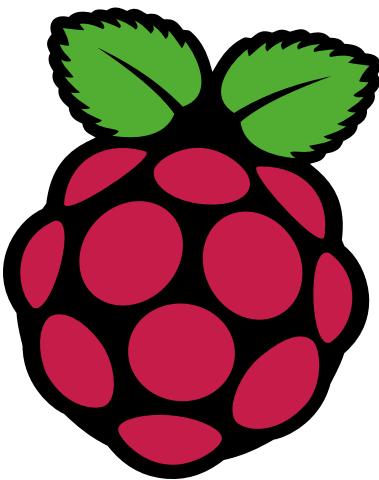




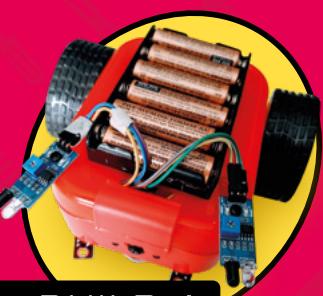
BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi



Issue 89 | January 2020 | magpi.cc

The official Raspberry Pi magazine



BUILD A SMART ROBOT

This wheeled racer responds to colours

MACHINE LEARNING FOR KIDS

Build a smart classroom assistant

50



KIT & GADGET GUIDE



CONNECT A CLASSIC COMPUTER TO THE WEB

Add internet to a retro computer

RASPBERRY PI TIPS & TOOLS



GRILL SANDWICHES WITH CHEESEBORG

THE CUTIEPI TABLET

MAKE AN AIRDRUM BEATBOX

Kickstart your new
project for 2020

£5.99



GLOBAL DELIVERY
magpi.cc/store

41 PAGES OF PROJECTS & TUTORIALS

Over 10,000 Different Boards and Modules In Stock

NO ONE KNOWS BOARDS LIKE DIGI-KEY!

- Open Source Community Boards
- Evaluation and Development Boards
- Wireless Modules
- Single Board Computers
- Reference Designs
- Programmers, Emulators and Debuggers

**DESIGNS START HERE!
DigiKey.com/Boards**

Digi-Key®
ELECTRONICS



#MAKEWITHDIGIKEY

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2019 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

WELCOME to *The MagPi* 89

Raspberry Pi is unlike any other computer. On the one hand, it's a fully powerful desktop computer, at a fraction of the regular cost. At the same time, thanks to its educational roots, Raspberry Pi is ready to become just about anything. Everything from solar-powered boats (page 8) to cheese-grilling machines (page 16) can be found inside this magazine.

Maybe you've got an idea for something to make. In which case our 50 Tips & Tools for Raspberry Pi feature (page 26) will help you kit out for 2020. This year could be your chance to finally build your crazy contraption.

If, like me, you've no idea what to make next, then take a look at our Raspberry Pi 4 accessories guide (page 68). Here you'll find kits and gadgets that can inspire you to build a friendly project.

I believe 2020 is going to be an incredible year for technology and making, and *The MagPi* sits at the heart of the most creative community on earth. This magazine is packed with incredible ideas and builds engineered by some of the most fun people we know.

Here's to another decade of making.

Lucy Hattersley Editor



**LUCY
HATTERSLEY**

Lucy is editor of *The MagPi* and cat whisperer. This month she's dressing up as a pantomime horse and running around Greenwich. As you do.

magpi.cc



Contents

► Issue 89 ► January 2019

Cover Feature

26 Tips & Tools

Regulars

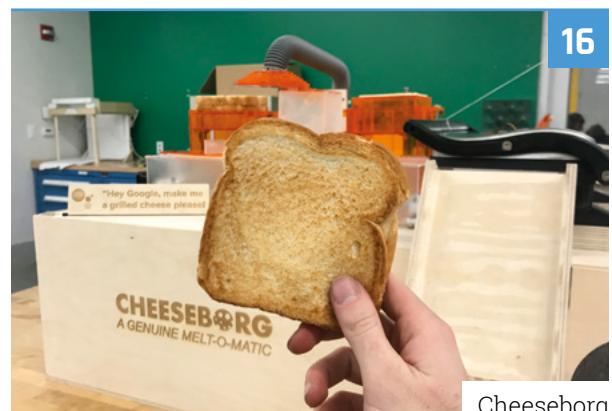
- 06 World of Raspberry Pi
- 92 Your letters
- 97 Next Month
- 98 The Final Word

Project Showcases

- 08 AMOS sailboat
- 12 Organnery pipe organ
- 14 Airdrum
- 16 Cheeseborg
- 20 CutiePi
- 22 Bit Time Rotary Dial Phone



Aquatic Mini Observation System



Cheeseborg

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.



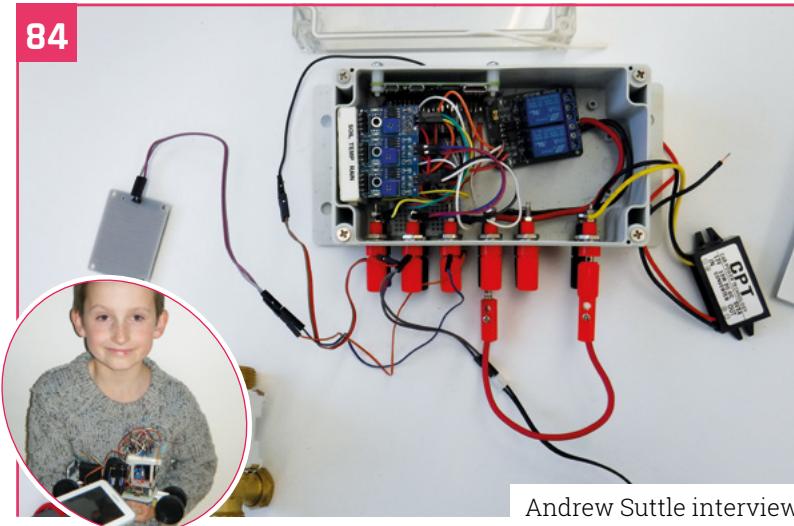
Add navigation to a robot



Design a 3D printer pendant



Add internet to a vintage computer



Andrew Suttle interview

Tutorials

- 38** Add navigation to a robot
- 44** Smart classroom assistant
- 48** Vintage computer internet
- 52** Hack GraviTrax – part 3
- 60** C and GTK custom widgets
- 64** Design a 3D printer pendant

The Big Feature

68



Raspberry Pi kit & gadget guide

Reviews

- 76** Enviro
- 78** SmartPi Touch 2
- 80** Top 10 advanced projects
- 82** Learn R with Raspberry Pi

Community

- 84** Andrew Suttle interview
- 86** This month in Raspberry Pi
- 90** Events calendar

WIN
ONE OF
FIVE

RASPBERRY PI 4
& CASE BUNDLES

95



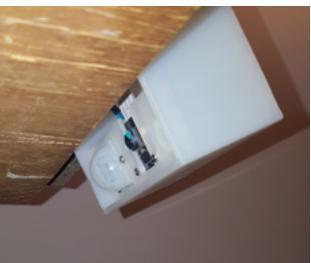


▲ A kestrel captured at night-time using Raspberry Pi NoIR infrared camera

The Nest Box

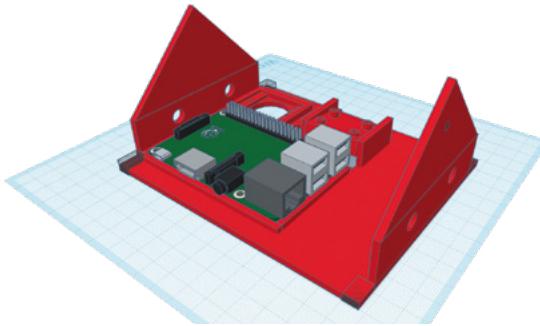
live-streams garden birds

This Springwatch-inspired nest box is used to lure Facebook users, such as **Rosie Hattersley**, away from cute cat videos



▲ A PIR sensor and camera fitted to a kestrel box

► The CAD design for the plastic tray at the top of a bird-box



It's a tried and tested social media formula: cute animals doing adorably cute things – or just being cuddly and wholesome. Click 'like' and share the heart-melting gorgeousness of the latest viral video. No wonder *The MagPi* magazine reader Jamie Wainwright's live stream of chicks in a nesting box caught our attention when we asked you about Raspberry Pi projects.

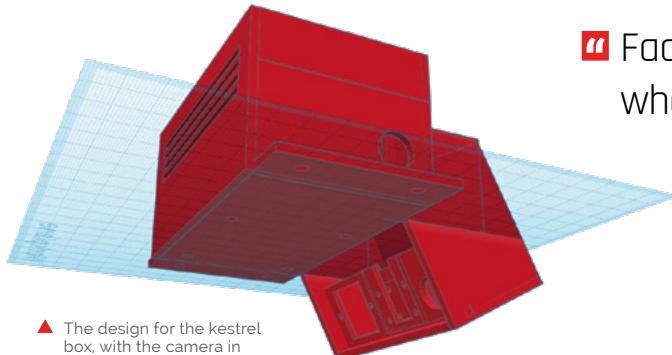
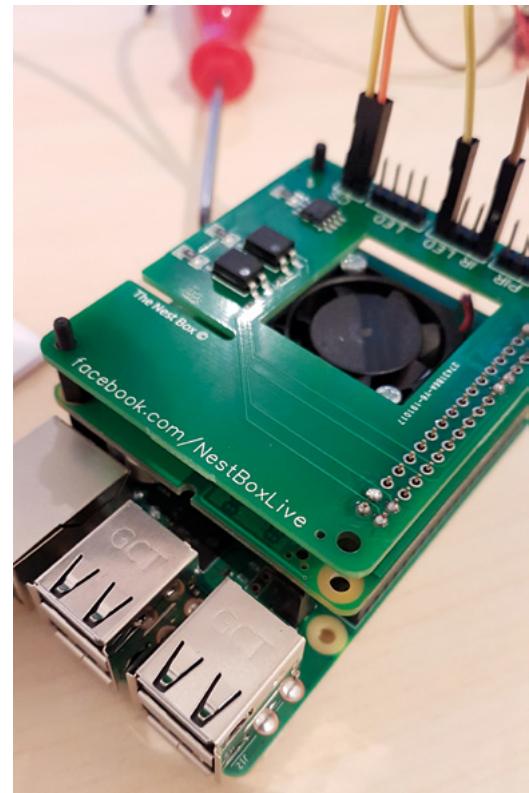
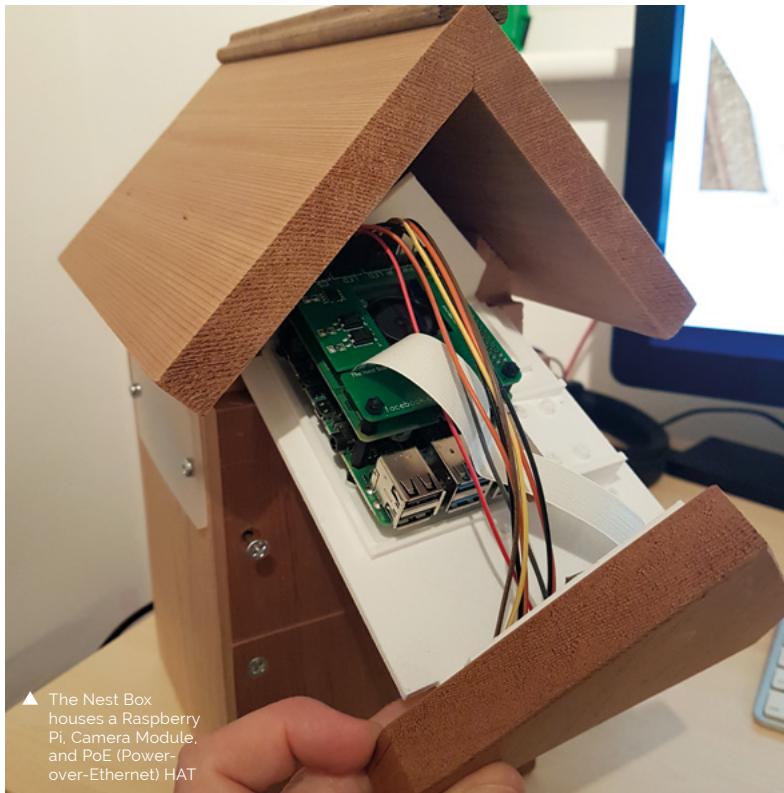
Liverpool-based Jamie introduces viewers to the cute residents of The Nest Box using impressively high-quality images and video from three Raspberry Pi-enabled British bird-boxes, with three more being set up.

These train video cameras on a kestrel box and also a barn owl box after Jamie proposed the camera feed to the Wirral Barn Owl Trust.

Instant replay

Throughout spring, all the cameras stream live to The Nest Box's 5500 Facebook wildlife fans (magpi.cc/nestbox). Out of season, visitors can enjoy a short movie recapping events from the nests earlier in 2019: magpi.cc/nestboxmovie.

Jamie says one of the beauties of The Nest Box setup is that it requires so little equipment: a Raspberry Pi and Camera Module, along with a Raspberry Pi PoE HAT (magpi.cc/poe) to provide both power and internet connectivity, so there's



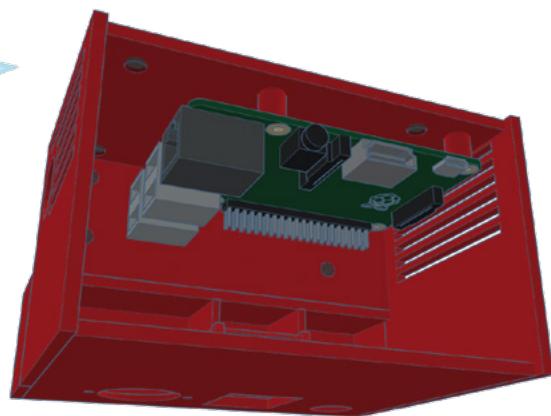
▲ The design for the kestrel box, with the camera in the lower section

only one cable connection to weather-proof. A custom HAT controls LED lights and the Raspberry Pi Camera Module, providing high-quality images day and night. Python code records visitors to the nest boxes. This causes the Facebook feed to go into live streaming mode whenever there's any nest box action.

Creature comforts

Thanks to some successful crowdfunding, Jamie was able to add sound via USB microphones attached to the Raspberry Pi rig. Furthermore, a schematic on The Nest Box's GoFundMe page (magpi.cc/nestboxfund) shows the smart bird-box's electronics hidden away in a small plastic box, with a fan to help keep everything cool during operation. ■

"Facebook goes into live streaming mode whenever there's any nest box action **"**



▲ Each installation has a different design depending on the bird being monitored – this one is for the barn owl box

◀ A 3D-printed section for the barn owl box

Aquatic Mini Observation System

What computer do you use when you want to have a solar-powered, autonomous boat that measures water quality? **Rob Zwetsloot** discovers that the answer is Raspberry Pi



MAKER

Murray
Lowery-
Simpson

Although coming from an engineering background, Murray has been developing software for a little over 20 years.

innaturerobotics.com

► AMOS will be useful to researchers, and a good base for hobbyists

There are many basic rules when it comes to electronics. One of them is that water and electronics don't mix, even if your average smartphone is able to take a brief dunk in the sink these days. So when someone comes to us talking about a waterborne project, we listen – and that's exactly what happened when Murray Lowery-Simpson emailed us about his solar-powered boat.

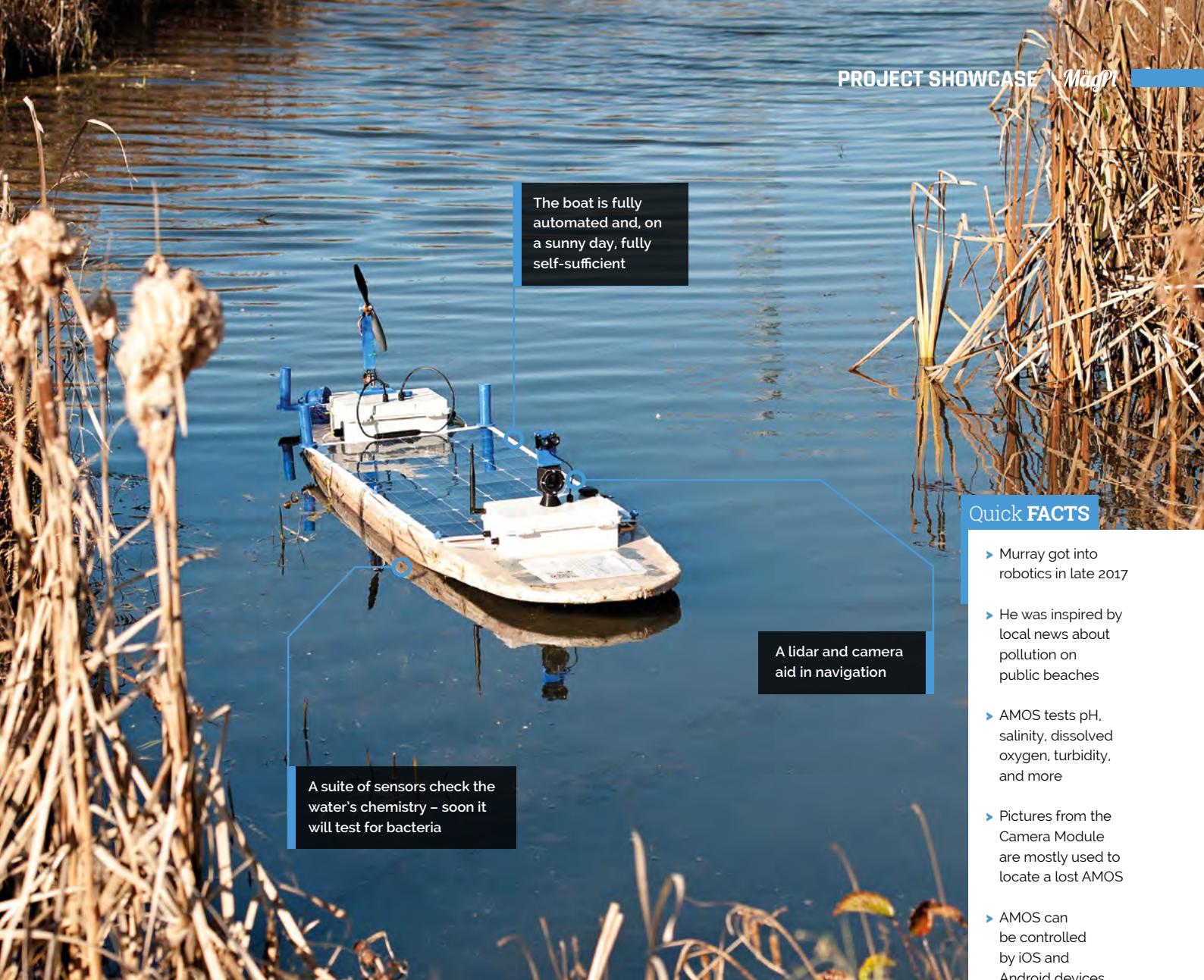
"My project is named AMOS (Aquatic Mini Observation System)," Murray tells us. "It is a solar-powered, autonomous airboat for measuring water quality over large, distributed areas."

Murray has worked on a couple of prototypes for the boat. The first one was made out of a kayak beer cooler (a small kayak that acts as a beer cooler) and had propellers that would end up getting gunked up. He also tested distance measuring with a Raspberry Pi Compute Module's stereo vision before settling on a lidar module and a Raspberry Pi 3B+.

"During this past winter, I built a second prototype, this time using a longer surfboard-type design constructed from glued-together insulation foam that was given a coat of fibreglass to give it some added strength and stiffness," Murray explains. "Instead of the water propellers, a single 10-inch drone propeller and motor were used and connected to a small waterproof servo motor at the stern end of the boat. This design was lighter (about 13 kg) and longer, and although the air propeller only produced about a tenth of the thrust provided by the dual water propellers, the improved draft and hydrodynamic shape made it slightly faster in the water."

A Raspberry Pi controls the speed and angle of the air propeller, takes sensor readings from the water, interacts with the lidar module, and has several other functions so that it knows its speed and heading.





Quick FACTS

- ▶ Murray got into robotics in late 2017
- ▶ He was inspired by local news about pollution on public beaches
- ▶ AMOS tests pH, salinity, dissolved oxygen, turbidity, and more
- ▶ Pictures from the Camera Module are mostly used to locate a lost AMOS
- ▶ AMOS can be controlled by iOS and Android devices





▲ This photo gives you an idea of the size of the boat

"I'm hoping that AMOS will be used for water testing by environmental services companies, and industrial customers such as mine operators that may be required by law to confirm that pollution limits in bodies of water surrounding their operations are not exceeded," Murray reveals. "I'm hoping also to be able to offer it at an attractive price point, with modular components so that researchers or robotic boat enthusiasts could also use it, or some subset of it, in their own projects."

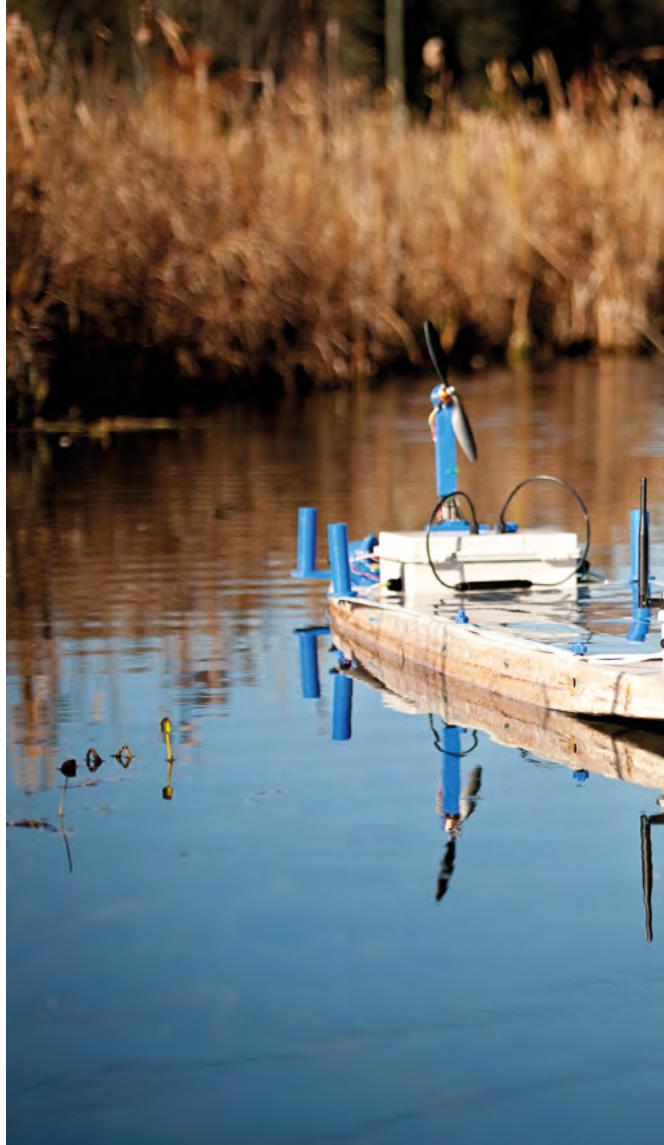
■ Approximately 150 km of testing has been completed on the second AMOS prototype ■

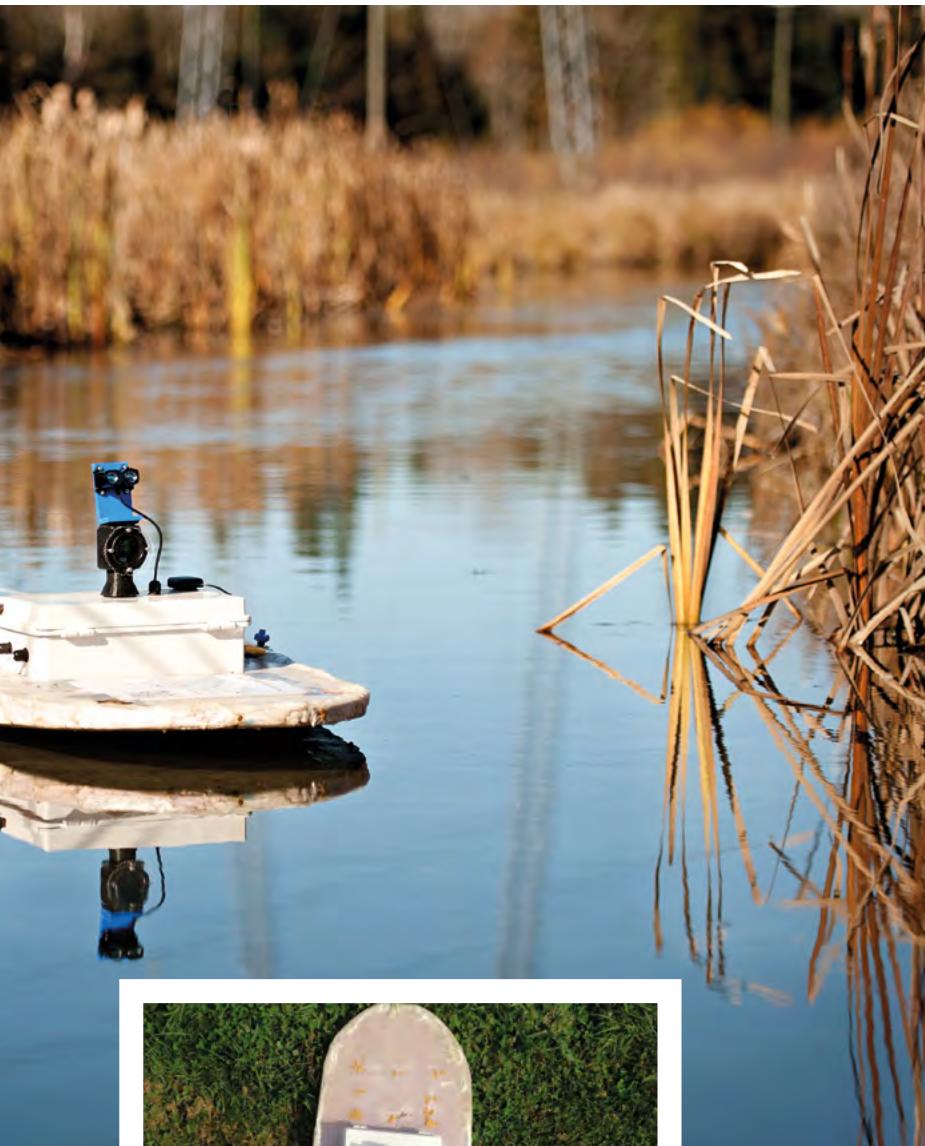
Major tests

The prototypes aren't just proofs of concept, either: they're fully functioning test beds, as Murray explains: "Approximately 150 km of testing has been completed on the second AMOS prototype in 2019. It can work well in shallow water (as little as 2 cm depth) and can travel through regions of water with lots of grass or other vegetation without any worries about getting stuck. Its airboat design works best under conditions of low wind (less than 20 km/h) and it can travel at a top speed of about 2.7 knots (5 km/h). Provided the sun is shining on a clear day and higher than about 40 degrees in the sky, AMOS can run at top speed without depleting the charge of its battery."

Murray plans for AMOS to be on sale in the summer, so you don't have too long to wait. ■

Our interview with Murray covered many more topics we couldn't squeeze into the magazine, so please look out for the full transcript at magpi.cc/amos.





▲ We think the AMOS looks quite majestic as it pootles about

◀ AMOS also works in cold weather, although a lack of sun means it can't operate for long

◀ The surfboard body makes it light, and its length allows for a large solar panel

Testing the waters



01 A boat needs to be launched. You can control the destination using a smartphone; there are apps for Android and iOS that you can find on the GitHub page: magpi.cc/amosGH.



02 Using a combination of GPS, lidar, and cameras, the boat navigates its way around waterways and rivers. This allows it to obtain a big sample from any given area, and see how it changes throughout.



03 On a sunny day, it can happily take readings all day without too much interference. On a cloudy day, it may only operate for up to an hour. Although a bigger battery might solve this, it will be heavier.

Organney pipe organ

A French company is marrying broken electronic pipe organs with a Raspberry Pi, and finding it's a match made in heaven, as **David Crookes** discovers



MAKER

Daniel James

Daniel James is the director of 64 Studio Ltd, which produces, integrates, and documents free and open-source software for media.

organney.com

When it comes to the work of Audiotronic founder Raphaël Mouneyres, silence is most certainly not golden. Given his job entails the repair of electronic pipe organs, a lack of sound points to a major problem – one that's happening more often and becoming trickier to resolve using original parts.

Raphaël tell us that he increasingly found components for first-generation digital pipe organs – many of which are installed in churches across the world – to be either obsolete, expensive, or simply unobtainable, threatening to consign many an instrument to the scrap-heap.

The solution has been to strip the organs of their old electronics and replace them with something new. “After some thoughts and calculations, we came up with a method that would retain the console, keyboard, and controls and instead make use of a small, affordable, and efficient single-board computer,” he says.



▲ Digital organs have been used in place of pipe organs within churches for a while, but damp and the scarcity of spare parts mean they're not lasting as long

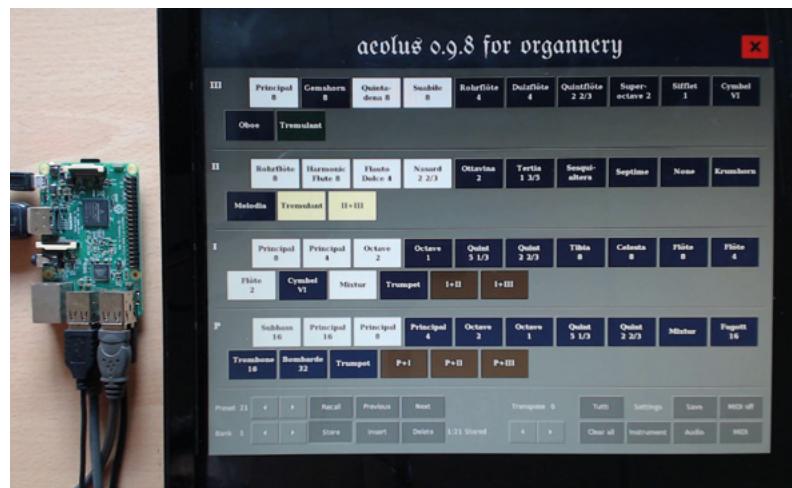
The chosen device was a Raspberry Pi, selected because of its worldwide availability, strong development community, and affordable price. It offered a perfect way of bringing pipe organs back to life so that they can be played as before, while allowing modern options such as a touchscreen.

“There are lots of good HATs available covering sound output, MIDI, and control interfaces,” says Daniel James, boss of 64 Studio, the company tasked with creating the software.

Fine tuning

As such, Daniel and colleague Chris Obbard began to work on a customised Debian GNU/Linux image called Organney that made use of the open-source classical organ modelling software Aeolus and Raspberry Pi’s capabilities. With eleven different temperaments, variable tuning, three or more manuals, and a pedal board, as well as MIDI in/out control and the flexibility to play different sounds, a personalised experience is possible.

“Organney needed to be set up as a single-use appliance, and the requirements came directly from the organists we’ve been working with to refine the system,” Daniel says. “We made adjustments to the Aeolus source code as well as to supporting programs. These included Mididings, a scriptable MIDI router and processor written in Python.”



▲ The Aeolus software allows a collection of stops to be saved to a USB memory stick for each organist

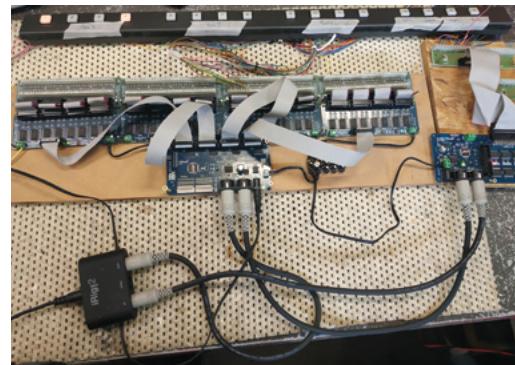


Daniel also created minimal system images based on Debian Buster using Dibby, which is a collection of scripts that leverage the distro's packaging tools. "This approach was a far leaner and more reproducible way to build appliances than starting from a standard image and stripping the system down," Daniel says.

Retro hits

So what's the verdict? "Organnery offers a major sound upgrade since the sound produced by the Aeolus software is of a much better quality than digital electronics from the 1980s," Raphaël says. "It gives the organ player access to new 'pipes' and complete control over sound placement in space, from a standard stereo system up to 3D Ambisonic diffusion."

Most Organnery systems are being retro-fitted into existing organs, but the system can be built from scratch using standard and affordable MIDI hardware. "Some of the smaller organs made for the domestic market have a five-pin MIDI Out socket and can be obtained for next to nothing," Daniel explains. "Aeolus and Organnery can also



The MIDI switches of an organ are being retro-fitted with new electronics, powered by Raspberry Pi

make learning the classical organ much more accessible than it has been in the past."

