex keyboard membrane connectors (1× 5-pin, 1× 8-pin)
- Push-to-make momentary contact button
- ZX Raspberry Keyboard Scanner **magpi.cc/2EbikTS**

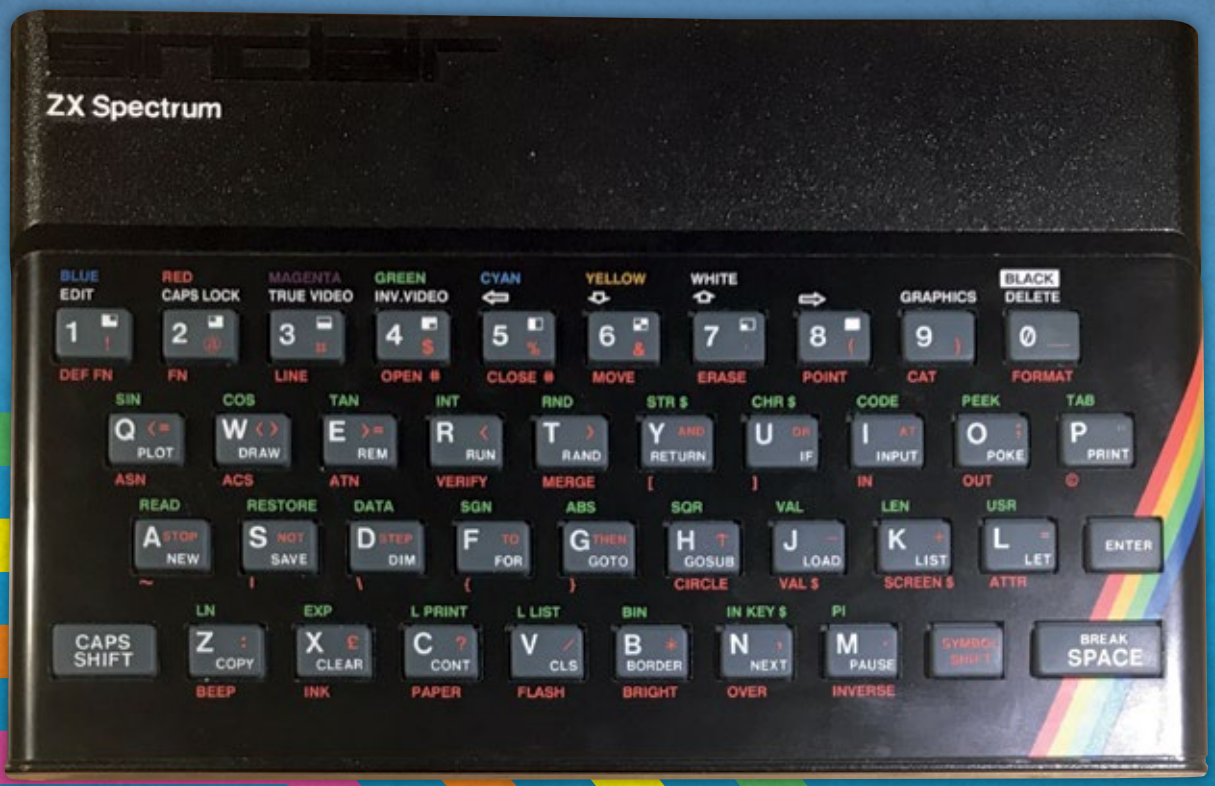
Luckily, many ZX Spectrums have made it to 2018 in working order, but the ones that weren't so fortunate can find new lives as a keyboard and case for the Raspberry Pi.

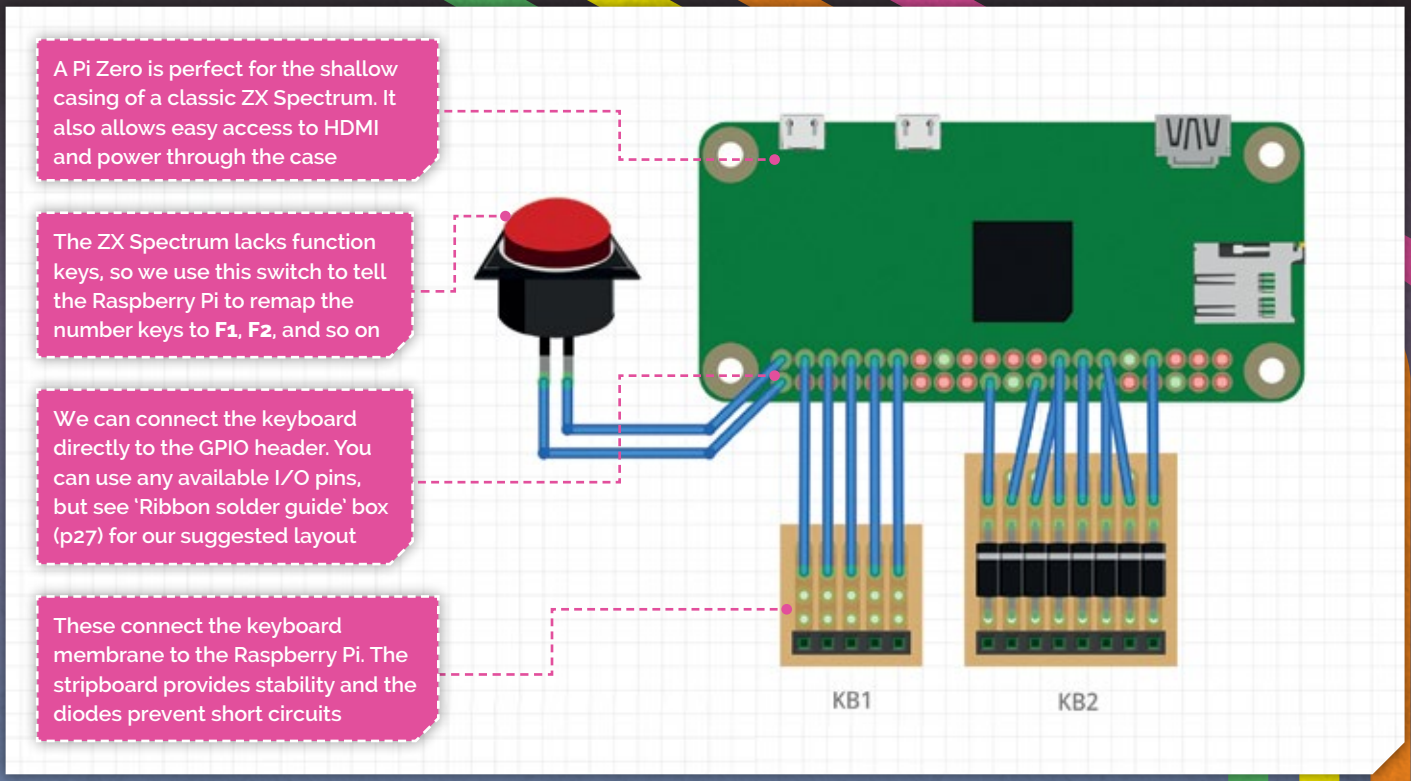
A Raspberry Pi Zero W fits most easily into a Spectrum case. There's space for larger models in the raised upper area of the case, but the wiring gets more complicated.

Always measure carefully, clean everything, double-check all positioning, and make sure you have enough cable to connect each component once everything's in place.

KEYS TO THE CASTLE

The Spectrum keyboard uses a thin plastic membrane sandwich with two layers of electrical tracks which form a matrix





grid of 5×8 tracks. When a key is pressed, it creates a circuit between two of these tracks. Each key 'lights up' a unique pair of tracks.

The tracks lead into two ribbon cables that go from the keyboard assembly to the motherboard.

> "The ribbons are extremely delicate and normally the first thing to fail on a Speccy"

The ribbon connectors consist of five data lines (KB1) and eight address lines (KB2). One wire on each will light up per key press.

The ribbons are extremely delicate and normally the first thing to fail on a Speccy, making keys on the broken line unresponsive. Newly manufactured replacements are available if needed.

The ribbon cables go to two connectors, which we'll need to connect them to the Raspberry Pi's GPIO. You can harvest these from a dead Spectrum motherboard or buy online.

Cut two pieces of stripboard, one for each connector. Carefully work out where to place these in the case, as the ribbons from the membrane must not be put under any stress. Ideally, model their positioning on the Spectrum's original motherboard, using masking tape to mark their position.

Spectrum keyboard connectors are very delicate. If required on an old one, manipulate the contacts carefully so they make a good connection with the ribbon.

Most importantly, make sure you solder them into position the correct way around! Only one side of the ribbon has its contacts exposed and these need to match up with the flat side of each connector. KB1 connects at the bottom, KB2 at the top.

To support multiple key presses, we need to ensure that the current from the address lines doesn't short-circuit, so a diode is used to protect each line. When soldering the stripboards, make sure you cut the tracks between the resistors and the diode.

FUNCTION KEYS

You'll also notice another component: a button. The Spectrum keyboard lacks function keys, but you're going to need them to control the Raspberry Pi.

The Keyboard Scanner utility includes an alternative keyboard

PJ EVANS

When not volunteering at The National Museum of Computing or running the Milton Keynes Raspberry Jam, PJ can be found installing Raspberry Pi devices where no-one asked them to be.

mrpjevans.com

CODE

LANGUAGE:
Unix

CODE FILE NAME:
startzxscanner.service

GITHUB LINK:
magpi.cc/2EbikTS

mapping mode. When it detects a press of our switch, it toggles between 'normal' mode and a second mode that converts the number keys into their function key equivalent.

TIME SCANNERS

To get your Raspberry Pi working with the membrane keyboard input, we use a Python script to scan the keyboard for presses and inject them into the kernel. The script outputs a current through all address lines (KB2) and drops the current to each one in turn about 60 times a second.

If a key is pressed, we can detect which data line it's on, and we know which address line was being checked, so we know which key was pressed. The result is passed to a piece of software called uinput, which turns it into a key press.

The ZX Raspberry Keyboard Scanner script is designed to work with RetroPie (see page 18) and the FUSE Spectrum emulator, but you'll find instructions for other distros on its home page.

We can use Git to pull down the script and its dependencies. First, let's make sure we have the essentials to install everything:

```
sudo apt install get libudev-dev python-dev python-pip
sudo pip install wiringpi
```

Now install Libsuinput:

```
cd
git clone github.com/tuomasjjrasanen/libsuinput
cd libsuinput
./autogen
./configure
make
sudo make install
```

Make sure uinput is loaded every time the Raspberry Pi boots. Add the following line to **/etc/modules-load.d/modules.conf**:

```
uinput
```

You can load it right away without rebooting by typing:

```
sudo modprobe uinput
```

And install Python-uinput:

```
cd
git clone github.com/tuomasjjrasanen/python-uinput
cd python-uinput
sudo python setup.py build
sudo python setup.py install
```

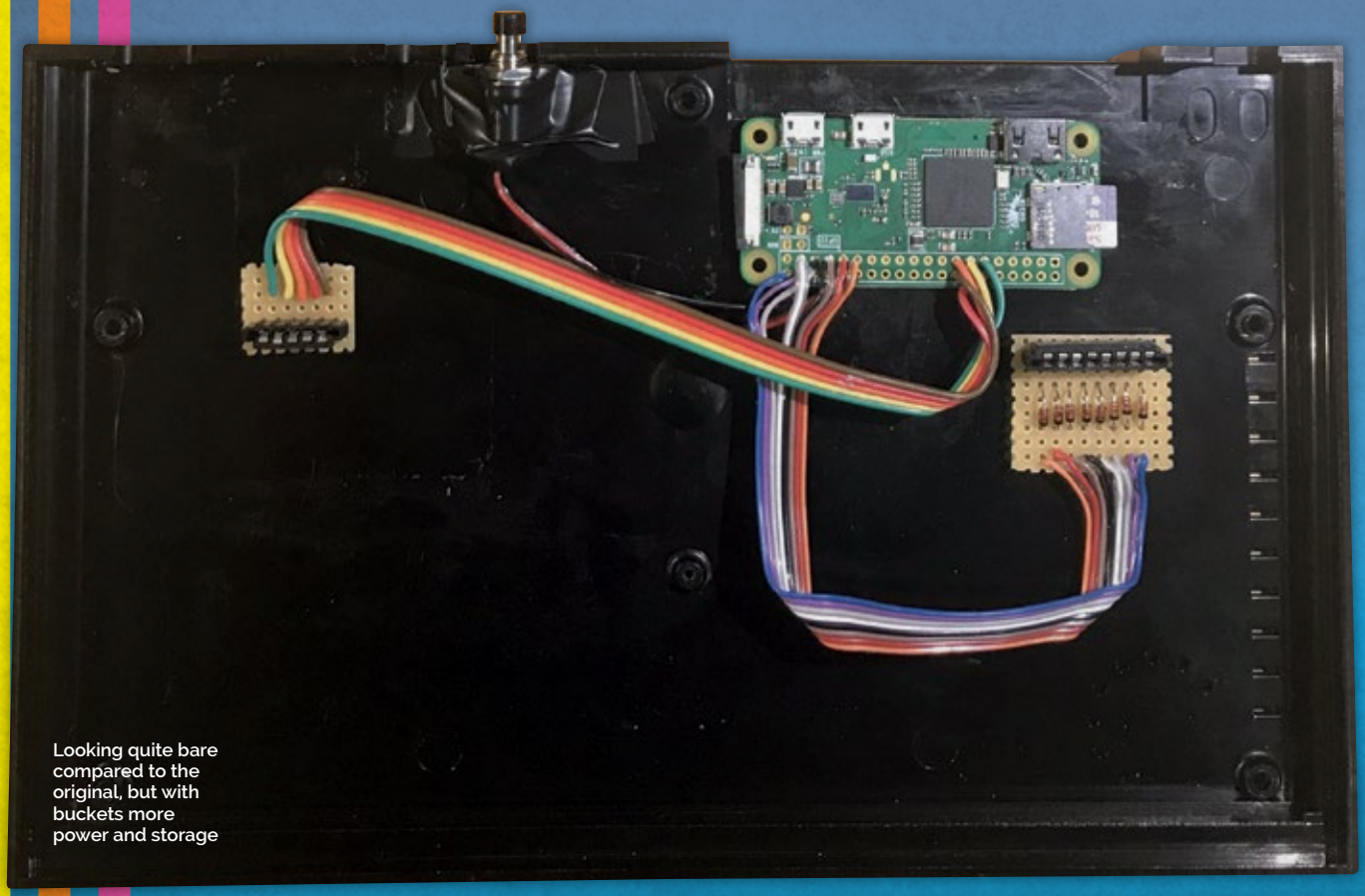
Download the scanner script:

```
cd
git clone github.com/mrpjevans/zxscanner
```

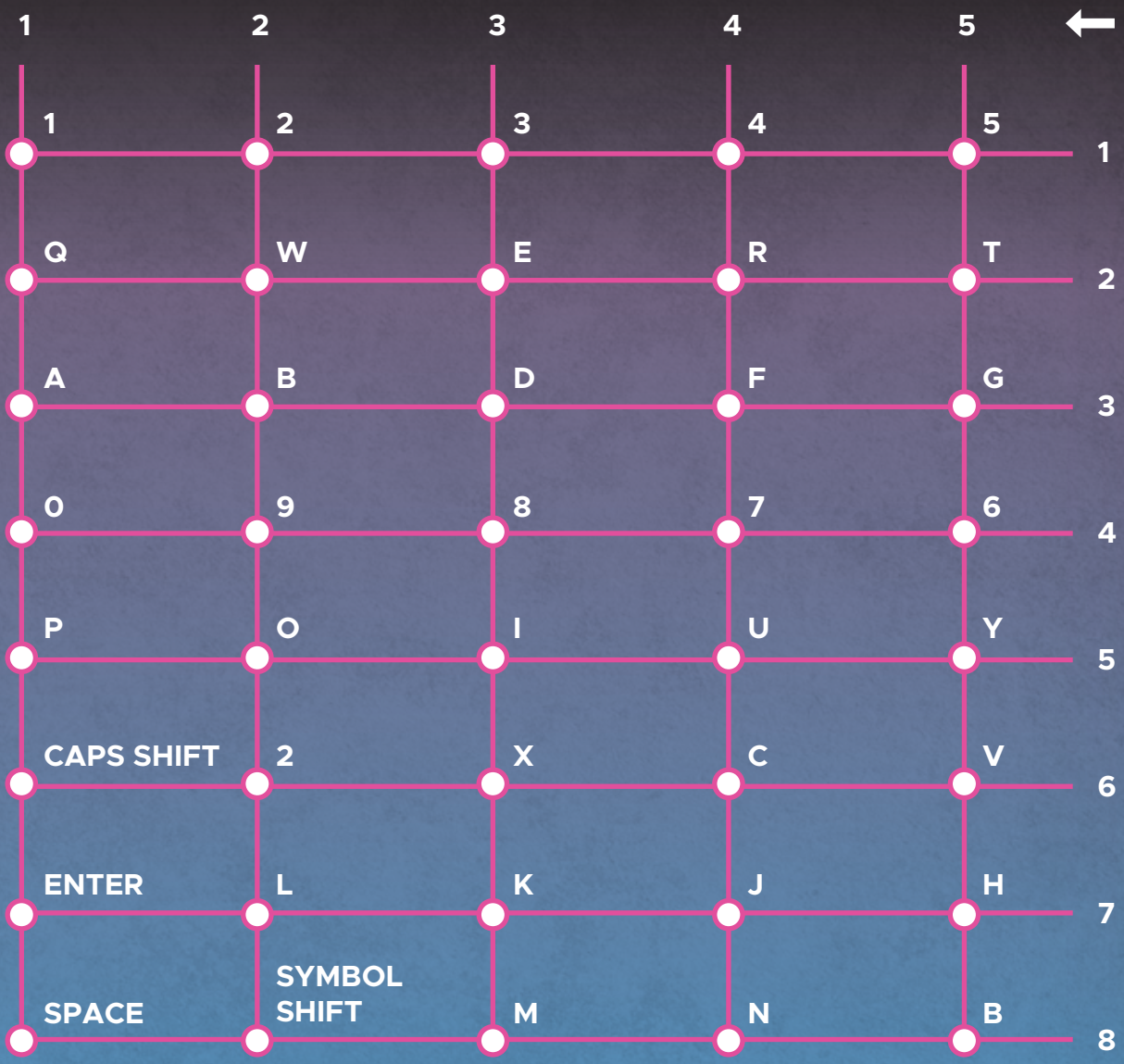
And test it. Carefully connect your ZX Spectrum membrane to the Molex connectors and run the following from the project directory (ideally from an SSH session on another computer):

```
cd ~/zxscanner
sudo python zxscanner.py
```

Now try the keyboard. Each key press should produce a letter



Looking quite bare compared to the original, but with buckets more power and storage



Left The Speccy's keyboard matrix allows you to detect inputs for 40 keys using just 13 lines. Perfect for the GPIO

on screen. The SSH session will show debug output. Check the switch too.

The script needs to run at boot. Start by making the file executable:

```
sudo chmod +x ~/zxscanner/zxscanner.py
```

Then create a service file and enter code from **startzxscanner.service**. Our **zxscanner.service** file needs to contain the following so the OS knows how to start the scanner:

```
sudo nano /usr/lib/systemd/zxscanner.service
```

Finally, we enable the service so the scanner is always running:

```
sudo systemctl enable /usr/lib/systemd/zxscanner.service
sudo systemctl start zxscanner.service
sudo systemctl daemon-reload
```

When used with RetroPie and FUSE, a tap on the button will cleanly close FUSE, although you'll have to configure the emulator not to prompt for confirmation. A 3-second press will switch the keyboard so keys 1–4 become F keys (so you can get to FUSE's menu) and 5–8 act as cursor keys.

A world of ZX Spectrum gaming and programming now awaits you. You can, of course, use your new ZX Raspberry to emulate other computers... even a Commodore 64.

RIBBON SOLDER GUIDE

Solder a ribbon cable from each connector to the GPIO as follows:

BROADCOM GPIO NUMBER	CONNECTOR	LINE
26	KB1	1
19	KB1	2
13	KB1	3
6	KB1	4
5	KB1	5
25	KB2	1
24	KB2	2
23	KB2	3
22	KB2	4
27	KB2	5
18	KB2	6
17	KB2	7
4	KB2	8
21	Switch	
GND	Switch	

CLASSIC COMMUNITY BUILDS

THE GREATEST RETRO COMPUTER PROJECTS BUILT USING RASPBERRY PI

A1200 BLUE

Dom Cresswell of Facebook's RetroPie Makers has built a Raspberry Pi-based Amiga emulator build running the Amiberry emulator inside a new A1200 reproduction case.

"The Raspberry Pi really inspired me, and since I had already ordered two new A1200 cases from the **A1200.net** Kickstarter campaign, the blue one seemed perfect for a Raspberry Pi 3, and the colour matches our living room, so when my wife said she was happy to let me start using it as our media centre, I really wanted it to be the best it could be.

"I am still waiting for a set of black keys to be produced, but I'm very happy with the rest. The keyboard works, and is controlled by the excellent KeyRah v2 USB device and the lights are Conny Larson's Bifrost replacement A1200 LEDs, although I had to do a bit of work getting them to work from the GPIO control voltage.

"Probably the part I am proudest of, though, is the back panel – all done without any changes to the original case; I bought some blue sheet plastic and did a lot of careful Dremel cutting to make some panels which could just be hot-glued in place, complete with the ports... the finish looks really great and much tidier than I had expected!"

COMMODORE PLUS4

When both his beloved Commodore Plus4 computers stopped work, Ian Hill turned to his Raspberry Pi to bring them back to life.

"The Plus4 was my first real computer, but they tend to overheat and burn out some of the chips inside, which are not available any more, meaning once they're dead, they stay dead.

"I have two dead Plus4s, so I replaced the internals of one of them with a Pi Zero, Keyrah V2, and Combian64 software. The result is a Commodore Plus4 that very closely resembles the original, and can switch between Plus4, C64, and VIC-20 modes, with all the available software for all three machines built in."

THE THROWBACK: FUZE

For those who want a retro-styled keyboard and chassis, but aren't up for the soldering involved, the FUZE (not to be confused with the FUSE Spectrum emulator) is a perfect compromise.

Modelled after the classic BBC Micro B, the FUZE Special Edition available from **fuze.co.uk** with a built-in Raspberry Pi 3 and a heap of electronics project accessories for £169.99 or as a stand-alone case for £90.

The FUZE BASIC programming language, an updated version of BBC BASIC, can be downloaded on any Pi and is a fantastic way to bring your retro programming knowledge into the modern day.

THE CASE MAKER

While some Raspberry Pi fans have converted favourite computer chassis into cases and keyboards, Stephen Eddy of **RetroPiCases.com** started remaking his favourites using a 3D printer and turned his passion into a vibrant small business.

"Each case model takes me around a month to design and produce. Initially I look at the source computer models, then look for design cues that I can incorporate into my own models. Because the Raspberry Pi is completely different dimensions to the original computers and the position of the ports are fixed, I use the design cues to make the model recognisable as a computer from the original family."

USE YOUR ORIGINAL CONTROLLERS

This is actually the reborn Competition PRO Retro made by Individual Components (**icomp.de**), but you can do the same project with any old joystick or game controller. It seems like sacrilege to cut the end off a functional controller, so your best bet is to order some connector ports (or pull some out of a dead computer or console).

There are some great dedicated connector boards to hook your controller into the Raspberry Pi's GPIO. An alternative is to use serial cable and some breadboard to connect it and feed into a driver module created by the RetroPie team.

COMMUNITY RESOURCES

RETROPIE MAKERS
> magpi.cc/2EqwwaE

KEYRAH USB KEYBOARD DEVICE
> magpi.cc/2ErTz50

BIFROST LEDS
> magpi.cc/2EnXXCa

IAN HILL'S COMMODORE PLUS 4 BUILD
> magpi.cc/2EpoJdj

RETROPIE CASES
> retropicases.com

SUBSCRIBE TODAY FROM JUST £4

SAVE UP TO 35%

NO OBLIGATION!
LEAVE ANY TIME*

Pricing

Rolling Subscription

- £4 a month
- Quick and easy to set up
- No long-term commitment

* Leave any time applies to Rolling Subscription only

Subscribe for a year:

£55 (UK)

£80 (EU)

$90 (USA)

£95 (Rest of World)

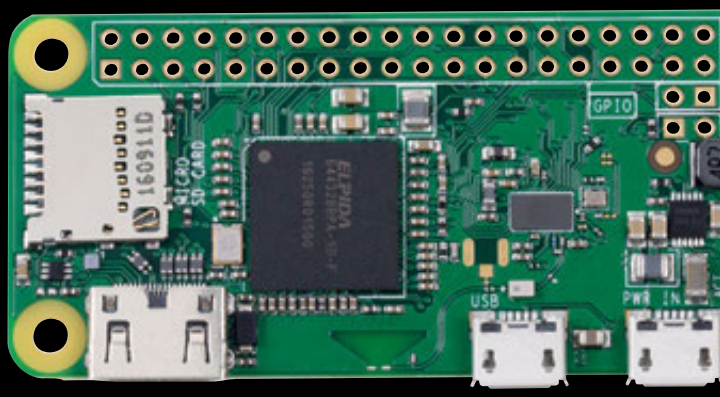
Subscription benefits:

- Free delivery to your door
- Exclusive Raspberry Pi offers and discounts
- Get every issue first (even before stores)

magpi.cc/subscribe

JOIN FOR 12 MONTHS AND GET A

FREE PI ZERO W



WORTH £20

Subscribe in print for 12 months today and you'll receive:

- **FREE** Pi Zero W
- **FREE** Pi Zero W case with three covers
- **FREE** USB and HDMI converter cables
- **FREE** Camera Module connector

SUBSCRIBE ON APP STORES

Available on the App Store

GET IT ON Google Play

FROM £2.29

raspberrypi.org/magpi

JOSH STARNES

Josh considers himself to be an amateur robot builder and his latest creation was the first time he had used a Raspberry Pi.
magpi.cc/2DNur9h

Quick Facts

- Josh wanted the robot to cost under $1000
- It's IR sensitive to record in the dark
- Four Raspberry Pi 3s are used
- It includes an anti-theft GPS 303D tracker
- Josh made extensive use of EZ-Robot's EZ-Builder v4

WATCHDOG SECURITY ROBOT

Big Brother on wheels or the potential future of law enforcement? **Josh Starnes**'s robot security guard is an arresting sight, as **David Crookes** discovers

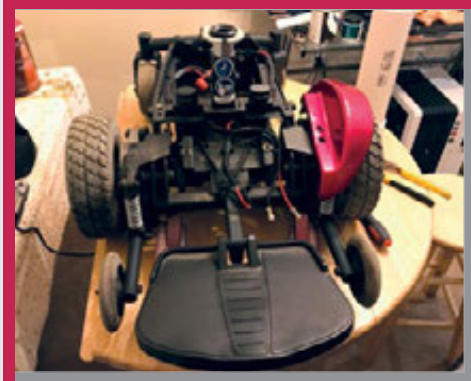
Although our city streets are packed with CCTV cameras, past reports have suggested that many are either in the wrong place or simply don't work.

"Someone can throw their hoodie on and hide from a high-mounted camera," argues Josh Starnes. "People are less likely to misbehave when someone is physically there to witness it."

So what is the answer? A lack of resources means flooding the streets with police isn't always economically viable. "But would you be surprised if a robot rolled up, recording or talking to you while you are up to no good?" asks Josh. "I would." Which is why he has created a security robot to do exactly that, called Watchdog.

Inspired by sci-fi movies such as *Short Circuit*, *Evolver*, and *Star Wars*, Josh got to work on an affordable DIY robot that can trundle around the streets, operate in cold and wet weather, work well in low light, and communicate. He wanted it to record HD video locally and stream via WiFi to a secondary source for online viewing. "I decided to use a Raspberry Pi. It has an excellent

BRINGING WATCHDOG TO LIFE



>STEP-01
Preparing the base
Josh tore into the Jazzy Select, pulling parts out of the power chair's chassis, including a non-working old circuit breaker. A water-resistant compartment was created in the emptied battery bay.

>STEP-02
Inserting the Pi boards
Four Raspberry Pi 3s connect to the four cameras and are set up to record at 800×640 resolution and 15 frames per second. Files are saved 600 seconds at a time to SanDisk 64GB cards.

>STEP-03
Wiring it up
A compartment in the front of the Jazzy chassis houses wires and the Sabertooth dual 12 A motor driver. New power wires and quick-disconnect plugs are installed, making it easy to swap batteries.

community and it's moved past being a new gadget to a useful tool."

Pi spy

At first, Josh spent time getting the chassis right, using the base of a Jazzy Select power chair. He stripped away unwanted parts such as the chair, footrest, original Jazzy electronics, and the front anti-tip wheels. He also bought a Sabertooth dual 12 A motor driver which is optimised for use in medium-powered robots and radio-controlled vehicles.

Struggling to find an off-the-shelf camera that met all his needs, he turned to the Pi 3: "It meant I could pick the camera module I needed, record to a hard drive, and have remote access." Josh used four Pi 3 boards, each connected to a camera. "In the event one fails, all the others are independent and would continue recording."

The project uses the Pi NoIR Camera Module, which has a 5MP sensor, night vision, and a 130° adjustable-focus lens. Four of these are mounted inside the robot's head – the upper half of a five-gallon bucket with a screw-on lid and a treated plywood bottom. Motion and ultrasonic sensors are also mounted within the head. These enable the robot to detect obstacles and movement.

> "The Raspberry Pi meant I could pick the camera module I needed"

Easy does it

With the Raspberry Pi handling the video recording and managing the broadcast, Josh was able to produce a robot that could scan its environment, roll up to wrongdoers, and record. "I used EZ-Robot's EZ-Builder software. They have some wonderful tools that allow use of radar-style ultrasonic navigation, machine vision, and movement controls." He found creating a movement routine to be straightforward: "EZ-Builder has modules you can insert that write complicated scripts for you."

This gave Josh more time to get the Raspberry Pi to record while broadcasting. After trying the motionEyeOS system but only achieving 10 frames per second, he turned to Python and wrote something himself. Meanwhile, he continued to work on the look of Watchdog, painting the robot a cool gunmetal grey. Today, while it's still a work in progress, it's almost ready for action.

