The MagPi
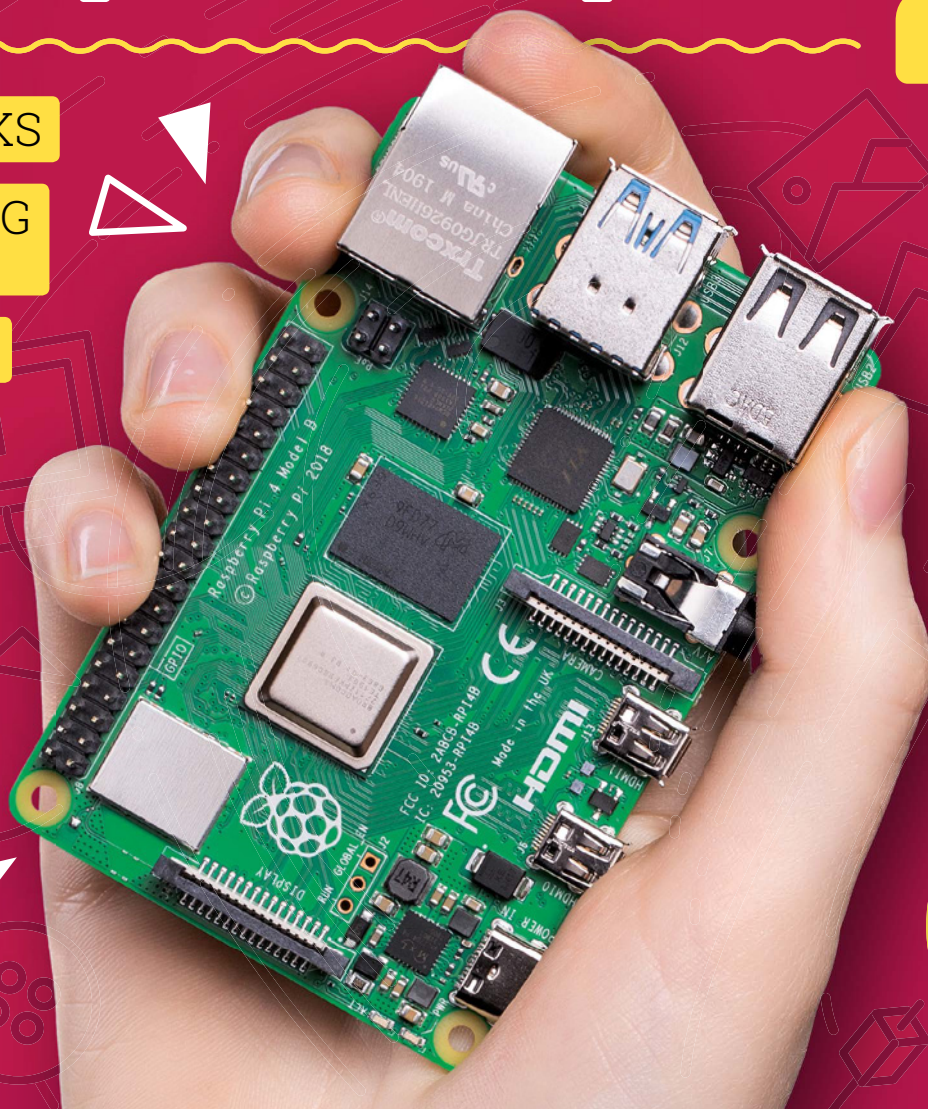
Introducing

Raspberry Pi 4

BENCHMARKS

ENGINEERING INTERVIEWS

TECH SPECS

+ Your **Ultimate Starter Guide**

£5.99

40 PAGES OF PROJECTS & GUIDES

WELCOME
to *The MagPi* 83

So here it is. The big secret is out and we can finally talk about it: Raspberry Pi 4! The new model is a serious step forward. Raspberry Pi has always been a capable computer, edging towards full desktop capability. That edge has arrived.

With up to 4GB of RAM, dual HDMI, quad-core ARM CPU, Gigabit Ethernet, and USB 3.0, Raspberry Pi 4 is a truly a desktop computer. I mean: Raspberry Pi has always been a computer, but now it's a *computer*.

Raspberry Pi 4 is a Linux-powered PC without compromise. Browsing the web is a breeze, apps start and videos play without stutter, and you can multitask several programs at once.

To celebrate its launch, we've got a full feature with benchmark tests and interviews with the engineering team (page 26).

For all its power, Raspberry Pi 4 remains a programmable computer. That bit is important. A tablet is a tablet, a phone is a phone, but a Raspberry Pi can become anything you want, used for any task.

This month, Raspberry Pi is a pop star's TV wall (page 12), an earthquake-detecting robot (page 16), a smart greenhouse (page 22), a programmable console (page 44), a squeezable controller (page 54), and a spacecraft (page 66). Try doing that with your smartphone.

Raspberry Pi is, and always will be, a programmable computer. Only now, it's a really fast one.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy spent this month juggling a Raspberry Pi launch and screwing bits of three-wheeled robots together. She also lost a game of Go quite badly to Google's AlphaGo AI engine.

magpi.cc

Contents

> Issue 83 > July 2019

Cover Feature

Regulars

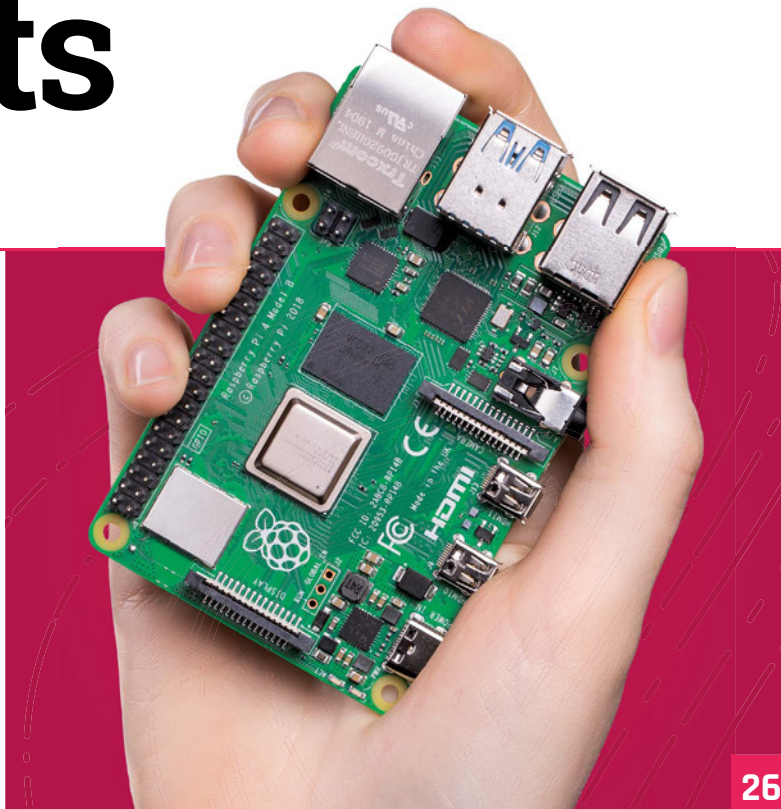## Project Showcases

26

12

22

Smart palm greenhouse

TV Wall

Get started with PICO-8

Squeeze controller

Enviro+

Chris Barnatt interview

WIN ONE OF TEN **RASPBERRY** PI 4 KITS **95**

Raspberry Pi smart
cars skip traffic jams

Cambridge University uses Raspberry Pi to build autonomous cars with superior self-management skills. Bring on a more efficient, crash-free future, exhorts **Rosie Hattersley**

Trials of autonomous cars on a track in Cambridge have proved that traffic congestion headaches could be largely avoided in the future.

The results of tests using a small fleet of Raspberry Pi-controlled cars by Cambridge University researchers were presented at the International Conference on Robotics and Automation in Montreal in June.

Mimicking motorway mayhem

Scale models of existing cars were fitted with motion-capture sensors and Raspberry Pi controllers, and communicated with each other over wireless LAN. Researchers were keen to establish the behaviour of self-driving cars when faced with obstacles, such as a broken-down vehicle.

Where the robot cars were operated independently from each other, a car stuck behind one that had broken down had to slow down or stop and wait for a gap in the traffic in order to pull out into the next lane and overtake the obstacle. So far, so similar to commuters' everyday experience.

However, in scenarios in which it was possible for the robot cars to communicate, the cars were able to drive 'co-operatively'. If a car in the inner lane slowed down or stopped, a signal was sent to all the other cars. Those in the outer lane slowed slightly so cars behind the slower one were able to pull out and avoid a collision, thus keeping both lanes of traffic flowing smoothly. With improved road safety it was possible to pack cars more closely, improving efficiency by

Cooperative driving increases throughput by at least 35%

◀ The fleet of miniature cars was built using Raspberry Pi
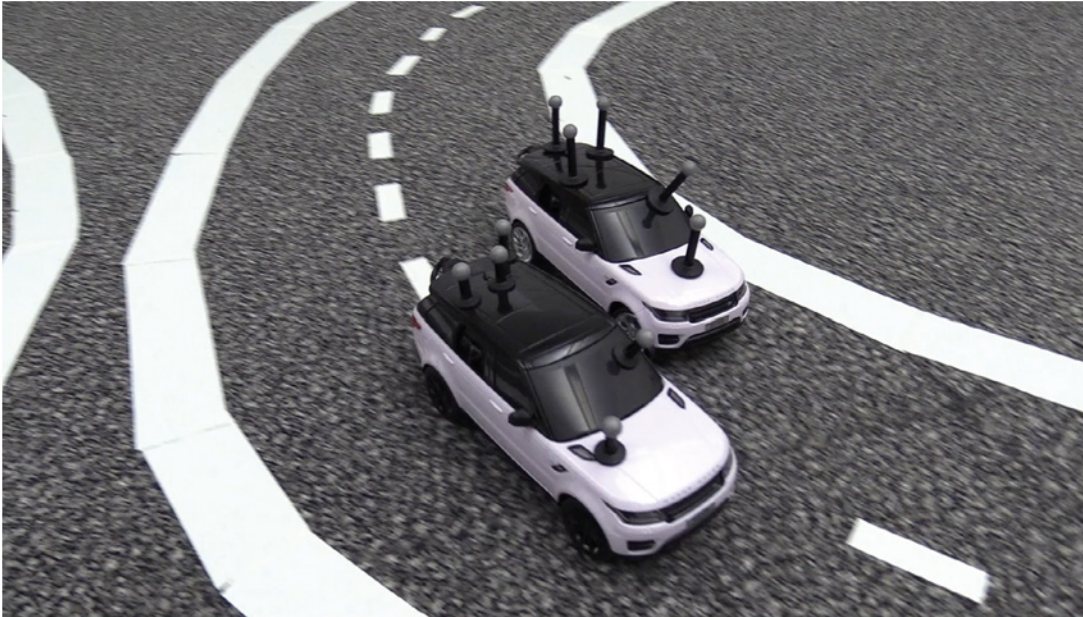
◄ The cars communicated with one another to drive safely and more efficiently

> **"** With improved road safety, it was possible to pack cars more closely **"**

between 35% and 45%, depending on whether the tests were modelling co-operative or 'aggressive' driving modes.

Cross-car communication

"Autonomous cars could fix a lot of different problems associated with driving in cities, but there needs to be a way for them to work together," says report co-author Michael He, an undergraduate student at St John's College who designed lane-changing algorithms for the experiment. The algorithm calculates whether it is safe to change lanes, and whether there's a compelling advantage to doing so – a perhaps more rational approach than some human drivers.

The researchers now plan to trial larger fleets of cars with more complex scenarios such as road junctions and a wider range of vehicle types.

More information about the experiment can be found on the University of Cambridge website: **magpi.cc/QGnWCY**. **M**

Motor power: 2.2 hours
Logic power: >5 hours
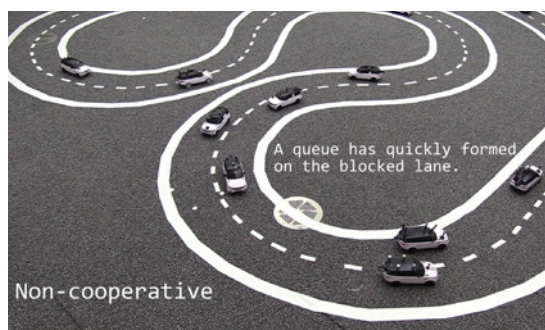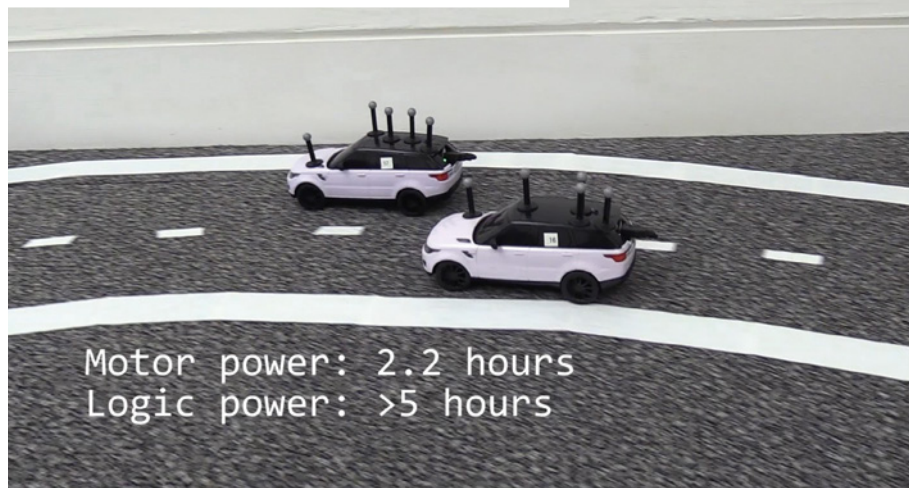
▲ Each car contained a Raspberry Pi Zero W to control the micro-servo arm and DC motors

◄ When the cars didn't communicate and co-operate, a broken-down vehicle soon caused a queue

A queue has quickly formed on the blocked lane.

Non-cooperative

Blender inspires
Code Club kids

Young imaginations are fired with a powerful 3D animation tool running on Raspberry Pi, discovers **Rosie Hattersley**

Film fans have come to expect blockbusters to deliver incredible wow moments. Commercial success often depends on highly anticipated, technically challenging animation scenes.

But big-screen visuals aren't always the result of incredibly expensive animation tools. Blender is a free, open-source, visualisation tool developed 20 years ago by Dutch animation company NeoGeo and has evolved through community input across multiple computer platforms.

Blender is available for the Raspberry Pi, and pupils can also make great use of it to learn coding skills. Children at Elton Primary School in Cheshire recently embraced the opportunity to get hands-on animation experience when Craig Fisk, one of the volunteers at their Code Club, showed them how it works (**magpi.cc/DdqXVz**).

> " Blender was used in superhero films such as Wonder Woman "
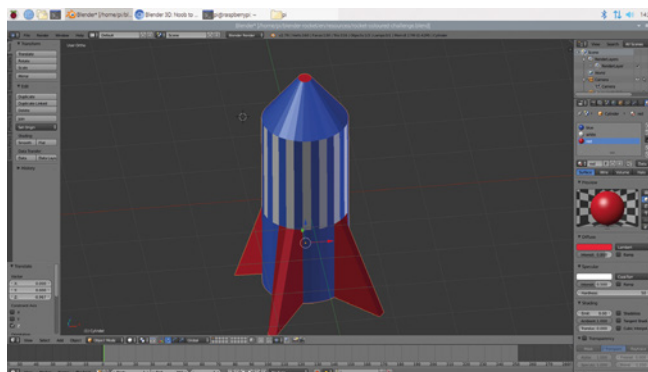
▲ Students learn to use Blender by colouring in a 3D rocket
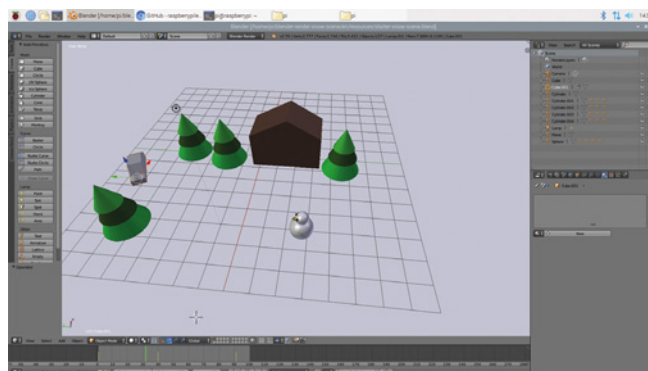
▲ This snow scene can be animated to learn 3D animation skills

They were especially excited when they realised Blender is the building block for the 3D animation used in many games and films. Blender was used in superhero films such as *Wonder Woman* and sequences in TV show *The Walking Dead*.

Blender is also used in the visuals that complement the virtual reality of Google Expeditions. Pupils use a smartphone and a cardboard viewer to create incredible virtual journeys from the comfort of their classrooms.

Model students

Projects hosted on the Code Club website (**magpi.cc/iovwYo**) allow Raspberry Pi users to create models and animations, such as making a car drive through a snow scene.

Craig says the pupils were so keen to use Blender that, once they'd mastered the basic controls to manipulate objects, they zipped through two projects in a single Code Club session. They then went on to master skills such as extruding and editing objects to create a 3D visualisation of a house, as well as creating an animated rocket.

Now that the group has worked through all the Blender projects showcased on the Code

Club website, they are beginning to work on original animations of their own. Craig says, "My plan is to get them to work collaboratively to create a new scene which they will animate. Each student will build something to put into the scene, which we will then combine, and then they can all have a go at animating it." **M**

▲ A student works on rendering a rocket model in Blender
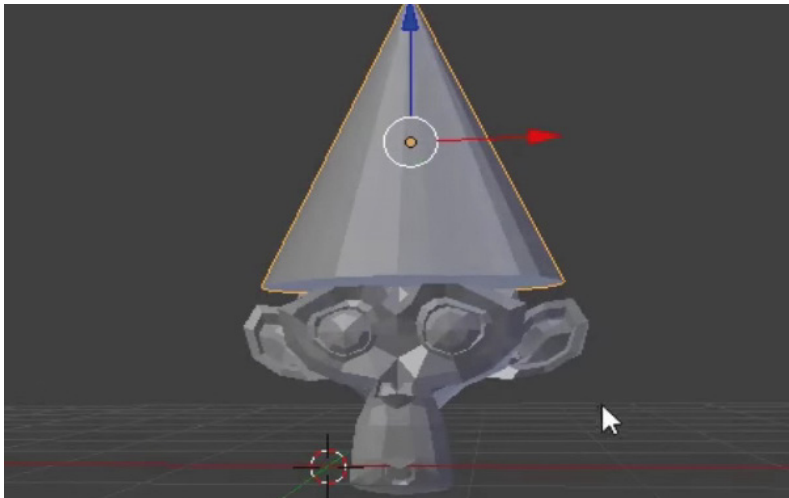
▲ One of the Code Club projects tasks students with creating a monkey with a party hat
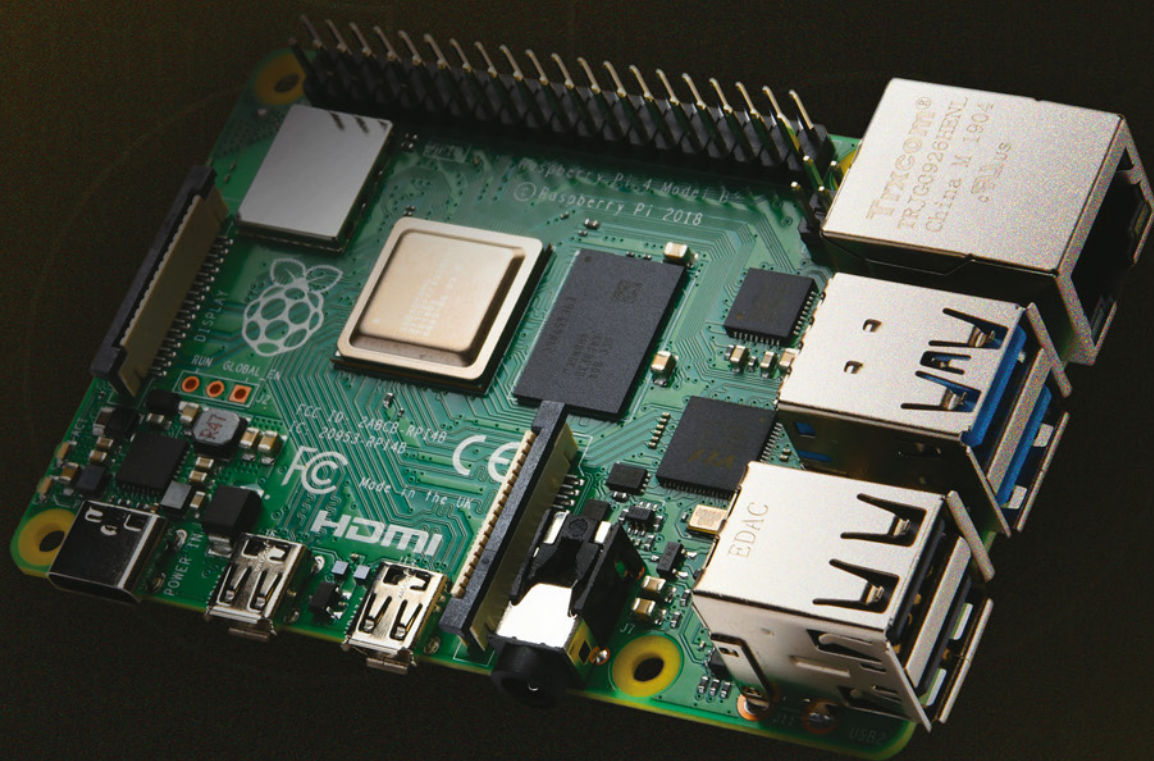
OKDO Pi4 SLIDE CASE
Protective case with a smooth
slide action. Available in Black,
White or Clear
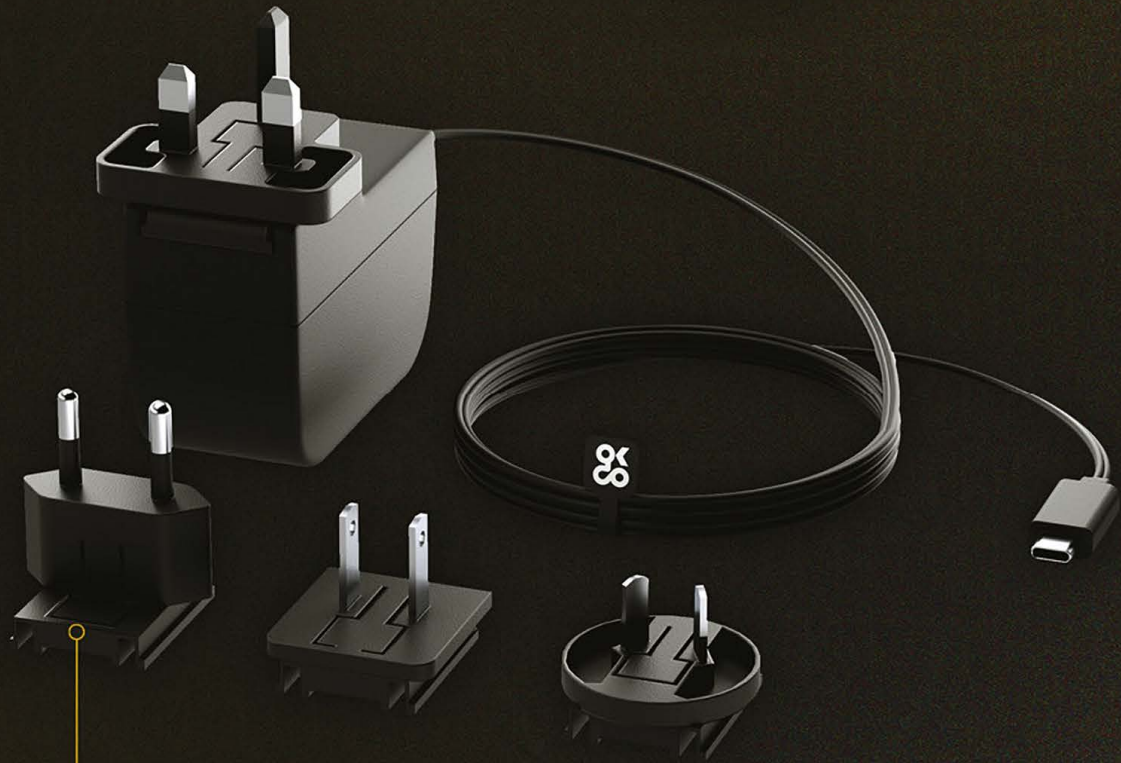
OKDO Pi4 POWER SUPPLY
Start straight from the box with
a power supply that easily adapts
to all international sockets.

OKDO™
DESIGN THE WORLD

TV Wall

Fans of analogue TVs and upcycling old tech will be wowed by this visually arresting Raspberry Pi-controlled display, reckons **Rosie Hattersley**

MAKER

Johanna Tano

Self-taught programmer Johanna's visual installations have appeared at music festivals, in a forest in Sweden, and at fashion shows. Each is controlled by Raspberry Pi.

johannatano.com

Where would the Wizard of Oz have been without the visual artifice that kept his myth alive? Being confronted with Johanna Tano's wall of TV screens surely has a similar disembodied effect to the sight that greeted Dorothy and pals when they entered the Emerald City. Harnessing the power of Raspberry Pi computers, Johanna has been able to sync up and independently control up to 30 analogue TV displays at once.

In her aptly named TV Wall, Johanna demonstrates the possibilities of using old tech and new in a highly engaging fashion. The TV Wall had its debut at Stockholm's Fashion Week in 2017, where the likes of singer Kelis partied to a backdrop of multi-screen live video mash-ups.

The sun always shines on TV

In its most recent iteration, the TV Wall is central to Swedish singer Zacharias Zachrisson's music video for *Shadow*. He says: "Together with the video director, Albin Eidhagen, we created custom-made videos and animations for the TV screens, which was live-mixed when we filmed it." The result is a decidedly eighties video, not unlike something by A-Ha.

Appropriately enough, the video came about when the singer saw the TV Wall displaying live code-generated visuals on Instagram and got in touch with Johanna. She had only a week to assemble the build for the music video shoot and says things "get complicated, fast" as the installation is scaled up and more TVs – and more IP addresses – are added.

Those inspired by her project can follow in her footsteps by breaking it down. "Figure out how to get a video signal from Raspberry Pi to a TV, then how to stream video from a computer to a Raspberry Pi," she advises. "Work out how to tell multiple Raspberry Pi boards to display different parts of the same video. By solving each step, you end up having a quite advanced system without even realising it."

❚❚ Work out how to tell multiple Raspberry Pi boards to display different parts of the same video ❚❚

▶ Johanna sourced the analogue TVs from a range of online sources, paying about 100 euros for each

Raspberry Pi boards control the displays of up to 30 networked analogue TVs

The custom-built web interface controls which bit of each video appears on-screen and when

The TV setup can be reconfigured since output is dependent on each on-board Raspberry Pi's instructions

Quick **FACTS**

> Johanna once created a fortune-teller from a rotary-dial telephone

> She harnessed real space data to generate visual art, controlled by an EEG

> She was part of the Berlin-based tech-art collective Lacuna Lab

> Her first Raspberry Pi project lit up Sweden with 45 LED light trees users controlled remotely

> She views Raspberry Pi as the ideal way to control hardware used in conceptual art projects

▶ Some impressive visual effects can be achieved by sending a different video feed to each TV
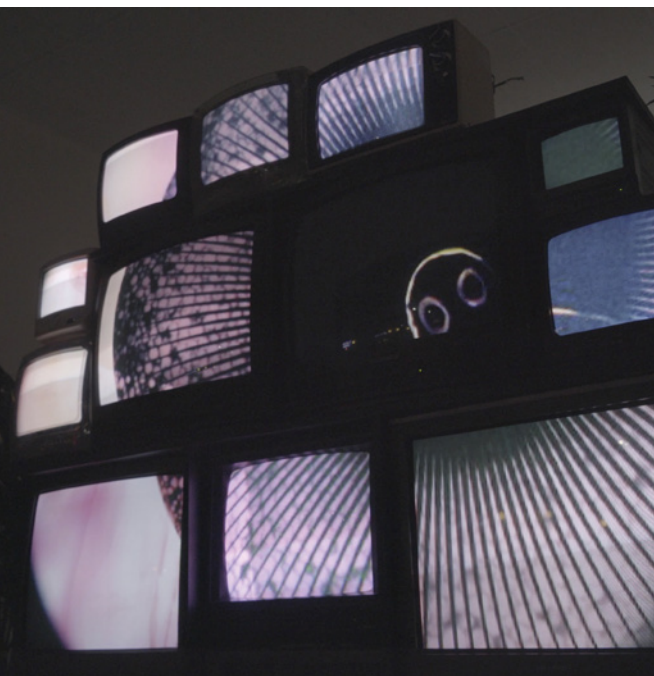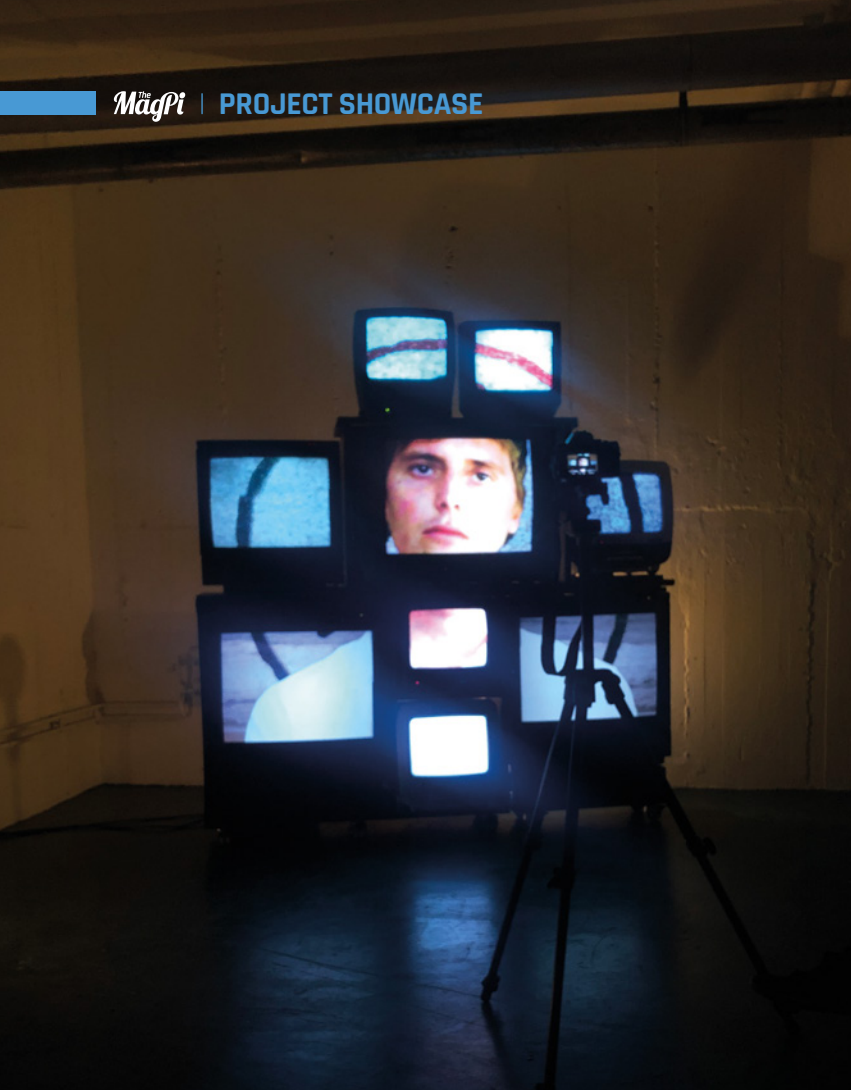
◄ Automatic mapping is controlled through the web interface, meaning she can easily modify each Raspberry Pi's settings, including which part of each video to use

▼ A TCP Syphon Server application connects to the TV Wall network, enabling the video or visual output generated by the VJ software to be displayed in real-time on the wall

Raspberry Pi on TV

01 Cables, adapters, network switches, and routers, plus the customisable TV stands, cost Johanna around 500 euros.

02 Johanna tried to make the monster stand a modular shelf that can be easily moved. It's even got tiny wheels!

03 Each TV has its own IP address, connects to the Node.js server on Johanna's laptop, and looks out for incoming events such as video feeds.

Stockholm-based Johanna is a self-taught programmer and maker. She was a web developer and digital producer at companies such as Acne and B-Reel. Four years ago she set up on her own, specialising in conceptual art and with a specific aim of working collaboratively across different media and learning new things. It led to her ongoing interest in Raspberry Pi.

"I started using Raspberry Pi [because] I wanted to find new areas where I could use my programming skills and wished to go outside the traditional computer screen with my output, bringing in more physical and real-world objects in my work," explains Johanna. "Raspberry Pi has been the ideal solution to use in most of my interactive art installations to control the hardware."

Superstar VJ

Those with designs on becoming video DJs themselves will need a Mac and software that supports video output for Syphon (**magpi.cc/FVJQXh**), an open-source client that works with lots of video streaming programs.

Johanna's technical expertise is also a critical element. Behind the scenes of her TV Wall, she's busy controlling which bits of video footage appear on which TV and when, via the web interface she built. She can modify each Raspberry Pi's settings, including which part of each video to crop out. With such audio-visual control at her command, it's no wonder she reminds us of a modern-day kind of Oz. *M*

PARSLEE

NASA scientist Dr Jamie Molaro tells **David Crookes** of her plans to conduct potentially ground-breaking research using a Raspberry Pi seismometer and a mini rover

MAKER

Dr Jamie Molaro

Jamie is a research scientist at the Planetary Science Institute and a contractor at the Caltech / Jet Propulsion Laboratory. She is a science team member of NASA's OSIRIS-REx mission to return samples from an asteroid to Earth.

magpi.cc/oQmeFR

I n the summer of 2018, engineers at NASA's Jet Propulsion Laboratory built a mini planetary rover with the aim of letting students, hobbyists, and enthusiasts create one for themselves. It uses commercial off-the-shelf parts and has a Raspberry Pi as its brain. But despite costing about £2000 in total, the Open Source Rover Project has proven rather popular, including among people who actually work for the USA's space agency.