Even so, bringing organs back to life, complete with the original woodwork and physical controls, is perhaps most satisfying. "Church organ consoles are also often made of expensive hardwoods so we are saving trees," Daniel says. For organists used to the age-old interface of their instruments, that's sure to be extra music to their ears. 

Quick FACTS

- ▶ Humid and cold churches mean organ electronics suffer
- ▶ Retro-fitting a Raspberry Pi revitalises broken organs
- ▶ Organnery can be a MIDI expander for new sounds too
- ▶ It runs Aeolus, allowing hundreds of potential controls
- ▶ A touchscreen can also be fitted

Airdrum

This drumset can be played with a hover of your hands and gives visual feedback too, as **David Crookes** learns



MAKER

Alessandro Verdieisen and Luuk van Kuijk

Alessandro is an electronics engineer with a passion for drones and speaker systems, while Luuk is studying for his master's degree in next level engineering.

magpi.cc/airdrum

We're always going to beat the drum for projects that seek to improve the lives of people with severe disabilities. To that end, we fell in love with the Airdrum, which was created to allow anyone – in particular those with a mental or physical condition – to play a traditional instrument.

Designed by two Dutch electrical engineering students, Alessandro Verdieisen and Luuk van Kuijk, the project came to life during their first year at university. "We aimed to develop a musical instrument that could be used to generate music by moving," explains Alessandro, who has recently been working on a fully modular version 2.0.

After speaking with therapists and health care institutions, the pair decided to make a drum that could be played by moving objects above a set of panels and they put a Raspberry Pi at its heart. "The basic functionality of the Airdrum is to detect the distance of an object above each connected panel and play a sound," says Alessandro. "These panels contain IR distance sensors and coloured LEDs for visual feedback."

Sorting the bass-ics

From the outset, Alessandro and Luuk needed their project to be accessible, affordable, adjustable and, in the latest iteration, modular, with each drummable section containing an Arduino Mini, an IR sensor, and LEDs. They also wanted the



► A lot of meticulous planning has gone into the creation of the Airdrum over the past few years



▲ Custom-made Airdrum detecting modules fit snugly into their 3D-printed cases and can be arranged in a full circle if you have enough of them

instrument to have a broader appeal and be suitable for everybody, including professional musicians, so it had to sound as good as it played.

"We needed it to be as versatile as it can be and allow people to choose custom sounds, colours, and lights while being a standalone instrument and a multi-purpose input/output device," Alessandro reveals. To make it easy to place the modules together, they used magnetic connections between the panels. This allowed them to be placed together in various configurations, with a minimum of two per Airdrum.

With a structured plan that divided milestones into electrical, mechanical, and software components, the pair used 3D printing for the enclosure, which enabled rapid prototyping for quick interactions. They used speaker panels to bookend the modules for auditive feedback.

Panel beating

Each of the panels includes a buck converter so that the current through the connectors can be drawn to a minimum. The master module panel contains a Raspberry Pi 3 running custom-made programs written in C and Python, as well

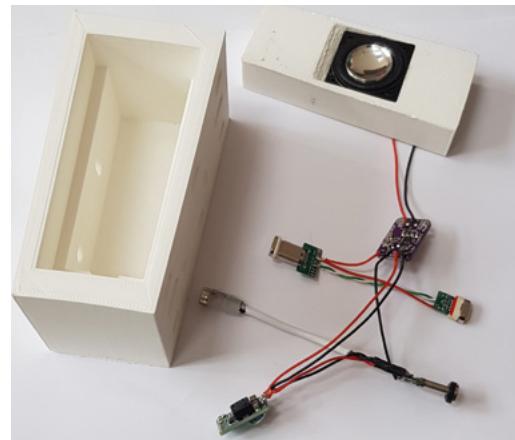


It also has a I²C bus to act as a data transfer master unit

as the free open-source software synthesiser FluidSynth. It connects to the other panels through I²C, constantly polling the panels for their measurements and for the configuration of their colour.

"If an object has been detected, the Raspberry Pi generates a sound and outputs it on the AUX audio jack," says Alessandro. "This output is then used by the mono D-class amplifiers in the speaker panels to make the tones audible."

The pair chose a Raspberry Pi because of its versatility and technical prowess. "The Airdrum needed something powerful enough to run software to generate audio through MIDI using the input from the panels and the Raspberry Pi is a great universal and low-cost development board with integrated DAC for audio," explains Alessandro. "It also has a I²C bus to act as a data transfer master unit and they're compact enough



to fit inside of the casing. The Raspberry Pi enables easy implementation of future upgrades, too."

Indeed, the pair want to explore the MIDI possibilities and connect the Airdrum with a smartphone or tablet. An app is being planned, as is a built-in synthesiser. "The people we have shown the Airdrum to have been very enthusiastic," Alessandro says. "That has been very motivating."

These speaker modules can be used to bookend the sensor panels, although the sound can be outputted via the Raspberry Pi to a different sound system too

Quick FACTS

- Disabled people can play without holding it
- A new modular design makes it versatile
- It can play MIDI files via a Raspberry Pi
- The designers also want pro musicians to enjoy it
- A basic two-module Airdrum would cost about £120

Cheeseborg

In wintertime, what could be more comforting than a grilled cheese sandwich?

Cheeseborg uses Raspberry Pi to serve them up, as **Rosie Hattersley** smacks her lips



MAKER

Taylor
Tabb

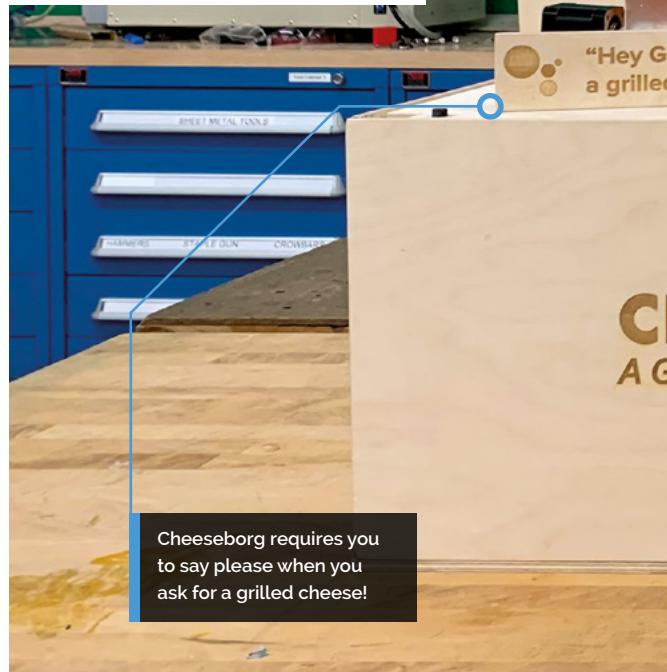
Taylor studied mechanical engineering at Carnegie Mellon University, exploring how people interact with hardware, and thinks Raspberry Pi is the ideal platform for this. He now works for creative agency DeepLocal, making 'wild and whimsical' robots.

tabb.me

Making a splash with your final-year university project can be the ideal way to gain the attention of potential employers and land yourself your dream job. Appealing to your lecturers' and classmates' stomachs is also a pretty effective way of getting them onside. Hearing from Taylor Tabb about the robot project he and friends Mitchell Riek and Evan Hill cooked up for their mechanical engineering degree, it's little surprise to learn that having graduated in the summer of 2019, he's already embarked on a fantastic-sounding Raspberry Pi-focused career.

"Cheeseborg was born out of our need to concept, design, and build a final prototype of a project in our senior engineering design class," says Taylor. "As hungry college students awake at hours that it's often hard to find food, we knew this was our calling – not just for us, but for grilled cheese eaters everywhere."

Cheeseborg is a dedicated, hands-free grilled cheese making robot that will satisfy cheese cravings at all hours. The Google Assistant SDK and AIY Voice Kit provide voice activation. Both "work fantastically with Raspberry Pi 3 Model B+," says Taylor. Their



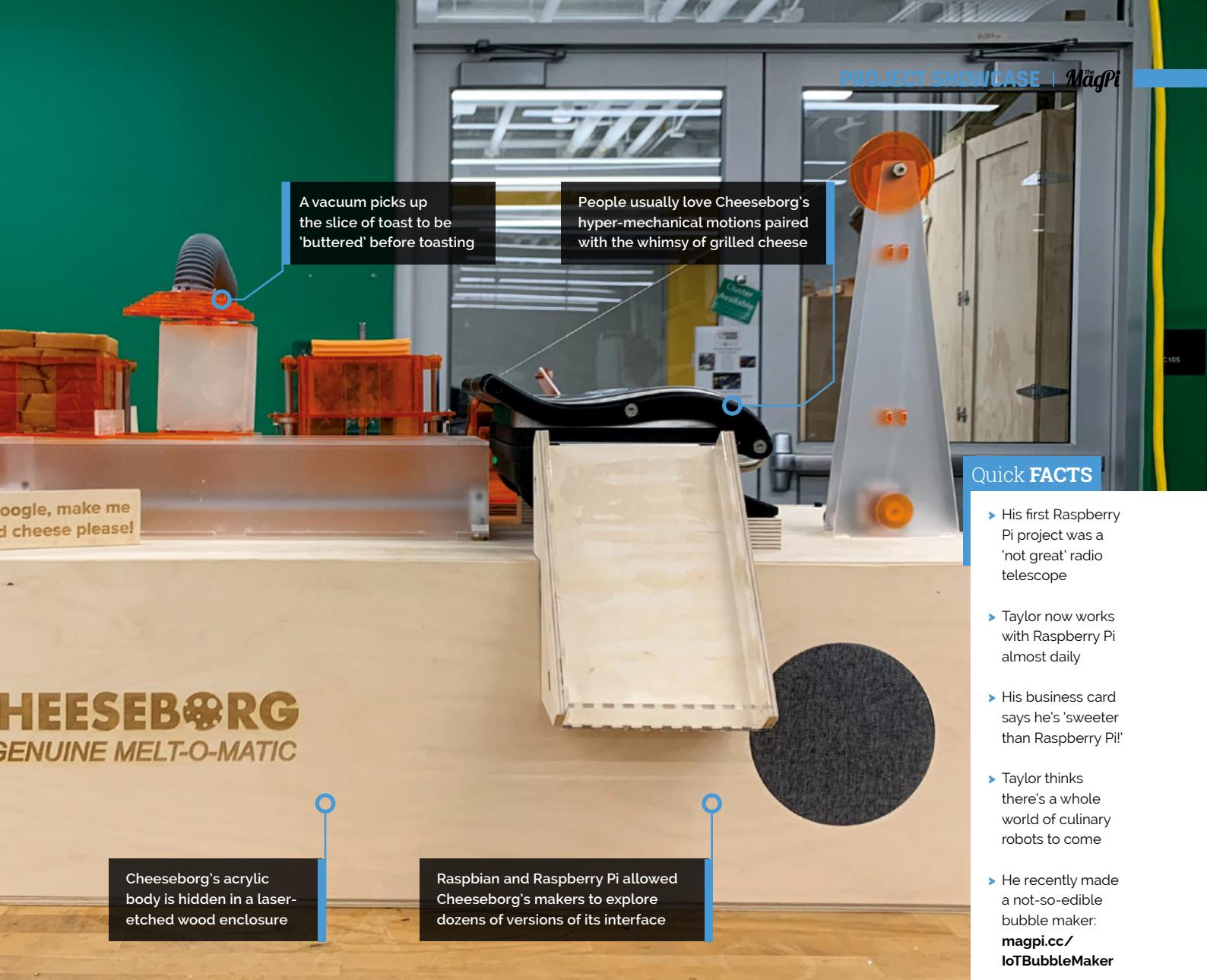
Cheeseborg requires you to say please when you ask for a grilled cheese!



Warning!
Hot mess!

Gooey butter is messy stuff. Be prepared for your kitchen, as well as your grilled cheese sandwich, to be buttered all over!





Quick FACTS

- His first Raspberry Pi project was a 'not great' radio telescope
- Taylor now works with Raspberry Pi almost daily
- His business card says he's 'sweeter than Raspberry Pi!'
- Taylor thinks there's a whole world of culinary robots to come
- He recently made a not-so-edible bubble maker: [magpi.cc/
IoTBubbleMaker](http://magpi.cc/IoTBubbleMaker)





“A dedicated, hands-free grilled cheese making robot that will satisfy cheese cravings at all hours”

customised build of Google Assistant, sourced from the AIY Kit and Google Cloud SDK, allowed them to add some personality: you can ask about the weather while awaiting your sandwich.

Raspberry Pi also triggers its assembly, while motor control circuits and motion are operated by Arduino Mega. Taylor reveals, “We wrote our own code, but also used open-source Arduino libraries,” dictating timings and the motions for each step.

Carnegie Mellon’s Department of Mechanical Engineering’s makerspace provided hardware commonly used in 3D printer kits, such as stepper motors and linear motion carriages. However,

the acrylic parts were designed and laser-cut by Mitchell, Evan, and Taylor. Impressively, the project cost just \$200 to build, and took approximately eight weeks to complete.

Taste test

The team undertook three weeks of customer research to identify grilled cheese preferences. “There was a lot of going back and forth between our build, CAD software, and laser cutter, adjusting parts as we learned more about the mechanics of bread and cheese,” recalls Taylor. They spent days just tuning the grilling time to get the gooeyness and crispiness just right, while experiments in applying the butter resulted in a thoroughly dairy-spattered makerspace.

Eventually, they went with a spray ‘butter’. “We aimed at the cooking plates and activated just before the bread was moved to be cooked,



thus buttering both sides [not to say the rest of Cheeseborg]. In the end, we found if we just spray the grill plates every five [sandwiches] or so, it still is enough to make a crispy, gooey grilled cheese!

"As with all good projects, we had no idea if it was going to work until 4am the morning before it was due, when Cheeseborg popped out its first fully hands-free grilled cheese sandwich."

Making your own grilled cheese robot

Interest levels online have been such that Taylor and his colleagues may consider offering a kit version. While he cautions that there's a huge amount of fine-tuning involved, he says, "If anyone out there is building a grilled cheese robot, we're more than happy to offer any insights." Contact him at tabb.me/grilledcheese. 

Making your own grilled cheese robot



01 This is the ideal project with which to use Google Assistant in Google's Raspberry Pi AIY Voice Kit, and the Google Cloud SDK for voice activation. Experiment with phrases to initiate the snack-assembly process.



02 Raspberry Pi now hands over to Arduino. Use Arduino Mega to control the robot that assembles the grilled cheese sandwich once Raspberry Pi instructs it. Open-source Arduino libraries are invaluable here.



03 Use tongs or suction to pick up slices of bread and add a chute for the snack's delivery. However, positioning bread to be buttered and adding cheese takes patience.

CutiePi

Want to liberate your projects from the desktop? This all-in-one Raspberry Pi tablet may provide the answer.

Phil King investigates



Penk Chan

A digital nomad wannabe from Taiwan, currently living in Tokyo and working as a principal software engineer at The Qt Company. He's leading a team of open-source enthusiasts to make the CutiePi tablet happen.

cutiepi.io

MAKER

Having long dreamt of owning a usable Linux-based portable device, a group of enthusiasts set out to create one and the **CutiePi tablet was born**. Based around a Raspberry Pi Compute Module 3+ Lite and custom carrier board, it features an 8-inch touchscreen, typical tablet features, and everything you need to make your Raspberry Pi projects portable.

"We tried to make the CutiePi tablet on par with normal tablets," says project lead Penk Chan. "You'll find a gyro, a microcontroller for battery and button monitoring, WiFi/Bluetooth, and a speaker. We also kept the camera connector and made the remaining GPIO pins available, keeping it hacking friendly." This will enable it to be used as a launchpad for users' portable Raspberry Pi projects.

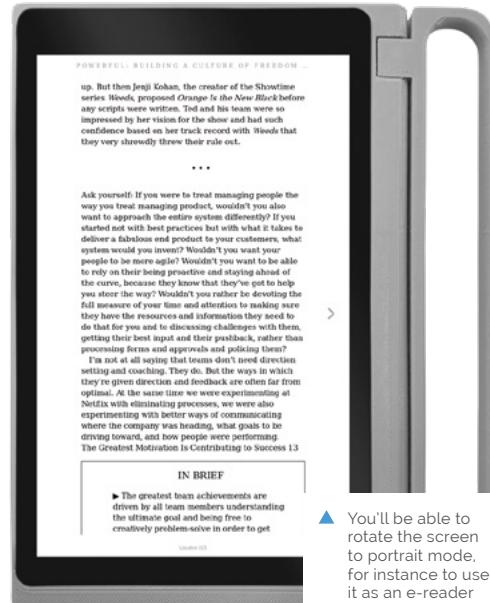
Making a portable device isn't easy, though. "It's not just about the Li-Po battery nor the DC-DC step-up converter," says Penk. "Those features that we take for granted in consumer electronics, like using the device while it's charging, reading remaining battery level, or simply detecting a power cable plug-in, are very hard to get right with modules and kits, let alone having a user interface that works. To top it all off, you need a case that houses all the hardware parts and cables."

Anyone who wants to produce it, or even build on it to make their next portable project, can freely do so ■

Custom carrier

The first CutiePi prototype was a cardboard box which housed an off-the-shelf HDMI display, a Raspberry Pi 3, and a power bank. For the second one, they stripped everything from the Raspberry Pi 3 board and soldered flex cables to replace the bulky HDMI connector, with the electronics now housed by a 3D-printed case.

While most hobby projects would have stopped there, the team went on to create a third prototype, based on a Compute Module 3+ Lite connected to



POWERFUL. BUILDING A CULTURE OF FREEDOM
up. But then Jeji Kohan, the creator of the Showtime series *Wheels*, proposed *Orange Is the New Black* before any scripts were written. Ted and his team were so impressed by her vision for the show and had such confidence based on her track record with *Wheels* that they very shrewdly threw their rule out.

Ask yourself: If you were to treat managing people the way you treat managing product, wouldn't you also want to approach the entire system differently? If you started not with best practices but with what it takes to deliver a fabulous end product, what would that mean for your business? Wouldn't you want your people to be more agile? Wouldn't you want to be able to rely on their being proactive and staying ahead of the curve, because they know that they've got to help you stay ahead? Wouldn't you want to be able to use the full resources of your team and attention to making sure they have the resources and information they need to do that for you and to discussing challenges with them, getting their best input and their perspective, rather than presenting them with a problem and a path forward?

I'm not at all saying that teams don't need direction setting and coaching. They do. But the ways in which they're given direction and feedback are often far from optimal. As we worked with our clients at Netflix on eliminating processors, we were also exploring with better ways of communicating where the company was heading, what goals to be driving toward, and how people were performing. *The Greatest Motivation Is Contributing to Success* 13

IN BRIEF

► The greatest team achievements are driven by all team members understanding the ultimate goal and being free to creatively problem-solve in order to get

▲ You'll be able to rotate the screen to portrait mode, for instance to use it as an e-reader

their own custom-designed carrier board. "Using the Compute Module allowed us to make the device a lot thinner, explore other form factors other than the regular Raspberry Pi 3's, and probably most important of all, it allowed us to mass-produce the CutiePi tablet," explains Penk.

Taking around three months to develop, the CutiePi carrier board is based on the reference designs made freely available by Raspberry Pi, and the team have open-sourced their now OSHWA-certified hardware: magpi.cc/CutiePiBoard. "At the heart of this project is our love for open-source, and CutiePi is our expression of that affection," says Penk. "All designs are available under open-source licence, and anyone who wants to produce it, or even build on it to make their next portable project, can freely do so. In fact, we advocate it."

Anyone who wants to produce it, or even build on it to make their next portable project, can freely do so ■

Multi-touch display

The CutiePi tablet features a 1280×800 MIPI-DSI display, with a five-point capacitive multi-touch panel. The user interface is built on top of Raspbian, and you can access the standard Raspberry Pi Desktop via a toggle switch. "When you toggle on over to our made-from-scratch, touch-friendly UI, you will have access to the CutiePi shell, including a WiFi settings manager, a Chromium-based web browser that supports all the common touch gestures, an on-screen keyboard with multiple



languages layout, and a terminal emulator, as part of our software version 1.0 release.”

Penk tells us that in the future it will be possible to support native Raspberry Pi apps in the CutiePi shell through use of XWayland, a fully-fledged X server implementation for the Qt Wayland Compositor being used for the display.

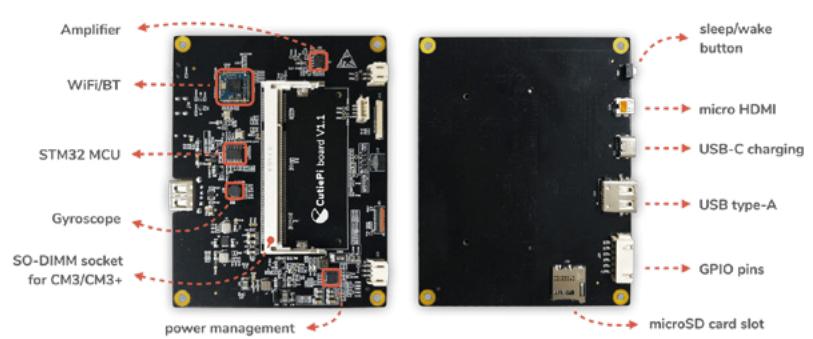
It all sounds very promising and, with the hardware parts near finalised, the team are focusing on the design for the final version of the tablet’s enclosure. They have decided to crowdfunding the project in order to mass-produce CutiePi, so keep an eye on those crowdfunding sites for it. **M**

◀ The OS is based on Raspbian, so you’ll be able to run standard applications and access a terminal to enter commands

▼ The CutiePi carrier board for the Compute Module 3+ Lite is crammed with functionality

Quick FACTS

- ▶ CutiePi gets its name from the Qt framework used
- ▶ The carrier board features a sleep/wake button
- ▶ The CutiePi prototype uses a 5000 mAh LiPo battery
- ▶ A 3D-printable case design file will be released very soon
- ▶ A Raspbian image will also be made available for testing



Bit Time – Rotary Dial Phone Project

A unique project in Essex explores how all ages engage with the tools that enable us to communicate, including using two upcycled retro phones. **Nicola King** dials in



Dave Norton
and Laura
Trevail

MAKER

Dave is a Digital Artist and Drama Practitioner, and his work ranges from large-scale interactive installations to live theatre performed in a virtual world. Laura is an Artist and Context Strategist with an outdoor theatre and live-art background.

magpi.cc/bittime

The Rotary Dial Phone project is part of a wider initiative called Bit Time – a project that has been running in Basildon over recent months. Lead artist Laura Trevail explains, “Bit Time is an intergenerational project, combining the skills and knowledge of the very young with those of our elders. As artists and facilitators, we’ve been bringing together these ideas and possibilities into playable works that in themselves keep that momentum going. These are projects about communication technology, but they are also communication technologies in themselves. By interacting with the art, you are interacting with each other.”

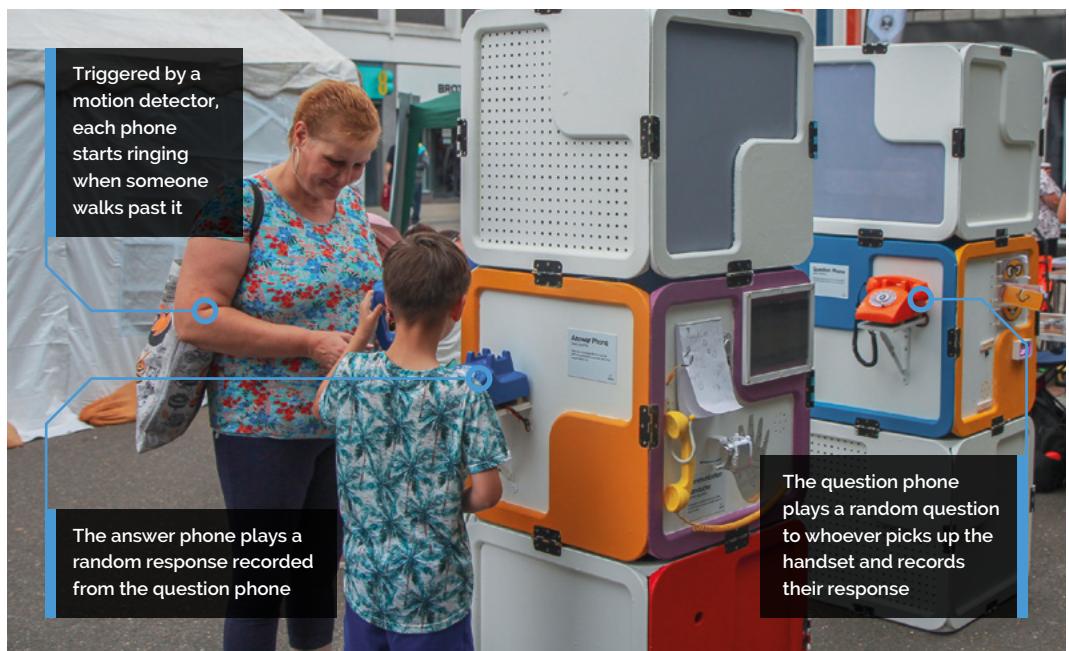
Cold calling

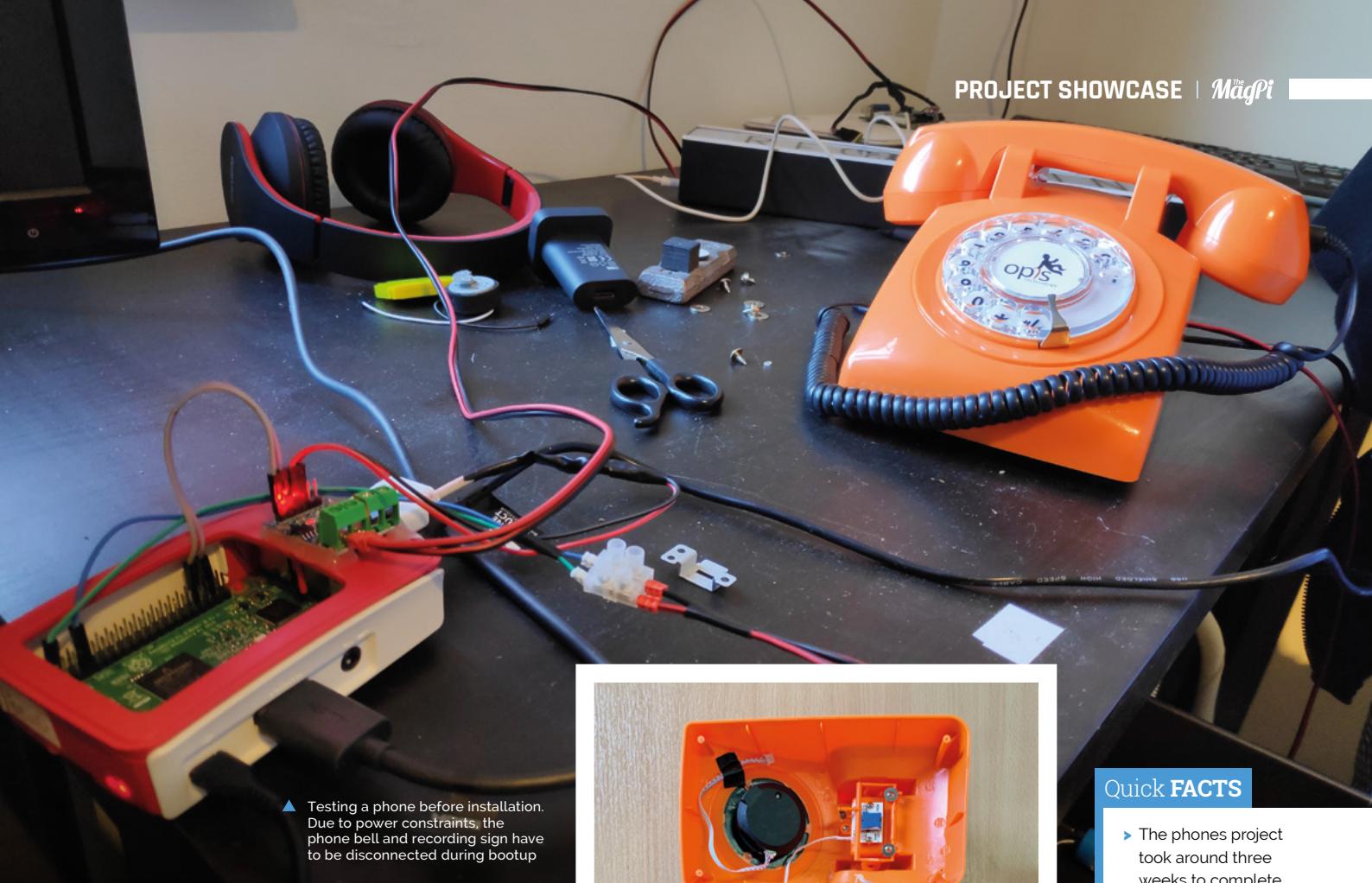
So, the retro phones... where do they fit into this story? Phone project artist Dave Norton says, “The

inspiration for the question/answer phones came from a desire to build a device that lets you share a message with someone you’ll never meet. A digital time capsule of anonymous thoughts, advice, stories, and memories that could be listened to by anyone. You have no idea who might hear your message and how it could affect them.”

He explains how the system works: “You walk past a phone and it starts ringing, you pick it up and the operator asks you to answer a question, e.g. ‘what was your first phone?’, ‘what will a phone of the future look like?’ A ‘recording’ light comes on and you leave your message and hang up. Later on, you see another phone that also rings as you walk past; you pick it up and it plays back a random message left by someone else.”

A motion detector identifies when someone walks past, while a push-button detects when the





▲ Testing a phone before installation. Due to power constraints, the phone bell and recording sign have to be disconnected during bootup

receiver is lifted. The phone's mic and speaker are hooked up to a Raspberry Pi, which chooses a random audio file question to play. "The mic starts recording the user's message for 15 seconds, or until they hang up, then the whole process restarts. The answer phone works in a similar way, but only chooses random audio files to play back."

Since the installation needed to work in any kind of location, it couldn't rely on WiFi to transfer audio files between the phones, particularly as the audio files needed to be checked manually before they could be shared with the public. "I ended up having to code a 'syncing' mode, which is activated when a USB drive is inserted into Raspberry Pi, which automatically transfers all the audio files to the drive," reveals Dave.

"Some people just hang up straight away, some people audibly freak out"

You talkin' to me?

The Bit Time project, including the rotary dial phones, ultimately became an exhibition which toured Basildon in the summer of 2019. Dave says, "There's something really unburdening about being anonymous, and something really pleasing about being given an open platform to speak your



▲ Dave says it was easy to snip the wires from each phone's speaker, mic, and hang-up button and connect them to Raspberry Pi

mind. I loved the idea that the installation starts as a blank slate and, as it travels to different events and locations, it collects a mixture of stories and thoughts and shares them with anyone who cares to listen, something akin to a travelling storyteller."

He says the phones elicited a variety of reactions. "Some people just hang up straight away, some people audibly freak out that they're actually being recorded, some yell bizarre phrases, but most people genuinely answer the question. No two answers have been similar and it makes for some really interesting listening... We've ended up with hundreds of varied audio responses – it would be lovely to build some sort of audio installation using all the clips." ■

Quick FACTS

- The phones project took around three weeks to complete
- Basildon Library Creator Space provided a location to construct the Bit Time artwork
- A 5V solenoid is used to ring each phone's bell
- Dave programmed each Raspberry Pi using Python
- He suggests the phones could be used in the foyer of a venue after an event, to glean honest views from attendees

SUBSCRIBE TODAY FROM ONLY £5

SAVE
UP TO **35%**



Subscriber Benefits

- **FREE Delivery**
Get it fast and for FREE
- **Exclusive Offers**
Great gifts, offers, and discounts
- **Great Savings**
Save up to 35% compared to stores

Rolling Monthly Subscription

- Low monthly cost (from £5)
- Cancel at any time
- Free delivery to your door
- Available worldwide

Subscribe for 12 Months

£55 (UK) £90 (USA & Rest of World)
£80 (EU)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero Kit with Rolling Monthly Subscription)

- ⌚ **Subscribe by phone: 01293 312193**
- 👉 **Subscribe online: magpi.cc/subscribe**

✉ Email: magpi@subscriptionhelpline.co.uk

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero W Starter Kit

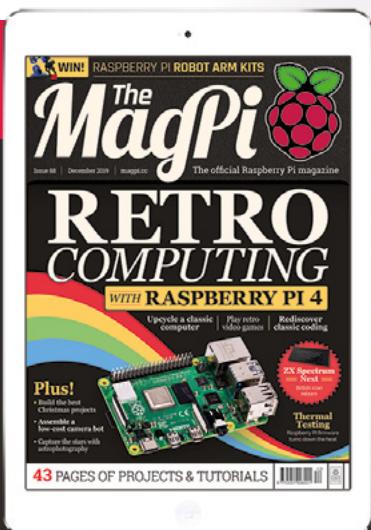
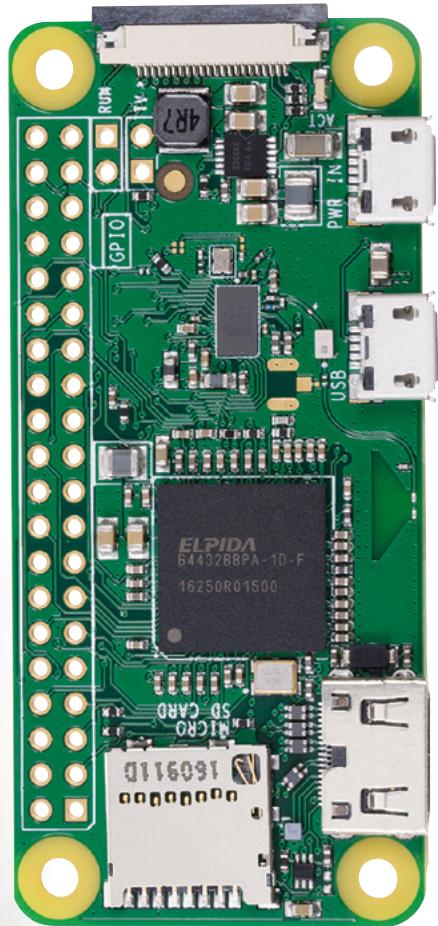
WITH YOUR SUBSCRIPTION

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

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

Offer subject to change or withdrawal at any time

**WORTH
£20**



Buy now: magpi.cc/subscribe



**SUBSCRIBE
on app stores**

From £2.29

Available on the
App Store

GET IT ON
Google Play

50

RASPBERRY PI 4 TIPS & TOOLS

THE RANGE OF TOOLS YOU CAN USE FOR MAKING IS HUGE.
HERE ARE 50 PHYSICAL AND DIGITAL TOOLS TO IMPROVE
YOUR MAKING SKILLS. BY **MARK VANSTONE**



Warning!