Above top Josh cut off the upper five inches of a five-gallon bucket, using a stepper bit to get the right-sized holes for the cameras and sensors

Above Josh picked up this Jazzy Select on Craigslist and used parts from other Jazzys to made the perfect chassis

MARC RENÉ SCHÄDLER

As a long-time GNU/Linux enthusiast and hearing researcher, Marc was eager to have openMHA running on the Raspberry Pi.
magpi.cc/2DPWc1j

HEARING AID RESEARCH

Quick Facts

- openMHA can process audio signals in real-time
- It can be adapted to an individual's hearing loss
- Delays between input and output audio is below 10 ms
- There's no GUI – except for fitting the amp parts
- Funding came from the US National Institutes of Health

When looking for a way of testing hearing aid algorithms outside of the lab, a team of researchers realised a Raspberry Pi could be a sound investment. **David Crookes** listens to their idea

For millions of people around the world, hearing aids are hugely beneficial. Not only do they allow people to hear better, they have been shown to lower the risk of dementia, the potential for loneliness, and the likelihood of people withdrawing from social situations.

But while a lot of research has gone into developing hearing aids over the years, Tobias Herzke, a signal processing engineer at HörTech in Oldenburg, Germany, says: "There is still a lot of potential for improvement, especially in acoustically difficult situations."

For that reason, the company – a spin-off of the University of Oldenburg – developed openMHA, which was designed to be a common, portable software platform for hearing aid research and teaching. "The openMHA platform allows for real-time audio signal processing with low delay," explains Hendrik Kayser, who develops signal-processing algorithms for digital hearing devices.

By providing a set of standard algorithms to form a complete hearing aid, openMHA can process the signal from a live microphone and perform different tasks such as amplification, directional filtering, noise reduction, and feedback suppression. Testing new algorithms is not always straightforward, however, which is where the Raspberry Pi can come into its own.

Above Hendrik Kayser (left) and Tobias Herzke (right)

Testing times

"The openMHA software can execute on Linux computers in a laboratory environment, but the sound environment in a lab will always differ from real sound environments encountered by

hearing aid users in real life," Tobias tells us. In the past, this has often led to skewed results that do not offer a true reflection of how hearing aids are used. Yet using ARM-based devices such as the Raspberry Pi offers a wonderful solution.

By running openMHA on a Raspberry Pi and taking advantage of its portable nature, the researchers can evaluate new algorithms in real-time and in realistic outdoor conditions. "It has allowed us to implement a new algorithm on a mobile device in order to find out how it sounds in real-time when running around with a hearing aid," says Tobias. As such, it gives research in this field a real boost by providing better quality and more relevant data to work with.

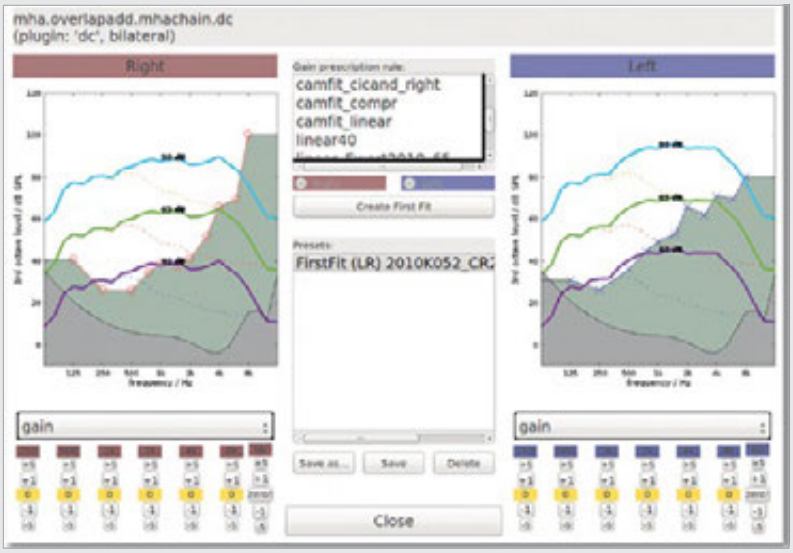
> "We have lowered the entry barrier for hearing aid algorithm development"

Loud and clear

It certainly beats lugging a Linux laptop around and it has also enabled the researchers to cut costs. "Small systems like the Pi provide decent computing capabilities at lower power consumption and small size," adds Hendrik, who says the Raspberry Pi has the best price-to-performance ratio and loves the bonus of on-board wireless. "They have allowed us to offer a complete, working hearing aid signal-processing chain and keep the cost below $300."

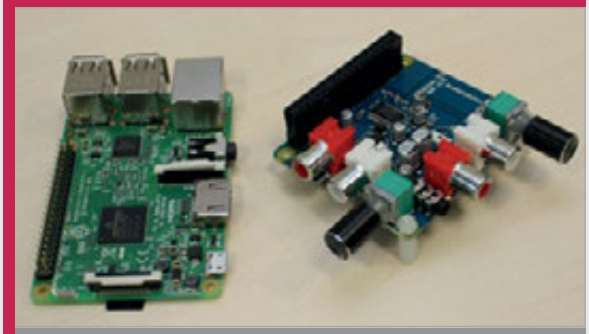
The knock-on effect of this has been significant. "By using the Pi, we have lowered the entry barrier for hearing aid algorithm development and evaluation, for students at university and amateur hearing enthusiasts who lack the resources to invest in large-scale setups," says Marc, who is leading a hearing search group at the university. What's more, by using openMHA for the software, Raspberry Pis, and commercially available headsets and sound cards, the team is able to ensure the system is stable and flexible.

"Low cost, high availability through off-the-shelf components, mobility, no proprietary solutions, community support, and easy access to hearing aid software were the most important driving points here," concludes Marc.



Below It's necessary to adapt the amplification and compression of the 'hearing aid' to the individual's hearing loss. This is done by plugging a monitor into the Raspberry Pi and firing up the Fitting GUI

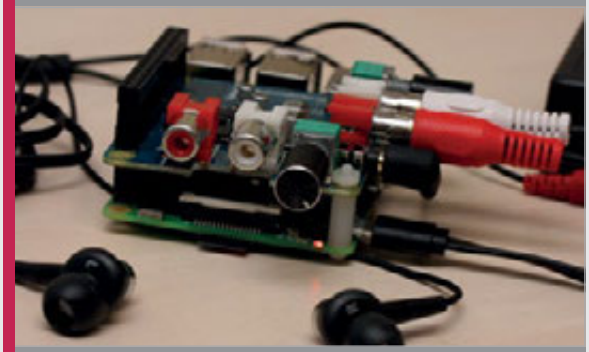
HEAR THAT SOUND



>STEP-01

Grab a Raspberry Pi

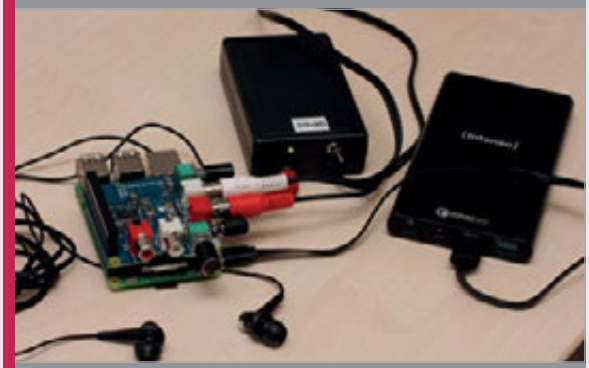
Originally, portable testing of the openMHA software platform was carried on on a Linux laptop. Marc René Schädler, leading a hearing research group at University of Oldenburg, felt a Raspberry Pi would work better.



>STEP-02

Add a sound card

Marc took a Raspberry Pi 3, loaded the Raspbian Stretch OS on to a 32GB microSD card, snapped up an Audio Injector Stereo sound card, and cloned the openMHA source code.



>STEP-03

Calibrate and set up

According to Marc, calibrating the setup before using it is vital. "Hearing aid algorithms need to know the absolute physical sound of the input," he says. This has been worked into the project.

GAWAIN HEWITT

Gawain Hewitt is a musician, technologist, and sound artist who works in the areas where technology and art meet.
gawainhewitt.co.uk

STEWART EASTON

Stewart Francis Easton is a multidisciplinary artist specialising in stitch, drawing, paint, and sound.
stewarteaston.net

THE NEXT VERSE

A rare fusion of handicraft and the Raspberry Pi has been displayed in the world-famous V&A Museum, alongside works by Raphael. **Sean McManus** reports

Quick Facts

- The title comes from a Philip Larkin poem
- The recurring bird in the canvas symbolises spirit
- The original drawing was a 3 metre panorama
- The Touch Board costs £60 ($85) from Bare Conductive
- Bare Conductive also makes conductive paint

Wandering through London's V&A Museum in December, you had a chance to see works from two major cultural movements: the Renaissance, and the maker movement. On the wall were paintings from 1515, designs by Raphael for his celebrated tapestries. On an easel was a modern work by Stewart Francis Easton that combines embroidery, music, and the Raspberry Pi 2. "I shared a room with Raphael. Just me and Raphael. That's the dream," Stewart says.

His work is called *The Next Verse*, and was part of a digital event at the gallery. It depicts the cycle of life, using scenes of a family along a snaking path through the canvas. "From childhood to death and round again," explains Stewart. "You begin with a father and child and end with that child's son releasing the spirit of his father to enable the cycle to begin once again."

The work is completed by a soundtrack. There is a 23-minute composition by Michael Tanner, performed on a psaltery, a stringed instrument often seen in medieval paintings. Additionally, viewers can trigger sounds. "Each

Above Viewers touch the conductive thread of the triangles in the canvas to trigger sounds relevant to each scene

illustrated event has a small triangle stitched next to it," reveals Stewart. "The viewer is required to touch these triangles (one at a time) to trigger the sound that corresponds to the storyline."

Ambient sounds

Gawain Hewitt created eight sounds for these points in the journey. "One of the sounds has a bellows playing – almost like a broken accordion with static sounds and whistling," says Stewart. Another has a recording of sounds made at his daughter's school playground. These sounds were all required to be able to be played along with the original soundtrack which plays on a loop.

Gawain also carried out the technical implementation, which is based on the Touch Board, made by Bare Conductive. Although the Touch Board can be used to trigger MP3s or play MIDI sounds itself, it can only play one audio file at a time. Using the Raspberry Pi enabled Gawain to code polyphonic software, and use high-quality uncompressed audio files.

Gawain wrote the software using Pure Data (Pd), an open-source visual programming language for multimedia. "One of the really beautiful things about Pd for me is the ability to code in an exploratory way," he says, "so that the act of coding software is a creative act. I have a background in audio engineering, so I find the interface and 'metaphor' used very empowering and intuitive."

Conductive thread

The triangles on the canvas use conductive thread to detect touch. So, what limitations does it present as a material? "From a purely aesthetic viewpoint it's the colour of the conductive thread," says Stewart. "I think you can only get either grey/silver or gold."

Gawain adds: "You do need to consider its inherent fragility, and try to avoid crossed lines in the wiring for better capacitive touch. Also, it can get dirty when touched!"

For the next verse of their creative collaboration, Stewart and Gawain have several works in progress, including some that might use lights triggered by sound running through a Raspberry Pi.

> "I shared a room with Raphael. Just me and Raphael. That's the dream"

Above Because viewers use headphones, interacting with the artwork is a highly personal experience

COMPOSING THE NEXT VERSE

>STEP-01
Adding the thread

Stewart stitched triangles alongside scenes in the canvas, using conductive thread. The triangles are connected on the back of the canvas with a wire loop that goes to the Touch Board.

>STEP-02
Connecting the Touch Board

The wires from the canvas are soldered to the electrodes along the edge of the Touch Board. The Touch Board can then be connected over USB to the Raspberry Pi.

>STEP-03
Using the Raspberry Pi

The Raspberry Pi uses software written using Pure Data to trigger the spot effects and loop the music. A Terratec USB sound card feeds audio into a headphone amplifier.

DANIEL DAVIS

Daniel is a retro-modern perpetual beta tester who is an IT specialist by day and a YouTuber by night. He lives to learn and loves to tinker.
tinknernut.com

Quick Facts

- The lunchbox measures 17 × 12 × 6.3cm
- A 5200 mAh power bank supplies the juice
- It'll run for about an hour per charge
- Build details can be found at **magpi.cc/2nwQtlE**
- Daniel is working on four new Pi projects

MINI LUNCHBOX ARCADE

Never mind a packed lunch, have a Pac-Man lunch with this mini retro arcade in a box. By **Phil King**

Having already built a full-size tabletop-style arcade cabinet powered by a Raspberry Pi, Daniel Davis wanted to make something a little more portable to feed his passion for retro gaming. So when he acquired an old metal lunchbox, the last thing on his mind was using it to store sandwiches. Instead, he opted to pack it with a retro arcade system built using a Raspberry Pi 3, Adafruit Backpack 5-inch HDMI touchscreen, Adafruit Arcade Bonnet, speaker, plus an analogue joystick and mini arcade buttons.

"The challenge was sourcing arcade buttons and components that would fit," says Daniel. "For instance, standard-sized arcade buttons were too large to fit inside. I also had to take into consideration the depth of the screen so that when the lid closed, it didn't smash into any of the other components."

Neat and tidy

Rather than simply cramming components into the box, Daniel wanted to make the system organised and easy to use, "so that anyone could open it up and be familiar with the control layout (which mimics that of an NES controller or Game Boy)."

To keep everything looking secure, and hide the wiring, Daniel 3D-printed a screen holder and controller board that he'd designed in a CAD program. "Once I had all the components laid out, I did a couple of test prints to make sure everything fit, but then I realised that I hadn't designed anywhere for the wires to go (especially for the screen). So I had to go back and redesign the casing to allow room for the various wiring."

The system runs the RetroPie retro gaming OS on a Raspberry Pi 3. A software fix was required to ensure the interface filled the screen

Daniel recalls that a lot of time was spent measuring, testing, and measuring again. "I went through several different types of arcade buttons before I found ones that did the job and fit in such a small space."

Ensuring that the control panel was low enough to allow enough room for the screen when the box was closed was another crucial factor: "If the lunchbox couldn't close properly, it defeated the whole purpose of the project."

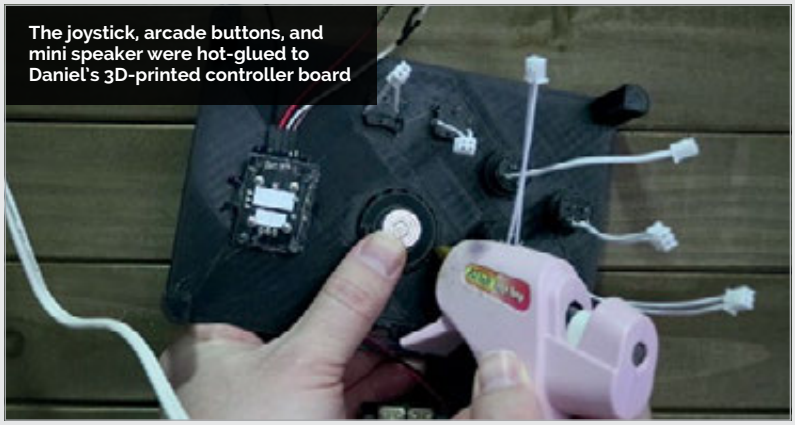
> "I went through several different types of arcade buttons before I found ones that did the job"

Arcade experience

Daniel tells us that his previous experience of building an arcade cabinet in 2013 helped when approaching this much smaller project. Since then, the build process has become a lot easier, thanks to the availability of various Raspberry Pi HATs and bonnets which "made everything a breeze to connect. Aside from that, the available software for the Pi to make arcade emulation possible is also something that has become a lot better over the years."

Daniel spent about 20 hours, spread over several weeks, to complete the build. The end result is a cute and compact arcade system that's simple to pick up and play (**youtu.be/h8nhqowESKg**).

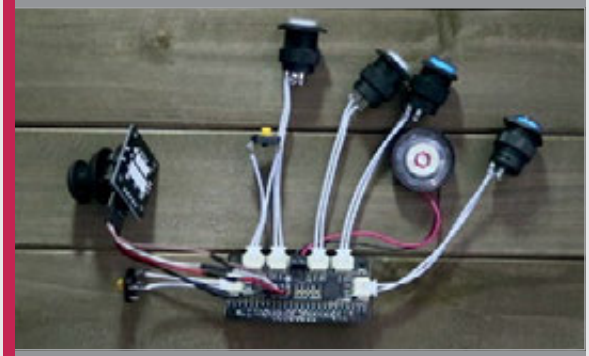
Upon being shown the Mini Lunchbox Arcade, Daniel's nephew was "really shocked to see what was inside. He was able to immediately figure out how to turn it on and start playing a game. He did say that if I could find a way to sell them, all of his friends would buy one and I'd make a million dollars. That's quite the endorsement!"


The joystick, arcade buttons, and mini speaker were hot-glued to Daniel's 3D-printed controller board

FILLING A LUNCHBOX

>STEP-01
Lining the box

One obvious problem with using a metal box is its conductivity, so to avoid any possibility of short circuits, Daniel lined it with Kaplon insulating tape before adding the electronics.



>STEP-02
Arcade controls

The analogue thumb joystick, six buttons, and 0.25 W 8 Ω speaker were connected to an Adafruit Arcade Bonnet. A couple of freely available scripts were then run to get everything working.

>STEP-03
3D-printed parts

After careful measuring, Daniel designed and 3D-printed a screen holder, which he hot-glued to the lunchbox lid, and a board to house the arcade controls and speaker.

WOLFRAM MATHEMATICA

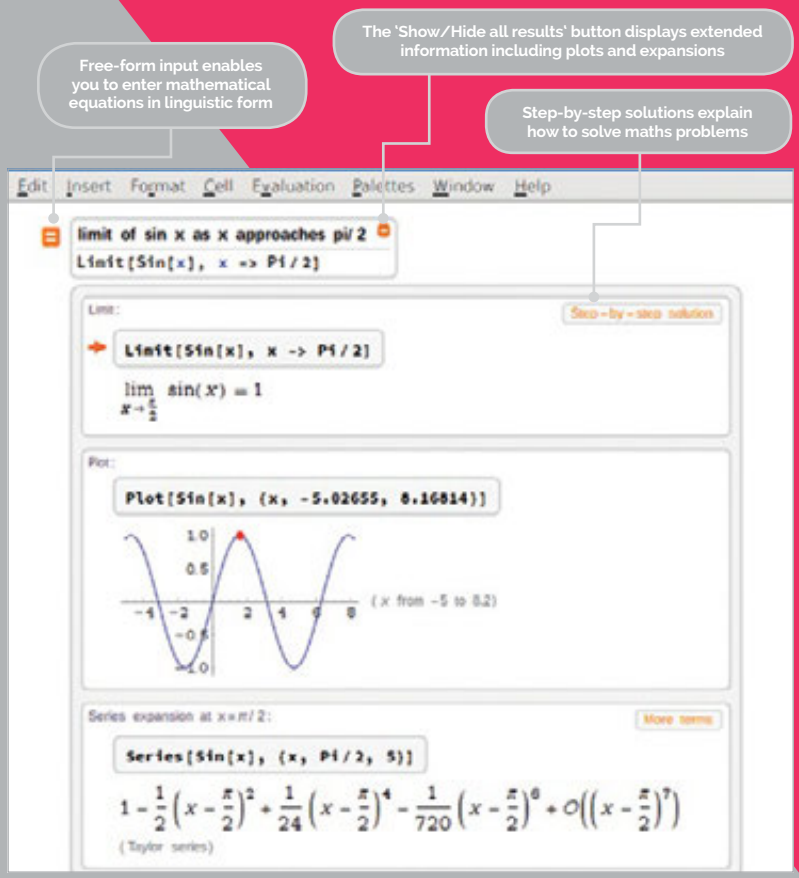
Perform advanced mathematics using the powerful program installed with Raspbian

You'll Need

- Raspberry Pi
- Raspbian desktop

Many professionals who work with numbers pay hundreds of pounds to purchase Mathematica. It's an extremely powerful mathematics and data processing program. Normally it costs hundreds of pounds to buy a licence, but Mathematica is free on the Raspberry Pi.

Back in issue 67 we looked at Wolfram Language. This programming syntax can be used in Mathematica, where its graphical interface provides detailed graphs, plots, and further information.



Anybody who is learning maths or works with numbers will value Mathematica.

Click on the Mathematica icon in the Task Bar to start the program, or choose Menu > Programming > Mathematica.

The default entry is Wolfram Language but unlike the command-line environment, you run lines using **SHIFT+ENTER**. Type out:

```
2+2
```

And press **SHIFT+ENTER** to get the result.

```
In[1]:= 2+ 2
Out[1]:= 4
```

To the right of the screen next to each are vertical lines knows as 'cell brackets'. There is one for the input and output, and a larger one grouping both the input and output. You can highlight, cut, and copy cells with these.

Getting rational

In Mathematica you can enter fractions by pressing **CTRL+/**. This splits the input into a numerator and denominator (indicated by empty boxes). Use your arrow keys and **SHIFT** to highlight the boxes and enter the amounts, such as:

```
In[n]:=  2   3
         - * -
         4   5

         3
Out[n]=  -
         10
```

There are lots of other techniques for entering math symbols. Such as **CTRL+2** to create and insert a square

root, and pressing **ESC**, **P**, **ESC** to enter the symbol (π) for pi.

Rather than memorise them all, it's better to open Palettes > Basic Math Assistant. Here you'll see an advanced calculator-style window, with buttons for math symbols and functions.

Free-form input

One of the more interesting advantages Mathematica has over Wolfram Language in the command line is free-form Input. This enables you to enter equations in plain English and get expanded information.

Press the equals key (=) once at the start of a new cell entry to switch to free-form input. Notice that the line now starts with an orange equals sign.

Before we go any further, press the = key a second time and the cursor will change again to an equals sign in a starburst shape. This is Wolfram|Alpha query, which we'll come to in a bit. Press **BACK SPACE** to get back to free-form input.

In free-form input, you can enter regular expressions such as:

```
2+2
```

As well as the result (4), you'll see a box appear around the input.

Look closely at the box and you'll see a small plus (+) symbol. Hover the mouse to see 'Show all results' in a pop-up balloon.

Click this and you'll see detailed information, such as the result, number line, manipulatives illustration, and even typical human computation times.

Now click on 'Step-by-step solution' and you'll get a detailed breakdown, including a number line (which you can change to 'math manipulatives'). It might seem silly for 2+2, but try it with something complex like these polynomials:

```
= (5*a^2-6*a-4) - (-7*a^2+3*a-9)
```

Now click 'Show all results' and 'Step-by-step solution' and you'll get a full breakdown of the intermediate steps. Handy if you're learning mathematics and want to check your own steps.

Talking maths

Free-form input enables you to enter commands with the formality of Wolfram Language. Enter:

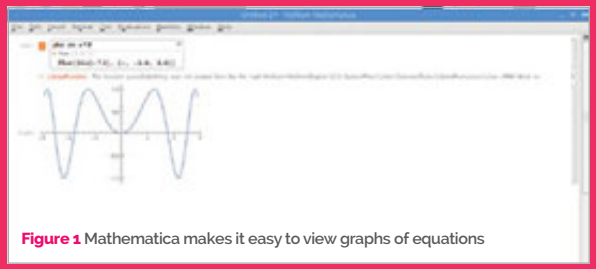


Figure 1 Mathematica makes it easy to view graphs of equations

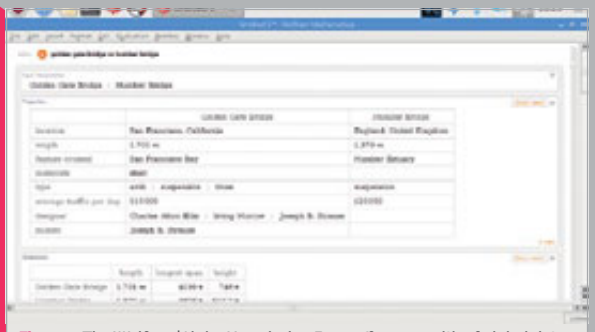


Figure 2 The Wolfram|Alpha Knowledge Base offers a wealth of global data

```
plot sin x^2
```

In free-form input this is converted to the formal Wolfram Language command: **`Plot[Sin[X^2, {x, -2.9, 2.9}]]`**.

You'll get to see a nice graph of the result (see **Figure 1**). Click the 'Show all results' button to see a range of subpods with sliders and controls, enabling you to fully explore the mathematical concept.

Free-form input is great for more advanced subjects, such as finding limits.

```
limit of sin x as x approaches pi/2
```

Click the 'Show all results' button to view the answer (1) as well as a plot of the graph with a red dot at the limit. Below it you'll even get a series expansion.

Wolfram|Alpha query

Beyond mathematics lies the wonderful world of the Wolfram|Alpha computational knowledge engine. This can also be used for advanced maths. However, its real strength is immediate access to a huge database of facts and stats. Want to know if more people live in London or New York?:

```
population of London > New York
```

```
True
```

Or get information on structures?:

```
golden gate bridge vs humber bridge
```

Wolfram|Alpha displays a wealth of information such as the location, length, and average traffic per day. It even displays photographs and a Wikipedia summary (see **Figure 2**). All of this data could be accessed by Wolfram Language, but the format is quite complex.

There's a lot of support from Wolfram. Mathematica is a very deep and powerful piece of software, and Wolfram has provided lots of resources and documentation, which you can access using Help > Online Documentation.

MATHEMATICA
INPUT

Wolfram Language input
(default)

Free-form input
(=)

Wolfram|Alpha query
(==)

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

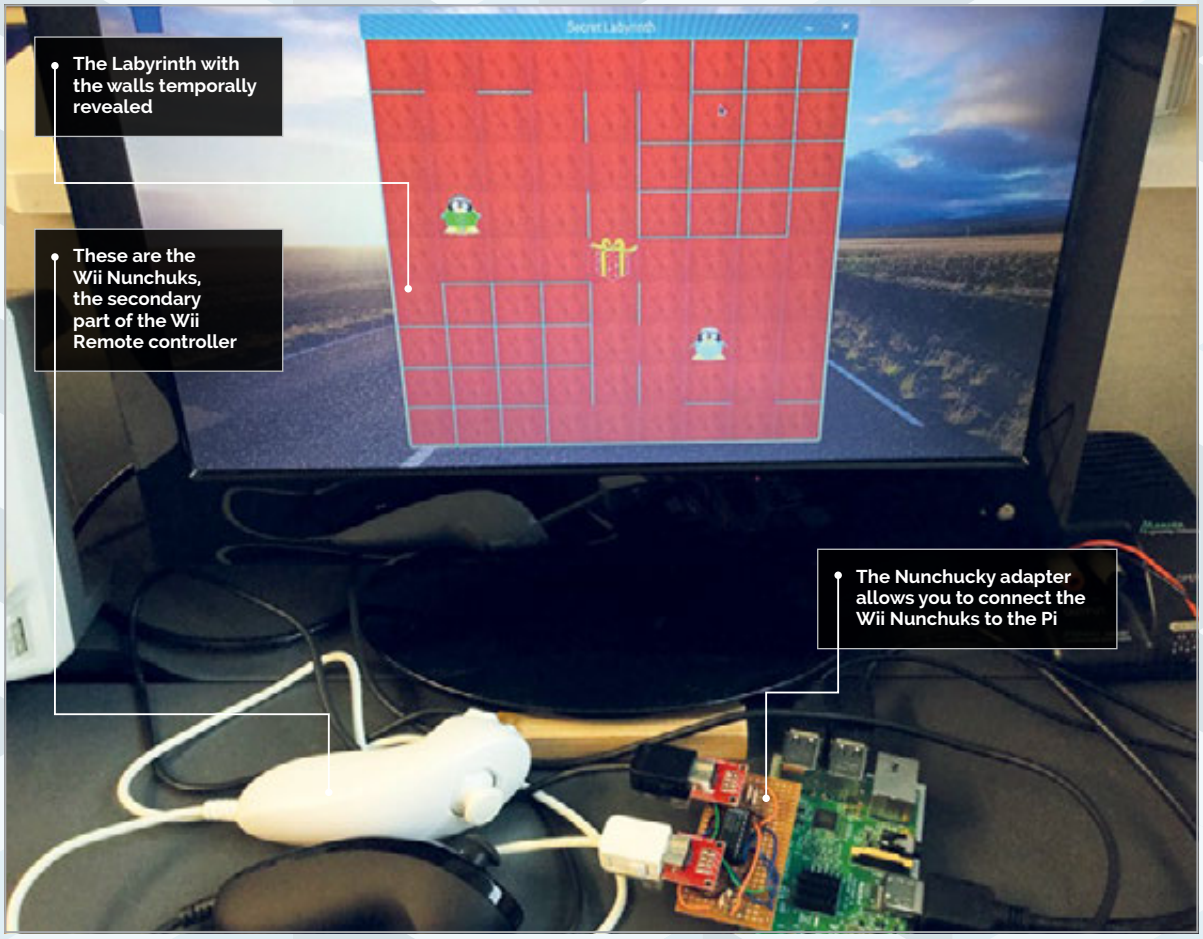
THE SECRET LABYRINTH

Use two Wii Nunchuks to play this unique fast-paced game

You'll Need

- 2 × Wii Nunchuks
- 2 × Nunchucky adapters **magpi.cc/2rzcndj**
- Stripboard: 24 strips of 13 holes
- 74HTC4052 analogue multiplexer **magpi.cc/2rArKCy**
- 16-pin IC socket
- Bottom-entry twin-row header sockets
- 4 × 68 kΩ resistors
- 0.1 µF ceramic capacitor

We love mazes here at the Bakery, but steering through one on a computer has its problems. Basically they are either too easy or too difficult. The Secret Labyrinth is hard but intriguing. It can be played as either a two-player or single-player game. The players begin at opposite corners of a maze and have to find the centre by using a series of horizontal and vertical moves controlled by a Wii Nunchuk. But the twist is that you can't see the walls in the maze: if you walk into a wall then you are instantly whisked back to your starting point. There are no player turns – just move as quickly as you can – and the maze is constructed so that the path from the bottom to the centre is the mirror image of the path from the top to the centre, so both players face an identical task.