One of those is Dr Jamie Molaro, a research scientist at the Planetary Science Institute. Her main focus is studying the surfaces of rocky and icy airless bodies such as comets, asteroids, and the moons orbiting Earth, Jupiter, and Saturn. So when she decided to create her mini-rover – which she dubbed PARSLEE, or Planetary Analog Remote Sensor and 'Lil Electronic Explorer – she also sought to shake things up a little.

Brought to life

Constructing the robot itself was, she says, rather straightforward: the instructions were detailed and she was able to draw upon the help of others in a forum. Jamie also built the robot with her husband, a software engineer at Adobe. "My interest in the Open Source Rover Project was driven by my scientific background, but not my ability to build it," she tells us, of what is essentially a miniature version of the Curiosity rover trundling over the surface of Mars.

Jamie's interest in science led to her considering the rover's potential payload before the couple had even finished building it. She added a GoPro camera and a Kestrel 500, which measures temperature, pressure, elevation, wind speed, and humidity. In addition, she opted to use a Raspberry Shake seismometer – a device costing a few hundred dollars which comprises a device sensor, circuit board, and digitiser – with a Raspberry Pi board and a preprogrammed microSD card.

The sensor records activity, converts the analogue signals to digital, and allows the recorded data to be read on Raspberry Shake servers. Jamie hopes to use PARSLEE to study the kinds of processes active at the surface of other planets. "A seismometer helps us understand our physical environment in a very different way than images from a camera," she says.

Seismic solutions

To that end, with funding, Jamie would like to heat and cool boulders and soils in the lab and in the field and analyse their seismic signature. "Thermally driven shallow moonquakes were recorded by instruments used by the Apollo astronauts," she says. "We believe these quakes may reflect signals from a thermal fracturing process that breaks down lunar boulders, or from the boulders and
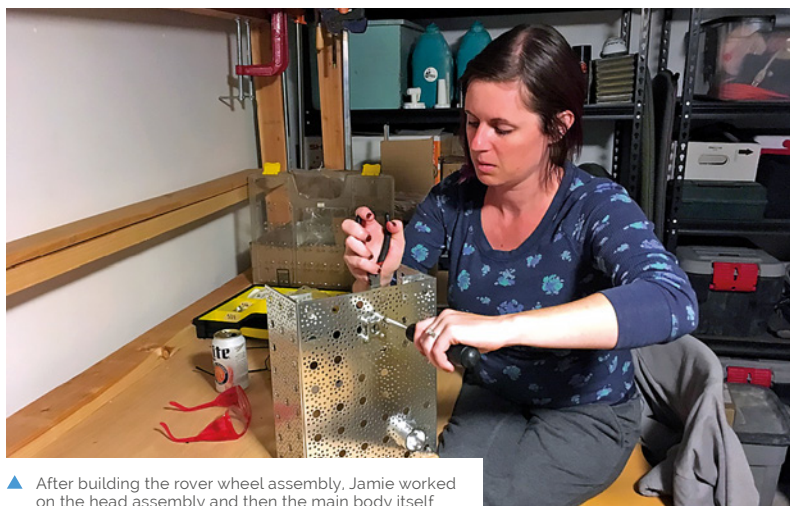
▲ After building the rover wheel assembly, Jamie worked on the head assembly and then the main body itself
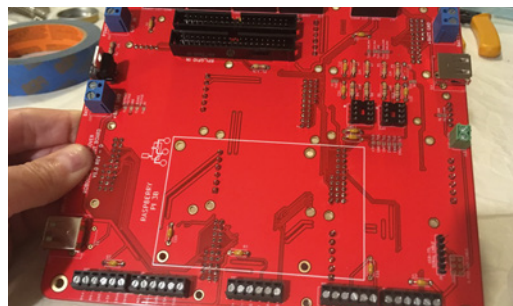
▲ With the electronics assembly complete, Jamie and her husband could get on with integrating PARSLEE's parts
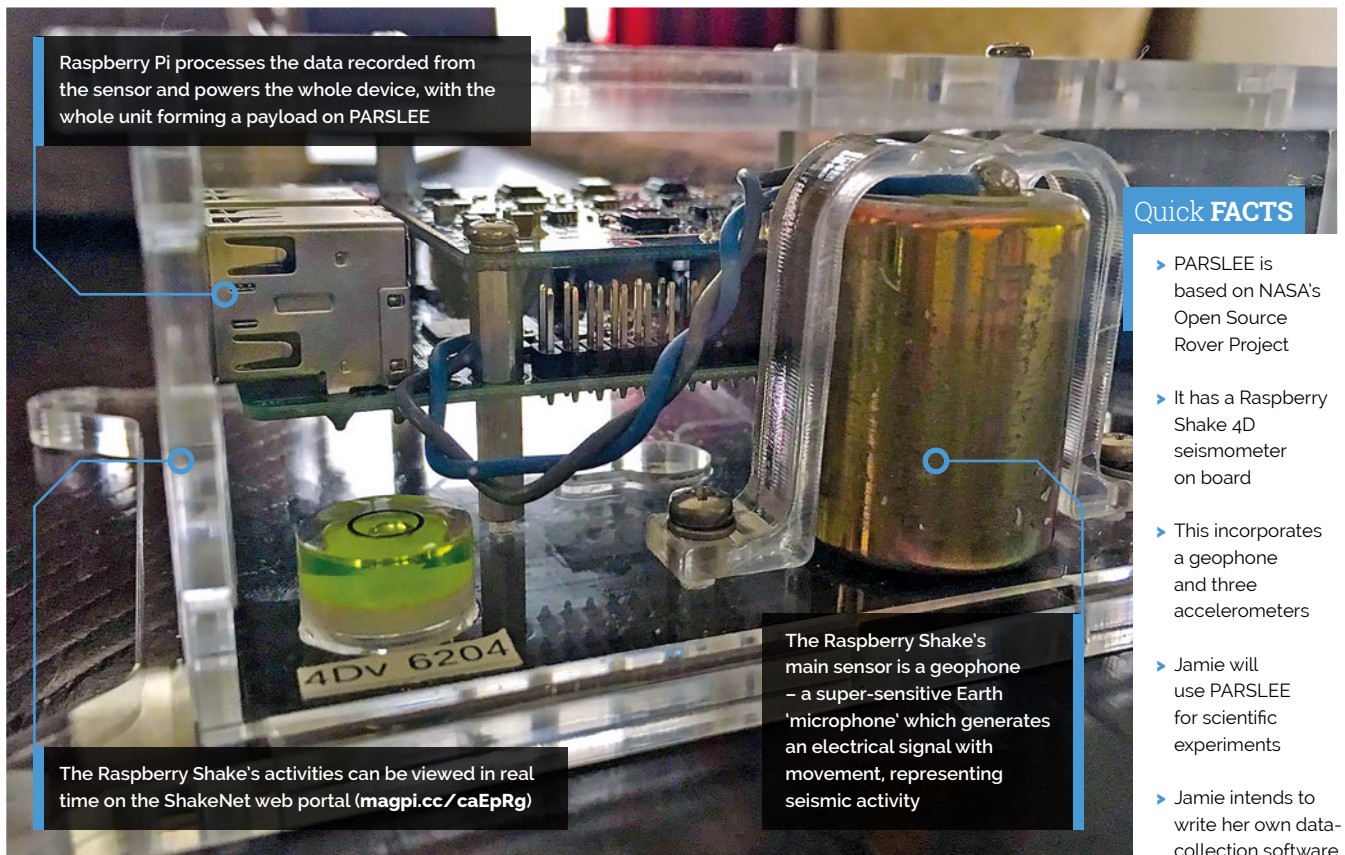
Raspberry Pi processes the data recorded from the sensor and powers the whole device, with the whole unit forming a payload on PARSLEE

The Raspberry Shake's activities can be viewed in real time on the ShakeNet web portal (**magpi.cc/caEpRg**)

The Raspberry Shake's main sensor is a geophone – a super-sensitive Earth 'microphone' which generates an electrical signal with movement, representing seismic activity

Quick **FACTS**

> PARSLEE is based on NASA's Open Source Rover Project

> It has a Raspberry Shake 4D seismometer on board

> This incorporates a geophone and three accelerometers

> Jamie will use PARSLEE for scientific experiments

> Jamie intends to write her own data-collection software

> **A seismometer helps us understand our physical environment in a very different way than images from a camera** ▮

surrounding soil shifting and settling as it changes temperature throughout the day. We can do experiments on Earth that mimic this process and use what we learn to help us understand the lunar seismic data."

Jamie is also toying with optimum locations for the Shake-fitted rover. "The best planetary analogue environments are usually deserts, due to the lack of moisture and low vegetation," she reveals. "Places like dry lake beds, lava flows, and sand dunes all provide good challenges in terms of testing the rover's ability to manoeuvre and collect data, as well as to try out technology being developed with and for it."

One thing's for sure, it is set to travel and potentially make a scientific breakthrough: "Anyone can use the rover for DIY science experiments." ▮

▲ Jamie has been working on a payload-laden version of NASA's Open Source Rover (**magpi.cc/tkOULg**). *Credit: NASA-JPL*

Community water sprinkler

Saving water, several thousand lawns at a time, **Rob Zwetsloot** takes a look at this award-winning IoT sprinkler system

MAKER

Adarsh Ambati

A ninth-grade student from San Jose, California, who likes to tinker. He's a member of the Environmental wing of Mitty Advocacy Program, and Youth Council member of San Jose.

magpi.cc/xieeLD

▼ A flow chart explains how the system works

At any CoderDojo Coolest Projects event, you're bound to see incredible things built by young makers. At Coolest Projects USA, we had the chance to talk to Adarsh Ambati about his community sprinkler and we were, frankly, amazed.

"The extreme, record-breaking drought in California inspired me to think of innovative ways to save water," Adarsh tells us. "While going to school in the rain one day, I saw one of my neighbours with their sprinklers on, creating run-offs. Through research, I found that 25 percent of water used in an average American household is wasted each day due to overwatering and inefficient watering methods. Thus, I developed a sprinkler system that is compliant with water regulations, to cost-effectively save water for entire neighbourhoods using a Raspberry Pi, moisture sensors, PyOWM (weather database), and by utilising free social media networks like Twitter."

Efficient watering

In California, it's very hot year round, so if you want a lush, green lawn you need to keep the grass watered. The record-breaking drought Adarsh was referring to resulted in extreme limitations on how much you could water your grass. The problem is, unless you have a very expensive sprinkler system, it's easy to water the grass when it's not needed.

"The goal of my project is to save water wasted during general-purpose landscape irrigation of an entire neighbourhood, by building a moisture sensor-based smart sprinkler system that integrates real-time weather forecast data to provide only optimum levels of water required," Adarsh explains. "It will also have Twitter capabilities that will be able to publish information about when and how long to turn on the sprinklers, through the social networks. The residents in the community will subscribe to this information by following an account on Twitter, and utilise it to prevent water wasted during general-purpose landscaping and stay compliant with water regulations imposed in each area."

Using the Raspberry Pi, Adarsh was able to build a prototype for about $50 – a lot cheaper than 'smart' sprinklers you can currently buy on the market.

"I piloted it with ten homes, so the cost per home is around $5," he reveals. "But since it has the potential to serve an entire community, the cost per home can be a few cents. For example, there are about 37,000 residents in Almaden Valley, San Jose (where I live). If there is an
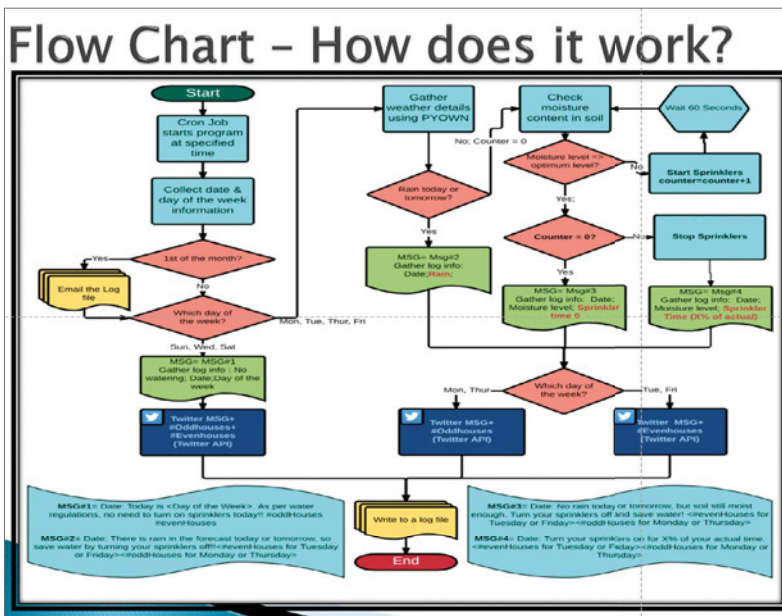
average of two to four residents per home, there should be 9250 to 18,500 homes. If I strategically place ten such prototypes, the cost per house would be five cents or less."

Massive saving

Adarsh continues, "Based on two months of data, 83% of the water used for outdoor landscape watering can be saved. The average household in northern California uses 100 gallons of water for outdoor landscaping on a daily basis. The ten homes in my pilot had the potential to save roughly 50,000 gallons over a two-month period, or 2500 gallons per month per home. At $0.007 per gallon, the savings equate to $209 per year, per home. For Almaden Valley alone, we have the potential to save around $2m to $4m per year!"

The results from Adarsh's test were presented to the San Jose City Council, and they were so impressed they're now considering putting similar systems in their public grass areas. Oh, and he also won the hardware project category at Coolest Projects USA. M
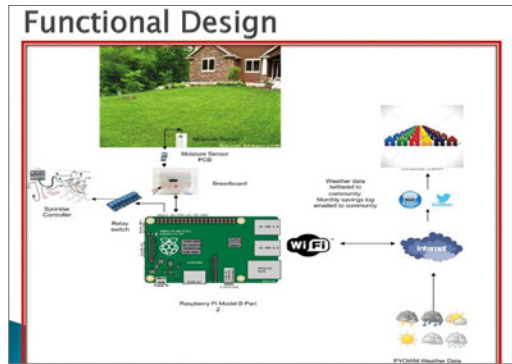
▲ The system could save a huge amount of water
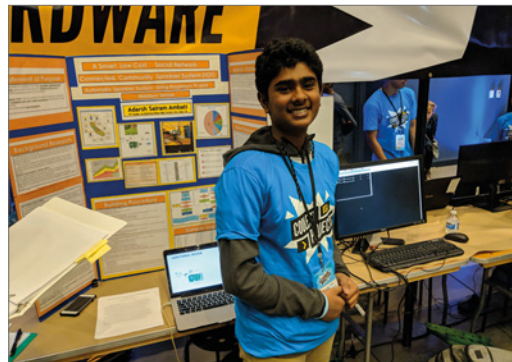
▲ Adarsh was a category winner at Coolest Projects USA 2019

> Based on two months of data, 83% of the water used for outdoor landscape watering can be saved

The system keeps detailed logs

The project was presented to Ardash's local government
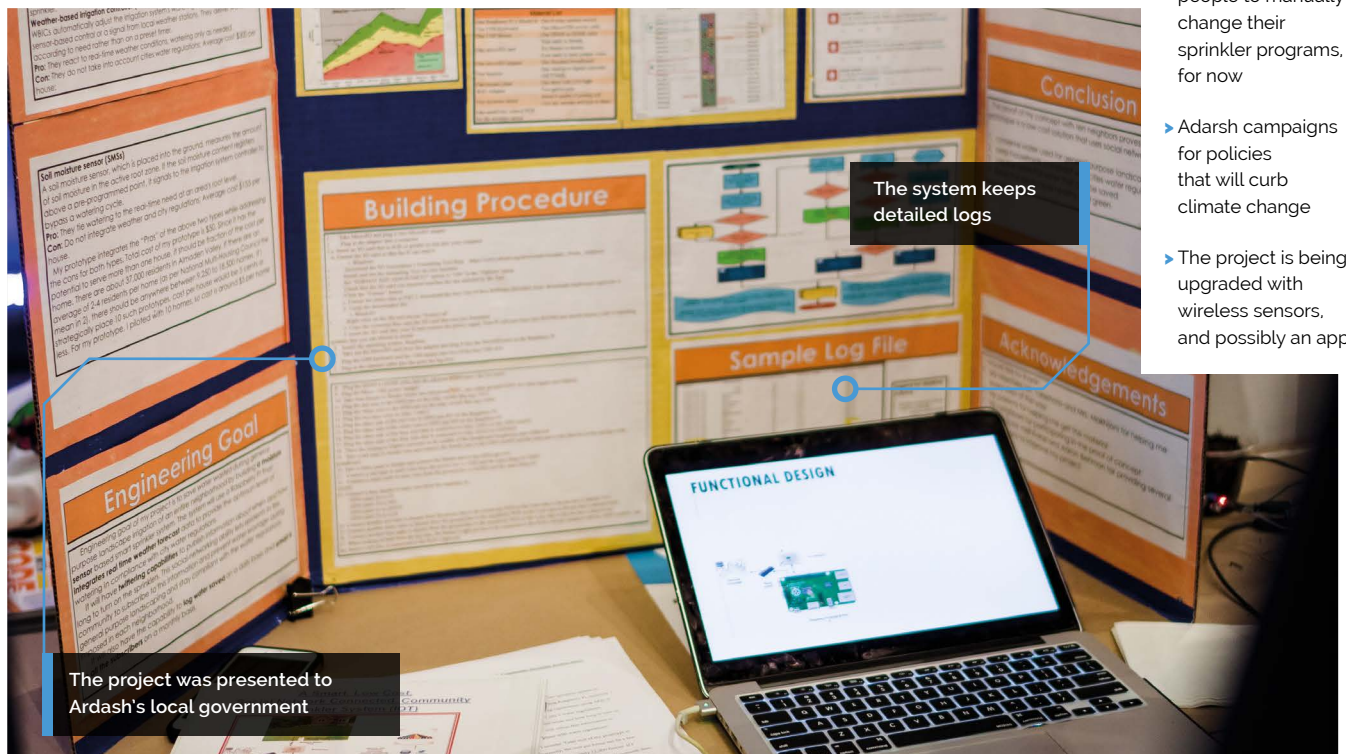
The LifeCam HD 3000 webcam includes a noise-cancelling microphone and records 720p HD video at 30 fps

When the button is pressed, a video call is made using Google Duo, which is installed on a Raspberry Pi 3 in a separate unit inside the house

PiNG uses a spray-painted intercom case from 1986, stripped out for fresh components such as a webcam, speaker, and wireless doorbell circuit

1986 PiNG Video Doorbell

Knock knock. Who's there? Martin Mander with a better way of announcing his arrival. **David Crookes** looks at his Raspberry Pi-based smart doorbell

MAKER

Martin Mander

Martin works as an analyst in Norwich where he lives with his wife, two children, and 13 Raspberry Pi boards, most of which are used in various projects around his home.

magpi.cc/jmdDHL

Smart doorbells have been ringing in the changes for home security for a while now, streaming video and audio to mobile devices whenever their button is pressed. They let you see and communicate with visitors regardless of whether you're in or out. While one of the most popular devices is Ring, however, Martin Mander opted to create his own.

"I took it as a personal challenge to build something with similar functions to Ring with a Raspberry Pi for a much lower cost," he says, indulging his passion for taking old tech and giving it a new lease of life using Raspberry Pi. And so, the idea for the PiNG video doorbell was born.

"At its most basic, it's just a Raspberry Pi, a webcam, a button, and an amplified speaker, which many folks have lying around already," Martin says. What's more, he reckons it is simple

enough for others to recreate and he actually created it with this in mind.

Securing the system

For a long time, Martin kept putting his project to one side as he sought an easy way to make video calls via Raspberry Pi. He noted the release of a browser-based version of Google Duo. "Before that, I was on the point of installing Android on a Raspberry Pi to see if I could use Duo that way," he says. "But the first test of Duo's web app was amazing. From that point, it was full speed ahead."

Martin found Duo worked best on a Raspberry Pi 3, which he originally connected to a Logitech webcam, plus a screen, mouse, and keyboard (later swapping the webcam for a LifeCam HD-3000 which has an integrated microphone). The real challenge was to make the setup work with a
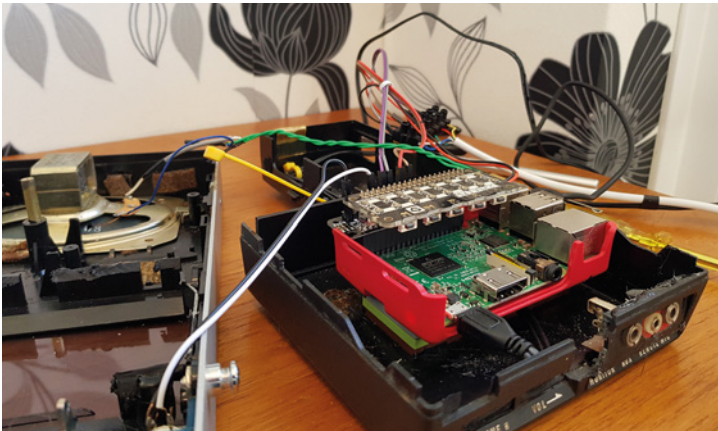
▲ Raspberry Pi and pHAT BEAT are placed inside an old cassette player wired to the intercom outside. The pHAT BEAT is also connected to the player's original speaker so that speech and the chime can be heard

◄ Call incoming…

▲ When the intercom button is pressed, a video call can be made with the visitor

Quick **FACTS**

> PiNG uses the video-calling app Google Duo

> A GPIO-connected lever microswitch starts calls

> Raspberry Pi 3B+ sends calls to mobiles within four seconds

> It would cost less than £50 to build

> A chime also sounds inside the house

❝ The first test of Duo's web app was amazing ❞

headless Raspberry Pi and buttons. This involved coding mouse movements, clicks, and keyboard strokes using the Python module PyUserInput.

"When someone presses PiNG's button (which is a standard doorbell I bought on eBay), the script uses the PyUserInput module to move the cursor around the screen, emulating mouse clicks and keystrokes to control the Google Duo web app, which is running all of the time," Martin says. "Once the call is ended, the script is ready straight away for the next button press." Now, Martin was ready to find his new creation a home of its own.

Playing it safe

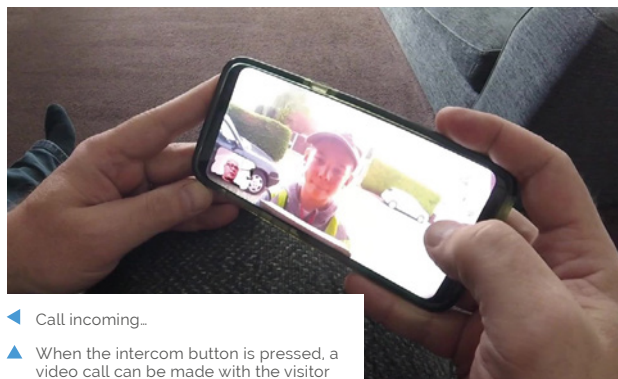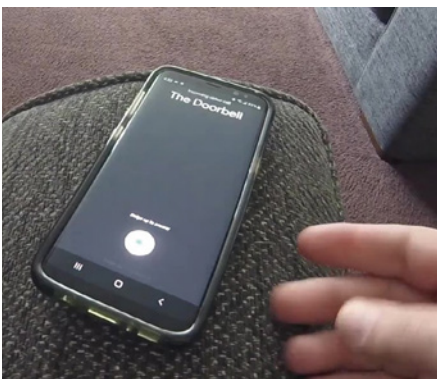A few years ago, Martin bought three intercoms originally released in 1986, and he decided to use one of those. "It's a bit unusual but still obviously a doorbell," he says. But since he didn't want Raspberry Pi or any components hanging outside the house "for security and connectivity reasons", he decided on a two-part construction. "The outside part is just a case for the button, speaker,
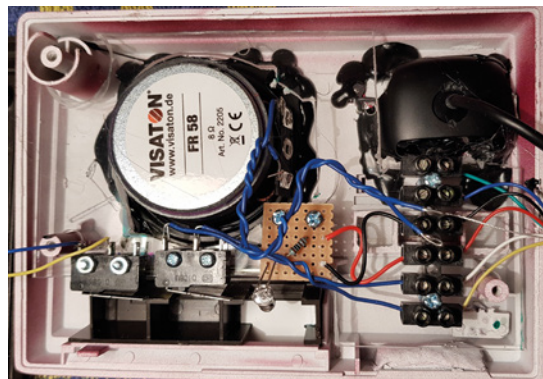
◄ Once these components were placed inside the intercom, the box was mounted to the exterior of the house using strong Velcro pads

and webcam," he reveals. The project's Raspberry Pi, meanwhile, is housed in an old, stripped-out tape machine located inside the porch and connected to the intercom using a six-core alarm cable with connection blocks at each end. Once it's fired up, the visitor presses the button and Martin can see and hear the person on his phone.

"I got a call from a delivery person while out for a lunchtime walk and it was a huge moment for me," he recalls of its first real life test. "Thankfully, he seemed unfazed by his intercom experience and we had a good two-way conversation about where to leave the parcel." Ⓜ

Smart Palm Greenhouse

Built to germinate the seeds of an endangered species of palm tree, this mini greenhouse features numerous sensors and automated plant-care systems. **Phil King** digs further

MAKER

Thomas Geers

Having worked as a CNC operator at a well-known German tyre manufacturer for 15 years, Thomas is now a freelance stock photographer.

thomas-geers.de

Thomas Geers is a fan of palm trees, although they're not very common in Germany, where he lives. Seeking to help conserve an endangered species of palm tree – the Chilean honey (or wine) palm, *Jubaea chi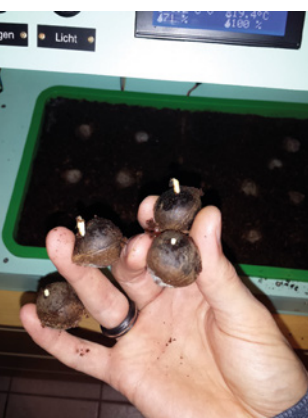lensis* – he bought some seeds in the summer of 2018. One issue, however, was their long germination time: up to a year. This inspired Thomas to build a fully automated greenhouse to ensure stable conditions for the seeds and possibly encourage faster germination.

After purchasing a standard mini greenhouse from a hardware store, he modified it with a homemade housing. "The housing consists of a wooden frame and 1 cm-thick plywood, which are bolted together," he tells us. "The wood is painted with a weatherproof wood glaze."

He then equipped it with various sensors and automated systems for watering, heating, lighting, and air humidity – all controlled by a Python script running on a Raspberry Pi 3B+ linked to a relay board.

> **"** Twelve spray nozzles in the housing cover will be activated as soon as the soil is too dry **"**

Automatic for the plants

"For automatic irrigation, there are twelve spray nozzles in the housing cover which will be activated as soon as the soil is too dry," explains Thomas. "For this purpose, there is a capacitive earth moisture sensor (Giesomat) in the earth."

Another key factor is maintaining an ideal temperature for the seeds. For this, Thomas placed a DS18B20 sensor in the soil and a heating mat underneath the container. "As soon as the earth temperature falls below a set value, the mat heats up the earth," says Thomas.

The automated lighting system comprises two LED plant lights which are switched on and off via a day/night time control – 12 hours on, 12 hours off.

Last but not least, there's an automated air humidity control system: "In the cover housing are two small fans, which are either time-controlled or, depending on the humidity in the greenhouse, allow an exchange of air to prevent mould."

In addition, Thomas has installed four switches so that he can turn all devices on and off individually and manually, if needed. A small LCD screen shows the temperature and moisture/humidity levels for the soil and air, with arrows indicating which values are currently being adjusted.

▼ Four of the germinated honey palm seeds. So far, out of 18 planted in the smart greenhouse, 11 have sprouted
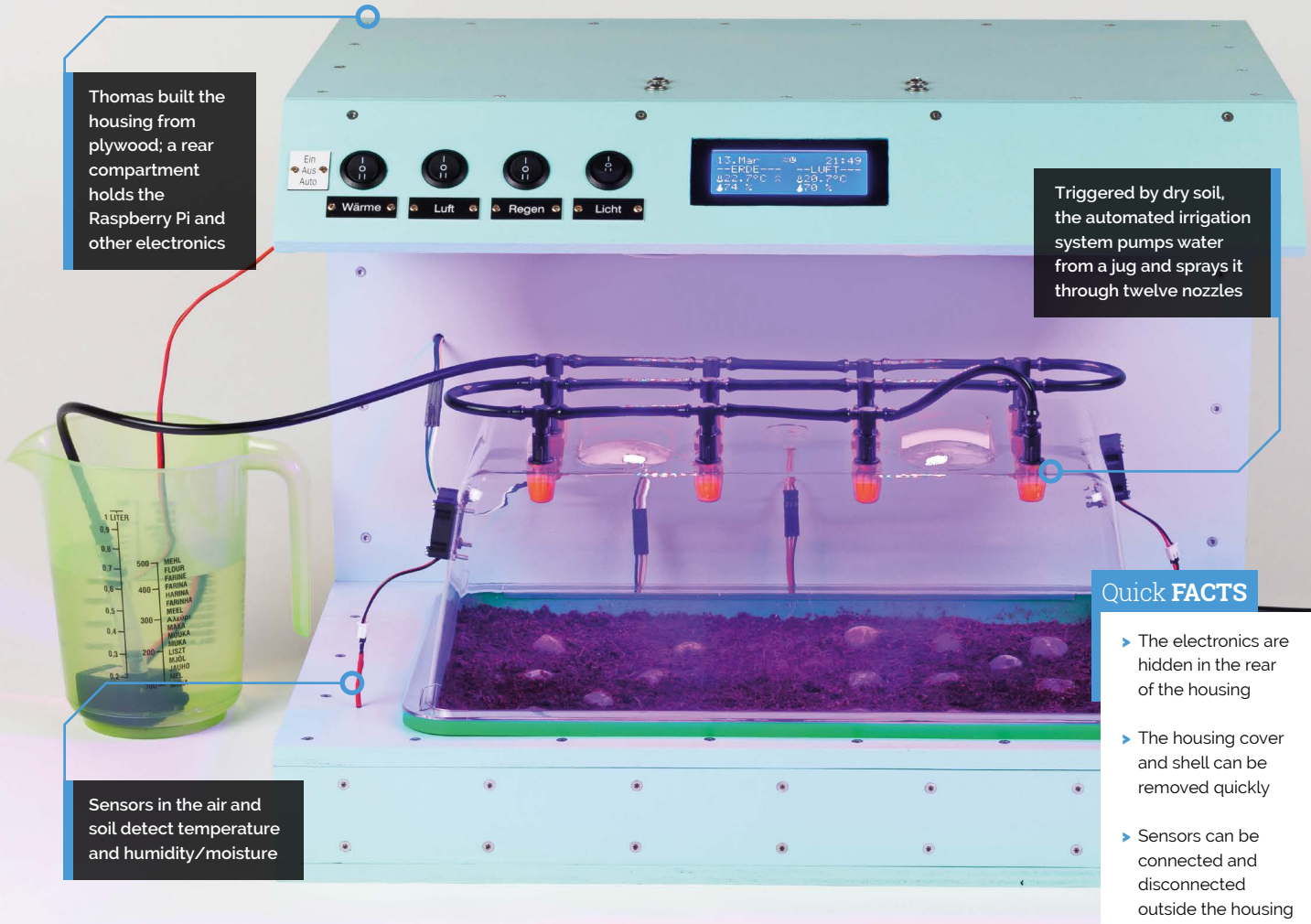
Thomas built the housing from plywood; a rear compartment holds the Raspberry Pi and other electronics

Triggered by dry soil, the automated irrigation system pumps water from a jug and sprays it through twelve nozzles

Sensors in the air and soil detect temperature and humidity/moisture

Quick FACTS

> The electronics are hidden in the rear of the housing

> The housing cover and shell can be removed quickly

> Sensors can be connected and disconnected outside the housing

> Thomas also has a squirrel/bird box observed by a PiNoir camera…

> Two great tits make this their home. See the live stream at **magpi.cc/DsAMqp**

▲ Four switches enable manual control of heat, fans, watering, and light, while an LCD screen displays current temperature and moisture levels

◀ With the rear panel removed, we can see the Raspberry Pi 3B+, relay board, and other electronics. In the housing cover, two LED plant lamps provide light

The project took Thomas two months to complete. "The hardest thing for me was programming, as this was my first Raspberry Pi project and I had no experience with Python," he reveals. "However, I had very good and friendly support from the German Raspberry Pi Forum."

Sprouting seeds

Since the honey palms come from Chile, he set the greenhouse temperature range to the average for that country, and the soil moisture and air humidity values to what felt right.

So far, the results have been impressive, with a much improved germination rate. In the seven months before the smart greenhouse was ready to use, only six out of the 24 seeds germinated in a normal thermostat-controlled greenhouse. "In the eighth month, all remaining 18 seeds were placed in the [smart] greenhouse; eleven seeds sprouted this month," says Thomas. "The palm trees will be planted in my garden someday." **M**

Warning!
Mains electricity

The lamps in this project use mains electricity. Please be careful!

