The MagPi

Issue 86 | October 2019 | magpi.cc

The official Raspberry Pi magazine

10 RETRO GAMING KITS
The best arcade controllers, kits and consoles

POWERFUL RASPBERRY PI PROJECTS

25 *rewarding builds for* **masterful makers**

SPOOKIFY YOUR HALLOWEEN COSTUME

LEARN TO CODE WITH SCRATCH 3
Easy programming with this block-based language

£5.99

BUILD AN INTERACTIVE ROLE-PLAYING TABLE

MAKE 8-BIT GAMES WITH RETRO MUSIC & SOUND FX

GLOBAL DELIVERY magpi.cc/store

Raspberry Pi PRESS

WELCOME
to *The MagPi* 86

We like to think we're a pretty welcoming bunch to newcomers here at Raspberry Pi Towers. After all, our favourite computer was designed to help students (of all ages) to learn computing skills. Everybody has to start somewhere, and there's nowhere better than *The MagPi*. This magazine, like every issue, is packed with great tutorials and projects for the absolute newbie.

If that sounds like you, stop reading now and head over to our Scratch 3 tutorial (page 40). In it, Sean McManus will show you how to code with the latest version of the ultra-friendly visual coding tool.

Right! Everybody else. There comes a point where every Raspberry Pi owner wants to get their teeth into something with more bite. A big ol' project that needs time, tools, and tenacity. So, this month our lead feature is Powerful Projects for Raspberry Pi (page 26). We've asked the community to let us know what ultra-powerful projects they've heard about, been testing out, and what they want to make next.

The result is 25 of the biggest, baddest, most breathtaking projects in existence. From space ballooning and soccer robots to networked knitting machines, intelligent cucumber sorters, and classic toy-hacks, right up to hydroponics.

PJ has interviewed the makers and got tips and advice. All you need is the get-up-and-go to get started. So dive into our feature, and let us know what you're going to make.

Lucy Hattersley Editor

Contents

> Issue 86 > October 2019

POWERFUL RASPBERRY PI PROJECTS
26

18
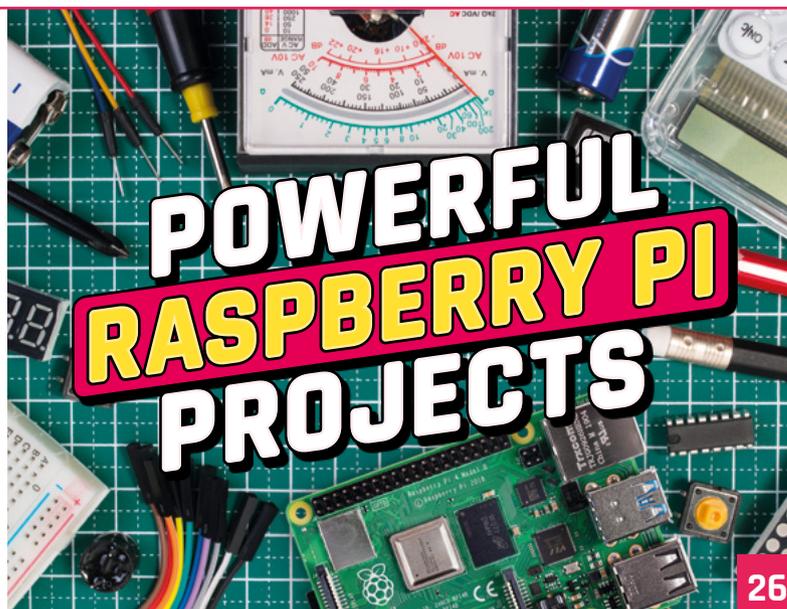
10
MudPi

Stonehenge Skyscape

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.

40

Throttle & Tilt

60

Make music in PICO-8

76

PecanPi

84

Sean Raser interview

Tutorials

The Big Feature

70

Halloween costume projects!

Reviews

Community
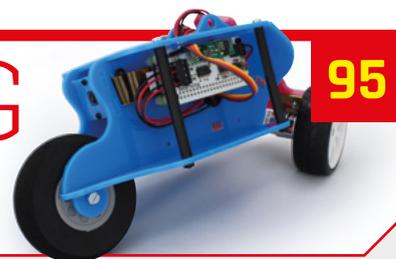
WIN
ONE OF FIVE ROCKYBORG **ROBOTS!**
95

Raspberry Pi
hosts Scratch Conference Europe

The cute coding cat is all grown up and plays a pretty big role in education, explains conference attendee **Rosie Hattersley**

The UK hosted the Scratch Conference Europe for the first time this August, welcoming 300 delegates and visitors from all over Europe and beyond.

There were 40 talks and 30 workshops, plus keynote speeches from MIT Media Lab's Mitchel Resnick and Raspberry Pi Foundation CEO Philip Colligan, among others.

The two-day conference focused on creative ways that educators and Code Club and CoderDoJo volunteers can inspire others using the visual coding platform. In part, it was a showcase for the possibilities of Scratch 3 – newly released in a dedicated Raspbian Buster version this summer.

▼ Teenage volunteer Thomas Southwell explains his role helping in the Scratch Bugs and Glitches forum

▲ Educators in Japan are getting primary school students programming

Mitchel Resnick talked about how both Scratch and Raspberry Pi had evolved from being "a little fringe activity" of interest to kids with an interest in technical careers. Embraced by governments and schools, both are now available to "more children from more backgrounds to design, create, experiment, explore, and learn through coding".

Mitchel presented the example of high school graduate CrazyNimbus, who credited Scratch with introducing her "to something she's truly passionate about" and has hopes of making computer science a key part of her career.

Fascinating workshops

Elsewhere at the Scratch Conference, there were presentations and workshops about overcoming common coding and learning barriers, but also about how to take Scratch and coding knowledge to the next level – becoming a Scratch Ninja and mentor for whole schools.

❝ More children from more backgrounds design, create, experiment, explore, and learn through coding ❞

We saw Scratch being used as a tool for learning different languages with visual vocabulary prompts, and Catrobat's Pocket Code app for iOS and Android that enables users of all ages to develop and then share their own mobile games and animations.

One fascinating workshop showed the Snap!-based embroidery design tool at work. Using Turtlestitch's Scratch-like Snap! interface, you can create geometric patterns, write words and phrases using a range of fonts, or make free-form designs. The instructions are then saved to a USB and printed out on a compatible embroidery machine.

Keynote speeches from the Scratch Conference Europe are online at **magpi.cc/rGwxnx**. ⓜ

▲ Joek von Montfort uses Turtlestitch to customise his conference swag bag

▶ MIT's Mitchel Resnick talks about the future of Scratch

Shaun the Sheep
enters space race

{ code club }

Aardman Animations has invited Raspberry Pi and Code Club on board its latest feature-length space mission, delighting **Rosie Hattersley**

I n a rather fantastic turn of events, Raspberry Pi has become the Educational Partner for the Oscar-winning animation studio's latest film launch, *A Shaun the Sheep Movie: Farmageddon*.

Both Aardman Animations and Raspberry Pi inspire young imaginations, so it seems appropriate that the partnership takes the form of a worldwide challenge for youngsters to create their own mini animation, telling the story of Shaun the Sheep and his new alien-friend Lu-La's space adventures.

The global Code Club competition is split into two age categories: 11-and-under and 12-and-above. The Scratch-based challenge asks entrants to tell the story of Lu-La and Shaun the Sheep. Entries will be judged on storytelling, creativity and coding, and must follow the rules of the competition.

An intergalactic challenge

Shaun the Sheep: Mission to Space is a project-based competition designed to be completed over approximately two Code Club sessions, resulting in an animation of up to 59 seconds long.

Instructions and resources are provided in eight different languages. You'll be competing

▶ Aardman Studios and Raspberry Pi have teamed up to run a project-based competition

▲ Shaun encounters Lu-La, an adorable alien with magical powers

▼ Shaun takes his extra-terrestrial friend on a road trip to Mossingham Forest to find her lost spaceship

for the prestige of having your animation hosted on the website, goody-bags, and a video message from Aardman.

The submission deadline is 25 October and entries can be submitted by individuals or groups of up to three children. The ten winning animations will be announced in early November on the Code Club blog, and on Twitter and Facebook.

Code Club exclusive

The competition is open exclusively to registered Code Clubs across the world. If you're interested in starting a Code Club in your community, register to get involved. Head to **codeclub.org** if you're based in the UK, USA or Republic of Ireland, or to **codeclubworld.org** to start a club in the rest of the world.

Character chase

Of course, as fans will know, Aardman has prior experience of space travel: in 1999, Wallace and Gromit jetted off to the moon in search of cheese in *A Grand Day Out*. The new film, *A Shaun the Sheep Movie: Farmageddon*, is due to hit cinemas this autumn. Ⓜ

" Compete for the prestige of having your animation hosted on the website "

MudPi

Want to automate a garden that grows delicious veg? **Rob Zwetsloot** gets out his gardening gloves to see how to do it

MAKER

Eric Davisson

Eric is a web developer who also has a small software and graphic design company that he's building with his wife.

mudpi.app

Warning!
Mains electricity

This project uses mains power. Be careful if you plan to recreate it

A couple of months ago, we looked at a Smart Herb Garden, a smaller version of the automated garden concept. A concept that we love here at *The MagPi*, so it was a delight to hear from Eric Davisson about his MudPi system.

"MudPi is an automated garden system, built open-source, that helps me grow the produce I love with less time, resources, and effort," Eric explains. "The system controls the irrigation to the garden and takes sensor readings to determine growing conditions. Soil moisture, temperature, humidity, and rain data is all collected and used to manage watering control."

It's highly customisable and scalable as well, to suit your garden and changing requirements during its lifespan. And it all runs on a Raspberry Pi.

▶ Sensors keep an eye on all the important environmental aspects like soil moisture and rain

How did a system like MudPi come about? Well, it turns out that Eric is a very keen gardener. "I have a passion for gardening and each year have expanded my garden with additional raised beds and growing areas," he tells us. "Maintaining and keeping the garden up is a task that quickly grows in demand. While looking for solutions to reduce

> ❝ It became apparent that automating the watering would be my first goal ❞

my workload, it became apparent that automating the watering would be my first goal."

He started off with a cheap hose timer. However, the lack of remote control and no way of sensing whether or not it was raining didn't get him far.

Automate gardening and growing produce

Sensors are placed around the garden and in the soil to keep an eye on the environment
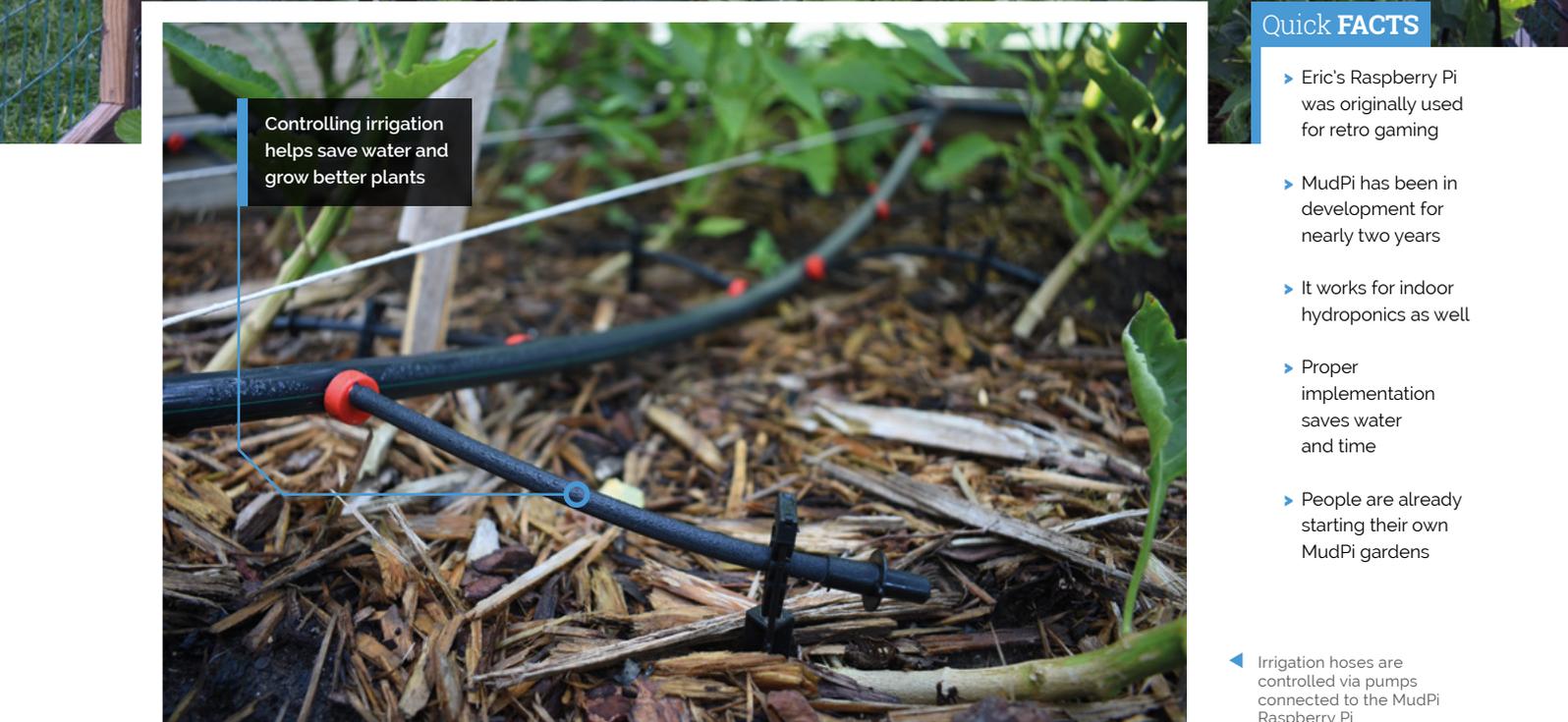
Controlling irrigation helps save water and grow better plants

Quick **FACTS**

> Eric's Raspberry Pi was originally used for retro gaming

> MudPi has been in development for nearly two years

> It works for indoor hydroponics as well

> Proper implementation saves water and time

> People are already starting their own MudPi gardens

◄ Irrigation hoses are controlled via pumps connected to the MudPi Raspberry Pi

▲ Indoor hydroponics can also be controlled by MudPi
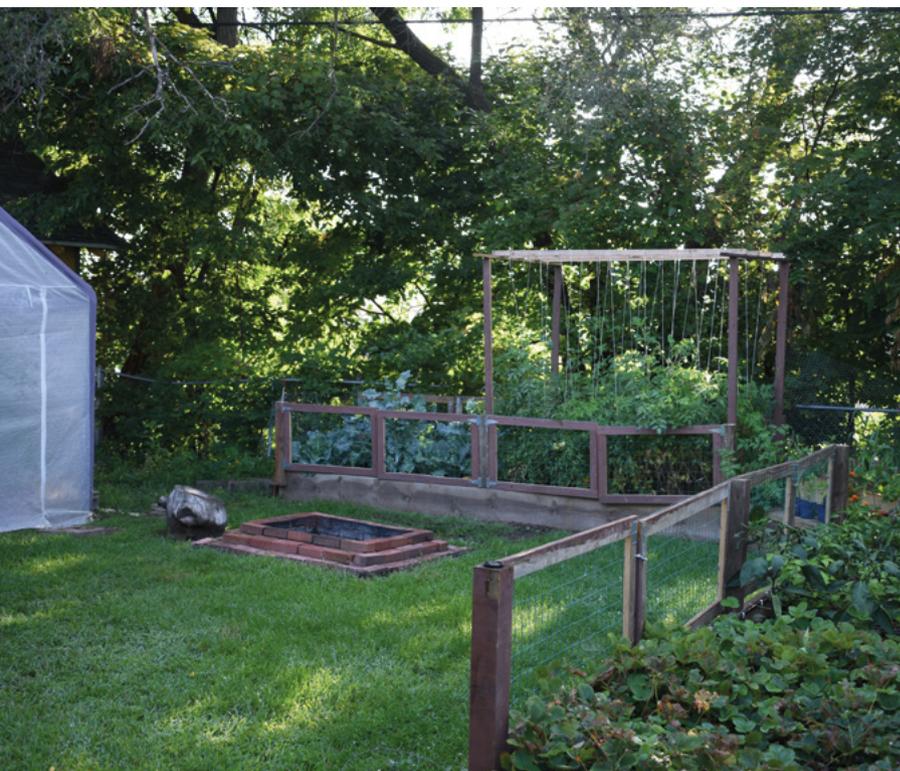
▲ Eric has a big garden that is constantly growing bigger

"I've had a Raspberry Pi sitting around that my wife bought for me and thought this would be the time to try and use it for this project," he says. "What started with just a simple task of turning on a pump has grown into a much more robust solution of helping maintain my garden that

> 🟦 Most of the resources I learn from are driven by communities of creative people supporting one another 🟦

others could benefit from too. Most of the resources I learn from are driven by communities of creative people supporting one another. MudPi is a great opportunity for me to give back to those communities."

Proof is in the parsnips
According to Eric, the project has been in development for just under two years and has already been a great solution to reduce the

▲ This MudPi system is hooked up to the mains – please be careful if you plan to make one yourself!
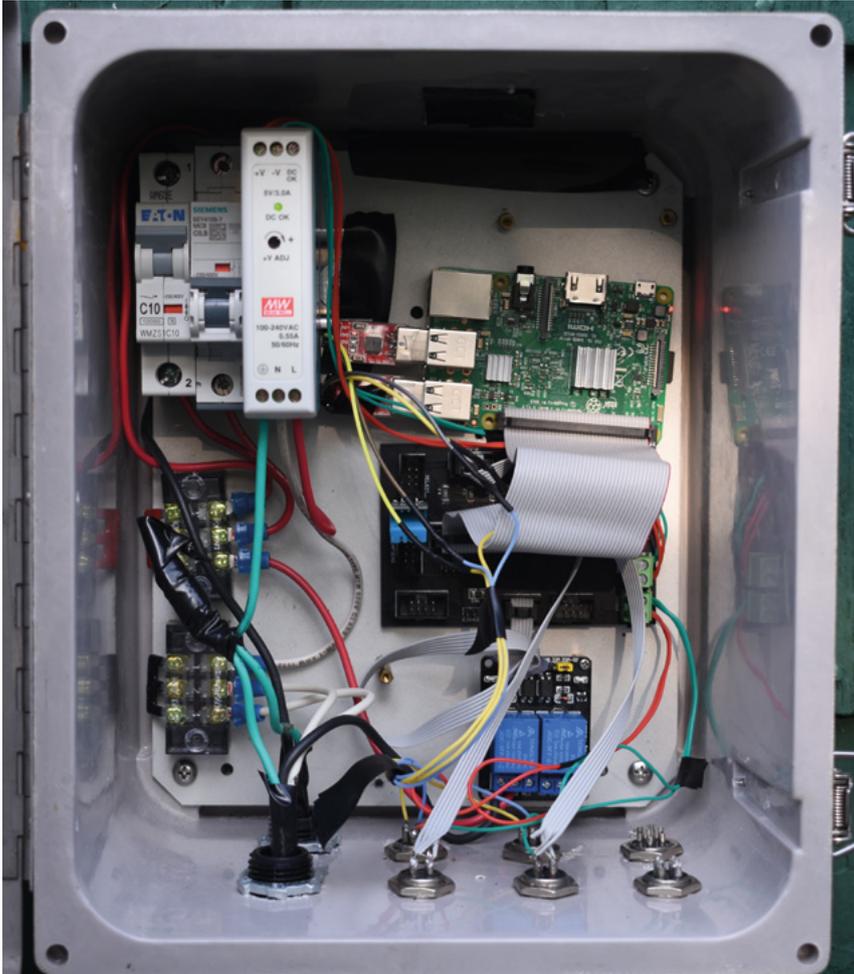
amount of time and water the garden needs. "Each year a new raised bed has been added and it was really easy to make another sensor unit to go along with it," he elaborates. "I spend less time having to water and weed the garden every year, which has been a joy. Not only does this work on my garden outside, but I have a grow shelf inside were another MudPi setup controls LED lights and small pumps. The benefit of the design is I can run a setup as minimal as a single temperature/humidity sensor for weather readings, all the way up to a setup with many attached units and components."

Feel a bit inspired yourself? There are detailed guides and project info on the MudPi website, **mudpi.app**, where Eric has created the resources he had a tough time finding when he started the project. There will even be kits available in the future. We hope you have a bit of fun in your garden thanks to MudPi. M

Build MudPi

01 Plan your garden – specifically, how you plan to irrigate it. Once you understand that, you can sort a system out however you wish, as MudPi will interact with the pump and solenoids to control it.

02 The MudPi software can be found on GitHub (**magpi.cc/zNOfgk**) and installed straight to a Raspberry Pi running Raspbian by just downloading a zip of the archive and extracting it all. You can then attach any components and extra sensors.
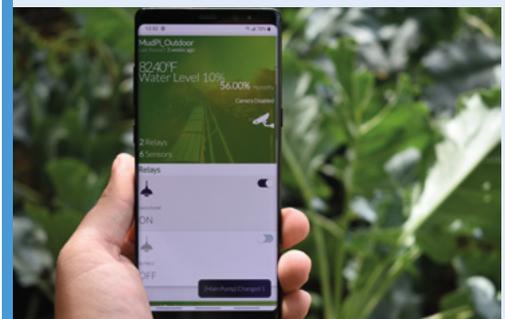
03 Once everything is connected and set up externally, you then need to add them all to the configuration file for MudPi. With that all done, you can run the main script, sort out scheduling, and (mostly) sit back and watch.

Press a single button to share captured dash-cam footage to your Dride.io cloud storage area

The adjustable-angle dash-cam attaches to a windscreen or mirror

Dride Zero
for Raspberry Pi

Dride Zero is a video-sharing dash-cam you can set up yourself and run from a Raspberry Pi. Intriguing, says **Rosie Hattersley**

MAKER

Yossi Neiman

Yossi Neimann is a full stack software engineer and developer based in San Francisco and Israel, primarily focused on Linux. He's a big fan of open-source and credits the open-source community with helping to improve Dride from its first iteration.

dride.io

Scary footage of vehicular near misses, crazily inattentive driving, and unfortunate encounters with wildlife, stray car parts and extreme weather events... Dash-cam footage has brought a new perspective to the everyday commute. It has also highlighted just how tenuous some people's control of their vehicles and sense of proportion can be.

▲ You can attach your Dride Zero behind the rear-view mirror, so that it doesn't obscure your vision

Engineer and developer Yossi Neiman, and partner Shiran Tasa, sensed a place for a 'fusion dash-cam and safety system' that has a social video-sharing element while also alerting others to poor driving etiquette and road dangers. There's a driving alert feature that kicks in when Dride Zero senses something crazy is happening, and it duly records it. Pressing a single button shares the 'driving event' on the owner's designated storage area, from where it can be shared on social media.

While you can buy the Dride Zero product from **dride.io**, there's also the option of creating your own Dride Zero with a Raspberry Pi. Both versions can be connected to Android and iOS apps. The startup company actively sought community feedback during development, and community members contributed code to the dash-cam's OS.

DrideOS uses a customised version of Raspbian Jessie. Originally, the commercial version of Dride Zero was based on Raspberry Pi Zero – and was intended to link up with Alexa which, back in 2016 when the project began, was being fêted for its in-car possibilities. Alexa has since been booted out.

You can make your own dash-cam powered by a Raspberry Pi Zero W

Quick **FACTS**

> Yossi's Linux experience made Raspberry Pi the obvious choice for the project

> The original Dride Zero product was based on Raspberry Pi Zero W

> Yossi hand-soldered every one of the first 150 Dride Zero units

> The original crowdfunding video rather goofy and features a horse (**magpi.cc/yCyAuH**)

> A Dride Modem that connects a dash-cam directly to the cloud is due to launch in early 2020

❝ The Dride Zero can be connected to Android and iOS apps ❞

Neiman cheerfully admits that his ten years in software development working primarily in Linux didn't fully prepare him for the world of hardware manufacturing. He soon realised that with hardware you need to be "a million times more focused" than you do with software. And sometimes there are things you just can't fix.

Dride on time
We first covered Dride back in 2017, when it was an Alexa project (**magpi.cc/yCyAuH**). We're pleased to see how far it's come on.
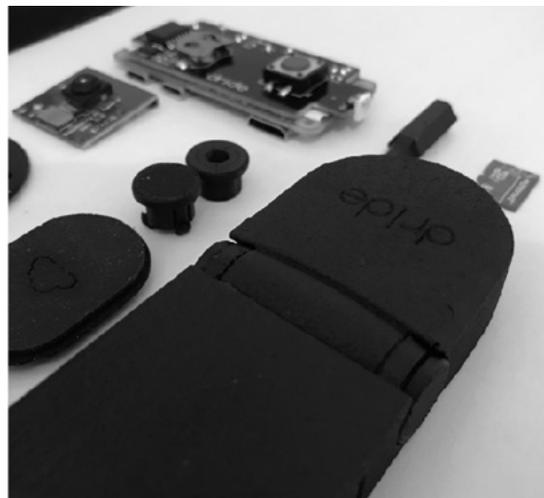
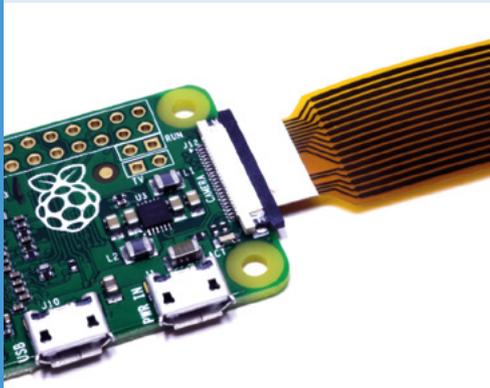▲ The Dride Kit enables you to build your own dash-cam

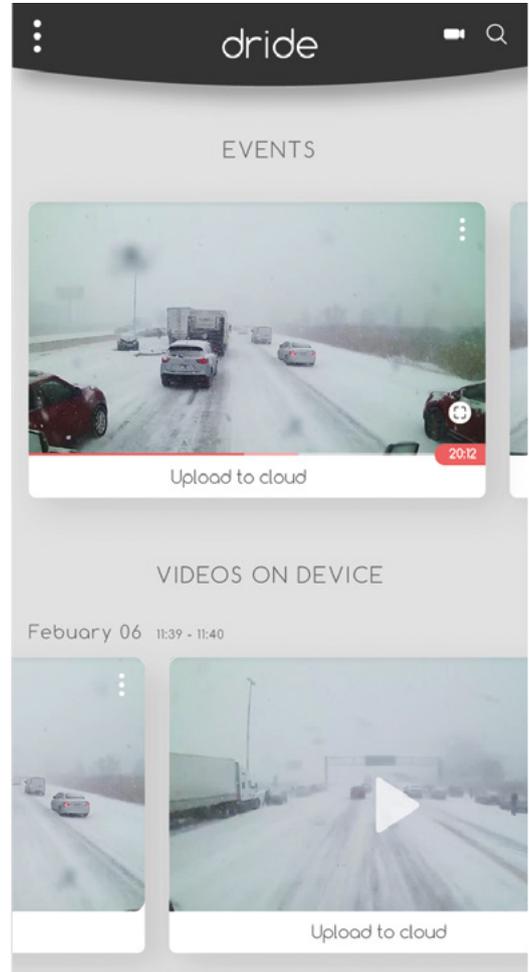## Share your ride

01 Install the DrideOS to a microSD card using Etcher to flash the image file.

02 Connect your Raspberry Pi Camera Module to Raspberry Pi 3 or Raspberry Pi Zero W with the blue and white strip facing upwards.
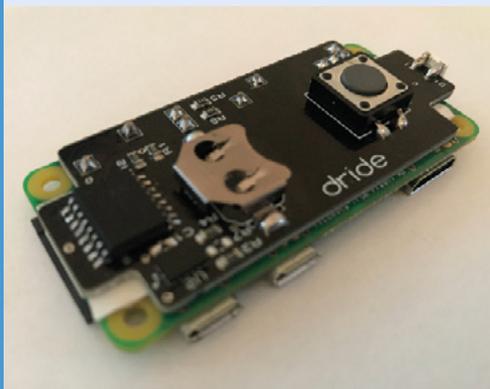
03 Solder the Dride HAT to Raspberry Pi and insert the unit into your 3D-printed Dride case.

▶ The Dride Universal app enables dash-cam footage to be shared by other dash-cam models

> **"** These days the units are built at a manufacturing plant in China before being shipped worldwide **"**

Feedback about the original prototype during the crowdfunding phase focused on the limitations of the video resolution: 4 fps just wasn't going to cut it. "There are zero things you can do to fix frame rate," says Yossi. However, the launch of Raspberry Pi Zero W in February 2017 changed everything. "A few days after the launch of Zero W it clicked," he says. "We need Zero W and we need to make this as fast as possible."

The wireless LAN-enabled Dride Zero with a sharing capability has proved a success and dovetails neatly with the Dride Universal iPhone and Android app, which can be paired with around 16 different models of dash-cam so they too can share video clips of near misses and inattentive drivers.

Yossi particularly recalls the first 150 Dride HATS: "I soldered every single one myself." These days the units are built at a manufacturing plant in China before being shipped worldwide. Customers still buy as backers on the crowdfunding platform, choosing between an £80 Dride Zero with a single 1.4MP camera or the £121 model that also has a rear camera. However, paying is not the only option…

Pimp your Dride

You can 3D-print and make your own Dride Zero with Raspberry Pi. Full details of Dride's OS, API, and how to SSH into the system can be found on the documentation page (**magpi.cc/Dtiapg**). To create your own Dride Zero, you'll need a Raspberry Pi Zero or Raspberry Pi 3 and a Dride HAT (**magpi.cc/hQrxNc**). Download the DrideOS and use Etcher to write it to an up to 32GB microSD card. You'll need to 3D-print your own case using the GitHub file (**magpi.cc/pkqydY**) – a boon if you're keen to avoid the standard black case of the original Dride Zero. At some stage, Dride may sell the case separately. M
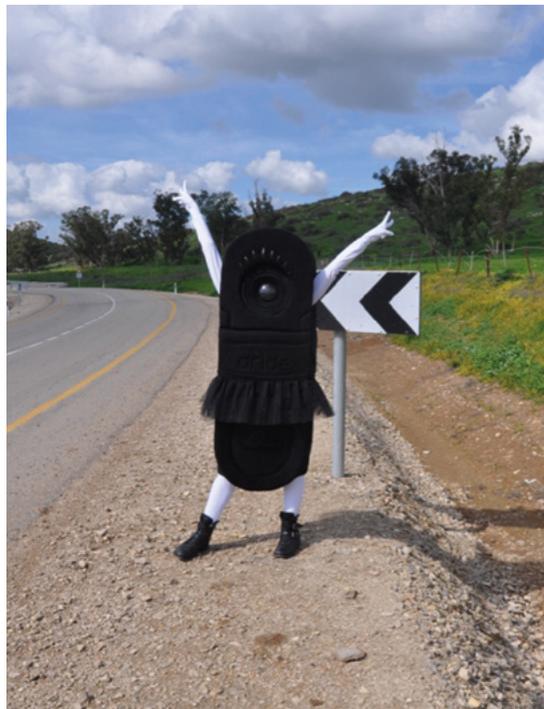
▲ With the Dride Zero dash-cam on your windscreen, you can record and share video footage

◄ The startup company created a mascot to get the message across

Warning!