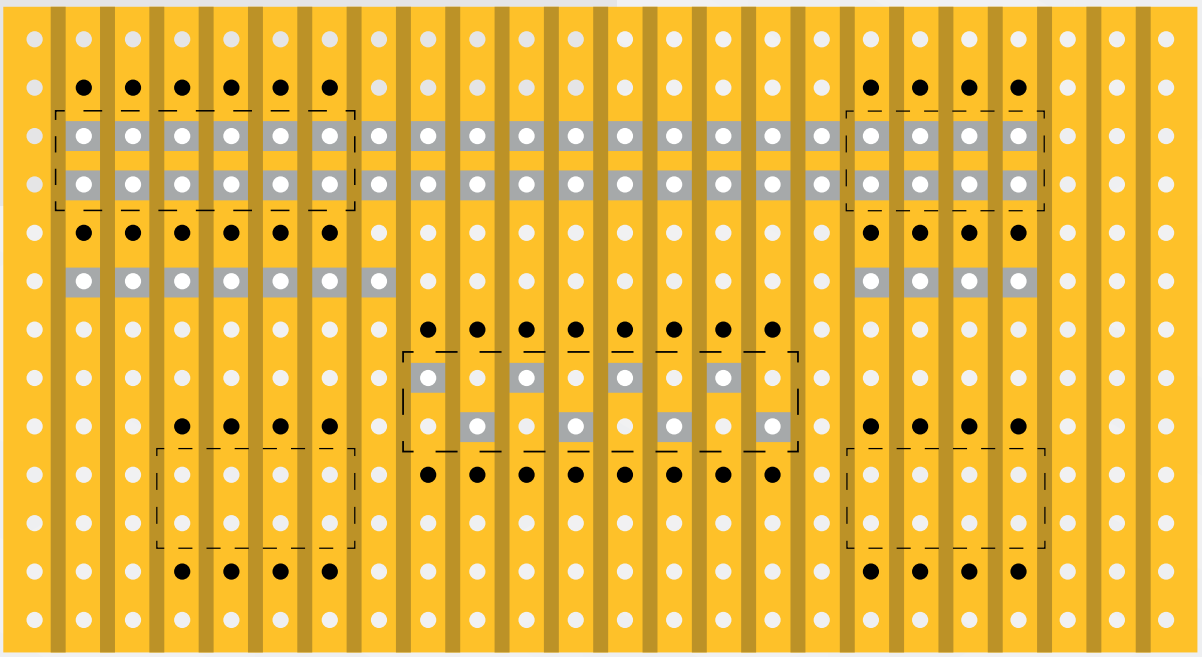
Nunchuks

The Wii Nunchuk is a cheap and remarkable interface. While in this project we will just be using the X-Y joystick, the Nunchuk also contains two push-buttons and a 10-bit three-axis accelerometer. They are interfaced through the I²C bus to the Raspberry Pi, but there is a small problem: each device on an I²C bus needs to have a unique address and all Nunchuks have the same address, fixed in the hardware, that can't be changed. So to get round this problem we have to split the bus by steering the I²C signal lines to one Nunchuk or the other by using an analogue multiplexer chip.

MAKING THE NUNCHUKS INTERFACE BOARD

>STEP-01
Preparing the board

Take a piece of stripboard 24 strips wide by 13 holes high, and break the tracks shown by the dark areas with a scalpel or spot face cutter. The dotted lines show the positions of the IC socket and the four bottom-entry sockets. The board will hang over the side of the Pi, not over the printed circuit board.



labyrinth.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

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

```
001. import sys, random
002. from smbus import SMBus
003. import RPi.GPIO as io
004. import pygame, os, time
005.
006. pygame.init()  # initialise graphics
     interface
007. pygame.mixer.quit()
008. pygame.mixer.init(frequency=22050,
     size=-16, channels=2, buffer=512)
009. os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
010. pygame.display.set_caption("Secret Labyrinth")
011. pygame.event.set_allowed(None)
012. pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
013.
014. gSpace = 70 # size of square
015. gSide = 9 # number of squares per side
016. nSquares = gSide*gSide
017. screenSize = gSpace * gSide
018. screen = pygame.display.set_mode([screenSize+2,screenSi
     ze+2],0,32)
019. random.seed() ; centre =  (nSquares-1) // 2
020. refresh = True ; joyLast = [-1,-1,-1,-1]
021. playPos = [0,0,0,0] # current and start position
022. maskFrom = [8,2,1,4]
023. dirInc = [-1,1,-gSide,gSide] # direction movement
024. wallColour = (0,255,255) # for revealed labyrinth
025.
026. def main():
027.     global refresh, done
028.     done = False
029.     initIO()
030.     init() # load in sound and graphics
031.     mazeGen() # create maze
```

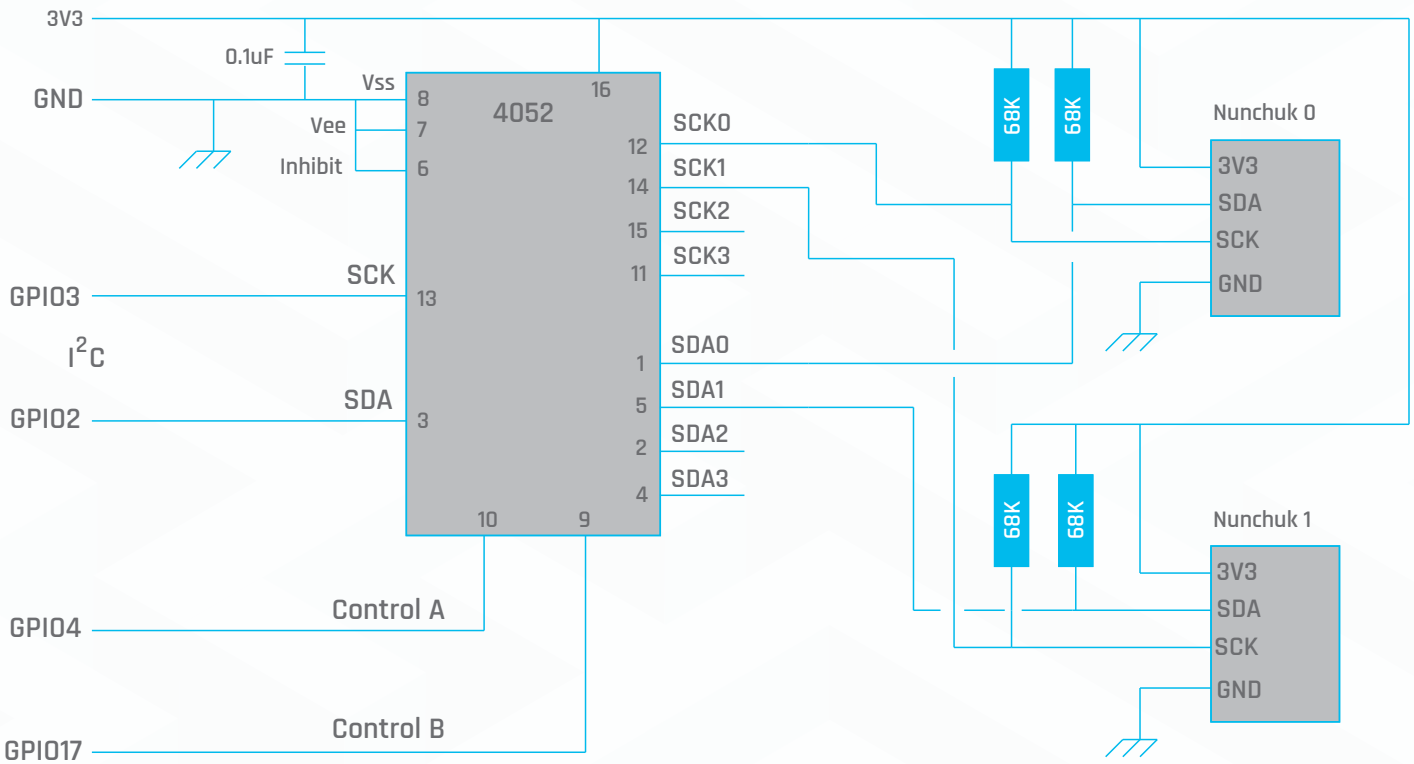
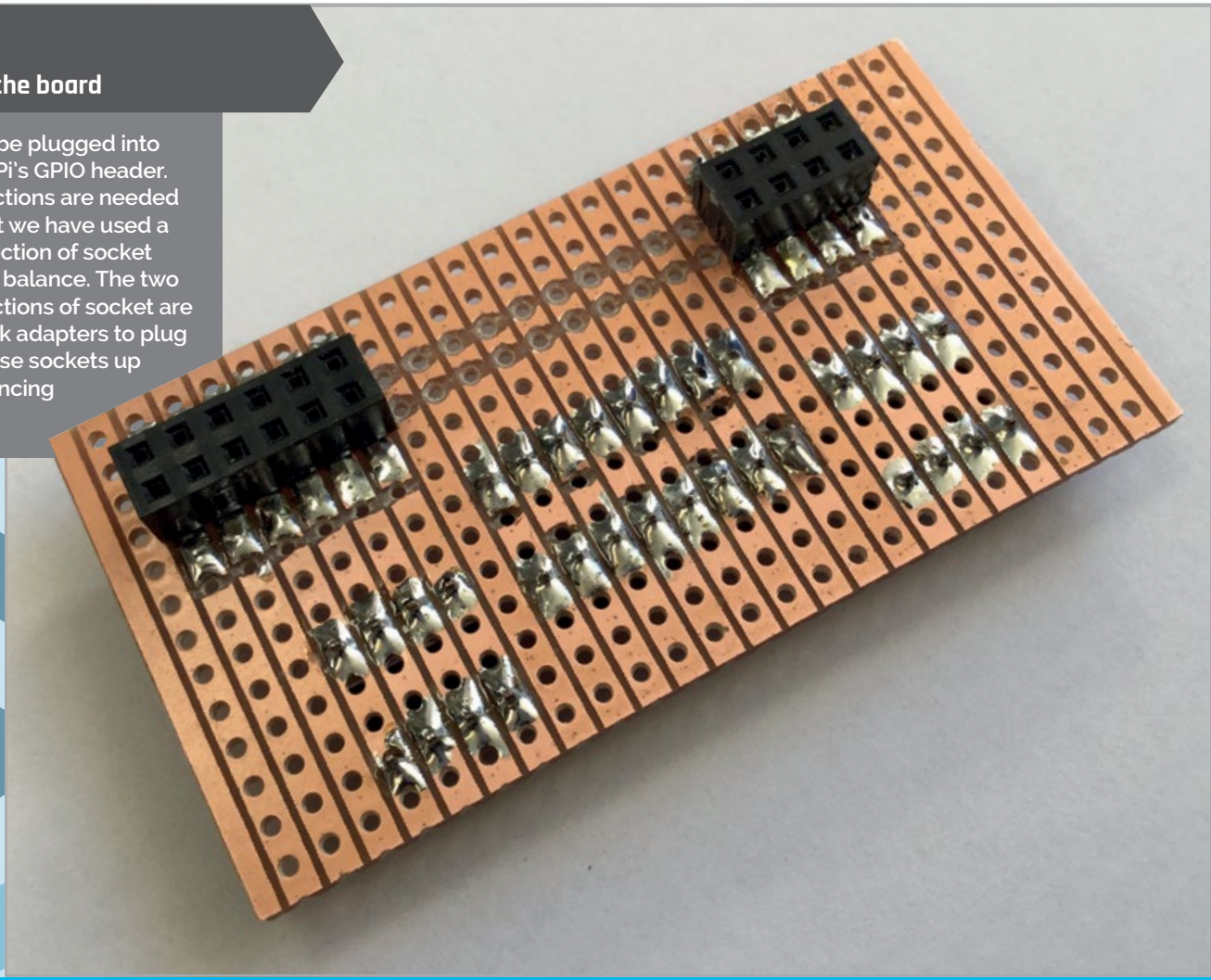


Figure 1 Schematic of the I²C multiplex board

>STEP-02
Underside of the board

The board will be plugged into the Raspberry Pi's GPIO header. Only six connections are needed electrically, but we have used a second four-section of socket for mechanical balance. The two four-length sections of socket are for the Nunchuk adapters to plug into. Solder these sockets up before commencing the wiring.



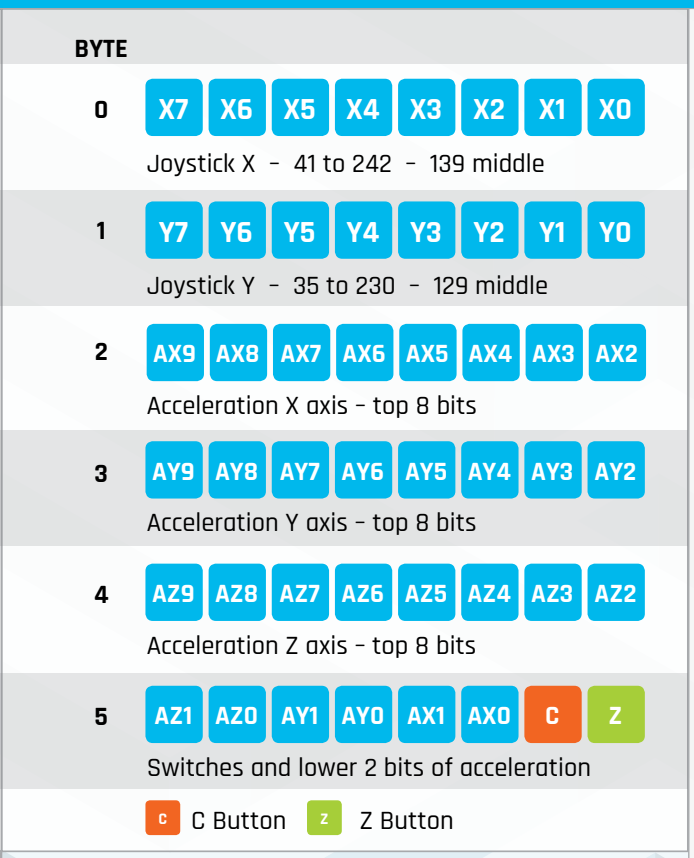


Figure 2 Registers returned by the Nunchuk

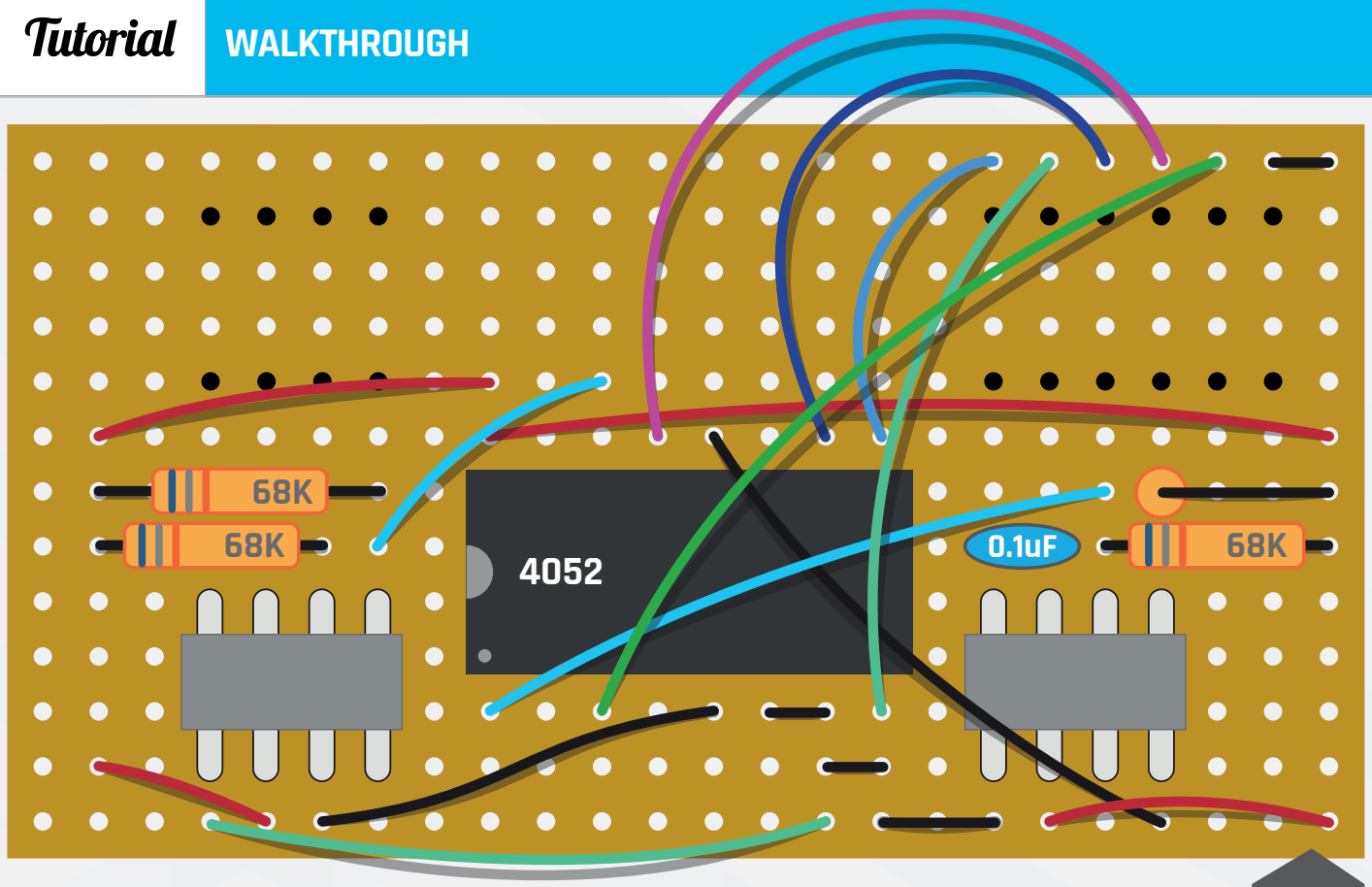
The multiplexer needs to be analogue because signals can go from the Raspberry Pi to the Nunchuk, or from the Nunchuk back into the Raspberry Pi. However, all we need to do is to set the multiplexer's select pins and talk to the Nunchuk normally, and the selected Nunchuk replies.

The schematic for this board is shown in **Figure 1** and full construction is shown in the step-by-step section. The Nunchuk itself when read, sends back a block of six bytes; each byte holds different data and a map of this data is shown in **Figure 2**. Note how the least significant two bits of the accelerator readings are in the last byte. This allows a simple access to an 8-bit reading as often the last two bits are noise. The two push-buttons are in this last byte as well.

The maze

The game is set in a 9×9 grid with players starting at the upper and lower corners. Due to the symmetry of the maze, it is in fact a 9×5 grid that contains the maze, but there is no restriction on a player going into the other half. Each square has four possible exits and these are marked as either blocked or clear in the list that holds the definition of the maze. So by checking the exit side of a square you want to move from, this will either allow free passage or send the player back to the start square. The maze is generated by starting at the centre and then moving in a random direction until a corner is reached. That corner becomes the start position for the player. On each random move,

```
032.    while 1:
033.       while not(done):
034.          checkForEvent()
035.          if refresh :
036.            drawScreen(False) # change to True to hide view
037.            refresh = False
038.       time.sleep(3.5) # see the walls
039.       mazeGen()
040.       done = False
041.
042. def drawScreen(laby):
043.     rSize = gSpace
044.     pygame.draw.rect(screen,(0,0,0),(0,0,screenSize,
     screenSize),0)
045.     #draw Tiles
046.     for s in range(0,nSquares):
047.       xp = findX(s)
048.       yp = findY(s)
049.       screen.blit(tiles[tilePlan[s]],(xp,yp))
050.     #draw grid
051.     if not(laby) :
052.        for y in range(0,gSide+1):
053.           pygame.draw.line(screen,(255,255,0),(gSpace*y,0),
     (gSpace*y,screenSize+2),2)
054.        for x in range(0,gSide+1):
055.           pygame.draw.line(screen,(255,255,0),(0,gSpace*x),
     (screenSize+2,gSpace*x),2)
056.
057.     #draw players & prize
058.     xp = findX(centre)
059.     yp = findY(centre)
060.     screen.blit(prize,(xp,yp))
061.     xp = findX(playPos[0])
062.     yp = findY(playPos[0])
063.     screen.blit(tux1,(xp,yp))
064.     xp = findX(playPos[1])
065.     yp = findY(playPos[1])
066.     screen.blit(tux2,(xp,yp))
067.
068.     # draw labyrinth
069.     if laby :
070.        for s in range (0,nSquares):
071.          xp = findX(s)
072.          yp = findY(s)
073.          for side in range(0,4):
074.             if not(maze[s] & 1) :
075.                pygame.draw.line(screen,wallColour,(xp,yp),
     (xp+rSize,yp),2)
076.             if not(maze[s] & 2) :
077.                pygame.draw.line(screen,wallColour,
     (xp+rSize,yp+rSize),(xp+rSize,yp),2)
078.             if not(maze[s] & 4) :
079.                pygame.draw.line(screen,wallColour,
     (xp+rSize,yp+rSize),(xp,yp+rSize),2)
080.             if not(maze[s] & 8) :
081.                pygame.draw.line(screen,wallColour,(xp,yp),
     (xp,yp+rSize),2)
082.     pygame.display.update()
083.
```

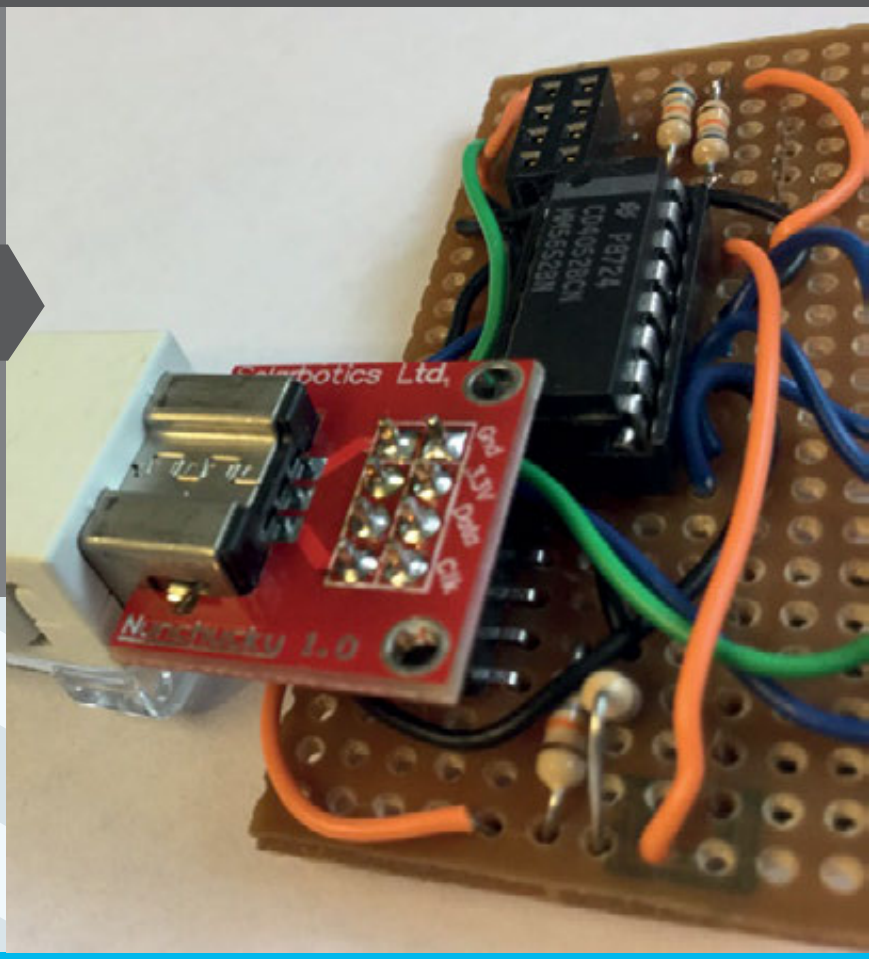


>STEP-03
Wiring up

Use stranded wire and make the links between the components, as shown in the diagram. The straight wires to adjacent holes can be made from tinned copper taken from the cut-off legs of the resistors. You can use different colour wires if you have them, but the electricity doesn't really care what colour wire it flows through.

>STEP-04
Wiring up

Solder the header pins onto the Nunchuk adaptors and plug the leads into them. Then insert them in the bottom-entry sockets on the board. Note that the Nunchuks are inserted in what we would consider to be upside-down. Plug the board into the Pi when it is unpowered, as with anything you plug into the GPIO header.



the side of the square just gone through is marked as clear and the side of the destination square is also marked as clear. This allows two-way movement between these two squares during the playing phase. The sides are marked as clear by setting the appropriate bit in the maze list element for that square. This is summarised in **Figure 3** (overleaf).

The side of each square is labelled clockwise from the top with the letters A to D. So if the

maze-generating program moves from the current square through side D, then bit 3 of the maze variable is set and the square it moves to (the one immediately to the left) has bit 1 for side B also set. This then becomes the current square and another random movement is made. There are constraints to stop the random movements occurring off the edge of the playing board, so that the whole board is ringed by blocked moves; this simplifies the playing logic. When any side of a square is cleared, the mirror image of that square and side are also cleared.

The resources

Each square of the board has a tile background chosen at random from a number of tiling strategies. Here we have implemented random tiles, diagonal rows of the same tile, displaced repeating, and all the same tile. Note how diagonal rows are simply displaced repeating with the displacement equal to one. We found that the tiling scheme as well as adding variety to the game also affected the way you play. We chose to use two versions of Tux, the Linux penguin, as the playing pieces. All tiles and playing pieces should be scaled to 72×72 pixels. We again raided the Scratch media files to get the sounds for when a player crosses a blocked boundary and is sent back to the start. A screen dump of a game in progress is shown in **Figure 4** (overleaf).

The software

The software for the project, **labyrinth.py**, uses our favourite Pygame frame work. It uses the smbus extension to handle the I²C interface – make sure it is enabled in the Interfaces tab of the Raspberry Pi Configuration tool. The main function is quite simple and consists of an infinite loop repeatedly playing different games. The inner `while` loop just plays one game by checking for an event or move and, if necessary, redrawing the screen. At the end of the game, the maze's hidden walls are revealed for a few seconds before generating a new maze and starting the play again. The `drawscreen` function takes in a Boolean value that determines whether the real walls are to be shown or if every square seems to have a wall. For very small people, you might want to play the game with the walls shown – simply change False to True in the `drawscreen` function call to do this.

The `mazeGen` function is perhaps the most complex in this code. It starts off by selecting a tile strategy at random and generating a `tilePlan` list which stores the tile number associated with each square. In the same loop, the old maze is erased. Then the centre square is set so that it has no barriers around it and the random walk clearing the walls is begun from the centre until either top corner is visited. That completes the maze and ensures at least one path

```
084. def findX(square):
085.    return (square % gSide)*gSpace
086.
087. def findY(square):
088.    return (square//gSide)*gSpace
089.
090. def initIO():
091.     global bus
092.     io.setmode(io.BCM)
093.     io.setwarnings(False)
094.     select = [4,17]
095.     io.setup(select,io.OUT)
096.     io.output(select,0) # select nunchuck 0
097.     print("Initialise I2C")
098.     if io.RPI_REVISION == 1:
099.       i2c_bus = 0
100.     else :
101.       i2c_bus = 1
102.     bus = SMBus(i2c_bus)
103.     for nun in range(0,2):
104.        bus.write_byte_data(0x52,0x40,0x00)
105.        io.output(4,1) # select nunchuck 1
106.        time.sleep(0.01)
107.
108. def init():
109.     global bus,sounds, maze, tux1, tux2, prize, tiles, tilePlan
110.     global finish, resetPlayer
111.     tux1 = pygame.image.load("images/Tux1.png").convert_alpha()
112.     tux2 = pygame.image.load("images/Tux2.png").convert_alpha()
113.     prize = pygame.image.load("images/present.png").convert_
    alpha()
114.     tiles = [pygame.image.load("images/Tile"+str(t)+".png").
    convert_alpha()
115.                 for t in range(0,15)]
116.     finish = pygame.mixer.Sound("sounds/finish.wav")
117.     resetPlayer = [pygame.mixer.Sound("sounds/Pop.wav"),
118.                    pygame.mixer.Sound("sounds/Rattle.wav")]
119.     maze = []
120.     tilePlan = []
121.     for sr in range(0,nSquares):
122.        maze.append(0) # create maze array
123.        tilePlan.append(0) # pattern of tiles
124.
125. def readNck(nunNum):
126. # the I2C drivers or something throws up an occasional error -
    this is the sticking plaster
127.     io.output(4,nunNum) # select the right nunchuck
128.     time.sleep(0.004)
129.     try:
130.        bus.write_byte(0x52,0)
131.     except:
132.        print("bus restart")
133.        time.sleep(0.1)
134.        initIO()
135.        bus.write_byte(0x52,0)
136.     time.sleep(0.002) #delay for Nunchuck to respond
137.     nCk = [((bus.read_byte(0x52) ^ 0x17) +0x17) for i in
138. range(0,6)]
139.     return nCk
```

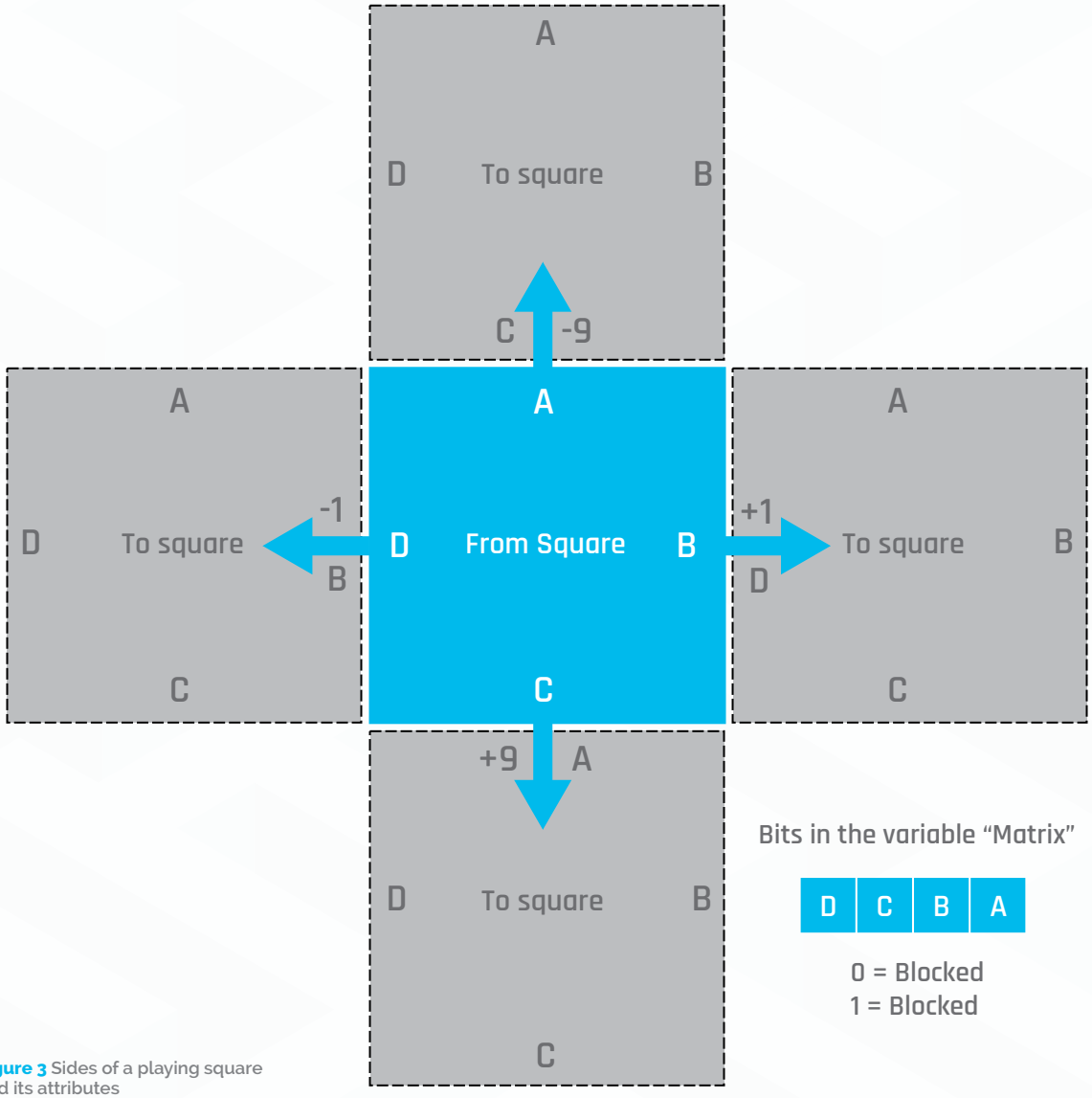


Figure 3 Sides of a playing square and its attributes

through it has been generated. We have a version of the program in our GitHub repository that uses only keyboard keys to move. Finally, the **G** key will give a glimpse of the walls for a second or two – only to be used if you are really stuck.