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
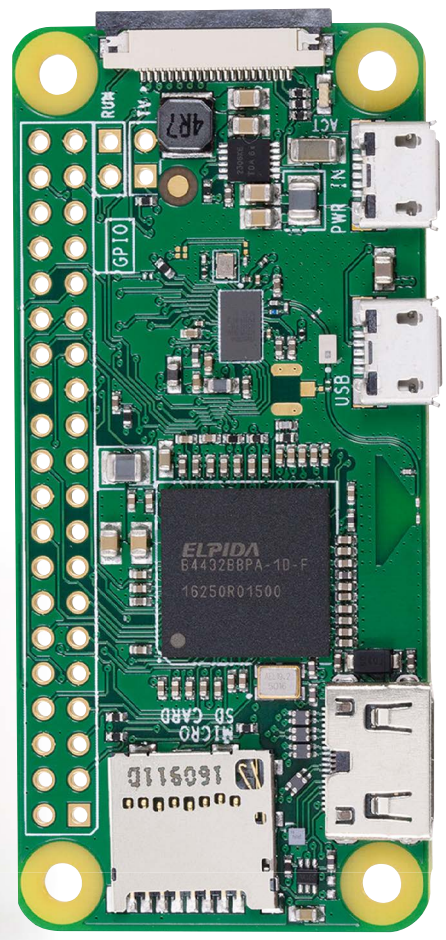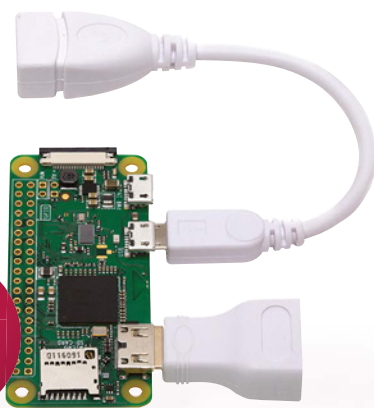
📨 Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

Introducing
Raspberry

The dual-display **Raspberry Pi 4** is here to redefine personal computing

By Gareth Halfacree

A new, 28 nm system-on-chip with powerful ARM Cortex-A72 processing cores. The first new graphics processor in Raspberry Pi's history. Up to 4GB of high-speed LPDDR4 memory. Two high-bandwidth USB 3.0 ports. Dual HDMI 2.0 outputs, capable of driving a pair of 4K displays (at 30 fps, or one at 60 fps). In short: very new, very powerful, and very exciting.

Designed as a true PC replacement for a lot of use-cases, Raspberry Pi 4 is the most impressive Raspberry Pi yet – and benchmark testing proves it's far from being all talk and no substance.

Pi 4
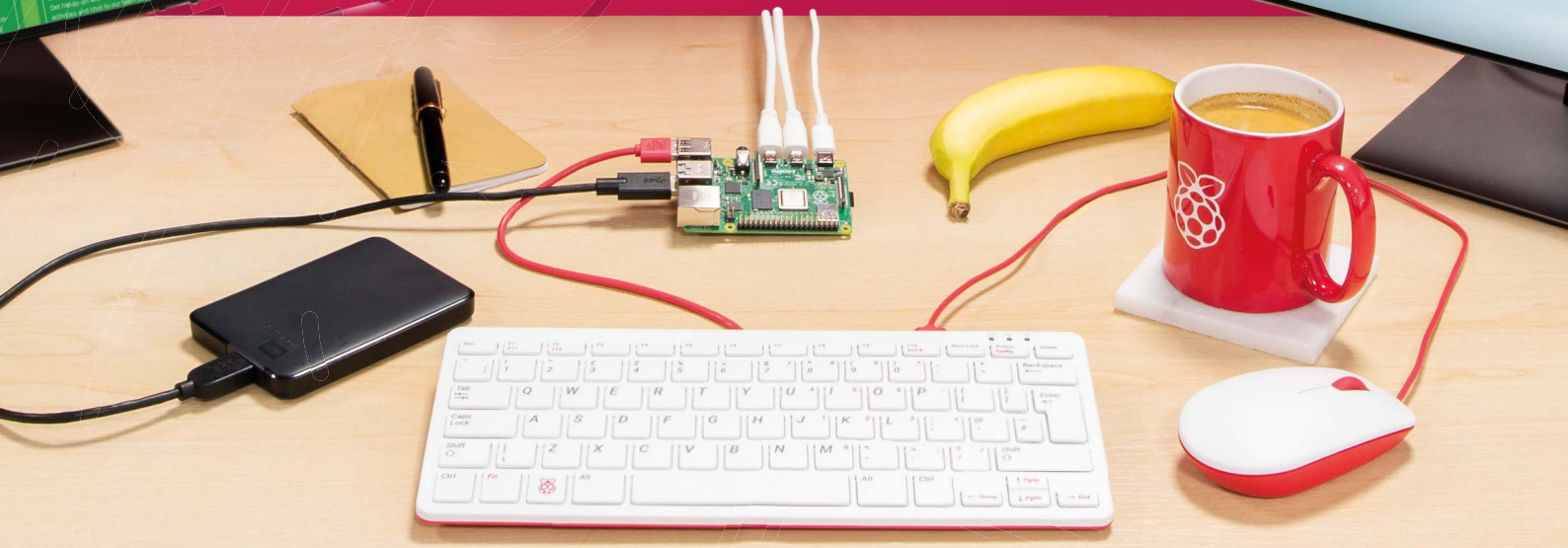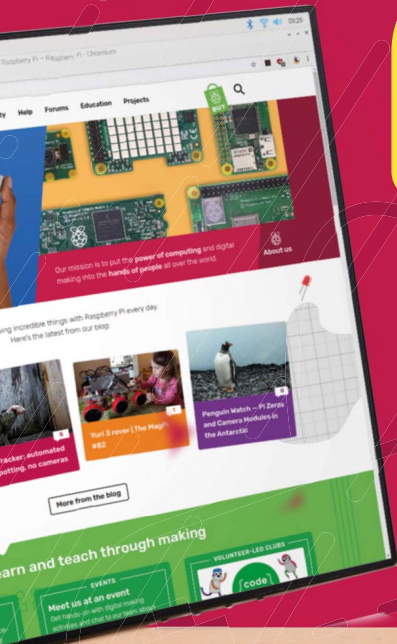
Get to know
Raspberry Pi 4

Raspberry Pi 4 marks a major Raspberry Pi family redesign

Specifications

SoC: Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit @ 1.5GHz

GPU: Broadcom VideoCore VI

NETWORKING: 2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN

RAM: 1GB, 2GB, or 4GB LPDDR4 SDRAM

BLUETOOTH: Bluetooth 5.0, Bluetooth Low Energy (BLE)

GPIO: 40-pin GPIO header, populated

STORAGE: microSD

PORTS: 2 × micro-HDMI 2.0, 3.5 mm analogue audio-video jack, 2 × USB 2.0, 2 × USB 3.0, Gigabit Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

DIMENSIONS: 88 mm × 58 mm × 19.5 mm, 46 g
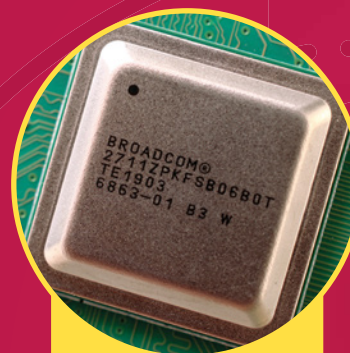
A | CPU
The new BCM2711B0 system-on-chip offers an impressive performance boost over its predecessors

B | POWER
The move to a USB Type-C connector for power allows Raspberry Pi 4 to support higher-current USB devices

QuickStart Guide

Raspberry Pi 4 is directly compatible with the 3B+ and all previous Raspberry Pi models, but the operating system has been updated to add support for the new system-on-chip which drives it. The easiest way to start from scratch is via our QuickStart Guide (page 38), which will guide you through setting up the latest NOOBS installer on a microSD card.
magpi.cc/quickstart
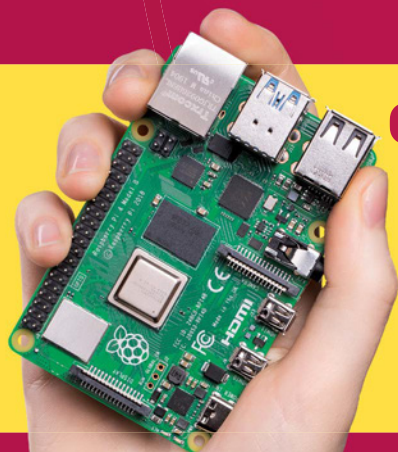
C | RAM

A move to up to 4GB of LPDDR4 memory, from the LPDDR2 of previous designs, increases performance further
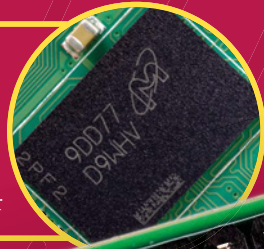
D | ETHERNET

The Ethernet port, relocated to the top-right of the board, now offers full-speed network connectivity with no bottlenecks

F | USB

Two USB 3.0 ports, centre, offer high-speed connectivity for external devices including storage and accelerator hardware

E | DUAL DISPLAYS

The two micro-HDMI connectors enable Raspberry Pi 4 to drive two 4K displays at up to 4Kp30, or a single display at up to 4Kp60

Eben Upton on Raspberry Pi 4

A brand-new processor, upgraded video capabilities, up to four times the memory – what did it take to make Raspberry Pi 4?

Eben Upton

Eben is the creator of Raspberry Pi and a co-founder of the Raspberry Pi Foundation. He is the CEO of Raspberry Pi Trading Ltd.

"I guess there's a question, which is 'why now, why not in a year's time?' Which is the sort of time line we previously indicated," says Eben Upton, co-founder of the Raspberry Pi Foundation, on the timing of Raspberry Pi 4's release. "Broadcom has been working on silicon for it for a little while, and the silicon came good earlier than I was expecting.

"This is the B0 step of the silicon. BCM2835, which was new on 40 nm, was equivalently radical at the time. The version we shipped there is 2835C2, so we'd had an A0, a B0, a C0, a C1, and a C2 to get to a shippable product. This one got shippable by B0, and that's taken a year out of the conservative schedule that we'd been communicating to people."
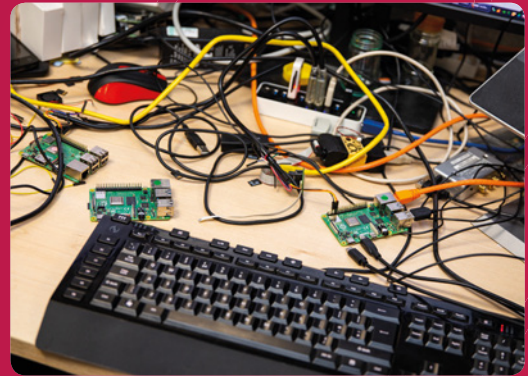
▲ Raspberry Pi 4 boards being tested and having software written and updated for them

Backwards compatibility

"It's very substantially backwards compatible," Eben promises. "You don't like to say 'perfectly backwards compatible,' because I'm sure people will find ways in which it's not. At launch, for example, I suspect there will be monitors that a Raspberry Pi 3 can drive that a Raspberry Pi 4 can't; but that will be fixed over time.

"I think we've met our goals for backwards compatibility. Which is good, because otherwise you tear your software team apart. You either have to sunset old products, which you know we hate doing, or you end up with two software teams: one to move the old product forward, and one to move the new product forward."

The pocket-sized PC

"It's a PC replacement. I mean, we've always talked about Raspberry Pi as being a PC, and that's become steadily more credible, I think, over the generations," says Eben. "I think this one takes it over the line where a lot of users will sit down in front of it and not really perceive a difference.

"You talk about the things that take you into PC land? PCs drive two displays. You know, you're not really a real PC if you don't drive two displays, right? If you think about the person you're speaking to on the phone in the bank, they'll have two monitors: one to put your account details on, and one to put the product that they're selling you on. We think this should break through very nicely into the thin client market and we're working with Citrix to make sure that their stack works on it on launch day.

A Raspberry Pi engineer's desk, where hardware is tested thoroughly and bananas are eaten

A challenging design

"Obviously the ports have moved around, and that's really a routing thing," Eben explains. "The board is within a millimetre of not working, and there wasn't enough routing resource to bring the Ethernet signalling down to the bottom-right of the board.

> 〃 The Easter egg is under the USB-C connector; it's James's signature 〃

"Probably the biggest challenge is the DRAM. If you look how close that SoC and that DRAM are to each other, you've got a 32-bit DRAM interface in that tiny little space, with some length-matching between the signals and the signals properly isolated from each other. If you were to desolder the USB-C connector, you'll see James [Adams, director of hardware] has signed the board. So, the Easter egg is under the USB-C connector; it's James's signature, because I think he feels it's the nicest piece of work he's ever done, and it was very close to not being doable.

Building the BCM2711

"This has been a more complicated development than previous ones because previous ones have been on the same process node and we've basically just been – 'just' been – bolting larger ARM cores onto an existing chip," says Eben of the work that has gone into the new system-on-chip (SoC).

"This one's on a new process node, so this one's on 28 nm. Obviously, it's got all these new features, so we've kind of moved it from being a 1080p-class chip to being a 4K-class chip. New process node, new memory technology, new multimedia IP [intellectual property], lots and lots of change. It's a full-chip project."

"The original prototypes, the A0 prototypes, are actually about five millimetres longer. They're five millimetres bigger in X than the historical board, but he was able to squeeze it back down. My contribution was largely to go to his desk and say 'is the board back at the right size yet?' every day for about six months. I should have signed the board as well – I deserve half-credit!"

Benchmarking
Raspberry Pi 4

A full-chip redesign, the first in the history of Raspberry Pi, has unlocked new levels of performance

I t's not hard to see where **Raspberry Pi 4 improves on its predecessor.** The brand-new BCM2711B0 system-on-chip has more powerful processing cores, the first upgrade to the graphics processor in the history of the project, and vastly improved bandwidth for both memory and external hardware. Gone is the single-lane USB bottleneck which hampered performance on older models, and Raspberry Pi 4 shines as a result.

Spec comparison

Internally, there's little left unchanged between the Raspberry Pi 3 family and Raspberry Pi 4. The SoC is now built on a 28 nm semiconductor process node, down from 40 nm, and packs the significantly more powerful ARM Cortex-A72 processor cores. The memory has moved from LPDDR2 to LPDDR4, skipping a generation and improving bandwidth, and is for the first time available in capacities over 1GB with 2GB and 4GB versions available on launch day.

Even the graphics processor has been upgraded: the Broadcom VideoCore IV, which has been a staple since the original Raspberry Pi Model B, has been replaced with the more powerful VideoCore VI, unlocking both performance and dual-4K-display capabilities.

Add in the full-speed Gigabit Ethernet and USB 3.0 ports and you've got a significant upgrade on your hands.

Linpack

A synthetic benchmark originally developed for supercomputers, Linpack offers a glimpse at just how far the Raspberry Pi family has come. This version – ported by Roy Longbottom – comes in three variants: the fast single-precision (SP), slower double-precision (DP), and a single-precision variant accelerated using the NEON instructions available in Raspberry Pi 2 and above (NEON).

Linpack (SP, MIPS)
Linpack (DP, MIPS)
Linpack (SP NEON, MIPS)
Higher is better

	Linpack (SP, MIPS)	Linpack (DP, MIPS)	Linpack (SP NEON, MIPS)
Raspberry Pi A	51.41	36.88	
Raspberry Pi B	50.26	35.81	
Raspberry Pi A+	51.14	37.26	
Raspberry Pi B+	49.99	35.74	
Raspberry Pi Zero	64.47	40.38	
Raspberry Pi Zero W	63.77	39.51	
Raspberry Pi 2	158.35	145.72	298.7
Raspberry Pi 3	193.79	179.87	460.91
Raspberry Pi 3 B+	224.89	209.23	526.75
Raspberry Pi 3 A+	225.9	208.39	536.23
Raspberry Pi 4 B	925.47	748.73	2037.33

Python GPIO Zero

Sitting somewhere between a synthetic and a real-world benchmark, here the Python GPIO Zero library is used to toggle a GPIO pin on and off as quickly as possible while a frequency counter measures the switching rate in kilohertz (kHz). This test is boosted by CPU speed.

GPIO Zero (kHz)
Higher is better

Model	kHz
Raspberry Pi A	1.301
Raspberry Pi B	1.201
Raspberry Pi A+	1.201
Raspberry Pi B+	1.201
Raspberry Pi Zero	1.702
Raspberry Pi Zero W	1.752
Raspberry Pi 2	8.307
Raspberry Pi 3	13.51
Raspberry Pi 3 B+	16.11
Raspberry Pi 3 A+	16.02
Raspberry Pi 4 B	50.84

File Compression

An example of a real-world workload, this benchmark takes a file and compresses it using the bzip2 algorithm and measures the elapsed time in seconds. For Raspberry Pi models with more than one processing core – the Raspberry Pi 2 and 3 family, and Raspberry Pi 4 – the test is run a second time using the multi-threaded lbzip2.

File Compression (single-threaded, seconds)
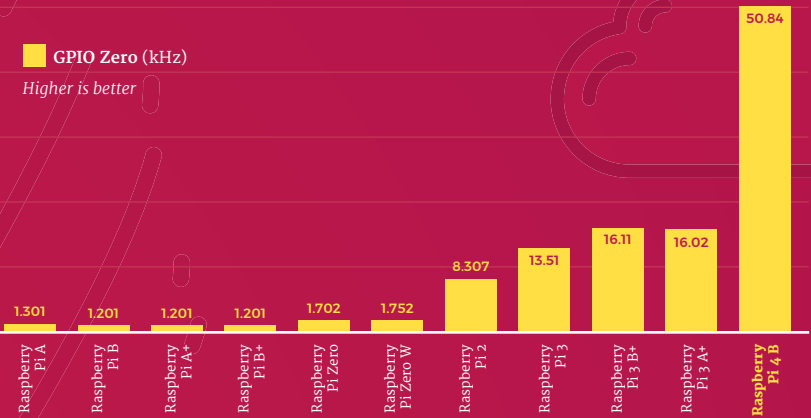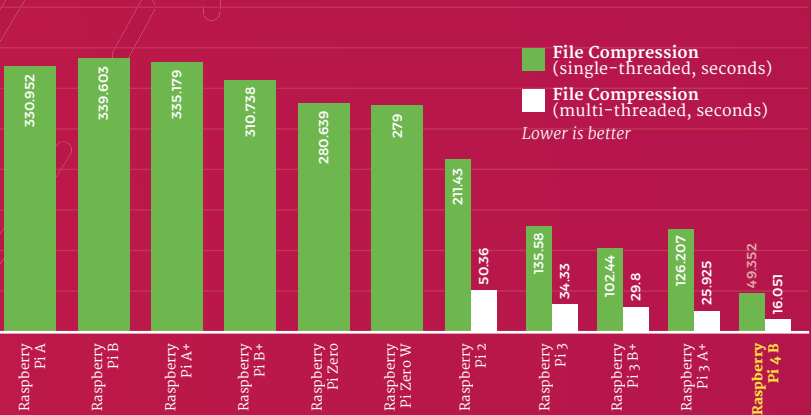**File Compression** (multi-threaded, seconds)
Lower is better

Model	Single-threaded	Multi-threaded
Raspberry Pi A	330.952	
Raspberry Pi B	359.603	
Raspberry Pi A+	335.179	
Raspberry Pi B+	310.738	
Raspberry Pi Zero	280.639	
Raspberry Pi Zero W	279	
Raspberry Pi 2	211.43	50.36
Raspberry Pi 3	135.58	34.33
Raspberry Pi 3 B+	102.44	29.8
Raspberry Pi 3 A+	126.207	25.925
Raspberry Pi 4 B	49.352	16.051

Speedometer 2.0

Speedometer 2.0 measures the responsiveness of the Chromium web browser by running a web application – a to-do list – and measuring how many times the application can be completed each minute. Here, performance hinges not only on CPU performance but on memory speed and capacity – the test proved too much for Raspberry Pi A+.

Speedometer 2.0 Browser Benchmark (runs per minute)
Higher is better

Model	runs/min
Raspberry Pi A	0.667
Raspberry Pi B	0.596
Raspberry Pi B+	0.623
Raspberry Pi Zero	1.04
Raspberry Pi Zero W	0.961
Raspberry Pi 2	4.138
Raspberry Pi 3	7.56
Raspberry Pi 3 B+	8.55
Raspberry Pi 3 A+	6.35
Raspberry Pi 4 B	16.91
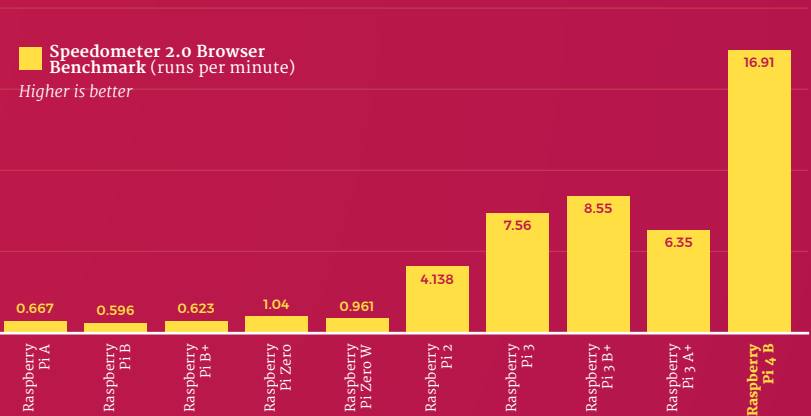
OpenArena Time Demo

The new VideoCore VI gives Raspberry Pi 4 a significant boost over its predecessors, as demonstrated in this gaming workload test: the Quake III-based OpenArena first-person shooter runs through its built-in demo as quickly as possible at a High Definition (1280×720) resolution, while the average frame rate in frames per second (fps) is recorded.

OpenArena Time Demo (fps)
Higher is better

Model	fps
Raspberry Pi 2	19.9
Raspberry Pi 3	27.6
Raspberry Pi 3 B+	27.8
Raspberry Pi 3 A+	27.2
Raspberry Pi 4 B	41.4
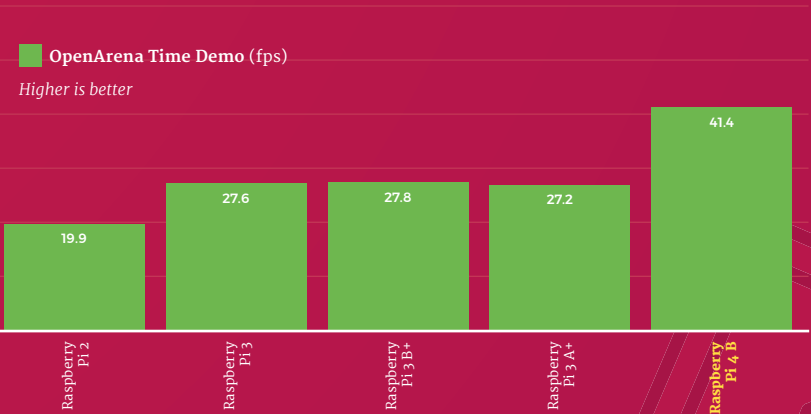
GIMP Image Editing

Another real-world test, the popular open-source GIMP image-editing suite is used to process a high-resolution image and save it as a PNG. Like the Speedometer 2.0 benchmark, this is heavily reliant on both CPU and memory performance – and extra memory really helps some of the models on test.

GIMP Image Editing (seconds)
Lower is better

Model	Value
Raspberry Pi A	459.899
Raspberry Pi B	486.239
Raspberry Pi A+	492.178
Raspberry Pi B+	419.946
Raspberry Pi Zero	411.685
Raspberry Pi Zero W	396.08
Raspberry Pi 2	126.22
Raspberry Pi 3	82.49
Raspberry Pi 3 B+	67.01
Raspberry Pi 3 A+	119.775
Raspberry Pi 4 B	47.35
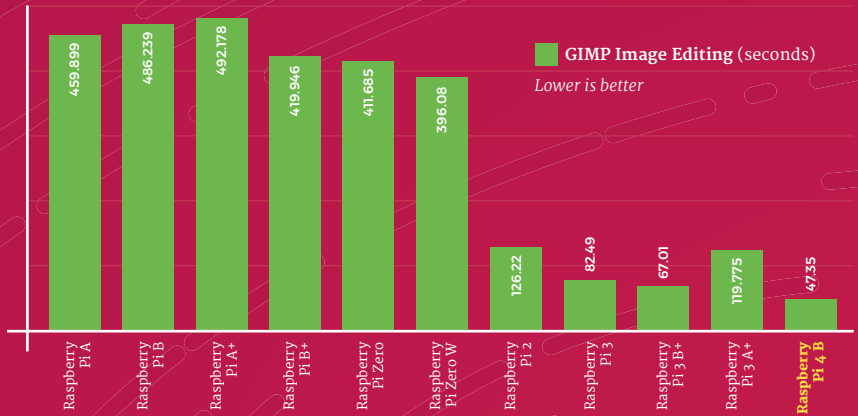
USB Storage Throughput

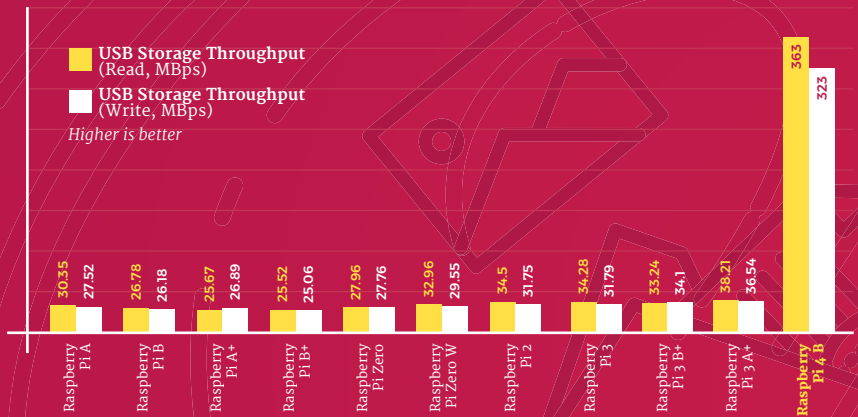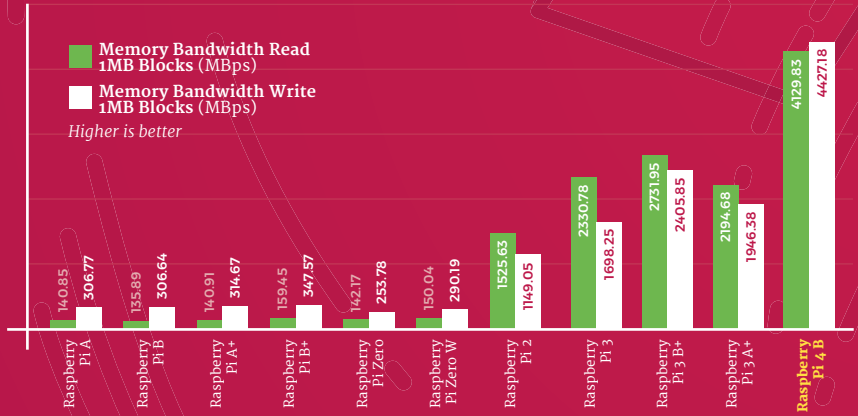Raspberry Pi 4's new USB 3.0 ports offer a massive bandwidth boost, which has a big impact on the performance of external storage devices. Here, a solid-state drive (SSD) is connected via a USB adapter and the average read and write throughput measured in megabytes per second (MBps).

USB Storage Throughput (Read, MBps)
USB Storage Throughput (Write, MBps)
Higher is better

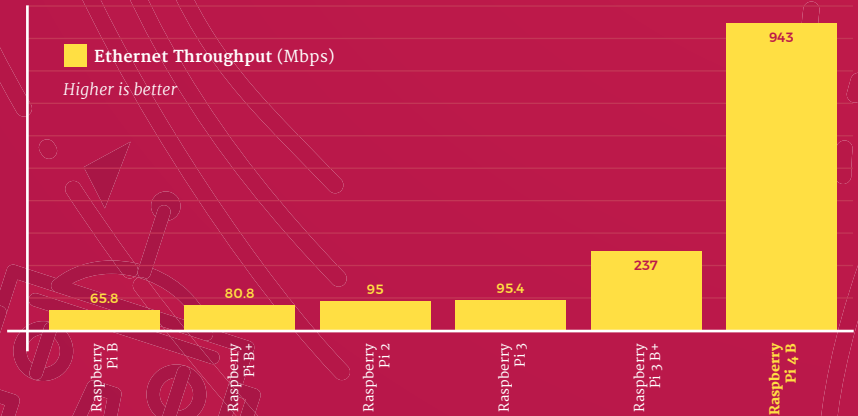| Model	Read	Write
Raspberry Pi A	30.35	27.52
Raspberry Pi B	26.78	26.18
Raspberry Pi A+	25.67	26.89
Raspberry Pi B+	25.52	25.06
Raspberry Pi Zero	27.96	27.76
Raspberry Pi Zero W	32.96	29.55
Raspberry Pi 2	34.5	31.75
Raspberry Pi 3	34.28	31.79
Raspberry Pi 3 B+	33.24	34.1
Raspberry Pi 3 A+	38.21	36.54
Raspberry Pi 4 B	363	323

Memory Bandwidth

Although many workloads are primarily limited by CPU speed, others rely on memory bandwidth – the rate at which data can be written to and read from RAM. In this benchmark, the RAMspeed/SMP tool is used to measure the read and write bandwidth for 1MB blocks in megabytes per second (MBps).

Memory Bandwidth Read 1MB Blocks (MBps)
Memory Bandwidth Write 1MB Blocks (MBps)
Higher is better

Model	Read	Write
Raspberry Pi A	140.85	306.77
Raspberry Pi B	135.89	306.64
Raspberry Pi A+	140.91	314.67
Raspberry Pi B+	159.45	347.57
Raspberry Pi Zero	142.17	253.78
Raspberry Pi Zero W	150.04	290.19
Raspberry Pi 2	1525.63	1149.05
Raspberry Pi 3	2330.78	1698.25
Raspberry Pi 3 B+	2731.95	2405.85
Raspberry Pi 3 A+	2194.68	1946.38
Raspberry Pi 4 B	4129.83	4427.18

Ethernet Throughput

While Raspberry Pi 3 Model B+ added Gigabit Ethernet connectivity, throughput on Raspberry Pi 4 is free from the single shared USB 2.0 channel to the SoC. The throughput of all Raspberry Pi models with a built-in Ethernet port is measured using the iperf3 tool, showing the average network throughput (in megabits per second) over several runs.
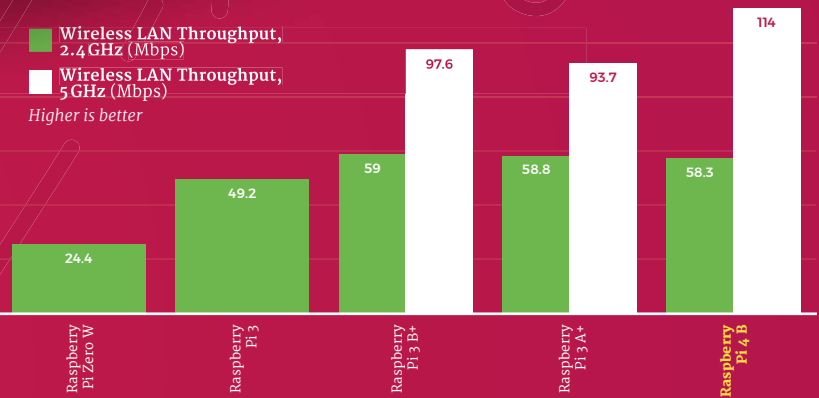
Ethernet Throughput (Mbps)
Higher is better

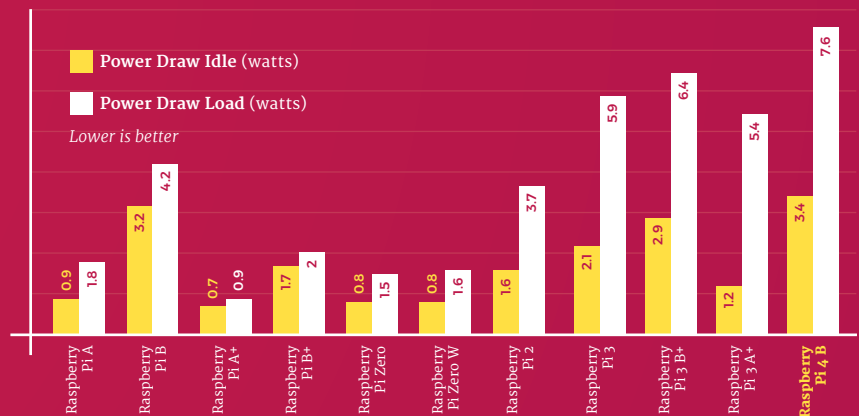| Model	Value
Raspberry Pi B	65.8
Raspberry Pi B+	80.8
Raspberry Pi 2	95
Raspberry Pi 3	95.4
Raspberry Pi 3 B+	237
Raspberry Pi 4 B	943

▶ The official case for Raspberry Pi 4

Wireless LAN Throughput

For this wireless networking test, an ideal environment is created: a Raspberry Pi is placed in line-of-sight of an 802.11ac router, and a wired laptop uses iperf3 to measure the average throughput over several runs. For models with dual-band 2.4 / 5 GHz radios, the test is run on both bands.

Wireless LAN Throughput, 2.4 GHz (Mbps)
Wireless LAN Throughput, 5 GHz (Mbps)
Higher is better

| Model	2.4 GHz	5 GHz
Raspberry Pi Zero W	24.4	
Raspberry Pi 3	49.2	
Raspberry Pi 3 B+	59	97.6
Raspberry Pi 3 A+	58.8	93.7
Raspberry Pi 4 B	58.3	114

Power Draw

More performance typically means more power, and here each Raspberry Pi model is left running a CPU-intensive benchmark while an HDMI display and a USB keyboard and mouse are connected. The peak power draw in watts is measured from the wall, and then an 'idle' draw with a Raspberry Pi sat at the Raspbian desktop is measured for comparison.

Power Draw Idle (watts)
Power Draw Load (watts)
Lower is better

Model	Idle	Load
Raspberry Pi A	0.9	1.8
Raspberry Pi B	3.2	4.2
Raspberry Pi A+	0.7	0.9
Raspberry Pi B+	1.7	2
Raspberry Pi Zero	0.8	1.5
Raspberry Pi Zero W	0.8	1.6
Raspberry Pi 2	1.6	3.7
Raspberry Pi 3	2.1	5.9
Raspberry Pi 3 B+	2.9	6.4
Raspberry Pi 3 A+	1.2	5.4
Raspberry Pi 4 B	3.4	7.6

Thermal Performance

A higher power draw naturally leads to more heat. Here, Raspberry Pi 3B+ and Raspberry Pi 4 are left running a CPU-intensive benchmark for ten minutes before a thermal image is captured, demonstrating where the heat is generated and how it spreads throughout the board.