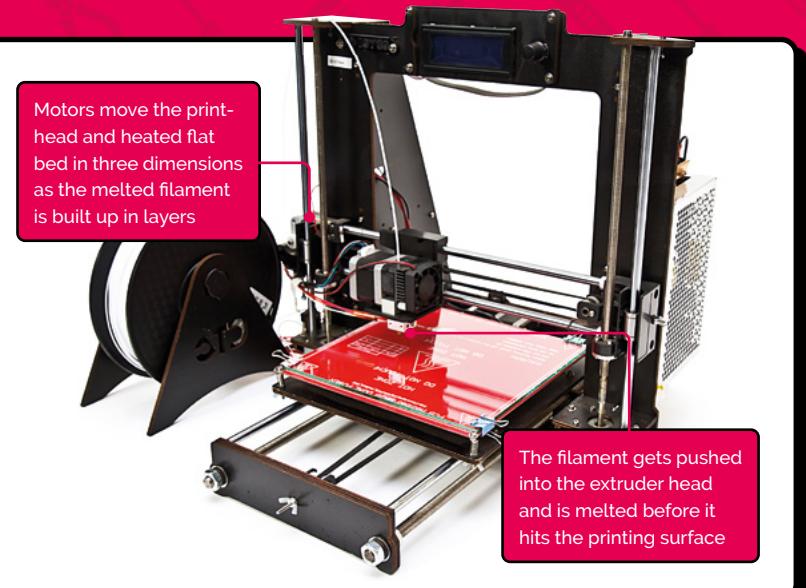
Some of these tools use mains electricity and heated parts. **Always** follow connection instructions and **never** leave heated elements unattended.

Whether you are new to making with Raspberry Pi or have been at it for years, there are always new tools and techniques to be learned. For the new arrival to the making community, this list will be a great resource to introduce you to tools that you may not have heard of before, and a short cut to get to the bits of software that work best for Raspberry Pi projects. If you've been doing this for a while, you may want to count how many of these tools are already in your makerspace. If you get to 50, you probably need a bigger makerspace.

01

3D PRINTER

 The 3D printer is a fairly recent addition to the maker scene. To be able to print something solid a few years ago was like science fiction, but now it's a reality. Printers come in many sizes and prices, so you can match your printer to your budget and requirements. There are several choices for filament too, such as PLA (a good beginner's choice) or ABS and many other more specialised types. If you have the time and patience, you can get a small unassembled one for less than £100, but for better quality (and less frustration) you may want to spend a bit more.



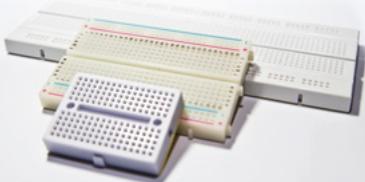
Printing something solid a few years ago was like science fiction

02

BREADBOARD

 Breadboards are solderless circuit boards and allow for fast prototyping of projects. The boards come in several sizes and consist of a matrix of small connector holes into which components and jumper wires can be inserted to make circuits.

magpi.cc/breadboard



03

SPEAKER

 Some projects need to make a noise, and the sound quality depends on the type of speaker you use. There are small piezoelectric speakers if all you need are beeps and pops; alternatively, you may need a larger dynamic speaker.

magpi.cc/piratemini

Affordable 3D printing

Pick up a copy of *The MagPi* #69 to learn all about affordable 3D printing. magpi.cc/69



04

LEDS



LEDs (light-emitting diodes) are a bit like very small and efficient light bulbs. They glow when current is passed through them, but they need to be connected correctly. Because they are diodes, the current only flows one way, unlike in conventional light bulbs.

magpi.cc/leds



05

RESISTORS

 Resistors are used in electrical circuits to reduce current flow. They are used for many reasons, including changing the brightness of an LED. They can be a fixed value, with stripes to indicate the resistance; or variable, with a slider or dial to change the resistance.

magpi.cc/resistorkit



07

DIODES

 Diodes are known as semiconductors because they only conduct electricity in one direction. This can be useful for a number of reasons; for example, to protect your Raspberry Pi from being damaged if you are connecting it to motors.

magpi.cc/diodes



09

CODE EDITOR

 There are several code editors for Raspberry Pi. IDLE used to be the standard Python editor in Raspbian, but now you might want to give Thonny or Geany a go. If you are feeling adventurous, you could try installing PyCharm.

magpi.cc/texteditors



06

MOTORS

 When a current is applied to an electric motor, a spindle in the centre rotates. It spins because of an electromagnetic field that is caused by the flow of current. Motors come in all kinds of sizes, from mini drone motors to industrial ones.

magpi.cc/motors

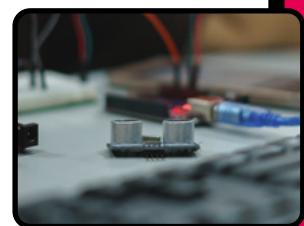


08

SENSORS

 There are many types of electronic sensors. You may want to detect light or vibration, sound, or heat. There are sensors for all of these and many more. Sensors allow us to capture data about the world around us.

magpi.cc/sensors

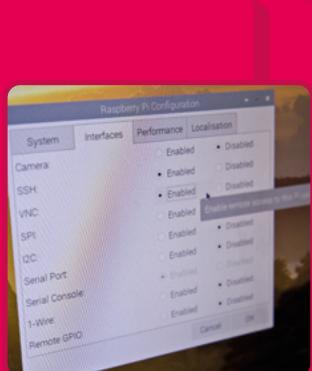


10

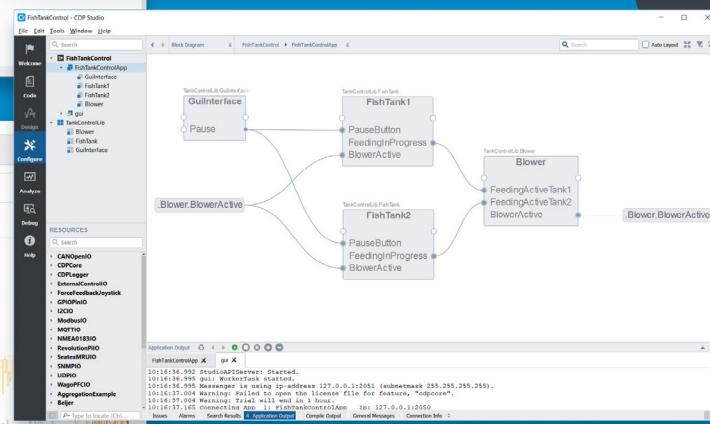
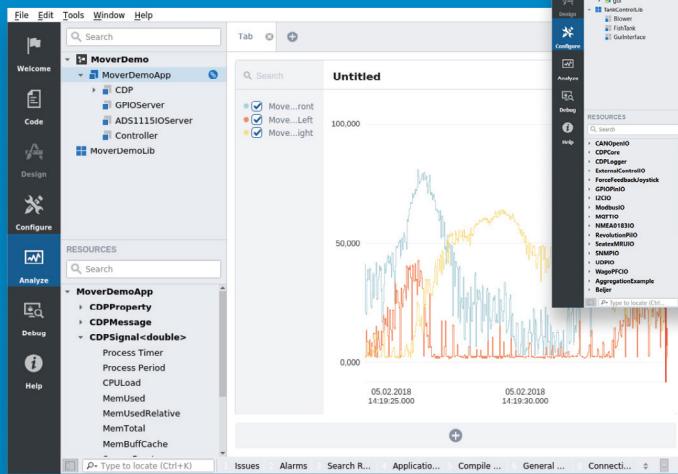
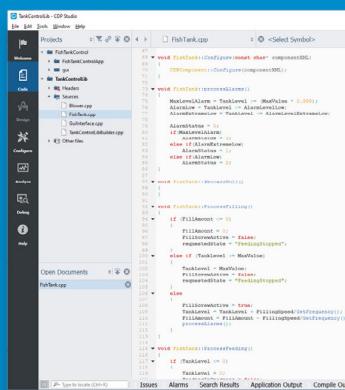
SSH AND VNC

 If you want to run your Raspberry Pi headless (without monitor, keyboard, and mouse), you will probably want to connect to it by typing in commands using SSH, which provides a terminal, or you can have a windowed version with VNC.

magpi.cc/vnc



Code



Analyze

CDP STUDIO

NEW GRAPHICAL EDITOR!

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.

TOOLS FOR BUILDING & PROTOTYPING

11 Clamps

For holding bits of your build together until it's stuck or you need to change it.

12 Crocodile clips

Connect one component to another without soldering, tape, or breadboard.

13 Crimping tool

Pinches connectors onto wire, fixing it to the connector. Can also can cut/strip wires.

14 Screwdrivers

You're going to need several different screwdrivers of different sizes/types.

15 Craft knife

Stanley knives are good for most work, but you might also want something lighter.

16 Tweezers

Tweezers can get you out of a fix when that little screw disappears.

17 Hot glue gun

Sometimes there is no substitute, but it's not suitable for the young 'uns.

18 Hand drill

Most of the time a cordless is best.

19 Saw

Hacksaws for metal/plastic parts, jigsaw/circular for big builds, mitre saws for angles.

20 Pliers

A good set of long-nose/snipe-nose pliers should be on hand at all times.



Because your Raspberry Pi generally relies on being booted from an operating system on a microSD card, you will require a way to write that data first. The open-source program you need for this is called Etcher from Balena.

magpi.cc/etcher

If you are making any sort of moving robot, you're probably going to need wheels

22 GEARS & WHEELS



If you are making any sort of moving robot, you're probably going to need wheels or gears. Some kits come with them, such as the CamJam EduKit 3. You can buy them separately or even 3D-print them.

magpi.cc/edukit3

TOOLS FOR STORAGE

23 USB stick

Add extra storage to your Raspberry Pi with a USB thumb drive. You can store more files than the microSD card, and you can transfer files from another computer to your Raspberry Pi.



24 Memory card

Make sure you have multiple microSD cards. An 8GB microSD card can be used to install Raspbian and other operating systems. It's a good idea to have your regular microSD card and at least one other card for testing out projects.

25 Storage box

When making projects, you quickly rack up additional kit – not to mention cables, components, and Raspberry Pi boards. Keep everything in one place by adding a storage box to your shopping list.

26 CARD TRIMMER



A card trimmer will enable you to cut crisp straight lines in paper or thin card. It's much quicker than a ruler and craft knife. Small ones are good for trimming photos, and a good-quality trimmer will last for years.

magpi.cc/papertrimmer



27 ROBOTIC CRAFT CUTTER



These machines are very much like plotters, but have a knife instead of a pen. They can cut very intricate designs out of paper and fairly thick card. They can be a bit temperamental and often need a sticky carrier sheet to work well.

magpi.cc/vinylcutters



28 LASER CUTTER OR ENGRAVER



Laser cutters and engravers fire a laser at a material to either cut through it or to leave a mark or indent on the surface, depending on the intensity of the laser. Very effective for cutting irregular wooden shapes.

magpi.cc/lasercutter



29 LAMINATOR



Lamination is the process of coating a material with a plastic film. This protects the material from moisture and other damage. It also makes the material thicker. Plastic laminate comes in various thicknesses and surface finishes.

magpi.cc/laminator



30

CONDUCTIVE PAINT



Bare Conductive® / Electric Paint®

Generally coloured black and supplied in a tube, this paint can be used to draw electrical circuits on paper or card, or to improve poor connections between components and even to create touch-sensitive areas on materials. Dries in a few minutes.

magpi.cc/electricpaint

32

POLYMORPHIC PLASTIC



This material is usually supplied as granules or beads. You can heat them up in boiling water and they will clump together in a mass. While still warm, you can mould the plastic before it sets hard in ten minutes or so.

magpi.cc/polymorphicplastic



TOOLS FOR CONNECTING ELECTRONICS

34 GPIO ruler/chart

GPIO (general-purpose input/output) pins are the interface between your Raspberry Pi and electronic components. Until you learn which pin is which, you'll need a guide. There are several GPIO rulers and guides available to make life easier.

35 Header connector

For some projects, you may want to connect a HAT or pHAT to your Raspberry Pi, but not place it right on top of the board. In this case, a female-to-male 40-way ribbon connector enables you to extend the reach of the GPIO pins.

36 Jumper wires

Jumper wires connect Raspberry Pi GPIO pins to electronic components. Use them with a breadboard to prototype your circuit, or solder them directly to components. Be sure to get a variety of different colours to make your project easier to understand.

31

ACRYLIC & THERMOPLASTIC SHEET



Acrylic plastic sheeting is very tough and can be cut using a saw or CNC router, whereas thermoplastic sheets are malleable when heated with a heat gun. Once cool again, thermoplastic retains its new shape.

magpi.cc/plasticsheets



33

ALUMINIUM FOIL



Available from convenience stores and supermarkets, foil can be used as a conductor in a circuit, as shielding to reduce electromagnetic interference, and also to reflect light and heat either away from or towards an area.

magpi.cc/alufoil

3 ISSUES FOR £5



- 📞 Subscribe by phone: **01293 312193**
- 👉 Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

TOOLS FOR SOLDERING

37 Desoldering kit

This equipment enables you to clean melted solder from components on a printed circuit board, allowing their removal or replacement.

38 Soldering station

A handy stand to place a soldering iron while it is still plugged in and hot. The sponge should be wet to clean the end of the iron.

39 Soldering iron

An absolute necessity for soldering or desoldering components onto printed circuit boards. For detailed work, an iron with a pointed end is best.

40 Third hand tool

Usually has a solid, heavy base with jointed arms with clips or holders at the end, and a magnifying glass for working on small components.



41 MULTIMETER



You need to be able to measure current, resistance, and voltage in your circuits. Digital multimeters can cost as little as £10.

magpi.cc/multimeter



42 GORILLA GLUE



Similar to superglue, but is supplied with a brush to apply. Sticks most materials and very good for 3D-printed parts. Usually takes around half an hour to dry.

magpi.cc/plasticsheets



43 NUTS, BOLTS & SCREWS



Having a good range of sizes and shapes of nuts, bolts, and screws is absolutely necessary for making anything that you are not going to glue. Keep a jar full of spares.

44 STICKY TAPE



For temporarily holding things in place or insulating wires. You can also use tape for wrapping handles of tools. Also good for removing cat fur from jumpers.

45 BLU TACK



Usually used for keeping posters on walls, but can also be used for holding components in place while soldering. It also rubs out pencil marks if you don't have an eraser.

46

POWER BANK / BATTERIES



Most maker projects need power, so it is a good idea to have a range of battery holders. Power banks for recharging phones can also be used for 5V supplies.

magpi.cc/pimoronipower



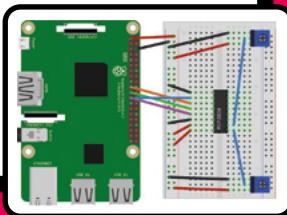
48

FRITZING



When you have completed your project, you may want to document how you did it. Fritzing is a great program for laying out circuit diagrams.

fritzing.org



50

DOCUMENTATION



One of the most important tools you will use for just about any maker project is reference material. Your Raspberry Pi may have come with some tips and hints about making, and many kits have worksheets and Frequently Asked Questions sections with them. Also, make best use of the online resources that detail other people's experiences: like Stack Overflow ([stack overflow .com](http://stackoverflow.com)) for information about any technical subject; blogs such as modmypi.com, recantha.co.uk, and blog.pimoroni.com; and of course the Raspberry Pi site (rpf.io) and magazines. If you're stuck getting something to work, it's likely someone else has had the same problem!

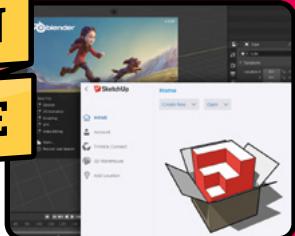


47

3D CREATION SOFTWARE



You may want to visualise your build before starting, and there are many 3D software packages to help you for free, like Blender; or with free trial versions, like SketchUp.

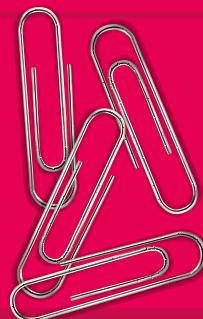


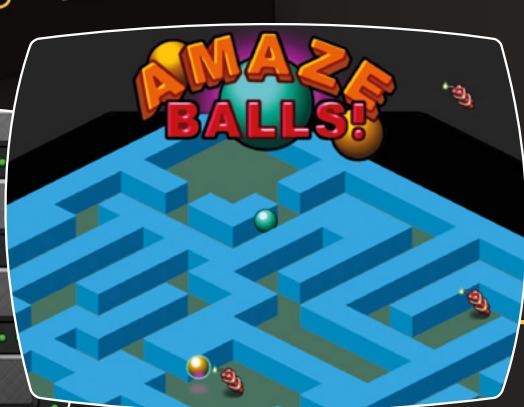
49

PAPER CLIPS



Can be used to connect components, clean out small nooks and crannies, or hold materials in place. Apparently, you can pick locks with them too!





RETRO GAMING WITH RASPBERRY PI

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games.

Build your own portable console, full-size arcade cabinet, and pinball machine with our step-by-step guides. And learn how to program your own games, using Python and Pygame Zero.

- *Set up your Raspberry Pi for retro gaming*
- *Emulate classic computers and consoles*
- *Learn to program retro-style games*
- *Build a portable console, arcade cabinet, and pinball machine*



BUY ONLINE: magpi.cc/store

Part 06

Add navigation to your low-cost robot



Danny Staple

MAKER

Danny makes robots with his kids as Orionrobots on YouTube, and is the author of *Learn Robotics Programming*.

orionrobots.co.uk

In the last article, readers added a camera to a Raspberry Pi-powered lunchbox robot. They got a photo from the robot's-eye view of the world. Now a robot builder gets to take this much further and make the robot use this camera to make decisions about the world.

This tutorial shows how to make an environment for testing computer vision. It demonstrates using OpenCV to condition images, to remove noise and simplify them. Then readers see how to extract data or check the content of an image and use this to make a robot turn.

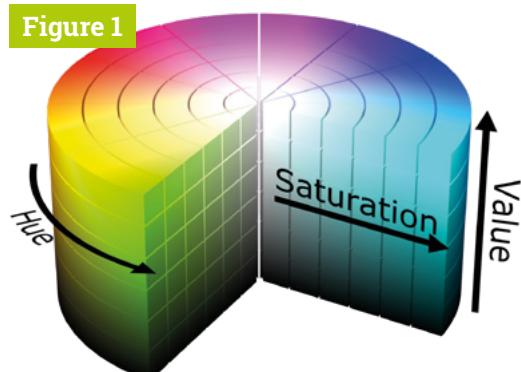
01 A test course

For trying out behaviours, robot builders make test courses. The goal is to create an environment with only the specific features to try out the robot.

Find a floor area in a neutral colour. Somewhere white or grey without patterns or colour.

Make walls with flat colours like red, blue, green, and yellow. Toy-boxes or coloured card work for

Figure 1



▲ Figure 1 The HSV colour space

You'll Need

- ▶ Lunchbox Robot with Camera magpi.cc/88
- ▶ A neutral floor area
- ▶ Good lighting
- ▶ One of:
Brightly coloured toy-boxes;
300 gsm brightly coloured card

this. Use white or neutral background walls. Cameras take better pictures with bright and consistent lighting. In good lighting, colours are clearer, making processing easier.

Good options are daylight or bright white indoor lighting. Avoid tinted or patchy lighting.

02 Installation

This step may take some time. Plug a mains-powered USB adapter into the robot's Raspberry Pi before proceeding.

Before installing the packages, make sure Raspbian is up to date with:

```
sudo apt update -allow-releaseinfo-change
```

There are some system packages needed for running the Python libraries.

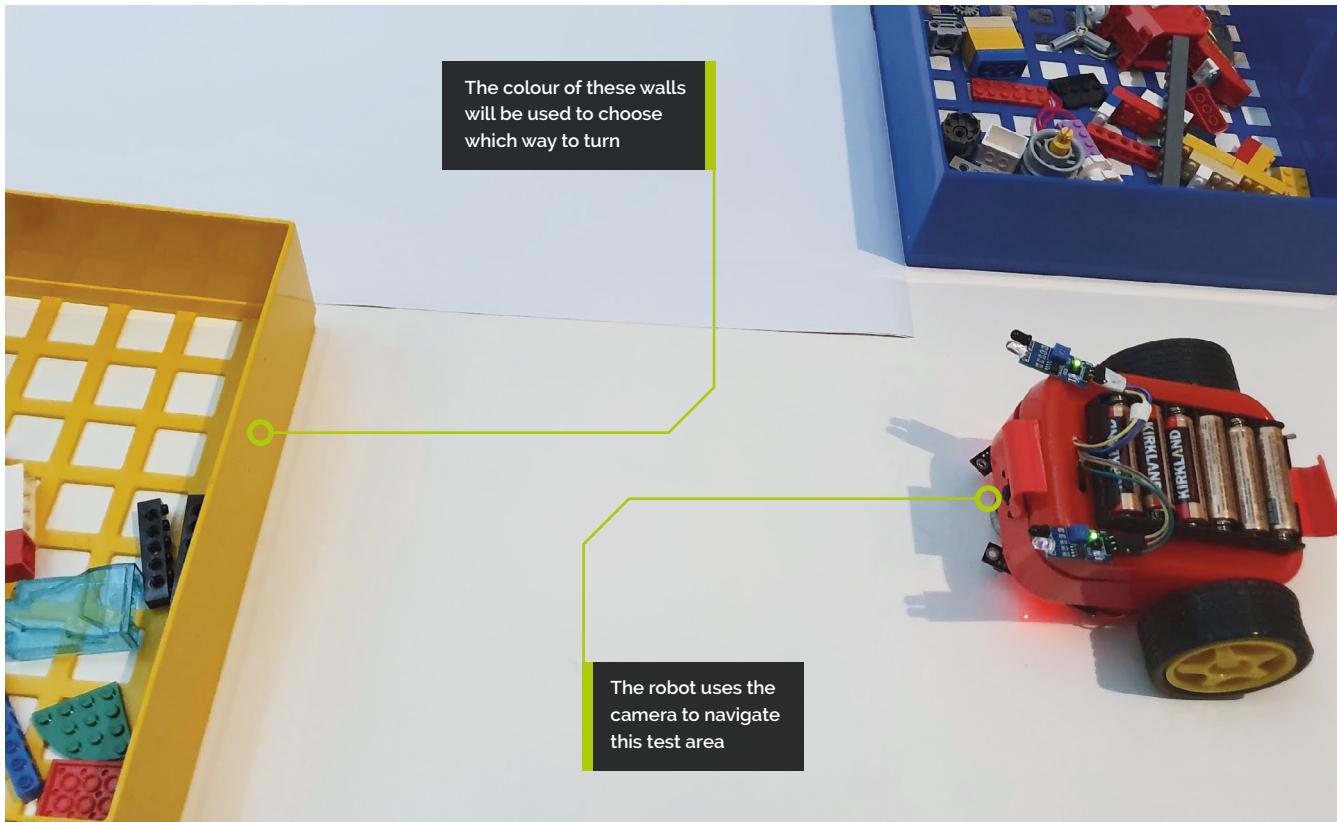
```
sudo apt install libcairo-gobject2 libwebp6
libilmbase23 libgdk-pixbuf2.0-0 libjasper1
libpango-1.0-0 libavcodec58 libavutil56
libcairo2 libswscale5 libatk1.0-0 libgtk-3-0
libtiff5 libpangocairo-1.0-0 libavformat58
libopenexr23 libfftw3 libgfortran5 libatlas3-base
```

Finally, install the Python packages needed for OpenCV, NumPy, and picamera:

```
sudo pip3 install opencv-python-headless
numpy imutils picamera[array]
```

03 Set up the camera

The function `setup_camera` in the file `find_contours.py` gets the camera ready.



“ Our robot’s camera is upside down, so the rotation is set to 180 degrees ”

For quick processing time, and to simplify the image, line 11 sets a camera resolution of 128×128. Our robot’s camera is upside down, so the rotation is set to 180 degrees. Using camera features saves processing on Raspberry Pi.

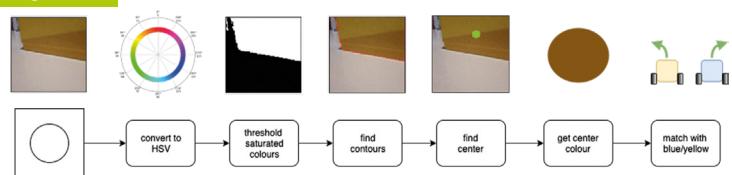
Line 14 creates `capture_buffer`, space to store image data from the camera. Lines 15 and 16 start the camera with two seconds of warm-up time.

With the robot in front of a coloured wall, run the following commands:

```
export LD_PRELOAD=/usr/lib/arm-linux-gnueabihf/libatomic.so.1
python3 find_contours.py
```

This code sends the camera’s captured image to the file `original.png`.

Figure 2



04 A little colour theory

Computers store colours as RGB or BGR, for red, green, and blue pixels.

In `findContours.py`, on line 21, we convert the image from BGR to the HSV colour system, which is suitable for this image processing.

Figure 1 shows how HSV works. Saturation measures how vivid or intense the colour is, from a low value being white or grey, to a full value being vivid. Hue indicates the colour – red, orange, blue, green, yellow, etc.

Transforming the image into HSV – Hue, Saturation, and Value – lets the robot pick out colour intensity (saturation) and then find its tint (hue), while mostly ignoring the colour brightness (value).

▲ **Figure 2** It takes a few steps for visual processing, with a number of transformations. A pipeline is a useful way to visualise this

findContours.py

▶ Language: Python 3

DOWNLOAD
THE FULL CODE:

 magpi.cc/robotCV1

```

001. import time
002. import imutils
003. import numpy as np
004. import cv2
005. from picamera.array import PiRGBArray
006. from picamera import PiCamera
007.
008.
009. def setup_camera():
010.     camera = PiCamera()
011.     camera.resolution = (128, 128)
012.     camera.rotation = 180
013.
014.     capture_buffer = PiRGBArray(camera, size=(128, 128))
015.     camera.start_preview()
016.     time.sleep(2)
017.     return camera, capture_buffer
018.
019.
020. def get_saturated_contours(image):
021.     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
022.     # Mask for vivid colours
023.     masked = cv2.inRange(hsv, np.array([0, 140, 30]),
024.                           np.array([255, 255, 255]))
025.     # Find Contours
026.     cnts = cv2.findContours(masked.copy(), cv2.RETR_EXTERNAL,
027.                             cv2.CHAIN_APPROX_SIMPLE)
028.     contours = imutils.grab_contours(cnts)
029.     contours = sorted(contours, key=cv2.contourArea, reverse=True)
030.
031.     colour = [0, 0, 0]
032.     if len(contours) > 0:
033.         # Find the center of the contour
034.         m = cv2.moments(contours[0])
035.         if m["m00"] > 0:
036.             cx = int(m["m10"] / m["m00"])
037.             cy = int(m["m01"] / m["m00"])
038.             colour = hsv[cy, cx]
039.
040.     if __name__ == '__main__':
041.         camera, capture_buffer = setup_camera()
042.         camera.capture(capture_buffer, format="bgr")
043.         image = capture_buffer.array
044.         masked, contours, found_colour = get_saturated_contours(image)
045.         cv2.imwrite('original.png', image)
046.         cv2.imwrite('masked.png', masked)
047.         cv2.drawContours(image, contours[:1], -1, (0, 255, 0), 1)
048.         cv2.imwrite('with_contours.png', image)
049.         print(found_colour)

```



Figure 3

▲ Figure 3 Example of a masked or thresholded image. Pixels are only on (white) or off (black)

05 Image processing pipelines

The code processes images from the camera through a series of transformations to find the colour of a wall. Each transform is a small step; for example, finding all the pixels that match a criteria or making an outline of an area.

Later stages use the transformed output of earlier ones. The outputs are joined to other inputs, forming a pipeline.

A diagram like **Figure 2** (previous page) shows where data flows from one process to another, making it easier to understand what is going on. Use images from real outputs, boxes for stages, and lines to show the flow of data.

06 Thresholding or masking

Thresholding tests if every pixel has values within a range. Line 22 of **findContours.py** uses **cv2.inRange** for this. It makes a new binary image, storing **True** if the pixel has values between the lower limits and the upper limits.



Figure 4

▲ Figure 4 This is the original image, after a contour has been drawn from the threshold image and drawn back on it

The `find_contours.py` range allows all hue values while filtering for saturation values over 140, for only vivid colours and the value component to values brighter than 30.

The output file `masked.png` shows the output, with coloured walls in white (see **Figure 3** for an example).

The S and V values of the lower bound on line 22 can be adjusted up if too much area is matching, or down if too little is.

OpenCV can inspect a black and white image and find outlines for different areas. It calls these outlines contours

07 Finding contours

OpenCV can inspect a black and white image and find outlines for different areas. It calls these outlines contours.

In `find_contours.py`, lines 28 and 29 obtain a list of contours. Each contour is a list of points describing the outline.

On line 30, the contours are sorted by area. By finding the first contour in this list (the biggest), the code has likely found the most significant coloured area.

On line 48, the contour is drawn out to a debug image `with_contours.png`. Run the code and download the image to see how the contours look (see **Figure 4** for an example).

Figure 5

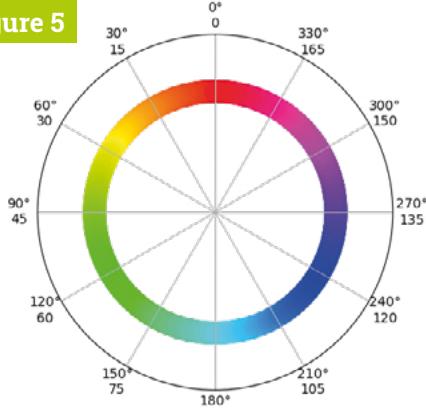


Figure 5 A hue colour wheel is handy for looking up colours. The figures under the degrees show the OpenCV values

08 Finding the colour

For this code to choose by colour, it needs the hue from the middle of the contour. It takes this colour from the original picture.

The robot uses OpenCV `moments` for finding the middle of a contour.

By dividing the sum of X coordinates (`m10`) by their count (`m00`), the code obtains the average X, their centre. The code also obtains the average and centre of the Y coordinates (`m01` divided by `m00`). The middle of the contour comes from combining these.

The code on line 36 of `find_contours.py` extracts the colour from the `hsv` output at the middle of the contour.

09 Using the pipeline in a robot

The `get_saturated_colours` function is imported from `find_contours.py`, enabling this code to reuse the pipeline from already tested code.

camera_nav.py

DOWNLOAD
THE FULL CODE:

► Language: Python 3

magpi.cc/robotCV2

```

001. import gpiozero
002. from camera_setup import setup_camera
003. from find_contours import get_saturated_contours
004.
005. robot = gpiozero.Robot(left=(27, 17), right=(24, 23))
006. camera, capture_buffer = setup_camera()
007.
008. for raw in camera.capture_continuous(capture_
009. buffer, format="bgr"):
010.     image = raw.array
011.     masked, contours, found_colour = get_saturated_
012.     contours(image)
013.     print(f"Colour {found_colour}, h value: {found_
014.     colour[0]}")
015.     if 5 < found_colour[0] < 40:
016.         print("yellow")
017.         robot.left()
018.     elif 100 < found_colour < 135:
019.         print("blue")
020.         robot.right()
021.     else:
022.         robot.forward()
023.     capture_buffer.truncate(0)

```

Top Tip

Lighting matters

Lighting should be neutral in colour, bright and diffused. Spotlights, low light, and coloured lights cause problems with visual processing.