Dash-cams are legal on UK roads, but you must ensure you have a clear field of vision when driving. A dash-cam must be fitted behind the rear-view mirror so it's in the centre of your windscreen, but not obtrusive to your line of sight.

magpi.cc/MzBaGy

Stonehenge
Skyscape

You can see how the skies above Stonehenge affect the iconic stones via a web browser thanks to a Raspberry Pi computer, as **David Crookes** discovers

MAKER

Mark Griffiths

Mark is the owner of The Bespoke Pixel which operates from Fareham, Hampshire and has created award-winning digital creations for art, music, and science.

magpi.cc/uZGDZz

Stonehenge is Britain's greatest monument and it currently attracts more than 1.5 million visitors each year. It's possible to walk around the iconic stone circle and visit the Neolithic houses outside the visitor centre. Yet, worries about potential damage have forced preservationists to limit access.

With that in mind, Eric Winbolt, Interim Head of Digital/Innovation at English Heritage, had a brainwave. "We decided to give people an idea of what it's like to see the sunrise and sunset within the circle, and allow them to enjoy the skies over Stonehenge in real-time without actually stepping inside," he explains.

This could have been achieved by permanently positioning a camera within the stone circle, but this was ruled out for fear of being too intrusive. Instead, Eric and developers from The Bespoke Pixel agency snapped a single panoramic shot of the circle's

This Raspberry Pi was also fitted with a Pimoroni Enviro pHAT containing atmospheric, air pressure, and light sensors. Captured light values from the sky image were then used to alter the colour values of the master image of the stones so that the light on Stonehenge, as seen via the web, reflected the ambient light of the sky.

What can you see?

"What it does is give a view of the stones as it looks right now, or at least within a few minutes," says Eric. "It also means the effect doesn't look like two images simply Photoshopped together."

> ❝ What it does is give a live view of the stones as it looks right now, or at least within a few minutes ❞

interior using a large 8K high-res, 360-degree camera when the shadows and light were quite neutral.

"We then took the sky out of the image with the aim of capturing an approximation of the view without impacting on the actual stones themselves," Eric says.

Stone me

By taking a separate hemispherical snapshot of the sky from a nearby position and merging it with the master photograph of the stones, the team discovered they could create a near real-time effect for online visitors. They used an off-the-shelf, upwards-pointing, 220-degree fish-eye lens camera connected to a Raspberry Pi 3 Model A+ computer, taking images once every four minutes.

Indeed, coder Mark Griffiths says the magic all runs from Node.js. "It uses a Python shell to get the sensor data and integrates with Amazon's AWS and an IoT messaging service called DweetPro to tie all the events together," he adds.

There was also a lot of experimentation. "We used the HAT via the I^2C connectors so that we could mount it away from the main board to get better temperature readings," says Mark, "We also tried a number of experiments with different cameras, lenses, and connections and it became clear that just connecting the camera via USB didn't allow access to the full functionality and resolutions."

Mark reverse-engineered the camera's WiFi connection and binary protocol to work out how to

By day, you see Stonehenge as it would appear to real-life visitors; at night, a computer-generated model of the sky is depicted with the live location of the stars and the five visible planets

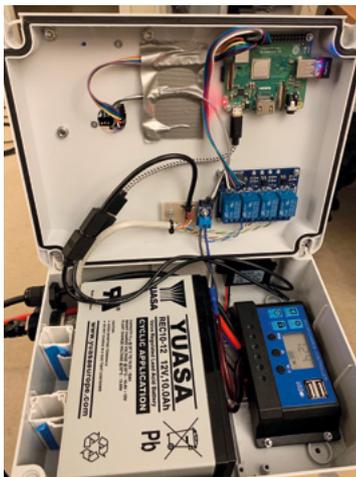**5:21** SUNSET 21:01:29

⊙ Ambient   ☾ Skyscape   ☰ Tour

Placed within a protective acrylic dome, the camera takes an image of the sky every four minutes; this is blended with the Stonehenge photo on the website

An ADSL line allows the sky images and the HAT data to be sent to the cloud, to an application called Stonehenge Skyscape which drives the website

A Raspberry Pi inside this box gathers data about the sky from the camera and the atmospheric, air pressure, and light sensors on the HAT and sends it to an application in the cloud

communicate with it via Raspberry Pi so that full-quality images could be taken and downloaded. "We also found the camera's WiFi connection would time out after several days," reveals Mark, "so we had to use a relay board connected via the GPIO pins."

With such issues resolved, the team then created an easy-to-use online interface that lets users click boxes and see the view over the past 24 hours. They also added a computer model to depict the night sky.

"Visitors can go to the website day and night and allow the tool to pan around Stonehenge or pause it and pan manually, viewing the stones as they would be at the time of visiting," Eric says. "It can look especially good on a smart television. It's very relaxing." ◾

◀ To protect the equipment, a Raspberry Pi computer is safely housed inside a weatherproof box attached to a wall of Stonehenge's security hub

▶ The setup is powered by solar panels (what else?) and it is hooked up with an ADSL line so that the sky images and HAT data can be sent to the cloud

Quick **FACTS**

> The hardware cost around £500 in total

> Marzipano was used for the 360-degree viewer

> Open-source projects including Node.js are utilised

> Suncalc2 and Ephemeris provide astronomical data

> The site had 80,000 unique visitors on its first day

Magic Wand with Computer Vision

What wizarding fan wouldn't want a wand that actually, really, properly makes something move? **Nicola King** explores a simply magical idea

MAKER

Jasmeet Singh

An electronics student from New Delhi, Jasmeet enjoys expanding his knowledge with hands-on projects. He also has a keen interest in the field of AI, and wants to bring the cool tech of movies to real life.

magpi.cc/MsBprC

Following a trip to Universal Studios (Japan) by his brother, who was able to perform some 'magical' tricks with a retroreflector-tipped wand while he was there, maker Jasmeet Singh was inspired to explore what he could do with a replica wand. "I have been highly inspired by tech portrayed in sci-fi movies – be it some cool gadget owned by a superhero, or wands that can cast incredible charms," he reveals. "That was the time I was exploring the field of computer vision and machine learning. Hence, I saw this as just the right opportunity to use the wand and create my own version of the 'wizarding experience' using computer vision and some basic electronics."

A wave of the wand

Jasmeet's project involves creating a motion capture system which enables him to open a box with a flick of his wand. To achieve this, he used a night-vision camera (Raspberry Pi NoIR Camera Module), a Raspberry Pi, some electronics, a Python program using OpenCV, and machine learning.

▶ Jasmeet used this original Universal Studios wand, but you can make your own
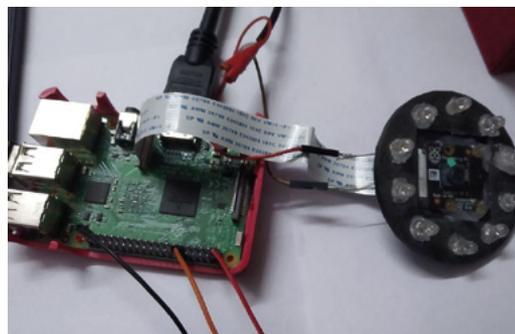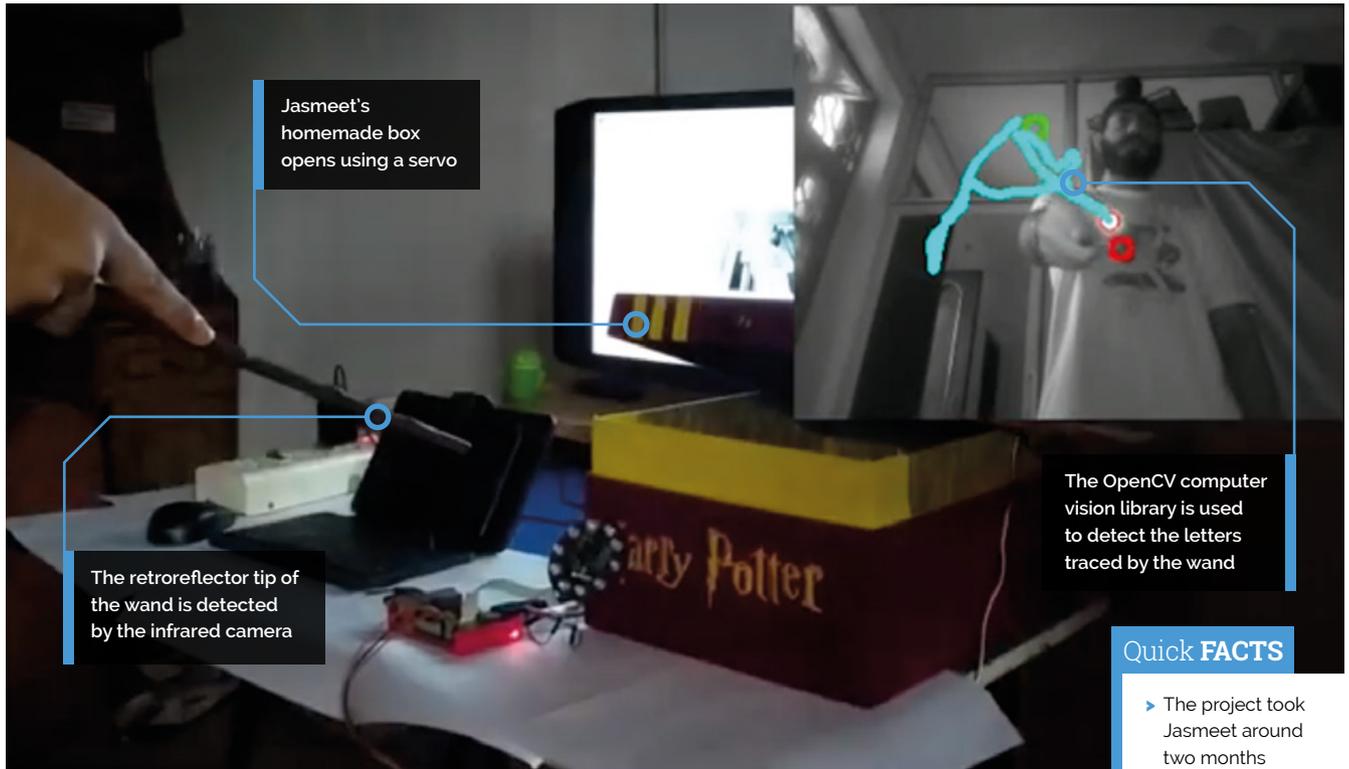

▲ The project's Raspberry Pi NoIR Camera Module is surrounded by a ring of infrared LEDs

So, how does it work exactly? Jasmeet elaborates, "My main goal for this project was to give its user a real wizarding experience with a wand, and that could only be achieved by attaching as little electronics as possible on the wand. The wands used at Universal Studios achieve this by using a retroreflector attached at the tip of the wand."

Jasmeet continues, "When the retroreflector at the tip of the wand is illuminated using an infrared light source, it can be seen as a bright circular blob in the real-time video from the [Raspberry Pi NoIR Camera Module]. Then, using a Python script running on Raspberry Pi with some basic functions of the OpenCV computer vision module, the path taken by the wand is traced after the tip of the wand's blob has entered a starting point in the frame. The person now has to draw the letter, which ends at the ending point, and then the last frame with the drawn letter is passed on to a trained machine-learning model which predicts the letter drawn. Depending on the prediction, Raspberry Pi commands a servo to open/close a box."

Stellar speller

Wizarding aficionados will be aware that 'Alohomora' is a spell that allows the opening of

Jasmeet's homemade box opens using a servo

The retroreflector tip of the wand is detected by the infrared camera

The OpenCV computer vision library is used to detect the letters traced by the wand

Quick **FACTS**

> The project took Jasmeet around two months to complete

> It was his first using Raspberry Pi

> You can use any stick with retroreflector tape or paint on the end

> Jasmeet plans to add a wizard duelling capability…

> …with a second wand and night-vision headsets!

locks. So, appropriately, if someone draws a letter 'A', Raspberry Pi commands the servo to open the box. By the same token, if someone draws a letter 'C' (for close), the box will close.

Jasmeet used a dataset from Kaggle to train the machine-learning algorithm and, as it contained a large number of images for each letter, the accuracy of predictions is an impressive 99%. In addition, he says, "Since the setup of the project uses a night-vision camera for detecting the wand in real time, it can work without any errors in a darkened room, as well as with an additional source of visible light."

Unsurprisingly, the feedback Jasmeet has received from friends and family has been very positive, as has the response from the Instructables and Hackster platforms where his written tutorial is available. All in all, it's a very fun and inspiring project, especially when you consider that Jasmeet was completely new to the fields of computer vision and machine learning when he started his wand-waving idea. Proof that having a go at something new can really pay dividends – "Project-based learning is the best and most efficient way to get yourself acquainted with new technologies," he tells us. **M**

> I saw this as just the right opportunity to create my own 'wizarding experience' using computer vision and some basic electronics **"**

◀ Jasmeet printed out images to stick on an old box – use your creativity

Night Vision

Unlike eating carrots, this ingenious infrared camera really does enable you to see in the dark. **Phil King** stops munching veg to take a look

MAKER

Dan Aldred

Raspberry Pi enthusiast, NCCE facilitator, teacher, and coder who enjoys creating new projects and hacks to inspire others to start learning. Still trying to get a Kinect 360, Python, and Raspberry Pi working with each other.

magpi.cc/rwepTs

It's late at night and all the lights are out in the house, but someone is sneaking around… Don't worry, it's just Dan Aldred testing out his latest Raspberry Pi-powered invention, a portable infrared camera enabling him to see in total darkness.

As a child, Dan was a fan of the Splinter Cell stealth shooter video game. "The iconic image from the game is the night-vision goggles that Sam Fisher wears," he recalls. "I have always been fascinated by the idea that you can see in the dark, and this formed the foundation of my idea to build a portable handheld night vision piece of equipment."
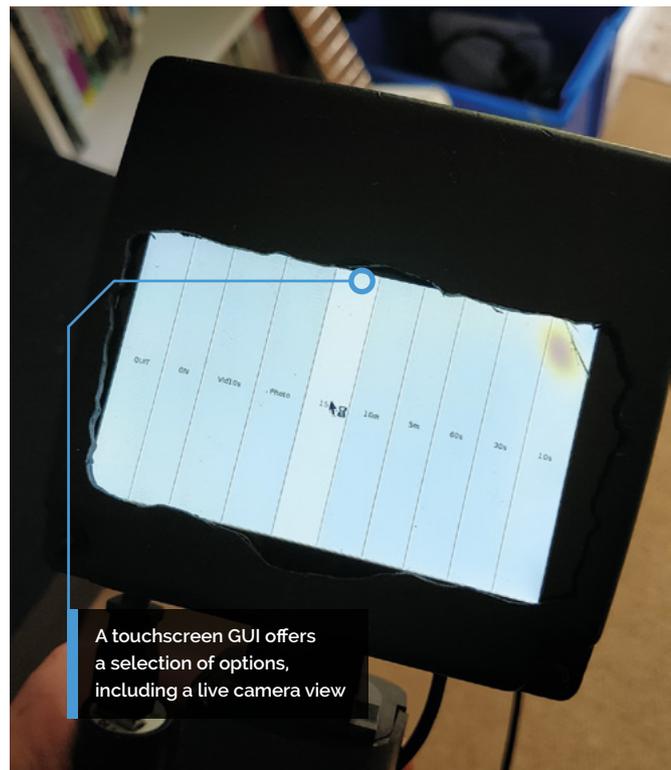
Success by stealth

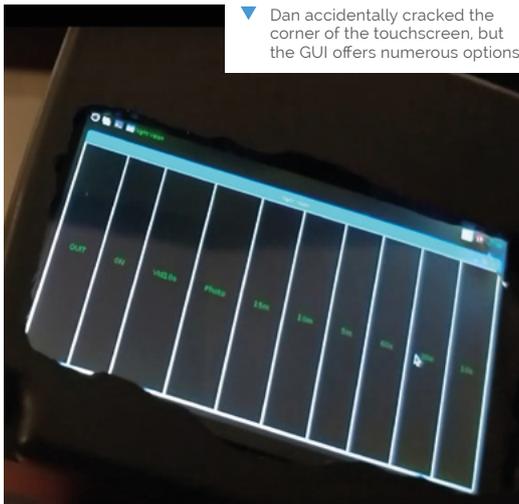Sam's Night Vision project features a Raspberry Pi Zero and a ZeroCam infrared camera equipped with IR LEDs for night-time 'illumination'. The live camera view can be shown on a Pimoroni HyperPixel 4-inch TFT touchscreen, which is also used for a graphical user interface.

For the housing, Dan upcycled a handheld camera flash unit. "The original plan was to use an old-fashioned camera and make a camera that took photos in the dark, but when I went to a car boot sale, all I found was the pivoting flash – it cost a pound, which was a bargain."

As well as being large enough to house a Raspberry Pi Zero, IR camera, and screen, the flash unit has the bonus of a tilting handle, enabling the user to angle the camera to the desired position.

One of the hardest challenges was that Dan wanted Night Vision to be standalone, portable, and bootable, with minimum interaction from the user to get it running. "The obvious solution was

The camera is housed in an old flash unit with a pivoting handle

A ZeroCam infrared camera with IR LEDs can see in the dark

A touchscreen GUI offers a selection of options, including a live camera view

▼ Dan accidentally cracked the corner of the touchscreen, but the GUI offers numerous options

4. jpg

0. jpg

▲ Photos taken by the camera in the dark, revealing a treasure trove of Raspberry Pi goodies

to use a crontab, but this did not load the GUI," he reveals. "I spent about two weeks trying to find a solution. In the end I opted for a desktop icon which when you click it the program runs and the GUI loads."
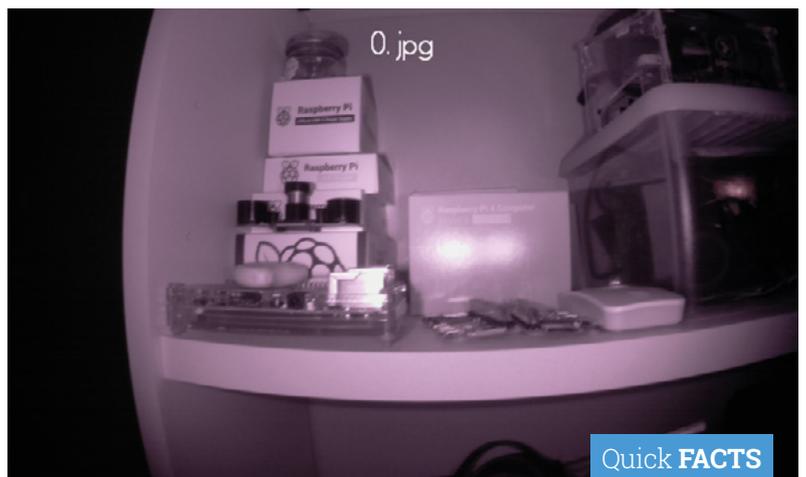
> ❝ The flash unit has the bonus of a tilting handle, enabling the user to angle the camera ❞

From vision to reality

While Dan planned out what he needed to do before he started putting the camera together, not everything went to plan. "There were several points at which I had to take a break and reassess how to find a solution to a problem," he admits.

"All was going well until I ran the program and realised that the camera image displayed back on the HyperPixel was upside down! Luckily at this point I was using elastic bands to hold the case and hardware together, so I could flip it around. But the IR LEDs were secured in, and breaking the Sugru might damage or split the plastic. As a stroke of fate, the IR LEDs are mounted into holders and guess what, you can pop the LEDs out of the holder. So, I popped them out and then rotated the camera unit and then popped it back into place!"

More trouble was to follow when Dan accidentally cracked the touchscreen. "The whole project was working and perfect, so I was left with the conundrum of either replacing the screen and risking all the inner hardware not fitting back in, or just 'make do and mend'. I settled for the latter, so the ten-second button does not work, but who wants to see in the dark for only ten seconds anyway!" Ⓜ

◄ The ZeroCam and its IR LEDs were mounted in drilled holes of a piece of clear plastic and secured with Sugru
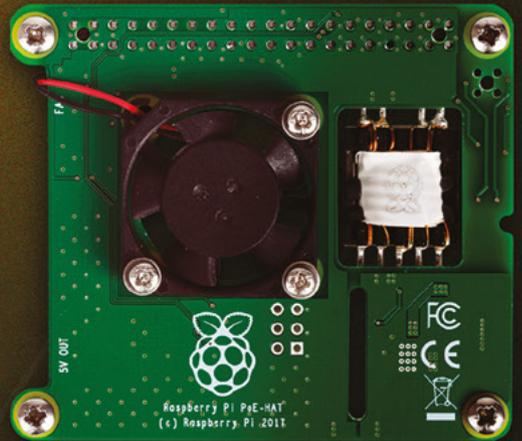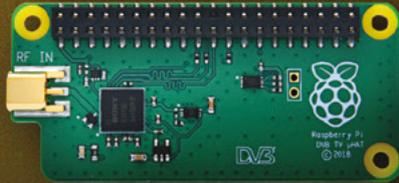
HEXIWEAR

MK20DX256
VLH7
3N64B
CTGR1738G

GND
VCC
RESET WES

Neo.Cortec
FeatherWing — rev.1

USB21

PPT
.PSF-1639

LATTEPANDA
2G RAM 32G eMMC
www.lattepanda.com
FCCID:2AIDMLPDFR0418

Movidius™
Neural Compute Stick

okdo
OFFICIAL
MANUFACTURER
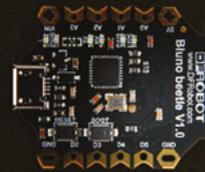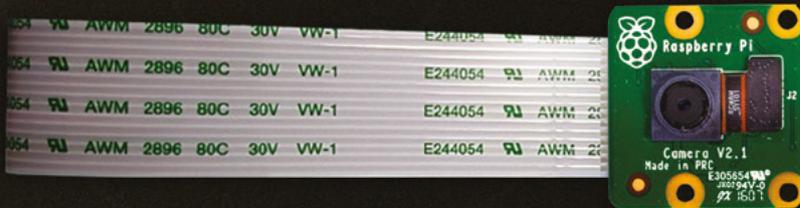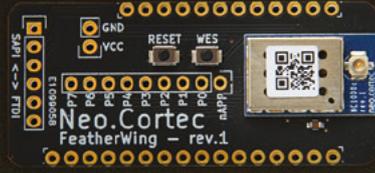
POWERFUL RASPBERRY PI PROJECTS

PJ EVANS MEETS THE MAKERS TAKING RASPBERRY PI PROJECTS TO THE NEXT LEVEL

I t's common to do a bit of head-scratching when you get your first Raspberry Pi. Sure, it's a cool bit of technology at an amazing price, but what to do with it?

Here at *The MagPi*, we work hard to stuff your imagination full of ideas that are hopefully accessible to all our readers. Still, we thought it might be fun to take a look at what happens when a maker gets ambitious and decides to go all-out on their new project. We present some of our favourite advanced Raspberry Pi makes…
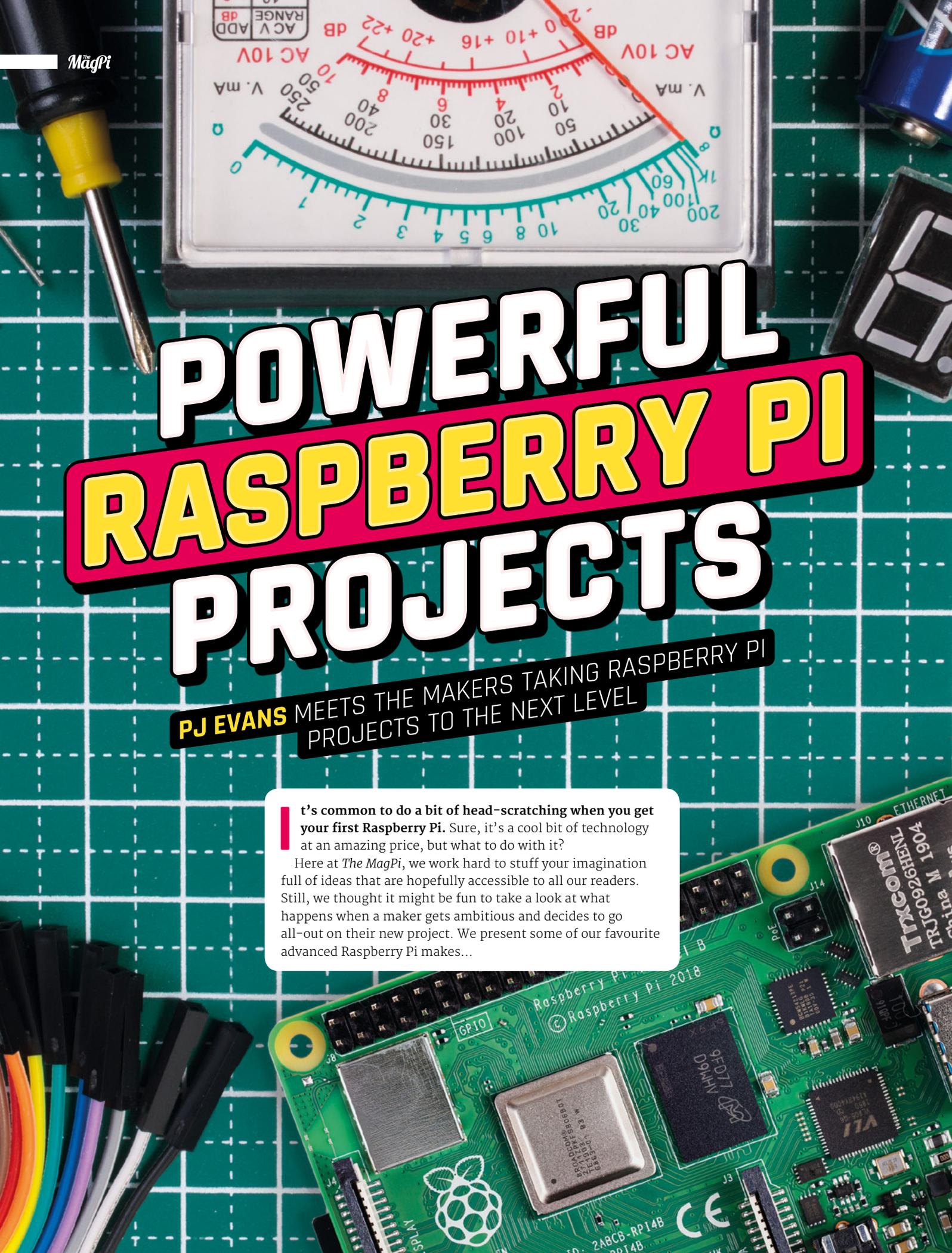
PHOTO PROJECTS

Take amazing shots with Raspberry Pi

DROP PI

Maker **David Hunt** | **magpi.cc/fvNmeA**

Take incredible shots of water droplets, using a Raspberry Pi as a controller for a solenoid valve and camera trigger. The valve is hooked up to the GPIO pins and a small piece of code opens the valve and triggers the camera. The code is timed for a valve 40 cm above the surface of the water. It's a great example of how Raspberry Pi can be used to control an environment and camera, plus a good excuse to learn how to control valves.

STEREOPI

Maker **Eugene Pomazov** | **magpi.cc/pGvBAp**

Since 2014, Raspbian has offered built-in support for stereoscopic photography. With two cameras attached to a Raspberry Pi, you can create 3D photographs and record 3D video. You'll need a Raspberry Pi Compute Module (which has support for two Camera Modules). The compact and light nature of StereoPi makes it particularly useful for attaching to drones and robots.
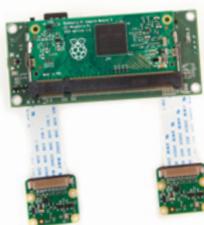
ADVANCED GAMING BUILDS

Strictly for fun, these masterful makes put a smile on your face

HEVERLEE SJOELEN

Maker **Grant Gibson** | **magpi.cc/DQiGwQ**

When Belgium beer brand Heverlee approached prolific maker Grant Gibson for promotional ideas, he was reminded of Sjoelen, a shuffleboard game popular in Germany and Belgium. The result was a Raspberry Pi-powered physical game and vending machine mash-up that dispensed cold cans of beer to the winners.

THE CLAW

Maker **Ryan Walmsley** | **magpi.cc/aqtCxF**

Ryan's ever-popular claw machine is often seen at Raspberry Pi events throughout the UK. An upcycled bar-top 'grabber' game, this one can be played over the internet. Use your computer or mobile phone to try to grab Babbage the Bear (gently) as the results are live-streamed to you.

OUTRUN BAR-TOP

Maker **Matt Brailsford** (aka Circuitbeard) | **magpi.cc/TjjVMH**

What separates Matt from the crowd is his exquisite attention to detail. This OutRun Deluxe bar-top features fully working controls, such as gear shifting and a steering wheel. Add the pedals, repurposed from an wheel controller, and custom bodywork and this is a classy project.

CREATIVE PURSUITS

Physical computing and the arts are a very good match, and here's why

ETCH-A-SKETCH ART WITH PYTHON

Maker **Sunny Balasubramanian** | **magpi.cc/UvGhfe**

A classic toy featuring in many people's childhoods, the Etch-a-Sketch allows you to draw pictures without making a mess, then wipe the board with a satisfying shake. Sunny used servos and a Raspberry Pi to take control of the dials, then used edge-finding image filters to create an Etch-a-Sketch 'camera'.
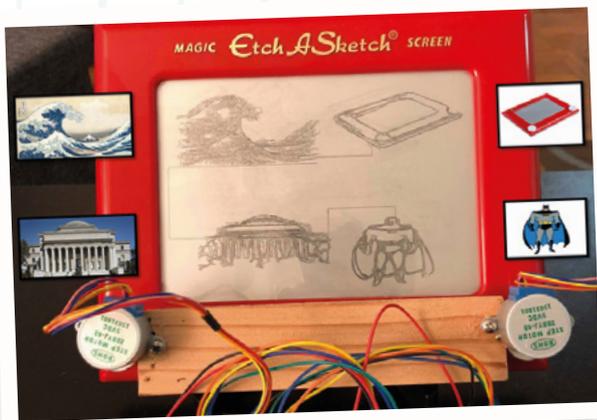
CUBERT

Maker **Lorraine Underwood** | **magpi.cc/dmSCSG**

A beautiful 8×8×8 structure of ping-pong balls, each containing a NeoPixel to create amazing colour patterns, and you can even play 3D Pac-Man. Of particular interest is Lorraine's blog and talks on the project, where she is brutally honest about the difficulties of making such a unique object.
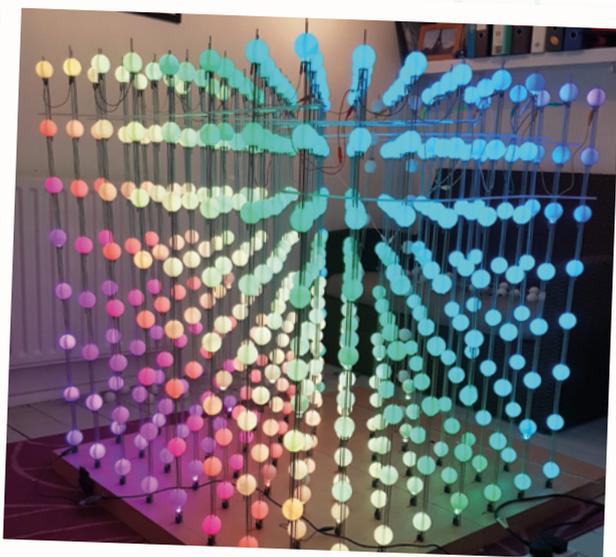
NETWORK KNITTING MACHINE

Maker **Sarah Spencer** | **magpi.cc/iSVNEm**

One of the must-see exhibits at 2018's Electromagnetic Field camp was 'Stargazing' by Sarah Spencer. This map of the universe measures a colossal 4.6 by 2.8 metres and was knitted on a hacked Brother mechanical knitting machine, controlled by an Arduino and Raspberry Pi. It's even networked.