Raspberry Pi 3 B+   Raspberry Pi 4 B
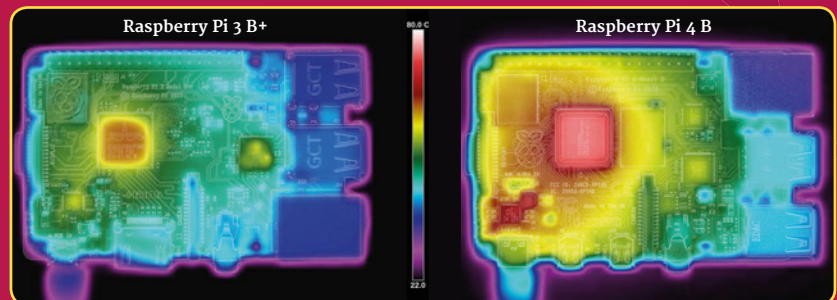
Simon Long on
Raspbian 'Buster'

User experience engineer **Simon Long** walks through the new features of Raspbian 'Buster' and its revamped user interface

Simon Long

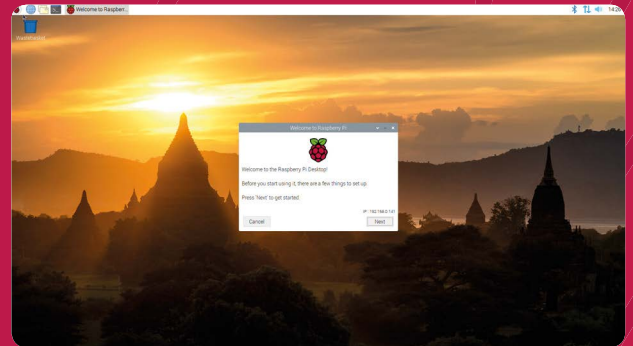Simon Long's work on user experience impacts everything you see and do on the Raspbian desktop

The launch of Raspberry Pi 4 brings not only new hardware but new software too: Raspbian 'Buster', a brand-new release – compatible, as always, with every Raspberry Pi model going right back to the pre-launch Alpha design – with a revamped, flatter user interface based on the upstream Debian 'Buster' Linux distribution.

Simon Long explains: "Due to the lack of obvious differences between Buster and Stretch, I wanted to do something to make it a bit more obvious that people actually had something new," of his new interface design. When we moved from Jessie to Stretch, there was a similar lack of major differences, and people wondered whether or not they actually had the new version – I wanted to avoid that this time. Also, the overall UI design in terms of the appearance of buttons, controls, and the like really hasn't changed significantly in the time I've been here – there have been some small tweaks, but it felt time for a change."

▲ The new Raspbian desktop offers a cleaner, more approachable interface – and lovely new wallpaper

Flatter is better

"The flatter appearance was driven by a few factors," Simon continues. "First, it does seem to be a general tendency in UI design in recent years that flatter, simpler designs are in, and fussier, more complex designs are out – iOS, Windows, and Android have all done the same sort of thing. Second, Eben is a big fan of flatter UIs, and he kept nudging me in that direction!

"It's a bit of a balancing act, though – you don't want to go too far and end up with just boring square boxes everywhere, which is why while things like corner radii have been reduced, they haven't been completely squared off.

"There's been a lot of experimentation with new designs; we toyed with things like changing the system font and considered numerous different ideas for the appearance of buttons, sliders, and scrollbars, and I think we've ended up with something that looks modern without looking too boring."

Mixing hardware and software

"Moving to a new Debian release is always a lot of work," Simon notes. "We have to take all the changes and patches we had created for the previous version and apply them to new versions of software in the current version, test it all, make sure it is still stable and that we haven't had performance regressions, and so on.

"That on its own is usually a challenge, but the fact that we were moving to new hardware at the same time added another dimension – when you find something has broken, you don't know if it's the new hardware, the new OS, or just that you've got something wrong yourself somewhere!"

The evolution of the desktop

"I'm really pleased with the way the new user interface design has come out," says Simon. "Because the design process was a gradual evolution over time, you don't realise the difference between where you started and where you've ended up, but once it was finished and I was applying the changes to existing images, the sudden switch from old to new just made everything look instantly better.

"I'd never really thought there was much wrong with the old design, but when you suddenly change to the new one, you think 'wow, that looks a lot better' – or at least I do!"

Under the hood

Not all improvements are immediately visible: "We're now using OpenGL to draw the desktop with hardware acceleration," Simon explains. "This is something which we've had as an experimental feature for a couple of years now – it's been an option in raspi-config to turn it on, but it's now the default mechanism. It means that any applications which use OpenGL should run significantly faster, and it means that things like OpenGL games are now usable on Raspberry Pi out of the box.

> **We are actually releasing Buster before Debian themselves do!**

"One interesting consequence of this is that we are actually releasing Buster before Debian themselves do! Some of the libraries which enable the OpenGL acceleration work much better in their Buster versions, so we have been using testing versions of Buster for several months and it makes more sense to release Buster software for Raspberry Pi 4 than it does to do all the work required to make this work on Stretch. Buster is in the final stages of testing by Debian – it is likely to be officially released within the next month or so – so this isn't a particularly risky thing to do, but it does mean that anyone using this release is getting it a bit early!" **M**

Places to Buy

UK & Ireland

Raspberry Pi Store
magpi.cc/retail-store

The Pi Hut
thepihut.com

OKdo
okdo.com

CPC
cpc.farnell.com

Pimoroni
pimoroni.com

North America

OKdo
okdo.com

PiShop.us
pishop.us

Adafruit
adafruit.com

Newark
newark.com

Micro Center
microcenter.com

Canakit
canakit.com

Europe

OKdo
okdo.com

BuyZero
buyzero.de

Kiwi Electronics
kiwi-electronics.nl

Sertronic
digitec.ch

Kubii
kubii.fr

SEMAF
electronics.semaf.at

Melopero
melopero.com

Totonic
pi-shop.ch

pi3g
pi3g.com

Electrokit
electrokit.com

Jkollerup
raspberrypi.dk

For a full list of approved resellers please go to **raspberrypi.org/products**

Raspberry Pi
QuickStart Guide

Setting up Raspberry Pi is pretty straightforward.
Just follow the advice of **Rosie Hattersley**

Congratulations on becoming a Raspberry Pi explorer. We're sure you'll enjoy discovering a whole new world of computing and the chance to handcraft your own games, control your own robots and machines, and share your experiences with other Raspberry Pi fanatics.

Getting started won't take long: just corral all the bits and bobs on our checklist, plus perhaps a funky case. Useful extras include some headphones or speakers if you're keen on using Raspberry Pi as a media centre or gaming machine.

To get set up, simply format your microSD card, download NOOBS, and run the Raspbian installer. This guide will lead through each step. You'll find the Raspbian OS, including coding programs and office software, all available to use. After that, the world of digital making with Raspberry Pi awaits you.

What you need
**All the bits and bobs you need
to set up a Raspberry Pi computer**

A Raspberry Pi
Whether you choose a Raspberry Pi 4, 3B+, 3B, Pi Zero, Zero W, or Zero WH (or an older model of Raspberry Pi), basic setup is the same. All Raspberry Pi computers run from a microSD card, use a USB power supply, and feature the same operating systems, programs, and games.
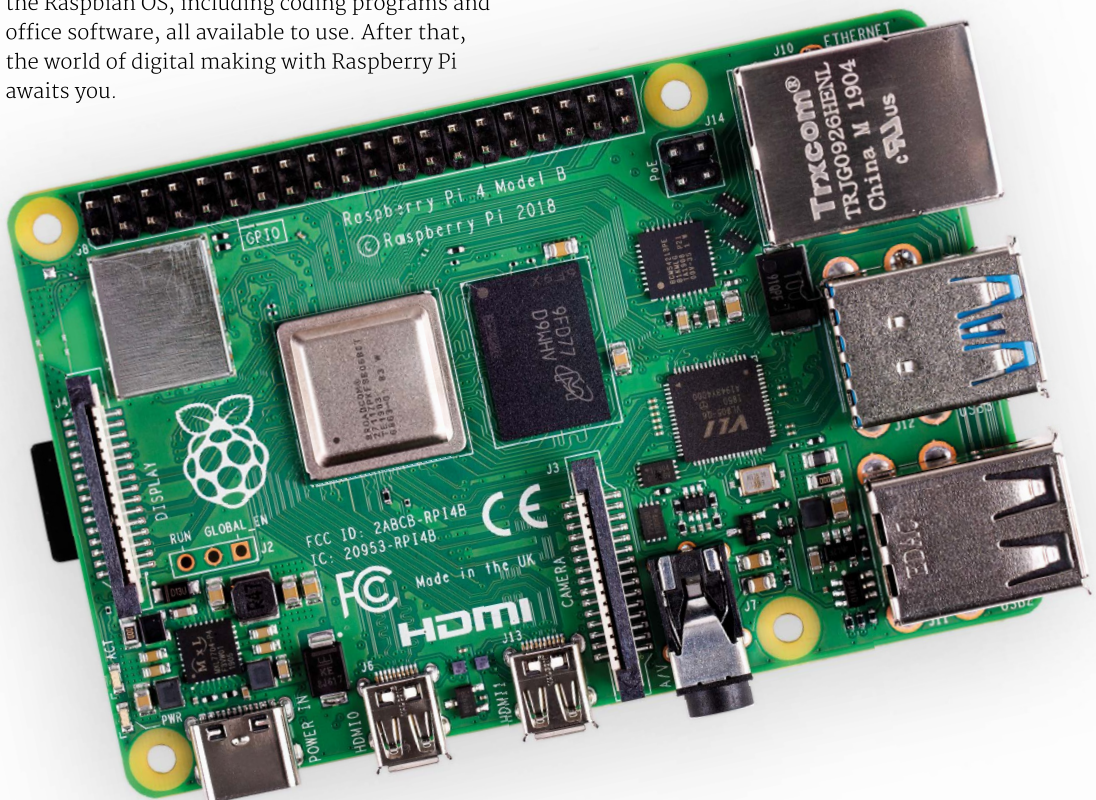
8GB microSD card

You'll need a microSD card with a capacity of 8GB or greater. Your Raspberry Pi uses it to store games, programs, and photo files and boots from your operating system, which runs from it. You'll also need a microSD card reader to connect the card to a PC, Mac, or Linux computer.

Mac or PC computer

You'll need a Windows or Linux PC, or an Apple Mac computer to format the microSD card and download the initial setup software for your Raspberry Pi. It doesn't matter what operating system this computer runs, because it's just for copying the files across.
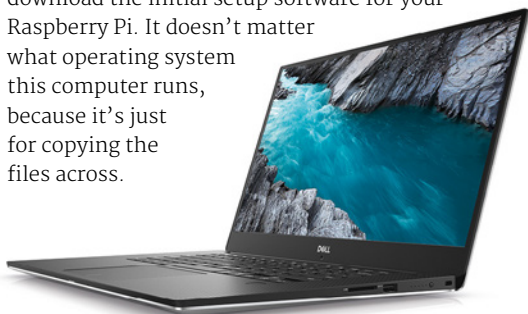
USB keyboard

Like any computer, you need a means to enter web addresses, type commands, and otherwise control Raspberry Pi. You can use a Bluetooth keyboard, but the initial setup process is much easier with a wired keyboard. Raspberry Pi sells an official Keyboard and Hub (**magpi.cc/keyboard**).
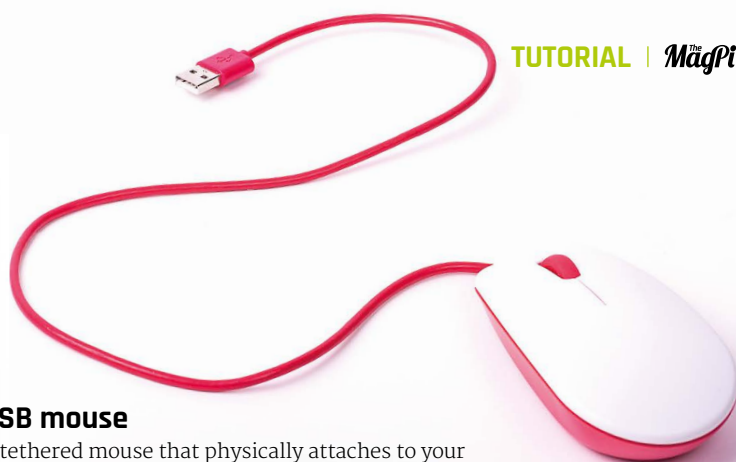
USB mouse

A tethered mouse that physically attaches to your Raspberry Pi via a USB port is simplest and, unlike a Bluetooth version, is less likely to get lost just when you need it. Like the keyboard, we think it's best to perform the setup with a wired mouse. Raspberry Pi sells an Official Mouse (**magpi.cc/mouse**).

Power supply

Raspberry Pi uses the same type of USB power connection as your average smartphone. So you can recycle an old USB to micro USB cable (or USB Type-C for Raspberry Pi 4) and a smartphone power supply. Raspberry Pi also sells official power supplies (**magpi.cc/products**), which provide a reliable source of power.

Display and HDMI cable

A standard PC monitor is ideal, as the screen will be large enough to read comfortably. It needs to have an HDMI connection, as that's what's fitted on your Raspberry Pi board. Raspberry Pi 3B+ and 3A+ both use regular HDMI cables. Raspberry Pi 4 can power two HDMI displays, but requires a less common micro-HDMI to HDMI cable (or adapter); Raspberry Pi Zero W needs a mini HDMI to HDMI cable (or adapter).

USB hub

Instead of standard-size USB ports, Raspberry Pi Zero has a micro USB port (and usually comes with a micro USB to USB adapter). To attach a keyboard and mouse (and other items) to a Raspberry Pi Zero W or 3A+, you should get a four-port USB hub (or use a keyboard with a hub built in).

Set up
Raspberry Pi

Raspberry Pi 4 / 3B+ / 3 has plenty of connections, making it easy to set up

01 Hook up the keyboard

Connect a regular wired PC (or Mac) keyboard to one of the four larger USB A sockets on a Raspberry Pi 4 / 3B+/ 3. It doesn't matter which USB A socket you connect it to. It is possible to connect a Bluetooth keyboard, but it's much better to use a wired keyboard to start with.

02 Connect a mouse

Connect a USB wired mouse to one of the other larger USB A sockets on Raspberry Pi. As with the keyboard, it is possible to use a Bluetooth wireless mouse, but setup is much easier with a wired connection.

03 HDMI cable

Next, connect Raspberry Pi to your display using an HDMI cable. This will connect to one of the micro-HDMI sockets on the side of a Raspberry Pi 4, or full-size HDMI socket on a Raspberry Pi 3/3B+. Connect the other end of the HDMI cable to an HDMI monitor or television.

A HDMI cable, such as ones used by most modern televisions, is used to connect Raspberry Pi to a TV or display. You'll need a micro-HDMI to HDMI cable (or two) to set up a Raspberry Pi 4. Or a regular HDMI cable for Raspberry Pi 3B+ / 3 (or older) models.
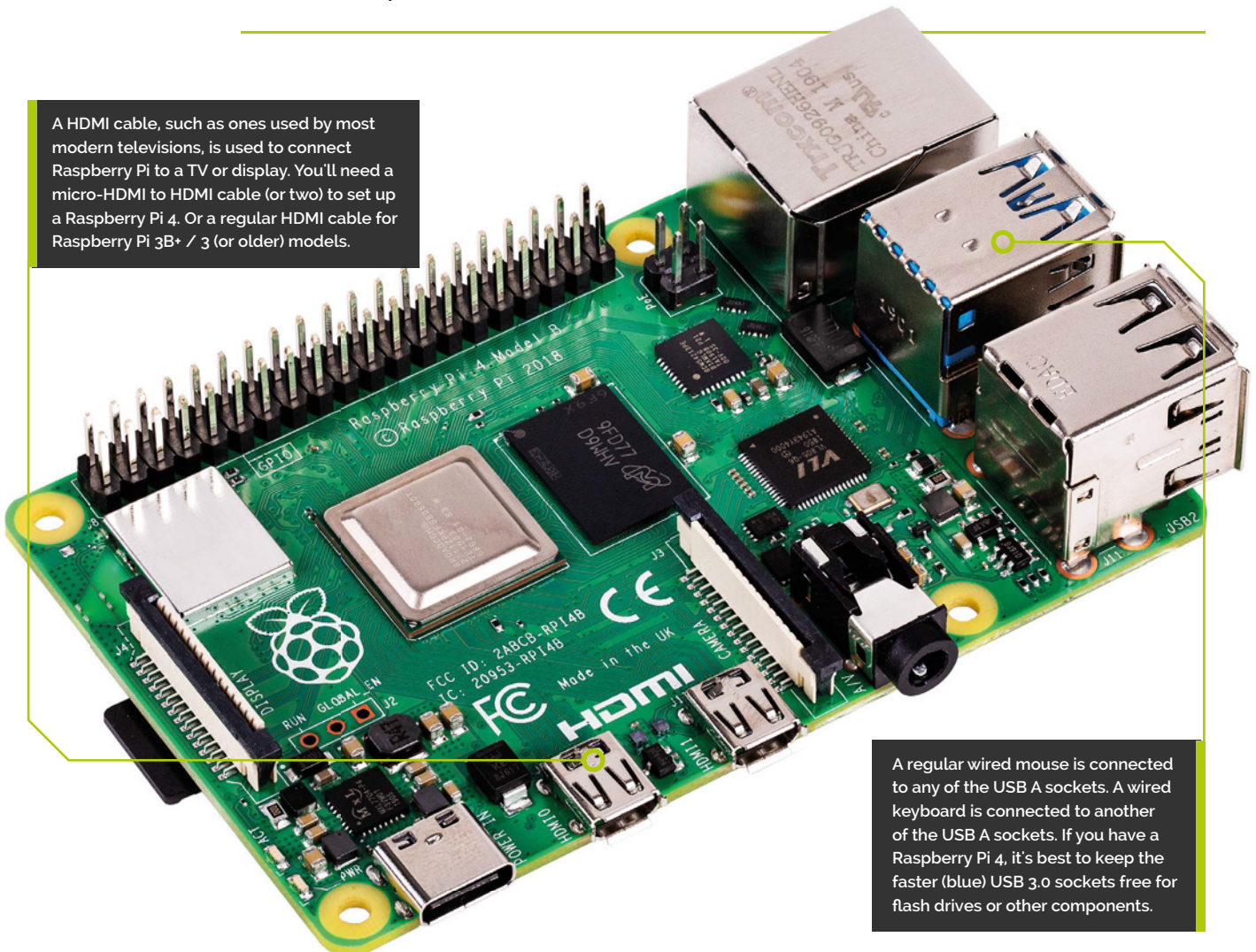
A regular wired mouse is connected to any of the USB A sockets. A wired keyboard is connected to another of the USB A sockets. If you have a Raspberry Pi 4, it's best to keep the faster (blue) USB 3.0 sockets free for flash drives or other components.

You'll need this micro USB to USB A adapter to connect wired USB devices such as a mouse and keyboard to your Raspberry Pi Zero W

Raspberry Pi Zero W features a mini-HDMI socket. You'll need a mini-HDMI to full-sized HDMI adapter like this to connect your Raspberry Pi Zero W to an HDMI display
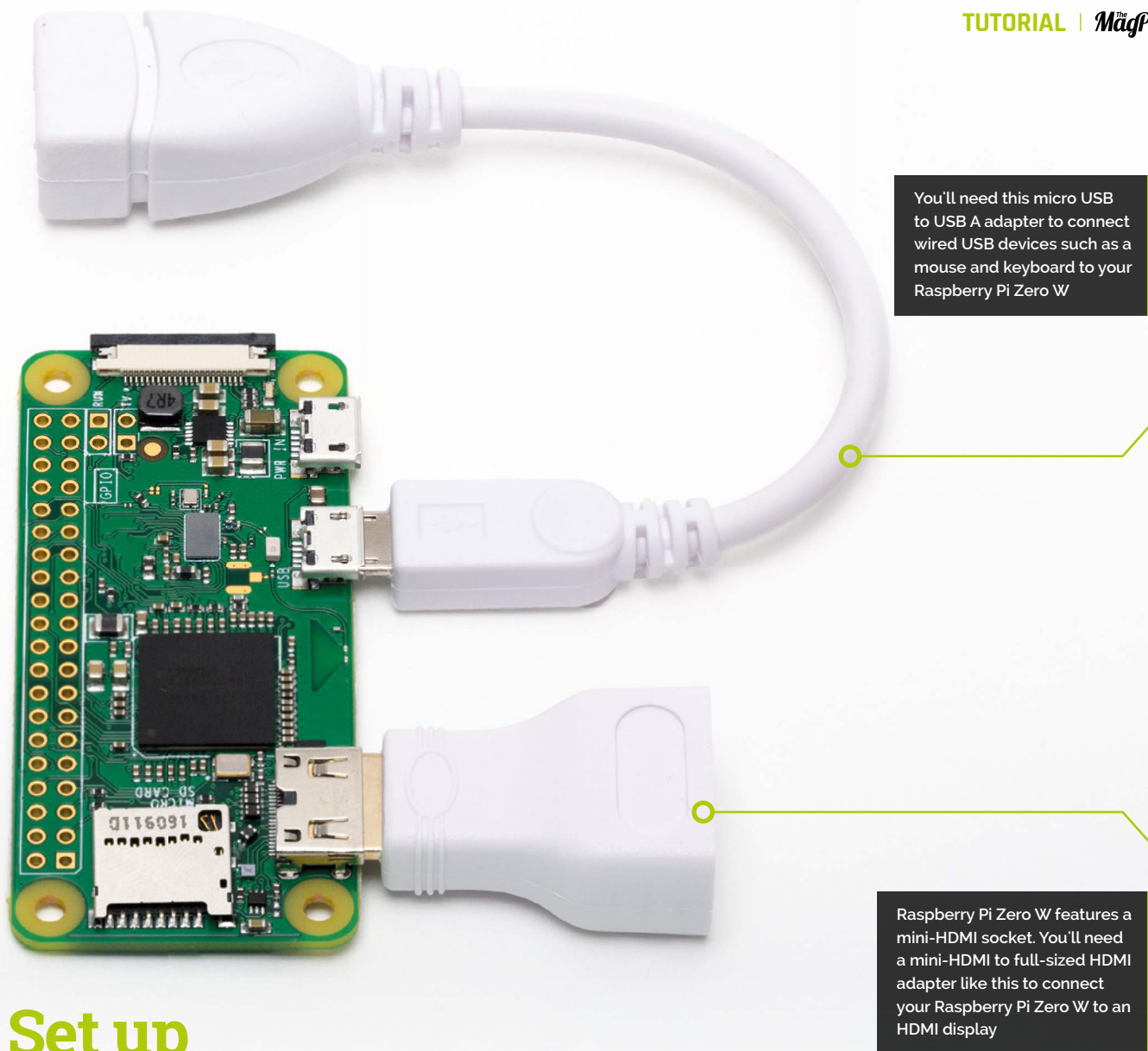
Set up
Raspberry
Pi Zero

You'll need a couple of adapters to set up a Raspberry Pi Zero / W / WH

01 Get it connected

If you're setting up a smaller Raspberry Pi Zero, you'll need to use a micro USB to USB A adapter cable to connect the keyboard to the smaller connection on a Raspberry Pi Zero W. The latter model has only a single micro USB port for connecting devices, which makes connecting both a mouse and keyboard slightly trickier than when using a larger Raspberry Pi.

02 Mouse and keyboard

You can either connect your mouse to a USB socket on your keyboard (if one is available), then connect the keyboard to the micro USB socket (via the micro USB to USB A adapter). Or, you can attach a USB hub to the micro USB to USB A adapter.

03 More connections

Now connect your full-sized HDMI cable to the mini-HDMI to HDMI adapter, and plug the adapter into the mini-HDMI port in the middle of your Raspberry Pi Zero W. Connect the other end of the HDMI cable to an HDMI monitor or television.

Set up
the software

Use NOOBS to install Raspbian OS on your microSD card and start your Raspberry Pi

Now you've got all the pieces together, it's time to install an operating system on your Raspberry Pi, so you can start using it.
Raspbian is the official OS for Raspberry Pi, and the easiest way to set up Raspbian on your Raspberry Pi is to use NOOBS (New Out Of Box Software).

If you bought a NOOBS pre-installed 16GB microSD card (**magpi.cc/huLdtN**), you can skip Steps 1 to 3. Otherwise, you'll need to format a microSD card and copy the NOOBS software to it.

02 **Format the microSD**
Choose the Quick Format option and then click Format (if using a Mac, you'll need to enter your admin password at this point). When the card has completed the formatting process, it's ready for use in your Raspberry Pi. Leave the microSD card in your computer for now and simply note the location of your duly formatted SD card. Windows will often assign it a hard drive letter, such as E; on a Mac it will appear in the Devices part of a Finder window.

You'll Need

- A Windows/Linux PC or Apple Mac computer

- A microSD card (8GB or larger)

- A microSD to USB adapter (or a microSD to SD adapter and SD card slot on your computer)

- SD Memory Card Formatter **rpf.io/sdcard**

- NOOBS **rpf.io/downloads**

01 **Prepare to format**
Start by downloading SD Card Formatter tool from the SD Card Association website (**rpf.io/sdcard**). Now attach the microSD card to your PC or Mac computer and launch SD Card Formatter (click Yes to allow Windows to run it). If the card isn't automatically recognised, remove and reattach it and click Refresh. The card should be selected automatically (or choose the right one from the list).

03 **Download NOOBS**
Download the NOOBS software from **rpf.io/downloads**. NOOBS (New Out Of Box System) provides a choice of Raspberry Pi operating systems and installs them for you. Click 'Download zip' and save the file to your Downloads folder. When the zip file download is complete, double-click to launch and uncompress the folder. You'll need to copy all the files from the NOOBS folder to your SD card. Press **CTRL+A** (⌘**+A** on a Mac) to select all the files, then drag all the files to the SD card folder. Once they've copied across, eject your SD card. Be careful to copy the *files inside* the NOOBS folder to the microSD card (not the NOOBS folder itself).

First, insert your microSD card into Raspberry Pi

With the microSD card fully inserted, connect your power supply cable to Raspberry Pi. A red light will appear on the board to indicate the presence of power

04 Assemble your Raspberry Pi

Now it's time to physically set up your Raspberry Pi. Plug your PC monitor into the mains and attach its HDMI cable to the corresponding HDMI port on your Raspberry Pi. Plug in the power supply but don't attach it to Raspberry Pi just yet. Use two USB ports to attach keyboard and mouse. Finally, remove the microSD card from the SD card adapter and slot it into the underside of your Raspberry Pi 4, 3B+, or 3. Raspberry Pi Zero W owners will need to attach a USB hub to connect mouse, keyboard, and monitor; the microSD card slot is on the top of its circuit board.
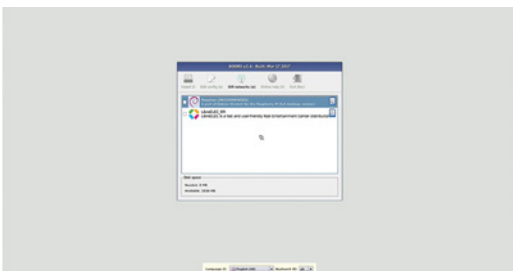
05 Power up

Plug in your Raspberry Pi power supply and, after a few seconds, the screen should come on. When the NOOBS installer appears, you'll see a choice of operating systems. We're going to install Raspbian, the first and most popular one. Tick this option and click Install, then click Yes to confirm. For more OS options, instead click 'Wifi networks' and enter your wireless password; more OS choices will appear. Installation takes its time but will complete – eventually. After this, a message confirming the success installation appears. Your Raspberry Pi will prompt you to click OK, after which it will reboot and load the Raspbian OS.
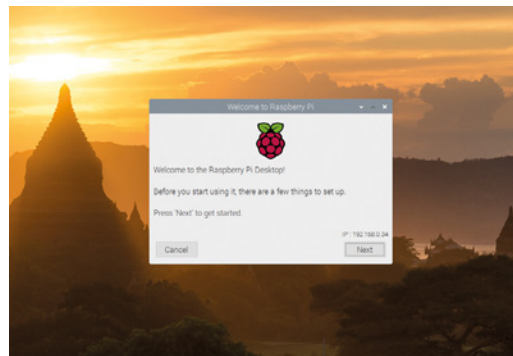
06 Get online

When Raspbian loads for the first time, you need to set a few preferences. Click Next, when prompted, then select your time zone and preferred language and create a login password. You're now ready to get online. Choose your WiFi network and type any required password. Once connected, click Next to allow Raspbian to check for any OS updates. When it's done so, it may ask to reboot so the updates can be applied.

Click the Raspberry icon at the top-left of the screen to access items such as programming IDEs, a web browser, media player, image viewer, games, and accessories such as a calculator, file manager, and text editor. You're all set to start enjoying your very own Raspberry Pi. M

Get started with
PICO-8 and Raspberry Pi

Turn your Raspberry Pi into the coolest console around
with PICO-8! Play, make, and share 8-bit games

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

At the forefront of a new wave of 'virtual' consoles, PICO-8 packs a pixel-perfect punch of 8-bit delights. Containing everything you need to make your own games, from code to graphics to sound, it has a super-active and supportive community of makers and creators behind it. With its own built-in game browser, you'll have access to a huge library of games straight out of the gate. The good news is you can get it running on your Raspberry Pi in just a few steps, meaning you'll be sitting cross-legged in front of the living room TV in no time.

You'll Need

- USB game controller(s) (Any generic 2-button controller will do)
- PICO-8 **magpi.cc/pico8**
- Keyboard, mouse, and WiFi. (You probably already have these!)
- Raspberry Pi 3 or Raspberry Pi Zero (or earlier)

▶ Cutesy pixel art, punchy music, and juicy game feel are all present in Just One Boss. Play this first, but beware of the brutal hard mode!

01 Prep a Raspberry Pi

First things first, you'll want a Raspberry Pi set up and ready to go. These steps are tested on a fresh install of Raspbian Stretch with Desktop, so you'll want to do the same. You can plug in your USB controller (PICO-8 supports one or two) and,
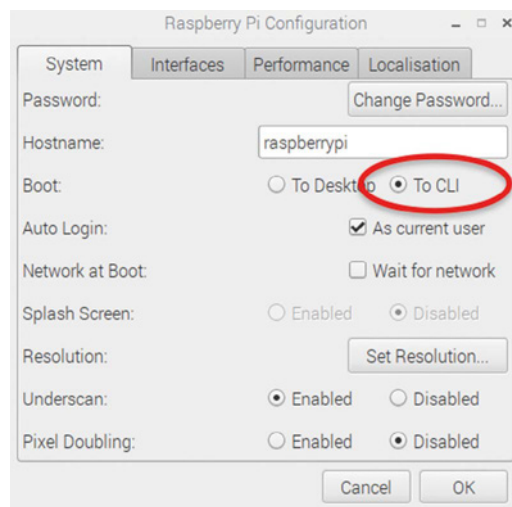
▲ To make Raspberry Pi boot straight into PICO-8, you'll need to set system preferences to boot into CLI after editing .bashrc

for these first few steps, you'll need to plug in a USB keyboard and mouse as well. Once you've got it all up and running, though, a controller and WiFi is all you'll need to download and play games.

02 The Lexaloffle shuffle

Next, you will want to download a copy of PICO-8 from the Lexaloffle website (**magpi.cc/PICO8**). To do this, you'll need to set up an account and buy a licence, which is a steal at around £12 at the time of writing. Once you've done that, go to the Downloads section of your new account on the website and download the Raspberry Pi version of PICO-8. You can use Chromium to download the zip file straight to your Raspberry Pi. Note: with your newly acquired licence, you can download Windows, macOS, and Linux builds of PICO-8 as well. A nice perk!

Retro controllers
for a retro console

PICO-8 is a virtual
console with an old heart,
and hundreds of games!

03 3, 2, 1... launch!

Extract the contents of the zip file into your **/home/pi** directory. Navigate to the newly unzipped folder and you will see a bunch of files. Most interesting for us is **pico8** (not the .dat), which is the virtual console itself, and **manual.txt**. The manual contains loads of valuable information, tips, and commands to get the most out of the console, and will help loads when you start to make your own games. But as we're focusing on playing them in this tutorial, ignore that for now and launch PICO-8 by double-clicking the **pico8** file.

04 Time to SPLORE

Beepidy beep! Wasn't that cute? All being well, you should be looking at the PICO-8 command prompt, complete with flashing cursor, begging for you to type something in. Type in **HELP** to see a list of commands, many of which might be familiar to you if you've used Terminal commands on Raspberry Pi before. Now that's all well and

> ❝ You should be looking at the PICO-8 command prompt, with flashing cursor, begging for you to type something in ❞

good, you say, but where are the games? Thankfully, PICO-8 has its own built-in game browser called SPLORE. Type in **SPLORE** and hit **ENTER**.

05 Taking control

Plug in a USB controller if you haven't already, and let's take it for a spin. PICO-8 refers to its games as 'carts', as in cartridges (remember those?). Use the D-pad to navigate between the tabs. You'll see a section for new carts, a search function, a tab for your favourite carts, and a file browser for your own creations. Let's scroll to the Featured tab, select Update, and push a button. PICO-8 should now connect to the game library and present you with a list. Scroll down to find one that piques your interest and push any button to start.