A continuous stream of images is needed to use the pipeline to drive the robot. Line 8 of **camera_nav.py** creates this stream; line 9 extracts the data. Line 8 sets up the main loop as a **for** loop that runs forever with a new image each time.

The main loop puts the image through the pipeline and uses the output to determine if the robot turns right, left, or goes forward. The camera's image rate sets the timing.

The colour returned by `get_saturated_colours` is HSV.

10 Matching the colour

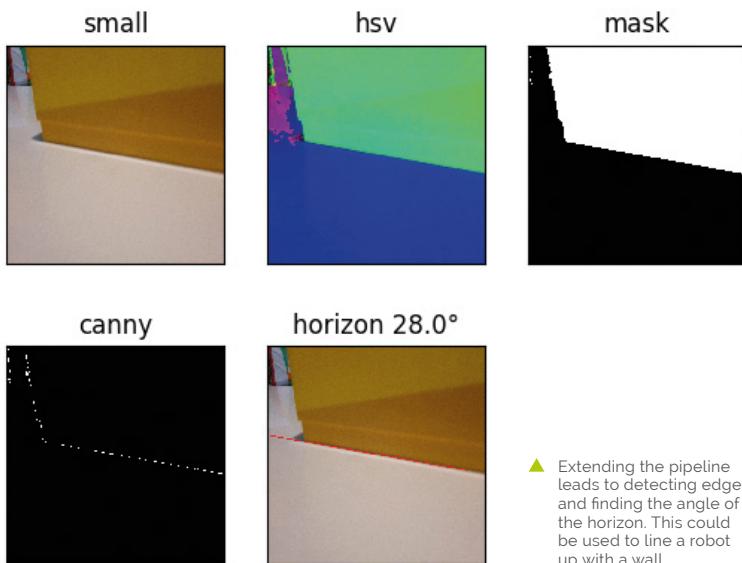
The **camera_nav.py** code uses the hue component from `get_saturated_colours`.

OpenCV stores a hue value as degrees divided by 2 to fit into 8 bits (up to 255). **Figure 5** (previous page) shows a colour wheel with hue values in degrees and OpenCV values.

The code in **camera_nav.py** matches a yellow range on line 12, and a blue range on line 15, printing the matched colour and turning the robot.

By setting up a series of walls of different colours, the robot can now navigate by wall colours. Expect to change these ranges for different test areas.

Ensure the robot is on battery power and in the test course before running this.



Extending the pipeline leads to detecting edges and finding the angle of the horizon. This could be used to line a robot up with a wall

■ A continuous stream of images is needed to use the pipeline to drive the robot ■

11 Improving robot vision

The `find_contours.py` code is a simple demonstration of computer vision. It's also easy to confuse it. Finding the image under the contour and averaging the colour would make it more stable.

The code could be combined with distance sensors, so only walls close enough were detected. Encoders or an inertial measurement unit (IMU) could be added to make a precise turn.

Advanced techniques like Canny Edge Detection with `HoughLines` could pick out the horizon, determining the angle and distance, so the robot could line up with a wall. OpenCV can do face detection and even has machine learning and neural network modules.

12 Further reading

Robot vision is a significant area of study in robotics, and this article has barely scratched the surface. It's one of the more rewarding and exciting spaces of robotics, worthy of further reading.

The PyImageSearch site (pyimagesearch.com) is a superb resource to learn more about computer vision and dig further into detecting different attributes from an image.

Your article author's book, *Learn Robotics Programming*, has a section on computer vision, building face- and object-following behaviours, and casting the camera and pipeline stages to a mobile phone browser to view in real time. ■

Top Tip

Reduce background clutter

A cluttered background causes the robot to detect random things. Neutral backgrounds without 'noise' make this easier to test.

MORE POWER TO STRATO PI AND IONO PI!

Strato Pi is a line of servers based on Raspberry Pi for the development of distributed/decentralized field control solutions. It is ideally suited for industrial applications where extreme reliability, ruggedness, and compliance with technical and safety directives are required.

Iono Pi is an extremely versatile I/O module that combines several digital and analog input lines, power relay outputs, and support for standard interfaces, typical of a modern PLC, with the powerful computing core of the Raspberry Pi.

On the trail of the new Raspberry Pi 4, we upgraded our products to fit its new design and capabilities. The re-engineered power-supply stage withstands the current demand of Raspberry Pi 4, including that of external devices powered via USB or through the auxiliary power supply output. The DIN-rail case now fits the new USB and Ethernet ports layout, as well as the new micro HDMI ports and USB-C power supply.



The new Strato Pi and Iono Pi are available on our website with pre-assembled Raspberry Pi 3 B+ or Raspberry Pi 4 2GB and 4GB.

Find out all the details on our website www.sferalabs.cc!

Smart classroom assistant



MAKER

Dale
Lane

Dale works for IBM and created the Machine Learning for Kids web-based tool.

magpi.cc/MLforkids

You can find this project and others on the Raspberry Pi Projects website!
magpi.cc/projectsonline

Create AI in Scratch using Machine Learning for Kids, and find out how machine learning really works!

In this project, you will use the Machine Learning for Kids tool (machinelearningforkids.co.uk) to make a smart virtual classroom assistant that reacts to what you say to it. You'll be able to control the classroom's virtual devices by typing in commands!

First, you'll create an assistant that uses a list of rules for understanding commands, and you'll learn why that approach isn't very good.

Next, you will teach the assistant to recognise commands for different devices by training it using examples of each command.

01 Get started

Head to machinelearningforkids.co.uk in a web browser. You'll then need to click on 'Get Started', and then click on 'Try it now'.

02 Create a project

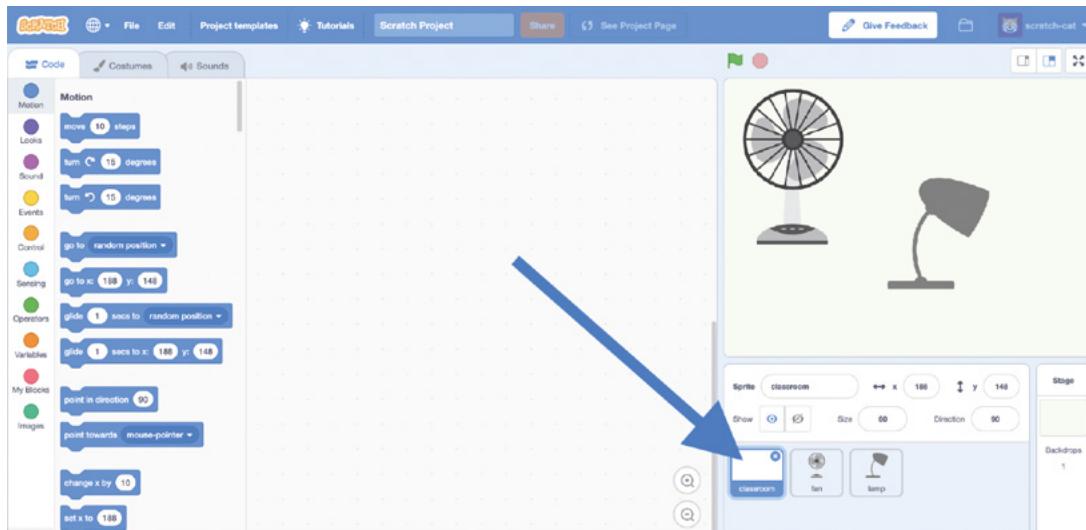
Click on Projects in the menu bar at the top, and then click on the '+ Add a new project' button. Name your project 'smart classroom' and set it to learn to recognise text, then click on Create. You should now see 'smart classroom' in the projects list; click on this project.

03 Prepare the project

Now we need to get a project ready in Scratch. Click on Make, click on Scratch 3, then click on 'Scratch by itself'. The page then warns you that you haven't done any machine learning yet. Ignore this and click on 'Scratch by itself' to launch Scratch.

Finally, click on 'Project templates' and then click on the 'Smart Classroom' template.





◀ Figure 1 Click on 'classroom' in the Scratch sprites pane

04 Add a list of rules

In this step, you will edit the project to include a list of rules to activate and deactivate the fan and the lamp. Click the classroom sprite to select it, as shown in **Figure 1**. Click on the Code tab and create the script shown in **Figure 2**. Once you've done that, click on File and then on 'Save to your computer', and save the program to a file.

05 First tests

Click on the green flag to test your program, and then type in a command and watch the program react! The following commands should all work:

Turn on the lamp

Turn off the lamp

Turn on the fan

Turn off the fan

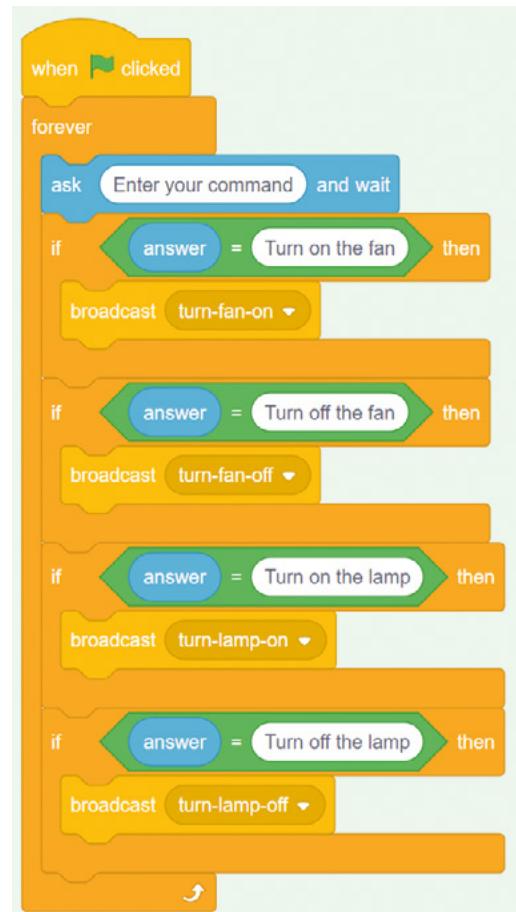
Type in anything else and your program does nothing! Even if you make a small spelling mistake, the program does not react.

06 Beyond rules

You're telling your virtual classroom assistant to react to commands using a simple rules-based approach. But if you wanted your program to understand commands that are phrased differently, you would need to add extra 'if' blocks.

The problem with this rules-based approach is that you need to exactly predict all the commands the smart classroom assistant will understand. Listing every possible command would take a very, very long time.

“The problem with this rules-based approach is that you need to exactly predict all the commands”



◀ Figure 2 The Scratch code for a rules-based approach

Top Tip

Machine learning

You need to tell an AI what to learn. The more you give it to learn with, the better it will be.

“ The more examples you give your program, the better the program should get at recognising your commands ”

Next, you will try a better approach: teaching the computer to recognise commands by itself.

07 Examples for training

Close the Scratch window and go back to the Training tool, then click on the ‘< Back to project’ link. Click on the Train button because you need to collect some examples so that you can train the computer. To collect different examples, you need to create ‘buckets’ to put the examples in.

To create a bucket, click on ‘+ Add new label’ and call the bucket ‘fan on’. Click on ‘+ Add new label’ again and create a second bucket called ‘fan off’. Create a third and a fourth bucket called ‘lamp on’ and ‘lamp off’.

Click on the ‘Add example’ button in the ‘fan on’ bucket, and type in a command asking for the fan to be turned on. For example, you could type ‘Please can you switch on the fan’. For the ‘fan off’ bucket, you’ll need to click ‘Add example’ again and then use something like ‘I want the fan off now’. Do the same for the ‘lamp on’ and ‘lamp off’ buckets.



▲ Add as many different and varied example phrases as you can for each command

learning model' button. If you have enough examples, the program should start to learn how to recognise commands from these examples.

10 Test the training

Wait for the training to complete. This might take a minute or two but once the training has completed, a test box appears. Test your machine learning model to see what it has learned by typing in one of the commands you added to a bucket, and then press **ENTER**. The command should be recognised.

Now type in commands that are not in the buckets. If you're not happy with how the computer recognises the commands, go back to the previous step and add some more examples. Then select the 'Train new machine learning model' button again.

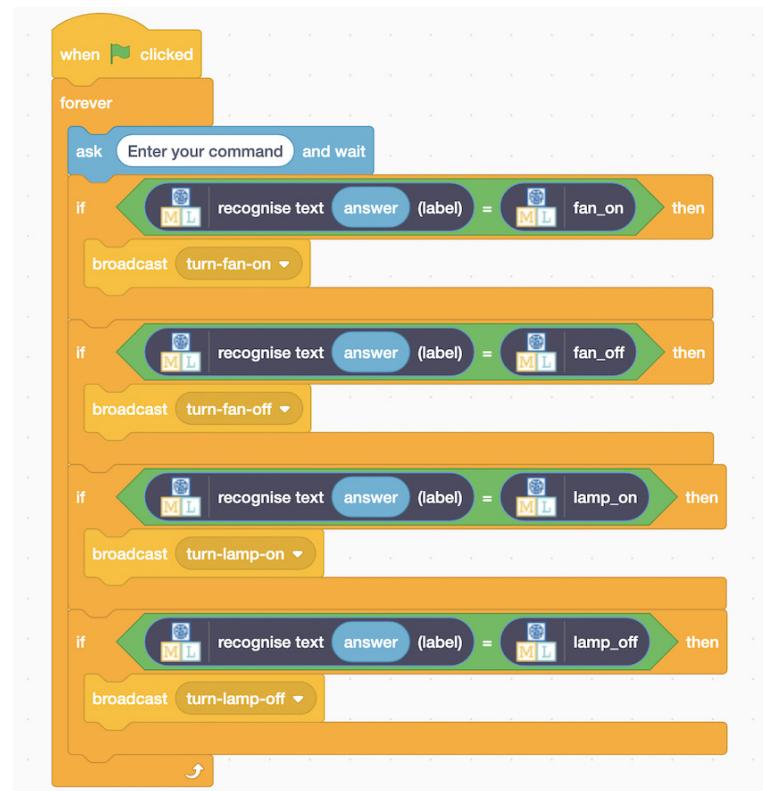
Instead of writing rules for the program, you are giving the program examples. The program uses the examples to train a machine learning model. Because you are supervising the program's training by giving examples, this machine learning approach is called *supervised learning*.

11 Use it in Scratch

Now update your Scratch program to include your machine learning model instead of the rules-based approach. Click on the '< Back to project' link, click on Make, then Scratch 3. Here you can read the instructions on the page to learn how to use machine learning blocks in Scratch.

Click on Open in Scratch 3, then on File and 'Load from your computer', and select the Scratch project you saved earlier. When Scratch asks you whether to replace the current project, click on OK.

Click on the Code tab, and update your Scratch code (**Figure 3**) to use your machine learning model instead of the rules you first added. The 'recognise text' block is a new block added by your project.



This new block can receive a message and return one of the four labels, based on the machine learning model you have trained.

▲ **Figure 3** Revised for a machine learning approach, the code features 'recognise text' blocks

12 Scratch AI

Click the green flag to test your new code. Test your project by typing a command and pressing **ENTER** on your keyboard. The fan or lamp should react to your command.

Make sure you test that this works even for commands that you didn't include as examples in the buckets.

Save your project as before. Your Scratch smart virtual classroom now uses a machine learning model instead of a rules-based approach. Using machine learning is better than using rules, because training a program to recognise commands for itself is much quicker than trying to make a list of every possible command.

The more examples you use to train the machine learning model, the better your program should get at recognising commands.

If you want to learn about how you can improve the model with 'confidence scores', head here: magpi.cc/smartclassroom

Top Tip

Go further

Can you get the model to tell you the weather or date? Give it a go!

Add internet access to a vintage computer



**PJ
Evans**

PJ is a writer, coder, and Milton Keynes Raspberry Jam wrangler. In the dial-up days he created pages for Prestel, which ages him greatly.

@mrpjevans

MAKER

Missing dial-up? Want to post to a BBS? Build an interface to the modern world for your old computer with a Pi232

The bulletin boards of the 1980s haven't died out quite yet, but they have gone online. There's only a handful of BBSes (bulletin board systems) that you can dial up with your old-time modem. If you like the idea of accessing an internet-based BBS from your Atari ST, Amiga, or BBC Micro, then we can use a Raspberry Pi computer to act as a translator. You can even use modern services like Twitter! All we need to do is get Raspberry Pi talking in one of the most popular communications protocols of all time: RS232.

circuitry features a further simplified version often referred to as UART (universal asynchronous receiver/transmitter) running at 3.3V, but we can adapt it to 'full' RS232 at 12V so that older computers can handle the signal.

You'll Need

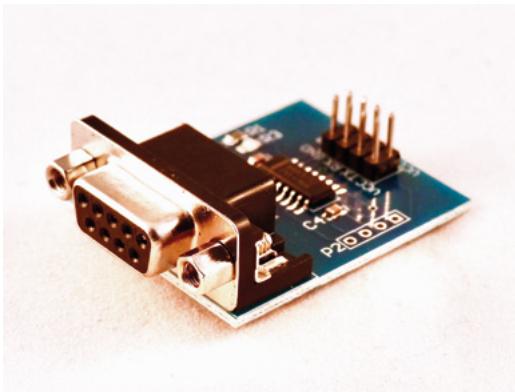
- MAX3232CPC
RS232 transceiver
magpi.cc/MAX3232
- Small piece of stripboard, e.g. magpi.cc/striptboard
- 5 x 100 nF capacitors
magpi.cc/100nF
- 1 x 9-pin D-sub socket
magpi.cc/D-sub
- 1 x PL2303-based USB/RS232 cable (optional) magpi.cc/USB-RS232
- Jumper leads (socket ending)

01 A little history

Every Raspberry Pi computer has a form of serial access, which is one of the oldest and simplest ways of communicating with computers. These days you're more likely to use TCP/IP, the protocol of the internet, to chat to a remote device, but back in the day it was RS232 that ruled the communications world. Raspberry Pi

02 Study and shop

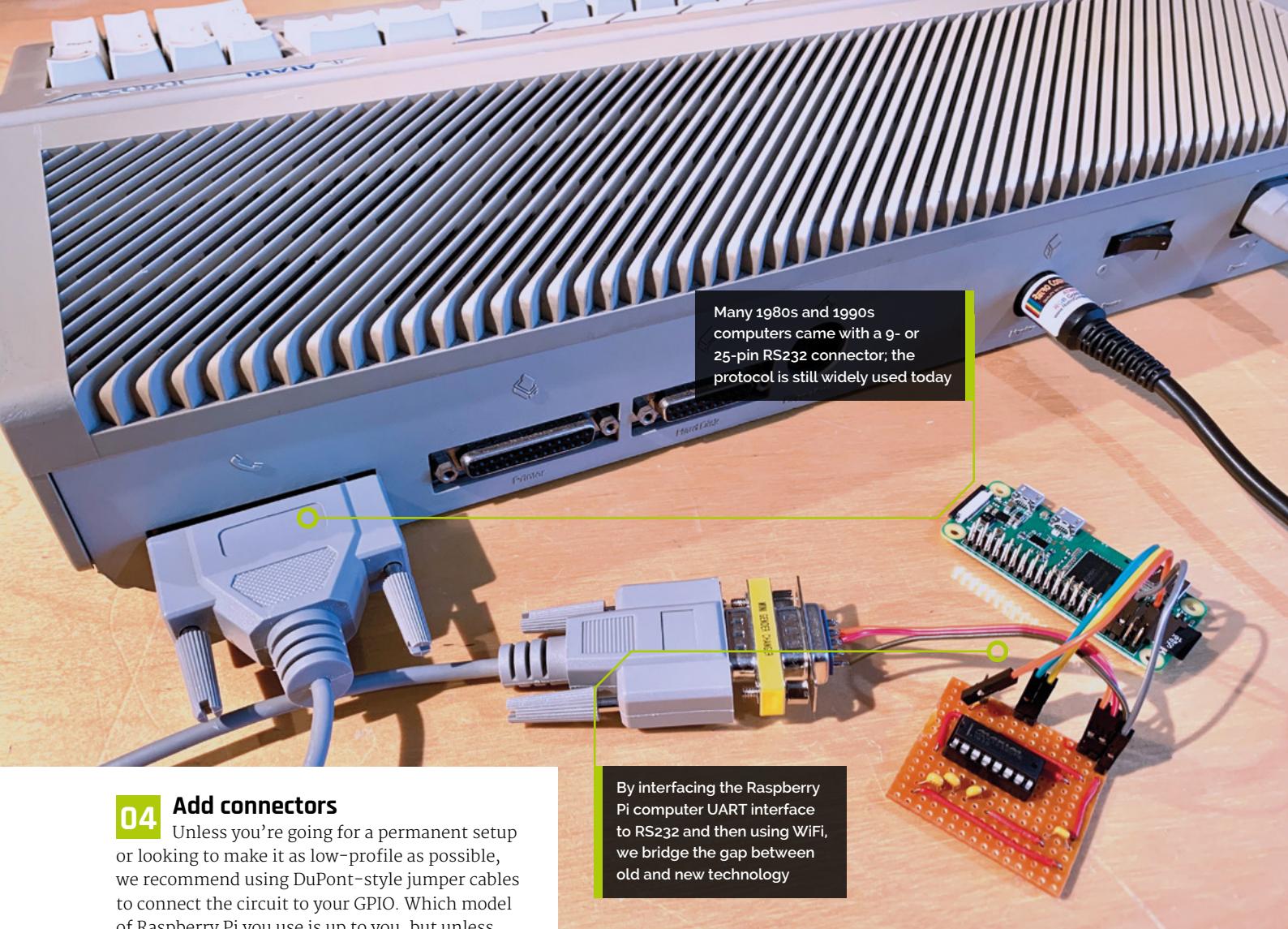
To 'downgrade' a Raspberry Pi to RS232, we'll need to build a small circuit to act as a transceiver so the vintage computer can 'hear' our virtual modem and the RS232 signals do not fry our delicate Raspberry Pi board. Luckily, all the heavy lifting for such a job can be done by the MAX3232CPC integrated circuit. All you need to do is wire it up, add a few capacitors, and you'll be BBSing in no time. Have a look at the circuit diagram (**Figure 1**, overleaf). We've deliberately made it larger than it needs to be for simplicity. More experienced makers will be able to reduce it in size if they wish.



▲ Cheat #1: If you don't fancy soldering, you can buy these pre-assembled units (for a few pounds) that can connect to the GPIO

03 Soldering the main circuit

Time to get building. Thankfully, the circuit itself is not very complicated, but it's also unforgiving, so make sure all the wiring is in the correct place or nothing will work! We recommend starting with the IC socket, then the wiring, followed by the capacitors, finishing with the headers (optional if you'd like to use jumper cables). The result takes the two transmit (TX) and receive (RX) lines from the GPIO and feeds them into the IC. The MAX3232CPC converts the data into RS232 standard and outputs them to the 9-pin D connector and vice versa. Check and double-check everything.



04 Add connectors

Unless you're going for a permanent setup or looking to make it as low-profile as possible, we recommend using DuPont-style jumper cables to connect the circuit to your GPIO. Which model of Raspberry Pi you use is up to you, but unless you have a specific use case in mind, a Raspberry Pi Zero W is the perfect choice as it's small and there's little horsepower required. You will also need to solder up the 9-pin D connector. The diagram (**Figure 1**) shows the connector from the solder (rear) side. Make sure you wire up correctly to pins 2 and 3 and ground it too.

The circuit itself is not very complicated, but it's also unforgiving

05 Raspberry Pi time

Your Raspberry Pi Zero W will need a bit of configuration before you can get going. Start with a fresh microSD card with Raspbian Lite (we don't need a desktop, but feel free to install 'full' Raspbian if you wish). This is also a great project for reusing any old low-capacity microSD cards you have, as there's not much software to install. Once booted up, make sure everything is up to date by running `sudo apt -y update && sudo apt -y upgrade`. Time to grab a refreshing beverage as Raspbian applies all the latest updates.

By interfacing the Raspberry Pi computer UART interface to RS232 and then using WiFi, we bridge the gap between old and new technology



06 Prepare your virtual modem

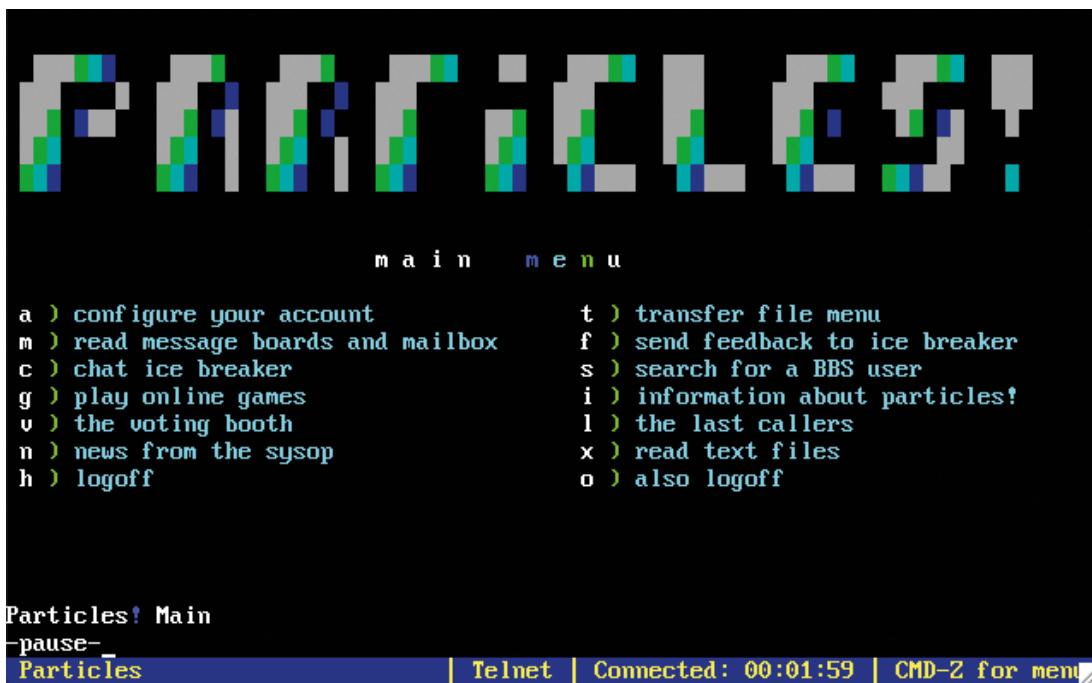
An 'out of the box' Raspbian installation isn't quite ready to go back to the 1980s just yet, so we need to do a little further configuration. From the command line, run `sudo raspi-config` to start the Raspberry Pi configuration utility. From the top of the menu, start by changing your password (optional, but recommended); then, under Network Options, set your host name (again, optional) and configure WiFi. Finally, under Interfacing Options, enable SSH and Serial. When asked 'Would you like a login shell?', select 'Yes'. Now exit the utility and shut down the computer (`sudo shutdown`).

Cheat #2: The simplest solution of all is the widely available USB-to-RS232 cables based on the Prolific PL2303 chipset

Top Tip

Not just vintage

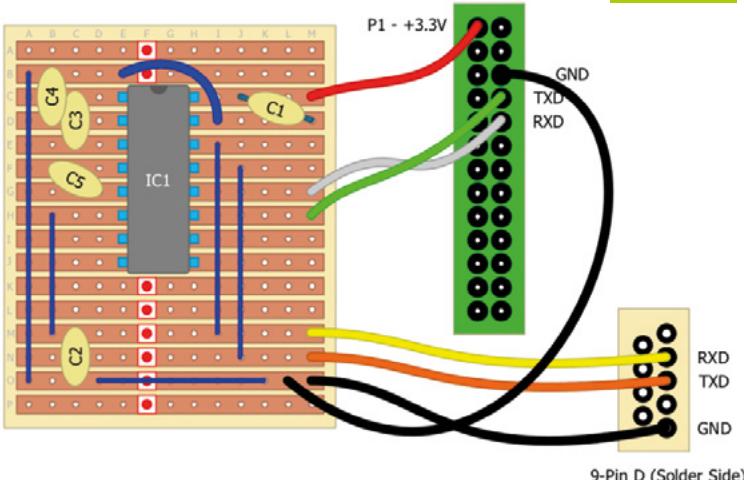
This project works well with older PCs. Although they refer to them as COM ports, their 9-pin connectors are really RS232.



► Many classic BBSes are still online, some even supporting PSTN modems. These days, most can be found at the end of an IP address

▼ **Figure 1** Here's the circuit we're going to build. It's not too complex, but it's imperative that the wiring is hooked up correctly. Check and double-check

Raspberry PI RS232 Adaptor



07 Testing time

Check your circuit board for any short circuits or dry solder joints. Once happy, connect it to the GPIO. There are four connection points: one for 3.3V (physical pin 1), ground (pin 9), and TX/RX (pins 8 and 10 respectively). We recommend getting an RS232/USB cable and testing with a modern computer first. Connect the 9-pin connectors together and boot your Raspberry Pi. Now, using your favourite Terminal emulator program (we like Serial for macOS), try to connect over RS232. Raspbian's default settings are 115,200bps (connection speed), eight data bits, no parity, one stop bit (aka 8N1).

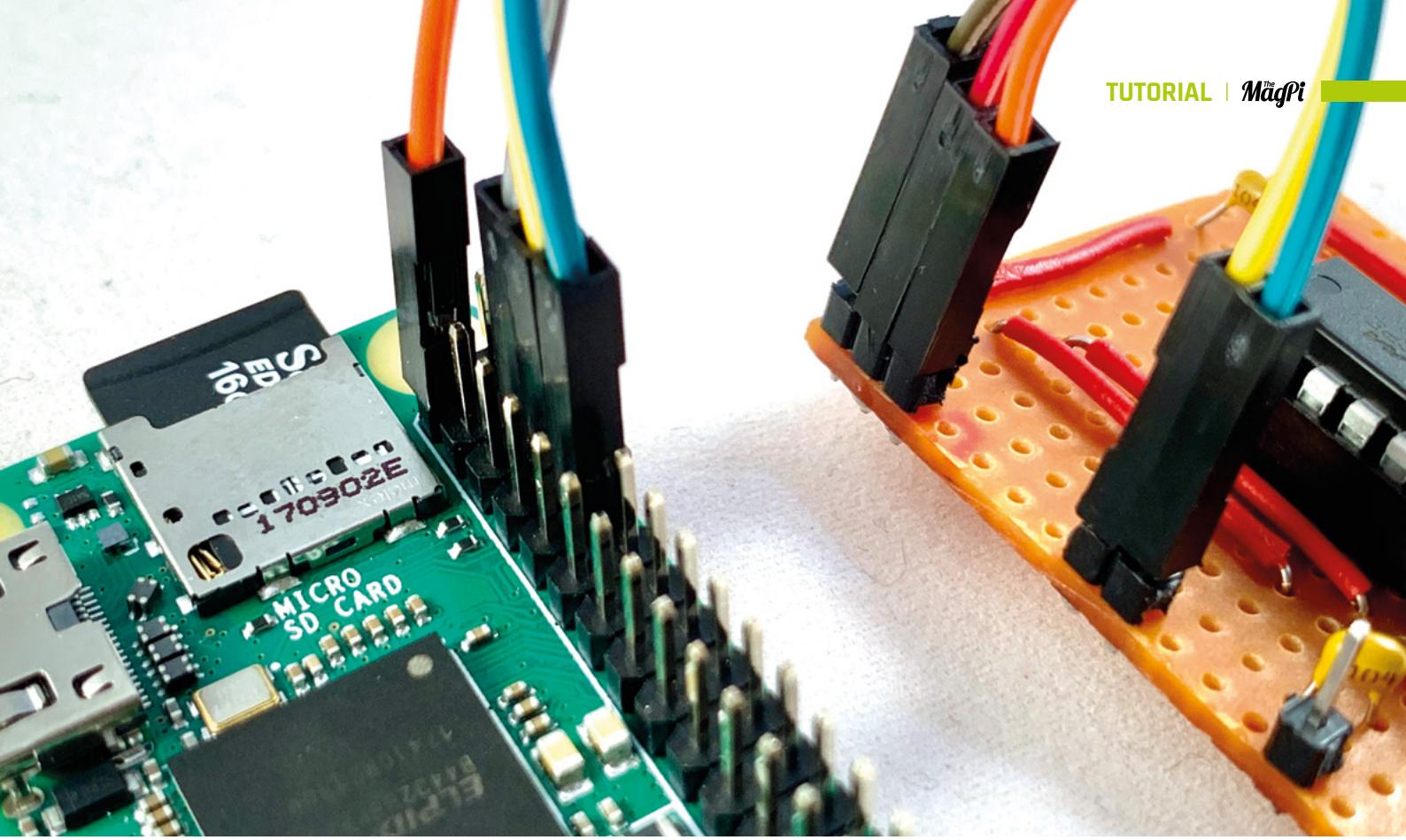
08 Troubleshooting

Having problems? RS232 can be a tricky beast as, unless everything is perfect, nothing will work at all. The most common issue will be polarity of the TX and RX lines. For things to work, the RX of the GPIO must be connected via the MAX3232CPC to the TX line of the computer and vice versa. Start by checking that your speed is correct and the protocol is set for 8N1. Also try 'reset' or 'send break' in your terminal emulator, which is sometimes needed to wake up the connection. Finally, try using a null modem cable (which crosses over the lines) or swapping the connectors to pins 8 and 10.