Taking it further

You could add a timer and keep a high score of the speed of a run. Or you could change the game into a turn-based affair. As the maze is symmetrical, you could – if you are clever – avoid some walls by remembering what your opponent has done. You could also change the ending point of the maze generation when it reaches the top or bottom row instead of a corner. For the more advanced, you could generate the mazes by hand, or another program, and store several of them in a file and pick the one to use at random. This would ensure a much greater variety of mazes, and allow you to make quite complex ones, but beware of making them too complex.

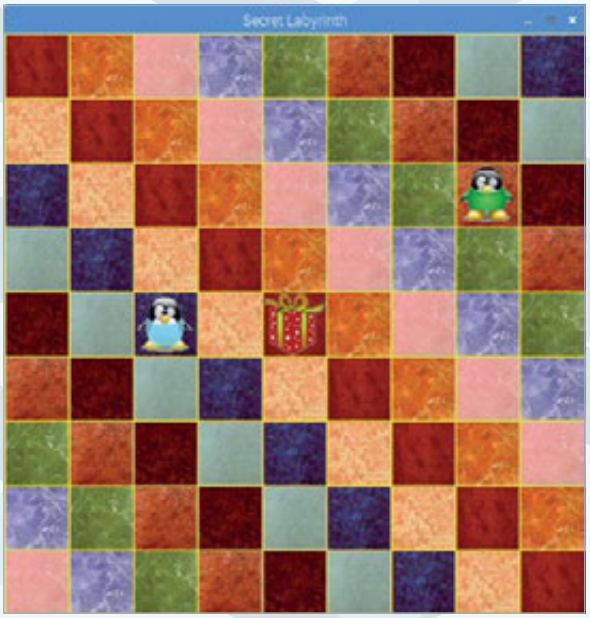


Figure 4 A game in action; all walls are hidden

```
140.
141. def mazeGen():
142.    global maze, current, last, tilePlan, playPos,
     refresh
143.    refresh = True
144.    maskTo = [2,8,4,1]
145.    thisTile = random.randint(0,14)
146.    thisMethod = random.randint(0,3)
147.    for s in range(0,nSquares):
148.      maze[s] = 0 # clear the maze
149.      if thisMethod == 0:
150.         tilePlan[s] = random.randint(0,14) # random
     tile
151.      if thisMethod == 1:
152.         tilePlan[s] = s % 10 # diagonal lines
153.      if thisMethod == 2:
154.         tilePlan[s] = thisTile # all the same but
     different for each maze
155.      if thisMethod == 3:
156.         tilePlan[s] = s % 12 # offset lines
157.    maze[centre] = 15
158.    current = centre ; last = centre
159.    while not(current == 0 or current == 8):
160.       blocked = True
161.       while blocked:
162.          move = random.randint(0,3)
163.          direction = dirInc[move] # initial direction
     of move
164.          test = direction + current
165.          if test >= 0 and test <= (nSquares/2 +gSide/2)
     and test != last: # within the board
166.             if not(direction ==1 and current % gSide ==
     gSide-1) and not(direction ==-1 and current % gSide ==
     0):
167.                blocked = False
168.                maze[last] = maze[last] | maskFrom[move]
     # clear exit square
169.                maze[nSquares-last-1] = swap(maze[last])
170.                current = test
171.                last = current
172.                maze[current] = maze[current] |
     maskTo[move] # clear entrance square
173.                maze[nSquares-current-1] =
     swap(maze[current]) #need to swap bits
174.    playPos[0] = current # player start positions
175.    playPos[1] = nSquares - 1 - current
176.    playPos[2] = playPos[0] ; playPos[3] = playPos[1] #
     start positions
177.
178. def movePlayer(pn, inc):
179.    global refresh, playPos, done
180.    refresh = True
181.    fromSq = playPos[pn]
182.    playPos[pn] += dirInc[inc]
183.    if not(maze[fromSq] & maskFrom[inc]) :  # a wall in
     the way
184.       playPos[pn] = playPos[pn+2] # back to start
185.       resetPlayer[pn].play() # reset noise
186.    if playPos[pn] == centre : # winner
187.       finish.play()
188.       refresh = False
189.       drawScreen(True)
190.       done = True
191.
192. def getMove():
193.    global joyLast
194.    joyNow = [-1,-1,-1,-1] # no move
195.    for nun in range(0,2): # read both nunchucks
196.       x = nun
197.       y = 2+nun
198.       bank = readNck(nun)
199.       if bank[0] > 190:
200.          joyNow[x] = 1 # move right
201.       if bank[0] < 60:
202.          joyNow[x] = 0 # move left
203.       if bank[1] > 190:
204.          joyNow[y] = 2 # move up
205.       if bank[1] < 60:
206.          joyNow[y] = 3 # move down
207.       if joyNow[x] != joyLast[x] and joyNow[x] > -1:
208.          movePlayer(nun,joyNow[x]) # move left or right
209.       if joyNow[y] != joyLast[y] and joyNow[y] > 1:
210.          movePlayer(nun,joyNow[y]) # move up or down
211.       joyLast[x] = joyNow[x]
212.       joyLast[y] = joyNow[y]
213.
214. def swap(value):
215.   result = (value & 1) << 2 | (value & 4) >>2 |
     (value & 2) << 2 | (value & 8) >> 2
216.   return result
217.
218. def terminate(): # close down the program
219.     print ("Closing down please wait")
220.     pygame.mixer.quit()
221.     pygame.quit() # close pygame
222.     os._exit(1)
223.
224. def checkForEvent(): # see if we need to quit
225.     global refresh
226.     event = pygame.event.poll()
227.     if event.type == pygame.QUIT :
228.          terminate()
229.     getMove()
230.     if event.type == pygame.KEYDOWN :
231.        if event.key == pygame.K_ESCAPE :
232.           terminate()
233.        getMove()
234.        if event.key == pygame.K_g : #glimpse maze walls
235.           drawScreen(True)
236.           time.sleep(2.0) # time to display
237.           refresh = True # draw over it
238.
239. # Main program logic:
240. if __name__ == '__main__':
241.     main()
```

BILL BALLARD

Bill is a retired physicist and former Radiological/Nuclear Countermeasures Program Manager who got back to his programming roots to find fun things to do with his ten Raspberry Pis.
github.com/wpballa

SIMPLE PI GEIGER COUNTER DISPLAY

This project connects a MightyOhm Geiger counter to a Pimoroni Scroll pHAT display for real-time radiation measurements

You'll Need

- Raspbian Lite **magpi.cc/2ejN6sk**
- MightyOhm Geiger counter with Geiger tube **magpi.cc/2Gc4CNT**
- Geiger counter case to protect you from high voltage **magpi.cc/2GejF9U**
- 5V USB power supply
- Pimoroni Scroll pHAT HD **magpi.cc/2wShYcf**

The MightyOhm Geiger counter is a great DIY project, and there is a simple way to connect it to a Raspberry Pi for data logging and displaying with a Pimoroni Scroll pHAT. The code supports either the newer HD Scroll pHAT or the older version with a simple variable change at the start of the Python program.

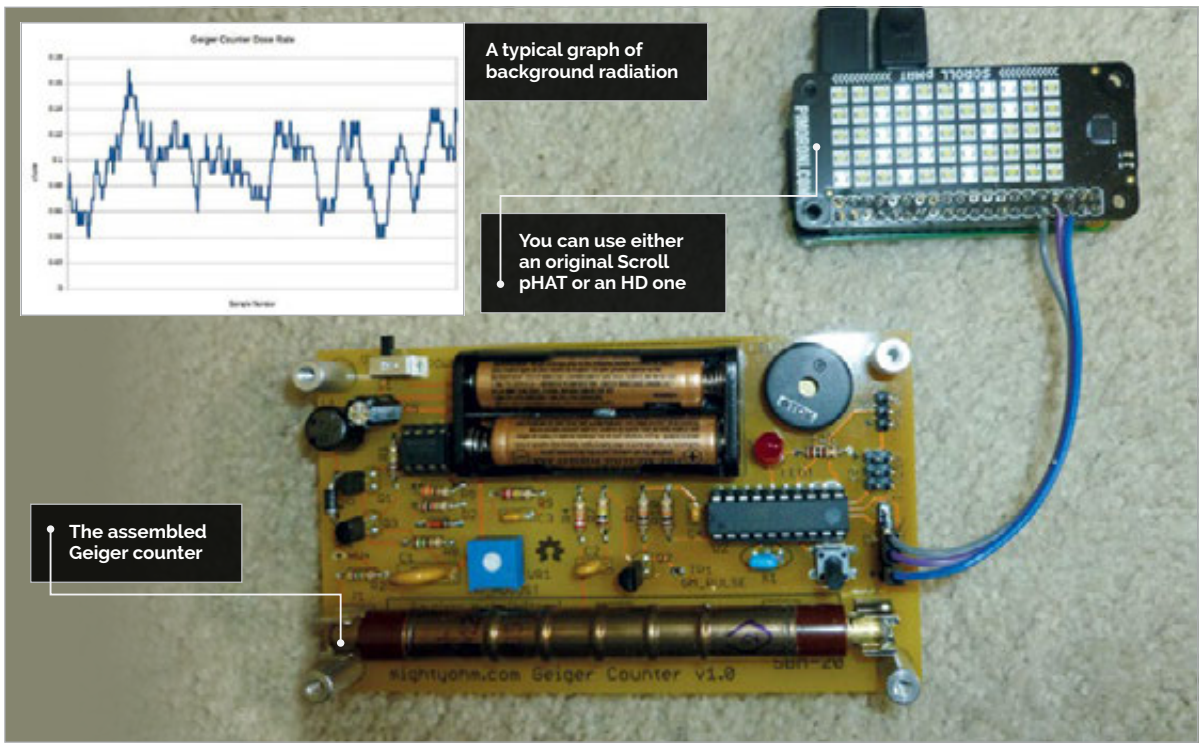
Software setup

Download and install the latest version of Raspbian Lite (**magpi.cc/2ejN6sk**) on your microSD card using any of the excellent instructions in the Raspberry Pi Forum. If you copy your **wpa_supplicant.conf** file to the **boot** partition before the first startup, your wireless network connection will be active immediately. Initialise Raspbian with raspi-config to the proper Internationalisation Options (time zone, keyboard, wireless country code, etc.), change the default password, and set the host name in Advanced Options to something like 'geiger'. Make sure that you enable I2C in the Advanced Options of raspi-config, and SSH if you will be connecting remotely. Next, edit:

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

...and remove:

```
console=serial0,115300
```



PLOT A GRAPH

The file **geiger.csv** is suitable for opening with LibreOffice Calc and then graphing. Be sure to select UTF-8 encoding and to merge delimiters when importing the CSV file.

SSH IN

Connecting to the Pi via SSH is simple. Use **geiger.local** as the address in your favourite SSH application – no need to know the IP address.

…from the line. Save, then exit the file and reboot the Raspberry Pi. Now you should do the usual update and upgrade to the operating system:

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

Next, install the Python pip and scrollphathd applications (just leave off the 'hd' if using an older Scroll pHAT):

```
sudo apt-get install python3-pip python3-scrollphathd
```

Finally, install pySerial and flask:

```
sudo pip3 install pySerial flask
```

…and you are all set.

Download the **geiger.py** file from GitHub (**magpi.cc/2Foqfzl**) and customise it for your desired log file location and your Scroll pHAT version (HD = True or False).

Hardware setup

Assuming you have already assembled the Geiger counter with the MightyOhm instructions, you only need to solder the header to the Scroll pHAT and connect a few wires. Optionally, you can use an extra-tall 2×20 stacking header so you can use a connector. To connect the pHAT to the Raspberry Pi, you need to solder three wires to the Scroll pHAT corresponding to GPIO physical pins 6, 8, and 10. Pin 6 (GND) connects to the Geiger counter's J7 connector pin 2, pin 8 (GPIO 14) connects to the J7 pin 4, and pin 10 (GPIO 15) connects to the J7 pin 5. Feel free to use a connector rather than direct wiring.

Run it

Connect the Pi to power, then turn on the Geiger counter (mute it if you wish, using the push-button). Log in and launch the program with:

```
python3 geiger.py &
```

The display should start showing the uncalibrated radiation level in microsieverts per hour, scrolling across the screen until you kill the process using:

```
kill -9 PID
```

…where PID was the number shown when you initially launched the program. Alternatively, you can use:

```
ps -ef | grep geiger
```

…to find the PID of the **geiger.py** process.

geiger.py

Language: >PYTHON 3

DOWNLOAD: magpi.cc/2Foqfzl

```
#! /usr/bin/python3

# set this line according to your version of the scroll pHAT
HD = True   # set to true for HD version, false for old version
import serial
import time
if (HD):
    import scrollphathd
    from scrollphathd.fonts import font5x7
else:
    import scrollphat
fname = "/home/pi/geiger.csv"
# open the mightyohm geiger counter terminal
ser = serial.Serial('/dev/ttyAMA0', baudrate=9600)
# initialize scrollphathd or scrollphat
if (HD):
    scrollphathd.set_brightness(0.2)
    scrollphathd.rotate(180)
    scrollphathd.clear()
else:
    scrollphat.set_brightness(32)
    scrollphat.set_rotate(True)
    scrollphat.clear()
# read each line of input, reformat byte to string and write to file
while True:
    try:
        lin=ser.readline()
        line=lin.decode('utf-8').split(',')
        outs = line[5] + line[4]
        if (HD) :
            scrollphathd.clear()
            scrollphathd.write_string(outs, x=1, y=0, font=font5x7)
            (buf1, buf2) = scrollphathd.get_buffer_shape()
            for i in range(buf1):
                scrollphathd.show()
                scrollphathd.scroll()
                time.sleep(0.1)
        else:
            scrollphat.clear()
            scrollphat.write_string(outs, 11)
            len = scrollphat.buffer_len()
            for i in range (len):
                scrollphat.scroll()
                time.sleep(0.1)
# now write to file, close each time in case you stop the program
        geig = open(fname, 'a')
        geig.write(lin.decode('utf-8'))
        geig.close()
    except (KeyboardInterrupt, SystemError, SystemExit):
        if (HD):
            scrollphathd.clear()
        else:
            scrollphat.clear()
        geig.close()
        ser.close()
```

BRIAN BEUKEN

Very old game programmer now teaching very young game programmers a lot of bad habits at Breda University of Applied Science in Breda NL.
scratchpadgames.net

NEVER BLAME THE MACHINE...

...Even when it is wrong. It is sometimes the machine, but only sometimes.

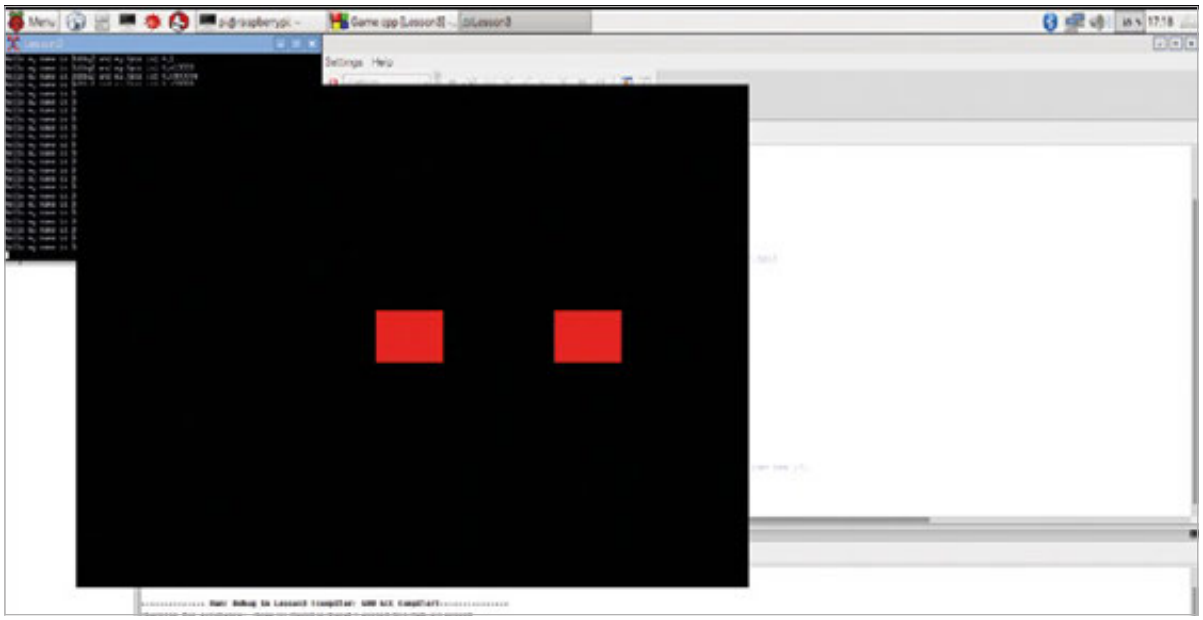


Figure 1 Let's be square(ish)

CODING GAMES ON THE RASPBERRY PI IN C/C++ PART 03

After our setup of systems, it's time to get some graphics sorted out

You'll Need

- Code::Blocks `sudo apt-get install codeblocks`
- STB **magpi.cc/2GCQe19**
- A pretty picture; not too big

We had to plough through a lot of tech stuff last time, but we still have a tiny bit more to do and need to make a few additions so we can start to build up a framework for a game. You should be comfortable with objects now, and it's time to use some code concepts to control them and create a basic code framework that will evolve into a playable game.

Last time we bounced a couple of triangles around, but triangles are not much use for games. We need to be able to move variable-sized objects around and have some control over the graphics they present us with to make our games fun and interesting.

But wait, all this tech is overwhelming. So far we haven't really learned much about code, so it's time to explain some of that and go through what we did last time so we can expand on it.

Time for some C/C++ foundations

All programming, regardless of the language, is about manipulating data, usually represented by variables, based on various conditions or needs, and in turn acting on that data to create other conditions or needs.

CPUs do that by simply performing one instruction at a time and then passing control to the next instruction; an instruction could be something which changes, inspects, or stores selected variables and then continues to the next instruction in line, or passes control to a different set of instructions.

In C/C++, instructions can be singular, like:

```
int a = 1;
```

Language
>C++

NAME:
MagPiHelloTriangle.cpp, SimpleObj.h/cpp, Game.h/cpp, OGL.h/cpp

DOWNLOAD:
magpi.cc/2suLVC2

Or grouped, using **{}** braces:

```
{
Int a = 1;
Int b = 2;
Int c = a+b;
}
```

Groupings make more sense when we want to do a number of different things following a test. Braces **{}** will become both your friend and your enemy – as will the **;** at the end of an instruction, which is 'nearly' always needed.

Both these singular and grouped instruction examples are setting up integer-type variables, and putting values into them. C/C++ insists you let it know what **type** of variable you are creating so it can be sure you put the right type into it – it'll usually complain if you don't! For now, we will use **int** and **float**; **int** types are for integer whole numbers, and **float** types are for more precise floating point numbers like pi:

```
float Pi = 3.14159265f;
```

Variables generally should be named something appropriate to their function. For example:

```
int MyAge = 54;
```

…makes more sense when reading it back than:

```
int ma = 54;
```

It's up to you what names you use, but generally it's best to make variables describe what they are holding. There is no limit on the size of your variable names, so:

```
int I_will_be_this_age_next_birthday =  MyAge + 1;
```

…is valid, but will get you some funny looks at a code review.

Figure 2 A familiar graphic

Not too much to remember

There aren't actually many instructions in the C/C++ language, but it does have a lot of very specific rules on how to use them. We'll stick with the simpler ideas to begin with, and expand our understanding as we need to.

When last time we added a **Speed** variable to our **Xpos** variable, we were able to test the resulting new position of our object and decide if we make a change to our **Speed** value, or not. A choice was made based on a condition test. We asked a simple question: **if** the Xpos is bigger (or less) than the value representing the edge of the screen, (then) do something to alter the direction, in this case negate the speed we move.

If, then, else concepts are important to most languages, even if they don't have the 'then' keyword; C/C++ does not actually have that word, but it's helpful to think it when you are writing code. It lets us make decisions based on the resulting truth or otherwise of a test.

> "If, then, else concepts are important to most languages, even if they don't have the 'then'"

How did that decision process make an entire triangle move? Well, it didn't: it made a reference point at the centre of that triangle move, and we simply told our GPU to build a triangle around that X,Y co-ordinate. We were able to abstract the concept of the triangle, down to one single point and modify that with simple arithmetic, adding a + or – speed to move it on each update.

So there are a few basic principles we should be able to get our head around from that.

Assignment: where we store a value into a variable.

Manipulation: where we modify the value of a variable.

Condition checks: where we test variables and potentially change the sequential flow of our program.

These are great foundations, but because a program usually goes from one instruction to the next, it will eventually reach the end of the code. That's where some special flow control methods we call loops come in. We've seen one type of so far unexplained loop, the **for** loop, which we used to repeat our updates.

A **for** loop, when set up, is given a counter variable, with an initial value, which it then automatically modifies at the end of each loop. It then compares the counter to see if the condition also stated in the

TYPE CAREFULLY

Typing slowly and carefully forces you to think about what you enter, and helps you to manage your code.

USE SENSIBLE VARIABLE NAMES

A variable that explains its own function is easier to keep track of.

for declaration is still true. If not, it exits the loop, otherwise it starts again, this time with the new value in the counter.

So:

```
for (int a =0;  a<20; a++)
{ /* this is the start of the loop*/
  //… do any amount of stuff between the {}
}/* the end brace is the end of the loop*/
```

…just makes the loop start at 0 (**int a=0**), and at the end of each loop it will increment **a** (**a++**) until such time as **a** is no longer less than 20 (**a<20**).

How that counter is modified is up to us, but usually we just settle for a simple increment **a++**,

> Triangles might be cute, but they're not a lot of use for games. We need something more functional

or decrement **a--**, which allows us to use the counter as a handy index to lists of things we want to process.

The code between brackets can be anything at all. Here it will repeat 20 times with this formatting and give us the ability, if we want it, to use the a variable for different things, each time being a value 1 more than the last time; we call this iteration.

There are a couple of other types of loop; we'll discuss them as we need them. For the meantime a **for** loop is simple to understand, and we will use it often.

Being able to test things and choose the next set of instructions we do is an important concept. This is what gives us program flow control and it's crucial to getting computers to do what we want them to do and change the sequence of instructions we want to follow.

All these concepts are easier to visualise when we can see some actual result on screen, but triangles are not quite what we need – let's expand our graphical abilities a bit more.

Time to get square

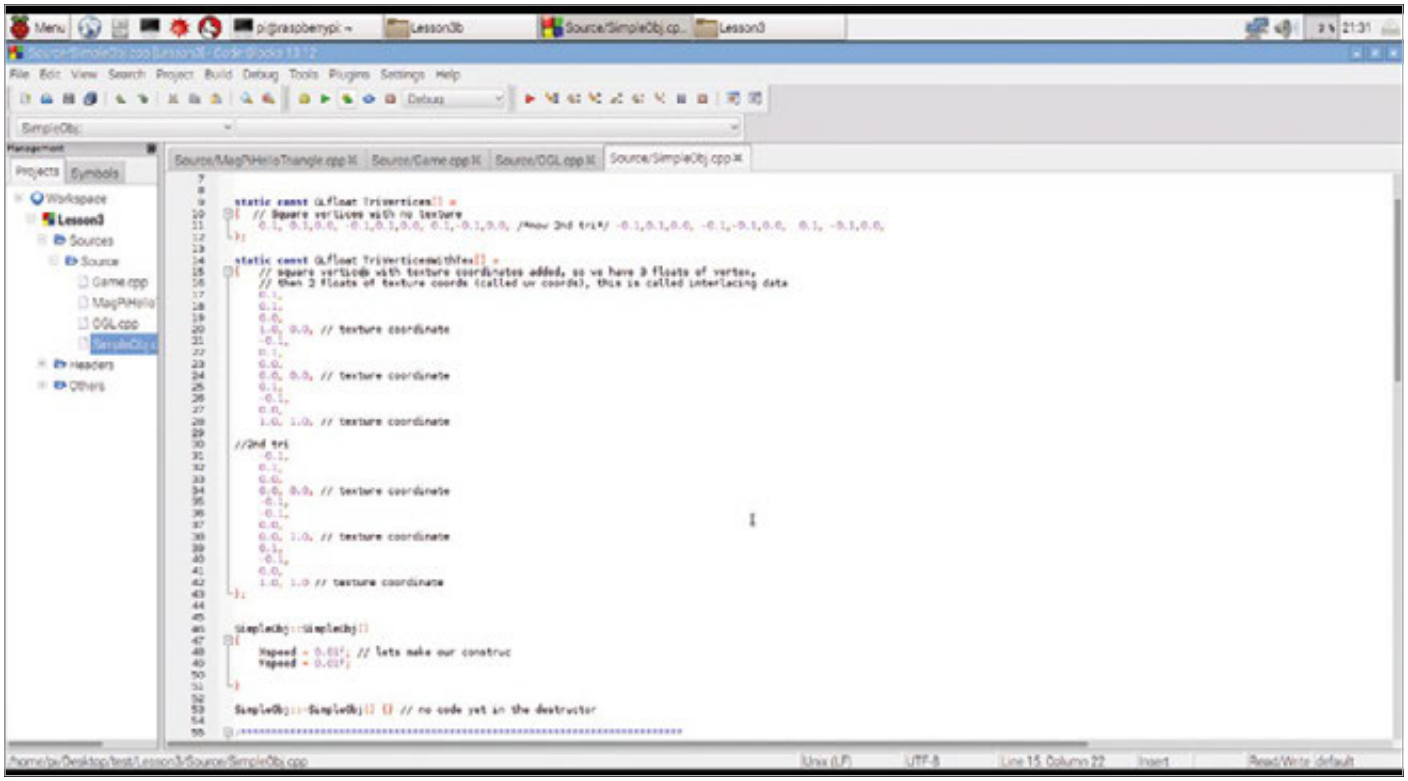
Triangles might be cute, but they're not a lot of use for games. We need something more functional, so we'll build our first games from two basic concepts: sprites and tiles.

Both are essentially square objects, which can be made from two right-angled triangles. Tiles can be used to create a background, and sprites can be used to represent our objects/players moving around. We'll focus on tiles this time.

Let's make a pretty simple change to our current Lesson 2 system, so we just send two triangles to the GPU and ask it to draw them as a square. In your **SimplObj.cpp** file, change the contents of **TriVertices** to this:

```
static const GLfloat TriVertices [] =
{ 0.1,0.1,0.0, -0.1,0.1,0.0, 0.1,-
0.1,0.0, -0.1,0.1,0.0, -0.1,-0.1,0.0,
0.1,-0.1,0.0, };
```

Figure 3
With or without texture



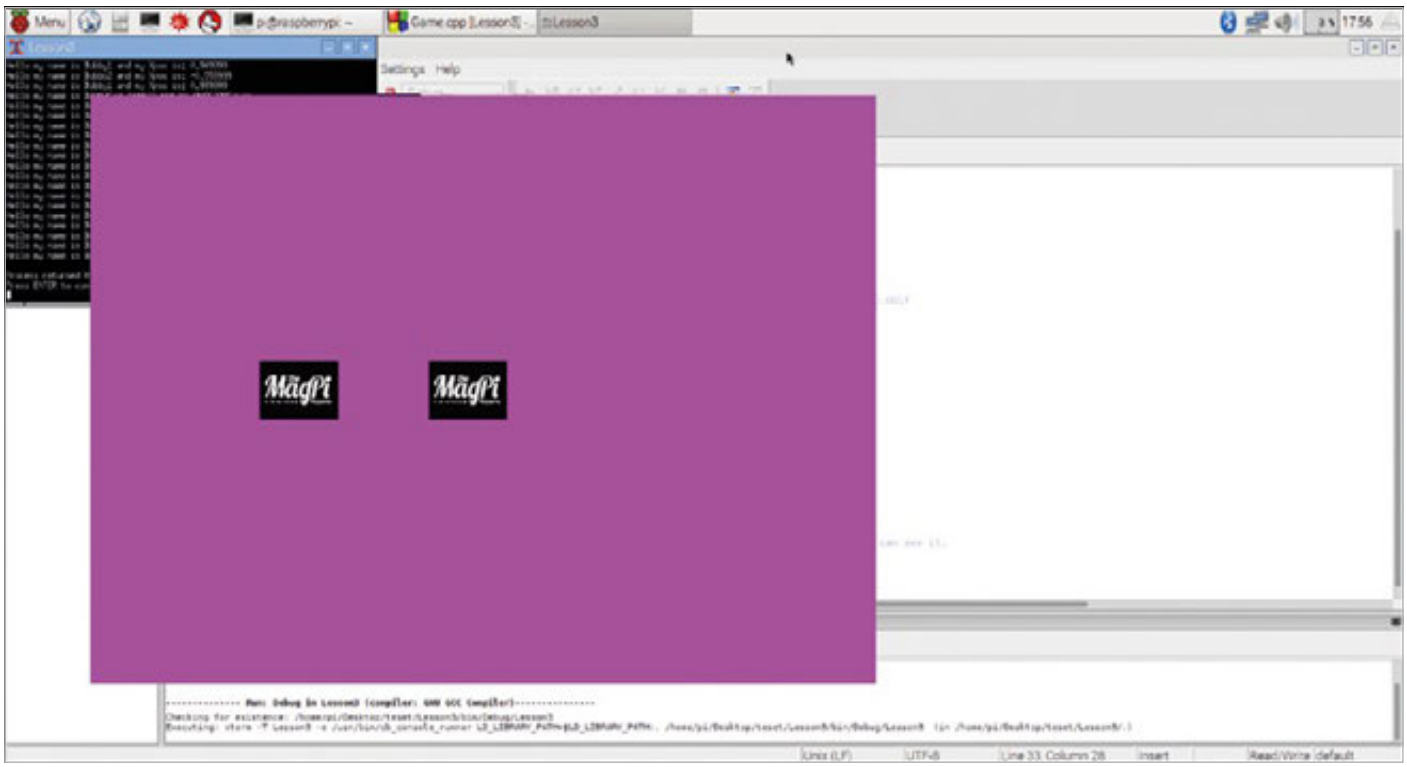


Figure 4
And now we can draw images

You should see there are now six points/vertices representing the tips of two right-angled triangles, some of them duplicates – we'll cover that later.