TRUE COMPACT CAMERA

Maker **Martin Parker** | **magpi.cc/dpbFJH**

Raspberry Pi Zero can fit into some tiny places, but how about a vintage 110 format compact camera? Martin replaced the internals with a Raspberry Pi Zero and Camera Module. Don't worry about film running out when your camera can take thousands of high-res snaps.

PI CLOCK 2

Maker **Tim Richardson** | **magpi.cc/dbqYPu**

Tim's clock is made up of two 64×32 pixel displays and displays information being relayed from a Raspberry Pi-controlled weather station. It features some clever energy-saving extras, such as a motion sensor to only update the screen when someone is in the room. It's been to Parliament as part of the one-millionth Raspberry Pi celebrations.

TAKE TO THE SEAS IN YOUR AUTONOMOUS YACHT

MAKER

AI Coventry (Coventry University)

Left to Right: Balazs Bordas, Mark Tyers, Sergiu Harjau, Shahzad Haider.

magpi.cc/uxjfhV

This is a wind-powered craft, so the sail is moved by motor

'The Rabbit', complete with Raspberry Pi Zero

The project's Raspberry Pi Zero is connected to a host of sensors to determine direction and speed

AI Coventry is making serious progress with autonomous vehicle technology. Sergiu Harjau and team entered their aquatic vehicle, 'The Rabbit', in an autonomous boat challenge in China. We asked him all about it.

What inspired you to build a self-sailing boat?

I first started having an interest in autonomous vehicles when I had to choose a project for a second-year module. I first built an autonomous RC Car, driven by Raspberry Pi Zero[…]. That got some traction in the university and then a lecturer offered me a spot on the autonomous boat team in Finland. We use the project as a way to broaden our skill set, both from a software standpoint but also when it comes to electrical engineering, and so far it's been working wonderfully.

What challenges did you face?

Autonomous vehicles are a bit like chess in some ways. It's very easy to understand how it's all meant to work, but it's really hard to go 'deep' and create beautiful systems which work without a single flaw[…]. In China, our biggest challenge which we didn't foresee was the weather. The humidity and extreme heat rendered some of our sensors faulty, spitting out random data at unpredictable times. Even still, we pursued our goals and in the end managed to fix some of the issues and came home with a pretty good result.

Are you happy with the outcome?

Yes, in our latest trip we did way better than our past ones, but even still we weren't perfect. We're very happy to call it a learning experience and go from there. On the flip side, we were very organised, more prepared than any team out there if I'm honest, and that allowed us to quickly fix our issues when we needed to. In the end, we managed to get third prize, and we're very happy with the result.

Any improvements planned?

We're going to be looking at spending a little bit extra on our compass sensor to ensure it doesn't get de-calibrated as often as it did in China. We suspected there were power lines under the lake, and that didn't help our autonomous sailing.

What plans do you have for your next vehicle?

Since autonomous vehicles and embedded systems are two of my favourite pastime activities, my next big project will again bring the two together. I'll be helping my lecturer Dr David Croft to deliver a hardware-software platform for a new master's course next year: 'Connected autonomous vehicles systems'. We're planning on building an RC car with capabilities to become autonomous on an ROS software interface. It's not going to be easy, to say the least, but I hope that through my other projects I have managed to gain the necessary skills to pursue yet another interesting endeavour.

A touch enclosure was required to withstand the rigours of space

All parts are 'commercial off-the-shelf', including the official Raspberry Pi Camera Module

The project's Raspberry Pi Zero required no modifications

TAKE PHOTOS FROM SPACE!

Space is hard. Space, when you're not SpaceX or NASA, is extremely hard, but that didn't stop Surrey Satellites. Having secured a slot on Soyuz, they launched the DoT-1 satellite, which had a Raspberry Pi and camera on board supplied by the University of Surrey. It had a simple task: take a photo from orbit using commercially available off-the-shelf parts. We spoke to Surrey Satellites' Director of Engineering, Rob Goddard.

What inspired this project?
Whilst the primary objective of the DoT-1 (Demonstration of Technology) mission was to fly the company's next-generation avionics, there was space for some additional experimental payloads, hopefully stimulating the interest of our younger engineers. One of those experiments, designed and implemented in conjunction with the University of Surrey Space Centre, was to

> " There could be some credible applications for low-cost computers and cameras of this type "

capture an image from space using a commercial-grade Raspberry Pi Zero computer and camera, store the data, and downlink it via a new data handling system on board the satellite.

What challenges did you face?
It was a surprisingly easy project! We performed some screening tests on three Raspberry Pi Zero computers to select the best performing over-temperature, and then packaged the computer and camera into a metal box. The standard camera lens was changed for a fish-eye lens. The remaining electronics were completely untouched.

Are you happy with the result?
We were certainly pleasantly surprised by the quality of both the still imagery and video capture from Raspberry PI Zero and camera. There could be some credible applications for low-cost computers and cameras of this type. We're considering flying them as inspection cameras to confirm deployment of solar panels, or to view robotic arm movement.

RADICAL ROBOTS

We couldn't leave our little robotic friends out, now could we?

ROBOT DINOSAURS

Maker **Dr Lucy Rogers** | **magpi.cc/QPHdBc**

It's not often dinosaurs are disappointing, but when the animatronic dinosaurs of the Isle of Wight theme park Blackgang Chine kept failing, Lucy came to the rescue. Retrofitting the bespoke mechanics with off-the-shelf parts controlled by a Raspberry Pi, she made them easier and cheaper to maintain, with more movement options.

SOCCERBOTS

Maker **Neil Lambeth** | **redrobotics.co.uk**

Neil wanted to show kids that robotics can be more than just a single 'bot moving around, so he developed the SoccerBots, a pair of remote-controlled, ball-firing robots that try to score goals against each other. They've inspired hundreds of kids around the UK.

BIOHEX

Maker **Harry Brenton** | **magpi.cc/eynBqs**

Hydroponics is all the rage. Growing plants without soil produces amazing results, but requires careful monitoring and care. Harry's BioHex is a 3D-printed modular plant-growing machine that uses a Raspberry Pi to analyse the environment, operate the air pump, and provide lighting control.

OPEN-SOURCE MARS ROVER

Maker **NASA Jet Propulsion Laboratory** | **magpi.cc/yhXYKp**

Want your own Mars Rover? Of course, you do. After a successful educational outreach programme demonstrating a small version of its real rovers, NASA's JPL created a new robot, ROVE-E, that is made of off-the-shelf parts and runs open-source software on a Raspberry Pi. It costs around $2500 to build.

INCREDIBLE AI PROJECTS

Build something super-smart with Raspberry Pi

GOOGLE CORAL

Maker **Google** | **magpi.cc/coral**

Coral is a range of AI products and projects made by Google. The latest star is the USB Accelerator. This dongle adds a Google TPU to Raspberry Pi, which rapidly boosts real-time classification – all local on the Raspberry Pi. Check out Teachable Machine in *The MagPi* 79 (**magpi.cc/79**).

TENSORFLOW CUCUMBER SORTER

Maker **Makoto Koike** | **magpi.cc/fDeLLK**

It's lovely to see one maker put AI tech into practice to solve a problem. Makato's father grows cucumbers and straight ones with lots of prickles command a high price. Makato trained an AI robot to spot and sort them. Perhaps there are a number of production lines that could benefit from a little AI.

THERE'S WALDO

Maker **Matt Reed** | **magpi.cc/LaGxbn**

Where's Wally? (Or Waldo as he's known in the US.) Matt Reed's There's Waldo robot is an irreverent demonstration of machine-learning. There's Waldo uses OpenCV to extract all the faces from the page, and then sends them to Google Auto ML Vision service. This locates the striped one and sends it back to a Raspberry Pi, which points him out using a uArm metal arm.
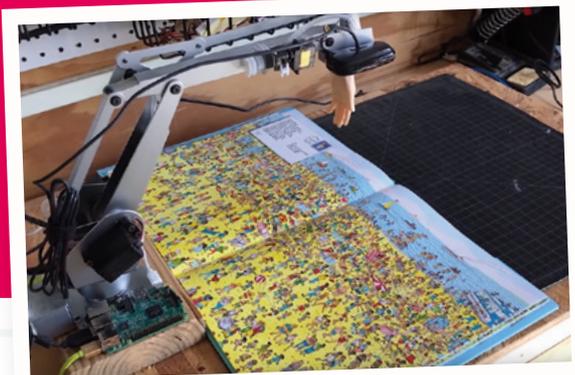
WHOA! WHAT IS THAT?

Some projects are just so 'out there', they defy categorisation

HIGH-ALTITUDE BALLOONING

Maker **Dave Akerman** | **daveakerman.com**

The world's best computer is no stranger to space, often hitching a lift on the ISS. If you'd like to send a Raspberry Pi skyward, it's easier than you think. Dave sends Raspberry Pi Zero computers up to 100,000 ft (30,480 m) in the air using helium balloons, taking amazing photos. His comprehensive blog shares an immense amount of knowledge.

Warning!
High Voltage

This project uses potentially dangerous levels of electricity

MUSICAL TESLA COIL

Maker **Derek Woodroffe** | **magpi.cc/WhJQRq**

A Tesla coil as a musical instrument? Why not? Derek is famous in the community for his high-voltage antics. Here, a Raspberry Pi Zero can load MIDI files and convert each note to information that is sent to two coil driver boards. The result is your favourite tune rendered at 200 volts.

3D SCANNER

Maker **Richard Garsthagen** | **pi3dscan.com**

This project gets a special place on our 'Whoa!' list simply for 98 Raspberry Pi computers in use. Split over 19 poles, they provide a high-resolution 3D model of anything placed within the scanning area. Richard has not only written a great series of posts as the project has evolved, but has open-sourced many of the plans and code used.

FLAPPY BRAIN

Maker **Albert Hickey** | **magpi.cc/jhspVm**

Mattel's 'Mind Flex' toy appears to read your mind by measuring your brain wave activity. Egham Jam organiser Albert Hickey added an Arduino and Raspberry Pi to decode the output so you can play Flappy Bird. Move the bird down by thinking hard, then clear your mind to go up!


FLOOD NETWORK SENSOR

Maker **Ben Ward** | **flood.network**

Citizen-scientist Ben Ward has invented a cheap, serviceable alternative to expensive flood-monitoring systems for his home town of Oxford, using Raspberry Pi computers. The system bounces sound waves off the water surface to calculate the level. The reading is then relayed by radio (LoRa) to a central database. The project is now spreading across the UK.

SPIN UP A DIGITAL ANIMATION
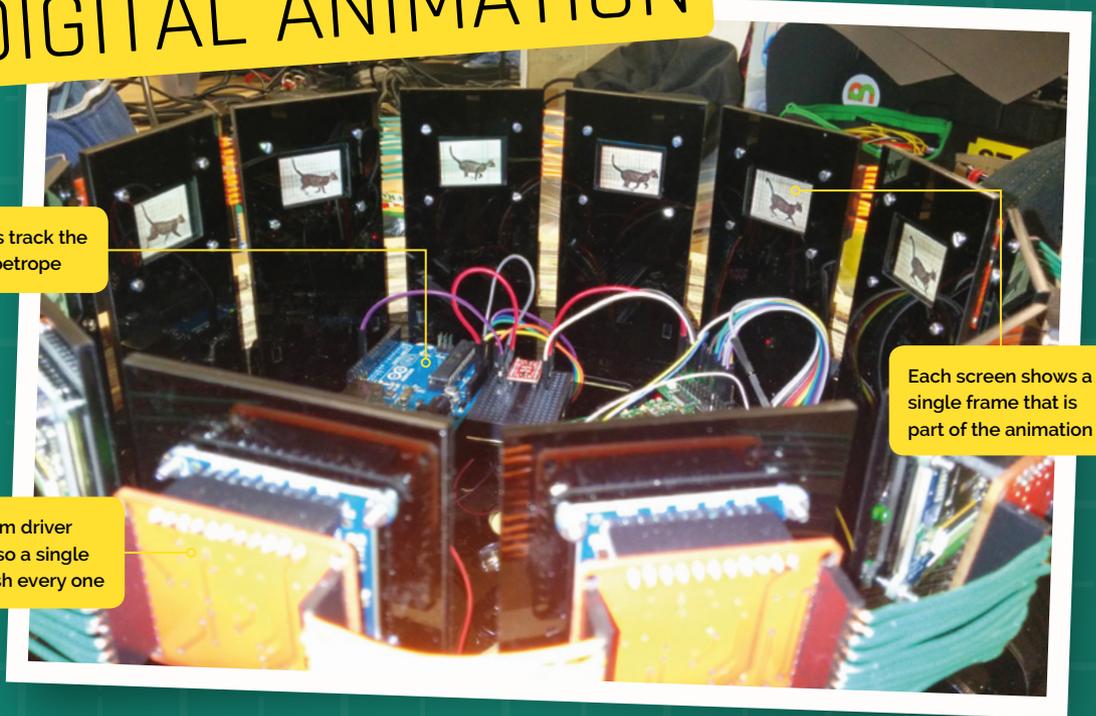
An Arduino helps track the rotation of the zoetrope

Each screen shows a single frame that is part of the animation

Brian designed a custom driver board for each screen so a single Raspberry Pi can refresh every one

MAKER

Brian Corteil

Brian is an award-winning robot maker, active member of the Raspberry Pi community, and self-titled 'head meat-bag' of Coretec Robotics.

magpi.cc/LjJFFO

Brian is a regular at Raspberry Pi events up and down the UK and is most often seen with his Pi Noon balloon-battling robots or FacePlant, the two-wheel balancing creation he entered in this year's Pi Wars. There is another one of his builds that caught our attention, a digital zoetrope (an early animation machine) that threw up some real technical challenges, in particular trying to drive twelve screens from one Raspberry Pi. The images on the screens can be updated in real-time when the zoetrope is spun.

What inspired you to build a zoetrope?

While I was researching the images of Eadweard Muybridge and the history of moving images, I was reminded of the zoetrope. I had a crazy idea that I could make a digital version bringing Eadweard Muybridge's images to life.

What challenges did you face?

I had to control twelve screens on a single Raspberry Pi and design a circuit to be able to select each screen. Then I needed to write software to render images for the screens, modifying the driver software to upload an image in four blocks.

Are you happy with the result?

I'm pleased with the way the Digital Zoetrope turn out after I changed the shiny black acrylic to matt black and the wiring to black. It ended up being as I first imagined it would be.

Any improvements planned?

To make it more interactive and be able to import cells by taking a photo of a hand-drawn sheet of a one-second short film.

Are there any more zoetropes in your future?

Well, I have two projects planned. Hopefully, I will be making a second larger version of the Digital Zoetrope, using e-paper displays for Electromagnetic Field 2020 camp as an art installation, using twelve Pi Zeros and a Raspberry Pi 4 networked together. And of course, my annual robot build for Pi Wars. ◪
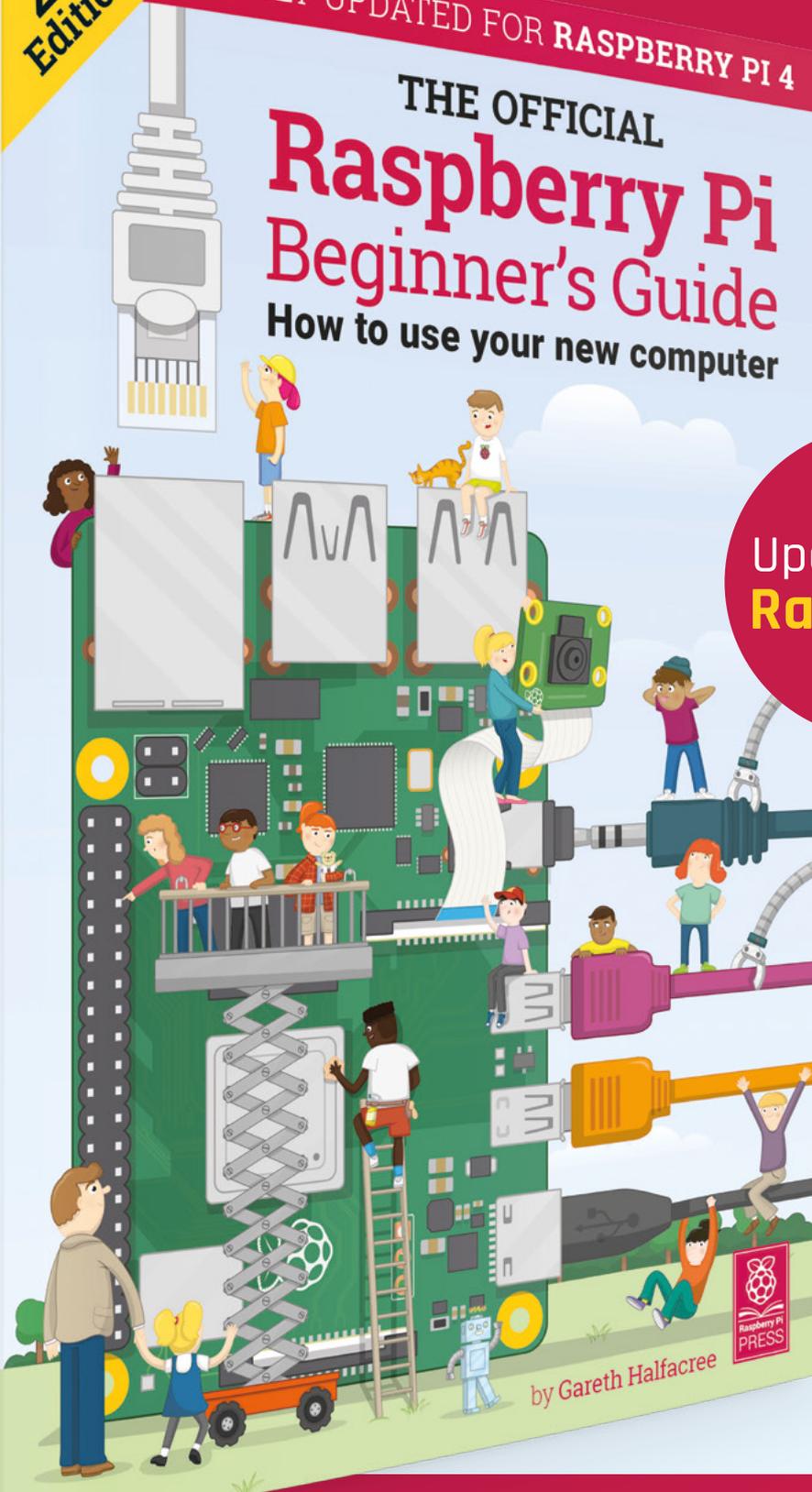
The official **Raspberry Pi** Beginner's Guide

2nd Edition
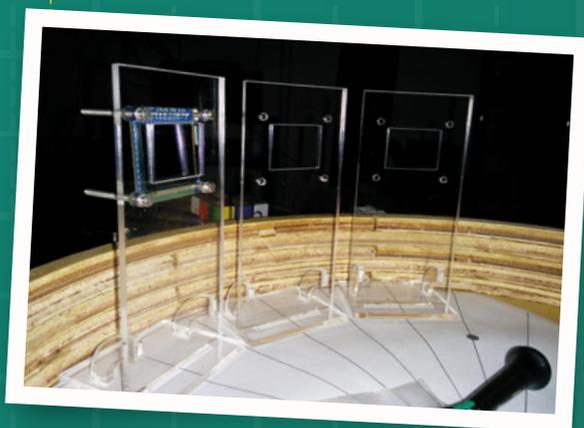
by **Gareth Halfacree**

FULLY UPDATED FOR **RASPBERRY PI 4**
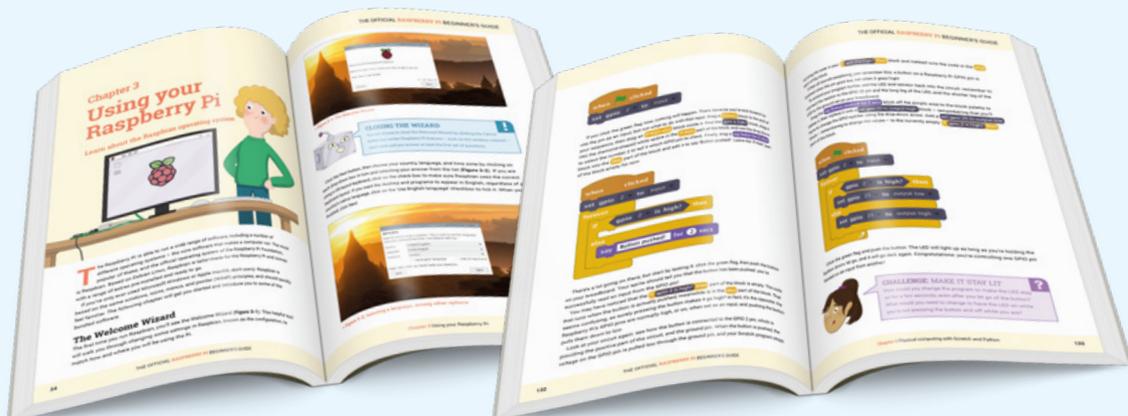
THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer

by Gareth Halfacree

Fully
Updated for
**Raspberry
Pi 4**

Interactive tabletop RPG table

Plunder dungeons and slay dragons in digital style with this simple TV conversion project for all your TTRPG needs

MAKER

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

Warning!
Power Tools

In this tutorial we use various hand and power tools that can be very dangerous. Please seek adult supervision.

You'll Need

> Flatscreen TV

> Roll20 account
  roll20.net

> Planed timber

> Wooden skirting boards

> Some brackets

> Carpentry tools

> Nails

Tabletop RPGs are back in fashion a serious way. Thanks to a recent tidal wave of podcasts, live streams, and YouTube series, games like Dungeons & Dragons, Vampire: The Masquerade, Warhammer 40,000, and other TTRPG systems have never been more popular.

Aside from finding a day all your friends are available, the most difficult part of playing can be handling and juggling different maps. Struggle no longer with paper printouts and upgrade your GM's table with an old TV, a Raspberry Pi, and a bit of ingenuity. Now grab your carpenter's tools and roll a skill check…

01 Choose your Raspberry Pi

Not every Raspberry Pi will be quite suited for this use. The main function of Raspberry Pi here is to launch into a web app from full-screen Chromium. It's not the lightest webpage in the world, so while something like a Raspberry Pi Zero

▶ Mitre boxes are very handy for cutting perfect angles on your wood

might seem like a no-brainer, we'd suggest doing some tests first. We're using a spare Raspberry Pi 3 and it works just fine, and a Raspberry Pi 3A+ would probably do the job as well, even with the reduction in RAM.

If you're buying a brand new Raspberry Pi for the job, go for the 4GB Raspberry Pi 4 so that you have more room to grow with possible future TV upgrades.

02 Set up Roll20

The real magic for this interactive table comes from Roll20 (**roll20.net**) – it's a free online service that is normally used to let people remotely connect to each other for a TTRPG game. As well as being able to talk to your group and GM, you can roll dice within the app and control tokens of your characters.

It also allows a GM to display a map that has a 'fog of war' – areas that the players cannot see. From their computer, they can reveal the map as the players explore. Sign up for an account – if you're the GM, you might want a secondary account to act as the 'player' account for the screen.

03 Initial screen setup

A fresh install of Raspbian is best for the table, we find. If this is your first time using a Raspberry Pi, check **magpi.cc/quickstart** for instructions on how to install Raspbian, including the normal first-time setup stuff like connecting to wireless LAN and setting a password, etc.

Once that's all done, open Chromium and log in to your Roll20 account that will be used for the screen. Go to the My Games section and right-
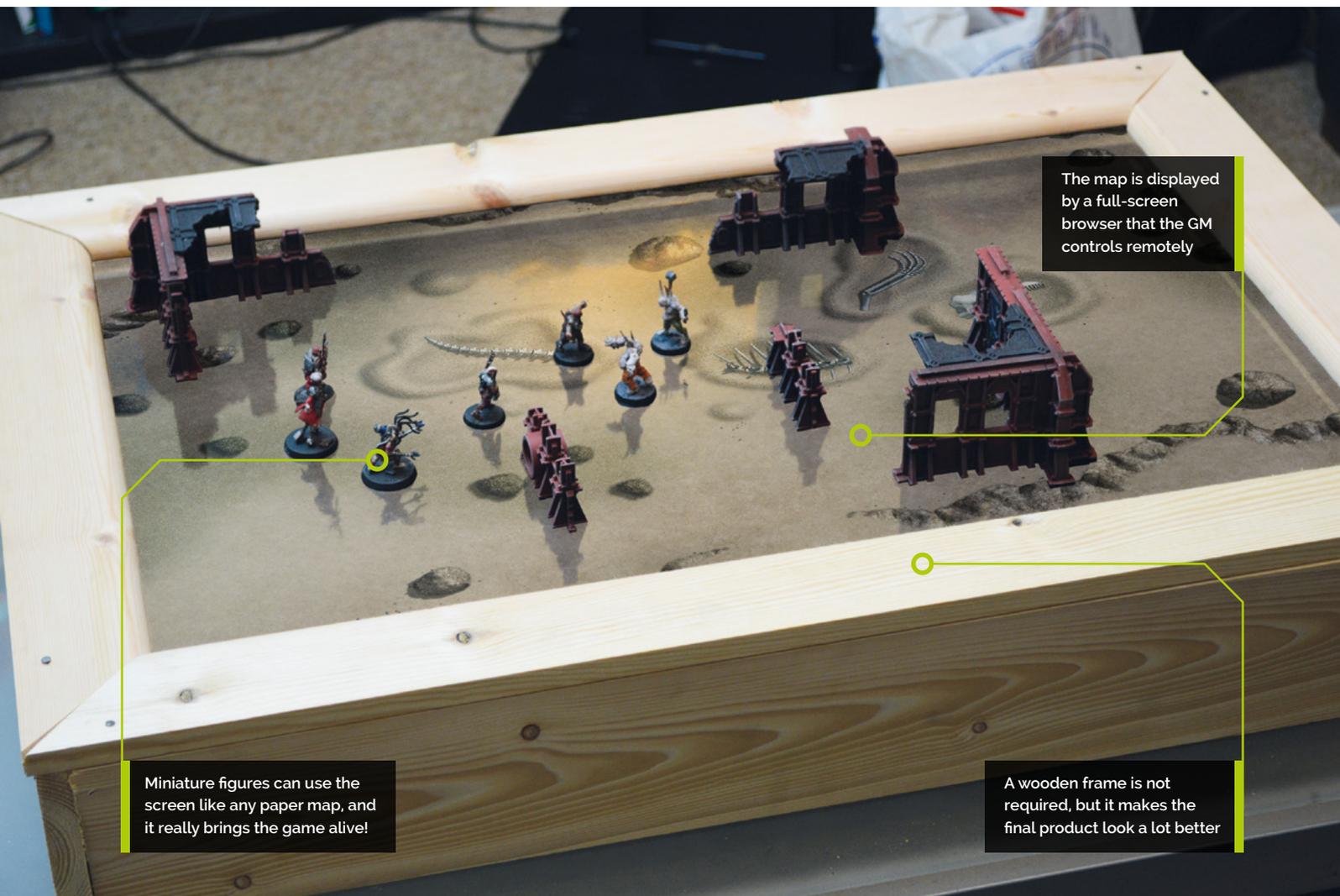
The map is displayed by a full-screen browser that the GM controls remotely

Miniature figures can use the screen like any paper map, and it really brings the game alive!

A wooden frame is not required, but it makes the final product look a lot better

click and 'Copy link address' on the Launch Game link for the campaign you want to show on the screen. Open Chromium's settings, scroll down to 'Open a specific page or pages' (under the 'On start-up' section), and paste the link into there.

04 Raspbian tweaks

To get Chromium to launch at startup, you can add a quick line to the bottom of the profile config file. Open the Terminal and use:
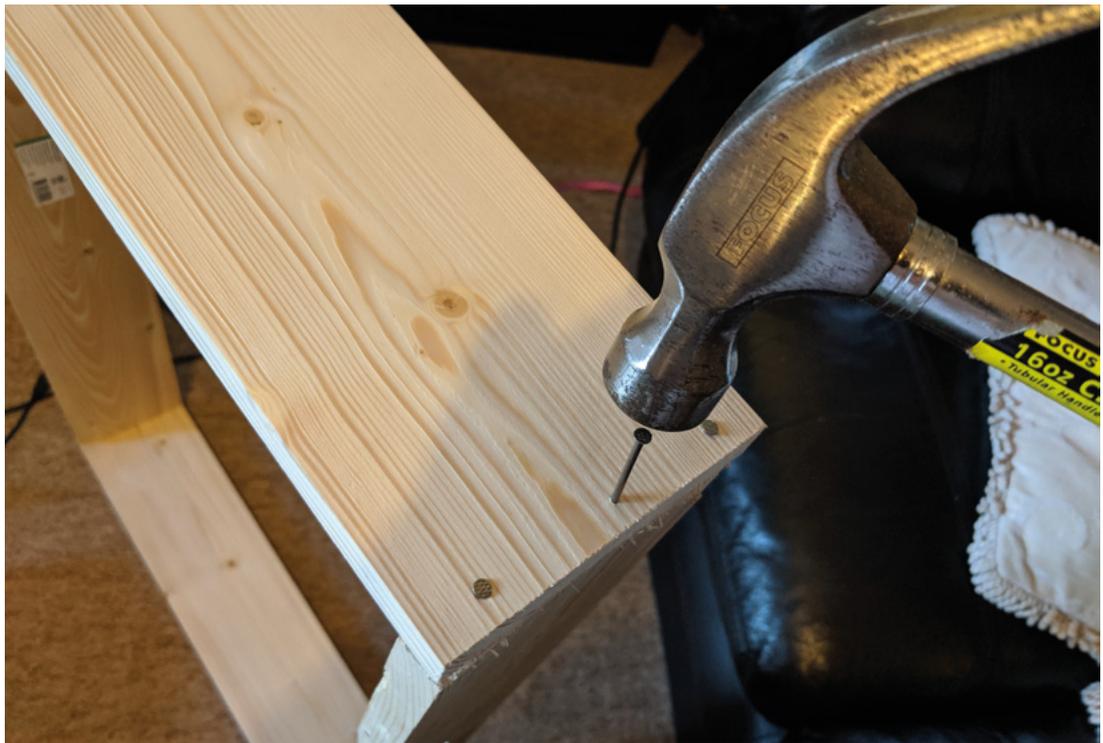
```
sudo nano /etc/profile
```

Use an arrow key to scroll to the bottom of the file and add:

```
chromium-browser --start-fullscreen &&
```

Press **CTRL+X**, then **Y** to save and exit. Now on a reboot, Raspbian will open Chromium in full-screen straight to your map.

▶ Make sure to test your build before trying to make a frame

▶ Be careful while nailing things, and try to make sure it's square

Top Tip 👍

Other software

Don't want to connect to the WiFi in every new location? You can use Dungeon Board, which allows you to power the map from the GM's computer: **magpi.cc/pJdHtp**

We did a bit of customisation of Raspbian as well. You can swap the Raspberry Pi Desktop splash screen by changing the **splash.png** file in **/usr/share/plymouth/themes/pix/** (we used the Dungeons & Dragons ampersand logo) and we also changed the wallpaper on the desktop.

05 Measure your screen

If you want to build a frame, you need to make sure you know how big your TV is. Measure the length and width precisely, as well as the depth and the bezels. For the frame, you want to use planed lengths of timber to create the sides of a sort of box.