09 Get online

If everything is working, you should now be greeted with the Raspbian terminal login (you might need to press **ENTER** a couple of times to wake it up). You can now log in as normal. If the connection is behaving, feel free to try it out on a real vintage machine. Some may require adapters to connect (1990s home computers favoured 25-pin D connectors). You will probably need to reduce the speed of the connection, as many older computers cannot handle the default 115,200bps. Unless your computer is very old, 9,600bps will probably work. To set the port to this, enter the following from an SSH session:

```
sudo stty -F /dev/serial0 9600
```



10 Connect to a BBS

Most BBSes are now online and talk using a protocol called Telnet. It is very insecure, sending plain text over the internet and has since been replaced with SSH (Secure Shell). However, many of these services are being run on original equipment that has no support for SSH, so Telnet it is. The Telnet client software is not installed by default, so run `sudo apt install telnet`. Once finished, try connecting to the popular Particles BBS (running on an Apple II/e!), by entering:

```
telnet particlesbbs.dyndns.org 6400
```

Within a few seconds you'll see a welcome message. After you've had a look around, use **CTRL+]** followed by 'exit' to leave Telnet.

11 Emulate a classic modem

If you want a genuine experience, and full compatibility with vintage BBS software, it's easy to emulate the classic Hayes AT command-set. To install the emulator:

```
sudo apt install tcpser
```

Before running `tcpser`, go back into `raspi-config`, select 'Interfacing Options', then 'Serial', and answer 'No' when asked if you want a login shell to be accessible and then 'Yes' if you want the serial port to be enabled. Now start `tcpser` using SSH:

For a genuine experience, and full compatibility with vintage BBS software, it's easy to emulate the classic Hayes AT command-set **■**

```
tcpser /dev/serial0 -s 9600 -l 7
```

(You can change 9600 to be your desired speed). Try connecting over serial again and this time you'll be able to enter AT commands, replacing phone numbers with domain names.

12 Do more!

Your original 'comms' software should be blissfully unaware that it is talking over the internet. There's now another web of vintage BBSes to explore on your original machine, but why stop there? The Raspbian repositories are full of command-line software that can be accessed by your classic machine. Tweeting from an old 8-bit is always fun, so why not install 't', a command-line Twitter client (github.com/sferik/t)? Of course, you can now do anything in the Raspbian shell that you can in an SSH session, so get your BBC Model B to operate LEDs and switches, or have your Commodore Amiga send Telegram messages! **■**

▲ Polarity is the most common problem encountered when working with RS232. Remember that TX (transmit) goes to RX (receive) and vice versa

Top Tip

Careful of weird wiring

Do your research on your choice of classic machine. Some, such as the Cambridge Z88, use non-standard wiring to their connectors.

Part 03

Hack GraviTrax with Raspberry Pi



MAKER

**Mike
Cook**

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

magpi.cc/MikeCook



Warning!
Drill safely

Please use safety goggles and a desk clamp when drilling

You'll Need

- Stripboard
magpi.cc/striplboard2
- 40-pin, 2-row header pins
magpi.cc/2rowpins
- 2 x 40-pin, 1-row header pins
magpi.cc/1rowpins

Make your GraviTrax layout trigger LEDs for a dazzling sound and light show

In the first part of this series, we explored how to monitor the presence of a ball and how to use that to trigger sound. Then, last month we saw how to make animated LED parts run in background threads. Now we bring these two parts together and show you how to control them both using a text script and interface board.

01 Making the connection

In the first part (magpi.cc/87), we simply mentioned that the ball sensors had to be connected to the GPIO pins with an external pull-up resistor and left it at that. Likewise, we just used several I²C devices on the same bus. However, for a practical useable system, we need to have all the input and output on a single board that allows simple flexibility for setting up different track layouts. So, we have designed a board which allows us to connect up to twelve sensors, five I²C devices, and eight NeoPixel channels.

02 The schematic

The schematic (Figure 1) might look a bit repetitive, but that is the nature of what we are making. First of all, it incorporates the NeoPixel driving board we saw last month, but adds an option for powering the LEDs from an external supply to allow for much brighter displays. Then we have the input connectors: each has a 5.6 kΩ (5K6) pull-up resistor to 3.3V (3V3) on the GPIO input pin. Note the centre pin is 3.3V and the end pin is the input. Finally, we have five distributing I²C connectors running on a single set of I²C drivers, which also carry 5V power and ground.

03 Beginning the board

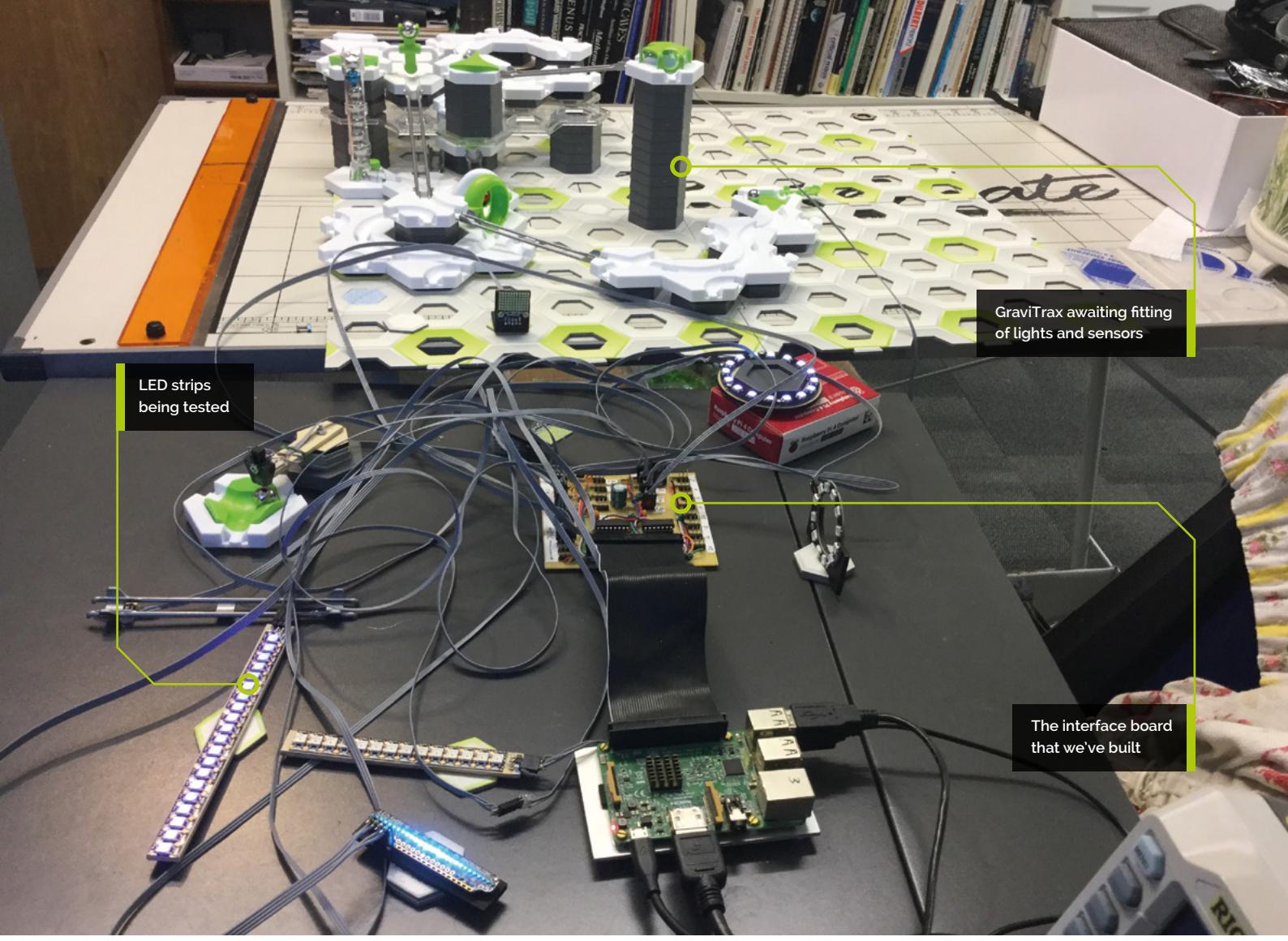
We used a piece of 39-row by 37-hole stripboard to make our distribution board, and a small length of 40-way ribbon cable to connect it to our Raspberry Pi. Figure 2 (overleaf) shows how we cut the tracks on the rear of the board. Note the cuts between holes we used for the ribbon cable's header pins; we did this with a scalpel, but you can use a small sharp knife for this – be careful! We found the best way to do this was to make two cuts across the track as close together as we could, and then remove the copper in between them with the blade.

We have designed a board which allows us to connect up to twelve sensors, five I²C devices, and eight NeoPixel channels.

04 Optional power jack

Wear goggles for this step

If you want to include a power jack for an external 5V supply for the LEDs, we recommend you prepare for that first. Unfortunately, the standard board-mounting jack connectors are not too friendly for being mounted on a 0.1 inch pitch board, and you need a bit of work to make it fit. We found the best way is using a 1mm routing bit in the drill; or, at a push, a Dremel. You need to run together several holes in order to make this fit. Figure 3 (overleaf) shows what you need to do. We recommend you practise this on a small scrap of stripboard first before you attempt it on the large board.



05 Construction

When faced with a large piece of stripboard, it is easy to get disorientated, so start by cutting the tracks for the header pin and soldering that to the board. Then cut the tracks for the two IC sockets and solder them in. This will give you a couple of good points of reference when building up the circuit. The final layout you are aiming for is shown in **Figure 4**, but we suggest you leave the flexible wires until last. If you want to see the layout before the flexible wires, there is an intermediate diagram on our GitHub page.

06 Construction order

We suggest that you first put in the copper wire links, shown as black lines. Then cut the tracks on the back. These are shown on the front view diagram as grey rectangles. Push a wire from the front into the hole of the track you want to cut, flip the board over, and cut the track at the point the wire shows. This way, we have found there is less chance of getting it wrong. Then add the header pins, followed by the resistors, capacitors, and power jack.

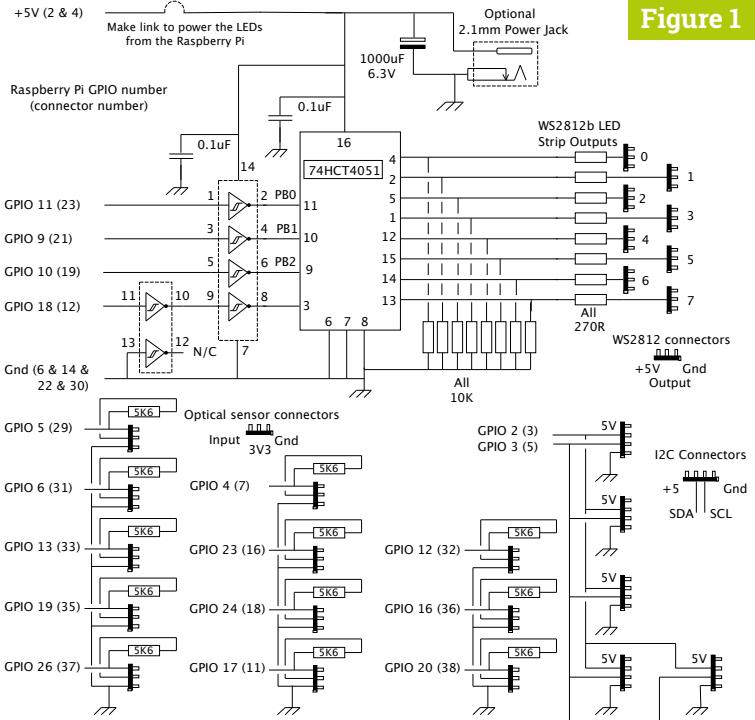
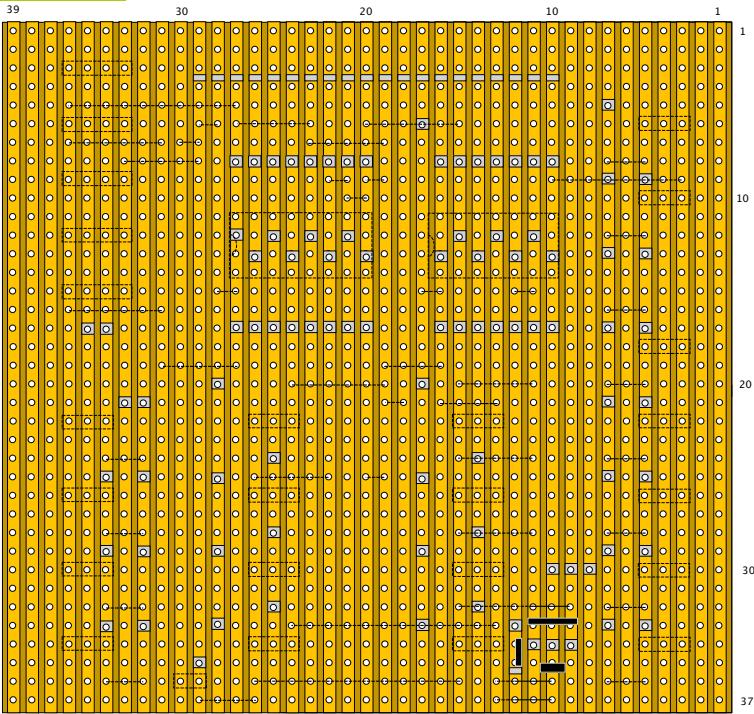


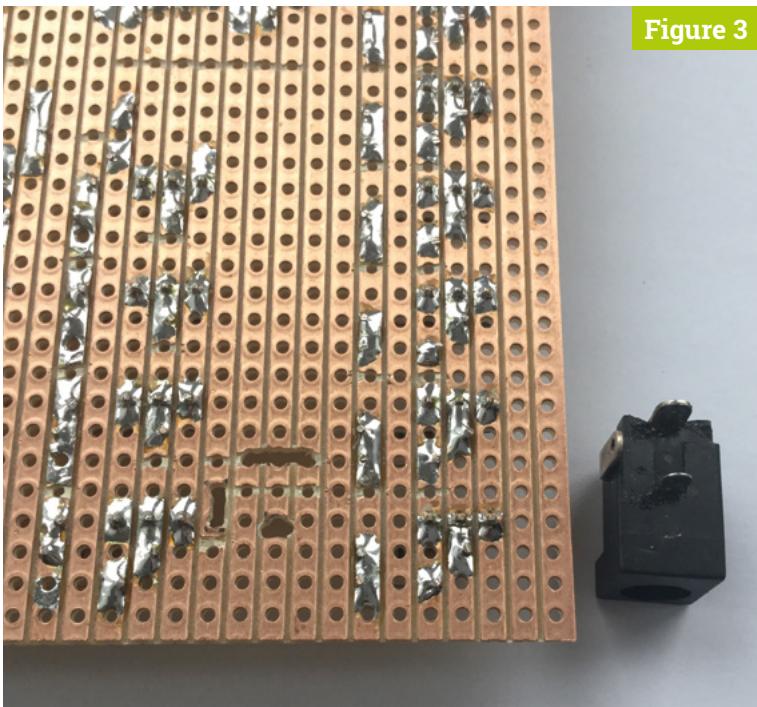
Figure 1

▲ **Figure 1** Here's a schematic of the interface board used in this project

Figure 2



▲ Figure 2 These are the tracks to cut on the rear side of the stripboard. Be careful!



▲ Figure 3 Here's what cutting the slots for the power jack looks like. Once again, be careful!

07 Flexible wiring

In order to complete the board, you need to add flexible wires to make the connections not made by tracks alone. Note that the layout diagram is designed to show what connections you need to make, and not necessarily what route they should take. For example, header pins 19, 21, and 23 are shown going over the top of the header; in practice, they should be routed round the left-hand side of the header pins, or even soldered on the underside of the board. Electricity is famously indifferent as to what colour wire it flows through – the colours we used are just to make the diagram more readable.

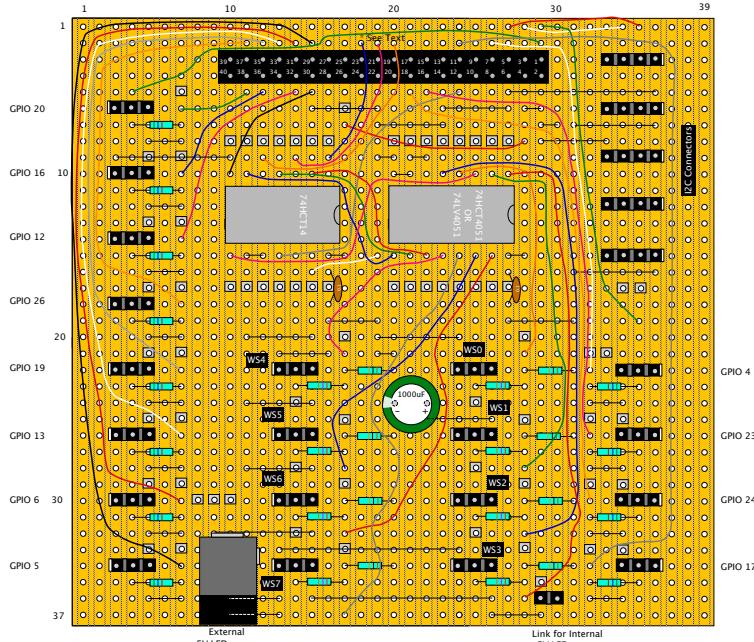
08 Finishing off

We gathered the wires together using small 2mm-wide cable ties. This is essential to prevent the wires resting in between the header pins and making them look untidy. We also gathered the wires away from the edges of the board to allow labels to be put onto the board. It is vital that when you connect a sensor, you know what GPIO pin it is connected to. We used sticky-backed labels cut to the appropriate size, and stuck them on the board, then wrote on them with a fine-tipped pen (Figure 5).

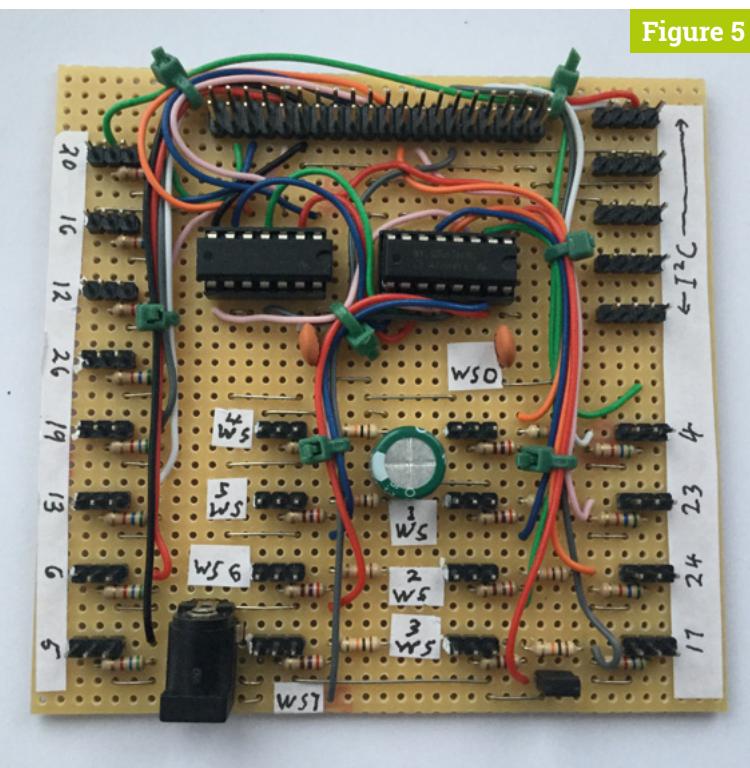
Cable ties are essential to prevent the wires resting in between the header pins and making them look untidy

09 The software

The software is written under the Pygame framework, mainly for the ease of sound generation, and is shown in the `traxscript.py` listing. The window itself has only two clickable words: one to load in a new script, and the other to reload it. The script is simply a text file (see Figure 6 for an example) containing instructions of what to do and when. Each condition should be on a new line, and there are only two conditions: ‘when’ and ‘now’. The ‘when’ condition syntax is: when (pin Number), pin state, delay value, play thread number or sound. The ‘now’ command simply plays a thread or sound immediately.

Figure 4

▲ Figure 4 Here's the full wiring of the interface board. Study it carefully



▲ Figure 5 The wires grouped and connections labelled in the finished product

10 How it works

The software reads in the script and builds up a list of pins to watch and what to do when they are triggered. A triggering event is a change in the pin reading; this can be the pin going high (rises), going low (falls), or just changing state (changes). Note that a pin may appear as many times in the list, triggering different things on different, or the same, events. By building a list of the pins we want to monitor, we don't waste time checking all the possible input pins.

11 Script errors

The beauty of using a script is that it documents and saves a specific layout. It does have its drawbacks, however. If a script contains errors, then the program is likely to crash. The most likely errors are that you specified a pin number that was not one of the twelve GPIO numbers the board uses, or you specified an FL3731 that used a device address not connected to the board. For an incorrect pin number, the one in error is printed out first. Adding a more rigorous file syntax-checking system could have been done, but would have made the code much longer.

You now have a system that can greatly enhance a GraviTrax layout. Its simple scripting function can be a great introduction to using scripts for children. We do have some ideas for adding other things to a GraviTrax system. For now, though, feel the tug of gravity with added sound and lights. **M**

Figure 6

► Language: Text

```
001. #, GraviTrax Script example
002. now, sound 0
003. now, ws 7
004. when 24, rises, delay 0.2, sound 4
005. when 24, falls, delay 1.5, sound 1
006. when 4, falls, delay 0.0, fl 6
007. when 4, rises, delay 0.0, ws 4
008. when 23, falls, delay 0.0, ws 7
009. now, fl 4
```

traxscript.py

► Language: **Python**

```

001. #!/usr/bin/env python
002. # TraxScript
003. # By Mike Cook November 2019
004.
005. import Neo_Thread as ws
006. import FL3731_Thread as fl
007. import RPi.GPIO as io
008. from copy import deepcopy
009. from tkinter import filedialog
010. from tkinter import *
011. import pygame
012. import time
013. import sys
014. import os
015. root = Tk()
016.
017. pygame.init()
018. os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
019. pygame.display.set_caption(
    "GraviTrax Script player")
020. screen = pygame.display.set_mode([320,40], 0, 32)
021. pygame.mixer.quit()
022. pygame.mixer.init(frequency=22050, size = -16,
023.                     channels = 2, buffer = 512)
024. pygame.event.set_allowed(None)
025. pygame.event.set_allowed([
    pygame.KEYDOWN, pygame.QUIT,
026.                         pygame.MOUSEBUTTONDOWN,
027.                         pygame.MOUSEBUTTONUP]
028.                     )
029. textHeight=24 ; black = (0, 0, 0)
030. font = pygame.font.Font(None, textHeight)
031. backCol = (120, 120, 120) ; lineCol = (196, 196, 0)
032.
033. ws.initIO()
034. fl.initI2C()
035.
036. def main():
037.     initIO()
038.     loadResources()
039.     drawScreen()
040.     getFile()
041.     loadFile(scriptName)
042.     setupScript()
043.     while 1 :
044.         traxRun()
045.

046. def initIO():
047.     global inPins, restartRec, loadRect,
048.         stopScript
049.     inPins = [17, 24, 23, 4, 5, 6, 13, 19, 26, 12,
050.               16,
051.               20]
052.     io.setmode(io.BCM); io.setwarnings(False)
053.     io.setup(
054.         inPins, io.IN, pull_up_down = io.PUD_UP)
055.     restartRec = pygame.Rect((238,7),(66,23))
056.     loadRect = pygame.Rect((10,7),(97,23))
057.     stopScript = 0
058.     soundFX, soundNames
059.     soundNames = ["owl", "Breaking Glass",
060.                   "ComputerBeeps1", "CymbalCrash",
061.                   "Fairydust", "Dog1", "Zoop", "Ya",
062.                   "Pop", "Duck", "Gong", "Laser1",
063.                   "Laser2", "running", "Screech",
064.                   "SpaceRipple", "Zoop", "Dog2",
065.                   "DirtyWhir", "ComputerBeeps2",
066.                   "AlienCreak2", "AlienCreak1"]
067.     soundFX = [pygame.mixer.Sound("sounds/" +
068.                               soundNames[effect]
069.                               + ".wav")
070.               for effect in
071.                   range(
072.                       0,
073.                       len(soundNames))
074.     ]
075.     def getFile():
076.         global scriptName, root
077.         success = False
078.         while not success:
079.             root.withdraw()
080.             scriptName = filedialog.askopenfilename(
081.                 initialdir = "/home/pi",
082.                 title = "Select GraviTrax script",
083.                 filetypes = (("txt files", "*.txt"),
084.                             ("all files", "*.*")))
085.             if ".txt" in scriptName :
086.                 success = True
087.             else :
```

**DOWNLOAD
THE FULL CODE:**

 magpi.cc/pibakery

```

087.         print("not a valid text file")
088.
089. def loadFile(fileName):
090.     global thingsToDo
091.     nameF = open(fileName, "r")
092.     thingsToDo = nameF.readlines()
093.     nameF.close()
094.
095. def setupScript():
096.     global pinToWatch, changeToMonitor,
097.         soundToPlay
098.     global actionDelay, wsToPlay, flToPlay,
099.         pendTime
100.    global nowIn, lastIn, trigNum, monitor,
101.        pendPlay
102.    pinToWatch = []
103.    changeToMonitor = []
104.    soundToPlay = []
105.    actionDelay = []
106.    wsToPlay = [] ; flToPlay = []
107.    print("\nloading the script", len(thingsToDo),
108.          "lines")
109.    for move in range(0,len(thingsToDo)):
110.        line = str(thingsToDo[move])
111.        now = False
112.        for val in line.split(","):
113.            if "when" in val:
114.                pin = int(val[-2:])
115.                if not(pin in inPins) :
116.                    print("Pin",pin,"is not valid")
117.                pinToWatch.append(pin)
118.            elif "now" in val:
119.                now = True
120.            elif "falls" in val and not now:
121.                changeToMonitor.append(1)
122.            elif "rises" in val and not now:
123.                changeToMonitor.append(2)
124.            elif "changes" in val and not now:
125.                changeToMonitor.append(3)
126.            elif "delay" in val and not now:
127.                actionDelay.append(float(val[-3:]))
128.            elif "sound" in val or "ws" in val or
129.                "fl" in val:
130.                pram = int(val[-2:])
131.                if "sound" in val:
132.                    if not now:
133.                        soundToPlay.append(pram)
134.                wsToPlay.append(-1)
135.                flToPlay.append(-1)
136.            else:
137.                soundFX[pram].play()
138.            elif "ws" in val:
139.                if not now:
140.                    soundToPlay.append(-1)
141.                    wsToPlay.append(pram)
142.                    flToPlay.append(-1)
143.            else:
144.                ws.startWs2812Thread(pram)
145.            elif "fl" in val:
146.                if not now:
147.                    soundToPlay.append(-1)
148.                    wsToPlay.append(-1)
149.                    flToPlay.append(pram)
150.                    ...
151.                    print("pin", pinToWatch)
152.                    print("change", changeToMonitor)
153.                    print("delay", actionDelay)
154.                    print("sound", soundToPlay)
155.                    print("ws animation", wsToPlay)
156.                    print("fl animation", flToPlay)
157.                    ...
158.                    monitor = len(pinToWatch)
159.                    nowIn = [0] * monitor
160.                    lastIn = [0] * monitor
161.                    pendPlay = [0] * monitor
162.                    pendTime = [0.0] * monitor
163.                    for i in range(0, monitor):
164.                        lastIn[i] = io.input(pinToWatch[i])
165.                    print("Current pin states",lastIn)
166.    def traxRun():
167.        global nowIn, lastIn, stopScript
168.        while stopScript == 0 :
169.            checkForEvent()
170.            for i in range(0, monitor):
171.                nowIn[i] = io.input(pinToWatch[i])
172.                if lastIn[i] != nowIn[i] :
173.                    #print("Current pin states", lastIn)
174.                    lastIn[i] = nowIn[i]
175.                    tmatch = changeToMonitor[i]-1 #
match

```

traxscript.py

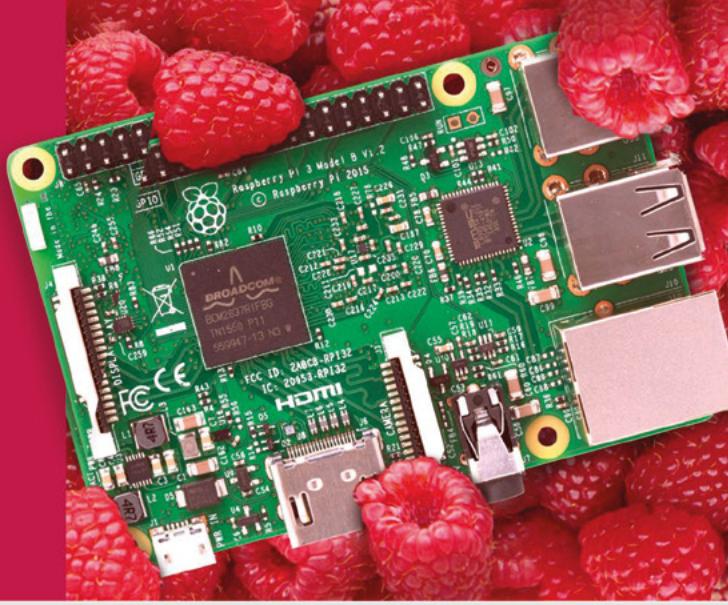
► Language: Python

```

176.             if tmatch == 2:
177.                 tmatch = nowIn[i]
178.                 if changeToMonitor[i] != 0 and
179.                     nowIn[i] == tmatch:
180.                         if soundToPlay[i] != -1 :
181.                             pendPlay[i] =
182.                                 soundToPlay[i]
183.                                 if wsToPlay[i] != -1 :
184.                                     pendPlay[i] = wsToPlay[i]
185.                                     if flToPlay[i] != -1 :
186.                                         pendPlay[i] = flToPlay[i]
187.                                         pendTime[i] = time.time() +
188.                                             actionDelay[i]
189.                                             for i in range(0, monitor): # what to
190.                                                 play now
191.                                                 if pendTime[i] > 0.0 and time.time()
192.                                                     >= pendTime[i]:
193.                                                         if soundToPlay[i] != -1 :
194.                                                             soundFX[soundToPlay[i]].play()
195.                                                             if wsToPlay[i] != -1 :
196.                                                                 ws.startWs2812Thread(wsToPlay[i])
197.                                                                 if flToPlay[i] != -1 :
198.                                                                     if flToPlay[i] == 99 : # stop
199.                                                                         clock
200.                                                                         f1.stopCount()
201.                                                                         else:
202.                                                                             f1.startFL3731Thread
203.                                                                             (flToPlay[i])
204.                                                                             pendTime[i] = 0.0
205.                                                                             if stopScript == 1 :
206.                                                                                 getFile()
207.                                                                                 loadFile(scriptName)
208.                                                                                 setupScript()
209.                                                                                 stopScript = 0
210.                                                                                 drawWords("Load script", 14, 8, black,
211.                                         backCol)
212.                                                                                 drawWords("Restart", 244, 8, black, backCol)
213.                                                                                 pygame.display.update()
214.                                         backCol)
215.                                         textRect = textSurface.get_rect()
216.                                         textRect.left = x # right for align right
217.                                         textRect.top = y
218.                                         screen.blit(textSurface, textRect)
219.                                         return textRect
220.
221.                                         def handleMouseDown(pos): # look at mouse down
222.                                             if loadRect.collidepoint(pos) :
223.                                                 pygame.draw.rect(screen, lineCol, loadRect, 0)
224.                                             elif restartRec.collidepoint(pos) :
225.                                                 pygame.draw.rect(screen, lineCol,
226.                                                     restartRec, 0)
227.                                                 pygame.display.update()
228.
229.                                         def handleMouseUp(pos): # look at mouse up
230.                                             global stopScript
231.                                             if loadRect.collidepoint(pos) :
232.                                                 stopScript = 1
233.                                             if restartRec.collidepoint(pos) :
234.                                                 stopScript = 2
235.                                                 drawScreen()
236.
237.                                         def constrain(val, min_val, max_val):
238.                                             return min(max_val, max(min_val, val))
239.
240.                                         def terminate(): # close down the program
241.                                             global root
242.                                             root.destroy()
243.                                             pygame.mixer.quit()
244.                                             pygame.quit() # close pygame
245.                                             os._exit(1)
246.
247.                                         def checkForEvent(): # see if we need to quit
248.                                             event = pygame.event.poll()
249.                                             if event.type == pygame.QUIT :
250.                                                 terminate()
251.                                             if event.type == pygame.KEYDOWN :
252.                                                 if event.key == pygame.K_ESCAPE :
253.                                                     terminate()
254.                                                 if event.type == pygame.MOUSEBUTTONDOWN :
255.                                                     handleMouseUp(pygame.mouse.get_pos())
256.                                                 if event.type == pygame.MOUSEBUTTONDOWN :
257.                                                     handleMouseDown(pygame.mouse.get_pos())
258.
259.                                         if __name__ == '__main__':
260.                                             main()

```

American Raspberry Pi Shop



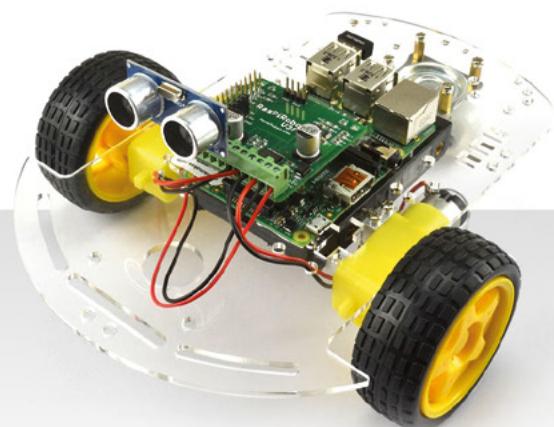
- Displays
- Cases
- Project Kits
- Add-on Boards
- HATs
- Arcade
- Cameras
- Cables and Connectors
- Sensors
- Swag
- Power Options
- GPIO and Prototyping

Partner and official reseller for top Pi brands:



and many others!

Price, service, design,
and logistics support for
VOLUME PROJECTS



Part 09

Custom widgets with C and GTK



Simon Long

MAKER

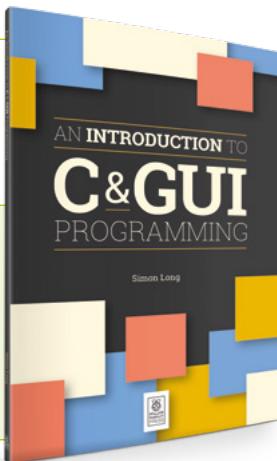
Simon Long is a software engineer working for Raspberry Pi, responsible for the Raspberry Pi Desktop on both Raspbian and Debian.

rpf.io

An Introduction to C & GUI Programming

For further tutorials on how to start coding in C and creating GUIs with GTK, take a look at our new book, *An Introduction to C & GUI Programming*. Its 156 pages are packed with all the information you need to get started – no previous experience of C or GTK is required!

magpi.cc/GUIbook



Change the properties of widgets to alter how they look

In all the examples we've seen so far, we have used widgets in their default state; we've just created the widget with the `gtk_<widget name>_new` function call and used it. However, GTK does allow a degree of customisation of widgets by setting the properties for each.

As an example, we are going to look at some of the properties of the basic GtkButton widget. Try this example:

```
void main (int argc, char *argv[])
{
    gtk_init (&argc, &argv);

    GtkWidget *win = gtk_window_new (
        GTK_WINDOW_TOPLEVEL);
    GtkWidget *btn =
        gtk_button_new_with_label (
            "Close window");
    g_signal_connect (btn, "clicked",
        G_CALLBACK (end_program), NULL);
    g_signal_connect (win, "delete_event",
        G_CALLBACK (end_program), NULL);

    GtkWidget *btn2 =
```

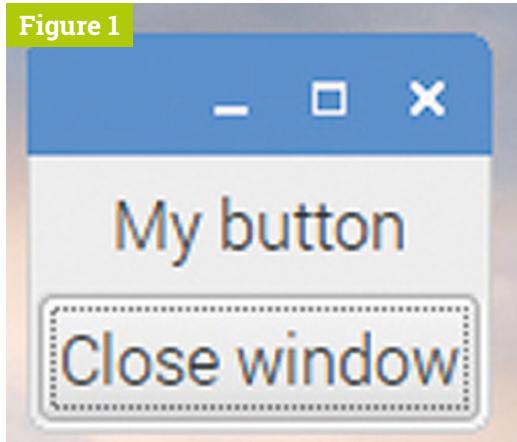
```
gtk_button_new_with_label (
    "My button");
g_object_set (G_OBJECT (btn2), "relief",
    GTK_RELIEF_NONE, NULL);

GtkWidget *box = gtk_vbox_new (FALSE, 5);
gtk_box_pack_start (GTK_BOX (box), btn2,
    TRUE, TRUE, 0);
gtk_box_pack_start (GTK_BOX (box), btn,
    TRUE, TRUE, 0);
gtk_container_add (GTK_CONTAINER (win),
    box);
gtk_widget_show_all (win);
gtk_main ();
```

This is familiar code from previous examples, but the highlighted line is new.

`g_object_set` takes as arguments the name of a widget, followed by a `NULL`-terminated list of property names and property values. In this case, we are setting the `relief` property of the `GtkButton` `btn2` to `GTK_RELIEF_NONE` (see **Figure 1**).

Figure 1



▲ **Figure 1** A `GtkButton` with the `relief` property set to `GTK_RELIEF_NONE`

The ‘relief’ of a GtkButton controls how the border looks. The borders of some GTK widgets have a degree of shading applied around them to provide a 3D appearance – by default, a GtkButton has this shading applied, which makes the button appear to stand out slightly from the window background. By setting the relief to `GTK_RELIEF_NONE`, this 3D shading is removed – if you run the program above, you should be able to clearly see the difference between the two buttons on the window. (You can use the `TAB` key to move the dotted outline between the buttons to show the difference more clearly.)

Here’s another example. Remove the setting of the relief property, and change the name of the button by adding an underscore:

```
GtkWidget *btn2 =
    gtk_button_new_with_label (
        "My_button");
```

You should end up with a button that looks like that in **Figure 2**.

If you now set the `use-underline` property:

```
g_object_set (G_OBJECT (btn2),
    "use-underline", TRUE, NULL);
```

...the underscore will vanish, but will reappear

“The borders of some GTK widgets have shading applied around them to provide a 3D appearance”

under the ‘b’ of ‘button’ if you hold down the `ALT` key on the keyboard (**Figure 3**, overleaf).

All widgets have properties that can be set like this. As another example, try replacing the GtkButton with a GtkLabel:

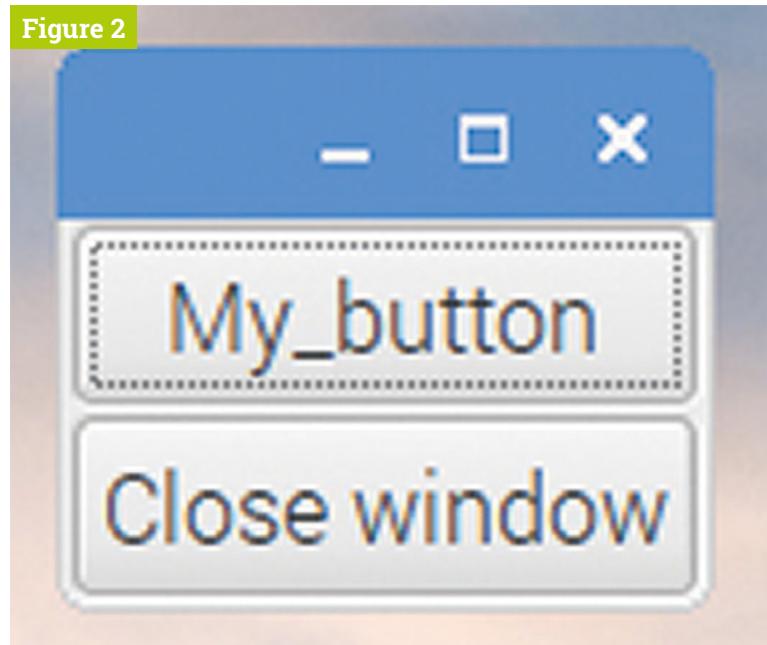
```
GtkWidget *lbl = gtk_label_new (
    "My label");
```

...and then setting the angle property of the label to 45 degrees:

```
g_object_set (G_OBJECT (lbl), "angle",
    45.0, NULL);
```

(Note that it is important to enter the angle as

Figure 2



45.0, rather than just as 45 – the value expected is a floating-point number, and adding the .0 to the end of the value ensures that the compiler treats it as such.)

You should end up with a window which looks like this, with the label text at a 45 degree angle to the horizontal (**Figure 4**, overleaf).

In many cases, widgets also have dedicated functions to set each property which can be used instead of the generic `g_object_set` function (in the examples above, `gtk_button_set_relief`, `gtk_button_set_use_underscore`, and `gtk_label_set_angle`, respectively). The advantage of `g_object_set` is that it can be used to set multiple properties in one line, which can shorten your code significantly.

The GTK online documentation page for each widget lists all the properties and dedicated functions to set their values. For the two examples above, this can be found at magpi.cc/GtkButton and magpi.cc/GtkLabel – it’s worth having a look through the options for any widget you want to use. (These pages are also a good way of finding out what signals are generated by a widget when a user interacts with it.)

An introduction to themes

The other way that GTK widgets can be customised is by the use of a theme. A theme affects the appearance of every instance of a widget in every GTK application, rather than changing the

▲ **Figure 2** A GtkButton with an underscore in the label and the `use-underline` property set to FALSE

appearance of individual widgets one at a time. There is a selection of themes installed in Raspbian (and in most other Linux desktop distributions), in the directory **/usr/share/themes**.

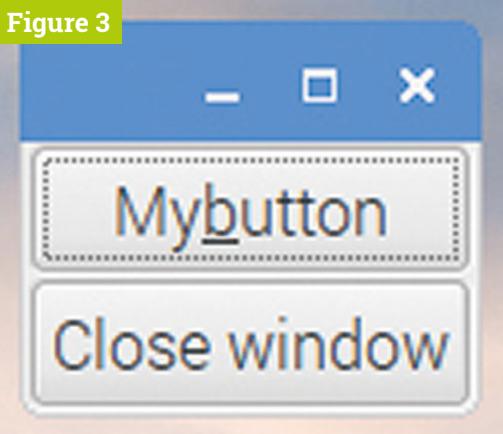
This directory contains a number of named folders, each of which is a theme for either GTK or other themeable applications. If a named folder contains a subfolder named **gtk-2.0**, then the name of that folder is also a valid GTK 2 theme name.

Which of the themes is currently used by GTK applications is usually controlled by the **xsettings daemon**, a process which runs in the background and provides configuration information to all desktop applications. On Raspbian, to change which theme is set in the daemon, you need to change a value in a configuration file.

“ Which theme is currently used by GTK applications is usually controlled by the **xsettings daemon** ”

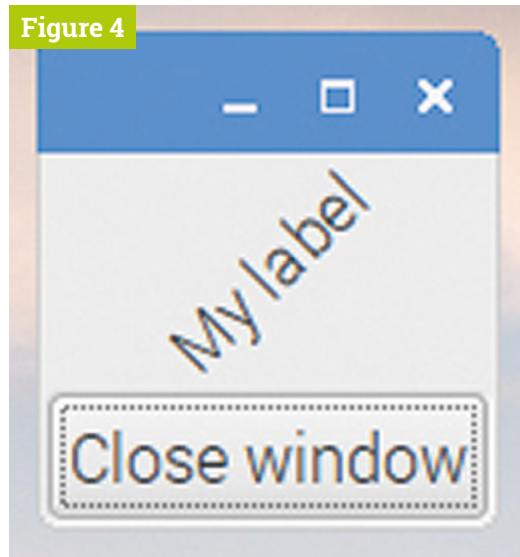
To do this, check to see if there is a file called **desktop.conf** in the directory **~/.config/lxsession/LXDE-pi**. If there isn't, create one by copying the file **/etc/xdg/lxsession/LXDE-pi/desktop.conf** into that directory.

If you then look in the **desktop.conf** file with a text editor, there is a section headed **[GTK]**. Somewhere under this heading is a line starting **sNet/ThemeName=**, which by default on Raspbian is set to **Pix**. If you change **Pix** in this line to the



► Figure 3 The same GtkButton, but with **use-underline** set to TRUE

Figure 4



▲ Figure 4 A GtkLabel with the **angle** property set to 45.0

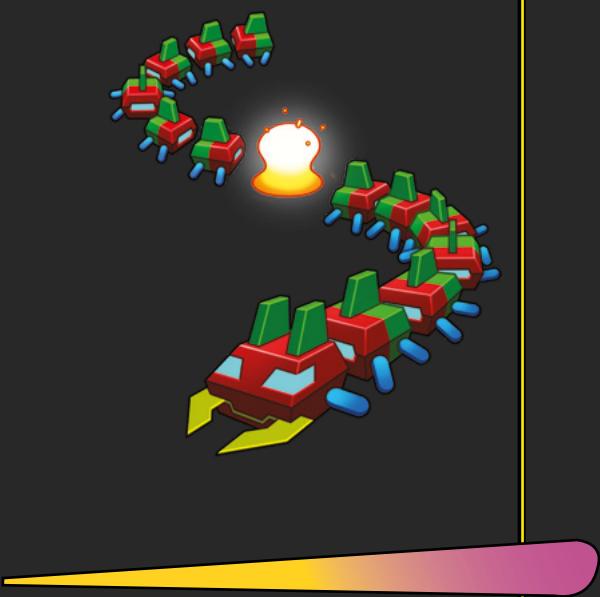
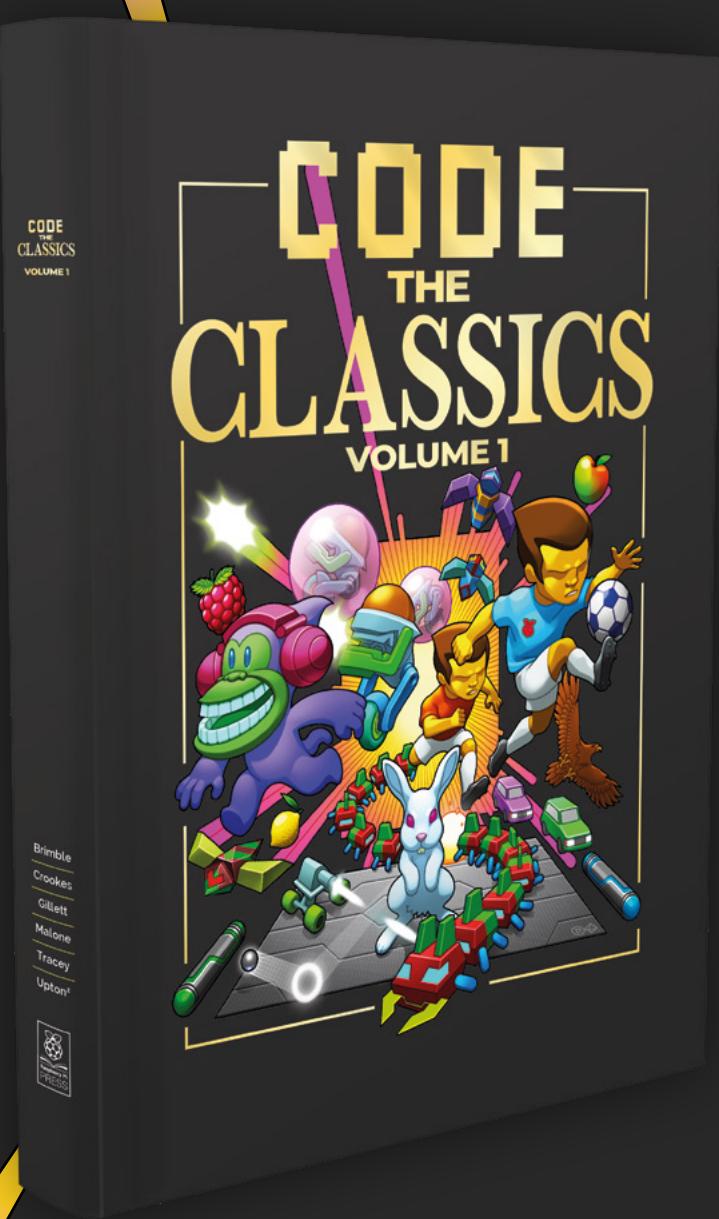
name of another GTK +2 theme (any directory in **/usr/share/themes** which includes a **gtk-2.0** subdirectory), the theme in use will automatically update and you should see every GTK application running redraw with the new theme.

Creating a theme is not for the faint-hearted, but if you are interested, look inside one of the **gtk-2.0** subfolders in the directory **/usr/share/themes**.

The theme itself is in a file in this folder named **gtkrc**; there may also be a number of other subfolders containing graphical elements and other resources used by the theme.

The **gtkrc** file is a plain text document which can be opened in your editor of choice. Most of them contain a number of style definitions, which consist of the word **style** followed by a number of lines contained within curly brackets. At the end of the file are lines which associate widgets with the styles previously defined.

To change the appearance of a widget, find it in the list of associations at the end of the file, and note which style applies to it. Then locate the definition of that style within the file and try changing some of the parameters to see what happens. Do back up the original **gtkrc** file before you do this – or even better, copy the whole theme folder and give it a new name of your choice; set the **ThemeName** in **desktop.conf** to the name of your new theme, and create your own theme by modifying an existing one! ■



- *Get game design tips and tricks from the masters*
- *Explore the code listings and find out how they work*
- *Download and play game examples by Eben Upton*
- *Learn how to code your own games with Pygame Zero*

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.

Available now: magpi.cc/store

Design a 3D printer pendant with BlocksCAD



MAKER

Dr Tracy Gardner

Tracy is a computer scientist who spent ten years working as a software architect at IBM. She now writes educational projects for the Raspberry Pi Foundation.

rpf.io/projects

Use BlocksCAD to design a 3D pendant based on the Flower of Life pattern

BlocksCAD is a 3D model editor that you use in a web browser, and it runs on Raspberry Pi. You drag and drop code blocks to design 3D models that can be exported for 3D printing.

In this project, you will use BlocksCAD to design a 3D pendant. The pendant uses a geometric design based on the ‘Flower of Life’, a design which is often found in historical art.

If you have access to a 3D printer, then you can print your pendant. The pendant is small and only uses a little bit of filament, and it is quick to 3D-print. There’s a hoop on top of the pendant so that you can put it on a necklace or cord. The pendant has a diameter of 40 mm, plus the hoop for hanging. It is 2 mm thick, so it will 3D-print quite quickly.

After this project, you’ll also be able to code your own design and create a custom pendant.

You'll Need

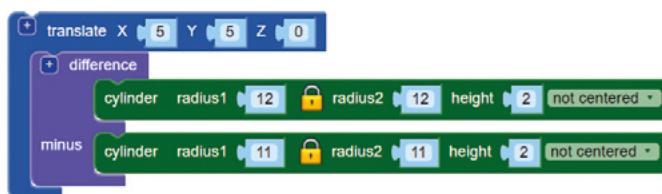
- ▶ Raspberry Pi 4
- ▶ BlocksCAD 3D blockscad3d.com
- ▶ 3D printer (magpi.cc/69)

01 Create a hoop

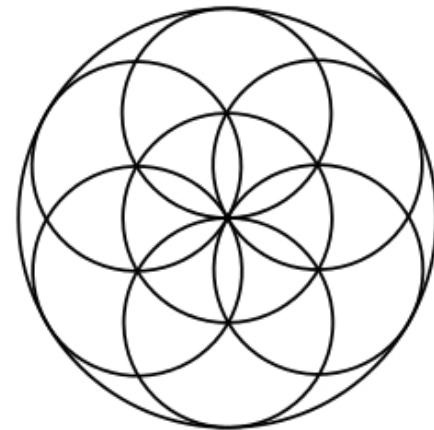
This project can be completed in a web browser using BlocksCAD (blockscad3d.com). Open Chromium and enter the BlocksCAD editor URL: blockscad3d.com/editor.

The design uses six interlocking hoops in the centre, and a larger hoop around the outside. As mentioned, the pendant is 40 mm wide, plus the hoop for hanging, and is 2 mm thick.

Click 3D Shapes and drag a **cylinder** block to the project. Create a cylinder with a radius of 12,



▶ Use a translate block to move the first hoop off centre



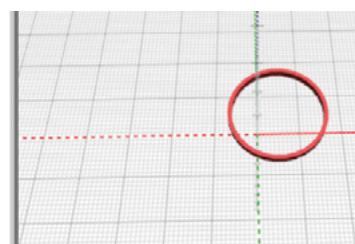
▲ The design is influenced by the Flower of Life pattern. If you like, you could research the Flower of Life and see if you can create a design based on the Seed of Life pattern

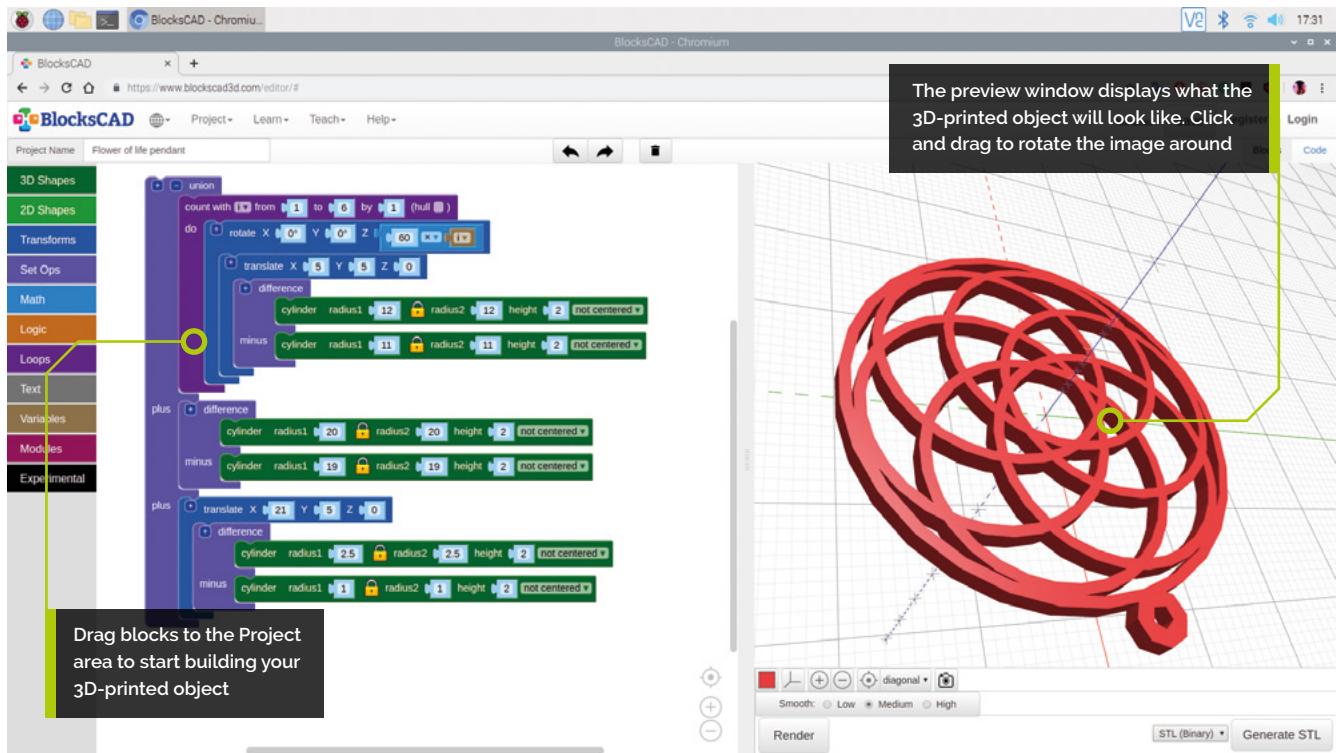
and a height of 2 (the unit here is millimetres). Cylinders are automatically centred along the X and Y axes. Select ‘not centered’ so that the pendant sits on the surface. (This means that the Z axis value is larger than 0.)

Click on the Render button after each change to your code to see the results.

02 Add more hoops

Now, drag a **difference** block from Set Ops to encase the cylinder. Add another **cylinder** block in the bottom space, and this time give it a radius of 11.





This will remove a smaller cylinder from the centre. This creates a hoop. Click Render again to see it.

If you like, you can click on the coloured square to change the colour used in the viewer. This does not affect the colour of your pendant, as that depends on the colour of the filament that you use.

The design uses six intersecting hoops, and each hoop is moved out from the centre and rotated a different number of degrees.

In the final design, there is no central hoop: the hoops are all moved out from the centre.

Drag a **translate** block (from **Transforms**) around your code, and set x and y to 5. This moves the first hoop into position.

03 Centre the hoop

Now the hoop is a little off-centre. You need multiple copies of this hoop, rotated around the centre. First, create three equally spaced hoops.

Add a **count Loops** block to create three hoops. To space the hoops, add a **rotate** **Transforms** block between the **count** loop and the **translate** block.

In the **count** block, set the i variable from 1 to 3. You'll need to insert an arithmetic block from **Math** and a variable (**i**) block from **Variables** into the Z field of the **rotate** block.

The rotation moves each hoop by $120 \times i$ degrees, so that the three hoops are distributed equally around the 360 degrees of a circle ($360 / 3 = 120$).

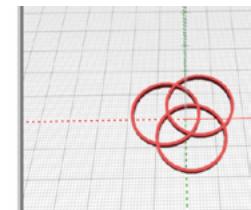
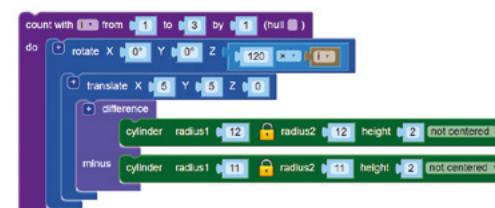
4 The design uses six intersecting hoops, each moved out from the centre and rotated **n**

Look at the code and make sure you understand how it works. The finished design has six hoops rather than three. In the **count** block, set i from 1 to 6, and set the Z rotation to 60, so it creates six equally spaced hoops.

04 Add a border

Next, add a border around the edge of the design. Create a centred hoop that touches the edges of the design. You can either do the maths to work out what the radius of the circle needs to be, or you can just create a circle and change the radius until it works. Either approach is fine!

▼ By using a count loop and a rotate block, you can quickly create multiple hoops





► You can 3D-print the pendant in any colour filament you like

Top Tip



Bigger hoops

The code adds a small hoop that can be used with elastic necklace thread. If you have a bigger cord or chain, you can adapt the code to create a bigger hanging hoop.

Encase your code with a **union** block from Set Ops, to join the border to the other hoops.

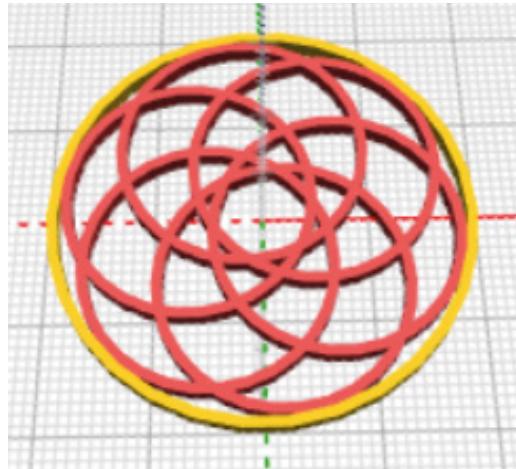
Add a **difference** block to the plus section of **union**, and two **cylinder** blocks to make the hoop.

The six hoops each have a radius of 12 mm, so the border cylinder that you are making needs to be bigger than that. You could try setting the radius to 24 mm.

To make a hoop, the radius of the second cylinder in the **difference** block needs to be 1mm smaller than the radius of the first cylinder.

Adjust the size of the cylinders until the border hoop just touches the outer edges of the six inner hoops.

The radius should be around 20. (As mentioned in the introduction, the finished pendant will be 40 mm in diameter.)



► The outer hoop should just touch the edges of the inner ones

05 Work it out

You could also use maths to work out the diameter. The diameter of each inner hoop is 24 mm. If the hoops met at the centre of the pendant, the border hoop would need to have a radius of 24 mm. But the inner hoops overlap, as they are translated 5 mm along the X and Y axes.

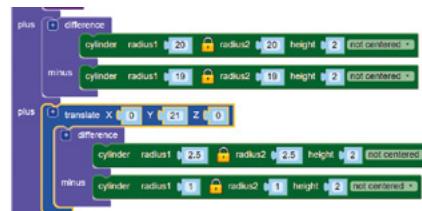
This removes a section from the radius. This section is on the arc, 5 mm from the origin, so we need to remove 5 mm from 24 mm. Thus the inner radius of the border hoop should be 19 mm.

Maths is really useful when you need to be accurate. But it's fine to just change things until you get the result you need.

06 Add a hanging hoop

Now, add a small hanging hoop through which you can thread a cord to make a necklace. Click the [+] on the **union** block to add another section to add the new hoop.

At the moment, the position of the hanging hoop isn't very visually pleasing.



► Adding blocks to create a small hanging hoop for the pendant's cord



Add a **rotate** block to move the inner hoops so that the hanging hoop is centred over one of the gaps between them.

07 Experiment with shapes

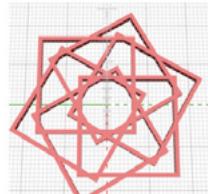
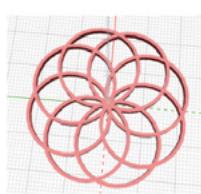
Experiment and change some values in your pendant. For example, change the number of hoops, or the rotation. You could also try to use cuboids (cubes) instead of cylinders to create a pattern.

08 Export to STL

BlocksCAD 3D can export an STL file for 3D printing. Render your model and then click on Generate STL. Remember where you save the STL file. Now 3D-print your pendant using a filament of the colour of your choice. Very carefully remove the 3D print from the print bed. The pendant is thin, so it's quite delicate.

You might need to remove small strands of filament (especially from the hanging hoop) to tidy up the print.

Thread the pendant on a chain or cord. If you want to use a thicker cord or necklace, then you can adjust the design to have a larger hanging hoop. ■



▲ Try adjusting the design and using different shapes

“Very carefully remove the 3D print from the print bed. The pendant is thin, so it's quite delicate”

Top Tip

Issue 69

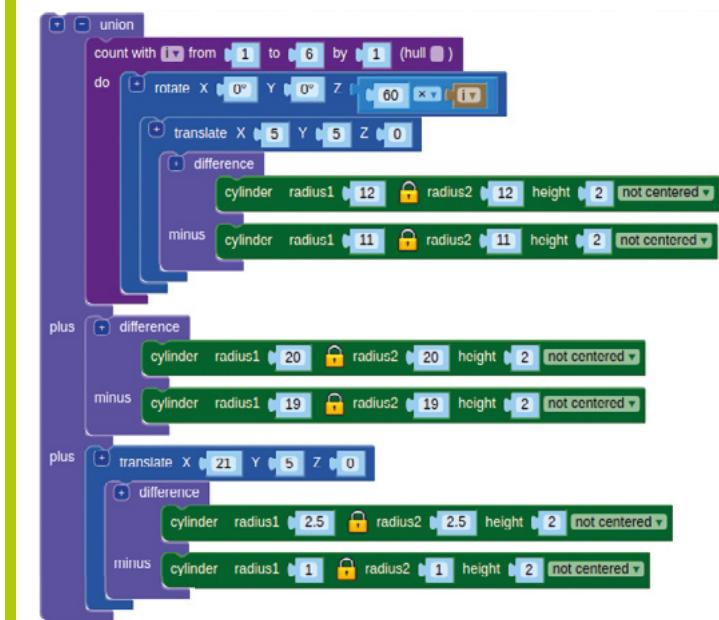
Learn all about 3D printing and setting up a low-cost 3D printer in issue 69 of *The MagPi* magazine (magpi.cc/69).

DOWNLOAD THE FULL CODE:

 magpi.cc/blockscad-pendant

Flower of life pendant

► Language: **BlocksCAD**



Raspberry Pi Kit & Gadget Guide

Do more with your Raspberry Pi with these amazing add-ons! Compiled by **Wes Archer**

We all know that Santa loves a mince pie, but did you know that he is often partial to leaving Raspberry Pi boards under the tree of those on his ‘nice’ list? Well, if you got a Raspberry Pi for Christmas, then you may want to know about some of the awesome accessories you can get for it to really get the most out of your brand new computer. With so many cool accessories available, it can be a minefield knowing which ones to go for, so let us help you make up your own mind based on some ideas of ours.



69 Cases

70 Essential add-ons

71 Cool HATs

72 Electronic starter kits

72 Gaming kits

73 Gaming accessories

74 Robot building kits



Raspberry Pi cases

We think Raspberry Pi looks cool as it is, but a case is a highly recommended accessory. Not only will it protect your Raspberry Pi and the delicate circuitry, it can also enhance the way some of the additional accessories work when used in combination.

Official Raspberry Pi 4 Case

If you want an affordable, reliable, and hackable case, then you cannot go wrong with the official Raspberry Pi case. Available in red and white or black and grey, this case will house your Raspberry Pi with ease.

If you're feeling adventurous, you can also hack the case to fit a small fan for cooling!

£5 | magpi.cc/case

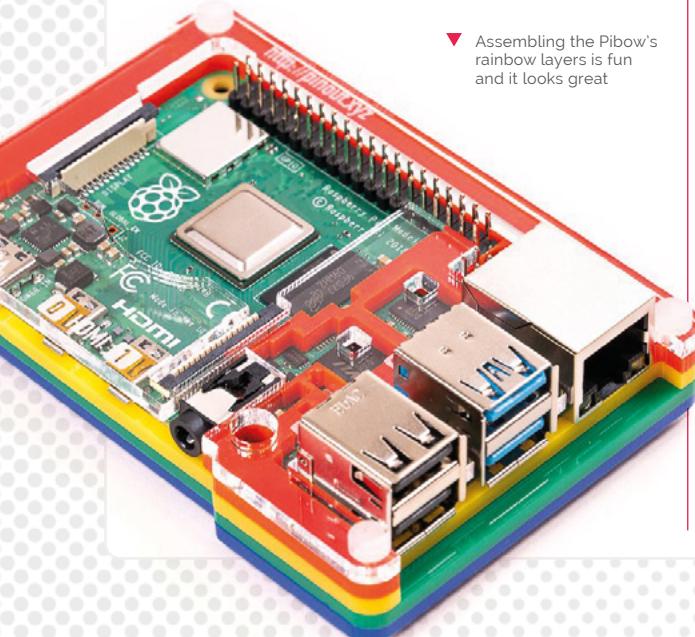


Pibow Coupé 4

The Pibow by Raspberry Pi veterans Pimoroni is a classic Raspberry Pi case. Designed to be quick, easy, and cheap, the Pibow is made up of multiple layers of laser-cut acrylic. The Coupé version is slimmed down and gives easy access to Raspberry Pi's GPIO and other inputs.

£8 | magpi.cc/pibow

▼ Assembling the Pibow's rainbow layers is fun and it looks great



SecurePi Case

The SecurePi case looks very futuristic, especially with those angles! This case provides protective covers for your microSD card, USB, Ethernet and micro HDMI ports, and also has venting which is ideal for providing airflow for keeping your Raspberry Pi cool. It has space for the PoE HAT or Fan SHIM too!

£10 | magpi.cc/securepi

◀ Protect your ports and your Raspberry Pi with the SecurePi case

Our biggest fan

Raspberry Pi 4 is the most powerful Raspberry Pi yet. All this horsepower means it can get a bit hot, though. The most effective cooling method is active cooling, which is typically accomplished with a fan. The Fan SHIM is perfect as it is low-profile, inexpensive and allows you to use the GPIO pins for other accessories.



Aluminium Raspberry Pi 4 Case

Aluminium is a great, lightweight metal that is also strong and is an ideal choice for a Raspberry Pi case because of these properties.

This case looks great, especially if used as part of a 4K home media setup. With the holes, the cooling potential is also fantastic.

£10 | magpi.cc/aluminium



Anidees Raspberry Pi 4 Case

Made of aluminium, the Anidees case provides ample protection for your precious Raspberry Pi. It comes in two colours – silver or black – and has an extra tall version to accommodate some HATs too. Oh, and it has a clear lid so you can see your Raspberry Pi in all its glory!

£37 | magpi.cc/anidees

▼ We love the clear acrylic lid that allows you to see inside



Essential add-ons

Some accessories are essential for any Raspberry Pi fans. These accessories will be useful in pretty much every project, so take a look below



▲ The official keyboard doubles as a USB hub with its three extra ports

Raspberry Pi Keyboard

You'll need a keyboard in pretty much every Raspberry Pi project going. The Raspberry Pi Keyboard comes in a variety of layout options, and is available in two colour schemes. Not only does the keyboard connect via USB, it also has three additional USB 2.0 ports to free up ports on your Raspberry Pi.

£16 | magpi.cc/keyboard

■ The Raspberry Pi Keyboard comes in a variety of layout options, and two colour schemes ■

Raspberry Pi Mouse

Something simple, yet extremely effective and an essential accessory for any Raspberry Pi project, allowing you to navigate through any graphical user interface. The Raspberry Pi Mouse, when combined with the Raspberry Pi Keyboard, can be powered from the keyboard's USB hub, keeping those precious ports free on your Raspberry Pi itself.

£7 | magpi.cc/mouse



Rii i8+ Mini Wireless Keyboard

If you want to go one step further, why not combine the keyboard and mouse into one and make it wireless while you're at it? With the Rii i8 Mini Wireless Keyboard with Touchpad, you can do just that! The supplied USB wireless dongle plugs into your Raspberry Pi and connects automatically.

£18 | magpi.cc/wirelesskeys



Retro Cube Bluetooth Speaker

Why use a USB port or cable when you can use Bluetooth to keep things wireless? This little speaker, by retro gamepad specialists 8bitdo, is a fantastic little Bluetooth speaker. Styled like a retro console controller, this rechargeable speaker provides up to eight hours play after one hour of charging.

£18 | magpi.cc/speaker



4 Port USB Hub

Whilst Raspberry Pi has four USB ports, they can be used up quickly depending on how you are using it. Having a dedicated USB hub is always handy, particularly if you have a Raspberry Pi Zero. This four-port hub has both USB and micro USB connectors, so works on any Raspberry Pi!

£8 | magpi.cc/usbhub



We recommend...

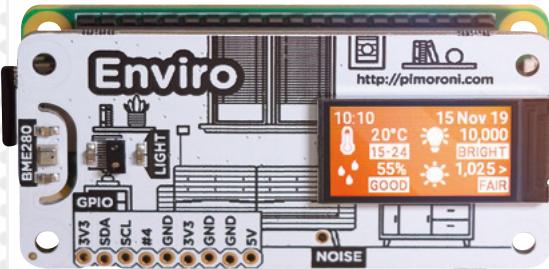
USB microSD card adapter

If you're regularly writing microSD card images for your Raspberry Pi, a USB microSD card adapter is a great tool to have, especially if your computer doesn't have an SD card slot!

magpi.cc/usbsd

Cool HATs

There are hundreds of HATs available for Raspberry Pi. As they are so easy to connect and set up, they are a perfect accessory



Enviro

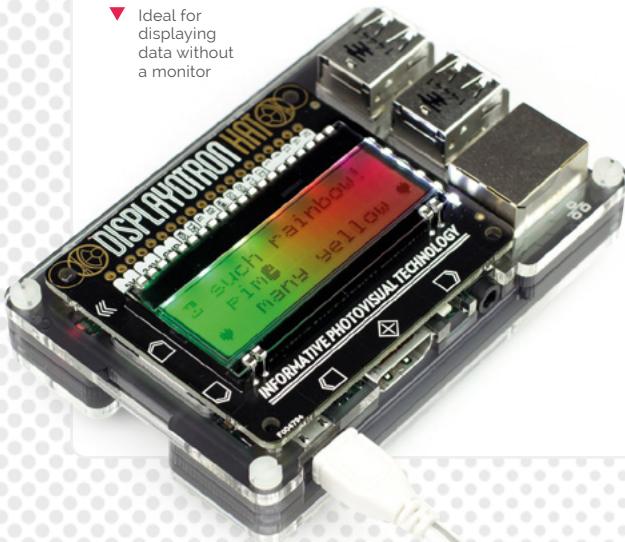
The Enviro is a fantastic piece of kit. It allows you to monitor a number of environmental factors, such as temperature, light, and sound. The fully-featured Enviro + Air Quality version also includes a gas sensor. Simply connect to your Raspberry Pi, install the code, and you'll have your very own monitoring station.

£28 | magpi.cc/enviro

Display-O-Tron HAT

The Display-O-Tron HAT is a fantastic little screen, backlit with controllable RGB LEDs, has six capacitive touch buttons, and also features a small LED bar graph! If you want to run your Raspberry Pi 'headless' (i.e. without a screen connected), then the Display-O-Tron HAT is ideal.

£23 | magpi.cc/displayotron



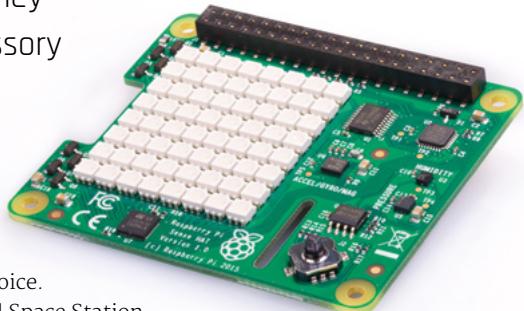
▼ Ideal for displaying data without a monitor

Sense HAT

If you want something a little more 'out of this world', then the Sense HAT is a perfect choice.

Used on the International Space Station as part of AstroPi, the Sense HAT monitors temperature, humidity, pressure, and orientation. It also has an 8x8 LED matrix on top for additional display purposes.

£30 | magpi.cc/sensehat



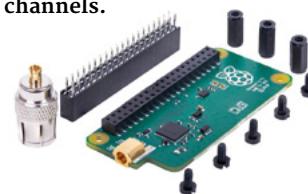
▲ As used on the ISS, the Sense HAT is packed with sensors and features

TV HAT

TV on a Raspberry Pi? Yes, that's right! With the TV HAT and a bit of configuration, you can set your Raspberry Pi to receive terrestrial television channels.

It is even possible to record TV shows so that you can watch them back at your leisure too!

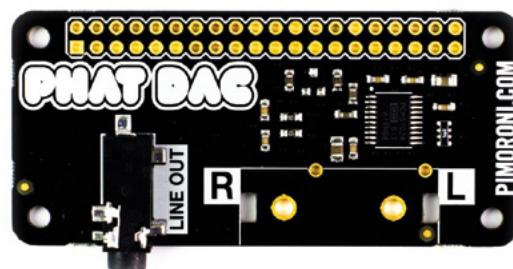
£20 | magpi.cc/tvhat



pHAT DAC

A DAC (digital-to-analogue converter) is a must for anybody who takes listening to music seriously. The quality of your music is much better when a DAC is used, and the pHAT DAC is a great little accessory that you can use to play music to your heart's content.

£13 | magpi.cc/phatdac



We recommend...

Resistor lead bending tool

Prototyping is essential, and this handy tool makes it easier to bend those resistors into breadboard-friendly form.

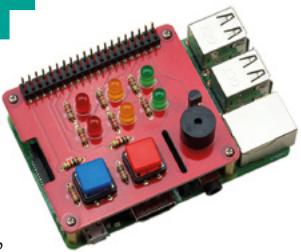
magpi.cc/resistorbend

Electronic starter kits

A Raspberry Pi can do more than play retro games or videos. Thanks to the GPIO pins, you can interact with a variety of sensors and devices

Jam HAT (LED & Buzzer Board)

If you're not that good at soldering and want something that is pre-assembled in a HAT form, then the Jam HAT is a great alternative. With LEDs, buttons and a buzzer, you can use the code examples provided to create your own unique projects, all for under a tenner!



£7 | magpi.cc/jamhat

CamJam EduKit

Prototyping is a great way to start experimenting with sensors, LEDs, buzzers, and everything else that can be connected to a Raspberry Pi. The CamJam EduKit contains a breadboard, an essential tool that allows you to make your own prototype circuits without soldering a thing, as well as other essential components.

£5 | magpi.cc/edukit



Gaming kits

Feeling adventurous? Have a go at building your own Raspberry Pi-powered gaming setup

TinyPi Pro

What good is a portable games console unless it fits in the smallest of pockets? Enter the TinyPi Pro – a do-it-yourself kit that is a small but perfectly formed games console. These sell like hot cakes, but are a real gem if you can get a hold of one, and you'll learn lots during the build.

£90 | magpi.cc/tinypipro



BASIC Monster Arcade Controller Kit

If the full Picade kit is a bit too lavish for you, then the Arcade Controller Kit by Monster is a great alternative. With this kit, you'll build an arcade stick that houses your Raspberry Pi, which can be connected to your TV for a more portable setup.

£60 | magpi.cc/monsterbasic



PiGRRL 2.0 kit

If you fancy 3D-printing your own case (designs are included) and putting your build skills to the test, then consider the PiGRRL 2.0 kit. You'll need to supply the Raspberry Pi and the case, but you'll have a great time putting it all together and testing it out when complete.

£56 | magpi.cc/pigrll2



Picade

When it comes to arcade kits, Pimoroni's Picade is king, and for very good reason! The kit is expertly crafted and has been refined since it was initially launched after a successful Kickstarter campaign. It comes in two options – with an 8-inch or 10-inch display – and with detailed step-by-step build instructions and videos.

£150 to £195 | magpi.cc/picade



Gaming accessories

Raspberry Pi is an excellent choice for emulating and playing retro games. But what accessories should you consider?



SN30 Pro+ Bluetooth Gamepad

There are so many controllers to choose from, but 8BitDo's wireless gamepads are an excellent choice. The quality and looks of these controllers really add that 'wow' factor to any retro gaming build. This one has analogue thumbsticks and comes in a choice of three colours.

£45 | magpi.cc/sn30pro

MEGAPI Case

If you're going to build a retro gaming system, what better than this fantastic scaled version of the Sega Mega Drive from RetroFlag? Their cases are spectacularly well designed and this one is the perfect combination of nostalgia and functionality, especially with the programmable shutdown buttons and cooling fan.

£25 | magpi.cc/megapi



GPi Case

Why not go one step further and make a portable retro gaming system? The GPi Case is a beautiful replica of a retro handheld console, and the attention to detail is breathtaking. A Raspberry Pi Zero (not supplied) is housed in a detachable cartridge and it even runs off regular AA batteries for gaming on-the-go.

£60 | magpi.cc/gpicase

“A Raspberry Pi Zero is housed in a detachable cartridge”



Classic USB Games Controller

If you want functionality without breaking the bank, then the classic USB game controller is an excellent choice. Modelled on a classic controller, this connects to your Raspberry Pi via USB – and a generous cable length means you don't need to sit too close to your TV to play!

£8 | magpi.cc/usbcontroller

▲ Create your own portable retro system with this stunning case

We recommend...

Micro USB to USB-C adapter

This little adapter lets you use your existing micro USB power supplies with the new-style USB-C ports on Raspberry Pi 4.

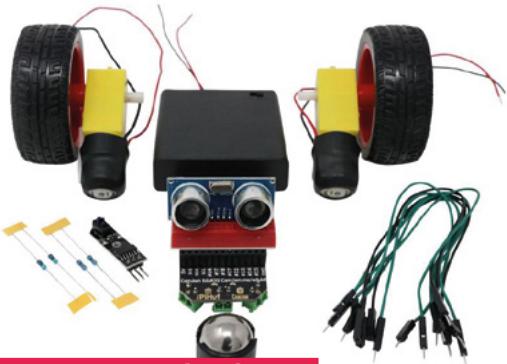
magpi.cc/microusb



◀ An extra-long 1.8 m cable gives this classic controller the edge

Robot building kits

R2-D2 or C-3PO? Or are you more of a BB-8 fan? No matter your favourite, you can always build your own with one of these kits



CamJam EduKit #3

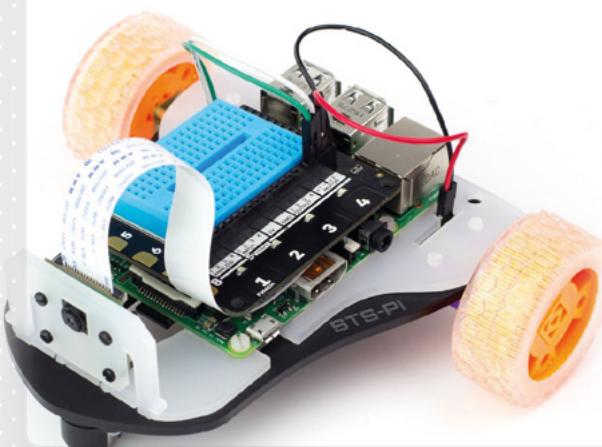
If you are after a budget kit, this CamJam one is a great introduction to robotics. You'll need to supply your own Raspberry Pi and chassis (something to attach the kit to), but it's a great way of getting into the world of robotics before delving into something a little more complex.

£18 | magpi.cc/edukit3

STS-Pi

The STS-Pi is a great little robot kit that gives you the bare bones to build a two-wheeled roving robot. You'll need to supply a Raspberry Pi, Camera Module, and motor driver (such as the Explorer pHAT), but you'll learn the basics of robotics with this nifty kit.

£23 | magpi.cc/stspi



MeArm

These types of robots are used in manufacturing and engineering plants – well, maybe not Raspberry Pi versions, but the same style. With the MeArm kit, you can build a robotic arm that is controlled using the two supplied thumbsticks (or with code). An ideal option for a budding robotics engineer!

£70 | magpi.cc/mearm



▲ Build your own robotic arm with this affordable kit



▼ One of the most rugged robots around, it's a real monster!

MonsterBorg

The title says it all here: this is the ultimate Raspberry Pi robot and is designed to withstand some punishment. The chassis is rugged and made of aluminium, and the wheels make it a great off-road choice, especially with the three hours run time. Oh, and it runs any side up, too!

£210 | magpi.cc/monsterborg

We recommend...

MotoZero

A motor driver capable of powering four motors, this board is a great and affordable choice for any robotic build.

magpi.cc/motozero

Picade X HAT USB-C

If you fancy building your own arcade setup without a kit, this add-on makes controller configuration a breeze. It works with the Pi 4 too!

magpi.cc/xhat

Picade Plasma kit

Want flashy LED arcade buttons instead of plain ones? This kit adds all the jazziness you'll need! It comes in six- or ten-button options.

magpi.cc/picadeplasma

BOOK OF **MAKING**

VOLUME 2



THE BEST
PROJECTS FROM
**HACKSPACE
MAGAZINE**

THE ULTIMATE SKILLS,
TRICKS, AND MAKES

**ONLY
£10**

WHSmith
BARNES & NOBLE

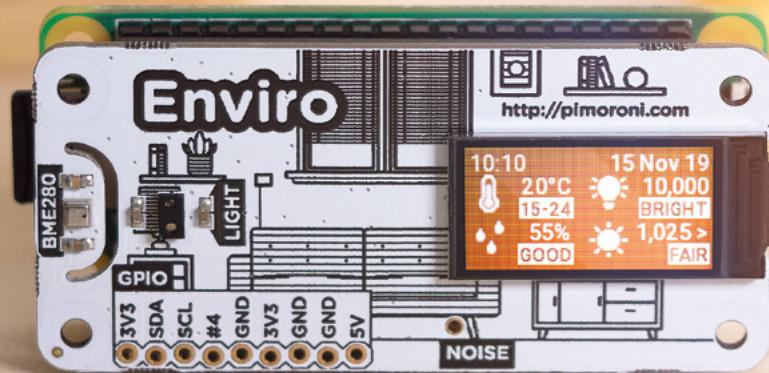
Available on the
App Store

GET IT ON
Google Play

AVAILABLE NOW

hsmag.cc/store

FROM THE MAKERS OF **HackSpace MAGAZINE**



Enviro

SPECS

BUILT-IN SENSORS:

BME280
temperature/
pressure/
humidity,
LTR559
light/proximity,
SPH0645LM4H-B
noise

DISPLAY:

0.96-inch
colour LCD
(160x80)

DIMENSIONS:

65x30x8.5 mm

► Pimoroni ► magpi.cc/enviro ► £28 / \$30

An environment-sensing board with a mini LCD screen to display data. By **Phil King**

Designed for indoor monitoring, the Enviro board enables you to measure temperature, pressure, humidity, light, and noise levels. While it lacks the gas sensor, ADC, and particular matter sensor connector of the Enviro + Air Quality version (reviewed in *The MagPi* #83, magpi.cc/83), it retains the cool mini LCD colour screen to display data – ideal when using your Raspberry Pi in a headless setup, without a monitor.

It also features the same slimline pHAT form factor to match the size of a Raspberry Pi Zero, although it can be used with any Raspberry Pi model.

Sensing the world

The Enviro's BME280 weather sensor monitors temperature, barometric pressure, and humidity. As on the Enviro +, this has been positioned at the left edge of the board, away from Raspberry Pi's CPU, and there's even a little smile-shaped slot around it to help reduce heat radiated through the board. Even so, you'll need to adjust its temperature reading for accuracy (by measuring that of the CPU itself and deducting a factor of it).

A smartphone-style LTR-559 light and proximity sensor detects the ambient light level and also proves handy as a substitute for a push-button when you put your finger on it. The board's tiny MEMS microphone measures sound levels, useful for monitoring noise pollution, and can also be used to record audio.



▲ The rear of the board, showing the pre-soldered female GPIO header

At the time of writing, most of the code examples provided with the Python library (magpi.cc/enviroGH) are aimed at the Enviro + Air Quality board. However, it's easy to edit the code for missing sensors from the all-in-one example to get it showing rolling graphs for temperature, pressure, humidity, and light level on the LCD. There are also a couple of examples that make use of the mic to plot noise levels and frequencies.

If you don't need to test air quality and just want a simple environmental sensor with a built-in display, the Enviro is ideal. The light sensor could also come in handy for use in IoT setups, such as to trigger your lights to come on when the light level falls below a certain level.

If you also require motion/direction sensing, the original Enviro pHAT is still available (albeit without the mini LCD). **M**

Verdict

Lacks the gas sensor of the Enviro + Air Quality board, but features the same mini LCD screen to display your environmental data without a monitor.

8/10

HackSpace

TECHNOLOGY IN YOUR HANDS

THE NEW MAGAZINE
FOR THE MODERN MAKER



ISSUE #26
OUT NOW

hsmag.cc



SmartPi Touch 2

► The Pi Hut ► magpi.cc/smartipi2 ► £24 / \$32

A very smart-looking case for the discerning touchscreen user.

Rob Zwetsloot puts one together for a test

We feel like we slightly overlooked the original SmartPi case, so we're happy to rectify this with the brand new SmartPi Touch 2.

This case is designed to work with the 7-inch official Raspberry Pi touchscreen, effectively turning a Raspberry Pi into a (bulky) tablet or fixed monitor. As there's no real room for rechargeable batteries as standard, it works better in a more fixed environment.

■ You can choose little feet for the display, or a foldable stand ■

One of the unique features of the SmartPi is the interchangeable faceplates on the front of the case. As well as allowing for a Raspberry Pi Camera Module to be mounted inside the case, it enables you to add a Lego-compatible plate. It's not large, so you can't do much with it, but it's a very neat addition nonetheless.

Smart build

Construction is extremely easy, just requiring you to poke some ribbon cables in the right place and fasten and tighten some screws to make sure everything is mounted properly. You can choose little feet for the display, or a foldable stand so you can angle the screen – both are easy to install and even replace. There's also a little case fan and vented rear panel to keep your Raspberry Pi cool.

It's extremely well thought out, and a very handy case if you need a compact 'Raspberry Pi with screen' solution that just looks nice. ■

Verdict

A great case if you need a stationary tablet, although making it mobile might be a little more tricky.

8/10



▲ It looks sleek and compact from the front



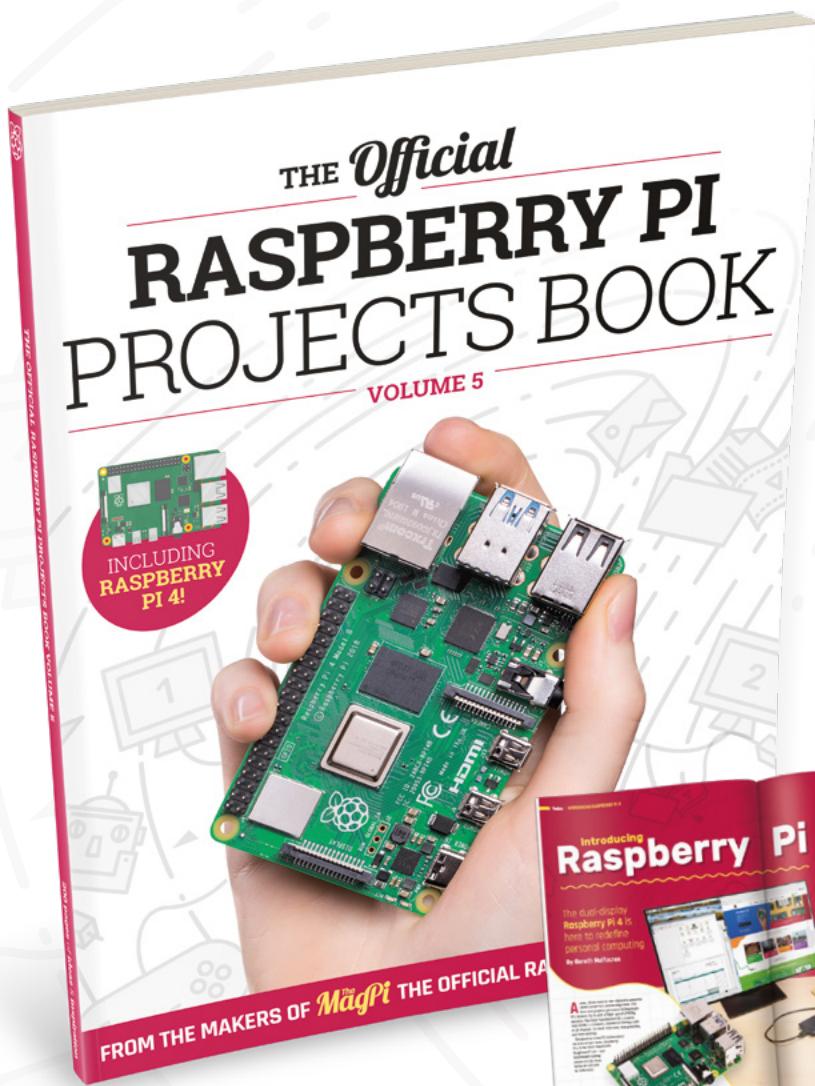
▲ The additional case fan and vents can be handy for intensive loads

THE *Official*

RASPBERRY PI

PROJECTS BOOK

VOLUME 5



200 PAGES OF RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store

10 Best:

Advanced kits and projects

Take your making skills to the next level with these advanced kits and projects

So, you've looked over the tips and tools (page 26), got yourself some cool accessories (page 68), and now you're looking for a challenge. Something fun to build. A big new year project. We have some great ideas for you... 



▲ DiddyBorg

A robotic beast

A serious robot, the DiddyBorg is diddy in name only and a true Raspberry Pi automated monster. Program it, remote-control it, or just have it sitting pretty in your workshop. Your choice.

£220 / \$299 | magpi.cc/diddyborg



▲ PiGrrl 2

DIY handheld gaming

We've covered this project in a previous issue of *The MagPi*, but this project deserves mentioning again: with some 3D-printed parts, and some ingenious Adafruit PCB parts, you too can make a handheld console.

£56 / \$60 | magpi.cc/pigrrl2

▼ Arcade machine

Retro-cool at home

This big build by Bob Clagett is incredibly thorough, and shows you how to build a wooden arcade cabinet from scratch, complete with lights and cool art.

magpi.cc/arcade



▲ Pinball table

High score mania

If an arcade machine is a bit too new for you, how about something a bit more classic and physical like a pinball table? This one repurposes an old bed.

magpi.cc/78

► Magic mirror

Raspberry Pi classic

This kind of project is a classic among the community, especially as the mirror software part is so easily done. Building a frame is quite fun and a great first-time carpentry project.

magimirror.builders



◀ Laptop

Take Raspberry Pi anywhere

This tiny laptop project allows you to bring your Raspberry Pi with you wherever you want to go, and do some work while you're there.

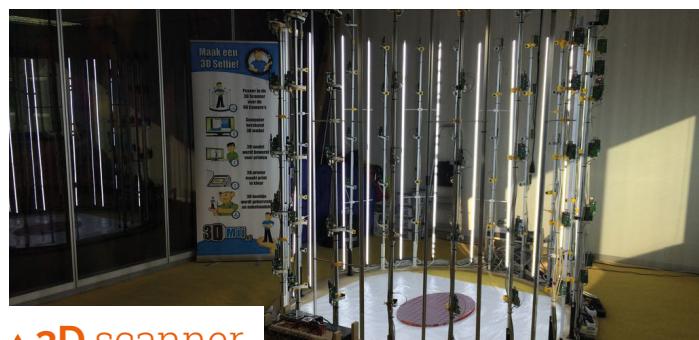
magpi.cc/74

Home assistant

Voice-controlled computer

A number of big voice services are available on Raspberry Pi. Alexa is one of the easiest to get onto Raspberry Pi, thanks to the excellent AlexaPi software.

magpi.cc/alexapi

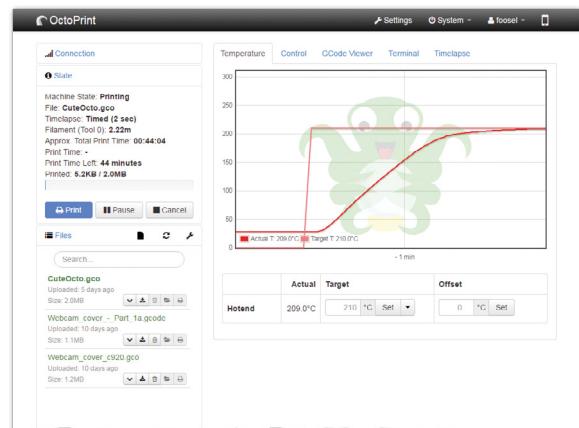


▲ 3D scanner

Scan your friends

This big project is great for showing off at your local Raspberry Jam or maker event. It takes a lot of Raspberry Pi boards and cameras to create the project, but it does capture impressive 3D scans.

magpi.cc/3dscanner

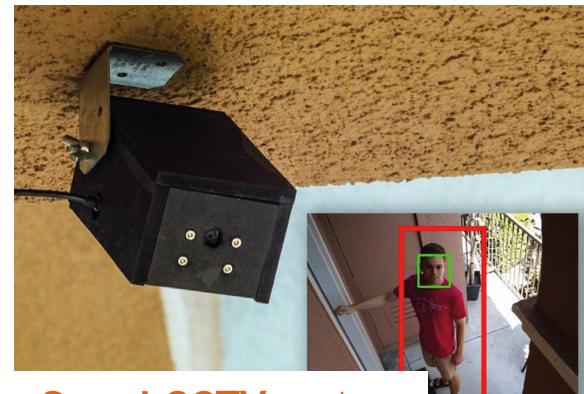


▲ 3D printer controller

Futuristic plastic printing

Have a new 3D printer? How about interfacing it with Raspberry Pi using the amazing OctoPrint? It's especially helpful if you think of something cool to print while on the go.

octoprint.org



▲ Smart CCTV system

Recognise your visitors

CCTV cameras with Raspberry Pi are not too difficult to put together. We like this project as it uses computer vision and face detection to recognise known people as well.

magpi.cc/smartCCTV

LEARN TO PROGRAM!

Got the tools and the skills to raise a barn and then some, but lacking the ability to print 'Hello World'? Take a look at our books on controlling electronics with Python and GPIO Zero (magpi.cc/GPIOZbook) and learning C (magpi.cc/CGUI).

Learn R **with Raspberry Pi**

Discover this statistical language and explore data science. By **Lucy Hattersley**



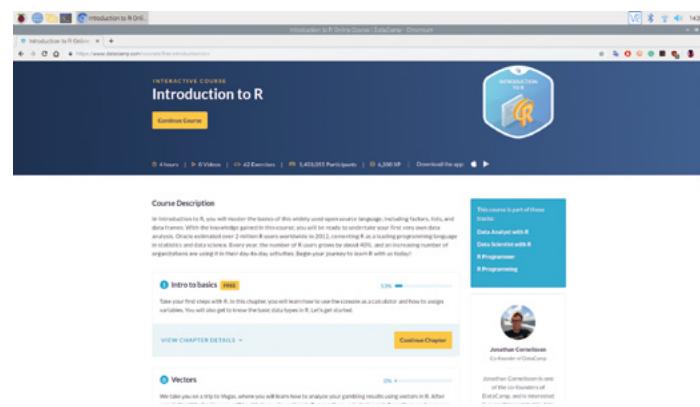
OR DataCamp

CREA
Price:
FREE (or \$568 per year)
datacamp.com

R is a language intrinsically linked to data and statistical analysis. Popular with scientists and number crunchers, it has fans around the globe.

If you've spent a lot of time in Python and other programming languages, some of the features of R are confusing at first. Assignment operators are arrows, and lists are one-indexed (with the first item starting at position one, rather than zero). All of this is designed to make working with large datasets more friendly.

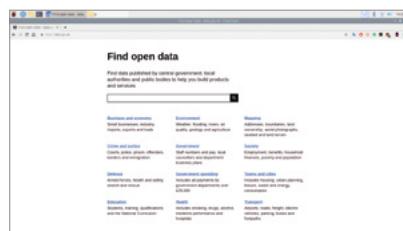
DataCamp is a great learning resource for R, Python, and SQL.