And also, in the **Draw** method, change the **glDrawArrays** to now draw six vertices. Compile and run.

Nice, we have red squares (well, squarish), as in **Figure 1**; it's half-way there. Next we need to make those squares into some kind of recognisable graphic. We can't really do that in the Lesson 2 code without a large listing, so let's get hold of Lesson 3 code, which will now load some graphics for us (**Figure 2**), create textures (**Figure 3**), and let us put a simple image on screen (**Figure 4**).

Loading graphics

Images are funny things; they come in many formats and resolutions, but for use as graphics in our games we need them in very raw byte-for-byte formats. That requires us to load the funny formats and convert them into raw data ready for OpenGL ES 2.0 to turn into a texture for display. That's quite a chore that most sane coders avoid, so we need some help with it.

Notice in the new **OGL.cpp** file that we also add a new header, called **stb_image.h**, which is a helper system that will let us load and convert images. We can largely ignore how it does what it does, just be confident that it does what we need it to do. It's part of several helper systems collectively known as STB, from coder Sean T Barrett, and is public domain so anyone can use them. See the 'You'll Need' box on the first page!

Let Git do its thing and you will now have a new directory called, unsurprisingly, **STB**, and in there will be the **stb_image.h** file we need. You can choose to copy it to your working directories, or just leave it there and make sure that Code::Blocks adds **/home/pi/stb** to its list of directories, as we have in the Lesson 3 project files for you.

A new initialisation

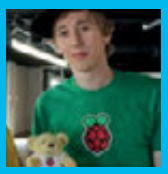
Looking at Lesson 3's source, you can see OGL has been adapted to load a familiar image and turn it into a texture. The shaders are also adapted to use that texture. Take a look again at the **TriVerticesWithTex** list in **SimpleObj.cpp** and you will see some more small changes and explanations in code.

It's still all a bit hard-coded, making it inflexible, but for tutorial purposes we can now see that we are able to draw a new kind of textured object. Now SimpleObj is able to display graphics!

So far so good, it's still not very elegant code, but we have managed to put our graphics on screen with very little effort. We could put a lot of them on, creating multiple instances of a SimpleObj. But 100 squares with the same graphics isn't fun. Next time we need to add some flexibility to load and control more graphics and have them do different things.

TRY YOUR OWN IMAGES

You can use almost any image, but keep it below 1024×1024 pixels. PNG is, we believe, probably your best option.



BEN NUTTALL

Raspberry Pi's Community Manager, creator or the GPIO Zero library, and the Foundation's resident Python expert.
twitter.com/ben_nuttall

PI ZERO GPIO EXPANDER

Access the GPIO pins on a Pi Zero from your PC, or double the available GPIOs on a Pi 3!

You'll Need

- PC (desktop or laptop running Windows, Mac or Linux)
- Pi Zero (any model) with GPIO header
- Debian Stretch with Raspberry Pi Desktop on USB stick **rpf.io/x86**
- micro USB cable or Pi Zero USB Stem **magpi.cc/2Gb2d67**
- LEDs and a button
- 3D Xmas Tree **magpi.cc/2GewFwe**
- SnowPi **magpi.cc/2GdupFA**

The latest release of the Raspberry Pi Desktop x86 operating system alongside Raspbian Stretch for Raspberry Pi includes a new GPIO expander tool. Now you can easily run GPIO programs from your PC by simply plugging in a Pi Zero with a single USB cable and no SD card is required. It's handy for prototyping, and can be really useful in coding clubs, libraries, and schools where the conventional Pi setup isn't possible. It has limitations, but it's ideal for lots of GPIO projects, such as many of the GPIO Zero ones you'll find in *The MagPi*.

The GPIO expander tool means you can connect a Pi Zero (original, W, or WH) to a regular PC with just a micro USB cable, no SD card required, and control the GPIO pins. The easiest way to get it working is by live-booting the Raspberry Pi Desktop x86 OS on your PC. Remember, when live-booting, once you shut down, just remove the USB stick and you can return to your usual operating system and get back to your boring spreadsheets or whatever.

Above As a Christmas decoration, we used the Pi Hut 3D Xmas Tree and Ryanteck SnowPi boards together by plugging a Pi Zero into a Pi 3

Boot into the Stretch release of Raspberry Pi Desktop, which comes with everything you need, ready to go. Plug in your Pi Zero and you'll be prompted to select a role for the device. Select 'GPIO expansion board' and continue. It will take 30 seconds or so to flash it, then the dialogue will disappear.

Once the dialogue disappears, that means your Pi Zero is ready. Try pinging it just to be sure it's connected. Open a Terminal and type:

```
ping6 fe80::1%usb0
```

This is the Pi Zero's IPv6 address. You should see some responses, so press **CTRL+C** to stop pinging. If you can't connect to it, try plugging it back in and trying again. You'll need to define two environment variables to set this as the default route for GPIO commands. Type:

```
leafpad .bashrc &
```

This will open a simple notepad editor with your user's Bash configuration. Scroll to the bottom and, on new lines, add:

```
export GPIOZERO_PIN_FACTORY=pigpio
export PIGPIO_ADDR=fe80::1%usb0
```

Save the file and close it. Now close the Terminal window and open up a Python editor or shell (such as Thonny or IDLE). Connect up an LED or a simple LED board like a Pi-Stop, and write the following code to flash a single LED (you'll need to know which GPIO pin number it's connected to – we're using GPIO 17):

```
from gpiozero import LED
led = LED(17)
led.blink()
```

You should see the LED on the Pi Zero blink on and off continuously. That's just the start! Take a look at the GPIO Zero documentation (**magpi.cc/2qd2MEb**), as well as plenty of tutorials in *The MagPi* for lots of different projects you can do with GPIO Zero.

Using two Pi devices together

As well as using GPIOs from your PC, you can connect a Pi Zero to another Raspberry Pi and use two sets of GPIOs together, controlled within the same script. This is similar to the principle of remote GPIO, which was covered in issue #60, but instead of using Pi devices over a LAN (local area network), where each Pi has to be running Raspbian from an SD card, this method again just requires a Pi Zero and a USB cable.

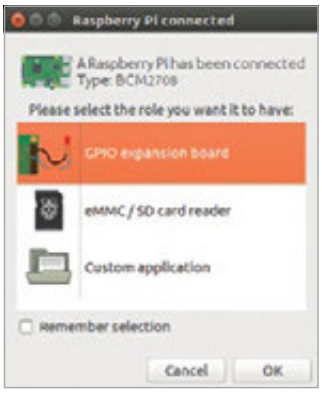
Now boot a Raspberry Pi (e.g. a Pi 3) running the latest Raspbian Stretch. The Pi Zero tool is not pre-installed in Raspbian, so you'll need to install it. Open a Terminal and type:

```
sudo apt install usbbootgui -y
```

When it's booted, connect a Pi Zero with a USB cable and this time, instead of setting the default pigpio address and pin factory, you'll define that within the code. Open a Python editor and type out:

```
from gpiozero import LED, Button
from gpiozero.pins.pigpio import
PiGPIOFactory
from signal import pause

pizero = PiGPIOFactory('fe80::1%usb0')

btn = Button(17)  # button on Pi 3
led = LED(17, pin_factory=pizero)
       # led on pi zero

led.source = btn.values

pause()
```

Here you have a button connected to GPIO 17 on the Pi 3, and an LED connected to GPIO 17 on the Pi Zero, which is connected via a USB cable. When the button is pressed, the LED lights up. Again, this is the simplest of examples, but it demonstrates the principle of controlling devices attached to two different Pis.



Above When you plug in a Pi Zero, you'll be prompted to use it as a GPIO expansion board

A better use for this would be to use two HATs or other add-on boards which use the whole pin header; for example, the Pi Hut 3D Xmas Tree and the Ryanteck SnowPi. The Christmas tree uses all available GPIO pins for its LEDs, and SnowPi uses nine of its own. But of course, you can simply place one board on the Pi 3, and another on the Pi Zero and when creating the relevant GPIO Zero device objects, use the remote pin factory for one:

```
tree = LEDBoard(*range(2, 28))
sp = SnowPi(pwm=True, pin_factory=pizero)
```

If you want to run a Python script on boot (using **@reboot** in cron), there's a trick you can use to wait for the Pi Zero to be booted over USB before you try to control it. The **PiGPIOFactory** connection will fail with an OSError if the Pi Zero is not available, so you can just keep trying that until it works using a try/except block:

```
pizero = None
while not pizero:
    try:
        pizero = PiGPIOFactory('fe80::1%usb0')
    except OSError:
        sleep(1)
```

...then proceed to create the **SnowPi** object on the pizero pin factory.

As you can see in our final code, we turn the tree star LED on, have the tree's red LEDs randomly flicker, and have the SnowPi's LEDs blink on and off.

Language
>PYTHON

NAME:
tree_sp_2.py

DOWNLOAD:
rpf.io/xmascode

USB MAGIC

The Pi Zero features a USB OTG (on-the-go) port, allowing you to boot over USB from a PC. Your PC sends the required boot firmware to the Pi over the USB cable, launching a mini version of Raspbian and booting it in RAM. The OS then starts the pigpio daemon, allowing 'remote' access over the USB cable.

INSTALLATION GUIDE

Discover how to install Raspberry Pi Desktop on a Mac or PC in *The MagPi* issue #60. Or watch our installation video online **magpi.cc/2wkkThL**

FREQUENTLY ASKED QUESTIONS

NEED A PROBLEM SOLVED?

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

Your technical hardware and software problems solved...

OLD HARDWARE

WHAT CAN I DO WITH MY OLD RASPBERRY PI?

Use it elsewhere

The original Raspberry Pi, while nowhere near as powerful as a Pi 3, is still no slouch at many computing tasks. If you're not using it for experimenting with projects, you can always turn it into a file server, or just use it for testing out simple circuits using Python.

Give it to a young maker

Know a child interested in computing but hasn't been introduced to Raspberry Pi yet? Gifting one might be the beginning of a long career in software engineering or otherwise. It would also give them a leg up on computer science lessons at school.

Recycle

Literally can't find a use for your Raspberry Pi? No one you can gift it to? No room to keep it? Do the responsible thing and sell it, or recycle it with an electronics recycler so the parts can go towards making other electronics without contributing to landfill.

BACKWARDS COMPATIBILITY

In general

Python is standard, and the GPIO pins (however many you have) all work the same way, so coding with the original Raspberry Pi is the same as programming with a Pi 3 or Pi Zero W. Even Minecraft Pi and Scratch 1.4 work on older Raspberry Pi models just fine.

Scratch 2.0

Unfortunately, Scratch 2.0 requires you to use a newer Raspberry Pi as it requires a lot of resources compared to Scratch 1.4. You can still use Scratch 2.0 on a web browser on your home computer, though.

HATs

HATs are backwards compatible with even the Pi B+ and A+, but not a Raspberry Pi with 26 GPIO pins like the original Model B and Model A. We still see some add-ons that use 26 pins to attach to the Raspberry Pi, though.

BUYING OLD PIS

Can I buy older models?

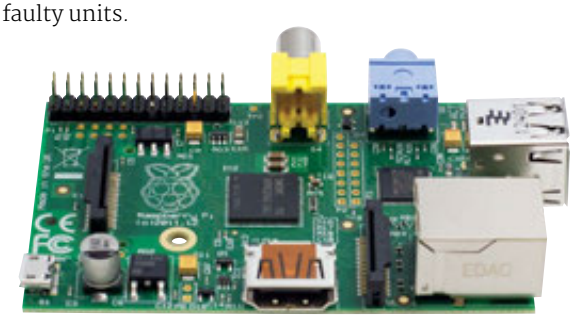
Yes! You can get brand new Model A+, Model B+, and Raspberry Pi 2 boards. We've also seen stock of Pi Zero v1.3 boards on Adafruit. They all tend to be priced low, although it does depend on where you shop.

Why would I want an older model?

Your needs may require a smaller Raspberry Pi (like the Model A+) or for your project to draw a little less power than a Raspberry Pi 3. You may even like the bulk-buy discount on the cheaper Pi boards. While we recommend getting a Raspberry Pi 3, that doesn't cover all cases.

Buying used Raspberry Pis

Like buying anything used, you need to understand the risks. Luckily, eBay has good buyer protection these days so if a Raspberry Pi you buy doesn't meet the description on the listing, then you should be able to get some money back. Raspberry Pi boards are fairly hardy, though, so you're unlikely to come across many faulty units.



FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

WHAT HARDWARE INTERFACES DOES IT HAVE?

Depending on the model, the Raspberry Pi has either 40 or 26 dedicated GPIO pins. In all cases, these include a UART, an I²C bus, an SPI bus with two chip selects, I²S audio, 3V3, 5V, and ground. The maximum number of GPIOs can theoretically be indefinitely expanded by making use of the I²C or SPI bus. There is also a dedicated CSI-2 port for the Raspberry Pi Camera Module, and a DSI display port for the Raspberry Pi LCD touchscreen display.

WHY IS THERE NO REAL-TIME CLOCK (RTC)?

The expectation is that non-network-connected units will have their clocks updated manually at startup. Adding an RTC is expensive once you factor in batteries, area, and components, and would have pushed us above our target price. You can add one yourself using the GPIO pins if you'd like an interesting electronics project.

CAN I ADD EXTRA MEMORY?

No. The RAM on the Model A, A+, B, B+, and Zero is a package on package (POP) on top of the system on chip (SoC), so it is not removable or swappable. The RAM on the Pi 2 and 3 Model B versions is on a separate chip on the bottom of the PCB, but 1GB is the maximum RAM that the SoC used by the Pi 2 and 3 Model B versions can support.

WHY DOESN'T THE RASPBERRY PI INCLUDE A CERTAIN PIECE OF HARDWARE OR A CERTAIN SORT OF PORT?

Our main aim is a charitable one: we are trying to build the cheapest possible computer that provides a certain basic level of functionality, and keeping the price low means we've had to make hard decisions about what hardware and interfaces to include.

WHAT IS ITS OPERATING TEMPERATURE?

The Raspberry Pi is built from commercial chips which are qualified to different temperature ranges; the LAN9514 (LAN9512 on older models with two USB ports) is specified by the manufacturers as being qualified from 0°C to 70°C, while the SoC is from -40°C to 85°C. You may find that the board will work outside those temperatures, but we're not qualifying the board itself to these extremes.

Privacy and security on, and with, your Raspberry Pi

Digital security is very important. With recent revelations over how people can exploit the vast majority of modern processors to access your private data, it's never been more important to learn how to improve your online security. While the Raspberry Pi may be immune to Spectre and Meltdown, it doesn't mean you shouldn't improve your own security.

So in this article, we'll help you to protect yourselves from nasty folks online, and even show how your Raspberry Pi can help you in this effort.

THE BASICS OF SECURITY

Before we secure your Pi, make sure to follow these basic security tenets

The most secure computer and network in the world won't be much use if your passwords can be cracked because you used password1 everywhere. Here's a refresher on how to keep your online presence safe.

ONLINE PRESENCE

TWEAK YOUR PRIVACY SETTINGS

Facebook, Twitter, and other social media services let you change the privacy of certain account aspects. Make as little public as possible.

LIMIT APP ACCESS TO YOUR ACCOUNTS

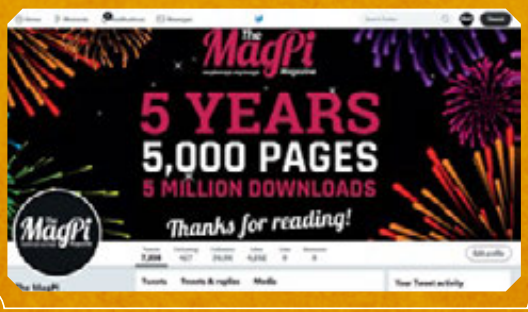
Some places let you log in via Facebook or Twitter, or will connect to your social media accounts. Regularly check what does and does not have access to your account, as old apps may end up compromised.

BE CAREFUL WHAT YOU SHARE

It may be common sense, but always think about what information you're giving out.

USE A PSEUDONYM

While perhaps an extreme step and not applicable on every social media network, it will aid your privacy and security.



PASSWORD ETIQUETTE

LEARN HOW TO CREATE A STRONG PASSWORD

We could write pages on this, so instead we'll point you towards this great guide: **magpi.cc/2rXYHcb**.

DON'T REUSE PASSWORDS

If one password is compromised, then all your accounts with that password are.

DON'T USE DICTIONARY WORDS

It might be common sense, but common cracking techniques start with dictionary words.

DON'T USE A COMMON PASSWORD

Check this list to make sure you're not doing anything silly: **magpi.cc/2rXjFI2**.

TURN ON TWO-FACTOR AUTHENTICATION

It may be a pain, but it means that even if someone works out your password, your more important accounts will be safe.

GET A PASSWORD MANAGER

Services like 1Password (**1password.com**) and KeePass (**keepass.info**) allow you to use unique, machine-generated passwords on accounts that you can then unlock with a master password.

BE PREPARED TO CHANGE IT

While the current school of thought is not to update passwords every 90 days any more, you may still need to change a password if there's a security leak or similar.

OTHER TIPS

CREATE A REDUNDANT EMAIL ACCOUNT

For accounts that aren't important or don't contain important information, use a second email account to sign up to them. That way, if someone cracks your secondary email, they won't have the keys to the kingdom.

KEEP YOUR SOFTWARE UPDATED

A common part of software updates include security fixes, so always update your software when you can.

BE AWARE OF PHISHING SCAMS

Always remember that hackers will send emails that look like they're from your bank or Facebook or similar asking you to log in. Check the sender and if in doubt, go to the original website.

ADD RECOVERY CONTACT INFO

Google recommends you add a phone number or alternate email to an important account so they can contact you if they think anything is wrong and confirm it is the real you.

SECURE YOUR RASPBERRY PI

Keep your Raspberry Pi, and any files on it, safe and secure

The Raspberry Pi is inherently fairly secure, especially if you're using the default operating system, Raspbian with Raspberry Pi Desktop. Raspbian is a form of Linux, which is much more resistant to hacking than Windows or macOS, and you're also unlikely to come across any viruses for it.

While secure, it doesn't meant Raspberry Pi and Raspbian are invincible. Here are some tweaks you can do to make sure your Raspberry Pi and everything on it are more secure.

IN RASPBERRY PI CONFIGURATION, YOU CAN TURN AUTOMATIC LOGIN OFF SO YOU NEED A PASSWORD TO GET INTO YOUR SYSTEM.

FURTHER READING:
As the pi username is the default, you may wish to create a new user account and remove the 'pi' user from being able to make changes to Raspbian. Find out how to do this here: magpi.cc/PiUsers.

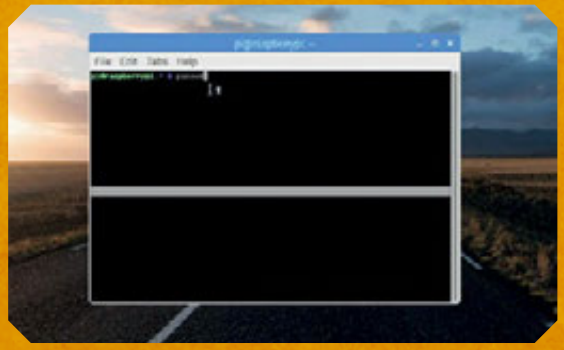
DEFAULT PASSWORD

As the Raspberry Pi is aimed at being educational, it makes sense that the default operating system Raspbian would be made easy to access by setting a standard username and password ('pi' and 'raspberry' respectively). For classroom environments where students are making games in Scratch or learning Python code, it's not an issue. If you're using the Raspberry Pi in a more sensitive environment, though, you should at least change the password for the default pi user.

Method:
Open the Terminal and type the following command:

```
passwd
```

You'll be asked to enter your current password (raspberry), then you can enter a new password. The screen will not show you typing in the password, so don't worry if no asterisks appear.

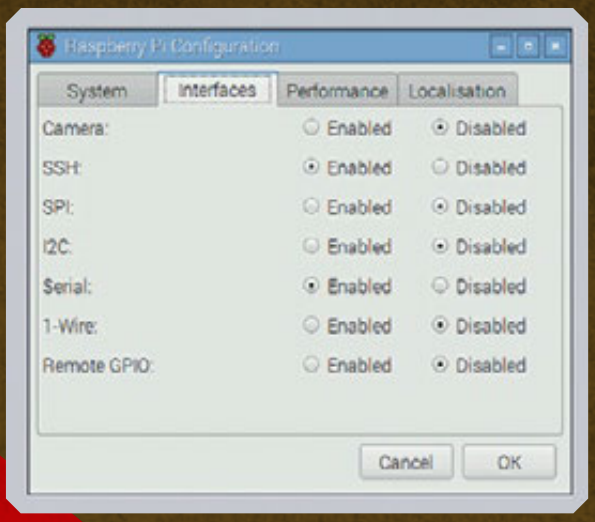


ALL ABOUT SSH

SSH (Secure Shell) is a way for computers to talk to each other over a network. It's like being able to open a Terminal window on another computer – although you have to know where the computer you want to talk to is, and the username and password.

On Raspbian this is made easy – by default all Raspbian-powered Pis use 'raspberrypi' as their name on a network. To SSH in, all you need to do is navigate to pi@raspberrypi and enter the password **raspberry**. SSH is turned off by default; however, if you do decide to turn it on, look at the above tip on how to change the password.

Turning SSH on or off:
From the Raspberry Menu, go to Preferences and find the Raspberry Pi Configuration option. From there, go to the Interfaces tab and click Enable next to SSH.



FURTHER READING:
You can find out more on how SSH works in Raspbian in the documentation, including how to turn it on when you don't have a monitor, here: magpi.cc/RaspbianSSH.

PI ENCRYPTION RESOURCES

Encrypt your Raspberry Pi's storage so that only you can access it

DISK ENCRYPTION
magpi.cc/SDEncrypt

Once you've set up Raspbian on your Raspberry Pi, you can encrypt the root partition (where the operating system is kept) to make sure that you need a password to access any of the files within.

You might be wondering why you would need to do this over just turning off automatic login – the reason is that the microSD card can be removed from the Raspberry Pi and the files read on another computer. If you encrypt the data, this can't happen.

ENCRYPTED OS
magpi.cc/KaliPi

If you need an all-in-one encrypted operating system with maximum security, try out Kali Linux. It's already extremely secure, and with a few modifications it can be fully encrypted as well!

Kali Linux is not your average Pi operating system, though – it's for computing professionals who need to test their networks for vulnerabilities. You won't be able to make the same kind of projects as you would in Raspbian.

WANT TO DO MORE ADVANCED SECURITY UPGRADES? Check out the full Raspberry Pi list of ways to improve Raspbian's security: magpi.cc/PiSecurity

SECURE YOUR COMPUTERS WITH PI-HOLE

Pi-hole turns your Raspberry Pi into an ad-blocker, making sure no malicious code from dodgy adverts gets to your PC

Online ads can be dangerous for your computer and your network. Malicious code can easily infiltrate ad networks and become a serious issue before the ads are eventually removed. While this kind of code is unlikely to affect your Raspberry Pi, it can infect other computers on your network. With some special software, you can create a network-wide ad block using just your Raspberry Pi, thanks to Pi-hole.

WANT TO WHITELIST WEBSITES YOU KNOW TO BE SAFE? Check out the details on how here: magpi.cc/2BZ8n64

INSTALL PI-HOLE

You'll need:
Raspberry Pi (any model)
Raspbian, connected to a network

>STEP-01
Get the software

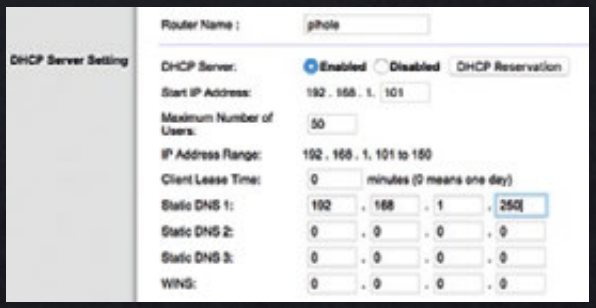
Pi-hole comes in a complete and easy-to-use installer that can be quickly used on your Raspberry Pi. Open a Terminal and use the command:

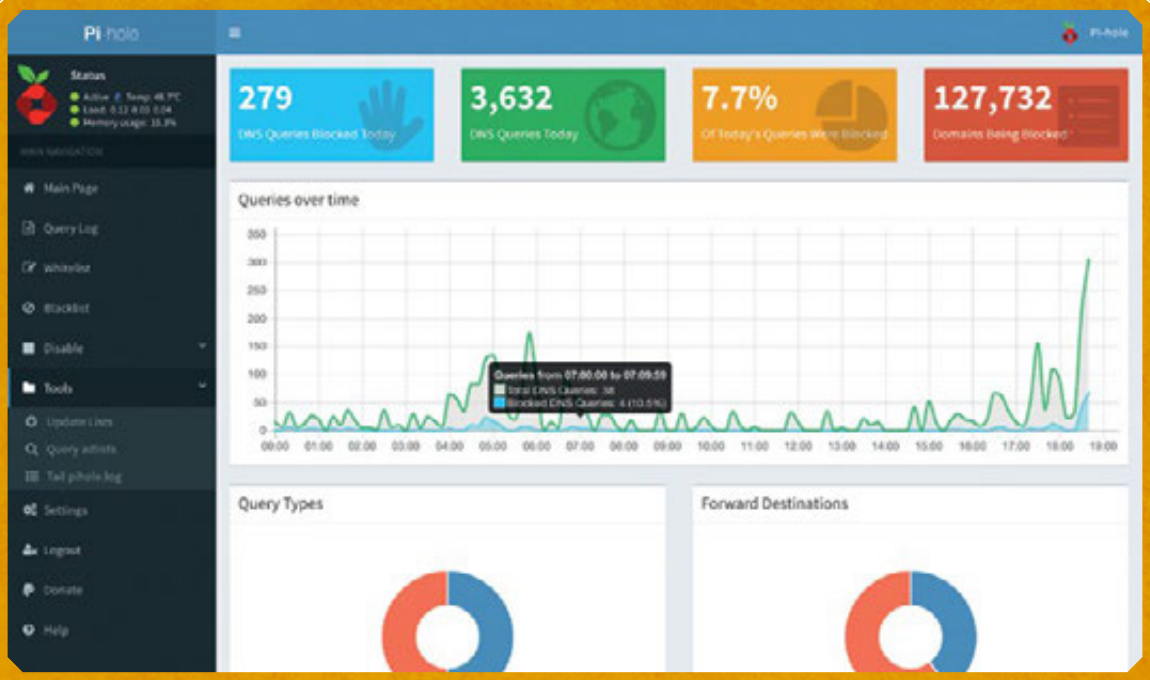
```
curl -sSL https://install.pi-hole.net | bash
```

This will download and then automatically install the Pi-hole software. You can do this in the command line (we recommend you turn off the desktop for your Pi-hole device anyway) as well as SSH.

>STEP-02
Configure your network

For the desired ad-blocking effect, you need to make it so all the computers on your network access the internet via your Pi-hole Raspberry Pi. To do this, you need to change the DNS settings on your router. This requires you to know your Pi's IP address on the network, so run the command **`ifconfig`** to find it.





Above Keep an eye on your home network traffic

BENEFITS OF PI-HOLE

While Pi-hole's primary use is as an ad-blocker, it also comes with other advantages to your network. Firstly, it blocks ads before they're even downloaded to your device; this frees up bandwidth so that pages will load faster, and also means you use up less of your data allowance (if you have one) day-to-day.

It also works with mobile devices and tablets, as the ads are blocked at the source rather than having to try to install resource-hungry ad-blockers on your phone.

Pi-hole has a very active community, so if you feel like you could give them a hand making the software better, head over to **pi-hole.net**.

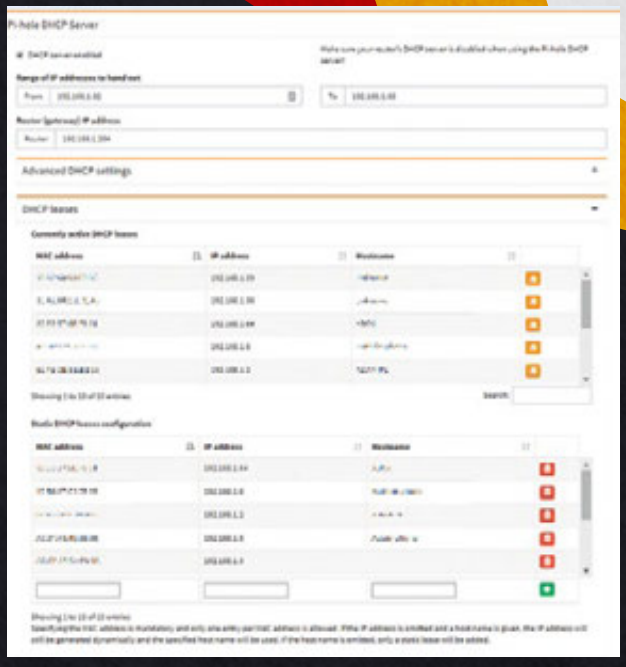
Your router will have instructions on how to access the admin settings, usually printed on a label on the back of the box. Log in to your router and find the settings for its DHCP server. This is used to automatically send details to your devices on how to connect to the internet. Change the value of the static IP address to that of your Raspberry Pi on the network. It might require a restart afterwards, but otherwise you're done!