Make sure the width of the timber is larger than the depth of the TV. We used 18 mm-deep cuts of wood to try to keep the weight down; however, you may choose thicker.

We used moulded skirting boards for the façade of the frame, meaning we didn't need to create a chamfer on a piece of wood. The width of the skirting should cover the depth of the main frame, as well as the bezel on the TV.

06 Cut your wood

We recommend using a mitre box to make sure your cuts are straight or perfectly angled. We created the frame by cutting the wood for the top

and bottom of the TV to the exact length of each side, and then cut the two side bits of wood to fit their size, plus twice the depth of the wood. This way, the side parts cover the ends of the wood used for the top and bottom.

For the façade, we measured how long each piece of wood would be at the top (side length plus twice the wood depth) and then cut at a 45-degree angle inwards so that the chamfered edge was the shorter edge of the wood.

07 Assemble the frame

This step looks easy, but is a bit tricky. You need to nail the shorter (side) lengths of wood to the longer (top and bottom) lengths of wood to create your basic box-like frame. Use clamps where you can to keep the angle at as close to 90 degrees as possible. We recommend nailing one of the short (side) parts of wood to both of the long pieces (top and bottom), before closing it off with the other side part of wood.

08 TV check

Before going further, make sure your measurements were correct and fit your TV into the box. At this point you might want to make note of any extra protrusions, or drill some holes to access buttons on the television if necessary.

Top Tip 👍

Elevation

One of the problems we came across with paper maps is that they take up valuable table space. So does this screen. A possible solution? Use legs to elevate it and regain some space for character sheets.

If it doesn't fit properly, you may need to take apart the frame and do it again, or make sure you measured correctly in the first place.

09 Assemble the façade

You should probably glue the edges of the façade together to make sure they're all even and nice before attempting to add it to the frame.

We didn't do that, and instead clamped each piece to their specific side and added a few nails to attach them. In our case, we had quite wide pieces, so we had to make sure they didn't dip inwards while we hammered them in. Gluing beforehand should allow you to avoid that.

10 Install the TV

Once the façade is added, turn over the frame and lower your TV in. We screwed in some wide brackets to the sides of the frame to keep the TV in place. This also means you can remove the TV if you need to at some point. Turn it over and admire your handiwork.

11 Final adjustments

You'll need to run a couple of power cables from beneath the frame to power the TV and Raspberry Pi – we used a rotary tool to carve out a little bump that would allow cables (along with a mouse cable) to run freely from under the frame.

Now it's time for the first proper test, so try to get your friends over (if their schedules allow it) and try not to spill any canned drinks onto the new table.

> " You'll need to run a couple of power cables from beneath the frame to power the TV and Raspberry Pi "

12 Further upgrades

The table looks nice, but there are always ways for it to look nicer. You could carve the façade parts to make it look more ornate, and give it a coat of dark varnish to give it a more woody look.

If the bezels on the TV you used aren't particularly deep, you could also add a layer of clear acrylic as protection to the real screen. Finally, maybe add carrying handles if you plan to move it around a lot. M

Make a Scratch 3
Sense HAT game

Discover what's new in Scratch 3 as you build this exciting racing game, using your Raspberry Pi as a steering wheel

MAKER

Sean McManus

Author of *Scratch Programming in Easy Steps, Cool Scratch Projects in Easy Steps,* and *Mission Python.* Get free chapters at Sean's website.

sean.co.uk

Scratch 3 is now on Raspberry Pi, with new blocks for the Sense HAT. In this game, you tilt your Raspberry Pi to turn left and right, and use the HAT's up and down joystick controls as an accelerator. Going off-road or colliding with the computer's racer damages your car. Can you complete three laps before your car is too beaten up to drive? If so, how fast is your qualifying time? This project will give you a tour through what's new in Scratch 3, including computer speech, new sound effects, and the extensions. Rev up!

01 Add extensions to Scratch

One of the best features of Scratch 3 is the extensions, which enable you to add new capabilities to Scratch. For example, you can add blocks for simple electronics, controlling the GPIO, video sensing, and using the Makey Makey input device. To simplify Scratch for first-timers, the Pen and Music blocks that used to be in the Blocks Palette have now been put into extensions too. For this project, you need to add the extensions for Raspberry Pi Sense HAT, Pen, and Text to Speech. The menu to add extensions is in the bottom left.

You'll Need

> Raspberry Pi 4 (2GB or 4GB memory) **magpi.cc/rpi4**

> Sense HAT **magpi.cc/ sense-hat**

> Scratch 3 Desktop **magpi.cc/rNWKPO**

> Pibow or other case compatible with Sense HAT **magpi.cc/bbSsQo**

> Speakers to hear the engines roar
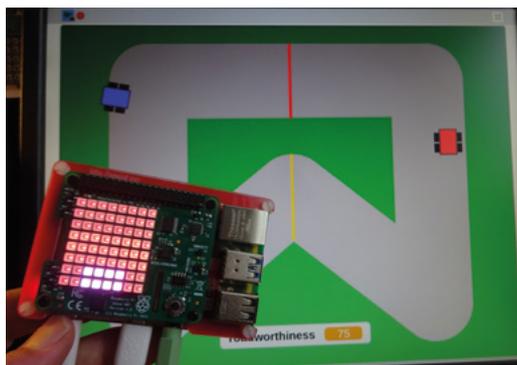
▲ The vertical bar chart in the middle of the LED matrix is the speedometer. The tiny joystick is your accelerator. Tilt to steer!

02 Draw the road

Hover over the icon in the bottom right to open the backdrop menu. Click the option to paint a backdrop and draw a green box that fills the Stage. Add **Listing 1** to Sprite1 (the cat). You'll need to use the menu in the first block to set up a new message called 'draw road'. Click the script to test it. You should see the road is drawn with two differently coloured checkpoints at the top and bottom.
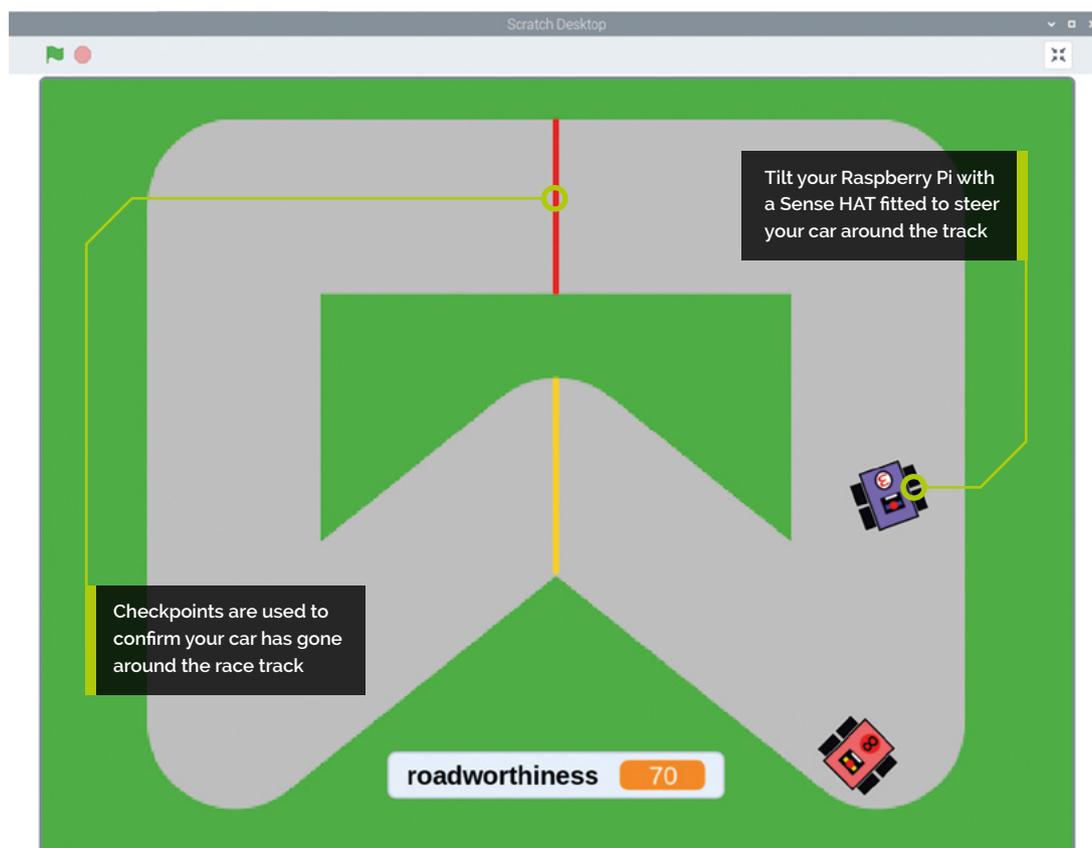
> ❝ Tilt your Raspberry Pi to steer, and use the Sense HAT's up and down joystick controls as an accelerator ❞

03 Draw the cars

Hover over the icon at the bottom of the Sprite List to add a sprite and choose Paint. The vector editor is now the default. Draw an overhead view of a race car, driving right. Centre your car on the cross-hairs target on the canvas, to make sure it turns correctly. Right-click your car in the Sprite List and duplicate it. With vector images, you can edit the shapes and colours afterwards, so you can easily make the second car look different. Click each car in the Sprite List and use the boxes above them to rename them to 'player car' and 'rival car'.

04 Start the rival car moving

The rival car is just an obstacle to avoid. Add the scripts in **Listing 2** to it. You'll need to create a new broadcast message for 'start race'. Click the green flag to position the car, then click the longer script. You should see it looping around the road, without veering off. If needed, adjust the size of the car in **Listing 2** (and **Listing 3** later) to fit both cars comfortably on the road. You might need to tweak the positions in the `glide` blocks in **Listing 2** too.

Tilt your Raspberry Pi with a Sense HAT fitted to steer your car around the track

Checkpoints are used to confirm your car has gone around the race track

roadworthiness 70

05 Create the variables

In Scratch 2, variables were created in the Data section of the Blocks Palette. Now, it's called Variables again, as it was in Scratch 1.4. Every project now starts with a variable created called `my variable` to help newcomers experiment. You need to make these variables: `half lap count`, `next checkpoint`, `qualifying time`, `roadworthiness`, `row to light`, `row to turn off`, and `speed`. They can be 'for all sprites'. Untick your variables in the Blocks Palette, except for `roadworthiness`, which we want to show on the Stage. Drag its box to the bottom middle of the Stage.

06 Set up the player's car

Click the player's car in the Sprite List and add **Listing 3** to it. Sense HAT and Text to Speech are both new in Scratch 3. The `display` block enables you to set a pixel pattern to show on the LED matrix, using the light colours you've set up with `set background` and `set colour` blocks. At the start, the game shows a chequered flag on the Sense HAT and the computer says, "Ready, Get Set, Go!"

07 Add player movement controls

You'll steer by holding your Raspberry Pi up, with the Sense HAT facing you, and tilting it left and right. You detect this axis of movement using the `pitch` block in the Sense HAT extension. Tilting left (up to 90 degrees) gives values of 0 to 90. Tilting right (up to 90 degrees) gives values of 360 to 270. **Listing 4** goes on your player's car. It turns your car if your Raspberry Pi is tilted left or right by more than 10 degrees, and uses the Sense HAT joystick to trigger the blocks to change your speed. To make the pink blocks, click My Blocks and use the button to make your own blocks, with the names you see in the listing. For the `touching color` blocks, click the colour box in the block and use the new, improved pipette to copy the checkpoint colours from the Stage.

08 Add sounds

There are fun new sounds to explore. For this game, we need the Cheer, Coin, Car Horn, Engine, and Skid effects. Click the Sounds tab on your player's car, and use the button in the bottom left to add them. You can now search for them by name.

Top Tip 👍

Random gliding

We're not using it here, but there's a new block to glide to a random position, which you might find handy in your games.

09 Add speed controls

Find the pink hat blocks for `define increase speed` and `define decrease speed`. They were created for you in the Code Area when you made those blocks in Step 7. Use them to build **Listing 5**. You'll need to create new blocks (using My Blocks again) for 'speed lights on' and 'speed lights off'. The `set pixel` Sense HAT block enables you to light an individual LED using its x and y position (both numbered from 0 to 7, from the top left). We're using it to turn the LED matrix into a speedometer, with a bar graph that grows with your speed, which can range from 0 to 8. Scratch 3 adds new sound effects blocks. We're making the engine sounds higher when you speed up. Don't confuse the sound pitch (how low or high it is) with the Sense HAT pitch (the angle of the device). Your colours in the 'speed lights off' script should match the Sense HAT background colour in **Listing 3**. Click the colour box in the block to check the numbers.

> " We're turning the LED matrix into a speedometer, with a bar graph that grows with your speed "

10 Add the lap counter

To confirm the car has done a lap of the track, we're using checkpoints. The car has to visit the red one, then the yellow one, and repeat until it's gone around three times. The next checkpoint variable remembers which checkpoint the car needs to visit next. The `half lap count` variable stores how many checkpoints have been

▼ When you lift your Raspberry Pi up to face you, it's still the pitch measurement you need if you move it like a see-saw above the desk

Yaw: Z

Pitch: Y

Roll: X
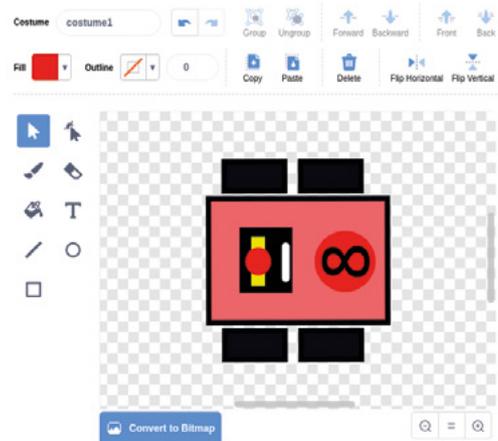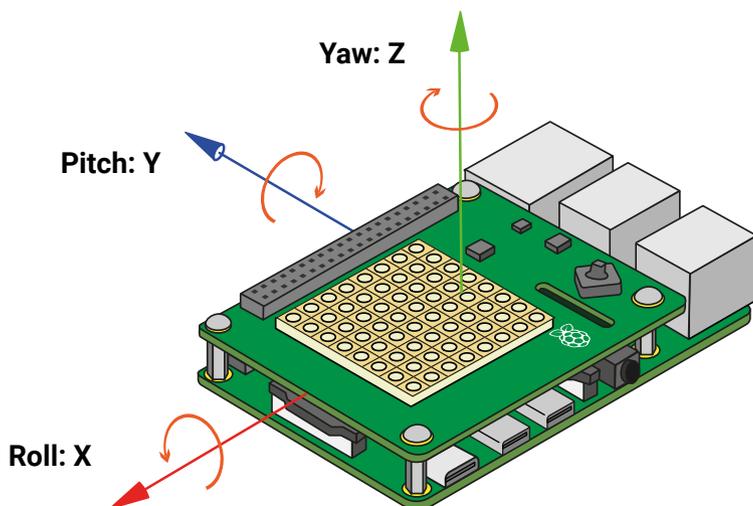
▲ The vector art editor is now the default. It makes it easy to modify the race cars to make two distinct designs

successfully crossed. Add **Listing 6** to your existing `define lap counter` block in the Code Area. Copy the red and yellow colours in the `touching color` blocks from the Stage.

11 Detect car damage

Your `roadworthiness` variable is like a health score. It goes down when you hit the grass or the other car. The two scripts in **Listing 7** continuously check whether you've hit something. If you have, they reduce your score and then pause briefly to stop your health sapping too fast. Use the pipette to copy the exact grass colour and the main body colour of your rival car (pink in our case) into your scripts. **Listing 7** goes on the player's car.

12 Add game-over sequences

There are two ways for the game to end. Either your roadworthiness gets to zero, or you successfully complete three laps. The `display text` block scrolls an appropriate message across the Sense HAT's LEDs. Attach **Listing 8** to the end of **Listing 4**, the script with movement controls on the player's car. Now you're ready to play! **M**

Top Tip 👍

Simplifying strings

There's a new Operators block called **apple contains a?** Use it to search a piece of text more easily.

throttleandtilt.sb3

> Language: **Scratch**

Listing 1

```
when I receive draw road
  set pen color to
  set pen size to 80
  go to x: -150 y: 120
  erase all
  pen down
  go to x: -150 y: -120
  go to x: 0 y: 0
  go to x: 150 y: -120
  go to x: 150 y: 120
  go to x: -150 y: 120
  pen up
  set pen size to 2
  set pen color to
  go to x: 0 y: 80
  pen down
  go to x: 0 y: 160
  pen up
  set pen size to 2
  set pen color to
  go to x: 0 y: -50
  pen down
  go to x: 0 y: 40
  hide
```

Listing 2

```
when [flag] clicked
  go to x: 150 y: 100
  point in direction -90
  set size to 40 %

when I receive start race
  forever
    point in direction -90
    glide 6 secs to x: -140 y: 100
    point in direction -135
    glide 0.5 secs to x: -170 y: 80
    point in direction 180
    glide 4 secs to x: -170 y: -130
    point in direction 45
    glide 4 secs to x: 0 y: -30
    point in direction 135
    glide 4 secs to x: 140 y: -135
    point in direction 45
    glide 0.5 secs to x: 170 y: -120
    point in direction 0
    glide 4 secs to x: 170 y: 80
    point in direction -45
    glide 0.5 secs to x: 150 y: 100
```

Listing 3

```
when [flag] clicked
  set background to
  set colour to
  display
  set roadworthiness to 100
  show variable roadworthiness
  hide variable qualifying time
  set next checkpoint to red
  set half lap count to 0
  set speed to 0
  broadcast draw road and wait
  clear sound effects
  set size to 40 %
  go to x: 150 y: 140
  point in direction -90
  speak Ready
  speak Get set!
  speak Go!
  clear display
  set background to
  broadcast start race
```

Listing 4

```
when I receive start race
  reset timer
  repeat until half lap count = 7 or roadworthiness < 1
    move speed / 2 steps
    if pitch < 90 and pitch > 10 then
      turn ↺ 10 degrees
    if pitch < 350 and pitch > 270 then
      turn ↻ 10 degrees
    if joystick pushed up ? then
      increase speed
    if joystick pushed down ? then
      decrease speed
    if touching color ? or touching color ? then
      lap counter
```

Listing 6

```
define lap counter
  if next checkpoint = red and touching color ? then
    change half lap count by 1
    set next checkpoint to yellow
    start sound Coin
  if next checkpoint = yellow and touching color ? then
    change half lap count by 1
    set next checkpoint to red
    start sound Coin
```

Listing 5

```
define increase speed
  change speed by 1
  set row to light to 8 - speed
  if speed > 8 then
    set speed to 8
  speed lights on
  set pitch effect to speed * 10
  start sound Engine
```

```
define decrease speed
  set row to turn off to 8 - speed
  speed lights off
  change speed by -1
  if speed < 0 then
    set speed to 0
  set pitch effect to speed * 10
  start sound Engine
```

```
define speed lights off
  set pixel x 2 y row to turn off to
  set pixel x 3 y row to turn off to
  set pixel x 4 y row to turn off to
  set pixel x 5 y row to turn off to
```

```
define speed lights on
  set pixel x 2 y row to light to
  set pixel x 3 y row to light to
  set pixel x 4 y row to light to
  set pixel x 5 y row to light to
```

Listing 7

```
when I receive start race
  forever
    if touching color ? and roadworthiness > 0 then
      change roadworthiness by -5
      start sound Skid
      wait 0.5 seconds
```

```
when I receive start race
  forever
    if touching color ? and roadworthiness > 0 then
      change roadworthiness by -10
      start sound Car Horn
      wait 0.5 seconds
```

Listing 8

```
if roadworthiness < 1 then
  speak Failed to qualify
  display text Game Over
else
  set qualifying time to timer
  show variable qualifying time
  clear sound effects
  start sound Cheer
  speak Congratulations! You qualified!
  display text Qualified!
stop all
```

Part 03

Build a low-cost
wheeled robot

A wheeled robot isn't much fun when it's stationary. Using the motor controller and code, it's time to get the robot moving!

MAKER

Danny Staple

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

orionrobots.co.uk

You'll Need

> Lunchbot

> Set of 6 × AA batteries (nice and fresh, alkaline)

> 16GB microSD card

> microSD card reader

> WiFi network

The previous part of this series showed how to build a robot using a lunchbox as a chassis (call it Lunchbot!). While that robot looks smart, and like it might work, it's not going to do much without code to do something on it. In this part, we'll get your Raspberry Pi ready for robotics and write some simple code to move the robot.

You'll set up the microSD card for running code over wireless LAN, add tools to connect to the robot's motor board, and write code to instruct the motors to turn.

01 Prepare the microSD card

When the robot is moving, Raspberry Pi can't be tethered to a screen or keyboard. A 'headless' configuration is needed. Headless means designed to be operated via the network. A NOOBs or Raspbian desktop image is unsuitable for this, so use Raspbian Lite.

Download a Raspbian Lite image from **magpi.cc/raspbian** and burn it onto the microSD card. Etcher (**balena.io/etcher**) is recommended

for this. When it is ready, eject the microSD card and put it back into the computer. Then follow the instructions at **magpi.cc/RhviuV** to configure WiFi (in the **wpa_supplicant.conf** file) and enable SSH.

02 Connect to your Raspberry Pi

The microSD card and USB power input on the robot's Raspberry Pi may not be accessible. If needed, loosen the bolts to get to it. Put the microSD card into the robot's Raspberry Pi.

You can power this Raspberry Pi from a USB cable during setup. When the lights stop flashing, Lunchbot's Raspberry Pi has booted and should be ready to contact.

Raspberry Pi has a guide on finding it on the network at **magpi.cc/kNOAbE**. It's called **raspberrypi.local** for now, but you can change this by following **magpi.cc/HZGBFR**.

Since it is headless, SSH is the right way to interact with the robot. See **magpi.cc/2sLqBmM** and **magpi.cc/ssh** for instructions on getting in.

03 Installing robotics software

This robot is programmed in Python 3, using GPIO Zero. Use SSH to access Lunchbot's Raspberry Pi, then install GPIO Zero with:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install python3-pip python3-gpiozero
```

Type `python3`. You should see something like:

```
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license"
```
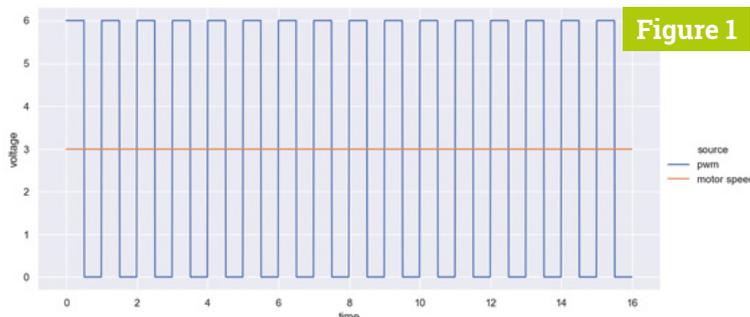

Figure 1

▲ **Figure 1** A PWM square wave in action, keeping a motor at a constant speed by turning the power on and off

Table 1

IN 1	IN 2	Motor
Low	Low	Off
Low	High	Spin Forward
High	Low	Spin Backward
High	High	Off

for more information.
>>>

Type `import gpiozero` to test GPIO Zero is ready. It should not show an error. Press **CTRL+D** to exit. Shut down your Raspberry Pi with `sudo poweroff`, remove the USB power, and bolt it in place.

04 Motor controller concepts

The motor controller has two halves. Each half controls a motor using two outputs, using power and three I/O input pins.

The motor outputs can handle higher currents and voltage than the inputs need. Connecting motors directly to Raspberry Pi would damage it.

The enable input pins turn on half of the controller. Labelled ENA for motor A or ENB for motor B, they are connected with a 'jumper wire' to a 5 V line (held high) to keep the half enabled. Each motor has two direction control pins: IN1 and IN2 for motor A; IN3 and IN4 for motor B.

05 Controlling speed and direction

The IN pins are used in pairs to control a motor. See **Table 1** for how the pins control the direction of the motor. You can substitute IN 3 and IN 4 for the behaviour on the other motor.
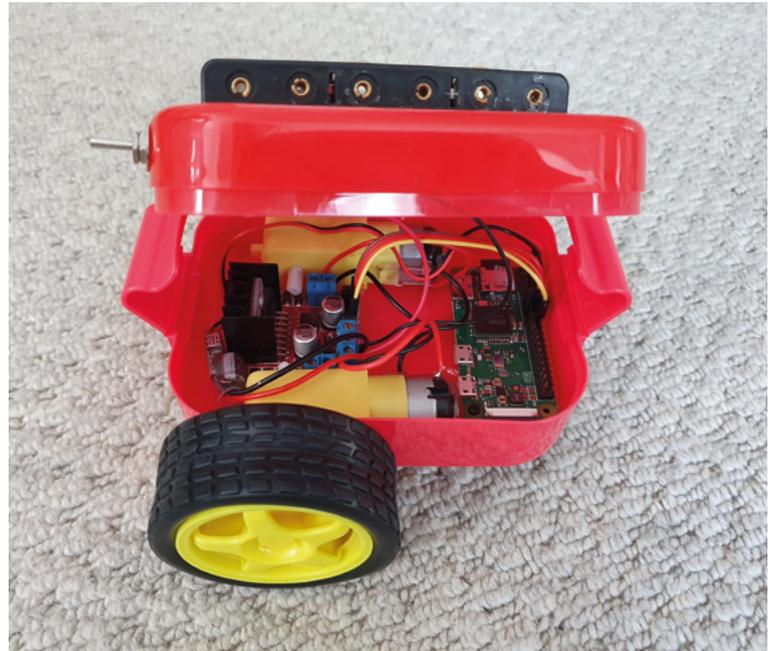
drive_forward.py

DOWNLOAD THE FULL CODE:
magpi.cc/QkkmqW

> Language: **Python 3**

```python
001. import gpiozero
002. import time
003.
004. robot = gpiozero.Robot(left=(27, 17), right=(24, 23))
005.
006. try:
007.     # Robot actions here
008.     robot.forward()
009.     time.sleep(1)
010. finally:
011.     robot.stop()
```
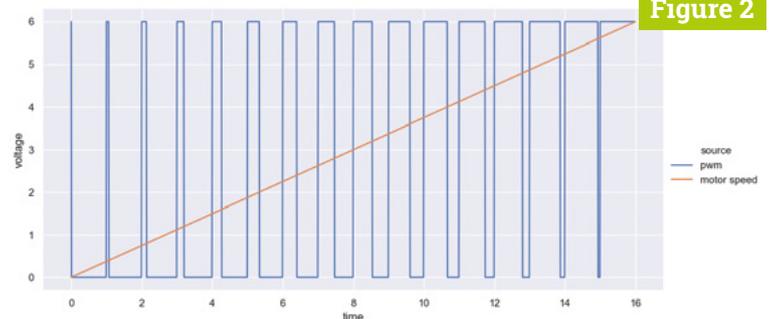

▲ **Figure 2** PWM can control the speed of a motor by changing the on/off ratio
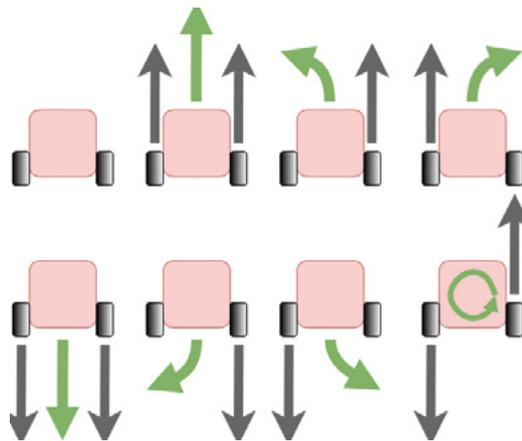
▶ How a wheeled robot moves. The dark arrows show the direction of the wheels; the green arrows show the movement of the robot

This controls the direction, but what about speed? By turning the motor power on and off rapidly (in square waves), the ratio of time on vs time off changes the speed of the motor. **Figures 1** and **2** show how this works. This is known as pulse-width modulation or PWM. The GPIO Zero library can set the direction and the PWM speed.

Top Tip 👍

Have plenty of driving space

The robot is going to move and needs space to do so. Don't test this on a tabletop!

06 Not driving off the table

Robots under test have a tendency to shoot off until they hit a wall or off a desk, and not in the direction you expect. Emergency stop measures are essential. First, ensure that motors are always turned off. Python's `try…finally` block applies in almost all cases. Even if the program crashes, it runs the `finally` code.