Top Tip 👍

Controller-only
searching

Want to search
for carts but don't
want to use a
keyboard? Use
your controller
D-pad to
scroll through
characters.
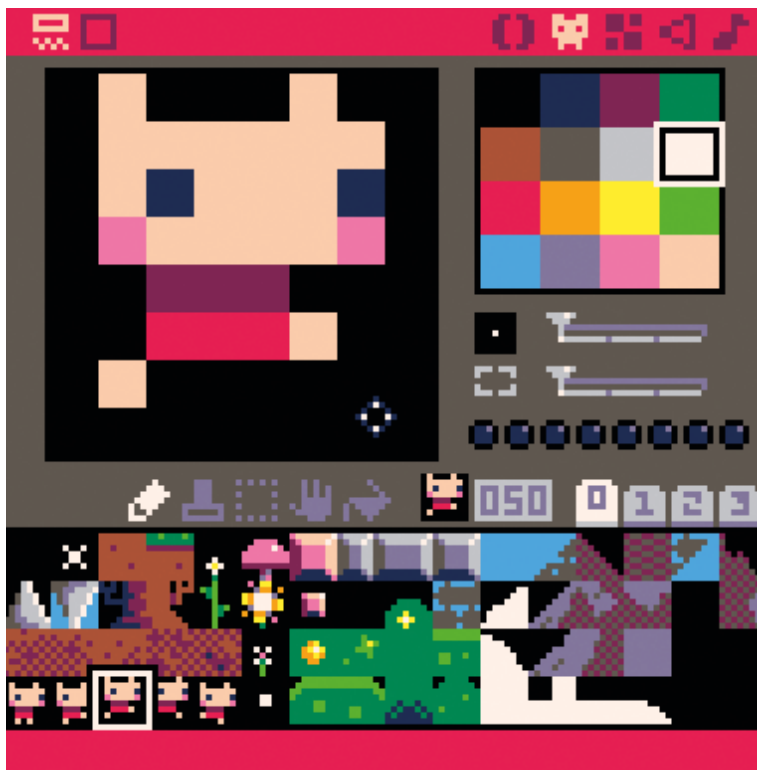
▲ PICO-8 contains everything you need to make your own games, like this nifty 8-bit sprite editor

▲ You can plug in one or two USB controllers to play. How about some retro-style ones?

06 A veritable cornucopia

Congratulations! You now have access to hundreds of homebrew wonders. You'll find everything from straight-out action arcade, to quirky indie titles, to demo-scene experiments, to de-makes of popular titles (where famous games are remade in 8-bit, just because). If you want to try something else, hit the Start button to open the menu, where you can return to SPLORE and jump into another title. You can also use this menu to 'favourite' carts, so that you can find them more easily.

07 My personal pick

You're probably feeling a little overwhelmed at the number of carts out there and are wondering where to start. Well, here are handful of your author's favourites. Enigmatic boss battler Just One Boss by Ayla Myers (**@bridgs_dev**), chunky squirmy puzzler Tomato Worm by Jose Guerra (**@guerragames**), mysterious puzzle platformer Dusk Child by Sophie Houlden (**@SophieH**), tear-jerking space-'em-up Orbiter Suite by Andrew Anderson (**@kittenm4ster**) and aquatic action game

SWORDFISH by yours truly (**@danhowardgames**). That should be enough to start with, but new ones are released almost daily.

08 Get dedicated

Now it's working and you've tried a few games, let's turn your Raspberry Pi into a dedicated PICO-8 console that launches straight into the cart browser when it boots. There are lots of ways to achieve this but the quickest, though possibly not the most elegant, solution is to edit **.bashrc**. This file contains a load of commands which run every time a new interactive shell is launched; for example, when opening a new Terminal window. Exit out of PICO-8, open a new Terminal, and enter `sudo nano /home/pi/.bashrc` to edit it.

09 Cart boot sale

Add `sudo /home/pi/pico-8/pico8 -splore` to the last line. This command will launch PICO-8 straight into the cart browser. Save the script and, on the desktop, go to the Raspberry Pi menu in the top left, then to Preferences, then Raspberry

Pi Configuration. In the System tab you'll need to change 'Boot:' to 'To CLI'. Now on boot, we'll go straight into PICO-8, no need for a keyboard or mouse. Save your changes and reboot to see your new favourite console in all its majesty.

> You'll now be staring at the code editor, and at the very code that runs the game you loaded

10 Behind the scenes

Great, now you have a dedicated PICO-8 console, well done, but what's next? Well, as mentioned, you can use PICO-8 to easily make your own games. We'll go into more detail about that in the next tutorial, but for now if you are curious, why not take a look under the hood? Plug in a keyboard and mouse, launch PICO-8 and load up a game, then return back to SPLORE. Next, hit **ESC** to exit to the PICO-8 console; once there, hit **ESC** again to swap to the code screen.

11 The tools at your disposal

You'll now be staring at the code editor, and at the very code that runs the game you

loaded. At the top right of the screen you'll also see icons that allow you to navigate between the other editors. From left to right, there are editors for code, sprites, maps, SFX, and music – all you could ever need! Have a click through to see what they look like. We'll go into detail on these in later tutorials, but for now why not try editing some of the sprites, then hit **CTRL+R** to reload the game with your changes?

12 Join the community

One of the most unique and exciting aspects of PICO-8 is the enthusiastic community of makers, tinkerers, and developers that has built up around it. People are very passionate about the console and are always willing to lend a hand and offer help. We highly recommend getting involved. You can find them on the official Lexaloffle forum (**magpi.cc/hqxiAL**), the PICO-8 Discord server, or by searching **#pico8** on Twitter. There are plenty of resources out there if you simply can't wait to get started making your own games. Happy making!

Top Tip

Two players, one keyboard

Don't have a controller? Don't worry! Play with a keyboard. P1 uses cursors + **N/M**, and P2 uses **SDFE** + **TAB/Q**.

▲ Featuring realistic gravity and inertia, Spaceman 8 sees a jet-pack-equipped astronaut transporting gems
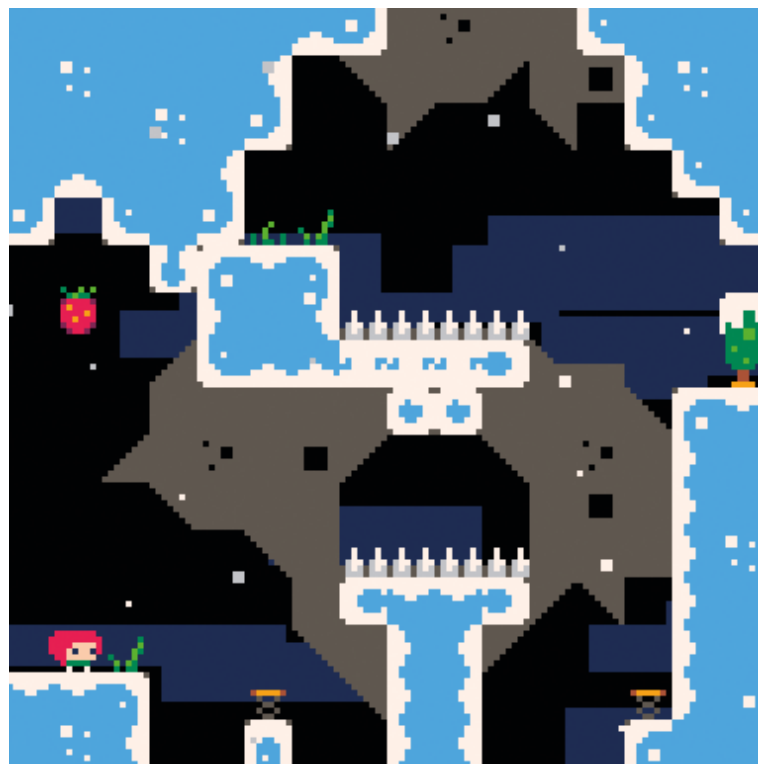
▲ Celeste, indie game of the year 2018, actually started life on PICO-8

Make a self-healing
Raspberry Pi

Get back to a fresh install with no need to wipe the microSD
card, download images, or use another computer

MAKER

**PJ
Evans**

PJ is a writer,
software engineer
and runs the Milton
Keynes Raspberry
Jam, which
involves a lot of
microSD cards.

@mrpjevans

Most modern operating systems come with a 'recovery partition', a reserved area of the drive containing everything needed to get the machine back to a clean install. So, if something goes badly wrong, you can start over. In the world of Raspbian, this normally means overwriting the image on the microSD card. This is perfectly fine, but what if you have a large number to do, say a classroom's worth, or you don't have access to another device to do the burning? We're going to create an alternative version of Raspbian featuring a recovery partition. Raspberry Pi, heal thyself!

01 Prepare your workspace

This tutorial will describe how to create a bootable image featuring a restore partition, but there's also a script to automate the process that you can download from **magpi.cc/junkPr**. This also contains the code shown some of the trickier commands shown later.

Make sure you have uuidgen installed by running it from the command line. If not, run:

You'll Need

> Raspbian Stretch
  Full image
  magpi.cc/raspbian

> Raspbian Stretch
  Lite image
  magpi.cc/raspbian

> Jumper (or F-F
  jumper cable)

> Lots of hard disk
  space

> Debian/Raspbian OS

▲ Create a custom image for a classroom and know you can
quickly get back to a clean, working machine after each session

```
sudo apt install uuid-runtime
```

Most commands here will need to be run as root. To avoid having to enter 'sudo' every time, you can switch to root using:

```
sudo su
```

Create a working directory on your machine and make sure you've downloaded both the Raspbian Full and Raspbian Lite images (we're using 2019-04-08). Unzip them as follows:

```
unzip 2019-04-08-raspbian-stretch-full.zip
unzip 2019-04-08-raspbian-stretch-lite.zip
```

02 Calculate the image size

Our image needs to be big enough for Raspbian Full, including its boot partition, and a second partition containing Raspbian Lite with an image of Raspbian Full. We measure disk sizes in sectors, each one 512 bytes in size. Find out how many sectors are used by the partitions:

```
fdisk -lu 2019-04-08-raspbian-stretch-full.img
fdisk -lu 2019-04-08-raspbian-stretch-lite.img
```

Each command produces output detailing how many sectors are required (see **Figure 1**, overleaf). The boot partition starts at sector 8192 to allow for the file allocation table. To calculate the size needed:

8192 + Raspbian Full Boot Partition + Raspbian Lite Main Partition + (Raspbian Full Main Partition × 2)

With these Raspbian versions, you will need 24,426,283 sectors.

This project complements the keyring project from last month, but works with any Raspberry Pi

Can't get access your Pi? A boot script detects this short and starts restoration

03 Create the blank image

We now need to create an empty file to contain our disk image. First convert the number of sectors required into 4MB blocks like so:

24,426,283 × 512 bytes = 12,506,256,896 bytes

12,506,256,896 / 4,194,304 = 2,982 4-megabyte blocks (rounded up)

Now create your target image:

```
dd if=/dev/zero bs=4M count=2982 status=progress
> 2019-04-08-raspbian-stretch-full.restore.img
```

You now have a large file full of zeroes.

04 Partition the image

Let's turn our blank file into a disk image. Start by generating some unique identifiers for the partitions:

```
UUID_RESTORE=$(uuidgen)
UUID_ROOTFS=$(uuidgen)
PARTUUID=$(tr -dc 'a-f0-9' < /dev/urandom
2>/dev/null | head -c8)
```

Now create the partition table:

```
sfdisk 2019-04-08-raspbian-stretch-full.
restore.img <<EOL
```

> ❝ This tutorial will describe how to create a bootable image featuring a restore partition ❞

```
label: dos
label-id: 0x${PARTUUID}
unit: sectors

2019-04-08-raspbian-stretch-full.restore.
img1 : start=8192, size=87851, type=c
2019-04-08-raspbian-stretch-full.restore.
img2 : start=96043, size=13877248, type=83
2019-04-08-raspbian-stretch-full.restore.
img3 : start=13973291, size=10452992, type=83
EOL
```

Careful! The sizes used here are specific to the version of Raspbian used. Other versions will have different sizes. Use fdisk to calculate them.

05 Mount the images

Our file of zeroes can now be accessed as a disk. We'll use the 'loopback' system so we can access it, along with the two versions of Raspbian.

```
losetup -v -f 2019-04-08-raspbian-stretch-
full.restore.img
partx -v --add /dev/loop0
```

```
    losetup --show -f -P 2019-04-08-raspbian-
stretch-lite.img
    losetup --show -f -P 2019-04-08-raspbian-
stretch-full.img
```

Now copy over the boot and root partitions from our Raspbian Full image to partitions one and three of the new image:

```
  dd if=/dev/loop2p1 of=/dev/loop0p1
status=progress bs=4M
  dd if=/dev/loop2p2 of=/dev/loop0p3
status=progress bs=4M
```

We can now install Raspbian Lite on the second partition:

```
  dd if=/dev/loop1p2 of=/dev/loop0p2
status=progress bs=4M
```

06 Configure and mount partitions

First, assign new unique IDs to the partitions and rename the recovery partition so we can tell them apart.

```
  tune2fs /dev/loop0p2 -U ${UUID_RESTORE}
  e2label /dev/loop0p2 recoveryfs
  tune2fs /dev/loop0p3 -U ${UUID_ROOTFS}
```

Although we have allocated enough space for the recovery partition to hold a copy of the Raspbian

❝ Each of the main partitions has an fstab file which tells Raspbian what disks to mount and where ❞

Full image, copying over the Lite image has reduced it. Luckily it's easy to fix this:

```
  e2fsck -f /dev/loop0p2
  resize2fs /dev/loop0p2
```

Now we're in a position to mount the new image's file systems:

```
  mkdir -p mnt/restore_boot
  mkdir -p mnt/restore_recovery
  mkdir -p mnt/restore_rootfs

  mount /dev/loop0p1 mnt/restore_boot
  mount /dev/loop0p2 mnt/restore_recovery
  mount /dev/loop0p3 mnt/restore_rootfs
```

07 Set the boot partition

Currently our image would not boot as it doesn't know which partition to use. Run this command and make a note of the eight characters after 'Disk identifier: 0x'.


```
pi@vb-rasbian:~/build_restore_files $ fdisk -lu 2019-04-08-raspbian-stretch-full.img
Disk 2019-04-08-raspbian-stretch-full.img: 5 GiB, 5402263552 bytes, 10551296 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x43c1e3bb

Device                                  Boot Start      End  Sectors  Size Id Type
2019-04-08-raspbian-stretch-full.img1        8192    96042    87851 42.9M  c W95 FAT32 (LBA)
2019-04-08-raspbian-stretch-full.img2       98304 10551295 10452992    5G 83 Linux
pi@vb-rasbian:~/build_restore_files $ fdisk -lu 2019-04-08-raspbian-stretch-lite.img
Disk 2019-04-08-raspbian-stretch-lite.img: 1.7 GiB, 1803550720 bytes, 3522560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc1dc39e5

Device                                  Boot Start      End Sectors  Size Id Type
2019-04-08-raspbian-stretch-lite.img1        8192    96042   87851 42.9M  c W95 FAT32 (LBA)
2019-04-08-raspbian-stretch-lite.img2       98304 3522559 3424256  1.6G 83 Linux
pi@vb-rasbian:~/build_restore_files $
```

▶ **Figure 1** Use fdisk to view the partitions and calculate the sizes you need
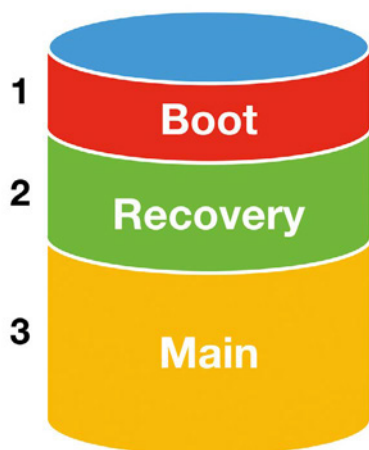
▲ Our recovery partition sits between the boot partition and main file system

```
fdisk -lu 2019-04-08-raspbian-stretch-full.restore.img
```

Then edit the **cmdline.txt** file to reset it:

```
nano mnt/restore_boot/cmdline.txt
```

Change the eight-character code after `PARTUUID=` to the value you noted and change the following `-02` to `-03`, telling Raspbian to boot to the third partition.

08 Create the reset scripts

To restore Raspbian your Raspberry Pi needs to boot to the second partition, containing Raspbian Lite, then overwrite the third partition with a snapshot image. We can automate this using scripts. Create the three scripts here in the **mnt/restore_boot** directory and make them executable:

```
chmod +x mnt/restore_boot/boot_to_root
chmod +x mnt/restore_boot/boot_to_recovery
chmod +x mnt/restore_boot/restore_root
```

Now make **restore_root** run at boot time on the recovery partition:

```
nano mnt/restore_recovery/etc/rc.local
```

Before the `exit 0` line, add:

```
/boot/restore_root
```

09 Fix fstab

Each of the main partitions has an **fstab** file which tells Raspbian what disks to mount and where. This needs correcting to match our new layout:

boot_to_recovery

> Language: **Bash**

```
001.  #!/bin/bash
002.  if [ "$EUID" -ne 0 ]
003.    then echo "Please run as root"
004.    exit
005.  fi
006.
007.  echo Rebooting to recovery partition in 5 seconds
008.
009.  if [ "$1" = "restore" ]; then
010.     echo Automatic restore selected
011.     touch /boot/restore
012.  fi
013.
014.  sleep 5
015.  sed -i 's/-03/-02/' /boot/cmdline.txt
016.  touch /boot/ssh
017.  reboot
018.
019.  exit 0
```

boot_to_rootfs

> Language: **Bash**

```
001.  #!/bin/bash
002.  if [ "$EUID" -ne 0 ]
003.    then echo "Please run as root"
004.    exit
005.  fi
006.
007.  echo Rebooting to root partition in 5 seconds
008.  sleep 5
009.  sed -i 's/-02/-03/' /boot/cmdline.txt
010.  touch /boot/ssh
011.  reboot
012.  exit 0
```

restore_root

> Language: **Bash**

```
001.  #!/bin/bash
002.  if [ -f "/boot/restore" ]; then
003.        echo Restoring rootfs
004.        dd if=/rootfs.img of=/dev/mmcblk0p3 conv=fsync
      status=progress bs=4M
005.        unlink /boot/restore
006.        /boot/boot_to_root
007.  fi
008.  exit 0
```

check_restore_gpio.py

> Language: **Python 3**

```
001.  import os
002.  from gpiozero import Button
003.
004.  button = Button(21)
005.
006.  if button.is_pressed:
007.      print("Restore button is pressed")
008.      os.system("/boot/boot_to_recovery restore")
009.  else:
010.      print("Restore button is not pressed")
```

▲ This simple script will trigger a restore when the GPIO 21 pin is connected to GND

```
UUID_BOOT=$(blkid -o export /dev/loop0p1 |
egrep '^UUID=' | cut -d'=' -f2)

cat << EOF > mnt/restore_rootfs/etc/fstab
 proc                   /proc  proc
defaults        0       0
 UUID=${UUID_BOOT} /boot  vfat    defaults
0       2
 UUID=${UUID_ROOTFS} /      ext4
defaults,noatime  0      1
 EOF

cat << EOF > mnt/restore_recovery/etc/fstab
 proc                   /proc  proc
defaults        0       0
 UUID=${UUID_BOOT}      /boot  vfat
defaults        0       2
 UUID=${UUID_RESTORE}   /      ext4
defaults,noatime  0      1
 EOF
```

10 **Take a snapshot**

As Raspbian Full has never been booted, it's a perfect time to make a copy of it for restoration. This command makes a copy of the main partition and saves it in the recovery partition as a file.

```
dd if=/dev/loop0p3 of=mnt/restore_recovery/
rootfs.img status=progress bs=4M
```

Now unmount everything:

```
umount -f mnt/restore_boot
umount -f mnt/restore_recovery
umount -f mnt/restore_rootfs
losetup --detach-all
```


▲ This project makes an ideal companion to last month's Keyring Pi – a Zero-to-go that can self-heal!

11 **Burn and test**

You should now have a complete and ready-to-go image. Copy it over to a suitably sized microSD card (Raspbian-specific example):

```
dd bs=4M if=2019-04-08-raspbian-stretch-
full.restore.img of=/dev/sda conv=fsync
status=progress
```

…or you can use any burning tool, such as Etcher. Your SD card should now boot as normal. To test partition swapping, open up a Terminal and type:

```
sudo ./boot/boot_to_recovery
```

The Raspberry Pi should reboot into Raspbian Lite. To go back:

```
sudo ./boot/boot_to_rootfs
```

To perform a fully automatic restore:

```
sudo ./boot/boot_to_recovery restore
```

12 **Physical reset**

What if you can't get terminal access to your Raspberry Pi? A Python script that runs on boot can check the state of a GPIO pin; if shorted, the restore process is triggered. Enter the **check_reset_gpio.py** code in **/boot** and make sure it runs on boot:

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

Before the `exit 0` line, add:

```
python3 /boot/check_reset_gpio.py
```

To trigger the restore, use a jumper wire between GND and GPIO 21 pins and boot your Raspberry Pi. M

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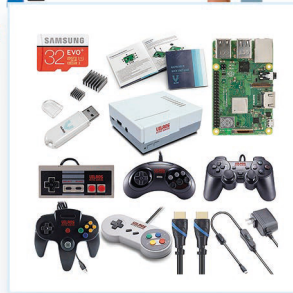
Vilros Raspberry Pi 3 Model B+
Retro Arcade Gaming Kit With
Retro Gaming Controller Set
~~$114.99~~ **$109.99**

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"

Part 01

The Squeeze:
A new way to play

Is it a joystick? Is it a paddle? No it's 'The Squeeze' – a new type of games controller

MAKER

Mike Cook

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

magpi.cc/TPaUfT

Have you seen those dynamo torches, the ones that you repeatedly squeeze to light three white LEDs? Well, this month we are going to take a pair of these and turn them into a unique games controller. A new type of controller offers the possibility of new types of games, or a better way to control some existing types of games.

a dynamo torch, the user repeatedly squeezes a lever to spin a magnet in a coil and generate electricity. We took one apart and measured the voltage the generator produced. As you can see in **Figure 1**, the output is AC with a peak-to-peak voltage of almost 80 V; when squeezed, the frequency rapidly rises to about 170Hz.

01 The torch

The torch, or flashlight for our American cousins, can come in many forms. These days there are lots of self-powering devices which involve actually generating the power needed to drive them by the efforts of the user. With

02 The dynamic signal

This voltage is very high, but is loaded down by putting a white LED across it; this shorts the negative voltage and limits the positive voltage to about 3 V, which is the forward voltage drop across the LED. This is a cheap and nasty design. You can

⚠️

Warning!
High voltage

The dynamo torches in this project can produce high voltage, so be careful.

You'll Need

> Two dynamo torches
magpi.cc/HkHuJM

> A/D converter (ADC), e.g. MCP3008

> Assorted electronic components

Signal-conditioning circuits and A/D converter
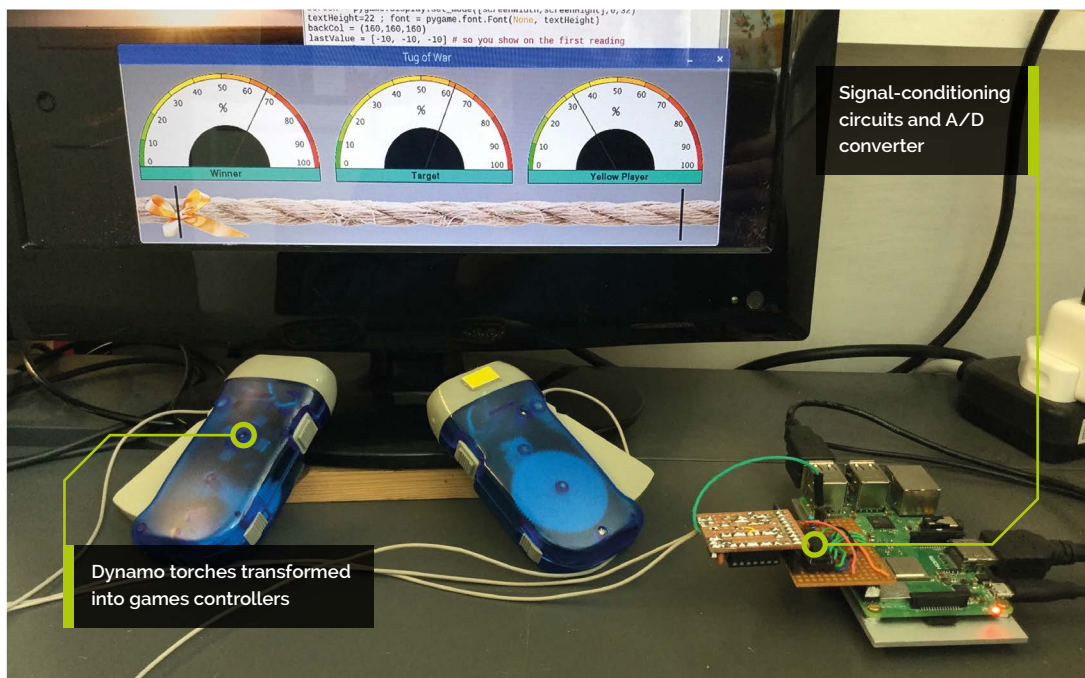
Dynamo torches transformed into games controllers
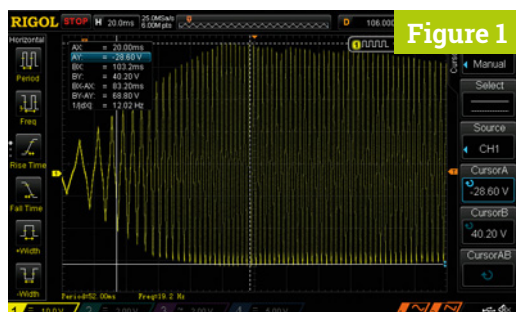
▲ **Figure 1** Oscilloscope trace of the raw output from the dynamo torch

pump the leaver to sustain a voltage or just squeeze once for a pulse, as shown in **Figure 2**. This shows 24 rapid squeezes followed by a single squeeze and release; it is measured over five seconds. Note how the trace is changing so rapidly that we can't see the individual waveform, only the envelope.

> 〞 The idea is that we can condition the electrical signal from the dynamo torch to make a games controller 〞

03 **Conditioning the signal**

The idea is that we can condition this signal to make a games controller. Basically, we need to make it into a DC signal by adding a series diode and then getting just the peaks of this signal with an envelope follower, which is sometimes called a peak detector. This uses a capacitor to hold the peak voltage and a discharging resistor which
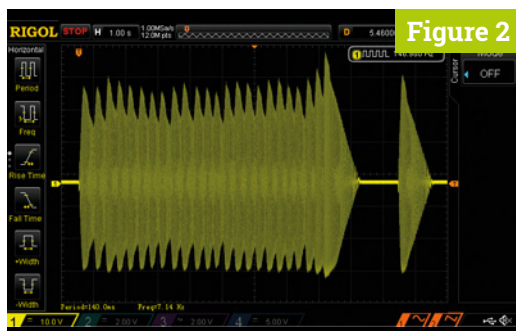
▲ **Figure 2** Oscilloscope trace of 24 rapid squeezes and then a single squeeze at the end

TugOfWar.py

> Language: **Python 3**

```python
001.  #!/usr/bin/env python3
002.  #Tug of war using squeeze controller
003.  # By Mike Cook June 2019
004.
005.  import math, spidev, time
006.  import os, pygame, sys, random
007.
008.  pygame.init()
009.  pygame.mixer.quit()
010.  pygame.mixer.init(frequency=22050, size=-16, channels=2,
       buffer=512)
011.  os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
012.  pygame.display.set_caption("Tug of War")
013.  pygame.event.set_allowed(None)
014.  pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
015.  screenWidth = 960 ; screenHight = 280 ; cp = screenWidth // 2
016.  screen = pygame.display.set_mode([screenWidth,screenHight],0,32)
017.  textHeight=22 ; font = pygame.font.Font(None, textHeight)
018.  backCol = (160,160,160)
019.  lastValue = [-10, -10, -10] # so you show on the first reading
020.  screenUpdate = True ; random.seed()
021.  nAv = 10 # number of samples to average
022.  avPoint = [0,0,0] ; p1 = [0] * nAv ; p2 = [0] * nAv
023.  runningAv = [p1,p2,[0]] ; average = [0.] * 3
024.  target = 0.5 ; timeChange = 0 ; scale = 700
025.
026.  def main():
027.    global tugState, gameOver, winner
028.    print("Tug of War")
029.    init()
030.    while(1): # do forever
031.      timeChange = 0
032.      tugState = -cp # middle of screen
033.      checkTarget()
034.      gameOver = False
035.      winner = -1 # no winer yet
036.      whistle.play() # start sound
037.      time.sleep(2.0)
038.      while not gameOver:
039.        checkForEvent()
040.        readVoltage()
041.        checkTug()
042.        checkTarget()
043.        if screenUpdate :
044.          drawScreen()
045.          updateMeters()
046.      if winner == 0:
```

Figure 3

Figure 4

▲ **Figure 3** Schematic of the signal-conditioning interface

▲ **Figure 4** Oscilloscope trace of the voltage output after conditioning

▶ **Figure 5** Physical layout of the signal-conditioning interface
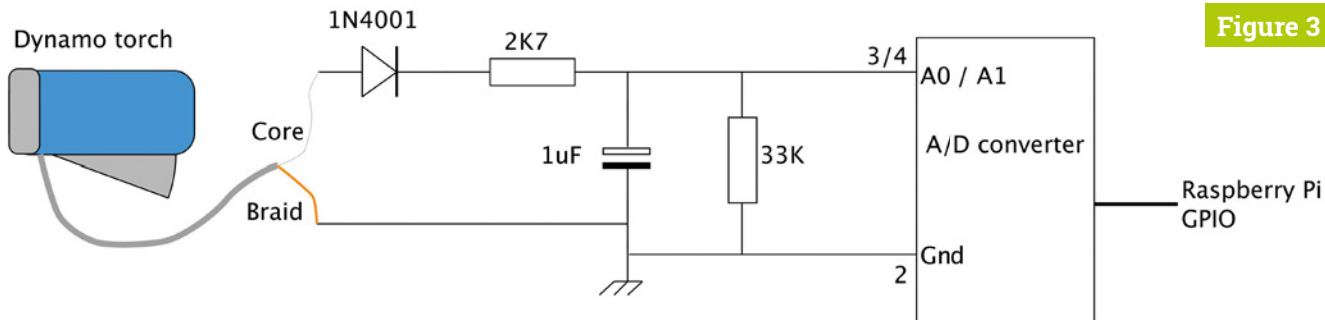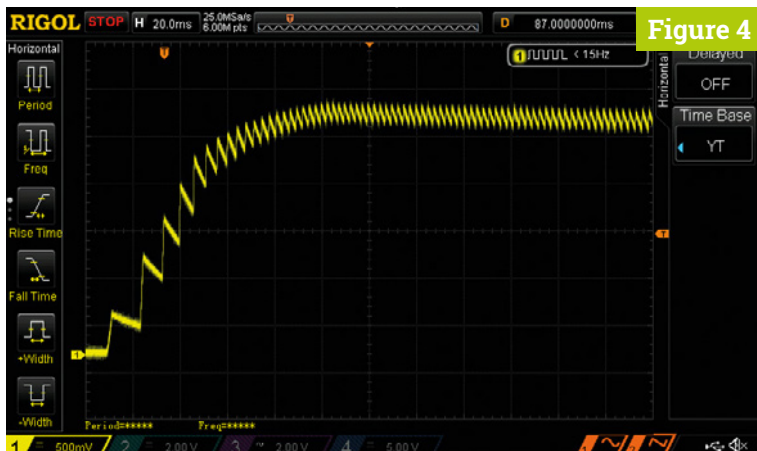
Figure 5

controls the release of the peak. The schematic for this is shown in **Figure 3**. When the signal is passed through this circuit, you get the waveform shown in **Figure 4**. You will need two of these circuits.

04 **Building the circuit**

We used a piece of 14 hole, by 10 row, stripboard and a single pin header row. This plugs into our ADC (see issue 68, page 42 – **magpi.cc/68**), component side down. The components were wired up as in **Figure 5**. Note that the track side shows where to cut the tracks and is flipped over right to left, just like you would see it. **Figure 6** shows a photograph of the conditioning circuit. The band on the diode marks the cathode, and the strip down the capacitor marks the negative wire. Make sure you get them the right way round.

> ❝ The band on the diode marks the cathode, and the strip down the capacitor marks the negative wire ❞

05 **Hacking the torch**

First off, drill a 2 mm hole in the body of the torch, close to the front, as shown in **Figure 7**. Then flip off the front cover and pull out the LED and battery assembly (**Figure 8**, overleaf). Be careful, because some of the wires are very thin and you don't want to snap them. Undo the two tiny screws holding the battery cover and remove the batteries. Now insert a length of 1.5 mm screened cable through the hole you previously drilled and strip off a 20 mm length at the end. Gather up the screen, twist it, and tin it.
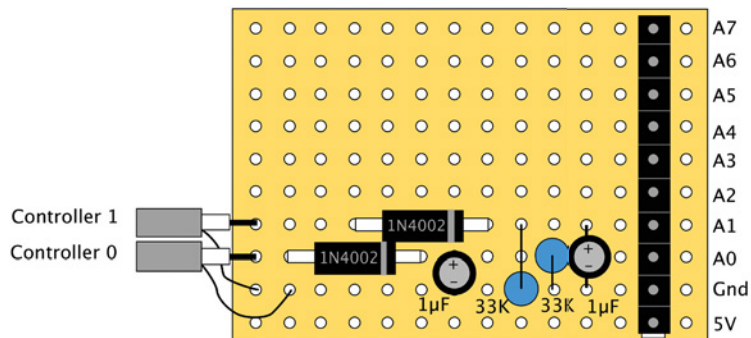
Figure 6
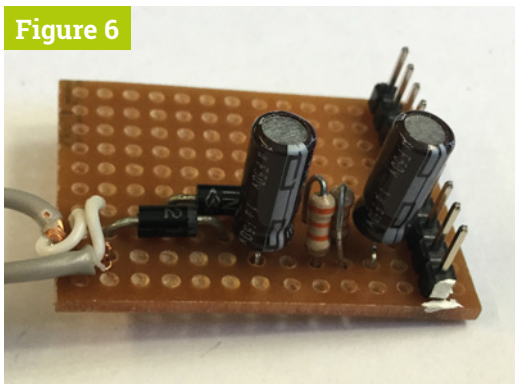
▲ **Figure 6** Photograph of the signal-conditioning interface

Top Tip 👍

Torches

We used two blue torches and identified the yellow player with a bit of painted card stuck on the lens cover. However, you can get other colour torches of the same type, so it would look better with a yellow torch.

06 Finishing off the torch

Use the cable's sleeving (that you cut off) to insulate the twisted screening and solder this to the sleeved side of the LEDs (**Figure 9**, overleaf). Then solder the core to the other side of the LED. Cut off the wire that used to go to the top of the battery housing and give all those long wires from the LED a bit of a trim. Glue the plastic lenses to the inside of the cover and slowly pull the cable back out of the torch. Fix the cable with a dab of hot-melt glue on the inside, before clipping the cover back in place.
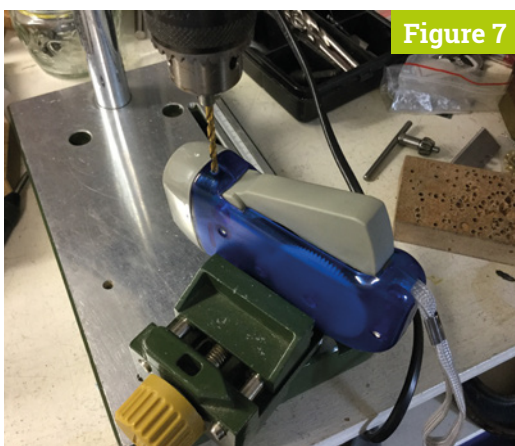
Figure 7

▲ **Figure 7** Drilling a hole for the connection wire

TugOfWar.py (continued)

> Language: **Python 3**

```python
047.        print("Blue Player is the winner")
048.        drawWords("Winner      ",123,159,(0,0,0),(20,178,155))
049.     else:
050.        print("Yellow Player is the winner")
051.        drawWords("Winner          ",742,159,(0,0,0),
     (20,178,155))
052.     pygame.display.update()
053.     end.play() # end sound
054.     print("Press space for another game")
055.     time.sleep(3.0)
056.     while gameOver:
057.       checkForEvent()
058.
059. def checkTug():
060.   global tugState,screenUpdate,gameOver,winner
061.   #check to see if anyone has won
062.   if tugState <= -869:
063.     gameOver = True
064.     winner = 0
065.     return
066.   if tugState >= -37:
067.     gameOver = True
068.     winner = 1
069.     return
070.   #check to see if anyone has scored
071.   p1 = abs(average[0] - average[2])
072.   p2 = abs(average[1] - average[2])
073.   if p1 < p2 : #player 1 closest
074.     if p1 < 40:
075.       tugState -= 1
076.       screenUpdate = True
077.   else:
078.     if p2 < 40:
079.       tugState += 1
080.       screenUpdate = True
081.
082. def checkTarget():
083.    global target, timeChange
084.    if time.time() < timeChange:
085.       return
086.    temp = random.uniform(0.2,0.8)
087.    target = int(temp*scale)
088.    average[2] = target
089.    timeChange = time.time() + random.uniform(3.2,6.8)
090.    drawScreen()
091.    updateMeters()
092.
```

Figure 8

▲ **Figure 8** Torch with the front removed

Figure 9

▲ **Figure 9** Attaching the connection wire

Top Tip 👍

Glue up the switch

The switch on the torch that was used to change over to battery operation, once it has been modified, disconnects the torch output. We found that this got knocked occasionally, leaving us to believe that the interface had stopped working. So we used polystyrene cement to make sure the output was always switched on.

07 Tug of War

What better way to show off a new interface than with a new game? So in the **TugOfWar.py** listing you will find our new, two-player, 'Tug of War' game especially designed for this interface. **Figure 10** shows the game in progress. The central meter shows the target, which is the reading you are aiming for. If your player's input is the closest to the target and is also within ten of the target, the rope is nudged in your direction. The first player to pull the rope over the finishing point is the winner. Pressing the **SPACE** bar starts another game.

▼ **Figure 10** Tug of War in play

Figure 10

08 A look at the code

The code follows the normal Pygame structure, and requires three images: rope, knot, and meter. It also requires a start sound and end sound. To smooth the input, a running average of the voltage readings is used. The `scale` variable is a sort of fiddle factor that allows you to adjust the output, so that you can get maximum meter deflection at the peak value from the torch. The `checkTarget` function will change the target you are aiming for at random intervals, to make the game a bit more challenging, so you need to look at the target and your input.

In conclusion

We hope you have fun with this. Another good game to implement using this interface would be a SpaceX rocket landing game similar to the classic Lunar Lander. However, next month we will show you how to use this interface to make a rather large LED Racer game. 🖥

Top Tip 👍

Removing the torch front

This can be tricky, but with a flat-blade screwdriver and some determination it can be removed. Mind that you don't stab yourself with the screwdriver – always push away from your body.

TugOfWar.py (continued)

```python
093.  def drawScreen():
094.      screen.fill(backCol)
095.      for i in range(0,3):
096.          screen.blit(meter, (meterPositionX[i],
          meterPositionY[i]) )
097.      screen.blit(rope, (tugState,190) )
098.      drawWords("Target",447,159,(0,0,0),(20,178,155))
099.      drawWords("Blue Player",123,159,(0,0,0),
          (20,178,155))
100.      drawWords("Yellow Player",742,159,(0,0,0),
          (20,178,155))
101.      pygame.draw.line(screen,(0,0,0),(64,188),
          (64,272),4)
102.      pygame.draw.line(screen,(0,0,0),(896,188),
          (896,272),4)
103.      pygame.display.update()
104.
105.  def drawWords(words,x,y,col,backCol) :
106.      textSurface = font.render(
          words, True, col, backCol)
107.      textRect = textSurface.get_rect()
108.      textRect.left = x # right for align right
109.      textRect.top = y
110.      screen.blit(textSurface, textRect)
111.      return textRect
112.
113.  def init():
114.      global meter, rope, meterPositionX,
          meterPositionY, spi,whistle, end
115.      whistle = pygame.mixer.Sound("sounds/whistle.ogg")
116.      end = pygame.mixer.Sound("sounds/end.ogg")
117.      meter = pygame.image.load(
          "images/MeterPC.png").convert_alpha()
118.      rope = pygame.image.load(
          "images/rope.png").convert_alpha()
119.      meterPositionX=[10,638,324]
120.      meterPositionY=[10,10,10]
121.      spi = spidev.SpiDev()
122.      spi.open(0,0)
123.      spi.max_speed_hz=1000000
124.
125.  def readVoltage():
126.      global screenUpdate, average, avPoint,lastValue,
          runningAv
127.      for i in range(0,2):
128.          adc = spi.xfer2([1,(8+i)<<4,0]) # request
          channel
129.          reading = (adc[1] & 3)<<8 | adc[2] # join two
          bytes together
130.          runningAv[i][avPoint[i]] = reading
131.          avPoint[i]+=1
132.          if avPoint[i] >= nAv:
133.            avPoint[i] = 0
134.          average[i] =  0
135.          for j in range(0,nAv): # calculate new running
          average
136.              average[i] += runningAv[i][j]
137.          average[i] = average[i] / nAv
138.          if abs(lastValue[i] - average[i]) > 8 or (
          average[i] == 0 and lastValue[i] !=0):
139.              lastValue[i] = average[i]
140.              screenUpdate = True
141.
142.  def updateMeters():
143.      global screenUpdate, average
144.      for i in range(0,3):
145.          plot = constrain(average[i]/scale,0.0,1.0)
146.          angle = (math.pi * ((-plot))) +
          (1.0 * math.pi)
147.          mpX = 146 + meterPositionX[i]
148.          mpY = 146 + meterPositionY[i]
149.          dx = mpX + 140 * math.cos(angle)
150.          dy = mpY - 140 * math.sin(angle)
151.          pygame.draw.line(screen,(50,50,50),(mpX,mpY),
          (dx,dy),2)
152.          screenUpdate = False
153.      pygame.display.update()
154.
155.  def constrain(val, min_val, max_val):
156.      return min(max_val, max(min_val, val))
157.
158.  def terminate(): # close down the program
159.      print ("Closing down")
160.      pygame.mixer.quit()
161.      pygame.quit() # close pygame
162.      os._exit(1)
163.
164.  def checkForEvent(): # see if we need to quit
165.      global reading, screenUpdate, average, gameOver
166.      event = pygame.event.poll()
167.      if event.type == pygame.QUIT :
168.          terminate()
169.      if event.type == pygame.KEYDOWN :
170.          if event.key == pygame.K_ESCAPE :
171.            terminate()
172.          if event.key == pygame.K_SPACE :
173.            gameOver = False
174.
175.  if __name__ == '__main__':
176.      main()
```

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

Part 03

More advanced
GUI layout

Learn how to expand your window, and position and resize buttons automatically, using C and GTK

We mentioned in last issue's tutorial that boxes can be either horizontal or vertical. Just to see the difference, try changing the line (in last month's code):

```
GtkWidget *box = gtk_vbox_new (FALSE, 5);
```

…to:

```
GtkWidget *box = gtk_hbox_new (FALSE, 5);
```

…and rebuild – you should then see a window like that in **Figure 1**.

This demonstrates the difference between a GtkVBox and a GtkHBox – they do exactly the same thing, but they just do it in different directions.

But what about those other arguments to `gtk_box_pack_start` that we glossed over earlier? These are designed to allow you to move things around within boxes and to give you more control over exactly where things end up within the window.

In the windows we have seen so far, this doesn't seem particularly useful, as the button and label both take up as much space as they need but no more.

▲ **Figure 1** The effect of using a GtkHBox instead of a GtkVBox

But GTK windows can usually be resized – the user can grab a corner of a window and stretch it to an arbitrary size in each dimension. GTK tries to resize windows and widgets intelligently so that things grow and shrink in proportion to each other; these additional arguments let you control what happens to the widgets when the window size changes.

The third argument to `gtk_box_pack_start` is called `expand`. If set to `TRUE`, then when a window is enlarged, the amount of space allocated for this widget is enlarged in proportion. If set to `FALSE`, the amount of space allocated for this widget is never more than the minimum it requires.

The fourth argument is called `fill`. This has no effect if `expand` is set to `FALSE`, but if `expand` is set to `TRUE`, it controls whether or not it is the space allocated to the widget or the widget itself which grows. If `expand` is `TRUE` and `fill` is `FALSE`, the widget itself stays the same size, but the space around it grows; if both `expand` and `fill` are `TRUE`, the widget itself grows to fill the increased space.

That is somewhat hard to follow when written down, but a few examples will help to get it straight. Change the GtkHBox back to a GtkVBox in the example above, and then set both `expand` and `fill` for all three `gtk_box_pack_start` calls in your example code to `FALSE`. If you then resize the window, the label and two buttons will all stay the same height, at the top of the window (**Figure 2**).

Now try the same thing, but change the `expand` parameter in the `gtk_box_pack_start` for `btn` to `TRUE` – the result should be like **Figure 3**.

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
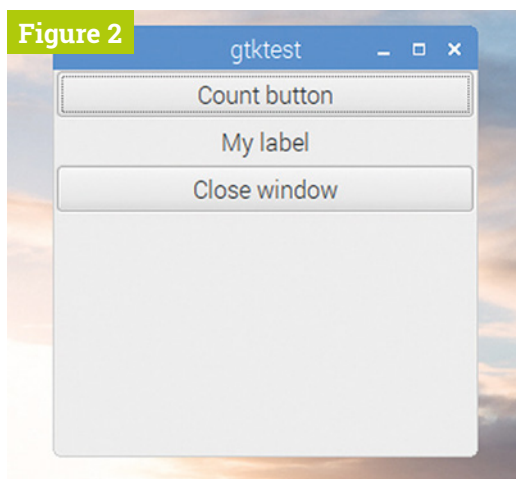
Figure 2

▲ **Figure 2** The layout with `expand` and `fill` both set to FALSE for all widgets

The button now moves down the window to fit in the middle of the increased space; the extra space has been allocated to the button, but the button itself is the same size.

Finally, try changing both `expand` and `fill` to `TRUE` for `btn` to see a result like **Figure 4** (overleaf).

Enlarging the window now causes the whole button to fill the extra space.

We can see that these parameters only affect the vertical size of the widgets; this is because they are used inside a GtkVBox. If we want to control the horizontal size of the widgets as well, we need to create a GtkVBox which contains a set of GtkHBoxes, which are then used to contain the widgets themselves.

The final argument in the call to `gtk_box_pack_start` is called `padding` – this is the amount of free space (in pixels) which is inserted at either end of this widget; this is in addition to the `spacing` which was requested between all widgets in the `gtk_box_new` call.

By careful tweaking of `expand`, `fill`, and `padding` parameters, you can create a window which displays all its widgets tidily and resizes exactly as you expect.

Tables

You might be wondering how you put, say, four items in a window with two on each of two lines. Well, it is perfectly acceptable to put a GtkHBox as one of the items inside a GtkVBox, or vice versa. It's even acceptable to put a GtkHBox as an item inside a GtkHBox (but it's generally a bit pointless). Using a combination of nested GtkHBoxes and GtkVBoxes, you can lay out all the widgets in a window the way you want them.

While nesting GtkHBoxes and GtkVBoxes works perfectly well for laying out widgets in two dimensions, and actually gives you the best control over how they appear, there is a simpler alternative: the GtkTable widget.

A GtkTable is a two-dimensional box, which is defined as having a certain number of rows and columns when it is created. Widgets can then be located at various row and column coordinates.

> ❝ A GtkTable is a two-dimensional box, which is defined as having a certain number of rows and columns ❞

This is actually not as flexible as using nested GtkHBoxes and GtkVBoxes, because every widget has to be aligned in both a row and a column – this means that any row is always at least as high as the tallest widget anywhere in it, and any column is always at least as wide as the widest widget anywhere in it; this can lead to a lot of wasted space. But quite often a simple arrangement like this is all that is needed, so a GtkTable works fine.

Here are the changes to our example if we use a GtkTable to position the label and buttons:

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

  GtkWidget *win = gtk_window_new (
GTK_WINDOW_TOPLEVEL);
```
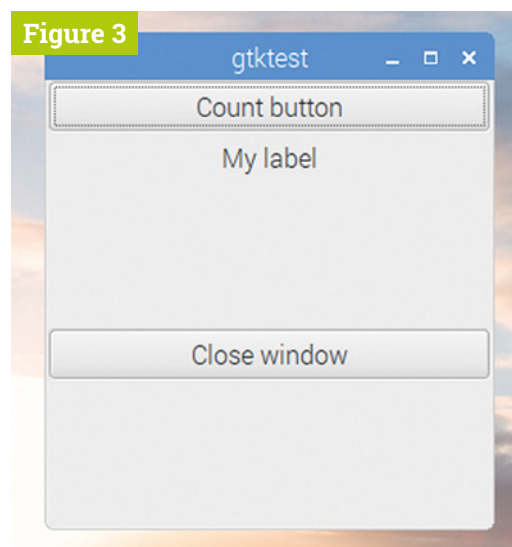

Figure 3

◄ **Figure 3** The layout with `expand` set to TRUE and `fill` set to FALSE for the 'Close window' button

Figure 4

gtktest

Count button

My label
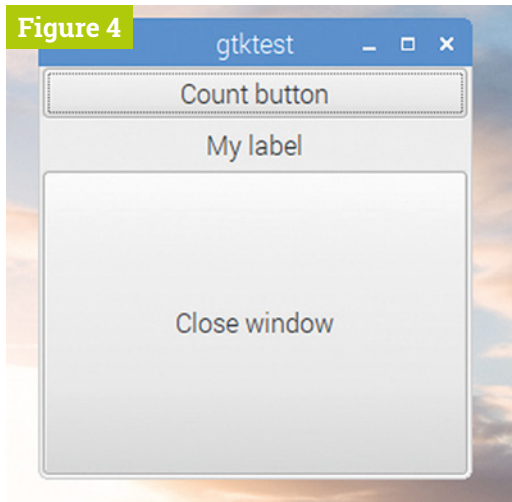
Close window

▶ **Figure 4** The layout with both `expand` and `fill` set to TRUE for the 'Close window' button

❝ We pass as arguments the names of the table and the widget we are putting into it ❞

```
    GtkWidget *btn = gtk_button_new_with_label
("Close window");
    g_signal_connect (btn, "clicked", G_CALLBACK
(end_program), NULL);
    g_signal_connect (win, "delete_event", G_
CALLBACK (end_program), NULL);

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

    GtkWidget *btn2 = gtk_button_new_with_label
("Count button");
    g_signal_connect (btn2, "clicked", G_CALLBACK
(count_button), lbl);

    GtkWidget *tbl = gtk_table_new (2, 2, TRUE);
    gtk_table_attach_defaults (GTK_TABLE (tbl),
lbl, 0, 1, 0, 1);
    gtk_table_attach_defaults (GTK_TABLE (tbl),
btn2, 1, 2, 0, 1);
    gtk_table_attach_defaults (GTK_TABLE (tbl),
btn, 0, 2, 1, 2);
    gtk_container_add (GTK_CONTAINER (win), tbl);
    gtk_widget_show_all (win);
    gtk_main ();
  }
```

We create a new GtkTable widget with:

```
    GtkWidget *tbl = gtk_table_new (
  2, 2, TRUE);
```

The arguments to the `gtk_table_new` function are the number of rows and the number of columns, respectively, and whether or not the table is homogeneous (which in the case of a table is whether or not every cell in the table is identically sized).

To insert a widget into the table, we use:

```
    gtk_table_attach_defaults (GTK_TABLE
(tbl), lbl, 0, 1, 0, 1);
```

We pass as arguments the names of the table and the widget we are putting into it. The following numbers are the coordinates where the widget is attached to the table – in order, they are the position of the left and right-hand edges of the widget, and then the top and bottom edges. So this positions the widget in the first column – between x positions 0 and 1 – and in the first row – between y positions 0 and 1.

Note that you can spread a widget across multiple rows and/or columns, as in how `btn` is placed:

```
    gtk_table_attach_defaults (GTK_TABLE
(tbl), btn, 0, 2, 1, 2);
```

In this case, the widget spans the first and second columns, because it is attached to 0 at the left and to 2 at the right.

For the sake of completeness, note that `gtk_table_attach_defaults` is a shortened version of the function `gtk_table_attach`, which has several other options. These other options include the ability to specify `expand`, `fill` and `padding` options for each widget, as in the `gtk_box_pack_` functions described in the previous tutorial. But these function calls then get rather long-winded to type out – using `gtk_table_attach_defaults` sets `padding` to `0` and sets `expand` and `fill` both to TRUE for each widget. Feel free to play with the full versions if you want!

If you build and run this code, you'll end up with a window that looks like the one in **Figure 5**. Note that the 'Close window' button spans both columns of the table, as described above. M

Figure 5

gtktest

My label   Count button

Close window
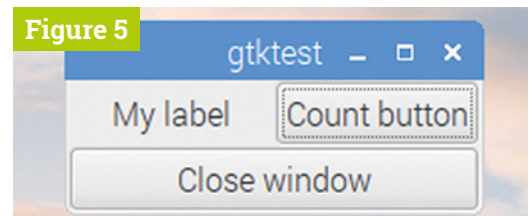
▲ **Figure 5** A GtkTable allows widgets to be aligned in rows and columns

Use your Pi to collect sensor data to the Google Cloud

UrsaLeo

Download our Raspbian package to turn your Pi into a Google cloud gateway. Display data on dashboards, store and download it, use it to drive emails, texts and other alerts.

Enter code `magpi1` at account creation to receive a free Pi debug board*
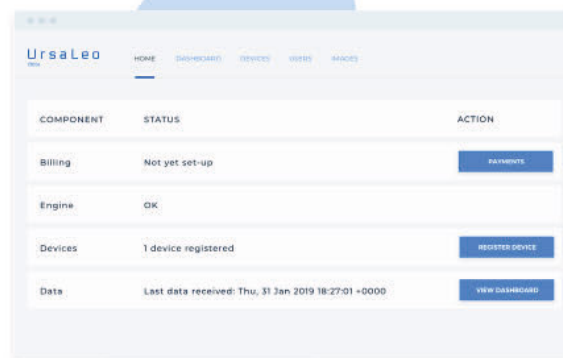
- LEDs indicate BLE, internet and cloud connectivity
- Console interface
- Safe power down switch

ursaleo.com/raspbian

*Offer available for North America and Europe only

Google Cloud Platform Partner

THE *Official*

RASPBERRY PI
PROJECTS BOOK

VOLUME 4

200 PAGES OF
IDEAS & INSPIRATION

DIY Games
Console
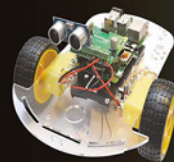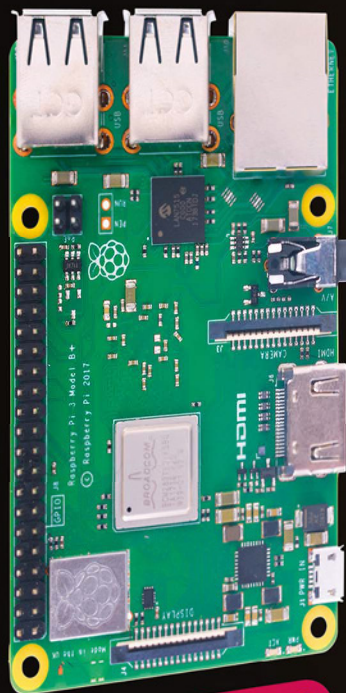
Build
a Robot
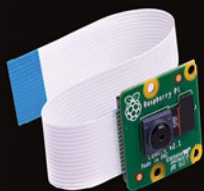
Set up
a Spy Cam

Make a
Magic Mirror

55 PROJECTS & GUIDES

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

THE *Official*

RASPBERRY PI
PROJECTS BOOK

VOLUME 4

£12.99
200 pages of
Raspberry Pi

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

Inside:

● How to get involved with the Pi community

● The most inspirational community projects

● Essential tutorials, guides, and ideas

● Expert reviews and buying advice

Available now **magpi.cc/store**

plus all good newsagents and:

WHSmith   BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

SCIENCE IN SPACE

TO CELEBRATE THE 50TH ANNIVERSARY OF THE MOON LANDING, WE USE RASPBERRY PI TO DO SCIENCE AMONG THE STARS

On 20 July 1969, humans set foot on the moon. It's one of the most important moments in the history of our civilisation, and this month it will have been 50 years since the momentous occasion.

While manned space missions don't leave a low Earth orbit these days, that doesn't mean exploration of space has stopped. It's only got better and more sophisticated. Experiments are performed daily on the International Space Station, while ground-based teams work through mountains of data from a vast network of satellites orbiting Earth, or spacecraft and rovers exploring our solar system.

Want to join in? We thought so – grab a Raspberry Pi and let's take it to the final frontier.
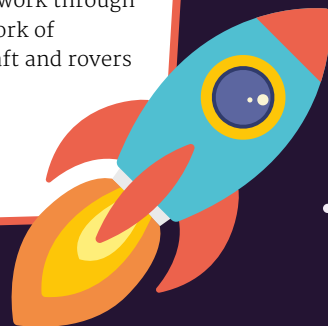
A BRIEF HISTORY
OF ASTRONOMY, ASTROPHYSICS, ET AL

From observing the stars, to flying among them

People of ancient times studied the movement of the cosmos using only the naked eye. Even so, they were able to map out star maps and even discover some of nearest planetary neighbours. The Babylonians and ancient Greeks became more sophisticated with their observation methods, and by the third century BCE, it was proposed that the Earth (and the other known planets) revolved around the sun.

The dark ages

Astronomy and astrology intertwined throughout the first millennium. By the end of the ninth century, Islamic scholars were setting up observatories to study the stars. The Andromeda Galaxy was described in 964 CE, while a supernova was observed around 1006 CE. Vitally important findings from the Middle East found their way to Europe, and by the 13th century medieval scientists were also making contributions to astronomy.

The telescope

The Renaissance brought about a scientific revolution in Europe. Copernicus proposed once again that the Earth revolved around the sun. Galileo, Kepler, and Newton all used telescopes,

▲ Celestial map from the 17th century, by the Dutch cartographer Frederik de Wit

▲ Apollo 14 Commander Alan Shepard on the moon. Yes, he did play a bit of lunar golf. *Source: NASA*

which let them peer even deeper into the heavens. People began seeing moons orbiting the other planets, and by 1781 Uranus was discovered, the first new planet observed since ancient times.

The modern era

Since then, planets, moons, celestial phenomenons, laws of motion, and more were being discovered; however, humans were still stuck on the ground. That was until the 1950s: advancements in technology during World War II resulted in the invention of rockets. Finally, there was a way for people to leave the Earth and explore space. By 1957, the first satellite, Sputnik 1, had been launched. In 1961, Yuri Gagarin became the first person to reach space. Eight years later, Neil Armstrong and Buzz Aldrin set foot on the moon.

Humans haven't rested on their laurels, though, as robotic rovers are exploring far-off planets and moons, while spaceships zip around the solar system to uncover other secrets. Sometimes they land on comets, which is supremely cool. Slightly closer to home, astronauts perform experiments on the ISS (sometimes using a Raspberry Pi!), and powerful radio telescopes probe the depths of the universe. The more we know about the universe, the more we discover about ourselves.

PHENOMENAL COSMIC PROJECTS

TURN YOUR RASPBERRY PI INTO A SPACE SCIENCE STATION WITH THESE PROJECTS

SHOOTING THE MOON

Take photos of the moon with a Raspberry Pi Camera Module

⭐ **The Raspberry Pi Camera Module is extremely versatile; however, you can't attach a standard Nikon or Canon camera lens to it.** Digital zoom is a bit rubbish, so how would you take a good photo of the moon with one? James Mitchell decided that, actually, he'd prefer to use a Canon DSLR lens, so he printed off a lens adapter. Then, with a bit of code, practice, and good luck, he was able to take some fantastic images of our very own natural satellite. How amazing is that?

magpi.cc/1Nd44Dx

SETI@HOME

With your help, we might find aliens

⭐ **Using software like BOINC (as seen in this tutorial from *The MagPi*: magpi.cc/2l2sQzI), you can turn your Raspberry Pi into one of thousands of nodes in a cluster of computers that crunch big numbers.** SETI@home is one of the most famous examples of these, allowing you to aid in the Search for Extra Terrestrial Intelligence (SETI) using an idle Raspberry Pi. Here's hoping they're friendly.

magpi.cc/kkhimA
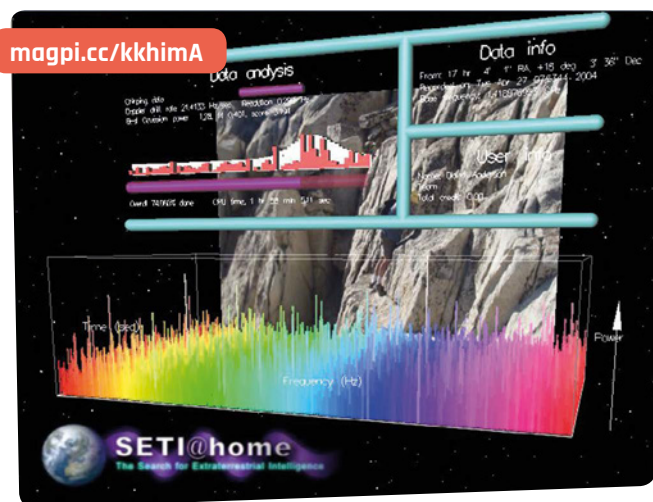
LUNAR PHASE CLOCK

Know when it's peak moon-watching time

⭐ **This project is one part art, one part science.** Using a bit of code and a lovely-looking custom-built clock, you can make sure you know exactly what the moon's phase is where you live. Whether you want to observe the moon when it's full, or take advantage of no moon for stargazing, the phase clock will help you out.

magpi.cc/HpJivV

ONGOING MISSIONS

Source: NASA
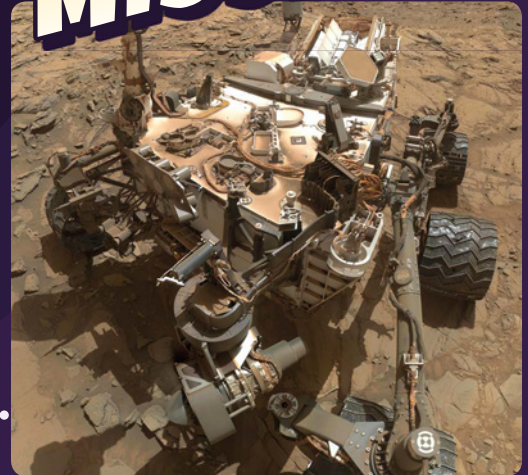
Curiosity/Mars Science Laboratory

The Curiosity rover has been on Mars for nearly seven years now, with four main goals: search for any signs of life; improve understanding of Mars geology; determine the planet's history; and keep an eye on surface radiation. It also likes to take selfies. We can relate to that bit. •

Hayabusa2

Did you know that we've landed spacecraft on asteroids? And that it was done remotely from thousands of miles away? Hayabusa2 specifically was able to land several rovers on the surface of an asteroid, and it will be returning to Earth at the end of next year with samples from it.

Juno

The Juno probe was launched in August 2011, and arrived at Jupiter in July 2016 . It's currently in polar orbit of Jupiter, and has a suite of scientific instruments on board in an attempt to more precisely measure the composition of Jupiter's gas clouds, the size of its core, its gravitational field, and more. In July 2021 it will plunge into Jupiter itself, so there's a few years of Jupiter-based discoveries left for it.

Source: NASA

SATNOGS GROUND STATION

Be part of a global network that listens into satellites

Building a single, solitary ground station to listen in on satellites won't be very efficient. You'll only ever receive a signal intermittently, maybe even once every few days if you're lucky. This is where SatNOGS comes in – it's an open-source ground station project that offers several designs you can use to join a network of other ground stations. You can use the software to set your ground station to look for specific satellites, or see what other stations are receiving. Read more about it in HackSpace Magazine #18: **magpi.cc/mynPsg**.

network.satnogs.org

BACK TO THE MOON?

Apollo 17 was the last manned mission to land on the moon, or any extraterrestrial body for that matter, on 7 December 1972. The expense of launching more moon missions makes the endeavour quite inhibitive; however Artemis 3, planned for launch in 2024, could end up as the first moon landing since Apollo 17. Next step: Mars in the 2030s.

APOLLO GUIDANCE COMPUTER

Emulate the actual computer used on the Apollo spacecraft

It turns out that the code for the AGC (Apollo Guidance Computer) is open-source and on GitHub. This is the same code from the famous photo of computer scientist Margaret Hamilton standing next to the stack of paper, now in a more usable form. Use it to emulate the special computer the Apollo mission used, and learn about the early days of computers at the same time.

magpi.cc/49

FUTURE MISSIONS

Source: NASA
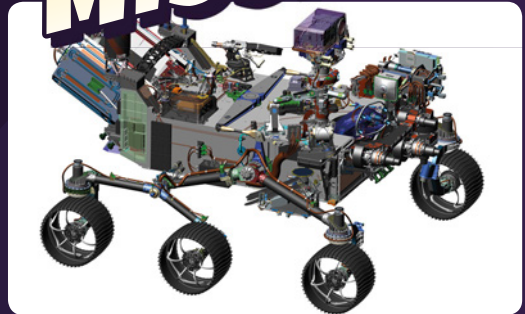
Mars 2020 *17 July 2020*

Slated to launch in mid 2020 and land early 2021, Mars 2020 is a rover that will study the geology of the Jezero crater, which is thought to have once held water. It will see if there are any signs of ancient life, and bring samples back to Earth for further study.

Source: NASA

Lucy *October 2021*

Lucy is planned as a probe that will study some of the so-called Jupiter trojans – asteroids that share Jupiter's orbit around the sun. Thought to contain the same materials that built the planets, studies could give us more knowledge about how our solar system was formed.

JUICE *June 2022*

The Jupiter Icy Moons Explorer, or JUICE, will be sent to Jupiter to study some of its largest moons for bodies of liquid water, specifically Ganymede, Calisto, and Europa. The current plan is for it to orbit Ganymede by 2032, becoming the first spacecraft to orbit a moon other than our own!

ASTRO PI

Want to send your code to space? The Raspberry Pi Foundation's Astro Pi programme has regular competitions to get projects created by school students and other groups of young people up onto the ISS. Keep an eye on the website for upcoming missions: **astro-pi.org**.

OPEN-SOURCE ROVER PROJECT

Build your own Curiosity rover

The Curiosity rover is very cool, and we love hearing about its various discoveries. While you won't be able to send a rover to Mars just yet, you can create a Mars rover to use here on Earth using the Open-Source Rover Project! The designs from the Jet Propulsion Lab are available on GitHub. They've been adapted for making a Raspberry Pi version.

magpi.cc/yZpbik

RASPBERRY PI-DRIVEN TELESCOPE MOUNT

Tracking stars precisely with a Raspberry Pi telescope

Computer-controlled telescopes are nothing very new; however, they are something that has always been relatively expensive. The benefits are pretty simple: you can easily program in a celestial body or just coordinates to look at, and the telescope will automatically look there. Using a Raspberry Pi Zero W, Dane Gardner was able to seriously upgrade his telescope and have it work wireless and flawlessly.

magpi.cc/EvwFdC
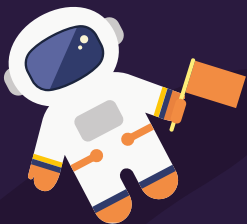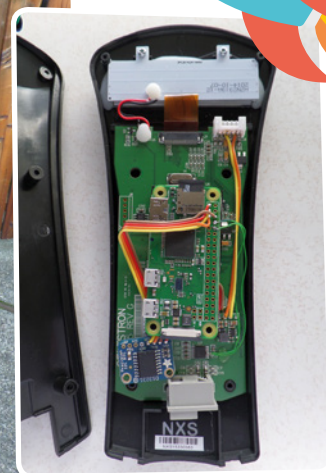
pi-top [4]

BE FIRST.

pi-top.com/kickstarter
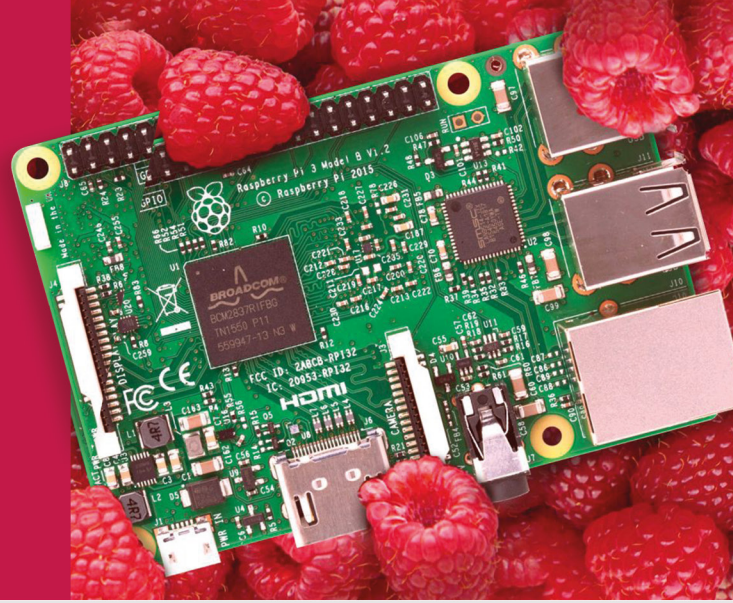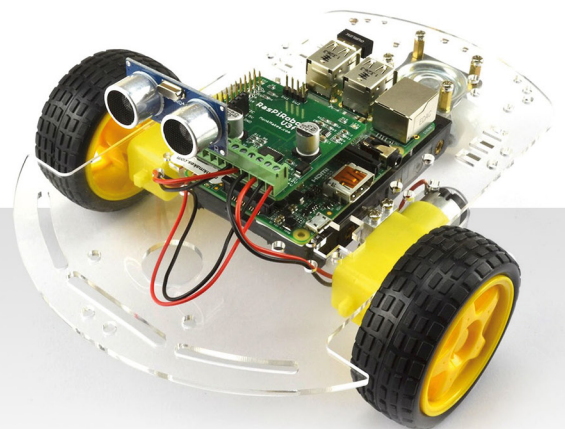
SEND A RASPBERRY PI TO (NEAR) SPACE!

HIGH-ALTITUDE BALLOONS CAN HELP YOUR RASPBERRY PI TOUCH OUTER SPACE

Unless you have access to a rocket or other kind of spaceship, it's a bit tricky to put your Raspberry Pi into space. However, with a very special balloon (and some luck with the weather) you can get pretty close. HABs (high-altitude balloons) can go over 30 km into the sky – high enough to get amazing photos of the curvature of the Earth.

Required hardware
magpi.cc/BnFynV

A HAB is made up of the balloon itself and the payload. At the very least you'll need to pack a tracker: there are GPS trackers that also use mobile data, SPOT trackers that use satellites, APRS which uses radio signals, and UKHAS radio trackers.

For photos, you can use a Raspberry Pi Camera Module or USB camera. You'll also need a parachute so it can land safely once the balloon bursts. A Raspberry Pi as the main computer will be required as well, along with a power supply for the electronics.

Programming

Software for the Raspberry Pi inside your payload is not too complicated: it's popular to live-stream the view of your camera, so using any internet connectivity you have in conjunction with the camera you've installed can be a good idea. Otherwise, you can have images save onto the Raspberry Pi as it goes.

A lot of trackers will usually have a lot of software already set up for them. Do your research and make sure you've got everything tested before attempting a launch. Dave Akerman has a great guide: **magpi.cc/2tc4pz2**.

Launch preparations

There's a long list of things to consider prior to launching your balloon. In the UK, you need permission from the Civil Aviation Authority. You'll need a location and date, and will need to watch the weather and flight path predictions.

For flight predictions, use the CUSF Flight Predictor (**magpi.cc/kmKTRA**). Use this to plan a safe flight path, where to launch, and where to try to retrieve the payload. Dave Akerman suggests making sure you're away from big cities and bodies of water, such as the sea.
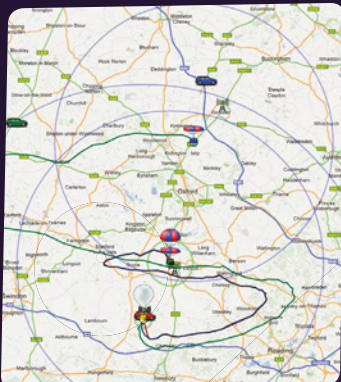
50TH ANNIVERSARY LAUNCH!

July is likely to be a very busy month for people in the space community, as the 50th anniversary of the Apollo 11 mission is celebrated by everyone! We got in touch with Pi HAB launcher extraordinaire Dave Akerman, who is working on an amazing launch for 16 July to commemorate the launch of Apollo 11 on 16 July 1969. Here's a small preview of the special model ship in the payload…

Enviro+

▶ Pimoroni ▶ **magpi.cc/ppkiiN** ▶ £45 / $48

Monitor your world with this all-in-one environmental sensing board. By **Phil King**

Developed in conjunction with Dr Nate Adams, a molecular biologist at the University of Sheffield, the Enviro+ turns your Raspberry Pi into a complete environmental monitoring station. For this it features four built-in sensors, some of them multifunctional, so it can gather plenty of useful data, including for air quality. Not only that, but if using it in a headless Raspberry Pi setup, without a monitor, its tiny colour LCD screen offers a convenient way of displaying readings. There's also the option of plugging in a particular matter sensor (not included).

Like the earlier Enviro pHAT – reviewed back in issue 49 (**magpi.cc/49**) and still available – the new board has a slimline pHAT form factor that matches Raspberry Pi Zero, although it can be used on any Raspberry Pi model. This time no soldering is required, as it comes with a female GPIO header attached.

Lacking the earlier board's motion sensors, the Enviro+ is intended purely for environmental monitoring. To this end, it incorporates a range of useful sensors.

Sensory overload

First up, a standard BME280 weather sensor is used to monitor temperature, barometric pressure, and humidity. This is positioned at the left edge of the board, away from the Raspberry Pi's CPU. Even so, you'll need to adjust its temperature reading (by measuring that of the CPU itself and deducting a factor of it).

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

Most notable is the inclusion of a MiCS6814 analogue gas sensor. This can detect three different groups of gases: reducing, oxidising, and NH_3 (ammonia). While levels of individual gases can't be discerned for the first two groups, the major ones are carbon monoxide (reducing) and nitrogen dioxide (oxidising).
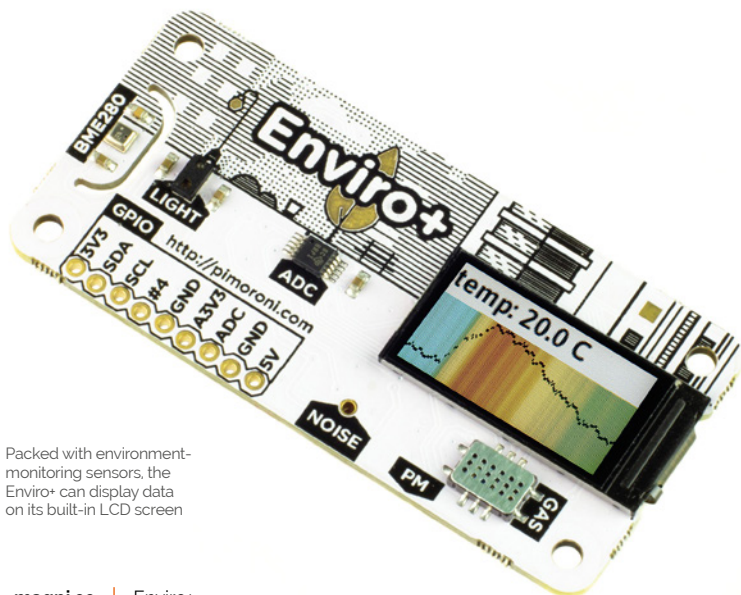
▶ Packed with environment-monitoring sensors, the Enviro+ can display data on its built-in LCD screen

▲ Available separately, the PMS5003 sensor measures the number of tiny particles – up to 1, 2.5, and 10 microns – in the air
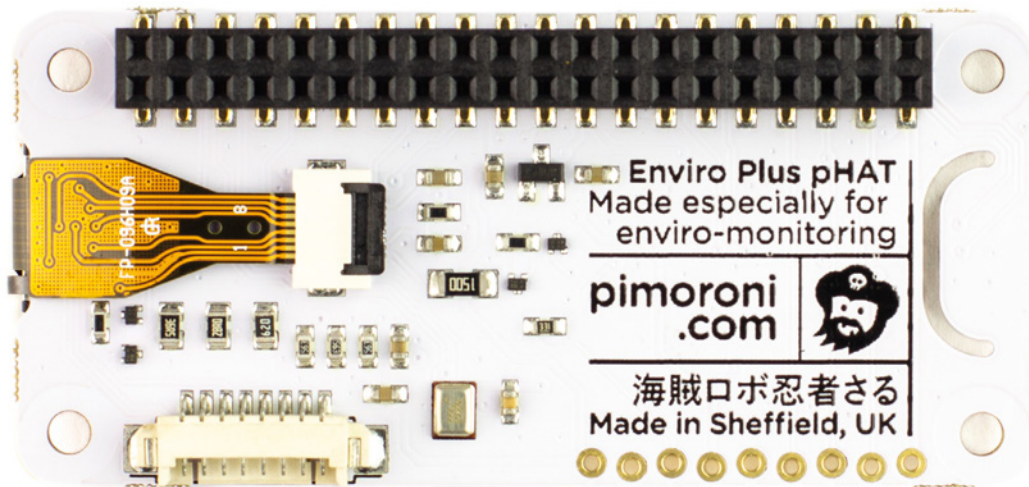
Near the gas sensor is a port to attach an optional particulate matter sensor, such as the Plantower PMS5003 (available separately for £25). This is used to measure numbers of tiny particles of sizes up to 1 micron (ultra-fine), 2.5 microns (combustion particles, organic compounds, metals), and 10 microns (dust, pollen, and mould spores). The board also features a nine-pin unpopulated header connected to selected GPIO pins. The finishing touch is the inclusion of a

Citizen science

Luftdaten is an open data project with a worldwide network of citizen scientists monitoring the air quality of their local environment – and, equipped with an Enviro+, you can become a part of it. Just run the **luftdaten.py** code example, register on the website (including your Raspberry Pi's displayed ID number), and you can start contributing your data – from the built-in BME680 weather sensor and add-on PMS5003 particulate matter sensor – which will then be shown on the site's world map.

0.96-inch colour LCD screen. It may be small, but it's ideal for displaying data out in the field, in a headless setup. It can even show some cool-looking scrolling graphs for live data, as shown in one of the Python code examples provided.

Environmental examples

Several code examples are included with the Enviro+ Python library for the board. Installation is simple enough, involving three terminal commands. The install script enables I²C, SPI, and serial interfaces on your Raspberry Pi, disables the serial console, and also enables a mini UART interface for the optional PMS5003 particulate matter sensor. If you ever need to revert this configuration change, there's an uninstall script.

The most impressive code example is **all-in-one.py**, which demonstrates most of the features of the board, taking readings from the various sensors (bar the mic) and displaying them in scrolling graph form on the mini LCD. To switch the latter between different readings, you simply tap the light sensor with your finger.

Another code example enables you to become a citizen scientist by uploading live data (from the BME280 and PMS5003) to the Luftdaten open-source air-quality monitoring project website (see 'Citizen science' box).

While the Enviro+ may seem a little pricey for a pHAT, it does cram a lot of useful sensors – which we reckon, if bought separately, would cost around £40 or more – into a handy package, along with that cool LCD screen to display your data. M

Verdict

If you want to create an air quality-monitoring project, this board is ideal, packing a raft of useful sensory tech into a small form factor, along with a handy LCD screen to display your data.

9/10

The official **Raspberry Pi** Beginner's Guide

2nd Edition

by **Gareth Halfacree**
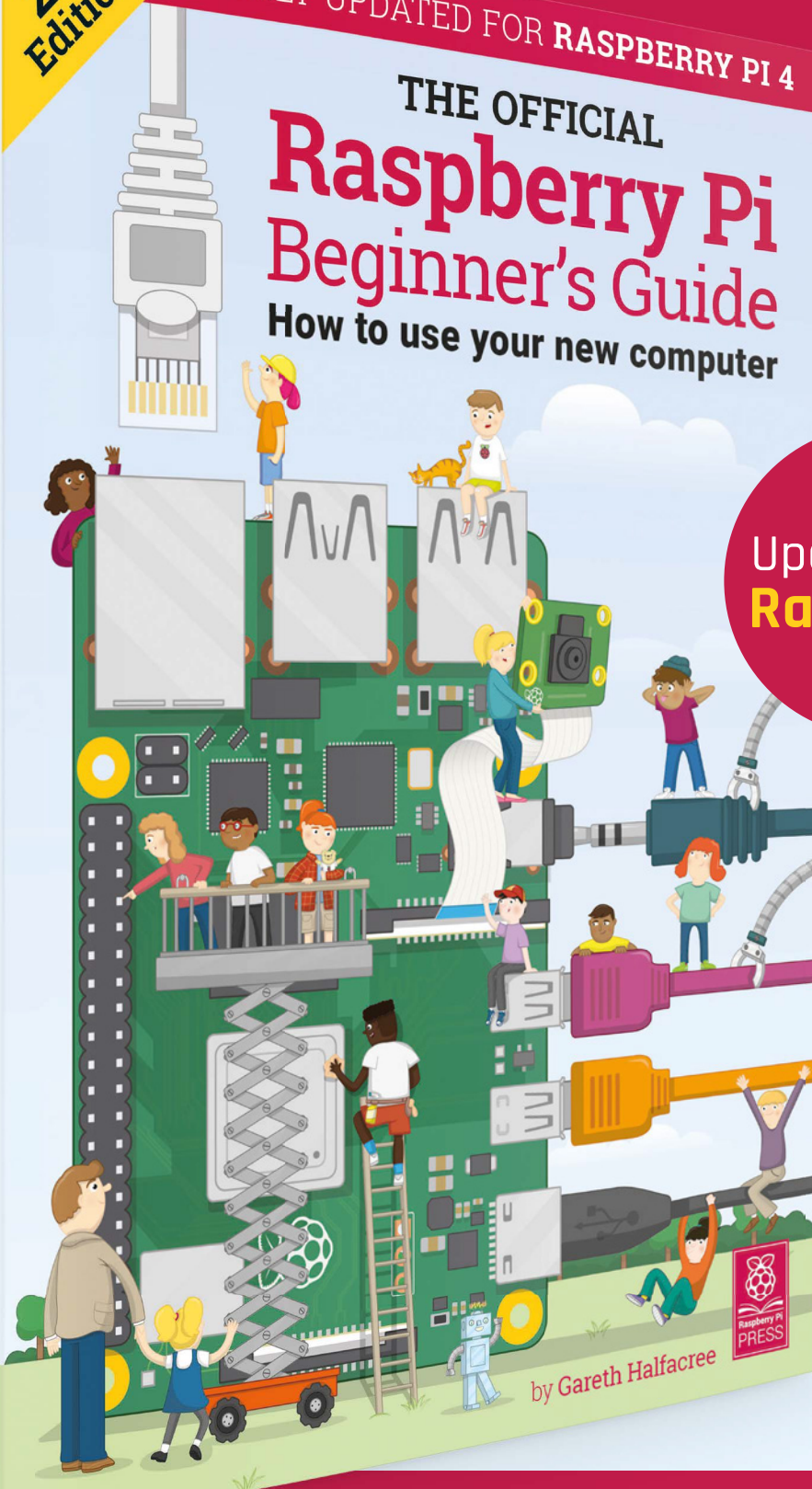
FULLY UPDATED FOR **RASPBERRY PI 4**

THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer

by Gareth Halfacree

Raspberry Pi
PRESS

Fully
Updated for
**Raspberry
Pi 4**
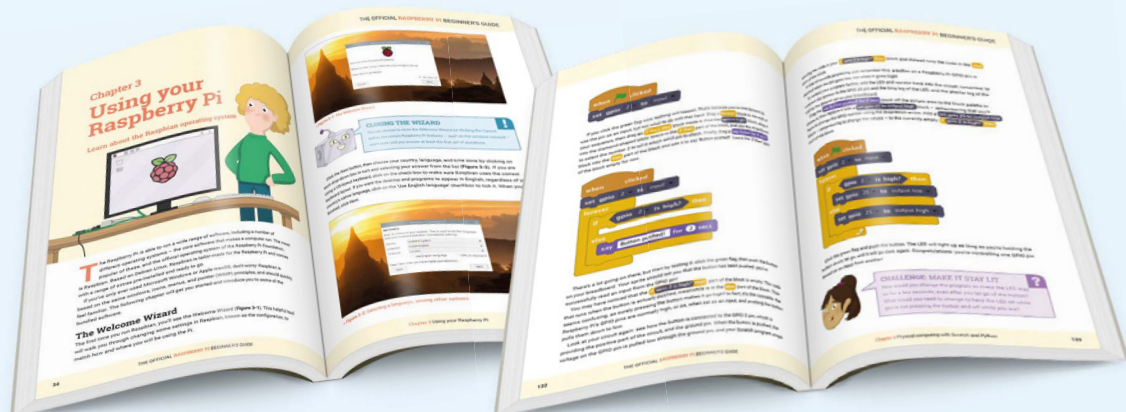
10 Best:

Raspberry Pi Zero projects

Get making with your Raspberry Pi Zero

We absolutely adore Raspberry Pi Zero. We've managed to fit the tiny, tiny computer into so many bits of electronics over the years to improve them that we sometimes wonder if we need to keep one in our pocket at all times. You know, for maker emergencies. If you've not had a chance to use a Raspberry Pi Zero yet, here are some projects that might inspire you to pick up the $10 tiny computer. **M**

ZeroPhone

Zero-sized mobile

This is truly a feat of excellent engineering: the ZeroPhone is a functional mobile phone that is fully open-source and just needs you to add a Raspberry Pi Zero and a SIM card. You might want to think about 3D-printing a nice case as well.

▶ £50
▶ **magpi.cc/rXgkZf**

Console-troller

Console in a controller

We made this in *The MagPi* way back when Raspberry Pi Zero launched, and it still works to this day, even if it is DIY. All it needs is a cheap USB game controller, a bit of Dremeling, and a touch of solder. A shout-out to RetroPie too for making this painfully easy.

▶ **magpi.cc/40**

Trinity Buoy Wharf Lighthouse

A model rock warning

Created for a wedding, these 3D-printed lighthouses not only light up very brightly, but also have a mini display at the base that lets people know the table number and the status of the meal. It's very fancy.

▶ **magpi.cc/dUiqTP**

The Seeing **Wand**

Magic pointing stick

This is a very DIY project, using some old PVC tubing as a device to hold a Raspberry Pi Zero and act as a guide to where you're pointing at. It uses a Raspberry Pi Camera Module in conjunction with Microsoft Cognitive Services to tell you what it's pointed at.

▶ **magpi.cc/pfpPwB**
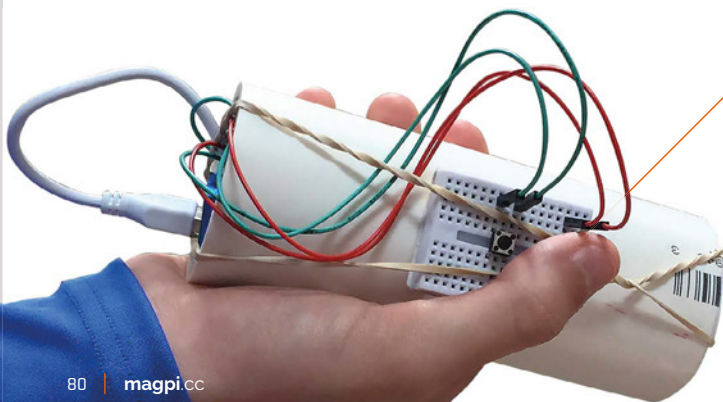
Zoe the Zero

A tiny flier

This small drone was built using a Raspberry Pi Zero from the cover of issue 40 of *The MagPi*, and in record time as well. Andy Baker built it within the week. He unfortunately passed away very recently, but we'll always remember his amazing Raspberry Pi work.

▶ **magpi.cc/piDYPq**

3D body scanner

A lot of cameras

The great thing about Raspberry Pi Zero is that it's very small. Also, it's cheap. So using a lot of them in one project is not tricky – like this 3D body scanner that takes incredible 3D photos.

▶ **magpi.cc/Vrdgov**

Tiny**Pi Pro**

The smallest console ever

This tiny DIY kit lets you turn a Raspberry Pi Zero into an absolutely minuscule handheld games console. Is it practical? Not entirely, but it's still extremely cool, and the build process is quite fun as well.

▶ £69
▶ **pi0cket.com**

Green sea **turtle tags**

Track turtles with a Raspberry Pi Zero

Nature preservation is a wonderful endeavour, and the Arribada Initiative is using Raspberry Pi Zero camera trackers to keep an eye on the well-being of green sea turtles. They're an endangered species and are at real risk from harm from discarded fishing nets, so this is vital work.

▶ **magpi.cc/NDiSwn**

PolaPi-**Zero**

A tiny, instant camera

Polaroid cameras are once again making a comeback of some kind, so it's only natural that people try to emulate the concept with a digital camera. One of the many tries at this is the PolaPi-Zero, which uses a thermal printer to print out instant images.

▶ **magpi.cc/ubWKNK**

All-in-one computer mouse

Zero in a mouse

Putting a Raspberry Pi Zero into something is everybody's dream. One person took this to the ultimate extreme and 3D-printed a mouse with a tiny screen, a pull-out keyboard, and a Raspberry Pi Zero inside. It's a full computer… only it's a mouse.

▶ **magpi.cc/kEzDbf**

Learn to upscale old technology
with Raspberry Pi

Take old, unused tech and give it a new lease of life with a Raspberry Pi. By **David Crookes**

Upcycled Technology: Clever Projects You Can Do With Your Discarded Tech
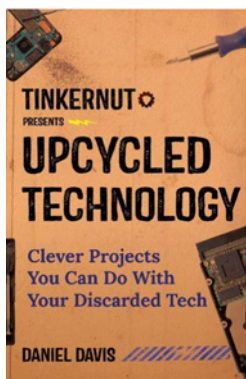
Half the battle of upcycling technology is coming up with a concept. The rest is identifying and working with devices that can help you achieve your aim.

This new book by creative technologist Daniel Davis hopes to inspire, guiding you through six projects that help you turn old webcams, laptops, CD-ROM drives, smartphones, iPods, and mobiles into backup cameras, projectors, 3D printers, security cams, and smartwatches.

In doing so, it examines key stages of the upcycling process, explaining why you should reconsider chucking away your old tech, and even describing where you may find stuff that others don't want.

In that sense, it's a handy resource. A crucial chapter picks apart a handful of tech and looks at the useful components they contain, and there's a good look at the tools you'll likely need for your projects – including a Raspberry Pi, of course.

That said, none of the projects actually uses a Raspberry Pi and, despite the steps clearly explaining the processes, it often skims when it could go in-depth. But you do get a solid grip of the various concepts and it's a good introduction to upcycling, particularly in its cheaper e-book form. **M**

AUTHOR
Daniel Davis

Price:
£20.99/$13.49 (hardback)
£6.43/£8.17 (Kindle e-book)

tinkernut.com

Video guides

See Raspberry Pi used in upcycled projects

OLD TECH. NEW SPEC.
Martin Mander loves breathing new life into old tech and his YouTube channel explores his Raspberry Pi-based projects to date, complete with links to Python scripts and well-written, clear instructions.
magpi.cc/e0idSo

INSTRUCTABLES CIRCUITS
There are loads of cool projects on Instructables, many of which upcycle old technology using a Raspberry Pi. The majority use videos to demonstrate the results alongside full instructions.
magpi.cc/BvVkgm

TINKERNUT
Yes, it's Tinkernut again (see the book), but for good reason: the site contains some lovely video tutorials about upcycled Raspberry Pi projects, a few of which are accompanied by step-by-step guides.
tinkernut.com

iFixit

You shouldn't generally tackle technology like a bull in a china shop and hack away at products' innards without thought.

It's always a good idea to see what goodies they contain first, so that you can determine what could be used in your upcycled technology projects and how they can best fit with a Raspberry Pi.

This wiki-based site has repair guides that take you under the skin of a device to look at specific parts. It also has full tear-downs which show you the safest way to dismantle lots of different technology – perfect for identifying any problems and every significant component you'll come across. Full lists of tools used are available, and there's a store where you can buy any you need, too. Ⓜ

AUTHOR

iFixit with help from users

Price:
Free

ifixit.com
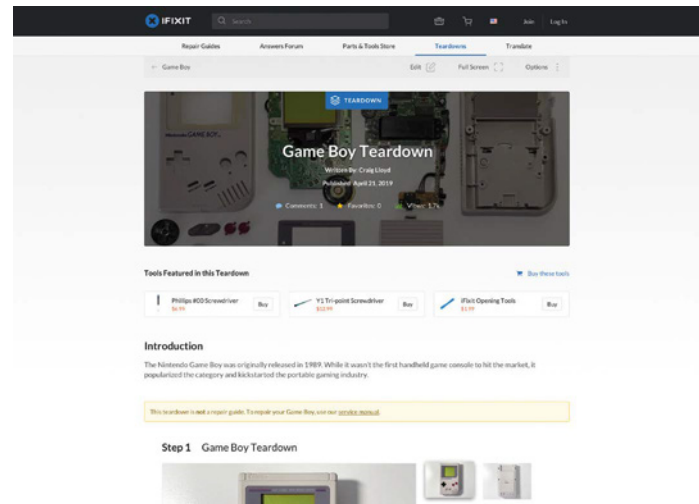
Retro Raspberry Pi Hands-On Hardware Projects

As well as running the excellent YouTube channel Adam Builds (magpi.cc/mebsYy), IT technician Constantin Adam has created a wonderful three-hour online course dedicated to upcycling using a Raspberry Pi.

Spread over eight multi-sectioned chapters, each of which lasts roughly 15 to 20 minutes, Adam carefully explains how to build a wireless Amazon Alexa from an old calculator, and an internet radio player from an original 1980s set. He converts a 1990s joystick into a games console, and turns an older television into a smart mirror. And that's only half of it.

The course includes some upcycling vitals, including exploring appropriate platforms, as well as planning and writing code using APIs. Available to view on mobile and computer, it also covers some Raspberry Pi basics, which means even beginners can ease themselves in. Ⓜ

AUTHOR

Constantin Adam

Price:
£8/$10 (special offer)

magpi.cc/HgOKAU
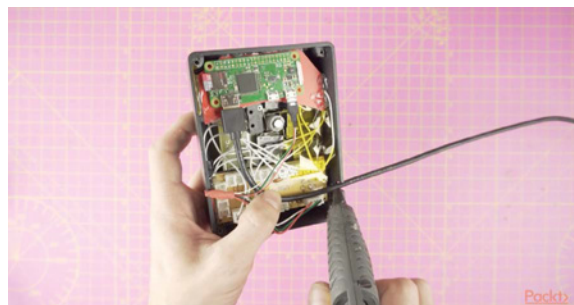
Manuals

Learn more about the tech you're adapting

MANUALS LIB
Finding most circuit diagrams for old tech will likely require a Google search, but we found a good number of scanned service and reference manuals among the millions of user guides here.
manualslib.com

ARCHIVE.ORG
Archive.org has 239 collections of manuals, datasheets, instructions, and user walkthroughs. You can filter them by media type, year, topic, creator, and language, or else perform a direct search.
magpi.cc/vNdWqm

VINTAGES HIFI
Upscaling audio equipment and need to look at a service manual? Vintages HiFi has lots of schematic guides, operating instructions, and diagrams, as well as brochures and catalogues.
magpi.cc/wbaoZu

Chris Barnatt

Want computers explained to you? Then meet
Chris Barnatt, futurist and tech YouTuber

> Day job **YouTuber** | > Website **magpi.cc/TMWLpb**

In the dark ages before YouTube, finding shows about technology, science, and computers on TV was a bit tricky. They were few and far between, so when *Tomorrow's World* and *How 2* were on, a lot of young people watched. Now, with 2020 just around the corner, there's a wealth of video channels online to satisfy your cravings for tech knowledge – and one of

▼ Raspberry Pi is now featured on Chris's ExplainingComputers YouTube channel

them is ExplainingComputers, run by Chris Barnatt.

"I remember reading the Ladybird book [*'How it works':
The Computer*] when I was six or seven years old," Chris tells us. "[I was] fascinated by the pictures of magnetic core storage, which were still in use at the time. When I was about 13, my parents bought me a ZX81 – in some ways the Raspberry Pi of its day – and I soon learnt to program it in BASIC and assembler, and wrote my first articles for computer magazines. I subsequently spent 25 years lecturing in computing and future studies in the University of Nottingham."

In 2007, Chris launched **ExplainingComputers.com** as an 'online computing textbook', which led to him starting the YouTube channel in 2008. This has covered a huge range of tech concepts and projects, and became a weekly show at the end of 2015. From last year, Chris has added Raspberry Pi videos. He makes use of his production experience at the BBC to add animations to his videos as well.

When did you first learn about the Raspberry Pi?
I remember watching a report on th Raspberry Pi on the BBC's *Click* TV show shortly after the board was first launched. But I

▲ The aptly named DEVASTATOR ROBOT is powered by a Raspberry Pi Zero
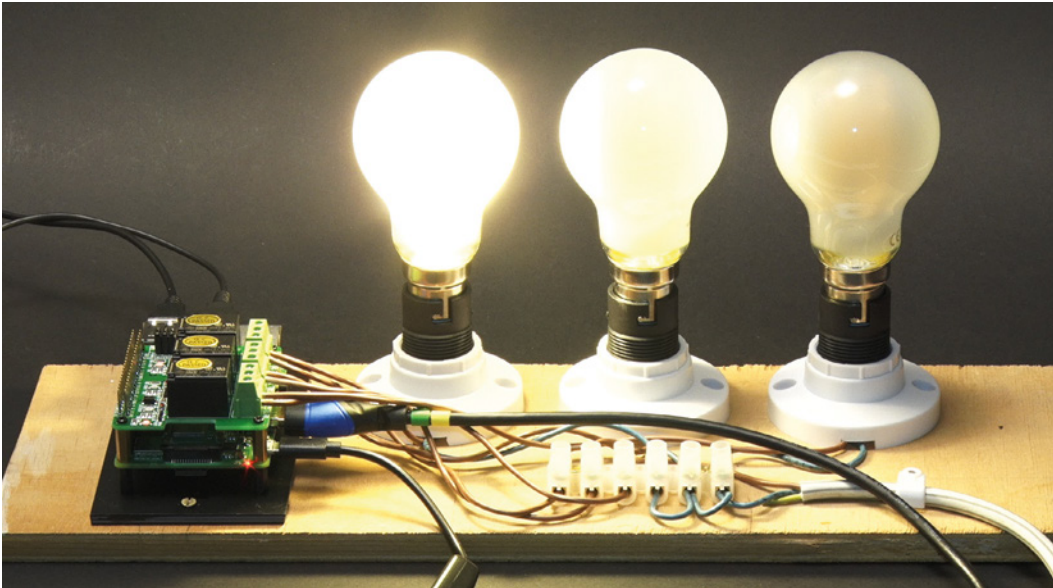
◄ A light-bulb moment. Always take great care when using mains electricity

didn't get one until a year or so later. Since that time, Raspberry Pi, and SBCs (single-board computers) more generally, have been an increasing part of my life! I now have about 40 SBCs, including every Raspberry Pi aside from the first Model A.

What are some of your favourite videos that you've made?
On ExplainingComputers, my favourite videos are often those that show viewers how to do things with low-cost hardware and/or free software – which is why I like working with Raspberry Pi, and why it is so popular with my audience. I remember making a video a few years ago called 'Raspberry Pi Week', when I used a Raspberry Pi 2 for a week as my only PC. This included delivering lectures for clients from a Raspberry Pi, which all worked, so this was a great experience. I've also enjoyed making videos about things like robotics and automation, often involving a Raspberry Pi, and have had great success with videos about Raspberry Pi cooling. Some of the

> 66 **I think I now have about 40 SBCs, including every Raspberry Pi aside from the first Model A** 99

techniques and test scripts I've shown in these cooling videos have been fairly widely adopted around the web, and it is always great to make YouTube content that others embrace like that, and which people find helpful.

More broadly, I've always like making videos – both on my ExplainingComputers YouTube channel, but also

my other channel called ExplainingTheFuture – that show people something that does not exist yet, and where I can include a lot of CG animation. So I'm proud of the videos I've made on subjects like brain-computer interfaces, bioprinting, and asteroid mining, even though they are not always the most popular. M
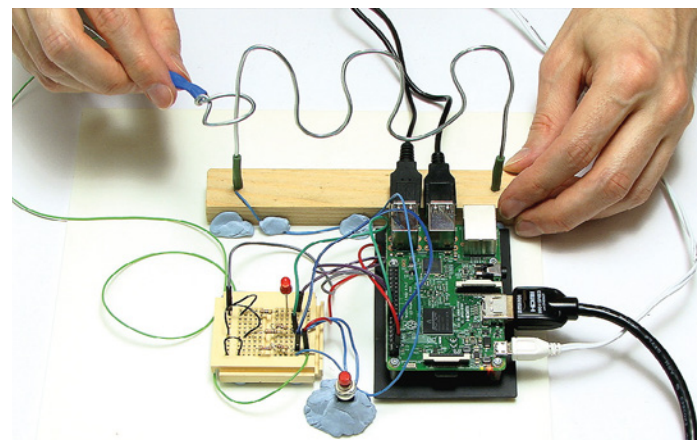
▲ A classic game, remade with Raspberry Pi

Raspberry Pi 4 and the future of computing

"I think it's a very exciting time in computing, with the lines between traditional desktops and SBCs continuing to blur as more computer power moves to the 'edge'. In this context, the new Raspberry Pi 4B is, I think, a signature development, as it is clearly capable of being used as a device for doing most of the home computing basics – surfing the web, email, playing media, running office apps, and things like that. I'm pretty sure we'll have desktop PCs for a good many years to come, but they will increasingly look more and more like a Raspberry Pi."

This Month in **Raspberry Pi**

A royal visit

His Royal Highness Prince Andrew visits Raspberry Pi HQ

Did you know that HRH Prince Andrew, the Duke of York, is the Patron of the Raspberry Pi Foundation? He's a big advocate of technology, and has hosted Raspberry Pi at Buckingham Palace and St James's Palace. He made a visit to the Raspberry Pi HQ last month to see what the Foundation was up to. **M**

01. The CoderDojo team from Ireland meet HRH

02. Foundation staff were on hand to show him what Raspberry Pi was up to

03. The Prince gets to know the Astro Pi

04. Code Club is an important part of the Raspberry Pi Foundation

05. The Prince and Raspberry Pi Foundation CEO Phillip Colligan inspect the maker space

06. The Duke of York talks to the Foundation Members and staff in attendance

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. This e-ink clock looks very nice!

02. A still image doesn't do this time-lapse video justice – take a look here: **magpi.cc/sSPYiF**

03. We always like seeing a nice, cleanly built robot

04. Robots are a good advertisement for Raspberry Pi

05. We wonder what the good Lady Ada has to say?

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

Strawberry4Pi 2

This IoT add-on makes it easier for you to control your Raspberry Pi home automation projects from a smartphone using a special app. Along with a temperature sensor and WiFi configuration button, it features four relays to switch connected devices on and off, which should make overall control a little simpler.
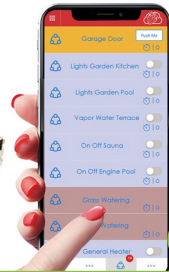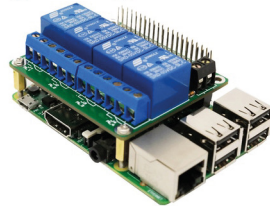
▶ **kck.st/2wOHy4o**

INcase ONE

The description as "The World's Best Single Board Computer Housing System" seems like a grand claim, but it looks like a neat case either way. Designed for Raspberry Pi, Arduino, BeagleBoards, etc., it's weatherproof and looks like it would make a good CCTV or motion sensor alarm.

▶ **kck.st/30IKaOM**

Best of the rest!

Here are some other great things we saw this month

ANDROID AUTO VIA RASPBERRY PI

This is a cool little hack – we much prefer Android Auto to some stock navigation and media software. Reddit user icecube45 was able to directly interface a Raspberry Pi 3 with the display's input to make this work.

▶ **magpi.cc/zMpAGQ**

DIY SMART GLASSES

Official smart glasses products seem to be making a return, but that hasn't stopped Reddit user Infranix from making his own amazing DIY pair using a Raspberry Pi Zero. We love the futuristic sci-fi look.

▶ **magpi.cc/0dcZVG**
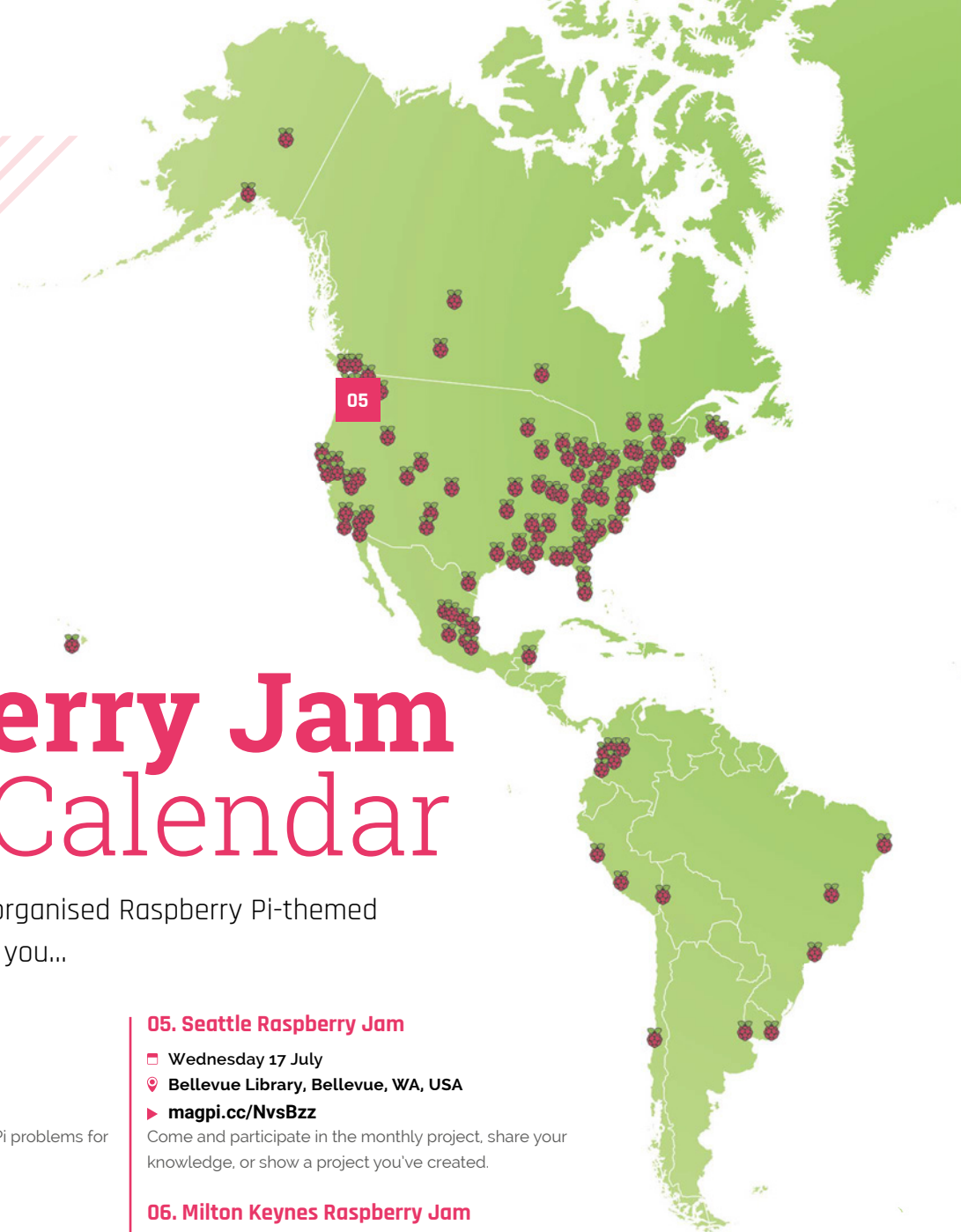
Raspberry Jam
Event Calendar

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

01. Exeter Raspberry Jam

📅 Saturday 6 July
📍 Exeter Library, Exeter, UK
▶ magpi.cc/vhtijy

Bring along your projects and Raspberry Pi problems for great conversations.

02. Potton Pi & Pints

📅 Saturday 6 July
📍 The Rising Sun, Potton, UK
▶ magpi.cc/QFwRbT

Raspberry Pi-related fun in a relaxed setting. Meet like-minded people at Potton Pi & Pints.

03. Stafford Raspberry Jam

📅 Tuesday 9 July
📍 Stafford Library, Stafford, UK
▶ magpi.cc/Swkveu

Welcoming anyone that wants to show off their projects, or see other people's builds!

04. Egham Raspberry jam

📅 Sunday 14 July
📍 Gartner UK HQ, Staines-upon-Thames, UK
▶ magpi.cc/WjFgSu

An opportunity to share what you have built, as well as the chance to learn from others.

05. Seattle Raspberry Jam

📅 Wednesday 17 July
📍 Bellevue Library, Bellevue, WA, USA
▶ magpi.cc/NvsBzz

Come and participate in the monthly project, share your knowledge, or show a project you've created.

06. Milton Keynes Raspberry Jam

📅 Saturday 20 July
📍 The National Museum of Computing, Bletchley, UK
▶ magpi.cc/hJEwZS

Meet other Raspberry Pi fans at the historic Bletchley Park, and make things over the summer break.

07. Preston Raspberry Jam

📅 Saturday 20 July
📍 Fulwood Library, Preston, UK
▶ magpi.cc/UoMSpi

A Saturday morning digital making event, with children, adults, and families, from beginners to experts.

08. York Pi Jam & Moonhack

📅 Saturday 20 July
📍 Acomb Explore Library, York, UK
▶ magpi.cc/JybjiC

A Jam for beginners, and also people who know a bit about Raspberry Pi computers!

FULL CALENDAR

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

rpf.io/jam
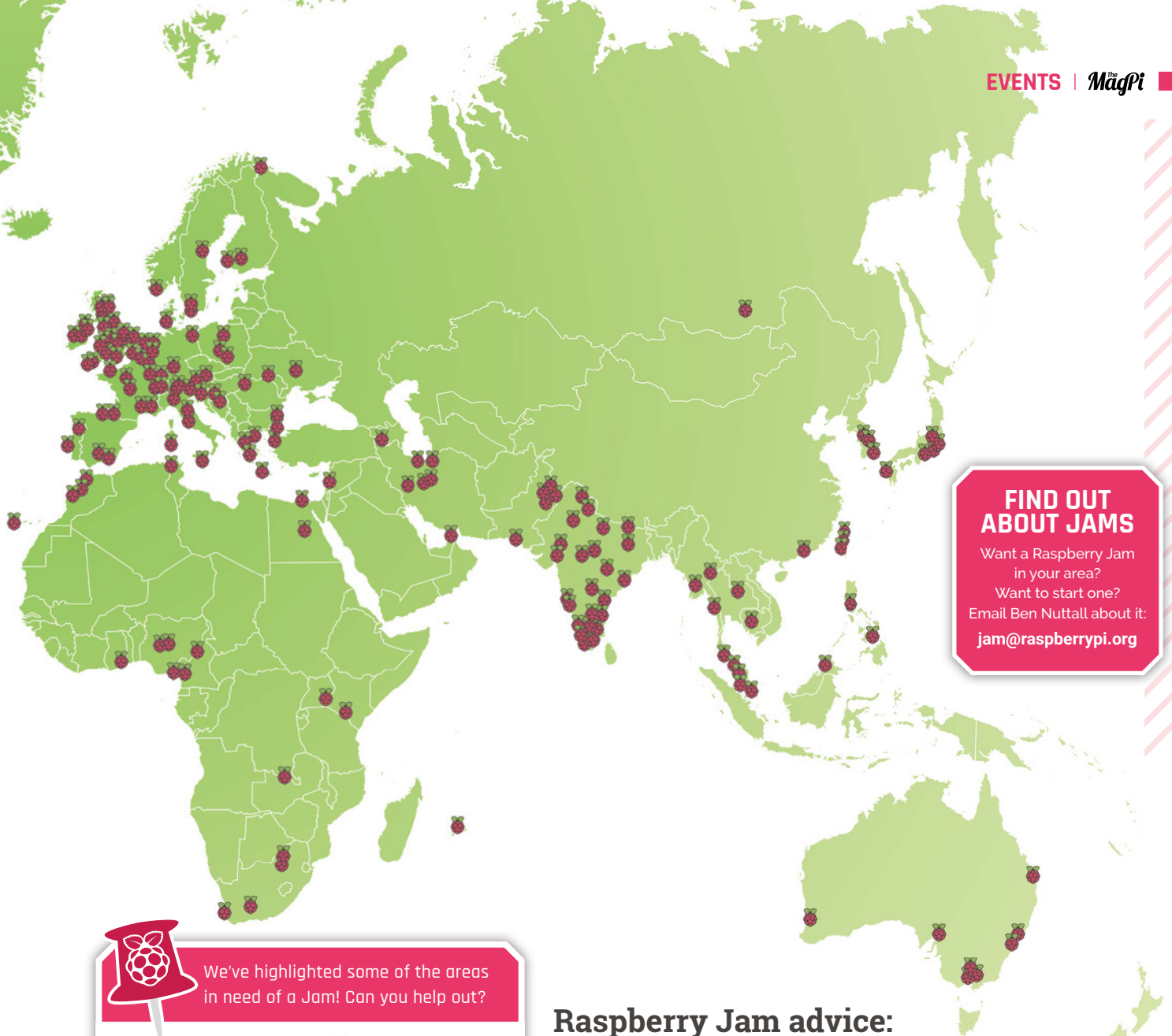
FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:

jam@raspberrypi.org

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

08

07

03

06

02

04

01
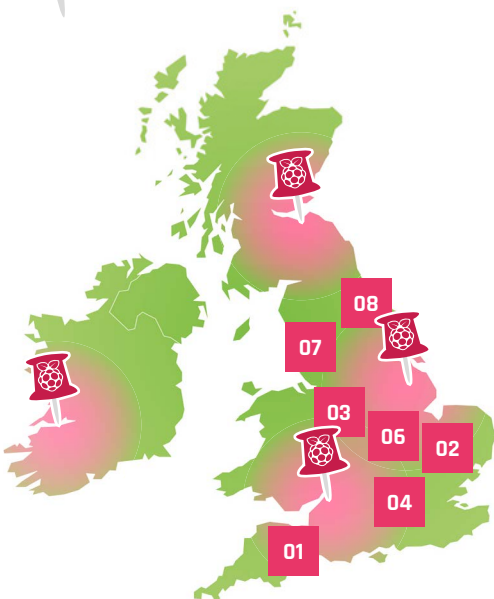
Raspberry Jam advice:
Catering

" took inspiration from other tech events I attended, and simply ordered pizza on the day of the event. Using money collected from donations and sponsorship, I would order as many pizzas as I thought we'd need, ensuring we had veggie options, and I'd tweet the order progress URL so people could follow along. While we waited for the pizza, I'd ask a volunteer to nip to the shop to buy a few large bottles of soft drink."

Ben Nuttall – Manchester 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: **magpi.cc/2q9DHfQ**
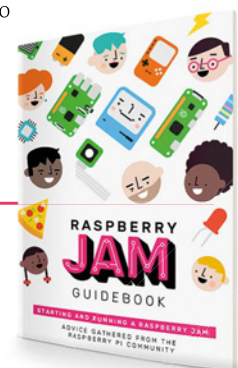
Your
Letters
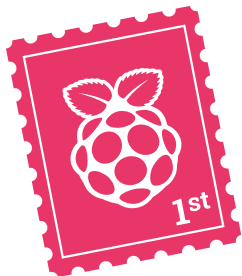
Subscription description

Can you explain what is in the gift pack for the initial subscription for the international option of £90? And what the delivery cost is for this option?

Which issues do you have in your back order that have hardware kits included in it? How can I order add-on kits such as Google Voice?

Joni via email

The twelve-month subscription, comes with a Raspberry Pi Zero W, along with a case and adapter cables. Postage is included in all versions of the twelve-month subscription, so there are no extra costs!

We no longer have any issues left with hardware kits included – they always sold quickly! Mouser stocks all Google's AIY Voice and AIY Vision kits, along with the the newer, Coral kits: **mouser.com**.

Contact us!

> Twitter **@TheMagPi**
> Facebook **magpi.cc/facebook**
> Email **magpi@raspberrypi.org**
> Online **raspberrypi.org/forums**
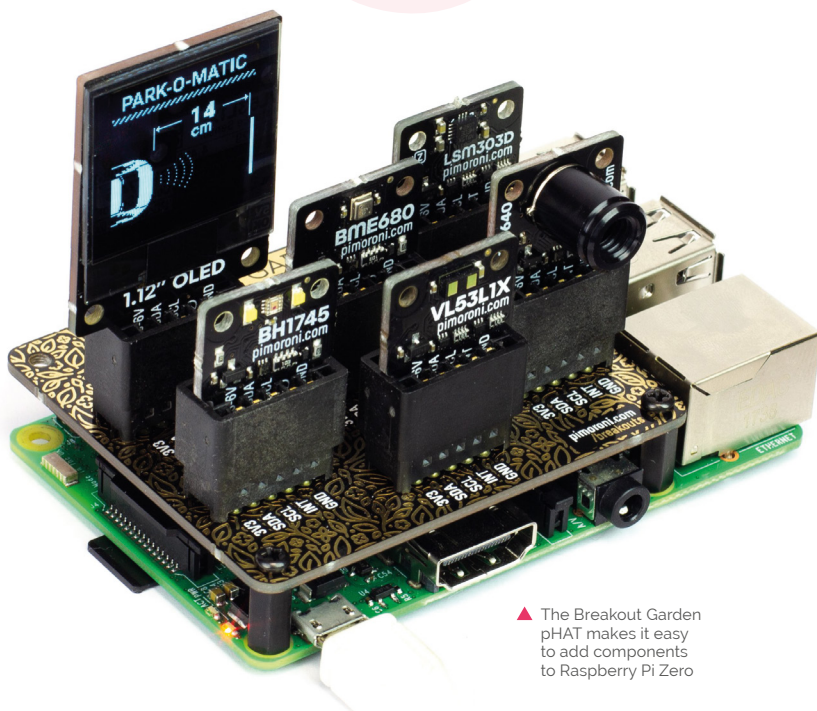
▲ The Breakout Garden pHAT makes it easy to add components to Raspberry Pi Zero

Eye squared see

Please can someone explain I²C and SPI? I want to attach an air quality monitor, and a set of LEDs to show the air quality state, to a Raspberry Pi Zero W. What do these terms mean, and how can I use devices with these two interfaces with Raspberry Pi Zero? I do have a Breakout Garden pHAT.

Anne via email

They're different ways for peripherals, components, and other devices you connect to the GPIO to communicate with the Raspberry Pi. Usually, it's a matter of adding the correct bit of code to your program to make sure you can read or control the thing you've connected to the Raspberry Pi, as well as connecting it to the correct pins.

Some of the pins on the Raspberry Pi's GPIO are reserved for SPI or I²C, so it's best to use those in conjunction with them.

As for the Breakout Garden pHAT, you'll be able to use the specific plug-in sensor with it without too much issue.

Cover Pi

I remember when you put out issue 40 with the Raspberry Pi Zero on it, and I was wondering if this is something you're ever likely to do in the future?

Sayem via Facebook

As you can see, we didn't include the Raspberry Pi 4 on the front of this issue of the magazine. Sorry! One of the reasons we were able to do it with Raspberry Pi Zero was because of its size – a full-sized Raspberry Pi board would make it extra-tricky.

Frankly though, we don't know. We're open to doing it again. Make sure you subscribe to ensure you get any special editions when they come out: **magpi.cc/subscribe**.

Backing up

I recently got a new Raspberry Pi and was wondering how to back up the software and documents on my SD card for transferring to my new Raspberry Pi. If it matters, I upgraded from a Raspberry Pi 2 to a Raspberry Pi 3B+.

Charis via Twitter

If you've kept your SD card up-to-date with dist-upgrades, you may be able to slot your SD card into your Raspberry Pi 3B+ and just boot it straight up. This doesn't always work, though, so you may have to install the software you need again.
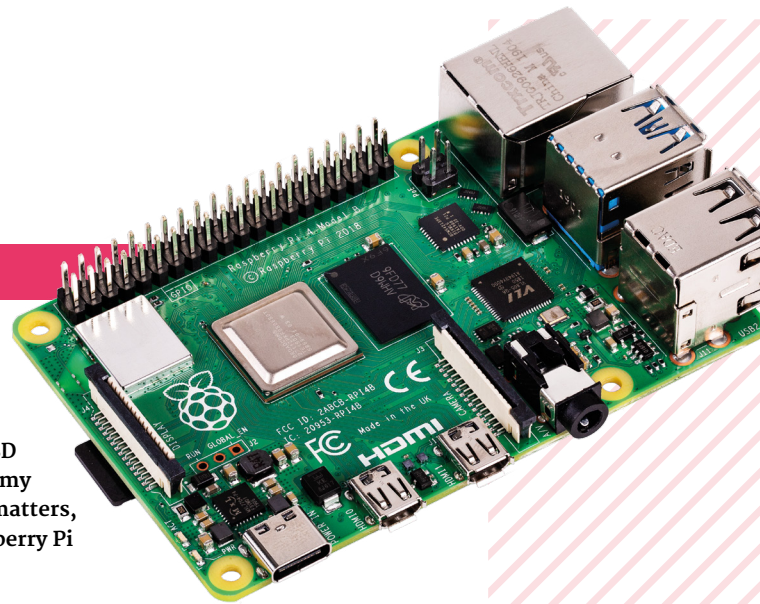
However, you can easily transfer documents over manually using a USB stick or network folders. It might be an idea to create an image from your original SD card in case of issues in the future – then you can go back and see what you may have done differently. M

WIN One of ten Raspberry Pi 4 Kits

Raspberry Pi 4 is here, and we're giving readers a chance to win one of ten of these amazing next-gen Raspberry Pi boards! Time to upgrade some projects…

Kit includes Raspberry Pi 4, official power supply, and a micro HDMI cable!

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

Terms & Conditions

Competition opens on **26 June 2019** and closes on **25 July 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.

THE
RASPBERRY PI
MANUAL

Your official guide to using the world's best computer
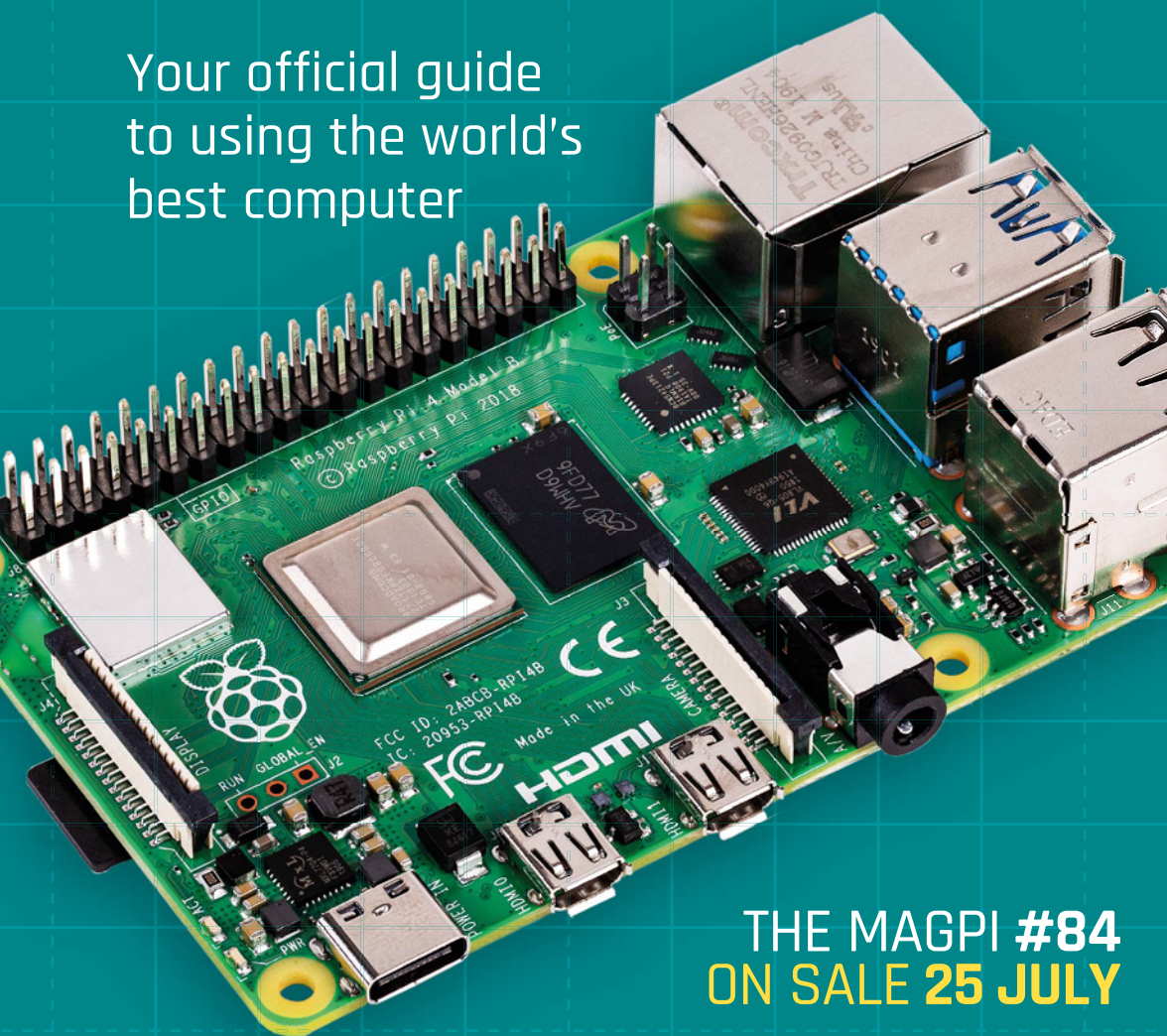
THE MAGPI #84
ON SALE 25 JULY

Plus!

Build a budget robot

Experiment with CircuitPython

Build the best summer projects

Play a game of Go against TensorFlow

DON'T MISS OUT!
magpi.cc/subscribe

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

How to make it

Taking things apart led to a career building incredible things.

By **Gordon Hollingworth**

I started engineering at age seven when my dad bought me a soldering iron to take a TV apart. Mostly, I learnt about how hot things burnt your fingers, and how resistors and capacitors can be levered out of the board effectively.

But these first experiments (the TV never worked afterwards) were indicative of my approach to all

> " I tend to think of everything – cycling, making companies, or managing people – as an engineering exercise "

engineering: take it apart and try to understand what it does, then use that understanding to develop new things.

The first one

At the age of ten I received my first computer, the Sinclair Spectrum 48K. My dad said I wasn't allowed to play games on it, so I had to learn to write them myself. My first game was a version of *Mole Attack*, which used crocodiles instead (because I found the graphics for those in a magazine).

My second computer was a BBC Micro Model B, and it was here that I tried to really understand how it worked: poring over the schematics in the advanced user's guide and having to fix the printer port because I'd shorted out the GPIOs. I left school at 16 with four GCSEs, did an apprenticeship for the MOD, and went to college. It was here I

realised how much I loved working on interesting things and worked out that it was worthwhile working hard (unlike at school.)

It's all just maths

After my apprenticeship, I went to the University of York and did an MEng in Electronics. Here I learnt about how everything is really just maths in the end, and that I really enjoyed really hard maths, finally ending up with a PhD (which was applying genetic algorithms to

electronics to make circuits that fixed themselves).

Most recently, I've graduated from the University of Cambridge Judge Business School with an executive MBA. Here, I have learnt about things other than engineering. I learnt about how companies work, how they interact, and how we can make companies that create value. Also, I've learnt a great deal about the people I was working with.

One thing I've found is that I tend to think of everything – cycling, making companies, managing people – as an engineering exercise: taking things apart, understanding how they work, then using this understanding to make things better. This process is one of the things I really enjoy, and something I get to do on a daily basis at Raspberry Pi. So, kids, if you think you're never going to make it because your GCSEs aren't quite good enough, just remember, Gordon did it… **M**
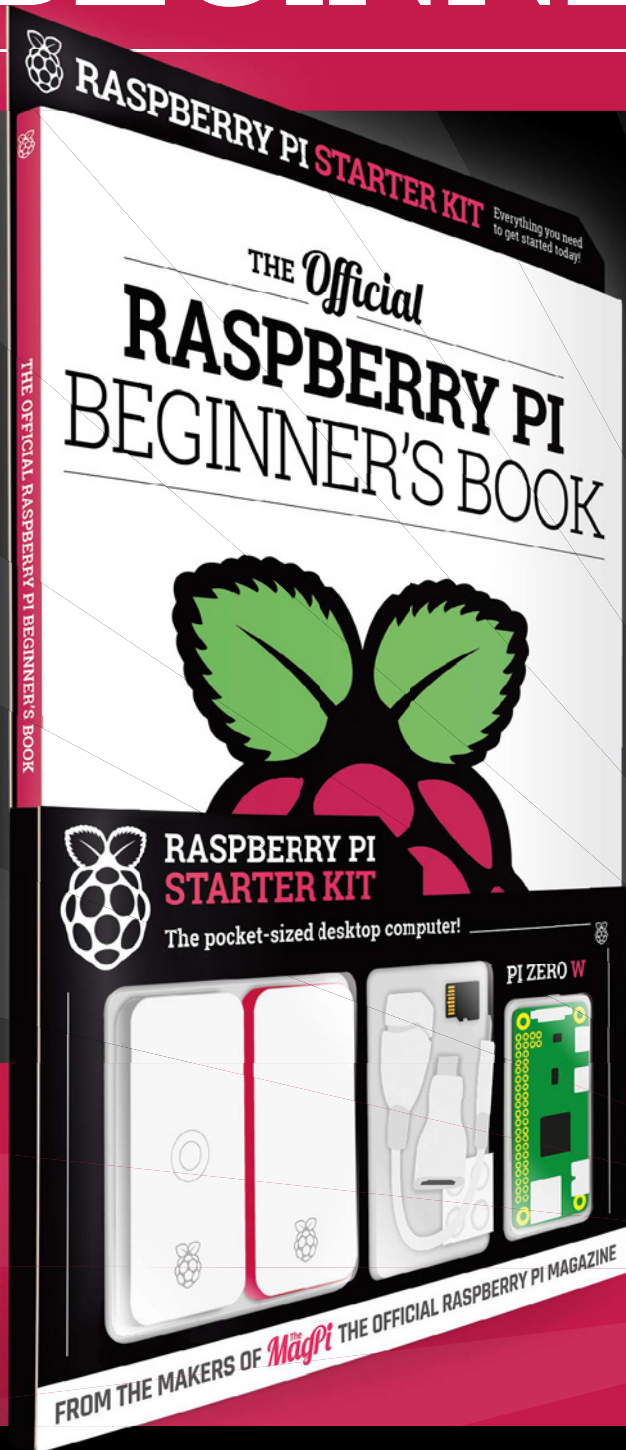
AUTHOR

Gordon Hollingworth

Director of Engineering at Raspberry Pi. Hacker, dad, cyclist, mountain biker.

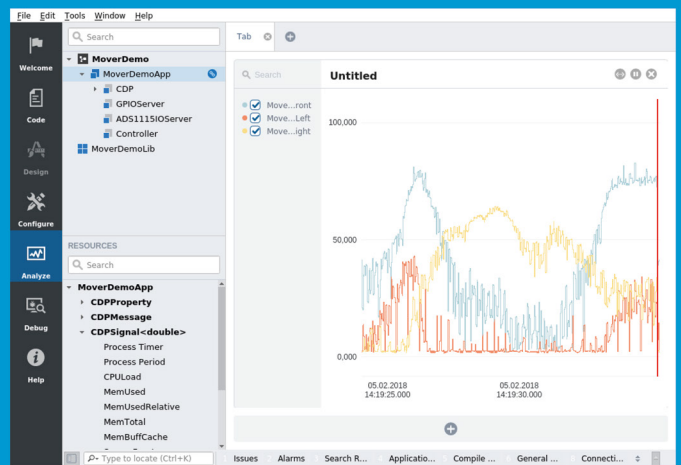**@gsholling**
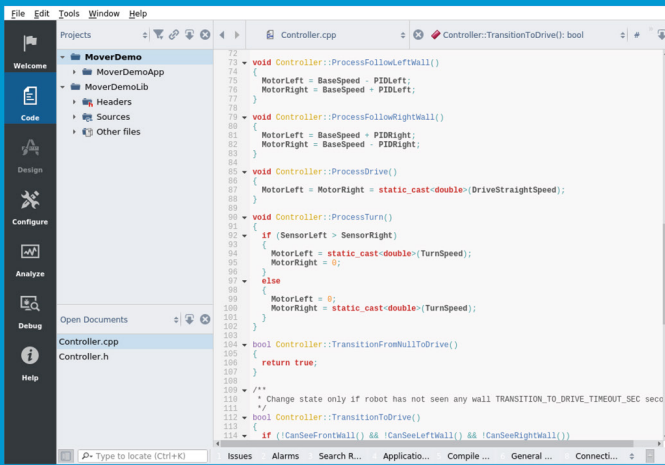
Code
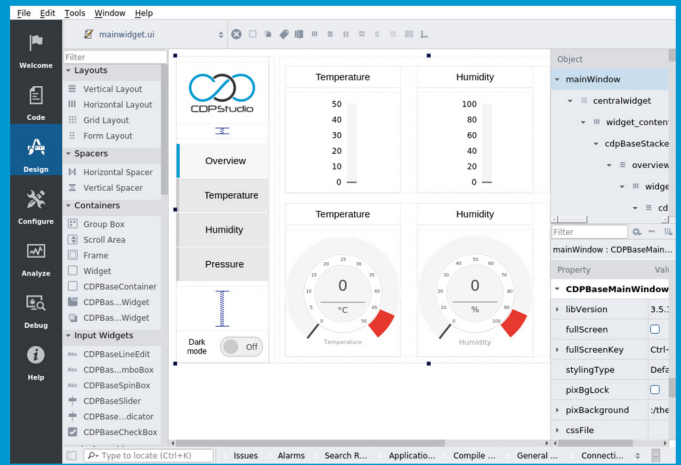
Design
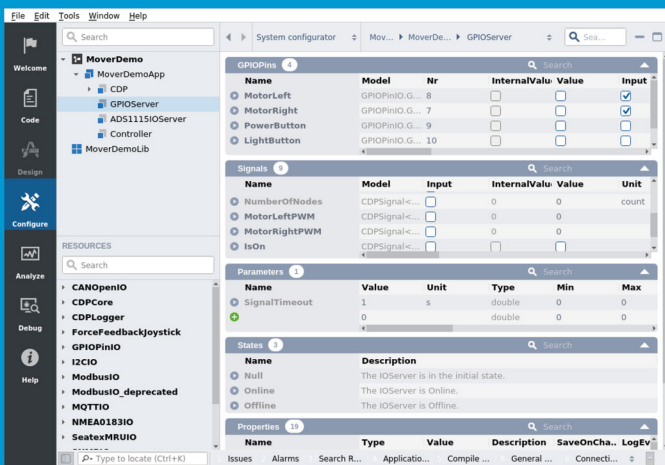
Configure

Analyze

Now free for home projects
A professional control system development tool

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

Free download on www.cdpstudio.com

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

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

CDPStudio