>STEP-03

Configure your network

If you can't change the DHCP settings on your router, you can turn on Pi-hole's own DHCP server which will then automatically manage devices on your network. You can find this option, along with many other settings, by going to **pi.hole** in a web browser.

There are other more advanced/manual methods for sorting out the DNS settings as well, and they can all be found in the Pi-hole documentation here: **magpi.cc/3CPyydP**.



SECURE YOUR NETWORK WITH TOR

Create an automatic Tor router that anonymises your online location, allowing you to maintain complete privacy

Privacy on the internet is tricky. Everywhere you go you accumulate cookies and servers will log your IP. For vulnerable people, this can be dangerous. Tor allows you to almost completely mask your presence online, and here's how you can use a Raspberry Pi to make your entire network private.

You'll need:

Raspberry Pi 3
Ethernet cable

First of all you need to turn your Raspberry Pi into an Access Point, which lets other computers wirelessly connect to it and use it as a bridge to your router. Connect your Pi to your router via the Ethernet cable, and leave the wireless LAN connection alone. To make the wireless LAN on the Pi 3 act like an AP, just execute the following commands in the Terminal:

```
git clone https://github.com/unixabg/RPI-Wireless-Hotspot.git
cd RPI-Wireless-Hotspot
sudo ./install
```

You'll need to agree to some terms and DNS options, but say no for the wireless LAN defaults. It will then prompt you to enter details for the wireless network you want to create and connect to. For the final two options, choose no and your Raspberry Pi will reboot.

Now we can install Tor on your Pi. First, update Raspbian with **sudo apt-get update** and then **sudo apt-get upgrade** to make sure everything is properly installed from the previous step, and then install Tor with:

```
sudo apt-get install tor
```

Once it's installed, you'll need to configure it. Open the config file with:

```
sudo nano /etc/tor/torrc
```

...and add the following just below the first set of comments:

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1
```

Next, you need to dump all the old rules from the IP table:

```
sudo iptables -F
sudo iptables -t nat -F
```

Now set the DNS to route through Tor:

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --to-ports 53
```

Route all TCP traffic through Tor:

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 9040
```

Make sure to check all these new routes with:

```
sudo iptables -t nat -L
```

Finally, save our new rules to the NAT table:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Once that's down, you should create a log file to track your Pi's Tor usage. Create an empty file and then change its permissions with:

```
sudo touch /var/log/tor/notices.log
sudo chown debian-tor /var/log/tor/notices.log
sudo chmod 644 /var/log/tor/notices.log
```

Once that's all done, you can start the Tor service with **sudo service tor start**. You can also then check it with **sudo service tor status** to make sure it's all running properly.

Finally, change the service to start on boot with:

```
sudo update-rc.d tor enable
```

KNOW HOW!

The great folks from the tech show *Know How* have created a similar setup in the past – check out their video for more info: magpi.cc/2sj1wF3

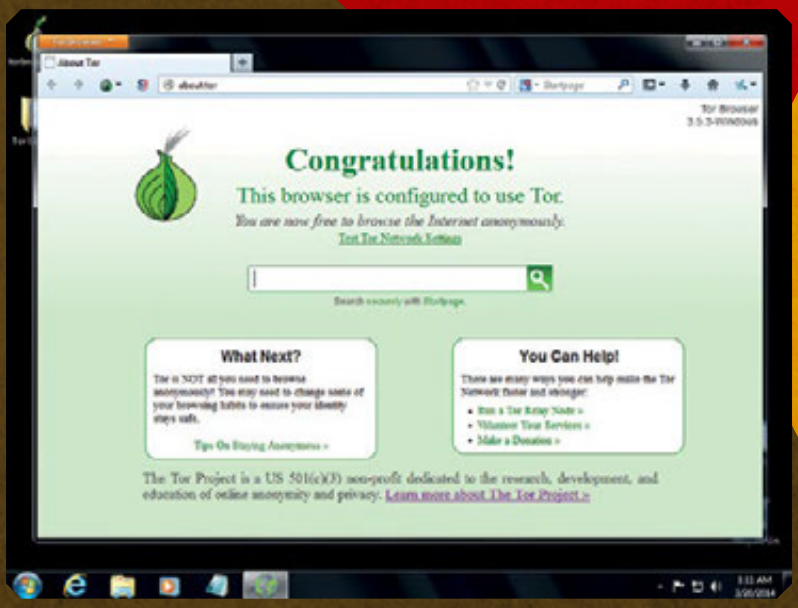
HOW TOR WORKS

Relay networking and how it keeps you safe online

Tor's main function is to protect your anonymity online. The way it does this is by routing web traffic through several encrypted servers around the world – in a sense, it bounces the contents of a webpage around different computers across the globe to then be read by your computer. A bit like in a film.

This allows users to conceal their location, and will stop anyone monitoring your network from figuring out what you're using the internet for. This can be anything as simple as concealing what you're getting a network engineer for their birthday, or for more serious uses such as staying hidden from an abusive partner.

Tor is not invincible though, and should be used with caution. Not every website allows access via Tor either. If you don't want to have your entire network running under Tor, you can always use the Tor Browser: magpi.cc/2smh6iY.



Above The Tor Browser is an alternate way of staying hidden on the internet instead of encrypting your entire network

FURTHER READING:

Want to learn more about Raspberry Pi and computing security? Check out these links.

GET SAFE ONLINE

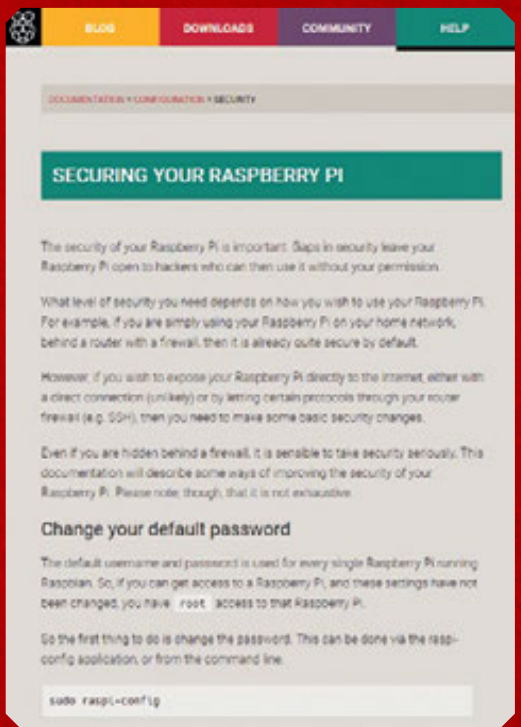
getsafeonline.org

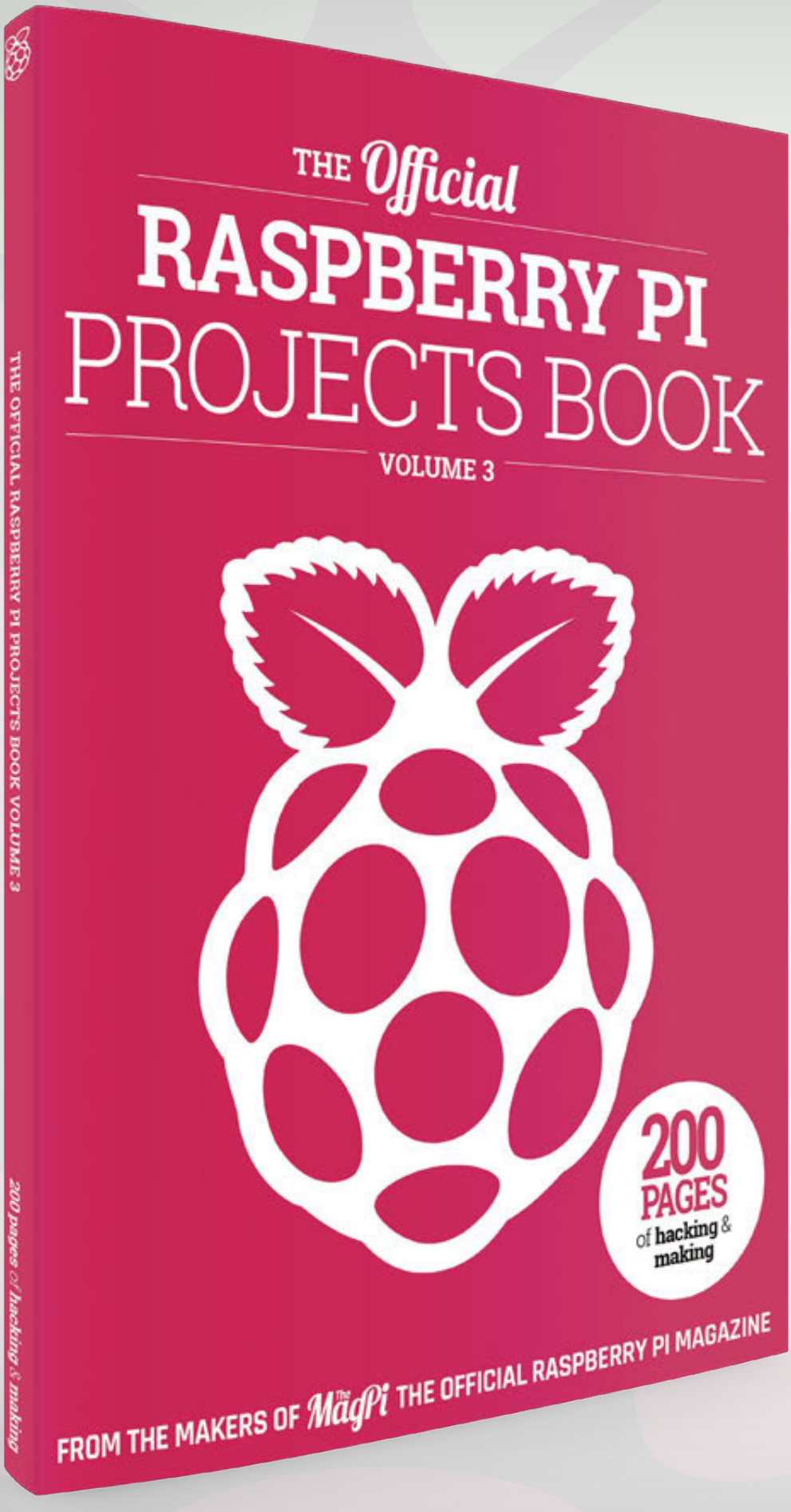
A general advice website that gives privacy tips on just about every part of your online experience – it even has sections on how parents can help safeguard their kids. Privacy and security online is always evolving, so it's worth checking regularly to see how you can improve your online safety.

SECURING YOUR RASPBERRY PI

magpi.cc/PiSecurity

Official advice from Raspberry Pi on ways you can improve the security on Raspbian and your Raspberry Pi. It covers some of the stuff we mentioned on previous pages but goes into a lot more detail, as well as offering more advanced methods of controlling your security on the Pi.




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

Maker Says

"Nova is the simplest plug-and-play modem for instant connectivity"
Hologram

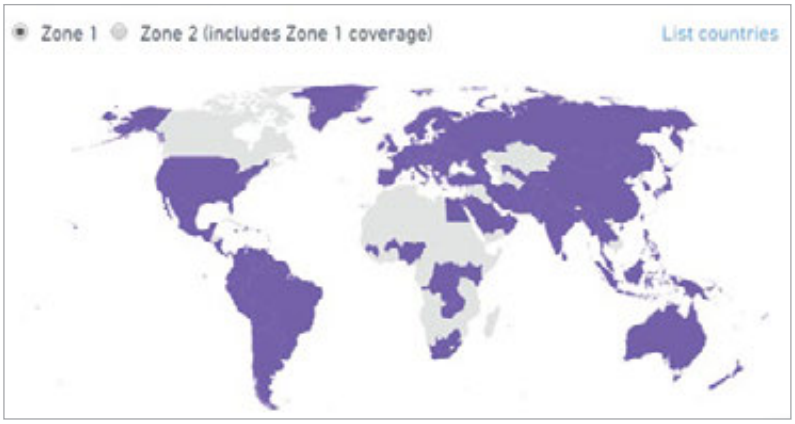
HOLOGRAM NOVA

Connecting a Raspberry Pi to a mobile network anywhere in the world is something **Rob Zwetsloot** is very interested in

In past issues we've talked about 'mobile' Raspberry Pi devices a few times in various contexts. However, one of the concerns with creating a truly mobile Raspberry Pi project is giving your Pi an internet connection wherever you go.

You could always hack about with a mobile hotspot on your phone, but cellular modems have always been an alternative. The Nova is one of the first made specifically with Raspberry Pis in mind, and it works really well.

The Nova itself is a very simple device. It's a USB dongle that you can slot a SIM card into (supplied with the Nova) and then plug into your Raspberry Pi. You can disassemble it to access the SIM card slot and swap between two included antennas depending on how much space you have and the strength of the signal where you're planning to use it.



Software & subscriptions

The Nova isn't just plug-and-play, though – you won't be able to just use it like a WiFi dongle on a Raspberry Pi. Luckily, you only need to install one bit of software which is available online (Hologram recommends you have a pre-existing network connection to set up your Nova before using it) to get yourself connected, with another piece of software required to use the SMS features.

While getting your Raspberry Pi to connect to a mobile network through the Nova is easy and quick, it doesn't mean you'll then have instant internet access. For that, you need to set up a subscription to Hologram's data plan.

It's easy enough to activate your SIM card on there. You just create an account, put in your SIM card's

Related

LEARN MORE ABOUT NOVA

You can find a tutorial on how to set up and use your Hologram Nova in issue 64 of *The MagPi*.

magpi.cc/2AqrV6q

magpi.cc/2C5YF1z

£36 / $49

serial number, and select from a few data plan options. By default, you can select from a couple of pay-as-you-go options that charge you by the megabyte (and SMS), with differences in price depending on what countries you'd like the Nova to work in.

> "The Nova is one of the first made specifically with Raspberry Pis in mind, and it works really well"

There's also a monthly contract plan, but it's currently only available to people in the US.

Pricing and regions

The cost of Nova is fairly cheap at $0.60 (£0.43) per minute for most of the world, along with $0.19 (£0.14) for each SMS. While it would be pretty expensive to use it to stream YouTube videos on the go, that's not really what it's meant for – it's more for quick data uploads and location pings for more advanced projects. Even for photos it's not too bad, especially as you can tweak the size and quality of photos taken by the Raspberry Pi Camera Module to keep the file size (and therefore upload price) fairly fixed and within a project's budget.

You can upgrade the plan from 'most of the world' (Zone 1) to 'almost all of the world' (Zone 2), which notably includes Canada among other countries. The price bump is about 33%, going from $0.60 to $0.80 (£0.58) per megabyte and up to $0.30 (£0.22) for a text. It's not huge but depending on the amount of data you're planning to use, it can be a significant jump. It also mostly affects less developed countries, although it's unlikely this would be their first port of call for internet access.

With the contract, you can save a lot of money on Zone 1 data (100MB for $15/£11) and some money on Zone 2 (100MB for $60/£43), although the price of SMS texts remain the same. Again, you need to be in the US to get the contract, but it's well worth it for the Zone 1 savings.

Usage and connection

While we weren't able to test it in every country, in the UK and the US we've found that the Nova works great. Focusing on sending small packets or limited file sizes of data is the way to go, although you're sometimes at the mercy of the mobile signal so we recommend making sure you check signal strength before deploying.

It's frankly a wonderful little product, and although it may be aimed more at a developer/business market, it ticks the right boxes for those specific uses.

Last word

The Nova is a great addition to the line-up of Raspberry Pi accessories that can really help improve some of your mobile projects.

★★★★★

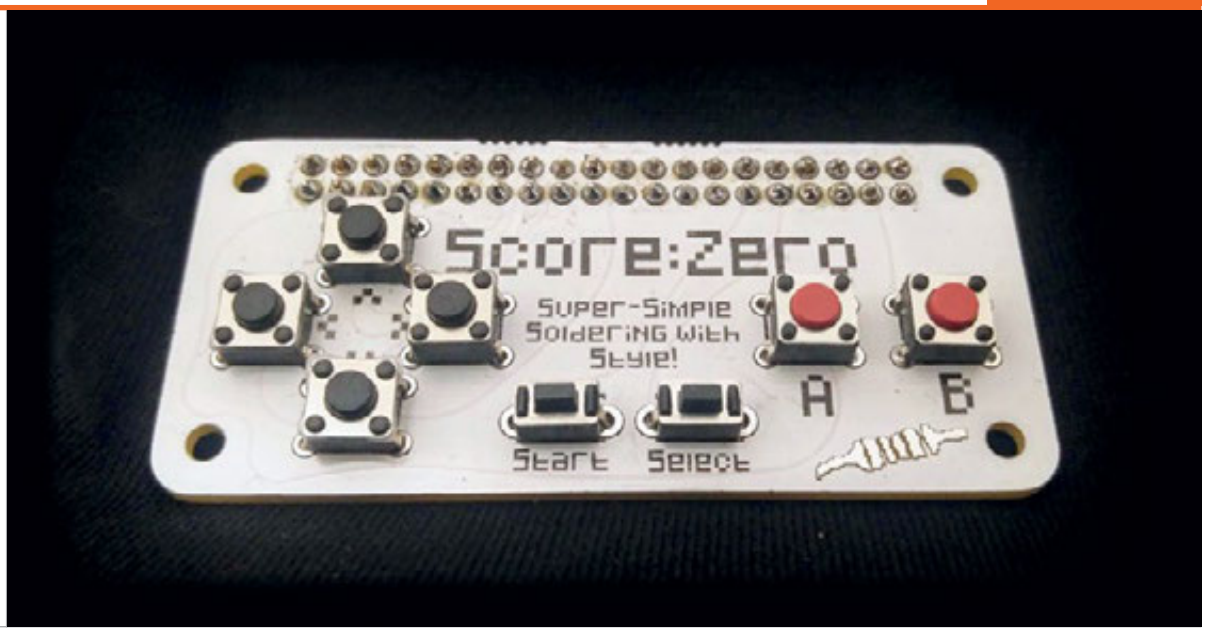
magpi.cc/2BNV5xw

£5 / $7

Maker Says

"Score:Zero is a super simple soldering project for your Raspberry Pi!"

The Wonky Resistor Co.



SCORE:ZERO

A tiny controller that also helps you practise your soldering

Ever since the Pi Zero came out, its use for video game console projects has been widely explored. From installing them into classic controllers to hacking Happy Meal toys into working Game Boys, it's been great to see just how inventive people have become with them. Sometimes though, you need something a little more practical, like the Score:Zero.

It's a plain, white PCB the size of a Pi Zero that comes with a 40-pin GPIO header and buttons. You solder these on and then place it on a Raspberry Pi's GPIO pins – a nice and simple concept. As there's not much to solder, it is fairly quick to do and is a good way to practise soldering a 40-pin GPIO header – in this case a female one. We did find some of the buttons a little small and fiddly, though, so make sure you're at a clear work area when opening the box and getting everything ready, lest you lose a button.

Small hands

Once you've soldered it up, you need to install some software to get the Score:Zero working on your Raspberry Pi. The Wonky Resistor website has a great little guide to help you do the soldering and installing – in fact, just one line of code will get you some example code to use with it.

The code is all done with GPIO Zero so you can easily put it into your own game projects without having to download any of the code. However, you can use the examples along with some other software to quickly use the Score:Zero as a keyboard or mouse.

With it sitting on top of a Pi Zero, we found the Score:Zero a bit awkward to use. The diminutive size of both hampered proper usage and the microswitches for the D-pad weren't very comfortable to use. On a full-size Raspberry Pi, though, the story was very different – we thought we'd have trouble reaching the buttons at all, but it was generally a lot more comfortable to use. For those with smaller hands who might get the most out of learning to solder and code with it, the Pi Zero size probably won't be too much of an issue, though.

Last word

A nice quick project that, while a bit small for adults, will be perfect for younger makers getting their start in soldering and programming.

★★★★☆

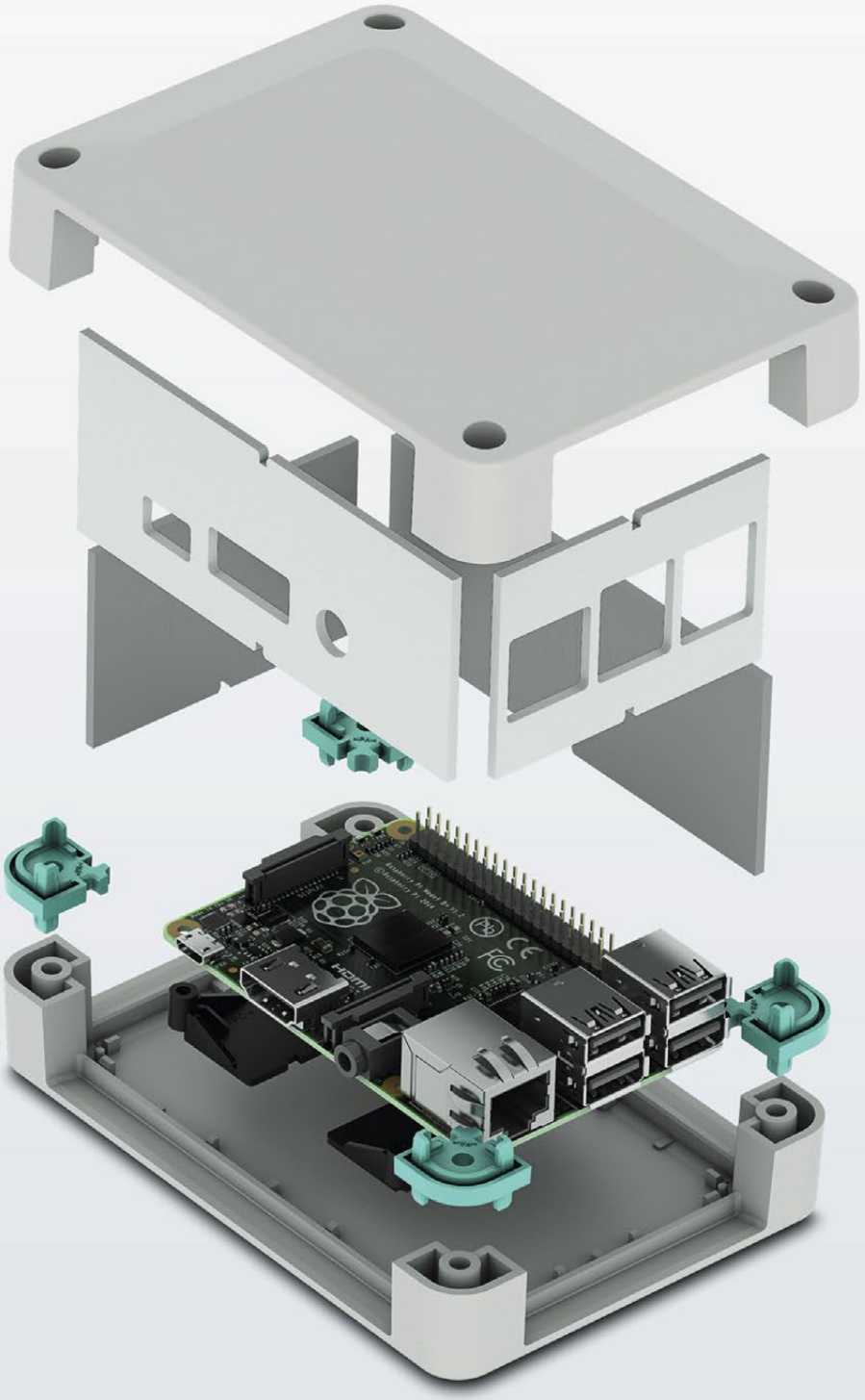
Related

JOY BONNET

A more expensive product, the Joy Bonnet features a mini joystick and six buttons. It comes fully assembled so no soldering is required.

£16 / $15

magpi.cc/2qaSfIT



Raspberry Pi finds a new home

Versatile enclosure for Raspberry Pi B+

The new UCS-RPI Universal Case System for Raspberry Pi B+ has pre-milled side walls for easy access to the I/O and power inputs. It is available in black or grey, in two size configurations and with glue dot location posts to secure the single board computer to the case.

For additional information call 0845 881 2222 or visit

phoenixcontact.co.uk/RPIBC

PHOENIX CONTACT
INSPIRING INNOVATIONS

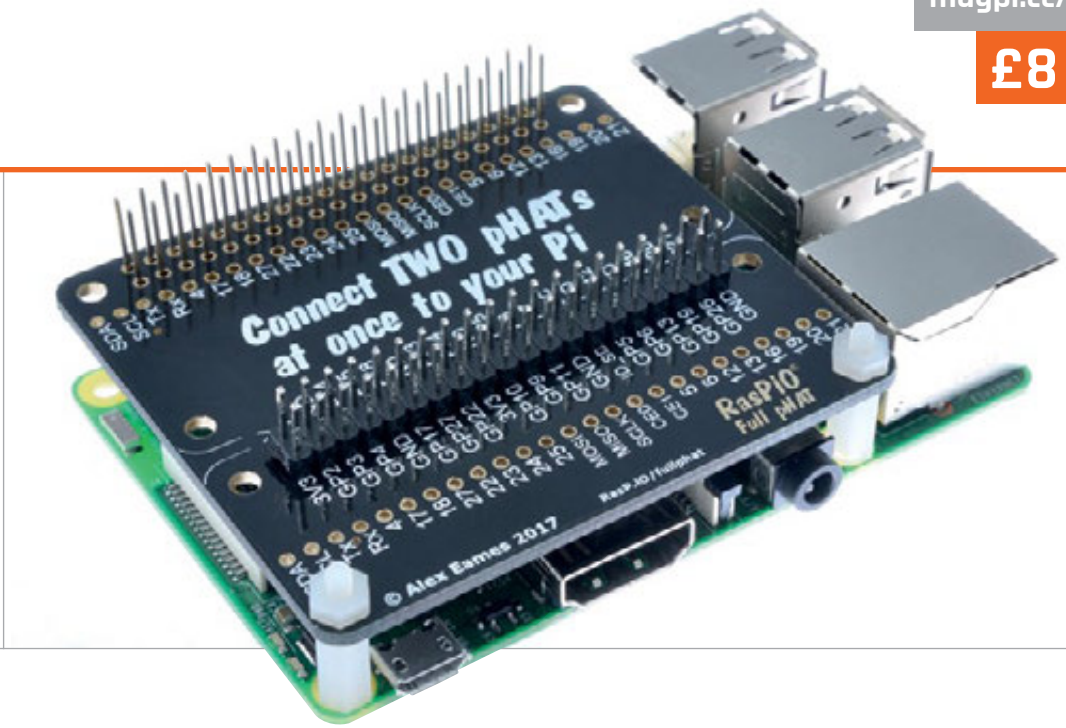
magpi.cc/2ku3dfc

£8 / $11

Maker Says

"Connect two pHATs to your Pi and still use spare GPIO ports

RasPiO



FULL PHAT

Have double the fun by plugging in two pHATs at once. By **Phil King**

You may recall that we reviewed Pimoroni's pHAT Stack last issue. The latter is a monster of a breakout board that enables the connection of up to five pHAT boards – or three HATs – to one Raspberry Pi.

While it works very well, we did mention that it might seem like overkill to some people. In which case the Full pHAT might well prove a more practical alternative with its smaller form factor. Fitting neatly on top of the Raspberry Pi, it allows you to connect two pHATs, or a pHAT and a HAT.

Well, it does once assembled. You first need to solder on a female stackable 40-pin header, and a male header. Two nylon standoffs are also supplied to provide a secure fit on top of the Raspberry Pi. For a low-profile project, you could even solder the Full pHAT directly to a Pi Zero and eliminate the main header altogether.

The GPIO ports (all bar GPIO 26) are broken out twice more in two single rows of holes, labelled with special functions (e.g. SDA, SCL, Tx, Rx) where applicable. In addition, there are three holes for 5V, 3V3, and GND. So there's plenty of scope for soldering wires directly, or via added pins, to connect various electronic components and sensors.

The two main GPIO headers are also clearly labelled with their GPIO numbers, which is a nice touch and reminiscent of RasPiO's Portsplus key-ring.

Spare ports

Once assembled and mounted on the Raspberry Pi, we found it easy to swap pHATs and HATs in and out – such as a Micro Dot pHAT and Unicorn HAT HD to simultaneously show scrolling text messages and a graphical display.

As with the pHAT Stack, however, one caveat is that you need to avoid combining two add-on boards that use the same GPIO pins (unless they're I²C with different addresses). Again, Pimoroni's 'pHAT Stack configurator' online tool (**magpi.cc/2BH6TSc**) comes in handy here – assuming you're using Pimoroni boards. Alternatively, you can just check which pins they use at **pinout.xyz**.

Related

PHAT STACK

Need to connect more add-on boards to your Pi? The pHAT Stack lets you use up to five pHATs, three HATs, or a combination.

£14 / $16

magpi.cc/2mcr2Gt

Last word

Fitting snugly on top of the Raspberry Pi, the Full pHAT is useful for combining two pHATs (or a pHAT and HAT) and also as a breakout board for projects. To this end, the clear labelling of all the GPIO ports is a great help, as is the provision of two extra rows of holes.

★★★★☆

magpi.cc/2sdQLDX

£148 / $169

Maker Says

"Designed with one goal in mind: absolute best S/PDIF output

Allo

DIGIONE PLAYER

Clever engineering and top-end components

For many, the audio on the Raspberry Pi's HDMI port – or even the stereo analogue audio, accessible on the multifunction 3.5 mm jack – is good enough. For others, 'good enough' is never good enough – and it's for these people that the Allo DigiOne Player has been designed.

Based around the DigiOne S/PDIF HAT (available separately, £86/$99), the DigiOne Player is designed to provide a peak-quality digital audio signal. While many will argue that digital audio is digital audio, being made up of zeros and ones which either arrive at the other end perfectly or don't arrive at all, audiophiles claim otherwise – and in particular, point to 'jitter,' the deviation of a period signal from its ideal timings, as a source of audible distortion.

The DigiOne, Allo claims, solves that problem. Using two on-board clocks as stable timing sources, as well as galvanic signal and power isolation and a variety of other engineering tricks, Allo claims to have reduced the jitter of the digital audio signal to 0.6 picoseconds – as close to perfect as practically possible.

That signal is available on either of the DigiOne's two outputs: an RCA jack and a BNC connector. In practice, there's no difference between the two, bar the BNC connector being more resistant to accidental disconnection. What you won't find is an optical output, a deliberate design choice by Allo to avoid what it claims is unacceptable 4 nanosecond (4000 picosecond) built-in jitter.

Complete bundle

Purchased as the Player bundle, the DigiOne HAT comes with a Raspberry Pi 3; 8GB SD card with a choice of DietPi, Volumio, or Max2Play audiophile-centric distributions pre-installed; a 5 V 3A power supply; and an attractive acrylic case.

Using the DigiOne Player is as simple as connecting a power supply and linking the HAT to your

digital-to-analogue converter (DAC), with the tested Volumio software providing a browser-based interface for control as well as acting as a streaming transport device for third-party software. When running, the audio quality from the DigiOne Player is undeniably high – but it is more obvious to scientific instruments than the fallible human ear.

Last word

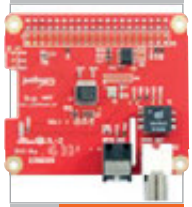
The DigiOne Player is a clever piece of engineering, but the problem it solves is arguably not worth the purchase price.

★★★☆☆

Related

JUSTBOOM DIGI HAT

It lacks the advanced features of the DigiOne, but the JustBoom Digi HAT provides an S/PDIF on optical and coaxial outputs for a fraction of the price.



£30 / $35

magpi.cc/2lMEYFx

Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.

Sign up today: rpf.io/makerspace

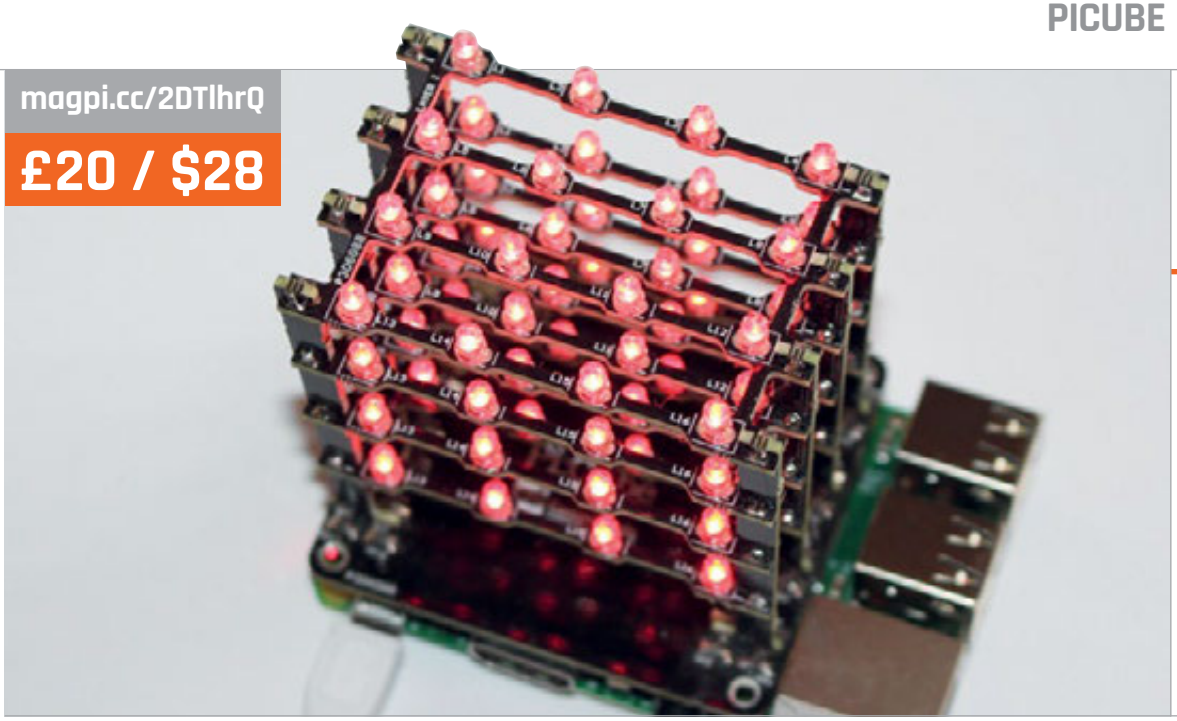
Raspberry Pi

The Raspberry Pi Foundation. UK registered charity 1129409

#RPiLearn

magpi.cc/2DTlhrQ

£20 / $28



Maker Says

"Strengthen your logic by typing complex code to draw patterns

SB Components

PICUBE

Add a tower of 64 LEDs to the top of your Pi and create hypnotic sequences. By **Sean McManus**

PiCube is a 4×4×4 grid of LEDs to sit on top of your Raspberry Pi 2, 3, or Zero. That means there's a total of 64 lights, many more than you could connect directly to the GPIO pins.

The secret is multiplexing: the pins are arranged in columns and layers. All the LEDs in the same column share a common anode, and all the LEDs in the same layer share a common cathode. That means each horizontal layer has the same lights switched on as the others, or is completely dark. As a result, PiCube isn't suitable for games or projects that would need the lights to be independently controllable. It's fun to experiment with, though, and could provide a highly visual way of teaching nested loops and nested lists, or experimenting with logic.

You don't need to install any software because you can control the PiCube using the RPi.GPIO Python package to turn pins on or off. The documentation shows which pins control which layers or columns, and the LEDs are numbered on the board to help you find your way around.

Cubist light

There are three mesmerising demos to download, showing different animations you can create. IDLE complained about inconsistent indentation in the first one, and wouldn't run it. The demos are unexpectedly in Python 2 (rather than 3) because of some old-style print statements, too. This all took seconds to fix, but it was disappointing that it didn't work immediately. A simple 'Hello world!' demo explaining how the layers work would have been helpful too.

PiCube comes flat-packed, ready for you to build. There are 64 LEDs to solder, as well as soldering the layers to the pillars, the pillars to the shield, and the shield to the header. We make that 144 joints, pins, or lights. That makes this a good project for those who are looking for a substantial soldering project with a bit more programmability than the usual LED displays.

We experienced a dead LED on two of the three units we tested. Our units arrived pre-soldered (thank you!), so we don't know how many knocks they took in transit, and this might not be typical. You can test LEDs before soldering them to avoid problems, as the manual recommends.

Last word

A nice tech demo that is fun to experiment with and a highly visual teaching aid. It has limited applications beyond education, though.

★★★★☆

Related

PIGLOW

A spiral nebula of 18 individually controllable LEDs supported with a Python library. Can be used for status output or purely for decorative patterns.

£9 / $12

magpi.cc/2nxGOvl

magpi.cc/2nvvkbI

£14 / $19

Maker Says

"Define complex on/off sequences with a simple script"

UUGear

WITTY PI MINI

Want to save power, or run a program intermittently? Witty Pi Mini turns your Pi on and off automatically. By **Sean McManus**

Witty Pi Mini adds a real-time clock to your Raspberry Pi so you can switch it on or off automatically, at set times.

To set it up, you can create a simple configuration file using a text editor. You specify the start and end time for the sequence, and then say how long the Pi should be switched on for, and then switched off for. The on/off cycle loops until the end time is reached. If you prefer, there's a nice online generator you can use to create the code, which includes a simulator showing you when your Pi would switch on and off using it.

To install your configuration file, you use a Bash script which can also show you the next scheduled shutdown and startup times, or reset the Witty Pi Mini. A selection of example scripts is provided, such as 'turn on for five minutes every 20 minutes'.

You can also use wildcards to set the shutdown or startup times, so that your Pi turns on at quarter past every hour, for example.

Power up

There's a graphical interface version of the Bash script, which might provide some comfort to those who prefer not to use the command line. It was extremely slow on our Pi Zero WH, though, with a significant delay after clicking a button. It takes several seconds to commit a time change, so setting the clock with second-accuracy was hard. It also appeared to lag a few seconds behind the time shown in the command line. There are a few usability niggles too, so we found we preferred the Bash script with its simple menu.

The real-time clock is powered using a capacitor, so the clock runs for about 17 hours without an external power supply.

The Witty Pi Mini uses four GPIO pins, two of which are customisable (with some soldering). You might be able to combine it with other HATs, and should be able to use it with your own electronics projects. A stacking header is available separately to enable you to access all of the GPIO pins.

When the Pi shuts down on schedule, it's merciless: whatever you're doing, it's going down. Make sure you remember if you set a shutdown schedule so you don't get any unpleasant surprises.

Related

WITTY PI 2

Its 'big sister' Witty Pi 2 keeps time for up to a year without external power. It can output 3A current to power peripherals too.

£17 / $24

magpi.cc/2nxB7gU

Last word

Witty Pi Mini is a great tool if you need your Pi to do something on a schedule. The Bash script, online generator, and manual will get you and your Pi running like clockwork.

★★★★☆

HackSpace

TECHNOLOGY IN YOUR HANDS

THE NEW MAGAZINE FOR THE MODERN MAKER

SUBSCRIBE AND SAVE UP TO 35% on the cover price

ISSUE #04 OUT NOW

hsmag.cc

Available on the App Store

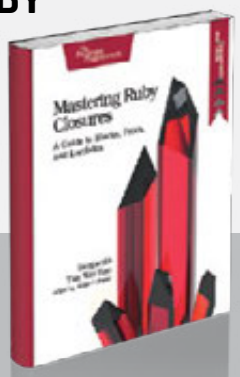
GET IT ON Google Play

RASPBERRY PI BESTSELLERS

PRAGMATIC EXPRESS

Introductions to extending interesting languages make for some of our favourites in Pragmatic's excellent Express series.

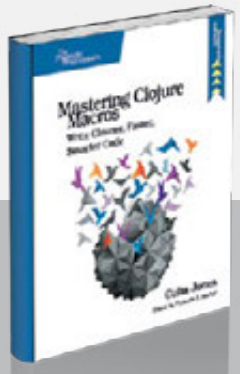
MASTERING RUBY CLOSURES



Author: Benjamin Tan Wei Hao
Publisher: OReilly
Price: £21.50
ISBN: 978-1680502619
magpi.cc/2GE91cz

An easy introduction to Ruby's secret weapon. The difficulty ramps up steadily, but examples and tests keep you on track, giving you closure understanding that you can take on to Haskell, JavaScript, or Elixir.

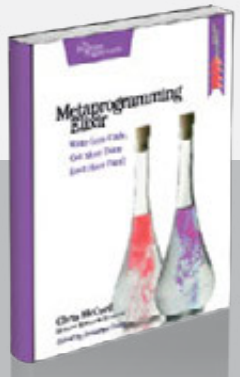
MASTERING CLOJURE MACROS



Author: Colin Jones
Publisher: OReilly
Price: £13.50
ISBN: 978-1941222225
magpi.cc/2GCWzda

Clojure's powerful macro system makes "hard things possible and normal things easy". Jones's easy style makes for a painless intro, with intriguing suggestions for when to use them, but also how to use new your powers wisely.

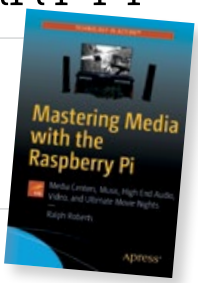
METAPROGRAMMING ELIXIR



Author: Chris McCord
Publisher: OReilly
Price: £13.50
ISBN: 978-1680500417
magpi.cc/2GCX6Mc

Not for the Elixir beginner, but a terrific introduction to accessing Elixir's Abstract Syntax Tree, and understanding metaprogramming through Elixir's surprisingly Lisp-like features. Useful examples include an internationalisation library, and an HTML chapter.

MASTERING MEDIA WITH THE RASPBERRY PI



Author: Ralph Roberts
Publisher: Apress
Price: £23.99
ISBN: 978-1484227275
magpi.cc/2GAKncO

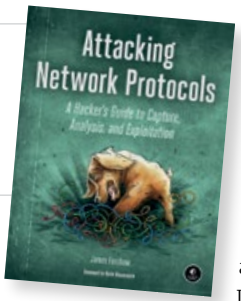
Despite all of its wonderful educational uses, it's as a media centre that many Raspberry Pis are destined to spend their days. And why not? It's more satisfying – and cheaper – than buying one off the shelf. Roberts provides a comprehensive guide to setting one up – yes, all of the information can be gathered from the web, but delegating that effort to Ralph Roberts will make your life easier.

There are several things to consider to get the correct setup, even at the hardware level. From careful consideration of getting the correct power supply – something which has occasionally tripped up new (and not so new) Pi users – and powering peripherals, to CEC over HDMI. The book is standardised on one digital-to-analogue converter (DAC), and doesn't examine closely any of the other DACs available. There are other quirks, too, but the choices made are all practical.

This is mostly a guide to Kodi, on LibreELEC – a sensible choice, and one with plenty of expansion options. These add-ons are also covered, and the guide is pitched such that you could give it to a fairly non-technical relative and be reasonably hopeful that they'd need no additional help! A useful resource.

Score ★★★★☆

ATTACKING NETWORK PROTOCOLS



Author: James Forshaw
Publisher: No Starch
Price: £39.99
ISBN: 978-1593277505
magpi.cc/2GyErRL

Forshaw is quite well known for finding a number of Microsoft security flaws, earning the first $100 000 bounty from Microsoft for using architectural-level flaws to bypass platform defences. Here he turns his attention to the kinds of network weaknesses which a proliferation of IoT devices will bring to homes everywhere.

Networking basics and capturing traffic with Wireshark open the book. The reader's hand is not held, as we gallop through the basics, and on to the binary protocol structures that underlie network traffic, then practical approaches to more difficult traffic capture, and a more detailed look at packet analysis, and some achievable reverse engineering.

Now you're ready to find (and even exploit) some vulnerabilities, with practical chapters using .NET Core. The chapter on network protocol security should give you a fresh appreciation of cryptography, and re-implementing them (in Java or C this time) may leave you nervous of trusting anything that you receive – time to dive into the root causes of vulnerabilities and gain a much better understanding not just of these but, thanks to Forshaw's examples, a better practical understanding of many aspects of code execution. Challenging in many ways, and rewarding in equal measure.

Score ★★★★☆

ELEGANT SCIPY

Authors: Juan Nunez-Iglesias, Stéfan van der Walt & Harriet Dashnow
Publisher: O'Reilly
Price: £31.99
ISBN: 978-1491922873
magpi.cc/2GC1jjp

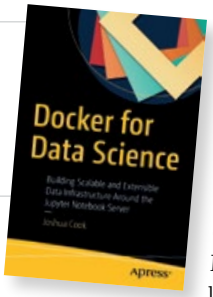
SciPy, NumPy, Pandas, and scikit-image give not just the data scientist, but biologists and other users of image and signal processing, almost everything they need to do great work. Something that we may have mentioned in previous reviews here. This book is not just about the tools and how to use them – but how to use them with code that is clear, concise, and efficient.

Along the way, this enjoyable and occasionally entertaining guide revels in the joy of elegant code, whilst demonstrating quite practical examples across a range of sciences – starting with NumPy's N-dimensional arrays with genome data, then sticking with the gene expression data, and moving on to quantile normalisation, then images as NumPy arrays.

Fast Fourier Transform with SciPy takes in birdsong, moon landing pictures, and radar data; the three authors ensure a diversity of examples, yet write with a unity of purpose and even humour. Chapters on sparse matrix problems and linear algebra present several exercises for you to stretch the SciPy skills you've gained so far, and the chance to consider PageRank's usefulness in examining ecosystems. The Toolz streaming library for large data sets rounds off a genuinely essential introduction to SciPy.

Score ★★★★☆

DOCKER FOR DATA SCIENCE



Author: Joshua Cook
Publisher: Apress
Price: £19.99
ISBN: 978-1484230114
magpi.cc/2GDzcjs

A book straddling many fields – aiming to teach Docker and its ecosystem, but using the field of data science throughout. Jupyter Notebook Server is the tool of choice, and a Jupyter/SciPy image on AWS. Docker and Jupyter are then brought together – first with an example program in C.

That done, we jump forward 30 years to IPython and NumPy, and spend the rest of the book more comfortably programming interactively in Python. Jupyter and Docker alternate as the main focus, meaning you get just enough of each, rather than the in-depth look at SciPy in the O'Reilly book reviewed above. If you need both depth and breadth, start with this one. Following on from data persistence with PostgreSQL, the chapter on Docker Compose (for multi-container applications) brings in examples with Redis, and with MongoDB, as well as scaling to larger AWS instances.

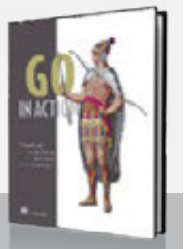
Cook is an experienced teacher, with multiple interests and a clear style – all of which enables him to pull off the tricky juggling act of helping you to overcome Python-based data science challenges in the 'infrastructure as code' world without this once seeming to be a book designed to squeeze as many hot topics in as possible. Quite an achievement!

Score ★★★★☆

ESSENTIAL READING: GO LEARNING

Not yet a decade old, Google's concurrency-friendly, typed language is not one you can ignore.

Go In Action



Authors: William Kennedy, Brian Ketelsen & Erik St. Martin
Publisher: Manning
Price: £27.99
ISBN: 978-1617291784
magpi.cc/2GEBboo

Gives you the why as well as the how, for real understanding in addition to practical knowledge – and is very good on concurrency.