```
try:
    pass
    # Code that starts motors
finally:
    # Code to stop motors
```

The other measure is an emergency stop button. Python comes with such a button built-in: pressing the keyboard combination **CTRL+C** will stop the code running and enter the `finally` code.

07 Setting up robot code

The setup for robot experiments is in **drive_forward.py**. Use this as a template for robot code. Line 1 and 2 make the GPIO Zero and time libraries available by importing them.

GPIO Zero has a handy robot object to control two motors. At line 4, a robot object is created with pin numbers for the motor controller.

Lines 6 and 10 provide the stop mechanism from Step 6, with line 10 stopping the robot.

Between these lines, after the comment (starting with a '#') on line 7 is the code to make the robot do something. This code can be swapped/changed for different movement.

08 Driving forwards

In **drive_forward.py**, line 8 instructs the controller to drive both motors forward.

The delay on line 9 is needed so that the robot drives for a little time before stopping the motors. `time.sleep()` uses a time in seconds.

Put in batteries to switch to battery power. If you don't have a switch, use the last battery as one.

Use SCP to put this code on the robot or use Raspberry Pi to edit it. Put it in **/home/pi**. Give the robot space to drive and run the code:

```
python3 drive_forward.py
```

It should drive forward for one second and stop. Press **CTRL+C** if it does not stop.

09 Troubleshooting driving forwards

If running **drive_forward.py** shows errors, check everything in Step 3, and that the code is exactly as listed.

If one motor (or both) doesn't move, inspect the motor controller wiring. Ensure the enable jumpers are fitted. If the motors whine but don't move, try fresh batteries. If a motor goes backwards, then swap its pin numbers on line 4.

The robot may veer to one side; check the motors are glued in well, without friction between wheels, axles, and robot. Some veer is unavoidable without sensors. This is when loose screws in the robot will show up and need fixing.

10 Simple turns and driving backwards

The robot object in GPIO Zero makes it very easy to do this for the simplest movements.

Where the code has `robot.forward()` on line 8 of **drive_forward.py**, this can be replaced with `robot.left()`. It is only now you will find out if you have the motors the correct way around! If the robot drives to the right, then swap the `left` and `right` keywords on line 4.

The robot is likely to turn too far, so reduce the timer value in line 9, this can be decimals, so try

0.3 seconds here. On line 8, try `robot.right` and `robot.backward` too.

11 Chaining movements

So far, this code has been straightforward with a single movement. A robot can do a sequence of actions after each other.

The code listing **chaining_movements.py** shows how. By tuning the timings, this will drive in a shape such as a hexagon.

Line 8 makes a loop, so the action will be repeated. Lines 9, 10 drive forward for a little time. Lines 11,12 then turn left for a bit of time too.

Lines 13 and 14 then do a victory spin to the right. An excellent, fun way to end autonomous events if you can determine a victory condition.

12 Moving at slower speeds

A robot's motors can do more than just change direction: PWM can be used to vary the speed. The listing **speed_control.py** shows how.

Line 8 sets up a loop, in this case counting down from 10 to 4. The speed is set in line 9 using `robot.value`, with the left value setting the left motor speed, and the right value setting the right. Values go from zero to 1, so divide by 10. Negative values go backwards.

Lines 12 to 15 do some turns with more precise control than the `robot.left` or `robot.right` commands. These will be handy for sensors. 🅼
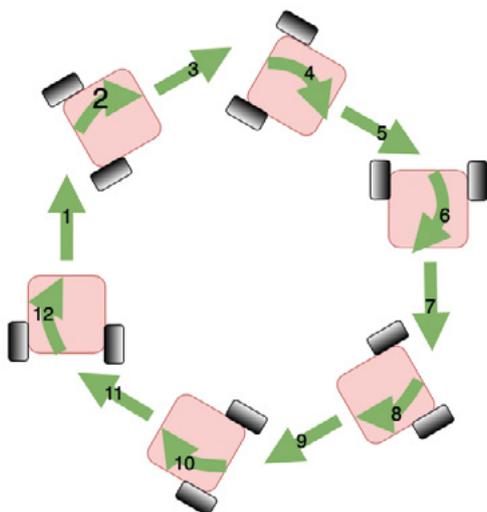
▲ By chaining movements together, a robot can make shapes on the floor. Add a pen and it could draw them too!

chaining_movements.py

> Language: **Python 3**

```
001.   import gpiozero
002.   import time
003.
004.   robot = gpiozero.Robot(left=(27, 17), right=(24, 23))
005.
006.   try:
007.       # Robot actions here
008.       for n in range(6):
009.           robot.forward()
010.           time.sleep(0.5)
011.           robot.left()
012.           time.sleep(0.3)
013.       robot.right()
014.       time.sleep(1)
015.   finally:
016.       robot.stop()
```

❝ A robot's motors can do more than just change direction: PWM can be used to vary the speed ❞

speed_control.py

> Language: **Python 3**

```
001.   import gpiozero
002.   import time
003.
004.   robot = gpiozero.Robot(left=(27, 17), right=(24, 23))
005.
006.   try:
007.       # Robot actions here
008.       for speed in range(10, 4, -1):
009.           robot.value = (speed/10, speed/10)
010.           time.sleep(0.4)
011.       # Smaller turns with forward motion
012.       robot.value = (0.5, 1) # left
013.       time.sleep(1)
014.       robot.value = (1, 0.5) # right
015.       time.sleep(1)
016.   finally:
017.       robot.stop()
```

The Photon Lab

Make accurate light level measurements by turning your Raspberry Pi into a powerful scientific tool to investigate glowing materials

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 jPi Projects for Dummies*.

magpi.cc/TPaUfT

You'll Need

> TSL2591 board
> **magpi.cc/Mnpiub**

> OVL-5521 LED
> **magpi.cc/eKNPyQ**

> 2N4401 transistor
> **magpi.cc/Zsssze**

Ever seen those glow-in-the-dark products? With the Photon Lab, you can learn how they work and test how effective they are. It can also be used for seeing how effective lights are at charging up as well, by comparing several photoluminescent materials. Learn about the concept of half-life, with light as safe radiation. You could even see how much of an LED's light capacity is lost when it is overdriven for a length of time.

01 What is photoluminescence?

When a material gives off light, it can be from different types of mechanisms. It seems that a lot of merchants on the web are confused and often their products are mislabelled. The 'photo' part of the word means that photons (quantum particles of light) are responsible for supplying the energy that produces the light. Yes, light produces light, but this often involves a colour change. With photofluorescence, light is absorbed by a material and almost immediately generates light of another colour. With phosphorescence, light is absorbed and then slowly releases the energy as light of another colour.

Figure 1
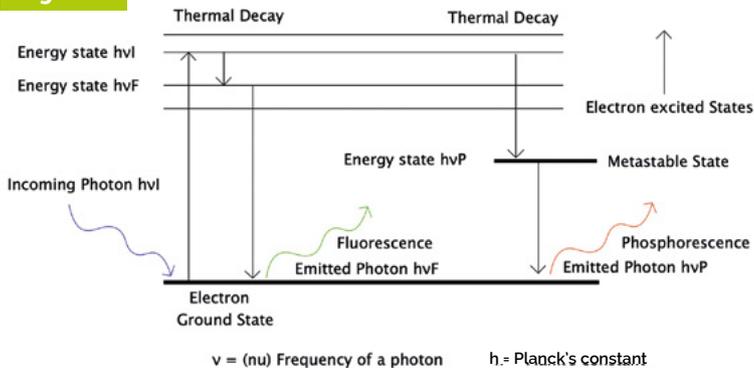
▲ **Figure 1** Electron energy diagram showing how photoluminescence works
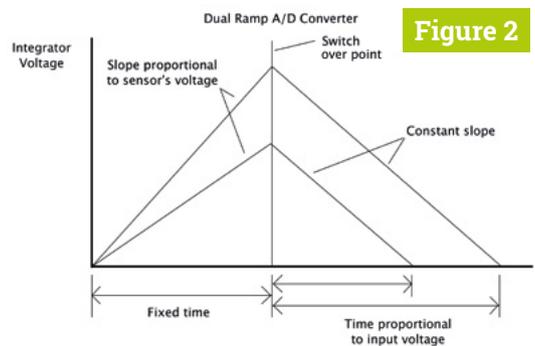
Figure 2

▲ **Figure 2** The dual ramp converter in the light sensor chip

02 How does this work?

Figure 1 shows these two processes, by means of an electron energy diagram. Electrons normally sit at a low energy state and can be excited into higher energy states when a photon hits the material. Due to the laws of quantum mechanics, the electron can only have certain discrete energy levels. Once excited to one of these levels, they decay in a very short time; this is fluorescence and is shown on the left. However, due to the structure of some materials, there are metastable states. These can hold excited electrons for much longer times – from seconds to hours – before they decay, as shown on the right.

03 Testing phosphorescence

Our Photon Lab will use a TSL2591 light sensor, but as these are hard to get hold of in small quantities, and very tricky to solder, we are going to use the Adafruit TSL2591 breakout board. This chip has two light sensors: a visible one and an IR (infrared) one, plus a dual slope A/D (analogue to digital converter) for each sensor. This sort of converter charges a capacitor with the sensor's voltage, and then discharges it through a known resistor. Timing how long it takes to discharge is a measure of the sensor's output voltage. See **Figure 2**.

Sample to test for photoluminescence

The Photon Lab sensor and LED
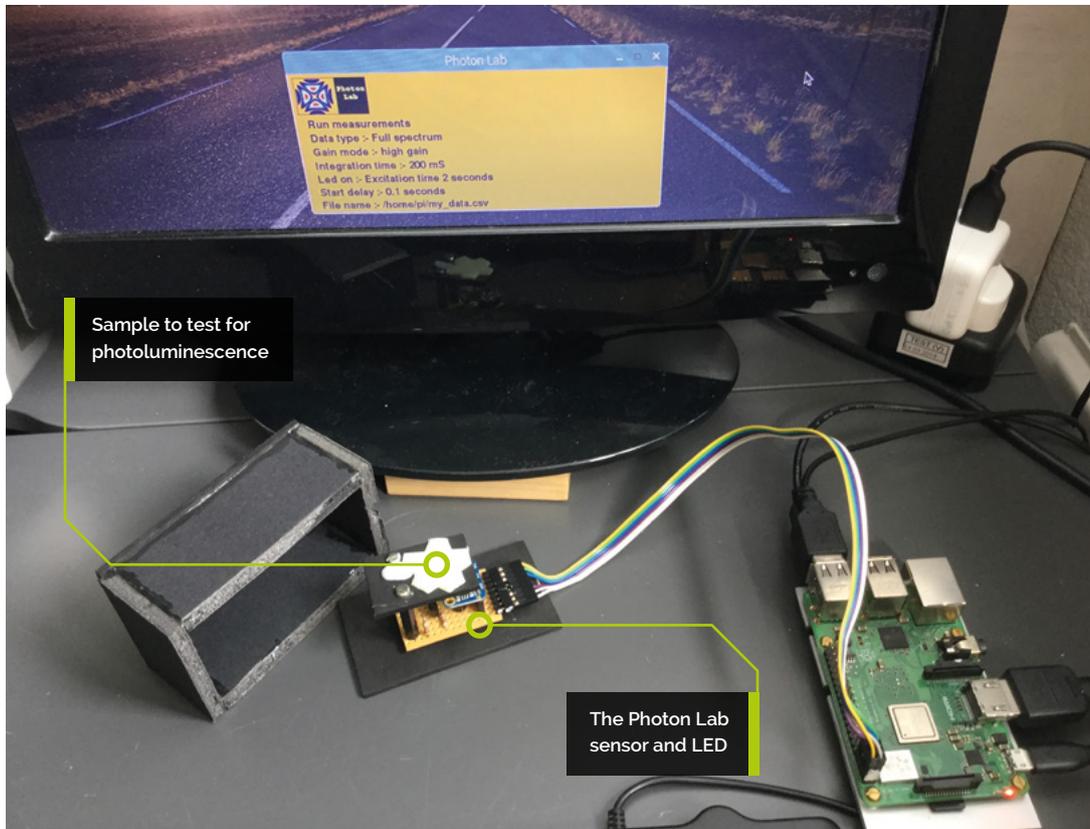
Top Tip 👍

The energy in a photon

A photon carries energy that depends on the frequency: the higher the frequency, the more energy it has. The frequency of a photon defines its colour, so a blue photon has more energy than a red one.
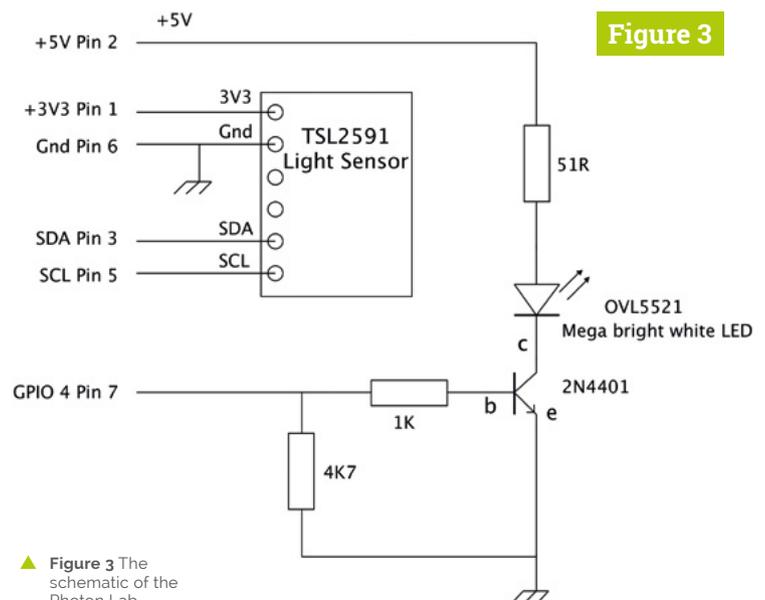
04 Our circuit

We need the TSL2591 light sensor board, plus an LED, to illuminate our target – and as we are dealing with very low light levels, this needs to be inside a light-tight enclosure. The schematic of our Lab is shown in **Figure 3**, and the physical layout diagram is shown in **Figure 4** (overleaf). The TSL259 communicates over the I²C bus and we need an LED to act as the excitation light. As this pin powers up as an input, with the internal pull-up resistors enabled, we need a 4.7 kΩ pull-down resistor to stop the LED from lighting from switch-on.

05 Making the box

First, cut the stripboard to size and drill the three holes in it. See **Figure 5** (overleaf) for the dimensions of the light-tight box. Glue the wood top to the four sides made from foam-filled mounting board. Make sure there are no light leaks by using fillets of glue on both inside and outside seams – we use a PVA glue for this. Cut the square hole for the sample holder by marking it out, and drilling a series of 3 mm holes close together round the inside, then cutting them together with a scalpel, and finally finishing with a file.

> ❝ Make sure there are no light leaks by using fillets of glue on both inside and outside seams ❞
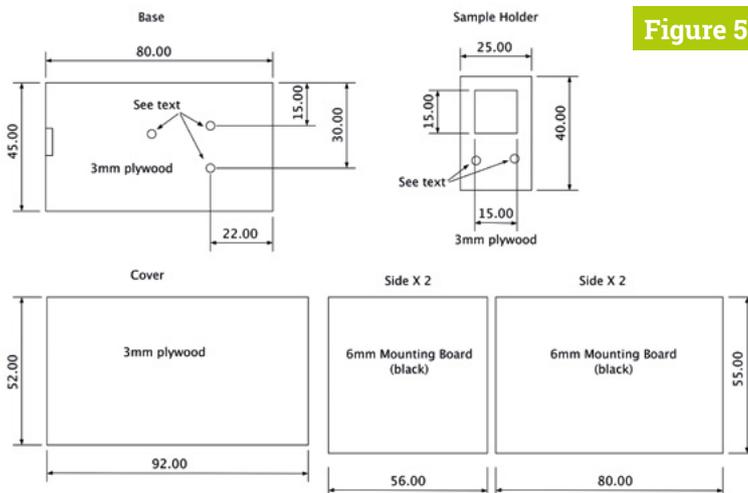

Figure 3

▲ **Figure 3** The schematic of the Photon Lab

> **Before assembly, paint everything matt black to stop any internal reflections**

integration time, along with other things like the file name to use to store the results. The way we made the software work is that the keyboard key that matches the first letter of the line controls the parameter. If just the key is pressed, the parameter increases; if **SHIFT** and the key is held down, the parameter decreases.
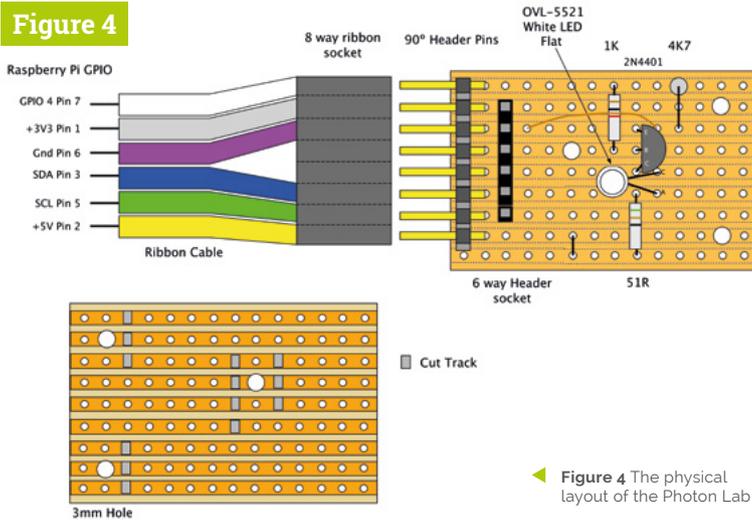
Top Tip 👍

Using the strip board as a template

When making the box, lay the stripboard, track side down, on the base, and drill through the three holes to the wood underneath. Do this for the sample holder as well.

06 Finishing off

Solder header pins onto the TSL2591 board and build up the sensor board. Make sure you leave the maximum length of lead on the LED, as this will need bending towards the target. Cut a small notch on the base plate to allow the ribbon cable to exit the box. Use countersunk screws and 2 mm nylon spacers to mount the stripboard, with 25 mm hex spacers holding the sample holder and a 20 mm spacer for the board rest. Before assembly, paint everything matt black to stop any internal reflections in the light-tight box (**Figures 6a**, **6b**).

08 The software

The software is shown in **Photon_Lab.py**. We decided not to use any of the existing libraries for this chip, as it is so simple to use once you find the right I²C commands in the data sheet. The `setGain` function sets both the gain and integration time of each measurement with a write to a single register. With the `readSensor` function checking the status register until it indicates a measurement is finished, it then reads the four bytes which contain the IR and visible readings. Then it kicks off the next reading while it processes this one.

07 Software design

For testing photoluminescence, we need to (optionally) light up an LED to charge the material, and then take light readings until we decide enough have been taken. What the software needs to do is to be able to set both the gain and

09 Using the software

First off, press the **F** key to bring up a dialog box enabling you to choose the file name and path for the results file. Press the **D** key to choose the data type; the Full Spectrum option is simply the addition of visible and IR sensor readings. Press the **G** and **I** keys to set the gain and integration time; use the **SHIFT** key to make these go down. The **L**
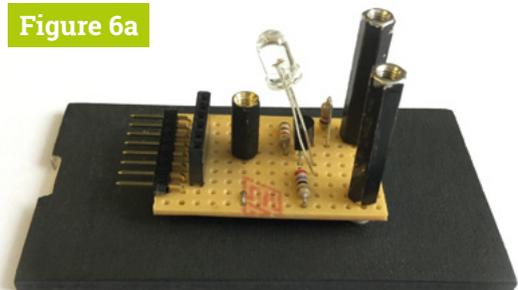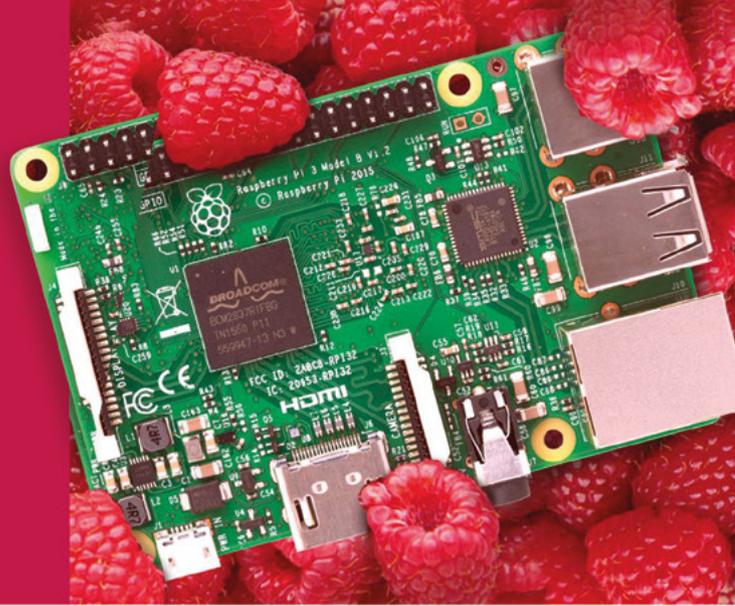
American
Raspberry Pi
Shop

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

Partner and official reseller for top Pi brands:

adafruit sparkfun ELECTRONICS PIMORONI HiFiBerry

and many others!

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

USA **PiShop.us**

Canada **BuyaPi.ca**

Raspberry Pi APPROVED RESELLER

key toggles if the excitation LED is turned on, with the **E** key determining how long it is on for, and the **S** key giving the length of delay after turning off the LED and starting the readings. This is needed because white LEDs do not turn off very quickly.

> 🔲 Note that if the light is too bright for the sensor, and it saturates, the measurement will be shown as -1 🔲

10 Making the measurements

When all the parameters are set up, the **R** key will run the measurement sequence. Once the sequence gets to the point of making and recording the light levels, a box appears in the left-hand corner showing the sample number and the reading. When you have reached sufficient samples, or the readings show you are not getting any meaningful measurements any more, you can press the **C** key to indicate your measurements are complete. Note that if the light is too bright for the sensor, and it saturates, the measurement will be shown as -1. So, back off the gain or integration time for the next run.

11 What to do with your results?

The measurements, along with the parameters you used, are stored in a CSV file and can be opened by any spreadsheet application. You

▼ **Figure 6b** The assembly of the Photon Lab
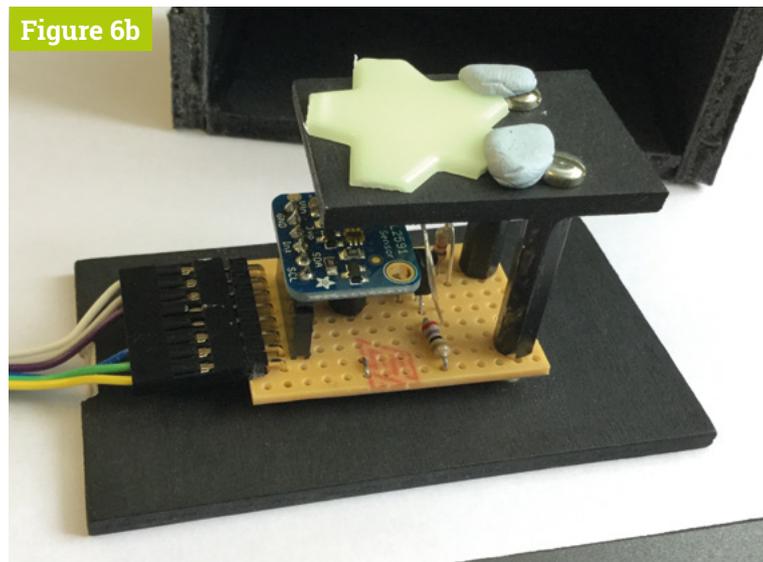
Figure 6b
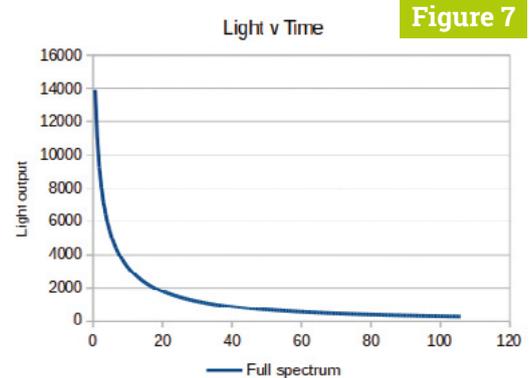
Figure 7

▲ **Figure 7** Graph of the output using LibreOffice Calc

fire up LibreOffice Calc and open the CSV file you have just created – click OK to accept the default import settings. The file will open in read-only mode, so click the Edit button. Then, select the first column and, holding down the **SHIFT** key, select the second column as well. Now click on the chart icon and select the X-Y scatter type of chart; click on your choice of points or joined line. You can also add axis labelling if you like, then save the spreadsheet as an ODS file – see **Figure 7**.

12 Understanding the results

What you should see is a sudden drop in light level followed by a much smaller drop with time. This is governed by how long photons can stay in the metastable state. This is not a single fixed time: it is a statistical probability, expressed as the amount of time it takes for half of the electrons to decay. This is known as the half-life of the metastable state. You might have heard of half-life in relation to radioactivity; this is exactly the same sort of measure. However, your curve is further complicated by the fact there are often several metastable states in any material, all having different half-lives.

We have put some teacher's notes on our GitHub page to allow you to explore the subject in a teaching environment. You can test several different types of glow-in-the-dark materials, as well as the miniature glow sticks used for fingernail decoration. These might require samples taken at a less regular interval than the software shown here, but the inclusion of a delay after the sample has been taken will do that. So, let's not be in the dark about low light level measurements. 🔲

Photon_Lab.py

> Language: **Python**

```python
001.  #!/usr/bin/env python3
002.  # Photon Lab
003.  # By Mike Cook August 2019
004.
005.  from smbus import SMBus
006.  import RPi.GPIO as io
007.  import os, pygame, sys, time
008.  from tkinter import filedialog
009.  from tkinter import *
010.
011.  pygame.init()
012.  os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
013.  pygame.display.set_caption("Photon Lab")
014.  pygame.event.set_allowed(None)
015.  pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
016.  sWidth = 500 ; sHight = 200 ; bk = (0,0,0)
017.  screen = pygame.display.set_mode([sWidth,sHight],0,32)
018.  textHeight = 22;font=pygame.font.Font(None,textHeight)
019.  backCol = (200,130,0);displayUpdate=True;running=False
020.
021.  def main():
022.      global displayUpdate, sampleTime, enabledS
023.      init()
024.      while 1:
025.          while not running: # set up parameters
026.              time.sleep(0.2) ; checkForEvent()
027.              if displayUpdate:
028.                  drawScreen(); displayUpdate = False
029.          # prepare to run
030.          data_file = open(fileName,'w')
031.          data_file.write("Time,"+dataTypeT[dataType]+",
      "+gainT[gain]+",Integration time:- "+intgTimeT[intgTime]
      +",Excitation time "+str(exiTime)+",Start delay:- "
      +str(startDelay/10)+" seconds \n")
032.          if ledOn : # Excitation phase
033.              io.output(4,1);time.sleep(exiTime)
034.              io.output(4,0);time.sleep(startDelay/10)
035.          sensorEnable() # start up sensor
036.          sampleTime=time.time()
037.          while running: # put samples in a file
038.              readSensor()
039.              data_file.write(str("%.3f" % round(
      sampleTime,3)+","+str(ch[dataType])+"\n"))
040.              updateScreen(
      str(ch[dataType])); checkForEvent()
041.          data_file.close() # finished run
042.          enabledS = False # shut down the light sensor
043.          readSensor() # remove the last reading
044.          sensorDisable() # shut it down
045.
046.  def readSensor():
047.      global ch0,sampleTime, sampleNumber
048.      done = False
049.      while not done:
050.          status = bus.read_byte_data(luxAdd, 0xA0 | 0x13)
051.          if status & 1 == 1:
052.              done = True
053.              bus.write_byte_data(luxAdd, 0xA0, 0x01)
054.      adc = bus.read_i2c_block_data(luxAdd,0xA0 | 0x14,4)
055.      ch[0] = adc[1] << 8 | adc[0] # IR
056.      if ch[0] == 0xffff: # overflow check
057.          ch[0] = -1
058.      ch[1] = adc[3] << 8 | adc[2] # Visible
059.      if ch[1] == 0xffff:
060.          ch[1] = -1
061.      ch[2] = -1
062.      if ch[0] != -1 and ch[1] != -1:
063.          ch[2] = ch[0] + ch[1]    # Full spectrum
064.      if enabledS : # start new conversion cycle
065.          bus.write_byte_data(luxAdd, 0xA0, 0x03) # next
066.          sampleTime = time.time() - startTime
067.          sampleNumber += 1
068.
069.  def init():
070.      global luxAdd, bus, gain, dataType, intgTime,gainT
071.      global ledOn, exiTime, fileName, startDelay, logo
072.      global sampleNumber, dataTypeT, intgTimeT, ch
073.      bus = SMBus(1) # use bus 1
074.      luxAdd = 0x29 # address of TSL2591
075.      io.setmode(io.BCM); io.setwarnings(False)
076.      io.setup(4,io.OUT) ; io.output(4,0)
077.      # confirm we have a sensor attached
078.      print("Checking for TSL2591 sensor")
079.      sec = bus.read_byte_data(luxAdd, 0xA0 | 0x12) # ID
080.      if sec != 0x50:
081.          print("TSL2591 not found")
082.      else :
083.          print("TSL2591 found")
084.      gain = 2 ; intgTime = 1 ;setGain() ;ledOn = True
085.      gainT = [
      "low gain","medium gain","high gain","maximum gain"]
086.      intgTimeT = [
      "100 mS","200 mS","300 mS","400 mS","500 mS","600 mS"]
087.      dataTypeT = ["IR","Visible","Full spectrum"]
088.      exiTime = 2 ; dataType = 2 ; startDelay = 1
089.      ch = [0,0,0]; sampleNumber = 0
090.      fileName = "/home/pi/my_data.csv"
091.      logo = pygame.image.load(
      "images/Logo.png").convert_alpha()
092.
093.  def sensorEnable():
094.      global enabledS,sampleNumber, startTime
095.      bus.write_byte_data(luxAdd, 0xA0, 0x03) # AEN on
096.      enabledS = True; sampleNumber = 0
097.      startTime = time.time()
098.
099.  def sensorDisable():
100.      global enabledS
101.      bus.write_byte_data(luxAdd, 0xA0, 0x01) # AEN off
102.      enabledS = False
103.
104.  def setGain(): # gain 00 to 11 - inter 000 to 101
```

Photon_Lab.py (continued)

> Language: **Python**

```python
105.        bus.write_byte_data(luxAdd,0xA1,(gain<<4)|intgTime)
106.
107.  def constrain(val, min_val, max_val):
108.      return min(max_val, max(min_val, val))
109.
110.  def wrap(newVal,min_val, max_val):
111.      retVal = newVal
112.      if newVal < min_val :
113.          retVal = max_val
114.      if newVal > max_val :
115.          retVal = min_val
116.      return retVal
117.
118.  def getFileName():
119.      global fileName
120.      root = Tk()
121.      root.filename = filedialog.asksaveasfilename(
initialdir = "/home/pi",title = "file to save data to",
filetypes = (("csv files","*.csv"),("all files","*.*")))
122.      temp = root.filename
123.      if len(temp) >3 :
124.          fileName = temp
125.      root.withdraw()
126.
127.  def updateScreen(data):
128.      white = (255,255,255)
129.      pygame.draw.rect(screen,white,(320,16,115,45),0)
130.      drawW("Sample "+str(sampleNumber),335,20,bk,white)
131.      drawW("Value "+data,335,40,bk,white)
132.      pygame.display.update()
133.
134.  def drawScreen():
135.      screen.fill(backCol)
136.      screen.blit(logo, (5,5) )
137.      if running :
138.          drawW("Complete measurements ",15,60,bk,backCol)
139.      else:
140.          drawW("Run measurements ",15,60,bk,backCol)
141.      drawW("Data type :- "+dataTypeT[dataType],
      15,80,bk,backCol)
142.      drawW("Gain mode :- "+gainT[gain],15,100,(0,0,0),backCol)
143.      drawW("Integration time :- "+intgTimeT[intgTime],
      15,120,bk,backCol)
144.      if ledOn:
145.          drawW("Led on :- Excitation time "+str(exiTime)+
      " seconds",15,140,bk,backCol)
146.      else:
147.          drawW("Led off",15,140,bk,backCol)
148.      drawW("Start delay :- "+str(startDelay/10)+
      " seconds",15,160,bk,backCol)
149.      drawW("File name :- "+fileName,15,180,bk,backCol)
150.      pygame.display.update()
151.
152.  def drawW(words,x,y,col,backCol) : # drawWords
153.      textSurface = font.render(words,True, col,backCol)
154.      textRect = textSurface.get_rect()
155.      textRect.left = x # right for align right
156.      textRect.top = y
157.      screen.blit(textSurface, textRect)
158.
159.  def terminate(): # close down the program
160.      print ("Closing down")
161.      sensorDisable() ; io.cleanup()
162.      pygame.quit() # close pygame
163.      os._exit(1)
164.
165.  def checkForEvent(): # see if we need to quit
166.      global gain, intgTime, displayUpdate, dataType
167.      global running, startDelay, exiTime, ledOn
168.      event = pygame.event.poll()
169.      if event.type == pygame.QUIT :
170.          terminate()
171.      if event.type == pygame.KEYDOWN :
172.          if event.key == pygame.K_ESCAPE :
173.              terminate()
174.          if event.key == pygame.K_c : # data run Complete
175.              running = False
176.              displayUpdate = True
177.        if not running:   # don't look if running
178.              displayUpdate = True
179.              if event.mod == pygame.KMOD_LSHIFT or event.mod
      == pygame.KMOD_RSHIFT :
180.                  inc = -1
181.              else:
182.                  inc = 1
183.              if event.key == pygame.K_g : # gain
184.                  gain += inc
185.                  gain = constrain(gain,0,3)
186.                  setGain()
187.              if event.key == pygame.K_i : #integration time
188.                  intgTime += inc
189.                  intgTime = constrain(intgTime,0,5)
190.                  setGain()
191.              if event.key == pygame.K_d: # data type to use
192.                  dataType += inc
193.                  dataType = wrap(dataType,0,2)
194.              if event.key == pygame.K_l : # excitation LED
195.                  ledOn = not ledOn
196.              if event.key == pygame.K_e : # excitation time
197.                  exiTime += inc
198.                  exiTime = constrain(exiTime,0,13)
199.                  if exiTime == 0:
200.                      ledOn = False
201.                  else:
202.                      ledOn = True
203.              if event.key == pygame.K_f : # file name
204.                  getFileName()
205.              if event.key == pygame.K_s : # start delay
206.                  startDelay += inc
207.                  startDelay = constrain(startDelay,0,20)
208.              if event.key == pygame.K_r : # run
209.                  running = True
210.
211.  # Main program logic:
212.  if __name__ == '__main__':
213.      main()
```
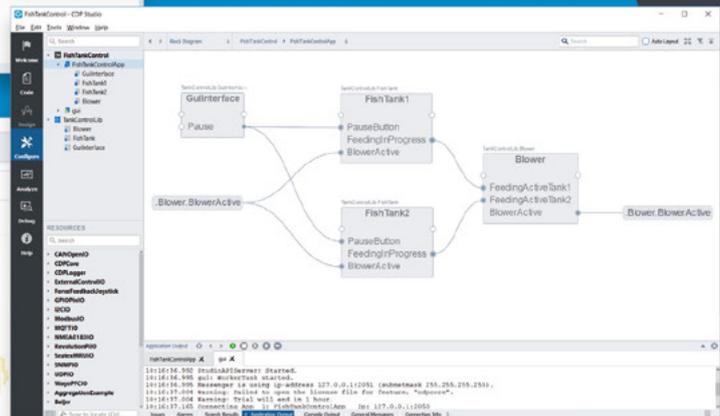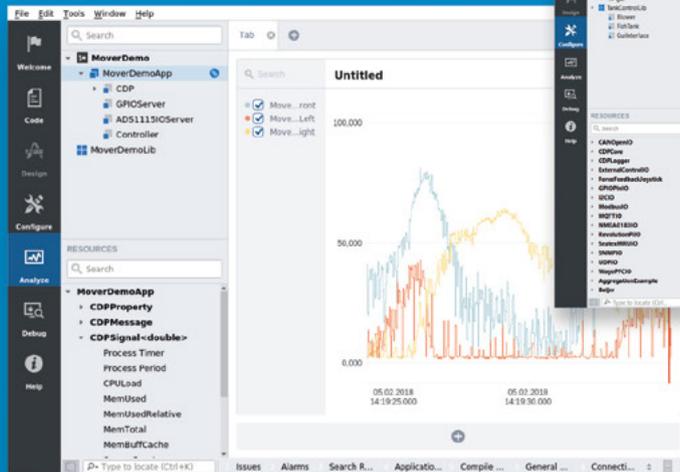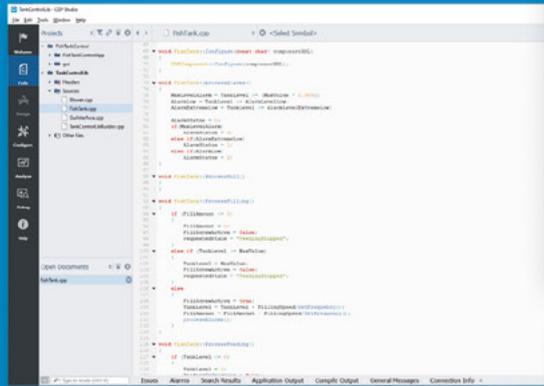
Tree views
with C and GTK

Use the GtkTreeView widget to display information and user input options

MAKER

Simon Long

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

rpf.io

As mentioned in part five, using a list store for a combo box can sometimes be overkill – although in some circumstances, it is exactly the right thing to do! But the situation for which a list store is really designed is as the data store for a widget called a GtkTreeView – a table of data arranged in columns. The table can show images as well as text, and can even contain other widgets, such as check buttons, alongside the data.

Creating a tree view

To demonstrate this, we're going to create a new list store with two entries per row, a text string, and an icon. Here's the code to do that:

```
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);
```

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

```
  g_signal_connect (win, "delete_event",
G_CALLBACK (end_program), NULL);

  int pos = 0;

  GtkListStore *ls = gtk_list_store_new (2,
G_TYPE_STRING, GDK_TYPE_PIXBUF);
  GdkPixbuf *icon = gtk_icon_theme_load_icon (
gtk_icon_theme_get_default (), "dialog-ok-apply",
32, 0, NULL);
  gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 1", 1, icon, -1);
  gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 2", 1, icon, -1);
  gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 3", 1, icon, -1);

  GtkWidget *tv = gtk_tree_view_new_with_model (
GTK_TREE_MODEL (ls));

  GtkCellRenderer *prend =
gtk_cell_renderer_pixbuf_new ();
  GtkCellRenderer *trend =
gtk_cell_renderer_text_new ();

  gtk_tree_view_insert_column_with_attributes (
GTK_TREE_VIEW (tv), -1, "Icon", prend, "pixbuf",
1, NULL);
  gtk_tree_view_insert_column_with_attributes (
GTK_TREE_VIEW (tv), -1, "Option", trend, "text",
0, NULL);

  GtkWidget *tbl = gtk_table_new (2, 2, FALSE);
  gtk_table_attach_defaults (GTK_TABLE (tbl), tv,
0, 2, 0, 1);
  gtk_table_attach_defaults (GTK_TABLE (tbl),
btn, 0, 1, 1, 2);
  gtk_container_add (GTK_CONTAINER (win), tbl);
  gtk_widget_show_all (win);
  gtk_main ();
}
```
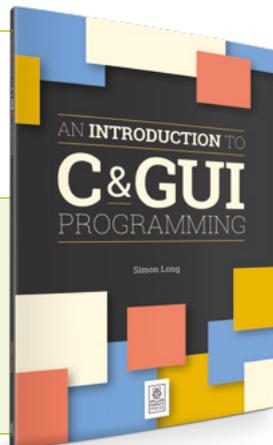
First, we call `gtk_list_store_new` to create a two-column list store; column 0 contains a text string, as in the previous part, while column 1 contains a pixbuf (short for 'pixel buffer'), which is used to store an image.

```
    GtkListStore *ls = gtk_list_store_new (2,
G_TYPE_STRING, GDK_TYPE_PIXBUF);
```

We then create a GdkPixbuf with a small image in it – in this case, we load an icon from the current icon theme into it, but a GdkPixbuf can store any sort of image you choose to display.

```
    GdkPixbuf *icon = gtk_icon_theme_load_icon (
gtk_icon_theme_get_default (),
"dialog-ok-apply", 32, 0, NULL);
```

We add three rows to the list store – in each case, we put a text string into column 0 of the row, and the pixbuf we created into column 1.

```
    gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 1", 1, icon, -1);
    gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 2", 1, icon, -1);
    gtk_list_store_insert_with_values (ls, NULL,
pos++, 0, "Option 3", 1, icon, -1);
```

We now need to create a GtkTreeView to display the list store. The first step of this is similar to the creation of a GtkComboBox in the previous part – we associate the new GtkTreeView with the GtkListStore we just created.

```
    GtkWidget *tv =
gtk_tree_view_new_with_model (
GTK_TREE_MODEL (ls));
```

As with the combo box, we need to create cell renderers for all the types of data we want to display; in this case, we have two – a text renderer and a pixbuf renderer.

```
    GtkCellRenderer *prend =
gtk_cell_renderer_pixbuf_new ();
    GtkCellRenderer *trend =
gtk_cell_renderer_text_new ();
```

Finally, we insert the columns we want to see into the tree view.

```
    gtk_tree_view_insert_column_with_attributes (
GTK_TREE_VIEW (tv), -1, "Icon", prend, "pixbuf",
```

```
1, NULL);
    gtk_tree_view_insert_column_with_attributes (
GTK_TREE_VIEW (tv), -1, "Option", trend, "text",
0, NULL);
```

Each call to `gtk_tree_view_insert_column_with_attributes` takes the tree view itself as the first argument. The second argument is the position of the column – this can either be the actual position, numbered from 0 at the left-hand side of the table, or as here, -1, which indicates that the column should be added as the next column; that is to the right of the rightmost column already in the table.

> ❝ The situation for which a list store is really designed is as the data store for a widget called a GtkTreeView ❞
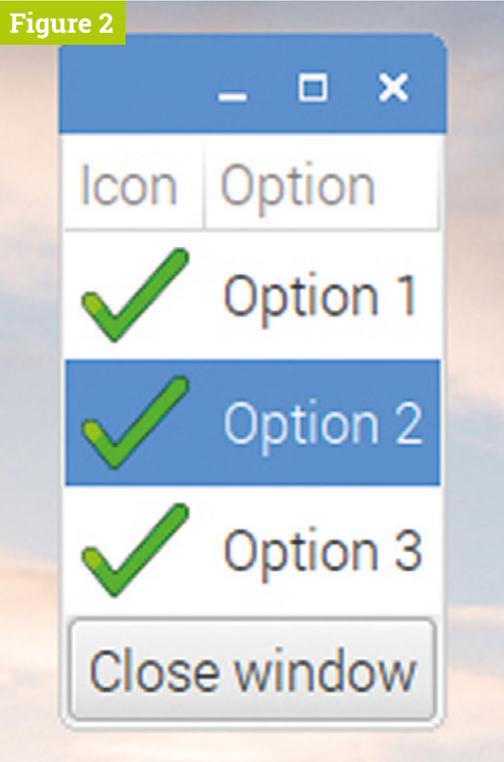
The next parameter is the title for the column (which can be shown at the top of the table), and then the renderer we've created to be used for the column.

The final arguments are passed to the cell renderer itself, and set an attribute of the renderer

Figure 1
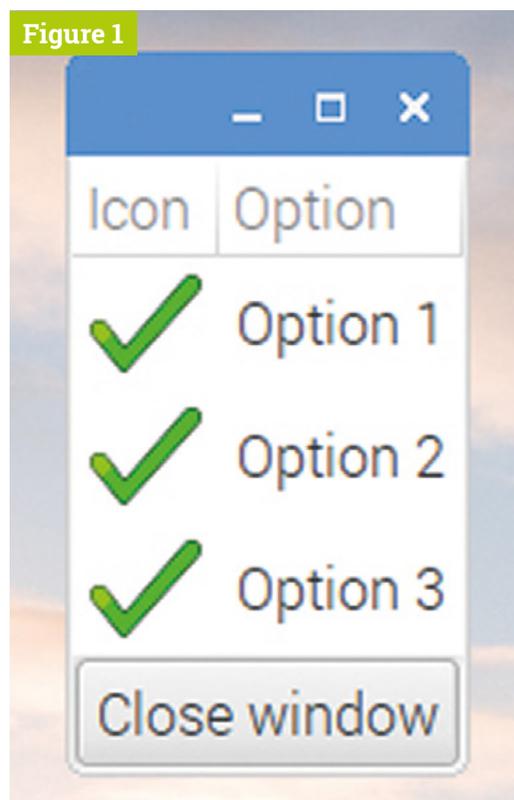
Figure 2

▶ **Figure 2** A GtkTreeView with a row selected by the user

to the data in a certain column of the list store – in the first instance we set the `pixbuf` attribute of the pixbuf renderer to the data in column 1, and in the second we set the `text` attribute of the text renderer to the data in column 0. (Because it is possible to set multiple attributes on some renderers, the final argument is a `NULL`, to indicate that there are no more attributes to set.)

Build and run the code, and see what happens – you should see something like **Figure 1**.

Getting input from a tree view

A tree view is useful as a display of information, but it can also be used to read user selections. If you click on one of the rows in the table, it will be highlighted – you can also move the highlight up and down through the table with the arrow keys on the keyboard (**Figure 2**).

Whenever the selected row is changed, the cursor-changed signal is generated by the tree view, so this can be used to call an appropriate callback:

```
    g_signal_connect (tv, "cursor-changed",
  G_CALLBACK (row_selected), NULL);
```

The callback should be as follows:

```
  void row_selected (GtkWidget *wid,
  gpointer ptr)
  {
    GtkTreeSelection *sel;
```

```
    GtkTreeModel *model;
    GtkTreeIter iter;
    char *option;

    sel = gtk_tree_view_get_selection (
GTK_TREE_VIEW (wid));
    if (gtk_tree_selection_get_selected (
sel, &model, &iter))
    {
        gtk_tree_model_get (model, &iter, 0,
&option, -1);
        printf ("The selected row contains the
text %s\n", option);
    }
  }
```

Finding out which row is selected requires an iterator – a data structure which stores the location of a row in a list store. First we need to get the GtkTreeSelection for the tree view – this stores which row (or rows) are currently highlighted.

```
    sel = gtk_tree_view_get_selection (
GTK_TREE_VIEW (wid));
```

From a GtkTreeSelection, we then get the GtkTreeIter and GtkTreeModel associated with it. The tree model is the list store which is providing the data for the tree view, and the iterator is the actual row selected.

```
    if (gtk_tree_selection_get_selected (sel,
&model, &iter))
```

(This function returns `FALSE` if there is no valid selection – it's important to check for this as otherwise the next line will cause a crash if no row is selected.) Finally, we can get the actual values from the model and iterator.

```
    gtk_tree_model_get (model, &iter, 0, &option,
-1);
```

As when setting a value in a list store, the arguments to this function, after the model and iterator, are a list of paired values, the first of each pair being the column to be read, and the second being a pointer to which the data should be written. Multiple values can be read from the same row with a single call, so -1 is used for the column value of the final entry in the list, to show there are no more values to read.

In this case, a pointer to the text in the first column of the selected row is returned as `option`. M

STRATO PI CM DUO

Dual SD card: for the first time on an industrial Raspberry Pi CM based device!

Strato Pi CM Duo is the perfect solution for high-reliability systems - featuring two distinct SD cards with high-speed switching matrix for separate OS/data storage, redundancy and in-field full-system upgrades.
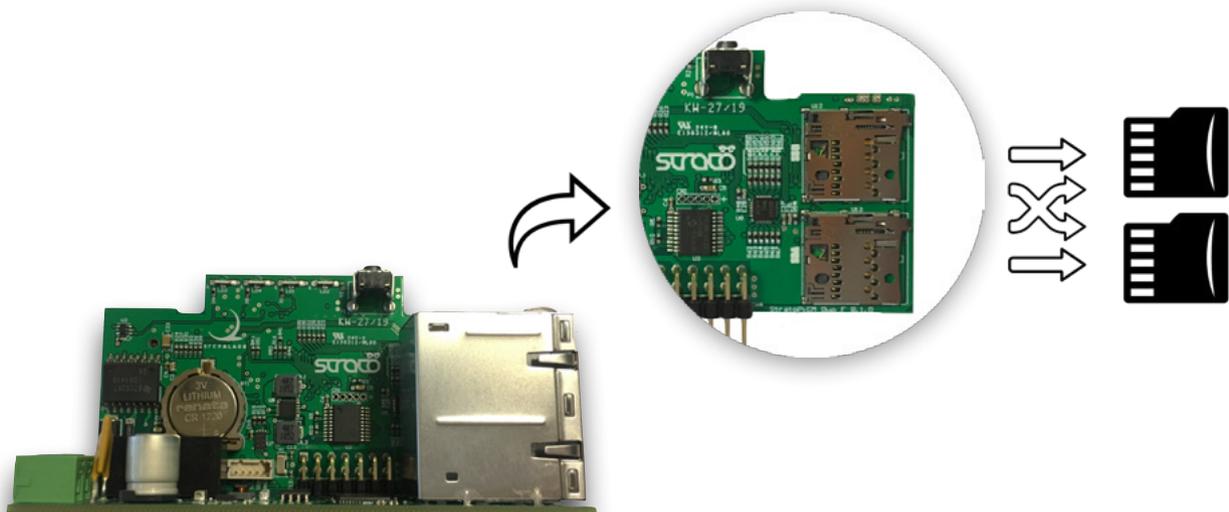
Fault tolerance is further enhanced by:
- Regulated wide-range power supply
- Hardware watchdog with boot SD switch control
- Power-controlled USB ports

Additional features:
- Opto-isolated RS-485 serial interface
- Real-Time Clock with a replaceable backup battery
- Secure element

All of this in a compact DIN-rail case!

SFERALABS

www.sferalabs.cc

Part 04

Add game music
and SFX in PICO-8

Use PICO-8's built-in synthesizer to create space-age sound effects and get your pulse pounding by composing some bangin' retro chiptunes

MAKER

Dan
Lambton-Howard

Dan is an independent game designer based in Newcastle upon Tyne, where he is lucky enough to make games for his PhD.

@danhowardgames

DOWNLOAD THE FULL CODE:

magpi.cc/fB0ksD

You'll Need

- PICO-8
  magpi.cc/pico8
- Raspberry Pi
- Keyboard and mouse

P ICO-8 comes with a comprehensive and surprisingly versatile set of music and sound effect editors. Featuring eight synth instruments and seven built-in effects – including slides, vibrato, and arpeggiators – it is an accessible and easy way to make everything from retro bleeps and bloops, up to entire chiptune symphonies. We'll walk you through each of the editors as we make some simple sound effects for our space shooter game. Then, we'll put some loops together to explore the basics of writing a retro techno soundtrack. All together now!

01 An anatomy of sound

Let's start with understanding how PICO-8 handles audio. A PICO-8 cartridge can hold up to 64 sound effects (SFX). These SFX are used for everything from gameplay sounds, like the 'boing' you hear when jumping in a platformer, to backing tracks and incidental music. Each SFX consists of 32 notes and each note contains information about its pitch (how high or low), its instrument (the overall timbre and aural quality), its volume (you guessed it), and effect (how that note is altered or modified). You can make a lot of noise with eight little bits!

02 The editors

Load up the space shooter game we've been building, and switch to the SFX editor. The first thing to know is that there are two modes. Pitch mode allows you to plot notes with the mouse and is best for creating sound effects, whilst tracker mode is more suited to music. Toggle between

them with the top-left button. Generally, they both show the same information, and both have the same speed, loop, and instrument selectors. The number at the top left is the index of the current SFX you are editing. Toggle to index 0 and let's get started.

03 Pew pew pew!

We think our space shooter could do with an old-school laser sound, don't you? To achieve this in pitch mode, draw a descending series of about eight notes (**Figure 1**). You can use any instrument you want, but we've alternated between square and pulse waves to give a little retro warble. We've also used noise for the last two notes, to create a sense of impact. Hit the **SPACE** bar at any point to play your SFX. We've chosen our SFX speed as 2 (1 is the quickest), but play around to see what works.

> ❝ We think our space shooter could do with an old-school laser sound, don't you? ❞

04 Impactful audio

To see how your laser sound works in game, add `SFX(0)` after the call to `create_laser()` around line 35. Reload your game to hear it in action. It really is that easy. Now, repeat this process to make an explosion sound. For ours, we've used around ten notes, starting with deep square wave, a surge of noise, then finishing on higher-pitched saw wave. This creates an initial impact noise and ends on a retro, dopamine-inducing chime. Visit

**Figure 1** Here's the same laser zap sound viewed in pitch mode (left) and tracker mode (right)

the project's GitHub repository (**magpi.cc/fBoksD**) to hear it in action. Now, add your explosion SFX in code alongside wherever we create explosions.

05 **Tracker mode**
Now we've got some retro zaps and booms, let's move onto the soundtrack. Toggle the editor to tracker mode. This mode is better for writing music as it is easier to see the relative spacing and timings of notes, as well as to adjust individual properties of each note. You will also be able to see the grey effect selector buttons in this view. Notes might look like a baffling array of letters and numbers, but are actually quite simple. Each line shows the pitch (A–G#), octave (0–5), instrument (0–7), volume (0–9), and effect (0–7) of a note.

06 **Kick and snare**
To write notes in tracker mode, you need to actually play notes with your keyboard, just like a piano – see the layout in **Figure 2** (overleaf). Have a little play around; you can hit **SPACE** to play your creation and if you set the loop controls to 0 and 32, it will play through all 32 notes before restarting. Once you're comfortable, plot a simple drum beat. Use **C1053** for a nice kick-drum sound, and **C3655** for a punchy snare. We've gone for a pretty standard rock beat with a speed of 12.
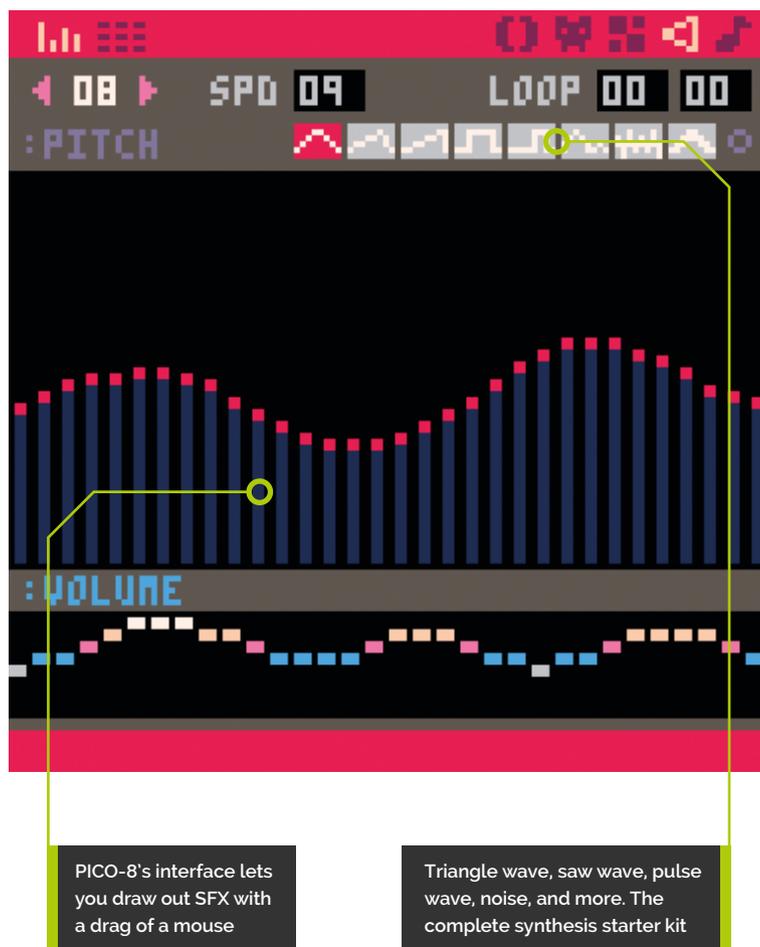
PICO-8's interface lets you draw out SFX with a drag of a mouse

Triangle wave, saw wave, pulse wave, noise, and more. The complete synthesis starter kit
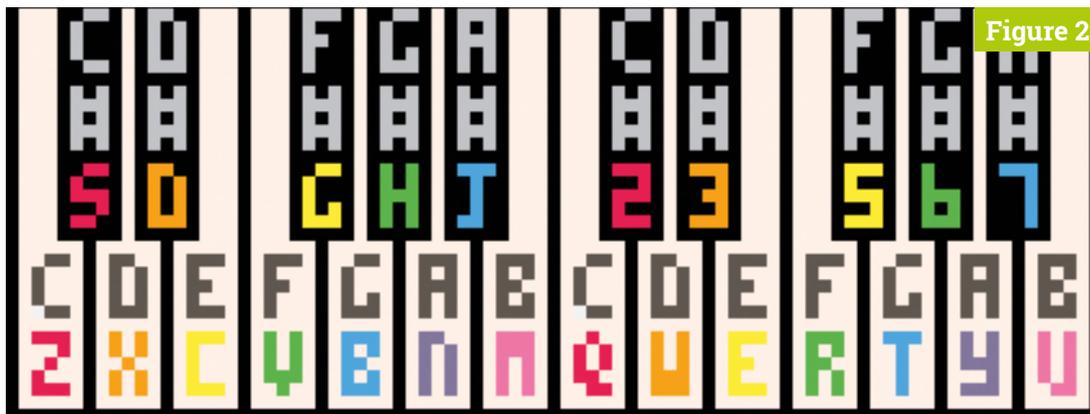
**Figure 2** Use this handy reference and soon you'll be playing your PICO-8 like Beethoven

07 The full kit and caboodle

Next, you might want to make a few variations on this basic beat and save these as new SFX. To make things easier, you can copy and paste individual notes or whole sections. Use **CTRL+A** to select an entire SFX. To help, a high note with fade out (5) makes a great hi-hat, and mid notes with the drop effect (3) work for toms. Layer it up! You can keep the SFX playing as you add notes and play around. Once you've made three or four variations, we'll use the music editor to put them together into a composition.

08 Seeing patterns

The music editor is where the magic happens. It's where individual SFX are layered and combined to make music. Switch to it now, and take a look. You'll see a list of patterns at the top of the screen. Each pattern can contain up to
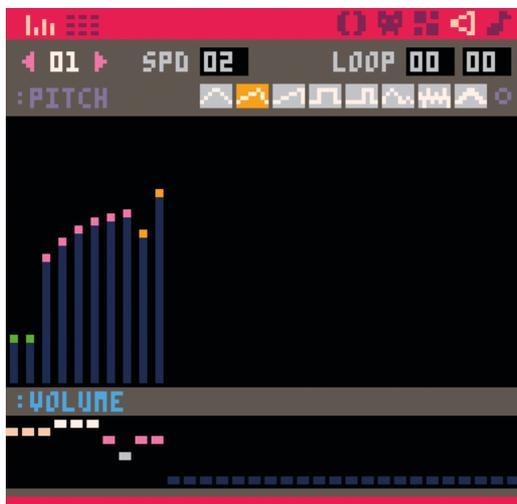
It might not look like much, but hit the **SPACE** bar and you'll hear a perfect retro explosion, boom!

four SFX, which are shown in the boxes below, and when it has finished playing, will move onto the next pattern. You can also use the controls on the top-right to set patterns to stop, or loop back to a previous pattern.

09 Tub thumping

Start adding your drum beat SFX to the leftmost slot of the first few patterns. You can see the index above the slot. Hit **SPACE** on the first pattern and you should hear your beat played, followed by the beat in the next pattern and so on. We've started with minimal drums for patterns 0–3, moved into fuller percussion for 4–7, and built our composition from there. Now, obviously, there is more to music than drums (though it's a good place to start), so we'll need to create some bass, rhythm, and melodies next. Cue the music!

> ❝ Try playing around with volumes and effects to make notes more interesting and dynamic ❞

10 Thank you for the music

You probably don't need us to tell you that low notes work well for bass, and high and mid octaves work for melodies and harmonies. For now, concentrate on creating an SFX for each. You can start adding these (and looping them) in your music editor to see how they work together. Try playing around with volumes and effects to make notes more interesting and dynamic. We've created a simple repetitive bassline to run through

▲ PICO-8's music editor in full swing. Here it is playing our composition on three of the four available audio channels

◀ Composers like Chris 'Gruber' Donnelly have produced entire albums on PICO-8, a testament to the versatility of the tool

most of the composition, and a catchy melody for the dynamic peaks. Download the cartridge from the project's GitHub repository to hear it.

11 Stops and starts

So now your masterpiece is composed, we need to get it into the game. Simply add `music(0)` to `_init()` and reload your game. Et voilà, your tune is now blasting out into space. For extra brownie points, you can add `music(-1)` to your code when game-over is triggered. This will stop the music from playing when the player is defeated. Or even better, why not compose a whole new game-over theme tune? Attempt to capture the sombre feeling of humanity's last hope being snuffed out, in 8-bit.

12 Deep cuts

We've barely scratched the surface of what you can do with sound effects and music with PICO-8. But you should have a good enough grasp of the tools to start exploring the creative potential of this little virtual console. If you want to go deeper, we highly recommend Chris 'Gruber' Donnelly's series of video tutorials: **magpi.cc/OngdaS**. To add to the possibilities, a recent PICO-8 update introduced SFX instruments, allowing makers to define and experiment with their own instruments on top of the basic eight. Plenty to get your teeth stuck into. M

Top Tip 👍

Only four channels

PICO-8 only has four audio channels, meaning only four SFX can be playing at any one time, including in music.

Access Raspberry Pi over
the internet with VNC

Set up Raspbian so you can remote access Raspberry Pi from
any other machine on your network or over the internet

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi*. She has a Raspberry Pi hooked up at home that she uses VNC to access, wherever she is in the world.

magpi.cc

There will be times when you can't or don't want to switch to your Raspberry Pi. Perhaps you're using another computer, or your Raspberry Pi is out of reach, behind your TV or a nest of cables. Or you're on the road and have left your Raspberry Pi at home.

Fortunately, with VNC (Virtual Network Computing) – free for non-commercial use and built into the Raspbian operating system – you can access Raspberry Pi remotely from any other computer, tablet, or smartphone. Either on your home network, or via the internet.

In this walkthrough, we'll be using VNC Viewer and VNC Server to connect on the fly to a Raspberry Pi from a Windows PC and an Android smartphone.

You'll Need

> Raspberry Pi with Raspbian

> Network and internet connection

> VNC Server and VNC Viewer

01 Enable VNC

First, Make sure both your Raspberry Pi and the other computer you're going to use are connected to the same network.

In Raspbian, click the applications menu icon (raspberry) at the top-left of the screen and select Preferences > Raspberry Pi Configuration.

▶ Activate VNC Server on your Raspberry Pi with Raspberry Pi Configuration tool. Choose the Interfaces tab and set VNC to Enabled

If you haven't changed Raspbian's password from the default 'raspberry', now is a good time to do so. Click the Change Password button and enter a new one.

Now click the Interfaces tab, and set the radio button next to VNC to Enabled. Click OK.

A VNC button appears in the top right of the screen, at the end of the menu bar. Click it to open VNC Server.

Your IP (internet protocol) address will appear in the VNC Server Window (below Connectivity). Note down the four numbers; they enable you to locate your Raspberry Pi over the network.
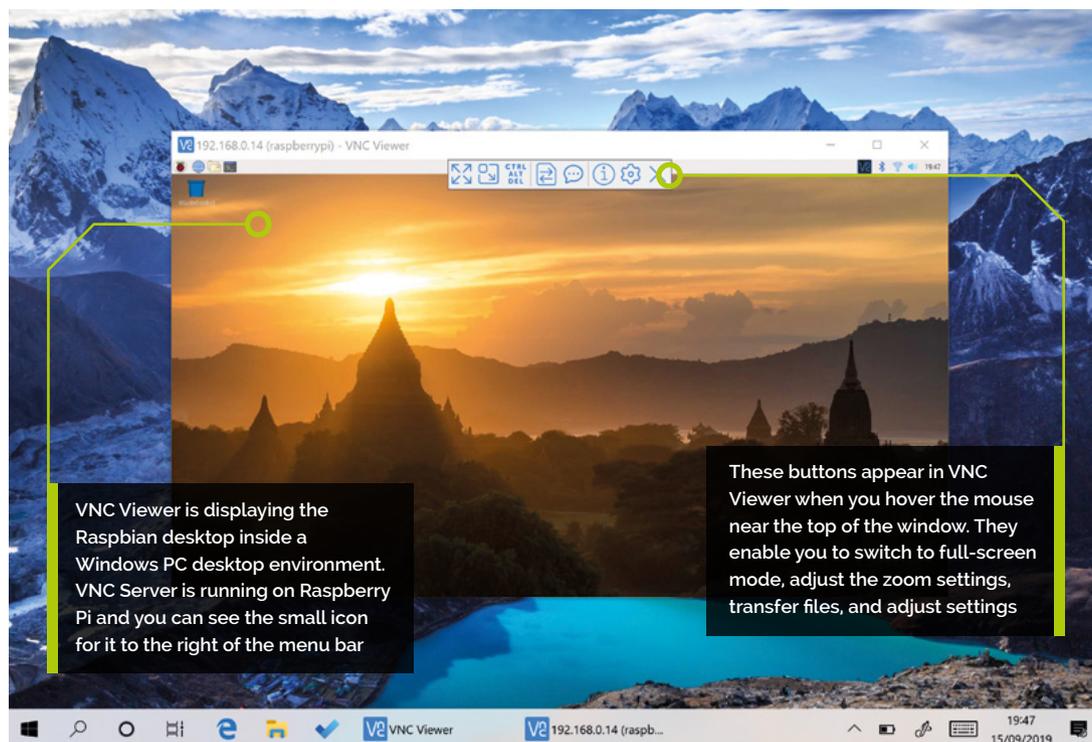
02 Open VNC Viewer

You're now ready to connect to your Raspberry Pi from another computer. We're going to use a Windows PC, but you can connect from a Mac, Linux PC, or even another Raspberry Pi on the same network.

Because we're using VNC Server on Raspberry Pi, it makes sense to use VNC Viewer on the Windows PC. VNC Viewer is available for macOS, Linux, Android, and iOS platforms, along with a web interface. Head to RealMac (**magpi.cc/hTpNBm**) and click Download VNC Viewer. Install and open the software.

03 Connect to Raspberry Pi

Enter the IP address of your Raspberry Pi (the four numbers displayed in VNC Server) into the search bar of VNC Viewer (the part saying 'Enter a VNC Server address or search'). Press **RETURN** to connect to Raspberry Pi.

The first time you do this, a window will appear with a warning: 'VNC has no record of connecting

VNC Viewer is displaying the Raspbian desktop inside a Windows PC desktop environment. VNC Server is running on Raspberry Pi and you can see the small icon for it to the right of the menu bar

These buttons appear in VNC Viewer when you hover the mouse near the top of the window. They enable you to switch to full-screen mode, adjust the zoom settings, transfer files, and adjust settings

to this VNC Server so its identity cannot be checked'. Click Continue.

You need to enter the username (typically 'pi') and password for your Raspberry Pi. Click on Remember Password and then OK to connect to Raspbian.

04 Remote control

A window appears on your Windows PC displaying the Raspbian desktop interface from your Raspberry Pi. Move the mouse around the window and you'll see the mouse moving around on your Raspberry Pi. You can now use this window to control your Raspberry Pi over the network (just as if you were sitting in front of it).

Hover the mouse at the top of the VNC Viewer window and a menu will drop down from the top of the screen. To the left of the Options is Enter Full Screen; click it and the preview window will take over the screen. Your Raspberry Pi display may not fit your PC display, so click the Scale button in the menu (so it's set to Scale Automatically).

You will now be able to open apps and use your Raspberry Pi as if you were working directly on the small computer.

> **You need to enter the username (typically 'pi') and password for your Raspberry Pi**

05 Looking at properties

You can access Properties from the menu at the top of the VNC Viewer preview window, but we're going to close down the preview window to access it from VNC Connect. Click End Session in the drop-down menu and Yes to disconnect.

VNC Viewer now displays a small preview of your Raspberry Pi desktop in the Address Book, along with the IP address. You can double-click this at any time to reopen the connection – but first, right-click and choose Properties.

Enter 'Raspberry Pi' in the Name window; this will make your screen friendlier. Now click Options. The Picture Quality setting is set to Automatic by default. If you have a slow connection, try setting it to Low; alternatively, if you have a fast connection, set it to High.

Finally, take a look at the Expert tab. Here you can find a variety of settings for all aspects of the system. Scroll down to find FullScreen and set the drop-down menu below from False to True.

Top Tip 👍

Technical information

You can find a wealth of detailed information on the RealVNC website. Take a look at this Technician's Guide PDF: **magpi.cc/JiSeZX**
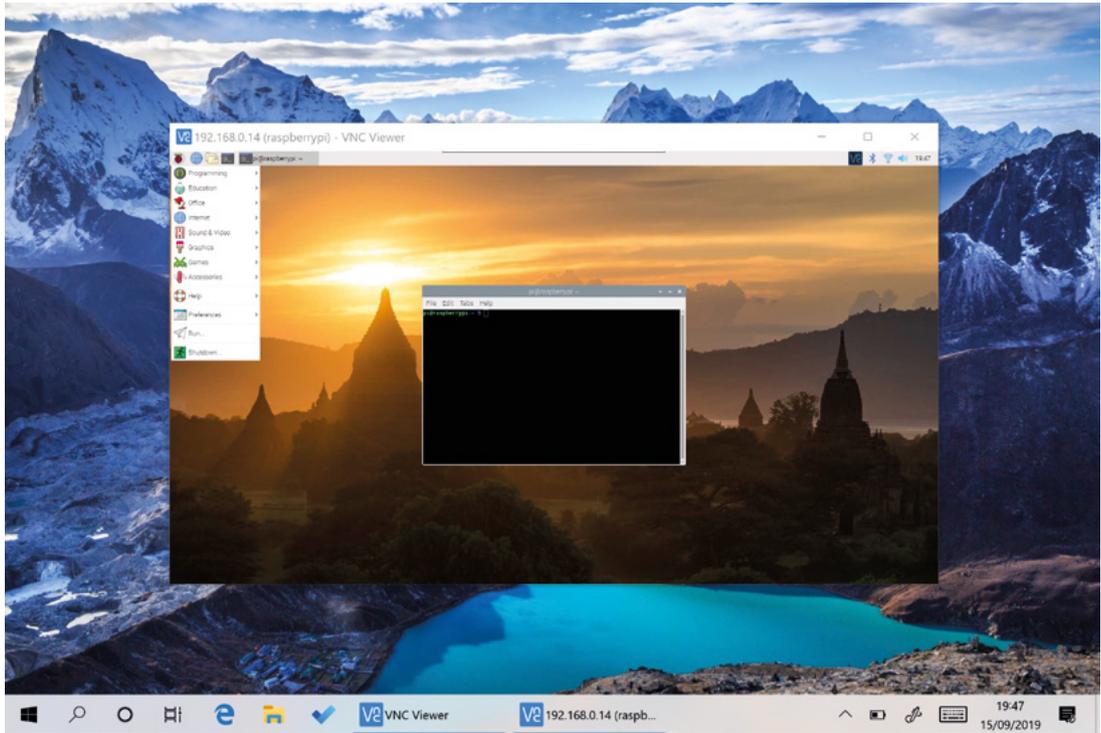
▶ VNC Viewer displays the Raspbian desktop interface remotely

Top Tip 👍

Speed it up

If your remote Raspberry Pi feels sluggish, hover at the top of the VNC Viewer window and click the cog on the menu that appears, then reduce the picture quality on the Options tab.

Now when you open Raspberry Pi in VNC Viewer, the preview window will automatically be in full-screen mode. Choose the options you prefer and click OK to set them.

06 Get online

So far we've been working on our local network. With a RealVNC account, you can access your Raspberry Pi across the wider internet.

Open VNC Viewer on your PC and click the Sign In button on the top left. It's most likely that you don't have an account already, so click 'Sign up online'. This will take you to the RealVNC website.

Enter your email address and click the 'I'm not a robot' checkbox. Now create a password. Please choose a long and complex password that is not easy to guess. We suggest using a Diceware

▶ Use File Transfer to send and retrieve files from your Raspberry Pi

> 🟢 With a RealVNC account, you can access your Raspberry Pi across the wider internet 🟢

password generator (**rempe.us/diceware**). Follow the rest of the setup process and click on the email authentication when it arrives.

07 Sign in

Now you need to sign in with the same account on both VNC Viewer on your PC and VNC Server on Raspberry Pi.

The Sign In window should still be displayed in the Sign In window in VNC Viewer on your PC. Open VNC Server on your Raspberry Pi and click Sign In to enable the cloud connectivity link (displayed just below your IP address under Connectivity).

Return to VNC Viewer on your PC. You'll still see a Raspberry Pi window in the Address Book section, but below it you'll see an option called Team (it will appear with your name). Click this to see your Raspberry Pi again. However, this is the connection using your VNC Connect account. Double-click it to launch into your Raspbian desktop and remote control.

Top Tip 👍

Got a Mac?

Mac users must change the Authentication protocol to VNC password to connect using macOS's native tools.

The difference? This account works remotely, and you can use this from other networks.

08 Send files

You can send and receive files from your Raspberry Pi and PC. We've created a blank test document in our **Documents** folder called **test.txt**.

To send the file to your Raspberry Pi, access Raspberry Pi using VNC Viewer. From the VNC Viewer preview window, click the menu at the top of the screen and choose Transfer Files.

Now click Send Files in the VNC Viewer – File Transfer window. Use the file picker to choose a file on your PC and click Open. The file will be sent to the desktop of your Raspberry Pi. A File Transfer window will appear with 'Download complete'; close it.

09 Retrieve files

You can retrieve a file from your Raspberry Pi using VNC Viewer, although the process is slightly different. This time, right-click the VNC Server icon in Raspbian's menu bar. Choose File Transfer from the drop-down menu to open the VNC Server – File Transfer window.

Click Send Files and choose a file on your Raspbian file system. Click OK to send the file to the desktop on your Windows PC.

Your Raspberry Pi is now set up for remote access. It's now perfectly possible to remove the screen and keyboard from your Raspberry Pi and leave it on the network. It'll be there waiting for you to connect from your PC whenever you're ready.

10 Android and iOS

It is also possible to connect to Raspberry Pi remotely from your smartphone. Use the VNC Viewer app for Android or iOS – download it from the Google Play Store or Apple App Store. Open and sign in using your VNC Connect account email and password.

You will see your Raspberry Pi under the Team menu. Click it and enter the username and password for your Raspberry Pi.

When it first launches, you'll need to run through the 'Control the computer' process. Click on Next and it will open the 'How to control' window. This screen outlines how to perform gestures such as mouse clicks from the touchscreen. Close this window and start using Raspberry Pi from your phone.

Use the touchscreen to move the pointer around. A virtual keyboard is accessed by tapping the Keyboard icon at the top of the app window.

You can now access Raspberry Pi from anywhere, including your phone. A great solution for remote monitoring. **M**

JOIN FOR 12 MONTHS AND GET A

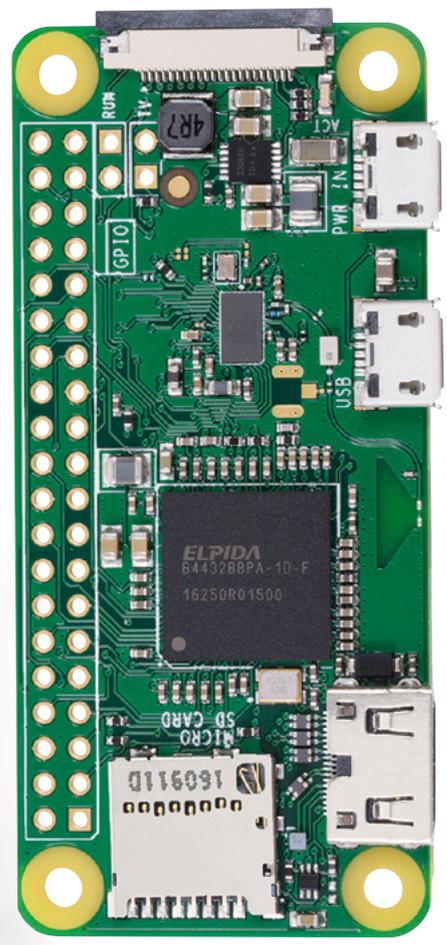# FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

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

WORTH
£20

▷ 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

⬤ Buy now: magpi.cc/subscribe

WIN! 1 OF 10 RASPBERRY PI 4 CASES

The MagPi
Issue 85 | September 2019 | magpi.cc | *The official Raspberry Pi magazine*

RASPBERRY PI 4
YOUR NEXT DESKTOP PC
Can you do everything with a $55 computer?

7-DAY TEST DIARY

BACK TO SCHOOL
Use Raspberry Pi to learn & teach coding

SUPER-FAST NAS DRIVE
Build a gigabit file server with Raspberry Pi

PLUS!
LEARN WEB DEVELOPMENT
MAKE AMAZING 8-BIT SPRITES
ROCKYBORG ROBOT REVIEWED!

40 PAGES OF TUTORIALS

The MagPi
YOUR OFFICIAL RASPBERRY PI MAGAZINE

RASPBERRY PI
STARTER GUIDE

30+ PAGES OF TUTORIALS

SUBSCRIBE on app stores
From **£2.29**

 Available on the
App Store

GET IT ON
Google Play

10 SPOOKTASTIC

HALL

STOMACH SHOT

MAKER

Luis Martin Nuez

magpi.cc/KiVVnM

Characters:
Zombie, skeleton

👻 **We interviewed Luis about this blood-curdling costume idea a few years ago.** It's very basically a Raspberry Pi with a Camera Module that displays what it sees on the screen live. The camera is attached to his back, with the screen on his stomach, and the preview mode is used to create the illusion that he has been shot through the stomach. Grisly.

As this is an older project, some of the parts may be unnecessary or easy to upgrade to better versions. We also recommend you use a screen solution that is powered by your Raspberry Pi so that you can use a portable battery charger instead.

▲ Luis created a latex frame of gruesome flesh to give the screen more realism

OWEEN

COSTUME PROJECTS

Use Raspberry Pi wearable tech to make amazing costumes this Halloween

GAME BOY COSTUME

MAKER

MikeHandidate

magpi.cc/yLdPXL

Characters:
BMO, robot, any other retro video game console

🎃 **The point of a Game Boy is that it's small and you can fit it into your pocket.** Turning yourself into a portable console as a costume sort of ruins that aspect of it, but at least you are then still portable.

The costume itself is very simple, with a couple of cardboard sheets painted to make it look like the classic Game Boy Color. It's got a pretty big screen though, so it may need a bit more power than your standard mobile battery. Having the gloves work as custom controllers adds an extra level of complexity to the build; however, GPIO-based buttons are supported in RetroPie, so they shouldn't be too hard to do.

▲ Instead of poking at Mike, you use controls on his gloves

ROBOSUIT

MAKER

Estefannie
Explains It All
magpi.cc/HPmPZx

Characters:
Darth Vader,
Power Loader,
Gort, Teletubby

🔊 **This build combines many things we enjoy.** Robots, simple electronics, googly eyes, flashing lights, and really fun, low-budget DIY construction.

Estefannie used several technologies to power this suit, including Arduino supplies to power LEDs around the suit, a voice changer to giver herself a more robotic persona, and a Raspberry Pi to power the screen on the chest with several functions which are powered by a series of buttons. The jaw even moves depending on how she speaks!

And of course, it's built using cardboard boxes and silver tape, meaning just about anyone can make the body.

COSPLAY & COSTUME SUPPLIES

🔊 Need some wigs, fabric, or armour-making supplies? Check out our friends at Coscraft (**coscraft.co.uk**) for a huge range of costuming and cosplay supplies for just about any wearable build.

HALLOWEEN VOICE CHANGER

MAKER

Olivier Ros
magpi.cc/PXEiJX

Characters:
Darth Vader,
Dalek, robot,
alien, Judge Doom

🔊 **Voice changers/amplifiers make for excellent parts in a scary costume, especially when installed in a scary skull mask like this.**

While seemingly a simple build – and it generally is – there's a few neat parts to this project. You need to remove the microphone part of an audio board add-on, and Olivier also teaches you how to correctly install the built device so that the microphone is isolated and to reduce feedback. There's even a button to change between voice types!

▶ Modelling clay is used to fit the device to the mask, and to reduce feedback

DOC BROWN

MAKER

Carl Monk

magpi.cc/giHLeF

Characters:
Darth Vader, Clock King, Rebel pilot, robot

👤 **A good Doc Brown costume can mostly be made with just a lab coat, cool shades, and a frizzy wig.** A great Doc Brown costume includes custom-made props from the *Back to the Future* series, such as the flux capacitor that makes time travel possible, and the time circuit for setting your destination.

The circuit part of the build is fairly simple, making use of several seven-segment displays to ape the style of the films, and they're housed in a simple box to complete the look. Carl has an exhaustive build tutorial on his website as well, if you wish to pick it apart.

◀ Still images don't really do it justice, but the effect in real life is excellent

NEOPIXEL SANS UNDERTALE EYES

MAKER

Mel Nurdin

magpi.cc/ipwbXp

Characters:
Replicants, Dr Manhattan, robots, Super Saiyan Goku

👤 **This full-on cosplay has an electronic secret – simple LED lights for eyes that can change into flashing circles of doom.** For those that know of the game Undertale, the light patterns represent how the character Sans looks when he unleashes his true power.

This simple build makes use of a Raspberry Pi Zero connected to a NeoPixel ring that is set behind some frosted acrylic. With the touch of a hidden pocket button, you can activate the eyes, turning them blue. Another press makes them flash yellow, and a final press turns them off.

Credit: Beth Dooner

STEAMPUNK TENTACLE HAT

MAKER

Derek Woodroffe

magpi.cc/iqDuUm

Characters:
Twi'lek, Andorians, other aliens, animal ears

🐚 **Have you ever wanted to have tentacles growing out of your head?** Probably not, we'd hope, but we can offer you the next best thing: a hat that makes it look like transdimensional tentacles are growing out of your head.

It's actually quite an ingenious build, using several servos and plastic tentacle skeletons to make the stocking-based suckers wiggle and move freely on top of the hat. Some latex, sealant, and paint bits later and you have a very weird and kinda scary hat.

◄ Each tentacle is controller by servos that run on a sine wave pattern to make them move more 'naturally'

POWER CONCERNS

🔥 Powering your costume can be tricky. Not everything can run off a Raspberry Pi, so you may find yourself needing a battery pack to provide a bit more oomph. In that case, you can also use a 5 V battery pack to power your Raspberry Pi via the GPIO pins – check out our full powering guide to find out how to do this safely: **magpi.cc/UiAJdk**

DISCO BALL

MAKER

Wolfie

magpi.cc/VsDiFi

Characters:
BB-8, a star, a planet

🔦 **For a costume that is essentially a series of LEDs, there is a lot going on here.** It's fully autonomous, playing a preprogrammed light display, and can connect to wireless LAN to ape the light show that the builder's house is playing as well.

The frame is welded together specifically to fit Wolfie's granddaughter; however, you can make it fit whomever you'd like if making it yourself. It uses some custom 3D-printed parts to keep a Raspberry Pi and some batteries in place – at the time of making, it would last several hours.

IRON MAN ARC REACTOR

MAKER

Team
MATRIX Labs
magpi.cc/PbaWFp

Characters:

War Machine,
robot, Optimus
Prime, Gundam

Voice modification and LEDs are cool, but how about voice-controlled (and reactive!) LEDs? This fantastic project from the MATRIX Labs team turns a cheap Iron Man costume into a party piece that is sure to attract a few nerds.

The voice-controlled magic is handled by Snips.ai (**snips.ai**), a customisable piece of tech that allows you to create voice assistants that work under your control. You can set keywords, train it to make sure it understand your voice, and then hook it up to your MATRIX board.
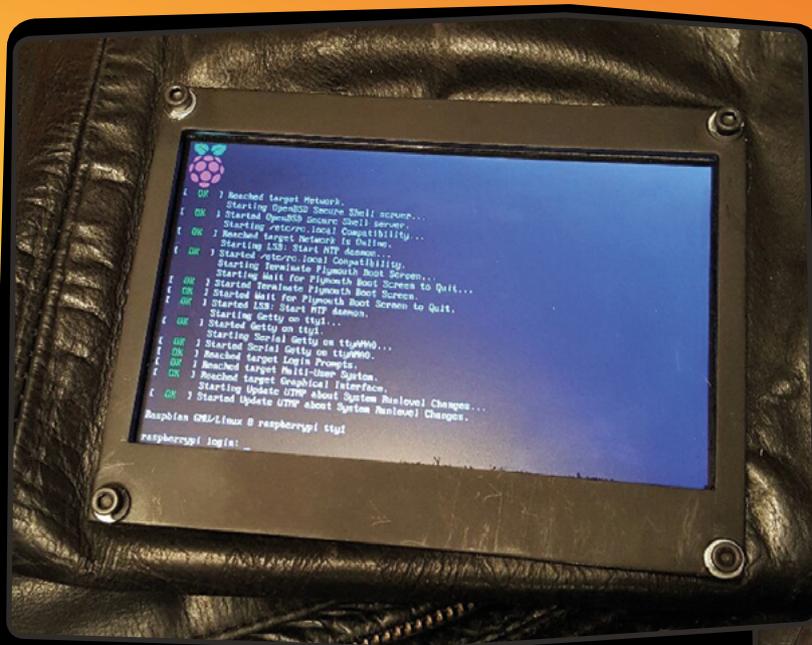
◄ The command-line interface is perfect as it looks like you're an eighties hacker, and it uses less power

THE JACKET 2.0

MAKER

CoreDump
magpi.cc/kUjSkV

Characters:

Buzz Lightyear,
Commander
Shepard, Fallout
character

The cyberpunk aesthetic will probably never go away, especially now that it's easier than ever to install electronics into clothes, a bit like 'The Jacket' that CoreDump likes to wear at Halloween.

It's a pretty standard leather jacket, albeit with some serious modifications. Not just style bits like neon lights and punk spikes, but also the fully functional Raspberry Pi installed into one of the sleeves for that 'hacking-on-the-go' look. There's also a hidden action camera in one of the spikes, perfect for first-person video fun.
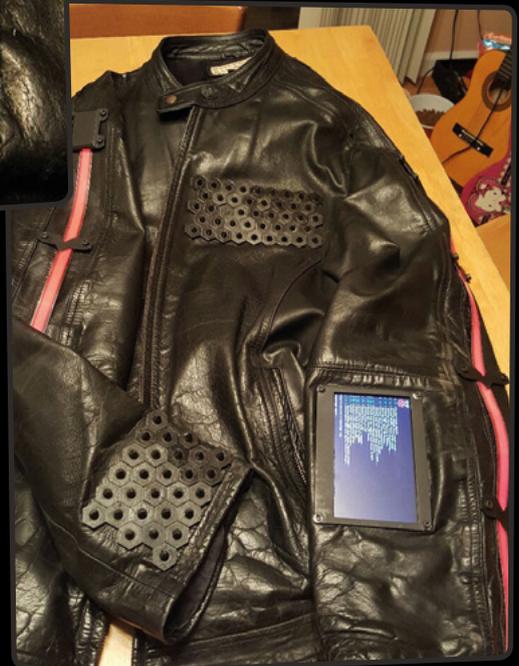
SPECS

SIGNAL TO NOISE RATIO:
130 dB (A-weighted)

DYNAMIC RANGE:
125 dB

TOTAL HARMONIC DISTORTION:
0.0003%

SAMPLING RATE:
Up to 192kHz

BIT RATES:
Up to 24-bit

DIMENSIONS:
125×125×70 mm

Pecan**Pi**

▶ Orchard Audio ▶ **magpi.cc/gedJyS** ▶ £325 / $399

We love DACs here at The MagPi. So what makes the PecanPi special? **PJ Evans** turns up the volume

Verdict

One for true audiophiles. If music is your passion or profession, and you're OK with the selection of available software, this might well be the best streaming device out there.

8/10

The market is crowded with audio streaming options available well under £100, so is there a place for another Raspberry Pi-based DAC (digital to analogue converter)? Presented in a brutalist black metal box, Orchard Audio's PecanPi is built with some of the highest-quality components available, aiming it squarely at the serious music lover. Inside you'll find a Raspberry Pi 3B and the DAC HAT itself. The outputs are analogue, with the two expected phono outputs accompanied by XLR connectors, making this suitable for professional setups.

Double DACs

Great care has been taken over component choice and the design uniquely features two of Texas Instruments' highest-specification DACs, one for each channel, with some of the industry's best-performing op-amps. The result is some impressive technical specifications, including total harmonic distortion of 0.0003% and a dynamic range of 125 dB. Even on-board WiFi has been disabled to prevent the radio interfering with
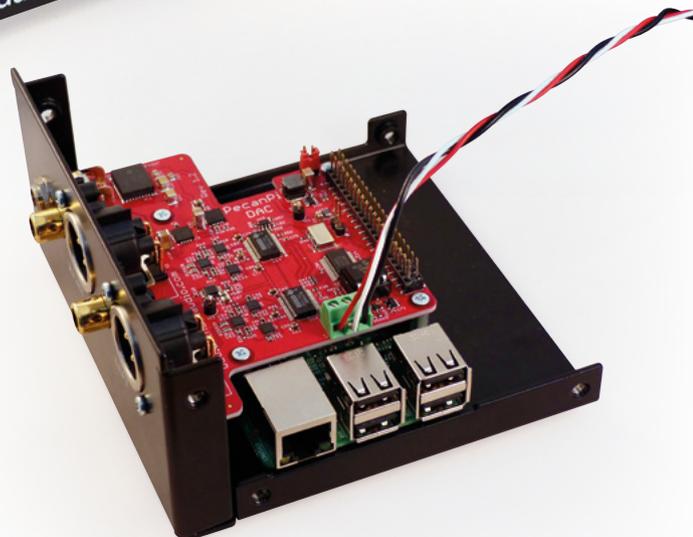
▲ The DAC is a HAT sitting on a Raspberry Pi 3B. WiFi is unavailable, to preserve sound quality, so you'll need Ethernet or an adapter

the sound processing (an optional WiFi/Ethernet bridge is available if needed).

We tested on a high-end amplifier with all its sound processing disabled, playing FLAC-encoded music. The result was crisp, detailed, and rich. We can see how this device would appeal to the enthusiast or professional requiring best-of-class.

A small let-down is the software. Volumio is an open-source playback web app that supports many different DACs. Although it aims to be pretty, we found it unintuitive and buggy, lacking the polish of other similar products like Plex. Luckily, other playback packages are compatible with PecanPi. **M**

NanoSTEM IOT
Weather Kit

▶ Nanomesher ▶ **magpi.cc/rpPidc** ▶ £64 / $80

A much smaller weather kit solution, complete with Nanomesher's signature tiny display. **Rob Zwetsloot** braves the elements to test it

◀ You can get a 3D case to enclose it all, while also allowing the sensors to still work

SPECS

DISPLAY:
1.3" High Contrast OLED (128×96 resolution)

PRESSURE SENSOR:
BMP180 (range 300–1100hPa)

HUMIDITY & TEMPERATURE SENSOR:
SHT31

AIR QUALITY SENSOR:
CCS811 (TVOC and eCO2 sensing)

Verdict

A great little IoT weather station that's easy to use and configure. It can be used around the house, or easily made portable with a mobile battery.

9 /10

Home weather kits are excellent for a number of reasons. With the right calibration, they can be more accurate than a lot of basic solutions. In addition, by nature of being a computer add-on, they allow you to keep track of data over a long period of time – great for science, and also good for stuff like timing your heating in the winter, or even just kitchen conditions when cooking particularly fussy foods.

The NanoSTEM IoT Weather Kit is one of the smallest solutions we've seen for this, and it's relatively cheap as well. Featuring an at-a-glance display for basic data readouts and an online dashboard with the full suite of data, it's basically the full package.

What do you get in the kit, then? Well, it's a HAT-like add-on board that covers the USB and Ethernet ports, along with the rest of a Raspberry Pi. This is the ProdBoard that Nanomesher makes, which includes a tiny 1.3" OLED display, but adding a pressure sensor, humidity and temperature sensor, and an air quality sensor attached to the I²C ports.

Plug and play

There's also a microSD card that is all preconfigured – very welcome, as we know how tricky these sensors can be to set up. Air quality monitors aren't something we see regularly in Western weather kits, but it's a big thing in built-up areas of China. Everything is ready to go out of the box – even the sensors are plugged into the board – so as long as you have access to power, you can get very quickly started with monitoring. 🅜

10 Best:

Gaming accessories

Level up your Raspberry Pi gaming with these accessories and kits

We've done several features and articles over the years on how to set up your own Raspberry Pi-based games console. So we have a good idea what kind of hardware is out there that will improve your experience on any Raspberry Pi gaming setup. Let's take a look… ▨

▼ Picade **X HAT**

Ultimate arcade board

Created for the amazing Picade, the X HAT is the absolute perfect accessory for turning a Raspberry Pi into an arcade machine. Just build the cabinet, add the buttons, and you're good to go.

▶ £16 / $16
▶ **magpi.cc/eaiQpA**
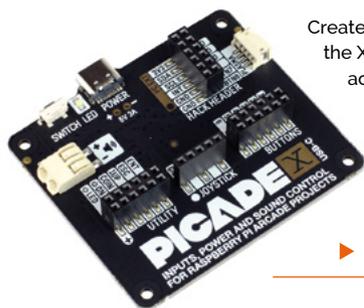
▶ **USB** Gamepad

A retro classic

Including the European/Japanese gamepad colour scheme, this retro-style controller has enough buttons to allow you to play just about any game you have prior to 1995, and some newer homebrew stuff.

▶ £6 / $8
▶ **magpi.cc/RRLFPM**

▲ **BASIC** Arcade Controller Kit

Arcade on your lap

This arcade stick-style kit allows you to build an arcade machine right into the stick. Monster Joysticks also makes a bigger and badder version with high-quality Sanwa arcade parts.

▶ £70 / $87
▶ **magpi.cc/ierNVd**

▼ TinyPi **Pro**

Pocket-sized gaming

This is a bit more of an accessory and more on the kit side – the TinyPi Pro is absolutely tiny, as the name suggests, yet still allows you to play a huge variety of games.

▶ £92 / $115
▶ **magpi.cc/EXKvig**

◄ Massive Arcade Button

For big hits

This huge button is 100 mm across and comes with an LED as well. It's perfect for very specific arcade and gaming builds, especially if you need to smack a big button very fast.

▶ £9 / $10
▶ **magpi.cc/LpMyCp**

▶ Joy Bonnet

Tiny controller HAT

This funky add-on for Raspberry Pi Zero turns it into a controller that is its own console. It's a bit fiddly, but it's also an excellent party piece and can even fit in your pocket.

▶ £14 / $14
▶ **magpi.cc/biqbgi**

▶ Wireless Gamepad

New school cool

Playing newer games thanks to Steam Link or other streaming tech? Then you may need a controller that can do it all, which also doesn't limit you to the length of a USB cable.

▶ £14 / $18
▶ **magpi.cc/fXqWFh**

▶ Cupcade

Mini arcade cabinet

Arcade machines can be huge and expensive to maintain. The Cupcade can fit on a DVD shelf and the parts are much cheaper. Plus, you don't need to hire a van to move it – bonus!

▶ £111 / $141
▶ **magpi.cc/KbHeYL**

▼ RetroFlag **MEGAPi Case**

Super stylish protection

RetroFlag makes a few retro console cases, and all of them are very cool. This design is a classic though, and even lets you use the buttons on the top like the original!

▶ £25 / $32
▶ **magpi.cc/ArHuJc**

▲ Keybow Kit

Extra hot keys

Need more keys for your gaming? Look no further than the Keybow, a Raspberry Pi-powered mechanical keypad. You can program the keys to do anything you want, so it's great for MMOs.

▶ £50 / $52
▶ **magpi.cc/MytktH**

Learn Scratch 3
with Raspberry Pi

Discover the delights of the remarkably easy-to-use programming language. By **Lucy Hattersley**

Raspberry Pi Resources

AUTHOR

Raspberry Pi Foundation

Price:
Free

magpi.cc/qxuVKi

Scratch is a wonderful programming language cleverly disguised as a multimedia interactive game.

With a mixture of characters (also known as 'sprites') and backgrounds, you can build your own interactive scenes. Throw in some music and sound effects and you have a video game, or an animated movie, or an interactive story.

Ostensibly aimed at younger students, Scratch is an excellent educational tool for learners of all levels. Armed with Scratch, you can learn important coding concepts such as loops, variables, and conditional branching.

Scratch 3 is the recently released update. And running Scratch 3 on a Raspberry Pi throws an extra level of learning into the mix, with extensions for GPIO pins and support for Sense HAT.

Where better to start than Raspberry Pi Projects? Choose Help > Projects from the menu in Raspbian, and click on Learn to Code with Scratch; or click on All Projects and change 'Any software' to Scratch. Many of these resources have been updated to work with Scratch 3 (those that have will proudly display so at the start). **M**
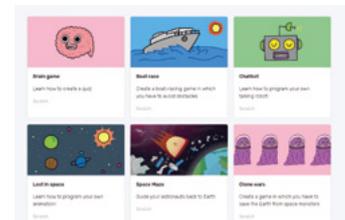
Books and cards

Cards, books, and more

CODING CARDS
MIT has produced a collection of Coding Cards that can be used to learn Scratch in the classroom. Scroll down its Ideas page (also useful) and click the Download Now link. There are 188 cards in total. **scratch.mit.edu/ideas**

CODE CLUB BOOK OF SCRATCH - VOLUME 1
Code Club's book on Scratch is a great primer for the programming language. Big and colourful, it has easy-to-follow projects and comes with

stickers and magic glasses to read hidden hints. It's a great book for kids. **magpi.cc/ccbook1**

25 SCRATCH 3 GAMES FOR KIDS
No Starch Press is quick off the block with book of Scratch 3 games. Games include catching snowflakes and eating tacos. The author, Max Wainewright, is a teacher which makes the book ideal for tutors. **magpi.cc/VMepSh**
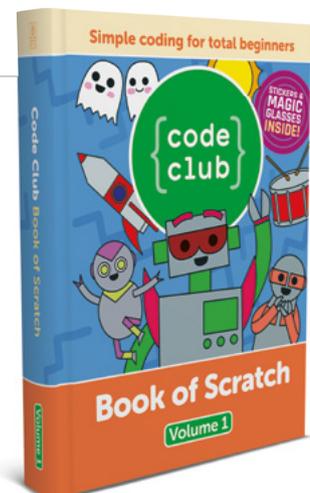
Udemy: Build 11 Games in Scratch 3.0 Bootcamp

AUTHOR

Dr Chris Mall

Price:
£150 / $150

magpi.cc/rxAAwp

Udemy has a selection of great video courses for Scratch. Most are being updated for Scratch 3, but and this gaming course is one of the first out. The course is by Dr Chris Mall and covers all the basics of programming using Scratch 3.

After the introduction to blocks and commands, you'll

view a range of games – The Snake Game, Scary Shark Game, Helicopter Game, Racing Car Game, Crossing the Road Game – all of which builds up to a version of the arcade game Arkanoid.

There are 46 lectures in all, with eight hours of video, along with two articles and downloadable resources for each game.

As with all Udemy courses, it can be expensive but is frequently available at a discount. The RRP is £150, but at press time it was on sale for a much more palatable £12.

Dr Mall also has another course, Scratch 3.0 for Teachers (**magpi.cc/qnSgEA**), with similar content, but aimed at a classroom environment. ◪
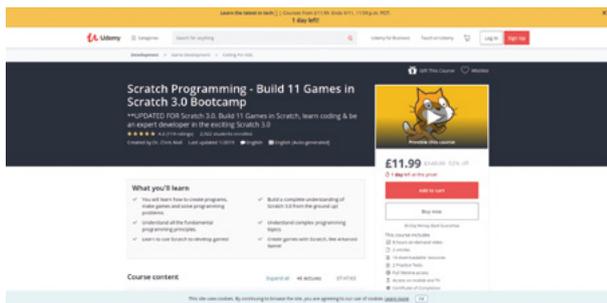
Scratch Tutorials

AUTHOR

The Scratch Team

Price:
Free

magpi.cc/jhzDsu

Why not cut to the chase and learn Scratch straight from the source? Open up Scratch 3 on your Raspberry Pi and click the Tutorials button.

Inside the Scratch 3 program, you'll find a Getting Started guide and plenty of other tutorials for learning key concepts in Scratch. As with the Raspberry Pi Projects, these are a varied bunch of imaginative projects.

You'll find such projects as Animate a Name, Make Music, Create a Story, and Create Animations that Talk. There are also a few games, such as Make a Chase Game and Pong Game. But there's a much
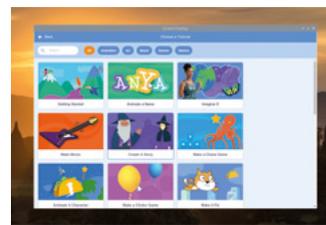
broader focus on animation and storytelling.

Some tutorials, such as Video Sensing, can work with Raspberry Pi. But there isn't the focus on hardware such as the Sense HAT or GPIO pins. For that, you'll need to look to Raspberry Pi itself. However, the built-in tutorials are a good source of fun and these projects are directly integrated with Scratch 3. ◪

Courses & projects

Learn computer science with these Scratch courses and online projects

HARVARD CREATIVE COMPUTING CURRICULUM

The Creative Computing Curriculum is a collection of ideas, strategies, and activities for an introductory creative computing experience using the Scratch programming language.
magpi.cc/yhJPtR

GOOGLE CS FIRST

A free computer science curriculum that makes coding easy to teach and fun to learn.
magpi.cc/mrppRV

CODE CLUB PROJECTS

Many Code Club projects appear on the Raspberry Pi website, but it also has a collection on the Code Club website. There is a great collection of additional projects here.
magpi.cc/kRHmrC

▲ California High School, where Sean teaches, joined in on the Big Birthday weekend

Sean Raser

Computer science teacher, Raspberry Jam organiser, and Code Club leader. Sean Raser sure loves to teach Raspberry Pi

> Day job **Teacher** | > Community role **Raspberry Jam organiser** | > Twitter **@sraser88**

For a lot of the people who do a lot with Raspberry Pi, they've been here since day one – 29 February 2012. Sean, on the other hand, didn't even learn about Raspberry Pi until early 2015, but that hasn't stopped him from teaching about it, putting on events about it, or running a Code Club using the Raspberry Pi Foundation resources.

"I have held Jams in both my local community (Castro Valley) and through my school district (San Ramon Valley Unified School District)," Sean tells us. "I also teach 'Physical Computing with Raspberry Pi' classes through our local Education Foundation, which take place after school, as well as during holiday breaks and summer. I am also the leader of the Castro Valley Code Club, which I started just around one year ago (right after I attended Picademy)."

The Castro Valley Jam is his big community Raspberry Jam, with attendees from all walks of life trying out Raspberry Pi projects. Sean's school-based Raspberry Jam is aimed more towards middle- and high-school students, though.

Do you use Raspberry Pi in your high-school classes?
I absolutely do! I have used them a few different ways. I have used them as their own separate unit,

▼ Sean leads projects for people to try, or they have a go at one from the Raspberry Pi Projects page
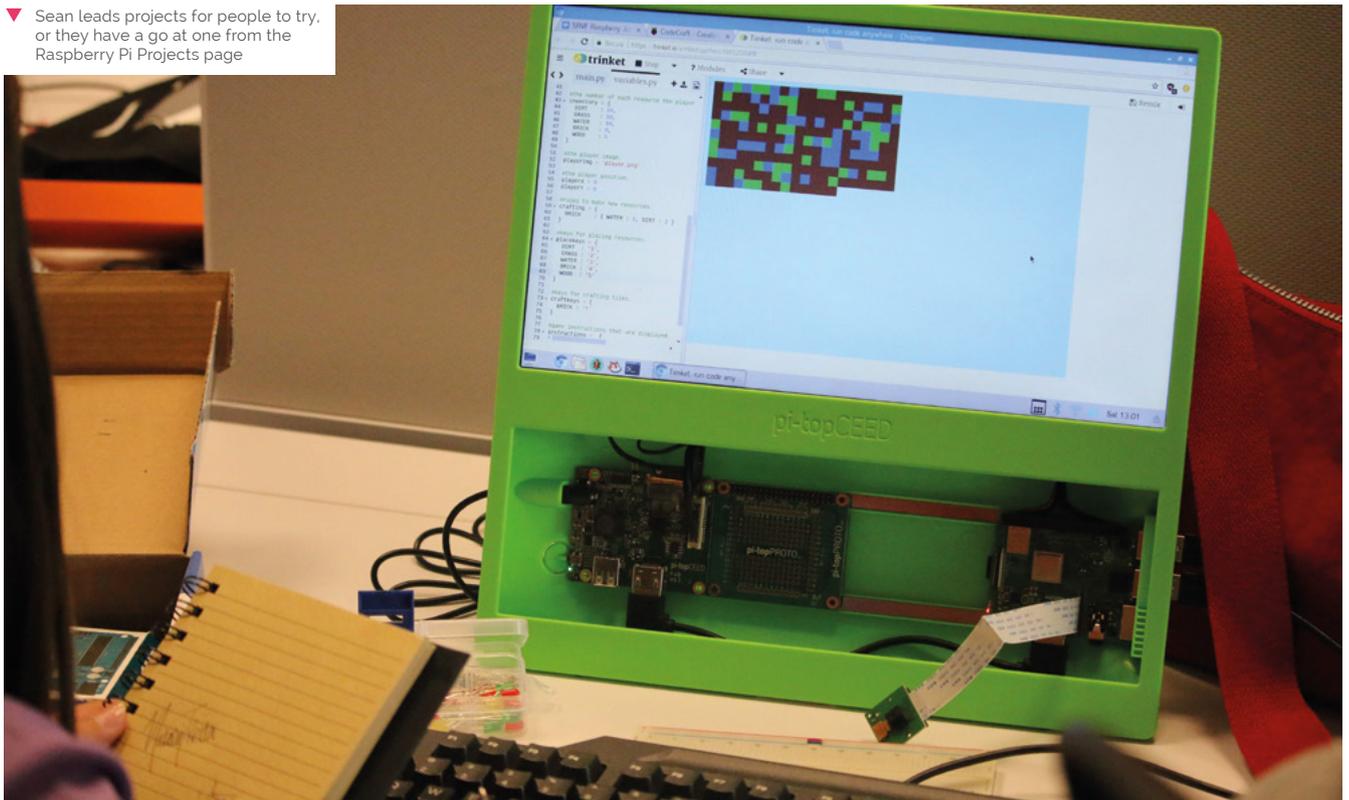
"When it comes to Jams, there is no right or wrong way that they have to be run or put together"

where we spent a chunk of time dedicated to Raspberry Pi mixed in with other content from the class. I first introduced them, and then afterwards I let the students work on their own projects that they either chose or found online. They had a few weeks total to spend using them.

Another way I tried involved working them in throughout the class all year long. For example, I dedicated each Friday to Raspberry Pi. At first, they were taught how to use them, but afterwards they were free to be creative and either work on a year-long project, or many projects over the year.

How was Picademy?

I attended Picademy Denver 2018, and it was, and has been, the greatest workshop I have

ever attended. I learned so many practical things that I could take (and have taken) right back to the class and start using right away. Many of the classes I teach outside of school are based around the Picademy lessons, and when I introduce Raspberry Pi in my high-school classes I use many similar lessons from Picademy.

I like how most of the lessons have the participants just jump right in to the code, and explain it after they have seen it running. This was different from what I did before, but I have had a better experience with it this way. I met so many wonderful educators at Picademy that I still keep in touch with. The network of educators that you are introduced to and stay in touch with is the most amazing,

beneficial part of Picademy. I would encourage anyone on the fence about applying to take the next step and apply!

Any Raspberry Jam advice?

When it comes to Jams, there is no right or wrong way that they have to be run or put together. I have found that they are all different, which is really what makes them such a great experience. Every situation is going to be different, and I have found that just going for it, while not always the best advice for every situation in life, has worked for me. I still get nervous the night before each Jam, but it's a good, excited nervous.

Also, I have always felt absolutely great after every single Jam. They don't always run exactly the way I had planned, and there are always bumps in the road, but at the end of the day I know that the people who attended were able to meet, talk, code, share, and learn. Who could ask for a better day than that?

Fun with Minecraft

"One of my personal favourite uses I have gotten from Raspberry Pi is being able to expose my young son to coding at such a young age. He loves Minecraft, so being able to show him any basic Python programs that modify Minecraft blows him away. The Minecraft camera that can take his picture then build it in his Minecraft world blows him away."

This Month in
Raspberry Pi

Scratch Conference Europe

The Raspberry Pi Foundation hosts a Scratch party

You've probably read all about the big Scratch Conference that the Raspberry Pi Foundation hosted last month on page 6. It was a great time for all involved, and we have the pictures to prove it! **M**

01. There were plenty of stalls to try out numerous Scratch projects

02. Talks were given on a huge variety of topics

03. There was a full schedule over three days with talks, workshops, and keynotes

04. Smaller workshops allowed people to get some hands-on experience, or present more niche subjects

05. People from around the world were here to talk about Scratch

06. Teaching each other was an important part of the weekend

07. Educators swapped tips with each other on how best to utilise Scratch

08. Live-streaming your Scratch Conference experience? Why not?

09. No table or auditorium? Have a chat on the show floor!

01

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. M

01. A secret mission to discover the Great Plumnapper

02. Can it talk as well, so we can ask annoying co-workers to 'get in the bin'?

03. This Raspberry Pi project is inside an Arnie biography and reads it to you

04. This solar-powered lawnmower is getting closer to completion, it seems!

05. Press the asteroid and a rocket ship appears from behind the canvas

06. If the MudPi project piqued your interest earlier in the mag, then this might be for you

redherring05
@redherring05
Replying to @TheMagPi
My night vision detector to know who eats my plums in the garden😫!

Remo TV
@remotvapp
Replying to @TheMagPi
A @Raspberry_Pi powered internet-connected trash can robot, controlled by the internet!
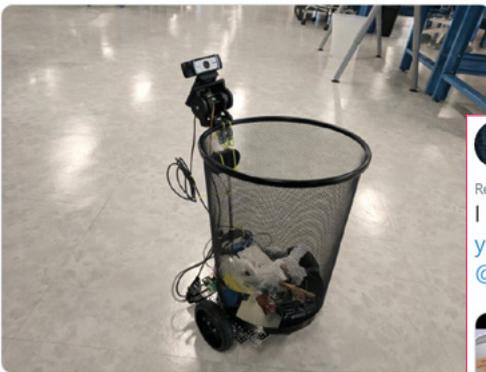
TeCoEd
@Dan_Aldred
Replying to @TheMagPi
I upgraded my Arnold Schwarzenegger speaking book.
youtu.be/WxUQ3rp6xLg
@Raspberry_Pi powered.

Arnold Schwarzenegger Talking Book
Get to the Choppa! An Arnold Schwarzenegger 'Total Recall' book (hacked with a Raspberry Pi) that talks to you and asks ...
youtube.com

TGD
@TGD_Consulting
Replying to @TheMagPi
This is the Raspberry Pi project I've been working on again.

#make your own #3dprinted #IoT #smart #robotics #lawnmower with #PiZero.

8 Bits and a Byte
@8BitsandaByte
Replying to @TheMagPi
An interactive painting you can touch! 🚀

Christopher Biggs
@unixbigot
Replying to @TheMagPi
Designed this garden irrigation hat for a green tech startup.

Crowdfund **this!** Raspberry Pi projects you can crowdfund this month

DACBERRY AMP²

- Class D Amplifier
- 60W Quad Stereo / 100W Dual Mono
- Up to 192kHz and 32bits
- THD+N @1kHz - 0.006 %
- Integrated Custom DSP
- 15V to 24V External Input
- 5V/3A Integrated Regulator for Pi
- Custom Heatsink for a better performance
- Isolated Digital and Analog Circuits
- Quad Low-Noise Power Supplies

Pi-oT – IoT Module

IoT modules for Raspberry Pi are quite common; however, this one is quite the beast. Not only is it designed with normal home automation in mind, but also with industrial automation. It certainly looks the part.

▶ **kck.st/2MELG0B**

DACBerry AMP²

With all the DACs available for Raspberry Pi, why should this be the one for you? Well, the blurb says: "The DACBerry AMP² provides a 4×60 W amplified dual stereo output in Dual Stereo mode and up to 2×100 W in Dual Mono mode. Integrates a 192kHz/32-bit DAC chip with hardware volume mixing and mini DSP."

▶ **kck.st/2Q3AYTO**

CROWDFUNDING A PI PROJECT?
If you've launched an irresistible Pi-related project, let us know!
magpi@raspberrypi.org

Best of the rest!

Here are some other great things we saw this month

RASPBERRY PI-CONTROLLED CRANE

This is a 13,000 lb (~6000 tonne), 45 ft (13.7 m) crane… being controlled with what looks like a third-party Xbox controller. We are worried this may not be the right tool for the job.

▶ **magpi.cc/tfJwaa**

$20 MAGIC MIRROR

Getting a second-hand monitor and some mirrored film for about $20 is a great find, especially when you can turn it in a very handy magic mirror.

▶ **magpi.cc/igkFed**

Raspberry Jam
Event Calendar

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

01. Raspberry Jam Estácio
- 📅 Friday 18 October 2019
- 📍 Estácio, Maceió, Brazil
- ▶ jam.maceio.br

This Brazilian Raspberry Jam is a great place to learn how to make your own projects

02. Milton Keynes Raspberry Jam
- 📅 Saturday 19 October
- 📍 The National Museum of Computing, Bletchley, UK
- ▶ magpi.cc/dgijby

Where better than the National Museum of Computing at Bletchley Park to show off your Raspberry Pi project?

03. Raspberry Pi Jamwich
- 📅 Friday 25 October
- 📍 The Forum, Norwich, UK
- ▶ magpi.cc/pYaUYd

Part of the Norwich Science Festival – you'll be able to try out workshops on lights, Minecraft, or whatever you want!

04. Patriot Pi Raspberry Jam
- 📅 Saturday 26 October
- 📍 Cherry Valley-Springfield Central School, Cherry Valley, NY, USA
- ▶ magpi.cc/bEDMUy

An educational experience for people new to Raspberry Pi or veterans, with hands-on activities.

05. Castro Valley Jam
- 📅 Saturday 5 October
- 📍 Castro Valley Library, Castro Valley, CA, USA
- ▶ magpi.cc/eXpRiF

If you're interested in computer coding and want to learn about Raspberry Pi, check out this Jam.

06. Exeter Raspberry Jam
- 📅 Saturday 5 October
- 📍 Exeter Library, Exeter, UK
- ▶ magpi.cc/LpKVCF

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

07. Stafford Raspberry Jam
- 📅 Tuesday 8 October
- 📍 Stafford Library, Stafford, UK
- ▶ magpi.cc/fNLiwT

A meet-up for folks who have a Raspberry Pi computer and want to learn more about it.

08. Dallas Young Makers Club
- 📅 Saturday 12 October
- 📍 Dallas Public Library, Dallas, TX, USA
- ▶ magpi.cc/hbGzPD

Free mentor-led, hands-on projects for kids with Raspberry Pi, Lego Mindstorm robots, and more.

FULL CALENDAR

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

rpf.io/jam

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Tom Hadfield about it:

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?

07

03

02

06
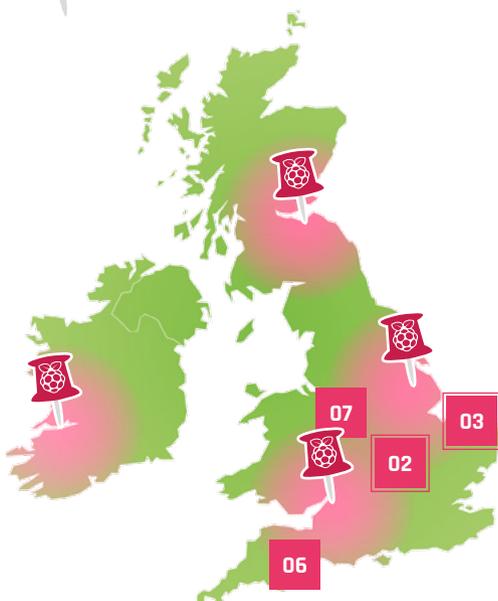
Raspberry Jam advice:

Ticketing

"**W**e make sure free tickets are always available for those who need them. We issue an 'optional donation' ticket with a suggested donation of £3 per person, or £5 if attending a workshop, but we make it very clear that it is entirely optional. We then also go round and shake a tin about half an hour before the end."

Andrew Oakley – Cotswold 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**

RASPBERRY JAM GUIDEBOOK

STARTING AND RUNNING A RASPBERRY JAM
ADVICE GATHERED FROM THE RASPBERRY PI COMMUNITY

Your Letters
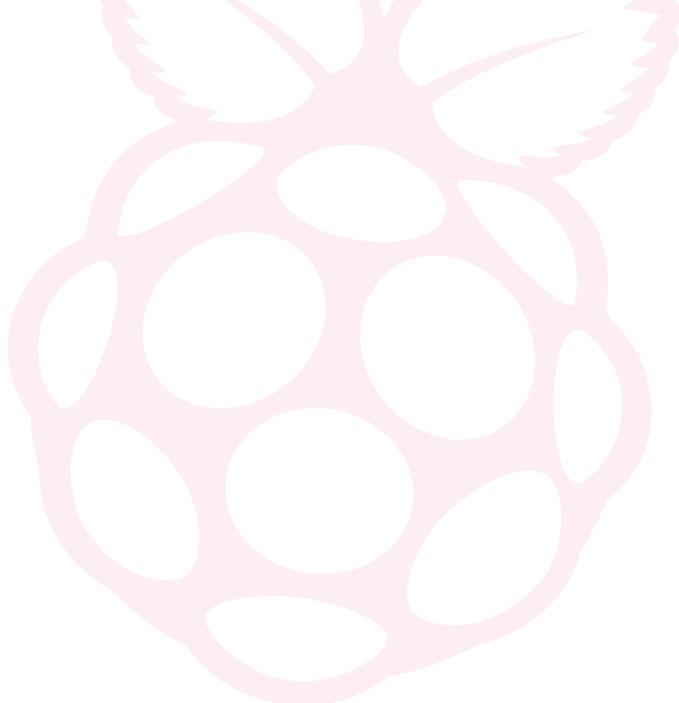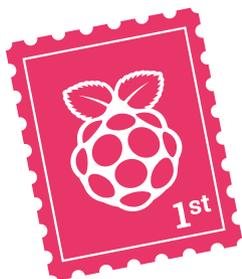
Adults in space!

I always love hearing about new Astro Pi projects; however, I think I'm a bit too old to participate in it! Is there anyway for us grown-ups to get involved? Would love to say I'd helped get something in space!

Kelly via Facebook

While you can't apply with a team of grown-ups yourself, you can help mentor a group of young makers! As a team mentor, you'll be the contact between your team and the Astro Pi team, Raspberry Pi Foundation, and even the ESA.

If you want to learn more about how you can help young makers with their Astro Pi Challenge, head here for more details: **magpi.cc/YeisCW**.

Contact us!

> Twitter **@TheMagPi**
> Facebook **magpi.cc/facebook**
> Email **magpi@raspberrypi.org**
> Online **raspberrypi.org/forums**

▲ This year we were judges at the 2019 Coolest Projects USA event – hopefully we can see your projects next year as well!

More Coolest Projects

It's been a while since the last Coolest Projects event – I was wondering if there were any plans for more Coolest Projects in 2020? I'd love to attend the International one – it looks like so much fun!

Ken via email

You're in luck – the full schedule for Coolest Projects has just been announced for 2020! Coolest Projects USA, International, and UK will all be making a return, roughly around the same time as last year:

- *USA, Discovery Cube Orange County, CA*: 7 March 2020

- *UK, The Sharp Project, Manchester*: 4 April 2020

- *International, RDS Main Hall, Dublin, Ireland*: 6 June 2020

We'll try to get along to one or two of them, so hopefully see you there!

Stretching time

In the article 'Control servos with CircuitPython and Raspberry Pi' in issue 84 (**magpi.cc/84**), Melissa LeBlanc-Williams used a Bosch BNO055 device on the I²C bus, which usually requires 'clock stretching'.

In step 7, clock stretching is said to be enabled, which so far was not supported by Raspberry Pi devices. The method used looks rather like slowing down the I²C clock permanently, than enabling clock stretching. Although that may be a suitable workaround for the problem, I would prefer to call things by their right name, than raise expectations that cannot be fulfilled. Please correct me if I²C clock stretching is supported by Raspberry Pi 4 and enabled that way.

Heimo via email

We admit we had to ask the Raspberry Pi engineering team about this one as we had not come across it before – they told us that you are indeed correct: I²C clock stretching was not supported on older Raspberry Pi models. This was a bug which was finally corrected with the release of Raspberry Pi 4, and the method used in the tutorial allowed you to make use of it.

More Picademy?

I'm an educator looking to learn more about Raspberry Pi and have been very interested in Picademy. Unfortunately, it always seems to be on at a time and place I'm unable to go to. Are there any plans to expand the amount of Picademy events in the country at all?

Terri via email

The Raspberry Pi Foundation has just announced something it's calling Picademy Bytes – a shorter, 60- to 90-minute session run by community volunteers who have been trained by the Foundation.

They'll be popping up around the UK, with four Community Trainers raring to go and get people started with RaspberryPi. You can find out more information on the blog here: **magpi.cc/dCVWRU**. Dates and venues will soon be announced! 🅼
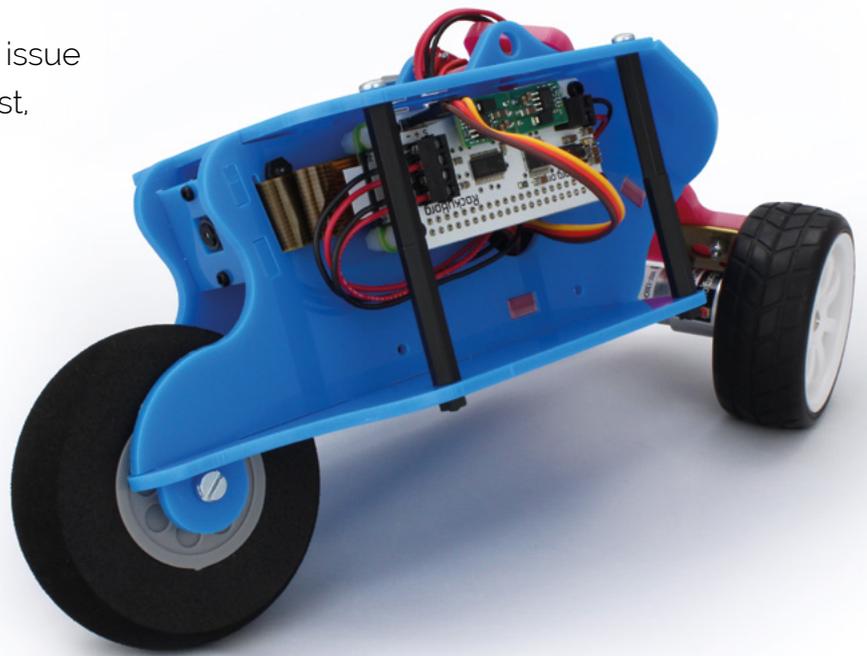
WIN
ONE OF FIVE
ROCKYBORG
ROBOTS!

We reviewed the RockyBorg last issue (**magpi.cc/85**) and loved it. It's fast, easy to program, and fun to use.

" A nimble robot with two 20 mm 180 rpm metal geared motors, a powerful metal gear servo for leaning into corners, and an all-new motor controller "

— PiBorg

In association with **PiBorg**

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

Terms & Conditions

Competition opens on **25 September** and closes on **31 October 2019**. 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.

BUILD A RASPBERRY PI

4K MEDIA CENTRE

THE MAGPI **#87**
ON SALE **31 OCTOBER**

Plus!

Make a Raspberry Pi digital camera

Retro gaming level design in PICO-8

Assemble a cluster computer

Control your low-cost wheeled robot

DON'T MISS OUT!

magpi.cc/subscribe

TWITTER	**@TheMagPi**
FACEBOOK	**fb.com/MagPiMagazine**
EMAIL	**magpi@raspberrypi.org**

The secret to success

Volunteering - why we do it and why you should too.

By **Michael Horne** and **Tim Richardson**

Our volunteer experience with Raspberry Pi started way back in May 2013. First, Mike resurrected the Cambridge Raspberry Jam. Two Jams later and Tim joined in to run the event.

Together, we turned it into one of the biggest Raspberry Jam meet-ups in the world. That is only part of the story, though, because it sounds like we did it all ourselves, which just is not true.

In order to make the Jam bigger, we needed a larger venue, and to be honest, we did not have a clue as to

> ❝ We had so many volunteers who contributed so much to the Jam ❞

how to get one. Fortunately, Andy Batey came forward and offered his workplace as a venue.

That venue, the Institute of Astronomy, in West Cambridge (**ast.cam.ac.uk**), was our home for the next four-and-a-half years, and it was only with the help of other people that we managed to keep it going. We had so many volunteers who contributed so much to the Jam. We had people who ran workshops, who manned the front desk, who helped us with setup

and clear-down, who gave talks, and who did show and tell. All of these volunteers gave up their own time so that we could support the aim of the Jam – which was primarily to help kids learn to both code and get involved in physical computing, as well as support the vast array of hobbyists that were using Raspberry Pi.

Rise of the robots

When we then started Pi Wars, the challenge-based robotics competition, we knew that it would only work with the help of others. We could sort out the courses and the logistics in the run-up to the event, but on the day we would need virtually an army of volunteers, to marshal and judge, and help us to tidy up at the end of the event. We knew that getting those volunteers would be a central aim of the organisation of the event and the 'other side of the coin' when it came to the event itself.

We are incredibly grateful to anyone who has volunteered at one of our events. We simply could not do it

without you. You are the backbone of every event that is run, the (sometimes hidden) element of any successful event.

Getting involved

If there is a message behind this article, it is to encourage anyone reading to support your local events; to get involved. If you are a hobbyist, consider spreading your knowledge and experience around. If you are bringing your child to an event, and you are on your own, of course you will want to keep an eye on them. If both parents are attending, though, one or both of you might consider giving up some of your time to support the event. Even if you do not have the technical skills to support in a workshop, you could volunteer to be on the front desk at a Jam, or be a judge at Pi Wars. Our point is: everyone has something to contribute, and it is one of the most rewarding things you can do. **M**

AUTHOR

Michael Horne & Tim Richardson

Between them, Mike & Tim have run the Cambridge and Potton Jams, Pi Wars, and the first two Raspberry Pi Big Birthday Weekends.

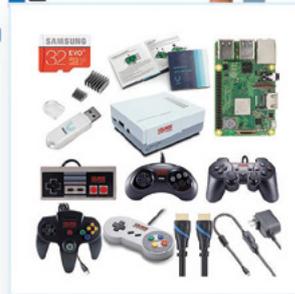Mike **@recantha** / Tim **@geeky_tim**

BUILD, PLAY, WIN!

VILROS

RETRO GAMING SALE

Vilros Raspberry Pi 3 Model B Retro Pi Arcade Gaming Kit with Classic USB Gamepad
~~$79.99~~ **$75.99**

Vilros Raspberry Pi 3 Model B+ Retro Arcade Gaming Kit With Retro Gaming Controller Set
~~$114.99~~ **$109.99**

Vilros Raspberry Pi 3 Model B Plus Retro Gaming Kit with 2 Gamepads & Fan-Cooled Retro Gaming Case
~~$89.99~~ **$85.49**

Vilros Retro Gaming USB Classic Controller Set of 5
~~$39.99~~ **$37.99**

USE THE CODE **RETRO** FOR 5% OFF   Check us out at **www.vilros.com** /vilrosusa

Crack Coupon Code for ADDITIONAL 20% Savings!
"*Restrictions Apply"   "**Offer Expires May 31, 2019"