It uses a web-based code editor (which admittedly, we have mixed feelings about). The basic course is free, and you can pay for a DataCamp subscription to access a wide range of advanced

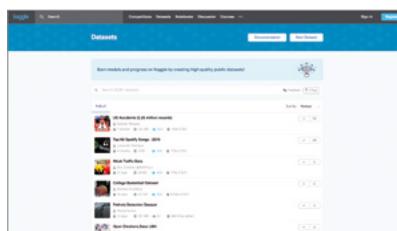
courses. A subscription isn't cheap though, coming in at over \$568 per year, although there are frequent half-price sales and it is aimed at budding data scientists. **M**

Data sources



DATA

The US and UK governments have made huge datasets open. Everything from business figures to the environment, through mapping and spending, can be found online.
data.gov.uk



KAGGLE

Kaggle is an online community owned by Google. It's a great resource for datasets, as well as featuring blogs, competitions, and tools.
[kaggle.com/datasets](https://www.kaggle.com/datasets)



DATAQUEST

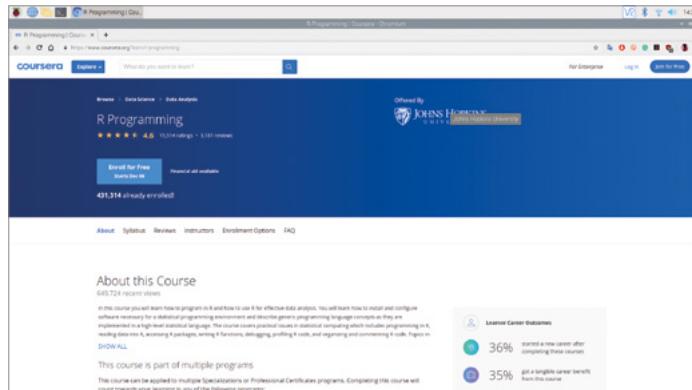
There's a range of datasets around, from Google, Wikipedia, and Amazon, and even news outlets like BuzzFeed. Dataquest has a great list of sources for you to bookmark.

magpi.cc/freedatasets

Coursera

CREATOR
Duke University & John Hopkins University

Price:
£38 / \$49 (per month)
coursera.org



Coursera offers a range of courses from universities. There are two that should be of interest. The first is Introduction to Probability and Data from Duke University (magpi.cc/courseraprobability), with a 4.7 star rating. Led by Mine Çetinkaya-Rundel, Associate Professor of the Practice Department of Statistical Science, the course features R, but it's more about learning to crank data. It gives

you a grounding in probability and Bayes' rule. It covers sampling methods, and forms part of a larger Statistics with R Specialization, which you can take to learn more about R.

The second suggested course is R Programming from John Hopkins University (magpi.cc/courserar). This will get you closer to the R language.

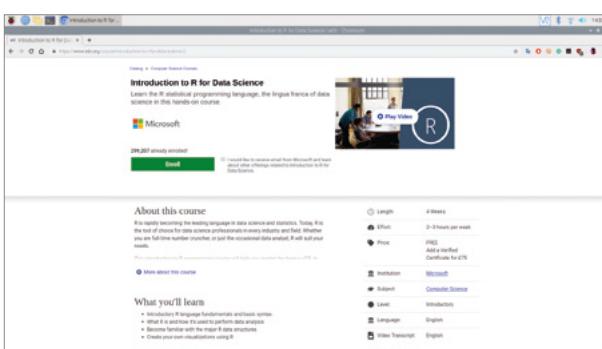
After a seven-day free trial, you'll pay Coursera a monthly fee to access the courses. **M**

Introduction to R for Data Science

CREATOR
Microsoft

Price: FREE (\$99 certificate)
magpi.cc/edxri

We're big fans of the edX platform, which offers a range of courses from respected universities and organisations.

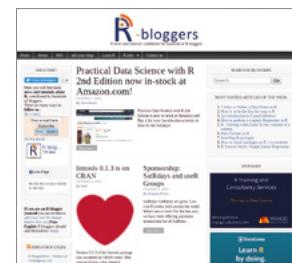


Its Introduction to R for Data Science course is provided by Microsoft, and runs on the DataCamp platform (so it's an interactive web approach). This is interspersed with video tutorials and short online quizzes. And the edX community is vibrant, with an active forum that is ready to answer any questions you might have.

It's an accessible course and, thanks to being on edX, you can enrol and take the course for free. You only need to pay to get a certificate at the end. **M**

R websites

Bookmark these webpages while learning R



R-BLOGGERS

R-bloggers is a website aggregator for blogs on R. In it, you'll find the latest contributions from hundreds of different R bloggers.

r-bloggers.com

R-EXERCISES

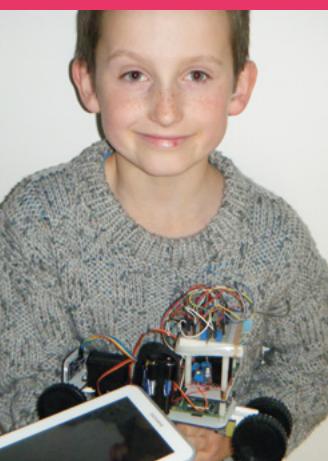
R-exercises aims to help people develop and improve their R programming skills. R-exercises was initiated and is maintained by Research for Decisions, a Dutch research/consulting firm.

r-exercises.com

REVOLUTIONS

Revolutions is a blog dedicated to news for the R community. It's a great place to find out recent developments and news.

blog.revolutionanalytics.com



Andrew Suttle

A young student who makes Raspberry Pi-powered robots at a university club

► Day job **Student** | ► Community role **Club attendee and promoter**

Although Andrew is currently in year nine, that hasn't stopped him from participating in what was once the very grown-up activity of building robots.

"My first Raspberry Jam was CamJam in autumn 2013, when I showed some programs I had written," Andrew tells us. "I also enjoy developing robots and doing electronics. I first went to the Colchester club run by John Woods in 2016, when I built a robot car using a Raspberry Pi. The Colchester club is run at the

university and I like going there very much. The lab we use has all new oscilloscopes and I used them recently with some op-amp circuits. I also built my own oscilloscope from a kit and John Woods helped me get it working, as I had not soldered one of the connections properly. I took my oscilloscope into school and used it there."

Andrew isn't the only young person at the club, either: "Several other children from my school have also gone to the club and I have built projects with

them. We made an 'I ❤ Pi' team and entered competitions. We also had a 'Pi Rates' team. My brother also goes to the club. He is building a security project with Arduino, and I might use some of his ideas in a programming course I am making."

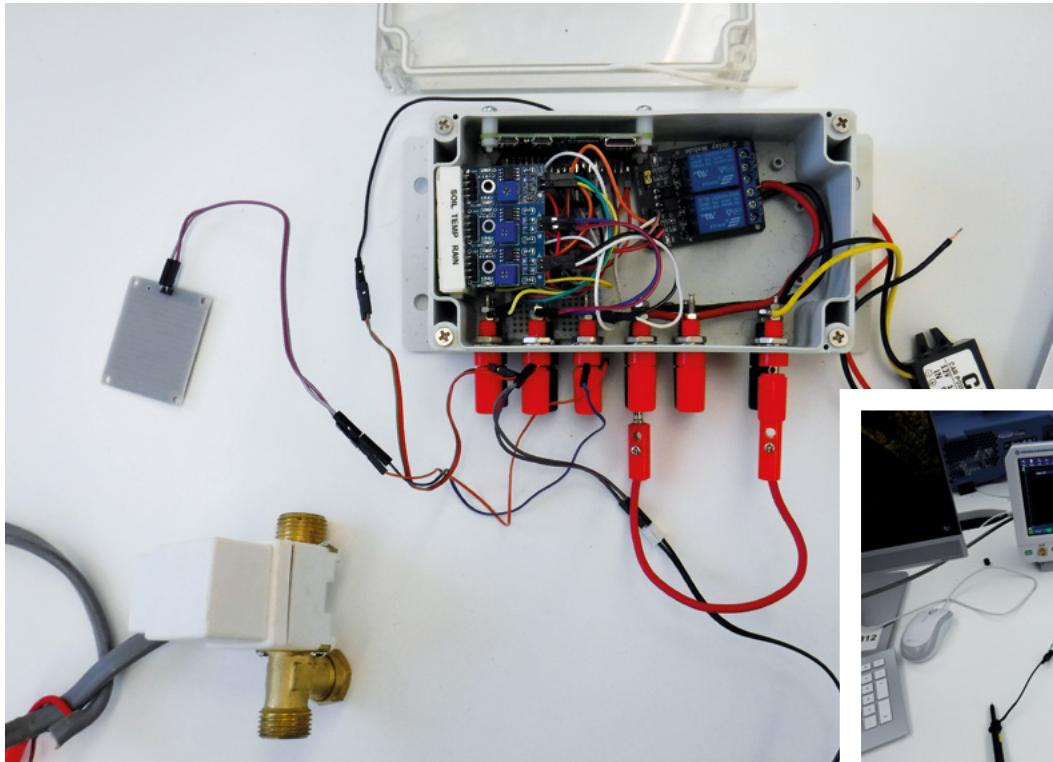
What kind of people attend?

At the moment, it is all children aged 8 to 14 who come along with their parents, but we don't have set age limits. The youngest children usually want to learn Python; the older children usually have a project of their own that they want to work on. People often drop in for a few sessions and solve a particular problem and then stop attending. We have had

▼ While some go there to build physical projects, others learn to code

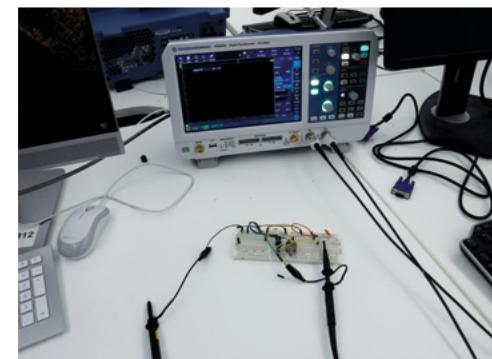


▲ Dr John Woods is a senior lecturer at the University of Essex, and also runs the club



◀ A plant monitor created by two children at the club

▼ Using an oscilloscope, attendees were able to explore modulation



some stay for years and they are all from different local schools.

One boy, Taylor, came regularly until he was 18. He developed several AI projects. He got a scholarship to Cambridge University, and graduates this year.

Other members have entered competitions and we have had members win in the 'BigBang', 'Pioneers', and 'EasternDigital' competitions. EasternDigital is an 'adult' competition for companies and one of the winners three years ago was ten years old; he was too young for their children's competition!

What projects have you or other people made there?

As well as the AI projects I have mentioned, we have had members developing robotic arms. Then there was a flight control system for model aircraft and several robot cars.

■ Some children are doing their own Python projects, with Python always a popular language ■

Several children have gravitated towards embedded systems. Two children, Zara and Andrew, developed a plant monitoring system.

What are you working on at the moment?

Some of the younger children who are new to the club are doing directed tasks. Samuel, aged eight, and Xingtong, aged eleven, are both working on TicTacToe. But we do very little teaching.

Some children are doing their own Python projects, with Python always a popular language. Robyn, aged twelve, has made a Mastermind-style

game linked with Hangman. Andrew, now aged 14, has developed a version of Conway's game of life.

There are also several Arduino projects on the go. Victor, aged eleven, has just finished his security control system which is based on an Arduino Uno and written in C.

There are always robots. We have a robot arena that is purpose-built for the development of robots, and we have a lot of knowledge and resources to support a robot project. The arena is 100 square metres in area, and has a six-metre-high ceiling to accommodate flying robots. ■

Colchester Coding and Robotics Club

"The club is located at Essex University and uses a lab run by the Computer Science and Electronic Engineering (CSEE) school. Getting to the university is easiest by car, with parking free on Sunday. The club is usually in the second-floor lab in building L77-1NW, which is located immediately opposite the STEM centre in Square 1. We meet most Sundays during university term time."

This Month in Raspberry Pi

MagPi Monday

Amazing projects direct from our Twitter

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made. Here is a small fraction of them. Follow along at the hashtag #MagPiMonday. 

01. 3D mapping is some dark magic, but Lorraine has managed to pull it off for her incredible animated tree lights
02. Amazing work from a near ten-year-old, and a great example of fun and simple Raspberry Pi projects
03. People don't mess about with Pi Wars, especially when materials can mean a win or loss
04. This is a very cool and simplistic info cube that we kinda want
05. Who could the mysterious beastie be?
06. This is a cool, abstract project that makes great use of the Twitter API
07. We love a tiny retro console, and this Raspberry Pi Zero-sized one is lovely
08. Razvan sent us a load of great projects with very practical uses
09. Another month, another excellent-looking Pip-Boy build

Lorraine Underwood
@LMcUnderwood

Replying to @TheMagPi

I 3d mapped the Christmas tree lights!

[twitter.com/LMcUnderwood/s...](https://twitter.com/LMcUnderwood/status/6443031000000000)

Blog post here: lorraine.mcunderwood.org/christmas-tree/

01



 Brian Cortell
@CannonFodder

Replying to @TheMagPi

Tom (10 very soon) soldered up his @ThePiHut Christmas Tree by @RachelRayns and setup the Raspberry with no help from me



0:06 932 views

02

 orionrobots
@orionrobots

Replying to @TheMagPi

I've been making parts for my piwars robot, specifically material study to see if I can make battery retaining clips that bend a little, in PLA. Blog post to come.



03

 Daniel Brady
@DanBrady

Replying to @TheMagPi

I finished off the code for this info display at home. Shows the time, buses due, and max temp. I'll also be adding precipitation and bin day reminders soon.

Made using a Raspberry Pi Zero and a @pimoroni Ubercom LED matrix.

#MagPiMonday



04

 Laurence Molloy
@MolloyLaurence

Replying to @TheMagPi

#MagPiMonday

We have a night time intruder in our house. It ripped a food pouch to shreds & sent our cat mental the other night

I set something up ready to catch our night time intruder in the act. I'll set it in front of our cat flap (which it broke in a previous forced entry)



05

 RaspberryPint
@RaspberryPint

Replying to @TheMagPi

I have modified one of my projects to be more Christmasy. It connects to Twitter and sets each of 180 neopixels to a colour representing someone it follows. Each pixel represents a tweet or retweet. It constantly changes as tweets arrive. I presented at last meetup. #MagPiMonday



06

 Mark
@TheMWord

Replying to @TheMagPi

Finished one of my #tinyretroportables. Has a 550mah battery, li-ion charging, 3w mono amp & the tiniest 5mm diameter speaker I could find! And the whole thing is only ~1cm thick



07

 Justin Hart
@jshart

Almost finished the pipboy, all systems working, just a bit more python and paint to go! #MagPiMonday



09

 Razvan T. Coloja
@cypress twist

Replying to @TheMagPi

Made a Raspberry Pi remote control for a fellow Psychologist. Simple Rx/Tx IR LED connected to three pins, LIRC working. Doubles as a webserver he can access through OpenVPN to start/stop his heating via two Wireless S20 Sonoff smart plugs I've installed Linux on.



08



Raspberry Pi turns eight in 2020!

Celebrate your favourite microcomputer on 29 February 2020 at a Raspberry Jam near you

2020 is one of those rare years: a leap year. February, the greatest month, receives an additional day, which just so happens to be the true birth-date of Raspberry Pi: 29 February.

It also lands on a Saturday, which makes it perfect for the return of the Birthday Jams! Here's a special message from Philip Colligan, CEO of the Raspberry Pi Foundation, on how you can get involved.

Celebrate at a Raspberry Jam

Raspberry Jams are community-led meet-ups that bring people together to share, connect, and

learn from each other. The first one was held in Manchester in 2012, and so far Jams have been held in more than 70 countries – and that's just the ones we know about.

While Jams take place throughout the year, there's a special tradition of Jams celebrating the birthday of the Raspberry Pi computer. This year, there were over 130 Raspberry Jam events in 39 countries, attended by 8000 people!

Register your Birthday Jam and we'll send you some special swag

Next year, because it's a big birthday, we'll be sending a special box of swag to any Jam that is taking place between Saturday 15 February and Sunday 15 March 2020.

It's really simple to register your Birthday Jam: just fill in the Raspberry Jam submission form (magpi.cc/jamform), including a valid event information URL linking to a webpage with more information about your event.

We'd prefer you to link to a public ticketing system (e.g. Eventbrite) if possible, but we know some libraries and community centres have restrictions that prevent them from doing this.

In order to ensure that your pack reaches you in time, we need you to register your Birthday Jam at least six weeks before your event.

As always, if you have any questions, please don't hesitate to ask us: jam@raspberrypi.org.



Crowdfund this!

Raspberry Pi projects you can crowdfund this month



Tablet Frame Kit

This kit is quite neat – it's part of an effort to create a low-cost computing solution that uses a Raspberry Pi for schools and classrooms. It can even fit a breadboard on the back of it. It's also able to act as a stand for several other low-cost tablets, such as Amazon Fire.

► kck.st/2OgRvkF

Vigibot, the ultimate web remote controller!



Make your own robot, learn and have fun!

Vigibot

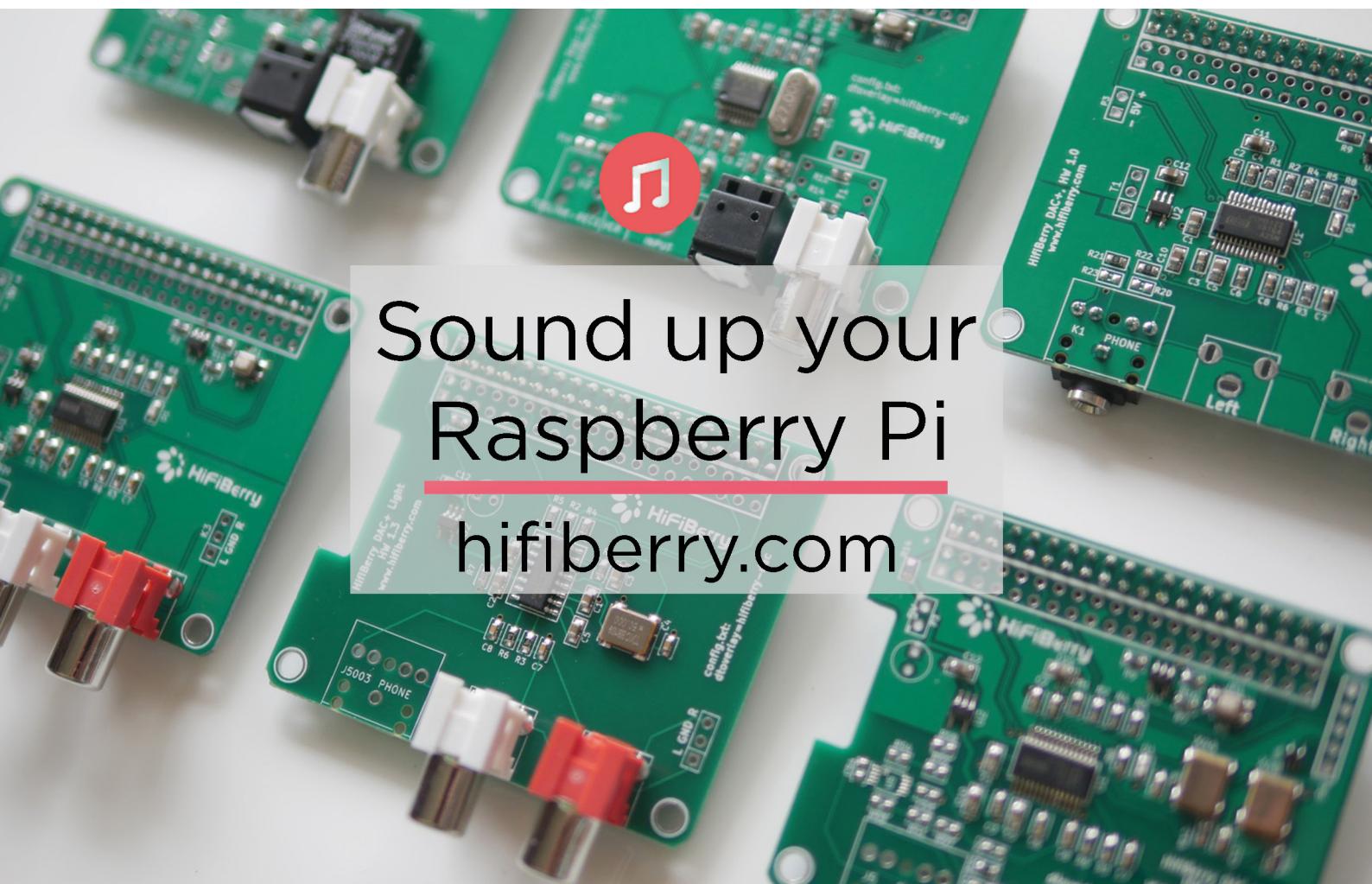
Vigibot is a web-based control scheme for Raspberry Pi-powered robots. It has some pretty great functionality for all kinds of movement and actions, and includes a nice display that can show a live view from a Raspberry Pi camera – with an optional pan-and-tilt mechanism.

► kck.st/2QuEvt3

**CROWDFUNDING
A PROJECT?**

If you've launched a Raspberry Pi-related project, let us know!
magpi@raspberrypi.org

Sound up your
Raspberry Pi
hifiberry.com



Raspberry Jam Event Calendar

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

01. Raspberry Jam Taiwan

- Monday 30 December
- 📍 2F, No.105, Sec. 1, Chongqing S. Rd, Taipei City, Taiwan
- ▶ magpi.cc/yXigDK

This Raspberry Jam will be focusing on uses of the Compute Module in solar power and more.

02. Castro Valley Jam

- Saturday 4 January
- 📍 Castro Valley Library, Castro Valley, CA, USA
- ▶ magpi.cc/7nrwp2

If you're interested in computer coding, you can tinker and code electronics at the Castro Valley Jam.

03. Exeter Raspberry Jam

- Saturday 4 January
- 📍 Exeter Library, Exeter, UK
- ▶ magpi.cc/TYDo6

A meeting for everyone interested in all things computers, microcontrollers, robotics, and making.

04. Cornwall Tech Jam

- Saturday 11 January
- 📍 Cornwall College Camborne, Camborne, UK
- ▶ cornwalltechjam.uk

For anyone interested in technology, of all ages and abilities. Ask questions and learn about programming.

05. South Devon Tech Jam

- Saturday 11 January
- 📍 Paignton Library and Information Centre, Paignton, UK
- ▶ magpi.cc/9vhGQ5

A monthly informal and friendly session for anyone interested in technology, regardless of age or ability.

06. WHPL Raspberry Jam

- Monday 13 January
- 📍 Public Library, West Hempstead, NY, USA
- ▶ whplibrary.org

Learn to code in this Raspberry Jam using the library's brand new Raspberry Pi computers.

07. Raspberry Jam Zelzate

- Saturday 25 January
- 📍 Openbare Bibliotheek Zelzate, Zelzate, Belgium
- ▶ magpi.cc/eUWnsH

Everyone is welcome to start, share, and work on their own projects in a fun and relaxed atmosphere.

08. Topsham Raspberry Jam

- Saturday 25 January
- 📍 Nancy Potter House, Exeter, UK
- ▶ magpi.cc/ysRxpX

A Code Club turned Jam. You'll need to call the number at the League of Friends website to enrol.

FULL CALENDAR

Get a full list of upcoming events for January and beyond here:

rpf.io/jam



Raspberry Jam advice: Volunteers

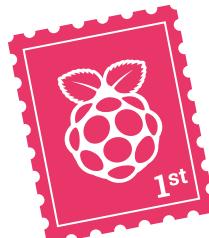
Our volunteers have badges that say 'Jam Maker' on them. Sometimes we have lanyards. Tim and I tend to wear our purple CamJam polo shirts. They weren't cheap, but we do stand out!"

Michael Horne – Cambridge Raspberry Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: rpf.io/guidebook



Your Letters



Case necessity?

Are cases for Raspberry Pi truly necessary? I understand they don't really come with one and have seen many people use one outside of a case.

Frankie via Twitter

For the longest time with our very first Raspberry Pi, we didn't use a case with it. Even now, if we're putting together a quick thing for the magazine, we'll just have a Raspberry Pi sitting loose on a (wooden) table. So it's not 100% necessary, but we always recommend getting a decent case – especially if your Raspberry Pi might sit in one place for a long time and feasibly get dusty.



▲ The official case is very sleek and stylish, and we really like the new black version!



▲ We hope to inspire you in the pages of *The MagPi*

New user

I'm getting my first Raspberry Pi for Christmas and am really excited by the possibilities. The only problem is that I feel a bit paralysed by choice! What should I do with my Raspberry Pi? Should I do many things? Will I need more? I hope you can help with these questions.

Eddy via email

It can be difficult to choose what to do with a Raspberry Pi once you have one! We've all been there ourselves. In this issue we have a few articles you might consider reading to get some ideas – first, there's the 50 tools and tips feature (page 26). Here you can learn about any extra gear you need to start your maker journey, and maybe spark some ideas of what kind of project you want to make. Then we have the accessories and add-ons feature (page 68), which can give you advice on stuff to get for your Raspberry Pi, and the advanced kits and projects list (page 80) might give you some ideas as well.

If all else fails, check out our latest *Official Raspberry Pi Projects Book, Volume 5*, for amazing ideas to help inspire you: magpi.cc/projects-book5.

Contact us!

- Twitter [@TheMagPi](https://twitter.com/TheMagPi)
- Facebook [magpi.cc/facebook](https://facebook.com/magpi.cc/facebook)
- Email magpi@raspberrypi.org
- Online raspberrypi.org/forums

SD card requirements

Whenever I see a sale for microSD cards online, I always feel like I should stock up on some for Raspberry Pi projects. Are there any limitations on size and speed and type of card, though? I've had issues in the past with cards getting corrupted in cameras and such, so want to make sure I don't get any bad ones!

Lou via email

Most microSD cards will work fine with a Raspberry Pi. However, we usually advise to get 8GB or 16GB cards, and no smaller unless you know what you're doing. Higher-capacity cards aren't always necessary, and we find that the larger you go, the less likely they are to work with Raspberry Pi. For a list of tested cards, take a look at magpi.cc/sdcards.

In our experience, though, you're better off getting a 16GB card and a big USB storage device.



▲ It's best to check if a particular microSD card works with Raspberry Pi before buying it



▲ Super Crate Box is a great game from the creators of Luftrausers

Raspberry Pi games

I've read a lot of stuff over the years about emulating old systems on Raspberry Pi, and using it as a streaming box for your gaming PC. However, I was wondering if there were any games that were released for Raspberry Pi natively?

Lee via Facebook

It's not a popular computer to release games on. There have been a few made for it, though. Most notably, a few indie games made in GameMaker: Studio were released on Raspberry Pi back in 2016 (you can read more here: magpi.cc/gamemaker), and Doom runs on Raspberry Pi if you can find the right files for it.

As well as GameMaker, Tilengine (tilengine.org) allows for games to be developed for Raspberry Pi, so if you fancy making anything yourself, check them out!

■ A few indie games were released on Raspberry Pi back in 2016 ■

CUSTOMPC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING

THE MAGAZINE FOR

PC HARDWARE ENTHUSIASTS

REVEALED THE HARDWARE YOU REALLY NEED FOR GAMING

CUSTOMPC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING / ISSUE 197

ALL THE CORES

ACCELERATE CONTENT CREATION WITHOUT LOSING GAMING PERFORMANCE

32 | AMD THREADRIPPER
Cores 3970X

24 | AMD THREADRIPPER
Cores 3960X

18 | INTEL CORE i9
Cores 10980XE

16 | AMD RYZEN 9
Cores 3950X

+ **PLUS**
GAMING LAPTOPS
GEFORCE RTX ON THE GO
FROM £1,699

FEBRUARY 2020 / £5.99

9 771140 744059

STEP DIVE WE DELVE INTO NVIDIA'S TURING ARCHITECTURE

CUSTOMPC

BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING / ISSUE 196

PC BUILDING MASTERCLASS

FULL STEP-BY-STEP GUIDE

MAKE A SMART PC JUST LIKE THE PROFESSIONALS

HARDWARE INSTALLATION

CABLE TIDYING

GETTING YOUR COOLING RIGHT

NEXT-GEN WI-FI

WE TEST THE LATEST 802.11AX ROUTERS

MAKING YOUR OWN VELCRO CABLE TIES

9 771140 744059

ISSUE 197 OUT NOW

VISIT CUSTOMPC.CO.UK TO LEARN MORE

WIN ONE OF FIVE RASPBERRY PI 4 & CASE BUNDLES

Still after a Raspberry Pi 4? We have five to give away, complete with one of the stylish official black and grey cases as well.



Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/raspberrypi4

Terms & Conditions

Competition opens on **18 December 2019** and closes on **30 January 2020**. 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. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

Wireframe

Join us as we lift the lid
on video games



Visit wfmag.cc to learn more

NEXT MONTH | *MagPi*

BUILD A RASPBERRY PI 4 **MAGIC MIRROR**

MAKE A FUTURISTIC **SUPER-SMART** SCREEN

THE MAGPI #90
ON SALE **30 JANUARY**

Plus!

**Starter projects for
the New Year**

The best Raspberry
Pi 4 thermal cases

**How to overclock
your Raspberry Pi**

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER @TheMagPi

FACEBOOK [fb.com/MagPiMagazine](https://www.facebook.com/MagPiMagazine)

EMAIL magpi@raspberrypi.org

EDITORIAL

Editor

Lucy Hattersley
lucy@raspberrypi.org

Features Editor

Rob Zwetsloot
rob.zwetsloot@raspberrypi.org

Sub Editors

Phil King and Nicola King

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.org
+44 (0)7725 368887

DESIGN

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Sam Ribbits, Harriet Knight,
Ty Logan

Illustrator

Sam Alder

CONTRIBUTORS

Wes Archer, Mike Cook,
David Crookes, Tracy Gardner,
Rosie Hattersley, Nicola King,
Dale Lane, Simon Long,
Danny Staple, Mark Vanstone

PUBLISHING

Publishing Director

Russell Barnes
russell@raspberrypi.org

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

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk



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

The MagPi magazine is published by Raspberry Pi (Trading) Ltd, Maurice Wilkes Building, St John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.





To a new era

Rob Zwetsloot looks back at the decade of Raspberry Pi and The MagPi

I remember sitting in my home at the start of the decade and feeling proud of myself. I'd cobbled together a media PC from a load of spare parts and cheap components from eBay, running the XBMC-based Boxee. A big old tower sitting next to a TV with a wireless keyboard and (ball) mouse for control. This was the future. It couldn't get better than this, even if it was over S-Video.

It's funny how much can change in ten years – or in this case, about three ■

It's funny how much can change in ten years – or in this case, about three, when a computer even cheaper and far smaller than my media PC came out: the humble Raspberry Pi Model B. I soon replaced the old machine with a Raspberry Pi with XBMC (now Kodi) installed and only recently upgraded it when Raspberry Pi 4 was released earlier this year.

At the time I was amazed at how much Raspberry Pi could do. Little did any of us know just what it could achieve in the 2010s. A revolution in home robotics, helping overhaul

school education, bringing free computing resources to everywhere in the world, and being part of a new wave of makers.

Eight years of Raspberry Pi

It's amazing to hear about the untold millions of Raspberry Pi computers sold over the last eight years, and even better seeing young folks enter the workforce who grew up with a Raspberry Pi. Yet still there are

new, young makers popping up with incredible Raspberry Pi projects.

Working on *The MagPi* for the last four years has been a dream as well. Helping to highlight amazing projects, amazing events, and amazing people throughout the community and across the globe, and there's no sign of that slowing down any time soon.

I have many personal highlights from the last 50+ issues I've worked on, but I always come back to one: writing the main feature that went along with issue 40 – the Raspberry

Pi Zero launch issue. We were all blown away by the response to its release, and I'm very proud that we were the first magazine to have a free computer on the cover. I still get emails asking if we have any left (not for several years, unfortunately), and Raspberry Pi Zero still amazes everyone I introduce it to.

Welcome to tomorrow!

Heading into 2020, I'm even more excited about the future of Raspberry Pi, making, and everything related to it. What incredible new projects will become trivial builds by 2030? How many more young people and kids will be enamoured with Raspberry Pi? Will Raspberry Pi go deeper into space? And will my media PC get smaller and better still?

Whatever happens, I'll be here to share and celebrate all that the community does with Raspberry Pi. Let's make the next decade amazing. ■

Rob Zwetsloot

Rob is amazing. He's also the Features Editor of *The MagPi*, a hobbyist maker, cosplayer, comic book writer, and extremely modest.

magpi.cc

AUTHOR

— Uni versal & Unique —



UniPiCase

for Raspberry Pi 4

Perfect option for your DIY or high-volume commercial projects



Fits all kinds of HATs
Ideal PoE HAT case



Use with HAT or Pi alone
Simple and professional



Rapid, tool-free assembly
GPIO cable pass-through



Two wall mount options
Well ventilated



www.UniPiCase.com



Designed and manufactured in North America



OKdo™
DESIGN THE WORLD
[f](#) [in](#) [t](#) [u](#) [y](#) [I](#)