Go In Practice

Authors: Matt Butcher & Matt Farina
Publisher: Manning
Price: £27.99
ISBN: 978-1633430075
magpi.cc/2vj0JBA

Excellent cookbook-style collection of Go recipes that will enlighten you on Go, as well as solving your practical problems.

The Little Go Book

Author: Karl Seguin
Publisher: OpenMyMind.net
Price: Free download
magpi.cc/2GyAbBE

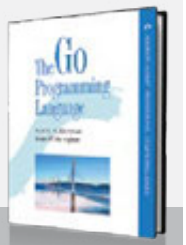
A short, CC-licensed introduction; if you're a developer looking for a quick start, this will get you up and running.

Go Web Programming

Author: Sau Sheong Chang
Publisher: Manning
Price: £27.99
ISBN: 978-1617292569
magpi.cc/2f0FF9a

You may be surprised to hear that Go is good for writing web apps; after reading this book you'll be convinced.

The Go Programming Language



Author: Alan A. A. Donovan & Brian Kernighan
Publisher: Addison-Wesley
Price: £29.99
ISBN: 978-0134190440
gopl.io

Not quite a K&R for Go, but this is an essential and practical guide to really learning the language in depth.

"It's a really fun educational computer kit that should really impress those who love Minecraft and building stuff" The MagPi

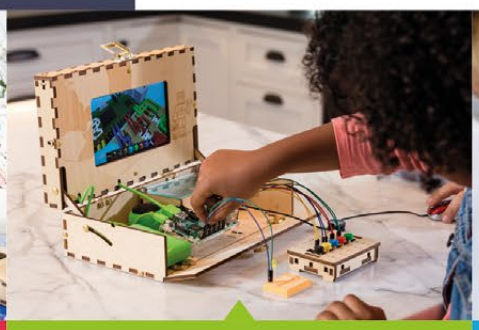
A STEM SOLUTION FOR SCHOOLS

Teachers all over the world use Piper to inspire kids to program, design, and engineer.

Want to learn how to bring Piper to your school?
buildpiper.com/EDU

THE PIPER EXPERIENCE

BUILD FROM SCRATCH



SOFTWARE-BASED CURRICULUM

CREATE YOUR ELECTRONICS

Supported by Top University Funds
Stanford-StartX Fund
AEF of Princeton University

Available at:

PIPER
BuildPiper.com

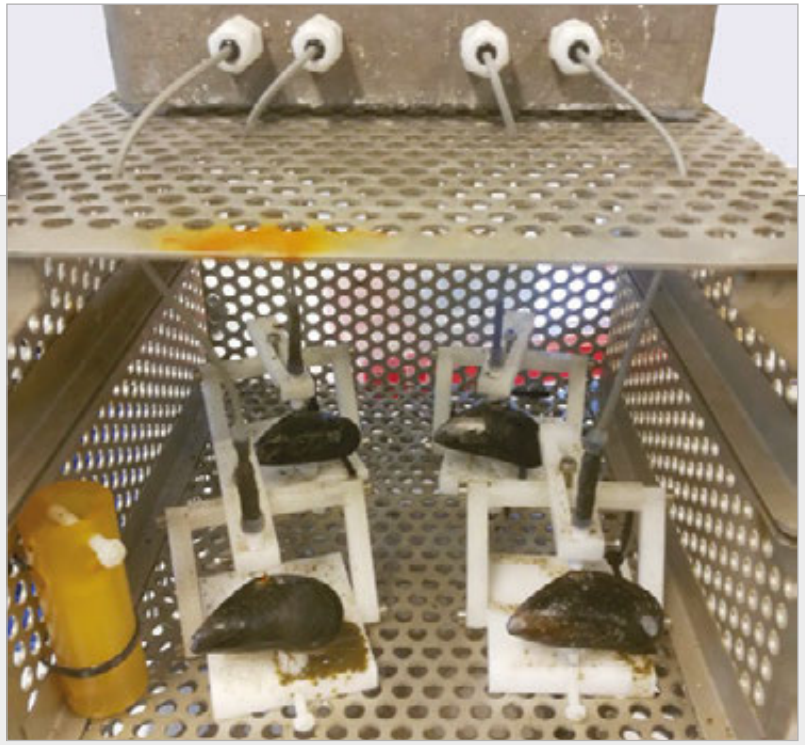
amazon

ToysRUs

BARNES&NOBLE

MARINE SCIENCE WITH RASPBERRY PI

Monitoring mussels and how humans affect their environment turned out to be a job for the Raspberry Pi



Jonathan Björklund

Occupation: Environmental Engineer

We like seeing the Raspberry Pi used for science, so when Jonathan approached us about how he'd been using Pi Zeros to study mussels for his masters thesis, we were immediately intrigued. Jonathan is a great lover of nature, and is an avid nature photographer as well as environmental scientist.

Can you tell us about your masters project?
Last summer I wrote my master thesis with the goal to create a battery-powered biological sensor system (a biosensor) that can monitor water quality. For this project I connected blue mussels (Mytilus edulis) to a Raspberry Pi Zero to monitor their movement. The project was very successful.

Why use the Raspberry Pi?
Since the Zero is cheap, uses little energy, and can easily be programmed with Python, it was the perfect match for this project.

What have you used the Raspberry Pi for in the past?
I have used the Raspberry Pi to control wireless cameras for monitoring purposes in hazardous environments. I have also used Raspberry Pi boards to control viewing screens at the local university.

What kind of results did you get?
The experiments showed that blue mussels' behaviour (valve gape movement) is altered by physical stimuli such as light, temperature, and access to food. The blue mussels are most active during night-time when it is less light. The biosensor system based on the Raspberry Pi proved very robust and reliable for this project.

Would you use a Raspberry Pi setup again?
I am planning to use the Raspberry Pi in a lot of future projects!

Above The cage that the mussels were kept in was submerged in water so they could be studied in their natural habitat

THE VIRTUAL RASPBERRY JAM

Luke Castle

Occupation: Student

Luke Castle holds a monthly Raspberry Jam, albeit not one you'll find at your local library

Raspberry Jams are great, but what if you can't attend one? York Pi Jam has found a way around that by offering an online Raspberry Jam (**virtual.yorkpi.tech**) that gives out tasks to its members. They have to complete a project in a week – like in a traditional game jam – and while prizes are given out, fun is the name of the game.

Can you tell us about the Virtual Jam?
A Virtual Pi Jam is an opportunity for people who have no knowledge in coding to have a go. We supply them with tutorials for Scratch, Python, HTML, etc. They have a weekend or a whole week to create a project and then submit it. The submissions then get judged and the winners get prizes. These prizes come from companies such as ModMyPi who have helped massively with the Virtual Pi Jams. The Virtual Pi Jams also allow people who have no local Raspberry Pi Jams to join in.

Why did you decide to start a Raspberry Jam?
There was a blog post on the Raspberry Pi website about starting new Jams and the idea intrigued me. Also, the fact that there wasn't really a Code Club at my school and I wanted to get people into physical computing and programming and give them the same opportunities I have had.

Why virtual?
Again, it gives such a massive opportunity for schools and individual groups to get involved and have a chance to win prizes. From this we have seen websites, coded traffic lights, and games on the Sense HAT. Not only does it allow opportunities, but also a chance for people with no knowledge or local jams to get involved. We also do love to see projects from coders with knowledge.

Below The winner of the joke-themed Jam created an entire website filled with jokes

How many people attend the Jam?
In our December Jam, for which we had the theme of 'Lost In Space', we had people submitting from five different countries including the USA, UK, and France. We also had a different range of groups taking part. Some were schools, some were friends. We roughly had 100 submissions that were whittled down so we could give out the prizes of £10 ModMyPi gift cards, some ModMyPi goody bags including some Pi cases and Haribos, and also a Devslopes Life Time Access Pass.

What sort of feedback have people given you about the Jam?
People are liking the fact that it has no geographical restrictions and there is no 'selling out'. Also, the fact we supply resources and by including prizes we are encouraging more people to get coding. It is great hearing feedback!

Would you ever consider doing the Jam in real life?
I am actually planning a real-life Jam for 24 March in York. We will be having some small workshops and it should hopefully be a great day.

What's been your favourite moment holding the Jams so far?
Getting to see the projects based on the monthly themes we set. Also, when people tweet us pictures of their prizes arriving by post, it is amazing. It is really fun.

THE MONTH IN RASPBERRY PI

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

BIG BIRTHDAY WEEKEND IS HERE!

CELEBRATE SIX YEARS OF RASPBERRY PI BY FINDING A LOCAL BIG BIRTHDAY WEEKEND JAM!

If you're reading this when the magazine comes out, it won't be long until Raspberry Pi turns six years old! To celebrate, over 100 Raspberry Jams are taking place around the world on Saturday 3 and Sunday 4 March. The list of Jams is still slowly growing and probably will continue to do so until the Birthday Weekend, so if you can't see one close to you on the list, keep checking to see if one is added.

Ben Nuttall, Raspberry Pi Community Manager, has this to say about the Birthday Jams:

"We're really looking forward to celebrating our birthday with thousands of people around the world. Over 48 hours, people of all ages will come together at more than 100 events to learn, share ideas, meet people, and make things during our Big Birthday Weekend.

"Since we released the first Raspberry Pi in 2012, we've sold 17 million of them. We're also reaching almost 200 000 children in 130 countries around the world through Code Club and CoderDojo, we've trained over 1500 Raspberry Pi Certified Educators, and we've sent code written by more than 6800 children into space. Our magazines are read by a quarter of a million people, and millions more use our free online learning resources. There's plenty to celebrate and even more still to do: we really hope you'll join us from a Jam near you on 3–4 March."

BIRTHDAY WEEKEND KITS!

Jams that signed before the deadline last month also got a special kit for their event, including copies of *The MagPi* as well as an excellent T-shirt to commemorate the event. Make sure you get any free swag you can from your Jam.

WHERE'S YOUR NEAREST BIRTHDAY JAM?

Want to get precise details on your local Raspberry Pi Birthday Jam? Head over to the Raspberry Jam page for an interactive map and links to the individual Jams here: **magpi.cc/2sumgtB**

Big Birthday Weekend | 3 March 2018 – 4 March 2018

- Ahmedabad, India — MyFirstPOC
- Amsterdam, Netherlands — AmsterJam
- Attard, Malta — MaltaPi
- Bedfordshire, UK — Potton Pi and Pints
- Belfast, UK — Northern Ireland Raspberry Jam
- Bletchley, UK — Milton Keynes Raspberry Jam
- Bogotá, Colombia — Bogotá Raspberry Jam
- Budapest, Hungary — Hungary Raspberry Pi BD Jam
- Cagliari, Italy — Raspberry Jam Cagliari
- California, USA — Discovery Cube Raspberry Jam
- California, USA — Morgan Hill Raspberry Jam
- California, USA — Riverside Raspberry Jam
- Cape Town, South Africa — RPiJamCT
- Cebu, Philippines — RPi Birthday with IoT Cebu
- Chelmsford, UK — Stokers Jam
- Cheltenham, UK — Cotswold Raspberry Jam
- Chennai, india — Next Tech Lab Jam
- Chennai, India — ResPro Raspberry Pi Jam Labs
- Chennai, India — RIT Raspberry Jam
- Cheshire, UK — Nantwich Raspberry Jam
- Chhattisgarh, India — BIT-D Raspberry Jam
- Chidambaram, India — jambigbirthday
- Cornwall, UK — Cornwall Raspberry Jam
- Cumbria, UK — Kendal Raspberry Jam
- Delhi, India — ILUGD Raspberry Jam
- Devon, UK — Exeter Raspberry Jam
- Dorset, UK — Meety Pi Club
- East Sussex, UK — Newhaven Raspberry Jam
- East Yorkshire, UK — Hull Raspberry Jam
- Ecatepec, Mexico — Piña Pi
- Essex, UK — Southend Raspberry Jam
- Florida, USA — Cocoa Raspberry Jam
- Florida, USA — Raspberry Pi Miami
- Forst, Germany — Ungeheuer Raspberry Jam
- Georgia, USA — MAKEmory Raspberry Jam
- Glenunga, Australia — Glenunga International High School Raspberry Jam
- Granada, Spain — Raspberry Jam Granada
- Greenwich, UK — Jam Ramma
- Gujarat, India — Surat Raspberry Pi Jam
- Hamamatsu, Japan — Japan Raspberry Jam in Hamamatsu 2018
- Indore, India — Indore Raspberry Jam
- Isfahan, Iran — Kajestan
- Istanbul, Turkey — 4th Annual Hisar Coding Summit
- Kanpur Nagar, India — Kanpur Raspberry Jam
- Kent, UK — Kent Raspberry Jam
- Kerala, India — Trivandrum Raspberry Jam
- Kisii, Kenya — Tabaka Boys' High School Raspberry Jam
- Lancashire, UK — Blackpool Raspberry Jam
- Lancashire, UK — Preston Raspberry Jam Sandwich
- Lima, Peru — Lima Raspberry Jam Verano 2018
- Limerick, Ireland — Limerick Raspberry Jam
- London, UK — East London and Covent Garden Jam
- London, UK — South London Raspberry Jam
- London, UK — TechTribe.UK Raspberry Jam Birthday Party
- Manchester, UK — Manchester Raspberry Jam
- Maryland, USA — Raspberry Pi Hack-a-thon
- Maryland, USA — Rock the Rasp
- Masvingo, Zimbabwe — Raspberry Team Masvingo
- Mazandaran, Iran — Babol Raspberry Jam
- Michigan, USA — Ann Arbor Raspberry Pi Birthday Jam
- Michigan, USA — West Bloomfield Jam
- Montréal, Canada — Pi Jam de Montréal
- Mumbai, India — Mumbai Raspberry Jam
- New York, USA — Long Beach Public Library
- New York, USA — Windsor Terrace / Kensington Raspberry Jam
- North Carolina, USA — Pi Kids Garage
- Nottingham, UK — Beeston Raspberry Jam
- Nottinghamshire, UK — Raspberry Jam @ Worksop College
- Odisha, India — Arihant's Jam
- Oklahoma, USA — Rose Cyber Jam
- Ontario, Canada — MyPi Toronto
- Ontario, Canada — Ottawa Jam - Big Birthday Bash
- Ontario, Canada — Pi Jammin
- Oxford, UK — Oxford Raspberry Jam
- Paris, France — Imen
- Penang, Malaysia — Raspberry Jam Penang
- Pennsylvania, USA — Bucks Raspberry Jam
- Pennsylvania, USA — Philly & Pi Birthday Jam
- Perth, Australia — Dojoberry Jam
- Saverne, France — Raspberry Pi Jam 2018
- Scotland, UK — Dundee Raspberry Jam
- Scotland, UK — Glasgow Raspberry Jam
- Somerset, UK — Taunton GlassBox Big Birthday Weekend Jam
- Somogy, Hungary — MalnaPi Jam
- Sucre, Bolivia — Sucre Raspberry Jam
- Tapei, Taiwan — Raspberry Pi Jam in Taipei 2018
- Terni, Italy — SuLuLab Raspberry Jam
- Texas, USA — ColleyVineJam
- Texas, USA — Nerdvana's 1st Raspberry Jam
- Texas, USA — RGVSA's First Raspberry Jam
- Tokyo, Japan — Raspberry Jam Tokyo
- Tyne and Wear, UK — Northumbria Jam
- Tyne and Wear, UK — Gateshead Raspberry Jam
- Tyrone, UK — Omagh
- Varna, Bulgaria — Varna Raspberry Jam Big Birthday Weekend 2018
- Vest-Agder, Norway — Kristiansand Raspberry Jam
- Virginia, USA — Roanoke Raspberry Jam
- Voi, Kenya — Voi Raspberry Jam
- Wales, UK — Cardiff Family Jam
- Wales, UK — Newport Family Jam
- Wales, UK — Swansea Jam
- Washington, USA — Key Tech Labs Raspberry Jam
- Washington, USA — Seattle Raspberry Jam
- West Midlands, UK — Birmingham Raspberry Jam
- West Sussex, UK — Bognor Regis Raspberry Jam
- West Virginia, USA — Raspberry Jam @ WVSU
- West Yorkshire, UK — Batley Library Raspberry Jam
- West Yorkshire, UK — Huddersfield Raspberry Jam
- West Yorkshire, UK — Leeds Raspberry Jam
- Wiltshire, UK — Jamming in Marlborough
- Wisconsin, USA — Norski Jam
- Yaoundé, Cameroon — Raspberry Jam Camer

RASPBERRY JAM Big Birthday Weekend 3-4 March 2018

STEAM AT BETT 2018

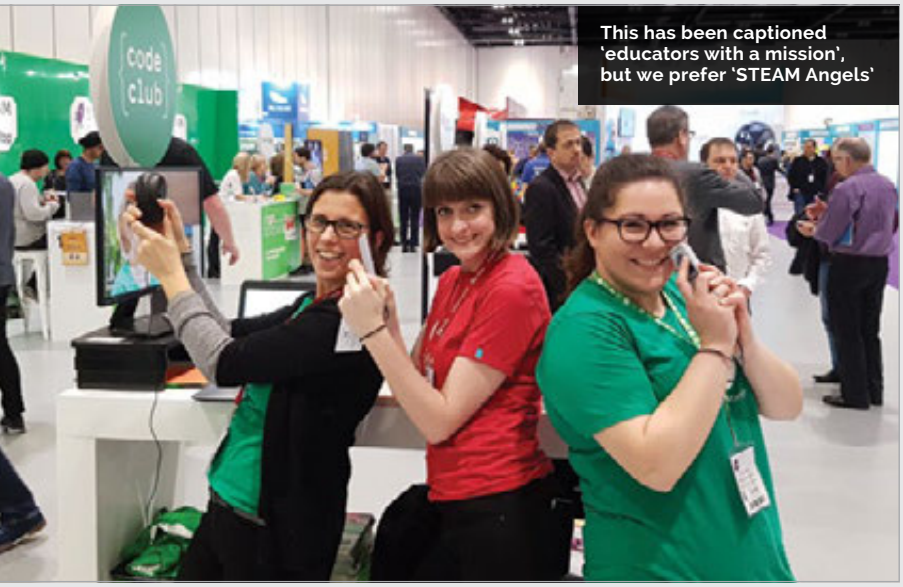
The annual education fair showcases new and upcoming teaching tech

Since 2014, Raspberry Pi has been attending Bett – an annual educational technology trade show – to show off all the free resources and teacher training programmes that the Foundation offers.

"Our setup this year consisted of four pods (imagine beefed up tables) in the STEAM village, and the mission of our highly trained team of education agents was to establish a new world record for highest number of teachers talked to in a four-day period," wrote Dan Fisher, Raspberry Pi Foundation Project Co-ordinator, in a blog post. "I'm only half-joking."

You can read his full report on the Raspberry Pi blog (**magpi.cc/2EC33L4**), but suffice to say it was a great weekend for promoting digital making.

The Raspberry Pi setup won Best Hands-On Experience, a great win for the hard-working Bett team



This has been captioned 'educators with a mission', but we prefer 'STEAM Angels'

Hello World is our educational sister magazine, seen here with American Raspberry Pi Certified Educator and all-round nice bloke Nick Provenzano (left)

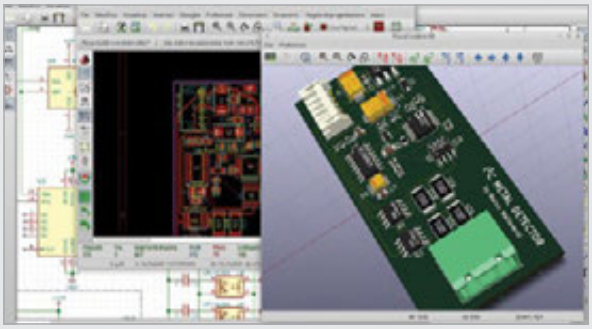
CROWDFUND THIS!

The best crowdfunding hits this month for you to check out...

NANOSOUND DIGITAL

kck.st/2GH6Chn

A follow-up to the NanoSound DAC Pro which we reviewed in a previous issue, the Digital version provides the same excellent setup while also letting you connect with S/PDIF. There's also a complete music-streaming box kit that will be funded with the new campaign, along with software updates. If you like using your Raspberry Pi as a music player, we recommend giving it a look!



I2C METAL DETECTOR

kck.st/2BiYVh5

We've been around the block with the cool and amazing Raspberry Pi projects that people in the community have made, but can't recall encountering any metal detectors – apart from the Discoverer robot featured in issue 48. This component might just help kickstart a movement of metal-detector projects. For people in the know, it uses a AD5933, a high-precision impedance converter which contains a Sine Wave DDS (direct digital synthesis) Generator from 1 kHz to 100 kHz. Hopefully those are magic words to the right people.

BEST OF THE REST

Here are some other great things we saw this month

"WHY DO YOU STARE AT YOUR MINTS ALL DAY?"

The 'Altoids tin computer' concept is something a lot of makers in the early days of the Pi Zero strived towards, but we've never seen a fully functional game system squeezed into one until this. Well done Wermy, you've made the ultimate incognito handheld.

magpi.cc/2Exlr7R

GOOD BURGER VHS KODI BOX

We love how daft this is. The garish Nickelodeon orange VHS cassette. The fact it's still got the tape inside. Also that it's forgettable mid-1990s kids comedy *Good Burger* starring Kenan and Kel. What a combination. We want one for our entertainment system.

magpi.cc/2EBCDsO

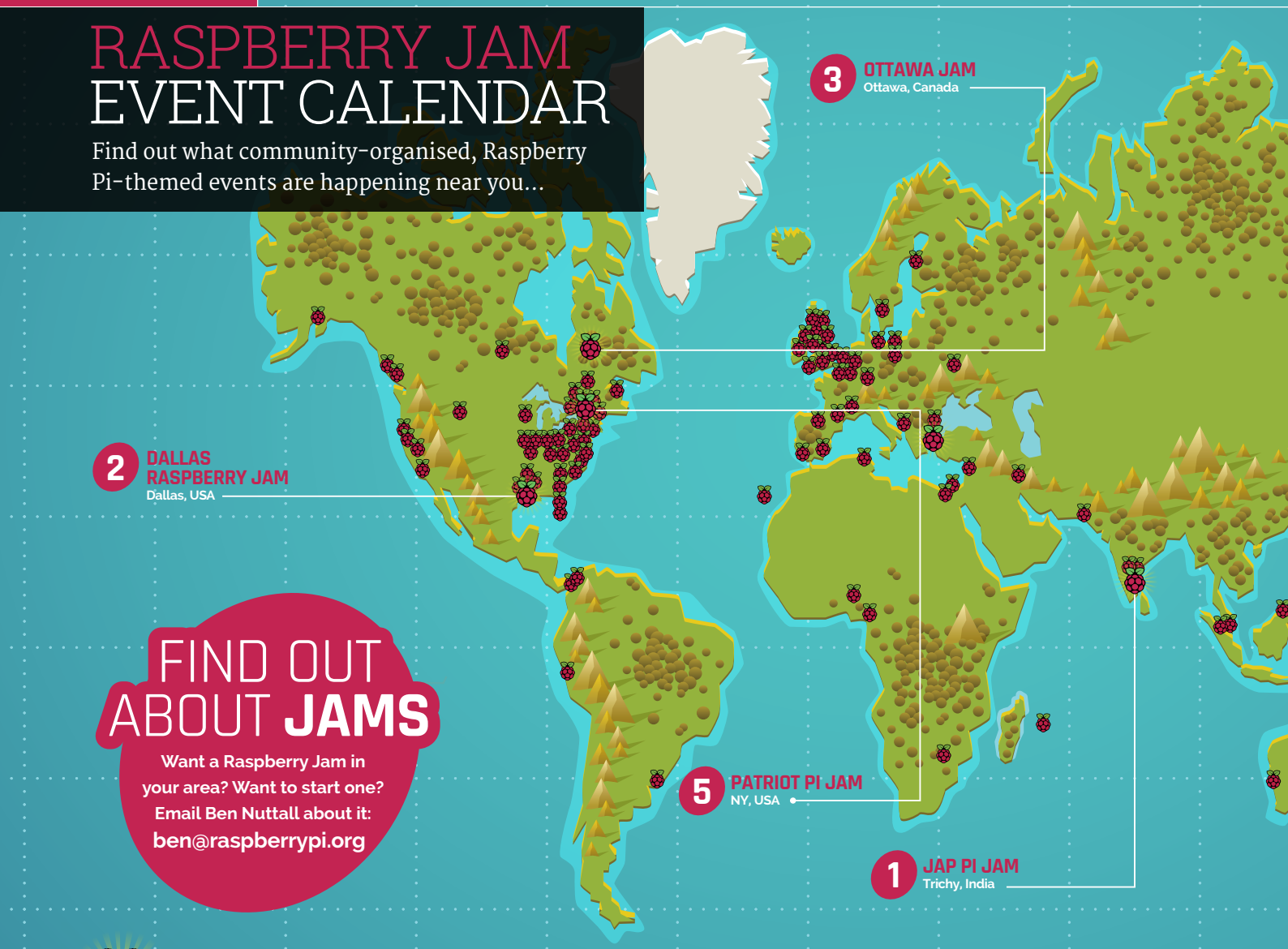
VICTORIAN MAGIC DRESSER

Magic mirrors are cool, but what is cooler is making the mirror section of a Victorian dresser into a magic mirror. It's almost steampunky – perhaps adding a Nixie tube aesthetic to the display would make it seem truly old? We love it anyway.

magpi.cc/2sq2CyG

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 JAP PI JAM

When: Thursday 1 March
Where: MAM College of Engineering and Technology, Trichy, India
magpi.cc/2EyZ2qS
A great way to find out more about the Pi, learn what you can do with it, and meet like-minded people.

2 DALLAS RASPBERRY JAM

When: Saturday 10 March
Where: Perot Museum of Nature and Science, Dallas, TX, USA
magpi.cc/2EDZB2u
A Raspberry Jam post-birthday celebration. Discover amazing gadgets and gizmos throughout the museum, and learn about coding.

3 OTTAWA JAM

When: Saturday 10 March
Where: Canada Science and Technology Museum, Ottawa, Ontario, Canada
magpi.cc/2EAFimz
Get your digits and neurons moving with hands-on coding and discover projects created by local tinkerers.

4 SHEFFIELD RASPBERRY JAM

When: Thursday 15 March
Where: Hertha Ayrton STEM Centre, Sheffield, UK
magpi.cc/2ExT89z
If you've built an interesting Pi project, bring it along and share it with the community for the chance to win an exciting prize.

5 PATRIOT PI JAM

When: Saturday 17 March
Where: Cherry Valley-Springfield Central School District, Cherry Valley, NY, USA
magpi.cc/2ED5MUy
Celebrate making with the Raspberry Pi at CV-S's first ever Jam. Come empty-handed or bring your own Pi.

6 YORK PI JAM FOR BEGINNERS

When: Saturday 24 March
Where: Robert Wilkinson Primary Academy, West End, York, UK
magpi.cc/2spChAL
A Raspberry Pi Jam for beginners, allowing them to learn more about the Raspberry Pi at this family-friendly event.

REGULAR EVENTS

7 PRESTON RASPBERRY JAM

When: Monday 5 February
Where: Media Factory Building, Preston, UK
magpi.cc/2EDfNBk
A community of people who meet in Preston each month to learn, create, and share the potential of the Raspberry Pi.

8 LONDON RASPBERRY PINT

When: Tuesday 27 March
Where: CodeNode, London, UK
magpi.cc/2swMPhN
Got a Raspberry Pi project you're working on? Show it off or get advice from others at Raspberry Pint, one of a number of capital events.

RASPBERRY JAM

BIG BIRTHDAY WEEKEND

Raspberry Pi turns six this year and to celebrate this, the Raspberry Pi Foundation is co-ordinating a huge number of Jams around the world to participate in the Big Birthday Weekend. There are over 100 Jams co-ordinating events on Saturday 3 March (and a few on Sunday 4 March). You can read more about them – including an up-to-date list! – in This Month in Raspberry Pi, on page 88.

YOUR LETTERS

SPACE **THANKS**

I would like to say a big thank-you to all those involved with organising the Astro Pi 2017 event. My son (Zack) had a fantastic time taking part in Mission Zero. It is amazing and very inspiring to know that two Raspberry Pi boards are up there on the International Space Station and that young people can make programs for them.
Gary Pickup

We're glad he had fun doing it! It's all down to the hard work of the Astro Pi team that young makers like Zack get to create code that actually gets to be used in space, and we're more than happy to help promote the latest challenges they set out.

If we were still kids, we would more than likely be trying to submit code for Astro Pi competitions as well!

A prized certificate showing that Zack here participated in the recent Astro Pi Mission Zero!

BIRTHDAY PARTY

I'm excited to be attending a Big Birthday Weekend Jam near me in March – I've not been able to come to the last two Cambridge-based birthday parties, so I'm happy that this year I'll be able to celebrate the birthday of Raspberry Pi with other people!

I was just wondering, though, will there be a normal birthday party like the last two years? I've not seen any info on it. I loved looking at the pictures of the last one and hearing about the cool stuff that went on. Either way, happy birthday to the Raspberry Pi!
Beth

It's great you've managed to find a local Jam to go to! The Big Birthday Weekend has been a great way to get people to come together and celebrate the Raspberry Pi without having to travel to Cambridge and try to get one of the limited tickets.

That being said, there will be a big, official event in Cambridge held by Raspberry Pi this summer. We don't have many details on it yet, other than it will be the weekend of 30 June – 1 July. Hopefully it means the weather will be a bit better! We'll definitely try to get there as well, so you'll have another chance to meet *The MagPi* team.

WEARING YOUR PI

I've recently been getting into wearables, making little light-up badges and patterns on old clothes I had. It's been really fun, but I wanted to expand what I did with wearables. Would the Raspberry Pi be suitable for this? What Raspberry Pi would be best for a wearable, do you think?
Lee

We've successfully used the Raspberry Pi for wearables in the past – even as far as using it for complex cosplay electronics! As the Raspberry Pi is nice and easy to program, with plenty of tutorials online on how to hook up electronics to it, it can be quite quick to set up your own custom wearable using one.

Other wearable makers and cosplayers may warn that the Raspberry Pi can draw a bit more power than other microcontrollers like Arduino. However, we've never experienced a problem with not having enough power to last for a significant amount of time.

As for what Raspberry Pi model we recommend – the Pi Zero is the natural choice for wearables due to its size. However, if you have the space, an A+ or even a full-size Raspberry Pi 3 will do the trick as they're still very small.

FROM THE FORUM: ADVENTURES IN TEXT

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**

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

The piece on making a text adventure in *The MagPi* #64 reminded me of the game 'Hampstead' on the ZX Spectrum. What makes that game stick in my mind is that it is a satirical game about social climbing (a typically British hang-up). Hampstead is an expensive London suburb. But in this game you can't just go to Hampstead and buy a house there: you have to 'attain Hampstead'. This you accomplish – as far as I remember – by marrying the boss's daughter, among other things.

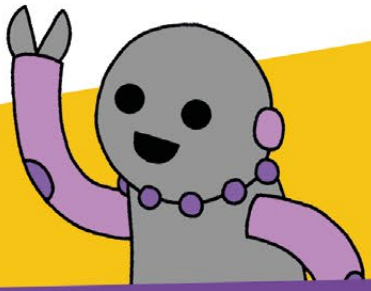
I bring this up to illustrate the point that it is not the sophistication of the technology that makes a game immortal (or we wouldn't mess with retro games), but the wit of the game's designers.

geoffreymh

We fully agree – while games are generally more complex these days due to advancements in both the technology we play them on and the ease with which games can be developed, a good story and a good idea is timeless, whether it's a simple text adventure, a ZX Spectrum game with limited colours, or a full-on VR experience.

Hopefully, the tutorial from issue 64 will show people just how simple they can start with game making, and work from there to make their magnum opus, whatever form it might take.

{code club}



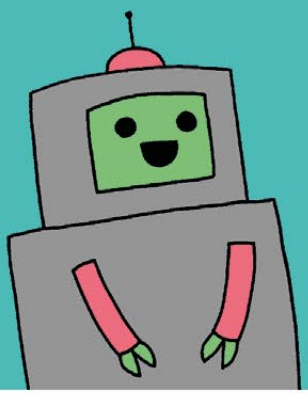
START A CODE CLUB IN YOUR SCHOOL!

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! 10 VISION KITS

IN ASSOCIATION WITH:

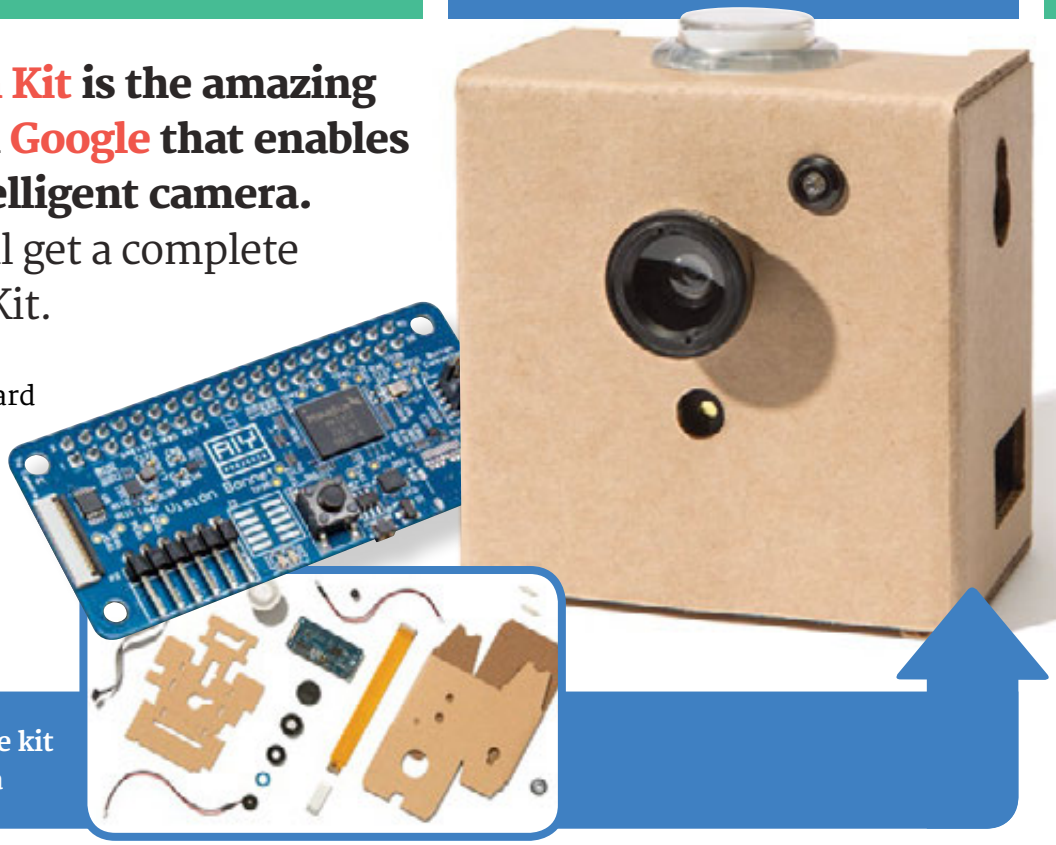
AIY PROJECTS

"This project lets you build an image-recognition device that can see and identify objects, powered by TensorFlow's machine learning models."

Learn more:
aiyprojects.withgoogle.com

AIY Projects Vision Kit is the amazing cardboard kit from Google that enables you to build an intelligent camera.
10 lucky winners will get a complete AIY Projects Vision Kit.

- VisionBonnet accessory board
- 24 mm RGB arcade button and nut
- Privacy LED
- Piezo buzzer
- Cardboard box and frame



Put everything together in the kit to build an intelligent camera

Enter now at magpi.cc/WinMar18

Terms & Conditions

Competition opens on **21 February 2018** and closes on **29 March 2018**. 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**.

MAKING AN IMPACT OUTSIDE THE CLASSROOM

A growing after-school movement provides more opportunities for young people to learn tech. **Matt Richardson** explains

Empowering young people with technology is a big part of what we do at the Raspberry Pi Foundation. In order to do that, we make computers as affordable as we possibly can. We also train educators for free and provide free content for them to teach digital making during the school day.

However, providing the opportunity for young people to learn about computers after school and on weekends is just as critical. They can use technology to explore their interests and passions, whether that's solving a problem, making a game, expressing themselves creatively, or just having fun.

This is why the Raspberry Pi Foundation supports Code Club and CoderDojo. Both programs take place outside of school time and provide young people with the opportunity to explore and create with technology.

> "We want this movement to be accessible to all young people and we need your help"

Code Clubs are for 9- to 13-year-olds and typically take place once a week for an hour at school, but after the end of the school day. We provide educators the content, structure, and volunteer matching so that their students can learn computational thinking with Scratch, make games or websites, and have fun at the same time.

A Dojo is for 7- to 17-year-olds and takes place in the evenings or on the weekends. They're typically hosted at community centres, libraries, universities, and corporate offices. CoderDojo is an opportunity for young people to create their own projects with technology with the help of volunteer mentors. Youth can create games, phone apps, websites, or even dip into physical computing.

An important aspect of these after-school programs is that they're an opportunity for the youth to learn in a self-directed manner. They can zip ahead or take it at an easy pace. If there's something they want to experiment with or get creative with, they can take a detour.

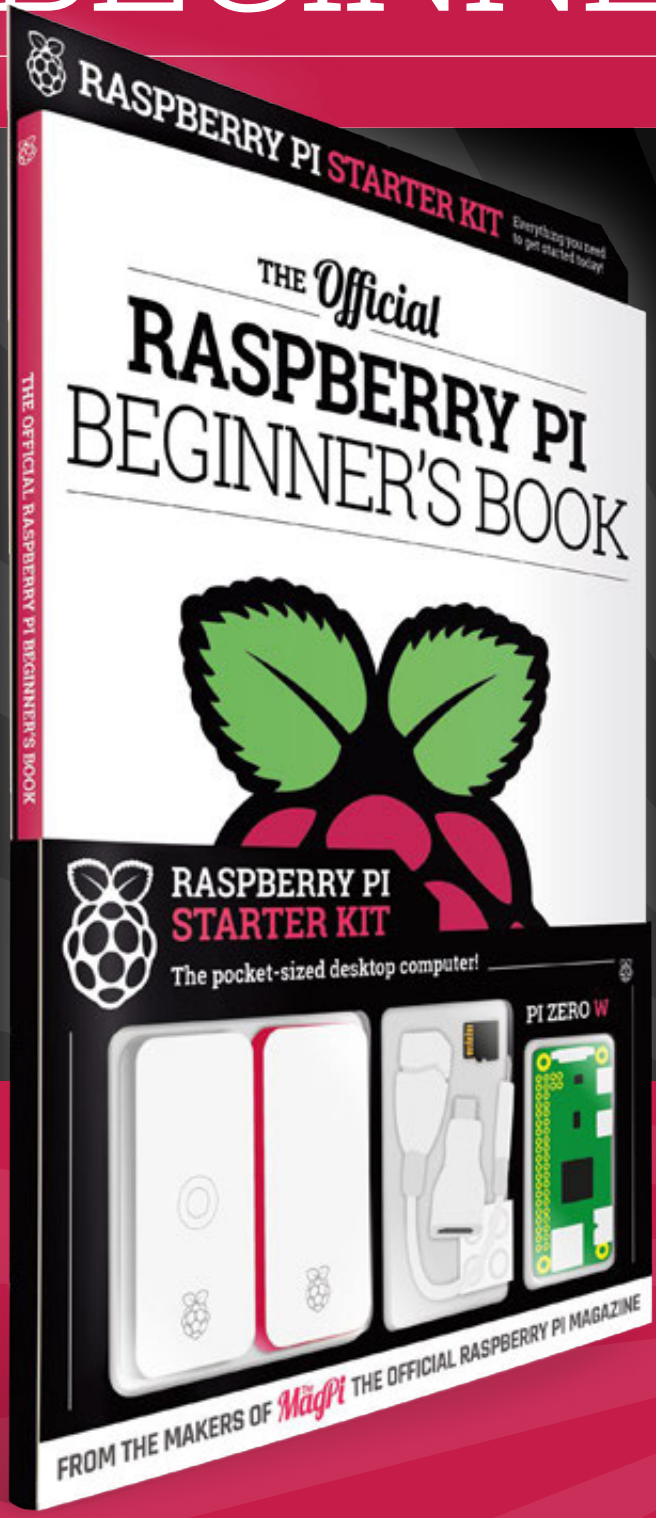
Different reasons

A young person may have different motivations for being at a Code Club or Dojo. Some may want to develop solutions to address real-world problems in their home or community. Some may want to create games; others to use code to express themselves creatively. What's important is that the youth are doing something they want to do, and not something that they're assigned to do.

Unlike in the classroom, a club isn't rigorous environment. There are no tests, reports, or papers. It's an opportunity to create, learn, collaborate, and share without pressure. I know from my own personal experience that not all youth thrive in the classroom, but can really shine in a more informal environment where they can explore their own passions.

We want this movement to be accessible to all young people and we need your help to make that happen. You can attend a local club with your kids, volunteer as a mentor, or encourage your local school or library to start one up. You can even start your own club for your community. Let's work together to put the power and potential of digital making into the hands of all young people, all over the world.

THE *Official* RASPBERRY PI BEGINNER'S BOOK



LEARN COMPUTING THE EASY WAY!

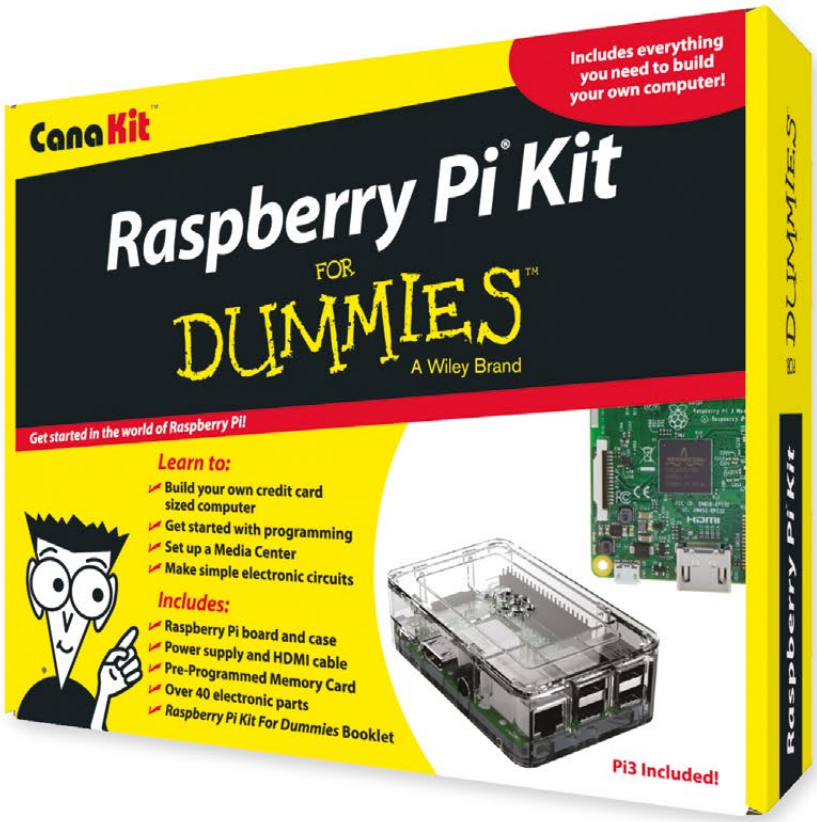
Includes

- Pi Zero W computer
- Official case with three covers
- USB and HDMI adaptors
- 8GB Micro SD card
- 116-page beginner's book

Available now

Buy online: store.rpipress.cc

CanaKit™



FOR DUMMIES®
A Wiley Brand

Kit Includes:

- *Raspberry Pi For Dummies* Booklet
- Raspberry Pi 3 Board
- Memory Card
- Plastic Case
- 2.5A Power Supply
- HDMI Cable
- Resistors
- LEDs
- Push Button Switches
- Prototyping Breadboard
- Jumper Wires
- Heat Sinks

Available for worldwide shipping at:

WWW.CANAKIT.COM

Available in Europe through RS Components

RS

$89.99 US DOLLARS

£69.99 EXCLUDING VAT

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO