# The MagPi

Issue 93 | May 2020 | magpi.cc | The official Raspberry Pi magazine

## HOW TO

# WORK FROM HOME

## WITH RASPBERRY PI

QuickStart Guide Inside!

Reviewed: ZX Spectrum Next Accelerated

Learn Java & electronics

Explore the Internet of Things

**Work online** with your team

**Set up** video chat
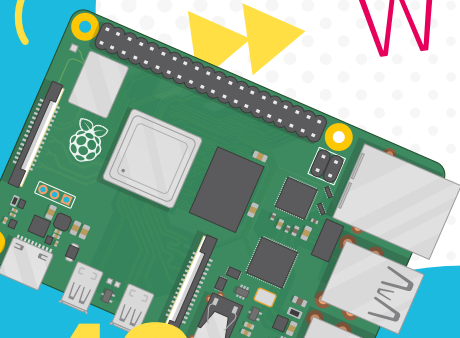
**Free office** software

# 43 PAGES OF PROJECTS & TUTORIALS

# WELCOME
## to *The MagPi* 93

**S**o it turns out a lot of people are now working at home with Raspberry Pi. Ever since Raspberry Pi 4 arrived, our favourite computer has become quite the reliable desktop workhorse.

Gareth takes us through How to Work from Home with Raspberry Pi (page 28). It's a great guide to some of the software and services that you can use from home with Raspberry Pi.

If you're new to Raspberry Pi, we have a couple of absolutely fantastic tutorials for you. Our QuickStart Guide (page 34) has been updated to work with the new Raspberry Pi Imager, and Rosie has written up a cracking tutorial that turns a Sense HAT into an LED Rainbow with a heart and thank-you message. It's perfect for putting in your window.

If you're after something a little more advanced, then PJ shows us how to Create Modules for your Magic Mirror (page 46) and Rob has a collection of amazing ideas in the At Home with the Internet of Things feature (page 70). Plus, there's an amazing range of projects, makes, and the best community builds.

I hope you enjoy this issue. It's not all work, and not all play. It's a lovely mixture of both.

**Lucy Hattersley** Editor

**EDITOR**

**Lucy Hattersley**

Lucy is editor of *The MagPi* magazine. She spends her mornings with Joe Wicks and the evenings with Joe Exotic.

**magpi.cc**

GET A **RASPBERRY ZERO W KIT**
WITH A SUBSCRIPTION!
PAGE **26**

# Contents

> Issue 93 > May 2020

## Cover Feature

## Regulars

## Project Showcases

**28**



Hot Spotter



RIoT Brick

Get started with the HQ Camera
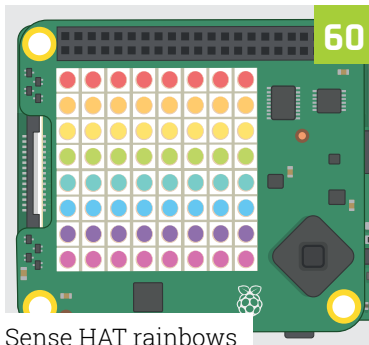

Sense HAT rainbows
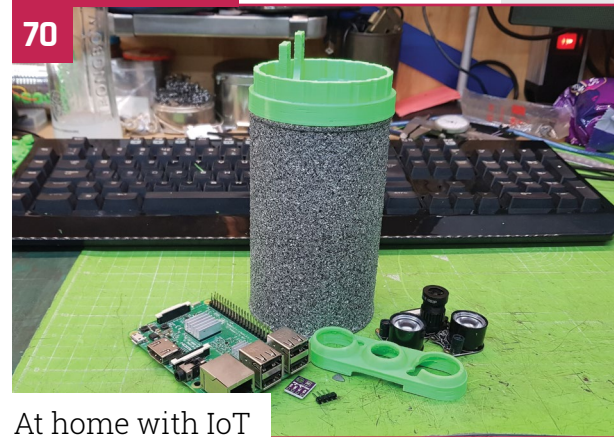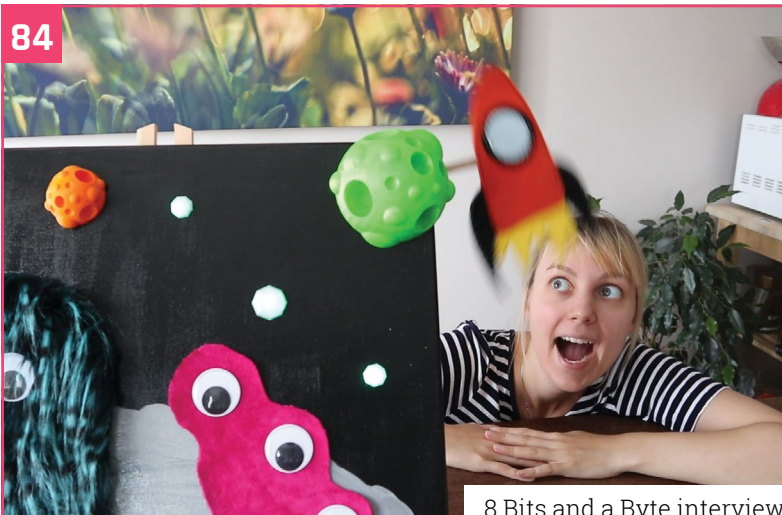

ZX Spectrum Next


8 Bits and a Byte interview


At home with IoT

**WIN** 1 OF 5 HQ CAMERAS **95**

# Digital Making at Home

Non-technical introduction to coding embraced by families thirsty for creative projects. By **Rosie Hattersley**

▲ Digital Making at Home features instructor-led tutorials

▶ If they don't have a computer available, learners can use a tablet

**R**aspberry Pi has had an enthusiastic response to the Digital Making at Home programme (**magpi.cc/DMAH**) which it launched in early April.

Thousands of people have started engaging with the Digital Making at Home content. They've also had kids sharing their projects (**magpi.cc/DMAHprojects**).

Digital Making at Home is for kids who want to get into making things with technology and need a few pointers. If you or your friends are looking for ideas of how to entertain youngsters or simply want new project ideas, the Raspberry Pi project portal (**magpi.cc/projects**) comes highly recommended.

Like all Raspberry Pi projects, the tutorials are both step-by-step and free. Videos explain all about coding platforms such as Scratch and Python, helping kids to quickly catch up.

## Young coders rule!

For families with no prior digital coding or making experience, Raspberry Pi staff members explain how they first introduced their own
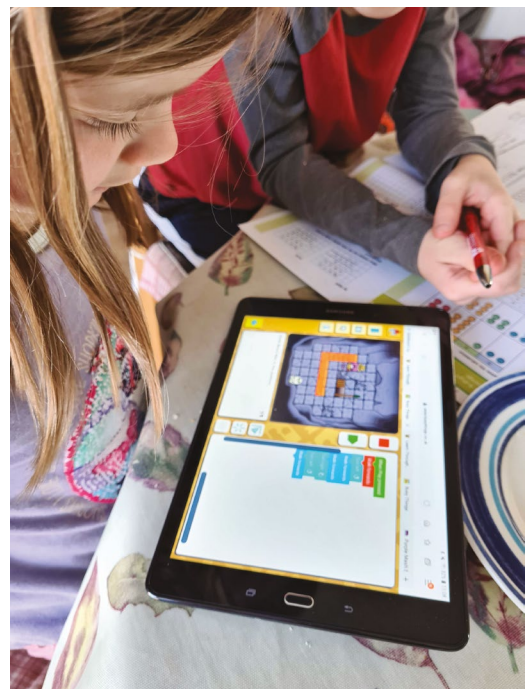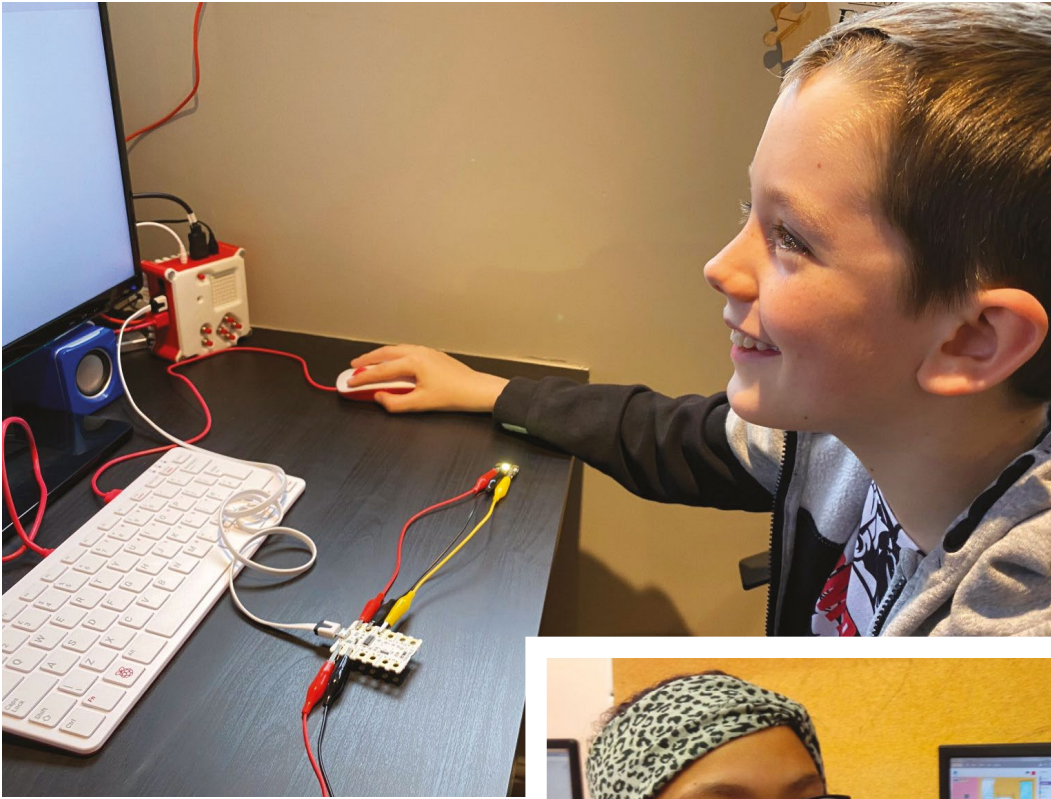
children to coding. Their initially tentative kids soon became confident, independent learners.

Children can use the game-making and coding skills they've learned to make their own versions based on their own interests. The Raspberry Pi Foundation's Senior Learning Manager, Marc Scott, related how his son used his new-found coding skills to create a random karate moves generator which helped him prepare for a karate exam.



> " Children can use the game-making and coding skills they've learned to make their own versions "

The projects, designed for all levels of experience, are self-contained and can be completed without much need for parental input, although many people tell of how much they enjoyed working through projects alongside their kids. They can be easily adapted to be more sensory, for example, so they suit learners with autism.

Marc cautions that before kids share their work online, they're prompted to consider what information they should be sharing about themselves. All schools now teach e-safety, so running through a checklist should be second nature.

### Video tutorials

There is a Digital Making at Home blog (**magpi.cc/athome**) which sees makers share the progress of their builds, troubleshoot issues that arise, and discuss ways to express their creativity. New instructor-led video posts are shared each week enabling viewers to code along with others. Subjects include making games and storytelling with code. Parents and carers can also sign up for tailored age-specific content via email sent bi-weekly. People can sign up at **rpf.io/parents-signup**. **M**
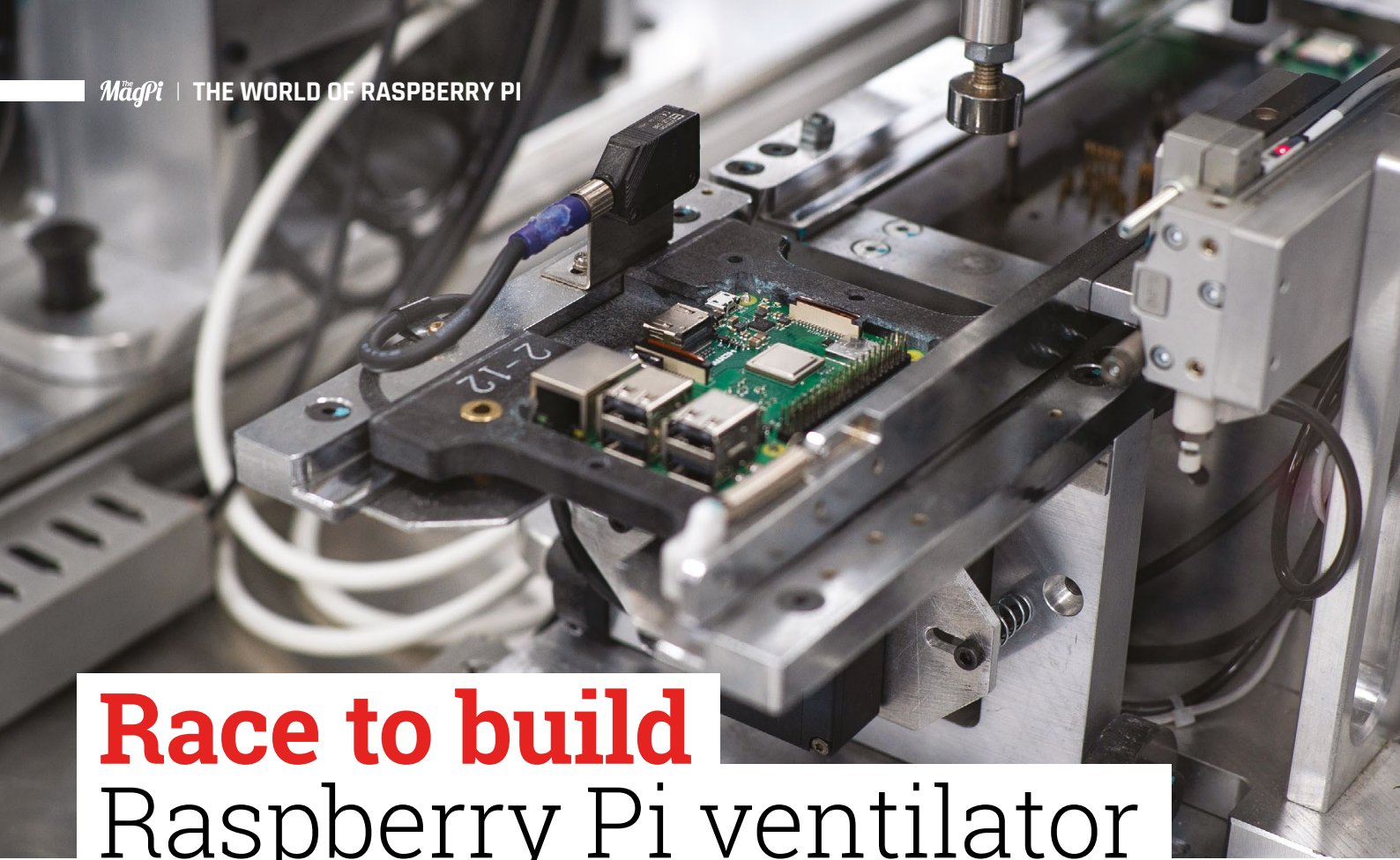
# Race to build
# Raspberry Pi ventilator

Raspberry Pi is being tested at the heart of ventilator equipment,
Eben Upton explains to **Rosie Hattersley**

▲ Raspberry Pi is seeing
incredible demand
for its products

**O**rganisations and individuals around the globe are racing to build a ventilator with Raspberry Pi.

One example uses the $5 Raspberry Pi Zero to run the user interface and direct an Arduino to open and close valves. This impressively low-cost ventilator is being trialled in hospitals in Colombia. It was put together based on details sketched out by robotics engineer Marco Mascorro, and uses widely available parts from plumbing and car part suppliers.

> ❝ People have been ordering 100 to test, then asking for 10,000 the next week ❞

Healthcare workers immediately responded to Mascorro's open-source Raspberry Pi-based design, with other countries likely to be making their own versions. Raspberry Pi founder and CEO Eben Upton told us that Colombia is the not only country making use of Raspberry Pi Zero, but says he does not believe it is used in any of the novel ventilators currently being trialled in the UK. He's sanguine about it, though: "We'd much rather it was the right design, rather than our product for the sake of it," he says.

Volume lead times on other industrial single-board computers are typically five to six months: too long to be of use during the anticipated surge in COVID-19 cases. However, Raspberry Pi builds to stock rather than building to order, and so has boards on hand. "People have been ordering 100 to test, then asking for 10,000 the next week, and every week after that," reveals Eben.

## A reliable source

Supplies of the British-built computer are not affected by Chinese setbacks to manufacturing. Eben says, "There have of course been some difficulties in the supply chain, but components are flowing fairly freely, and assembly is going well. Staff at the Sony factory in Pencoed in Wales (where Raspberry Pi Zero is assembled) have been working overtime and weekends to meet demand." Social distancing measures are in place for safe working.

Demand for Raspberry Pi is stronger than ever as people make use of unexpected free time to tinker, or home-schooling their kids by exploring physical computing projects. Mindful of both hobbyist and industrial demand, Eben told technology website Tom's Hardware, "Raspberry Pi plans to produce 250,000 units of Zero-line (Pi Zero / Zero W) products per month" from now on. 🅼

# remote.it

# Enable remote work with remote.itPi

Adding remote.it transforms a Raspberry Pi into a fully-featured Remote Access Appliance – no complex VPN nor exposed ports required. Your company's team members can easily connect to office computers, Network Attached Storage systems and other shared resources.
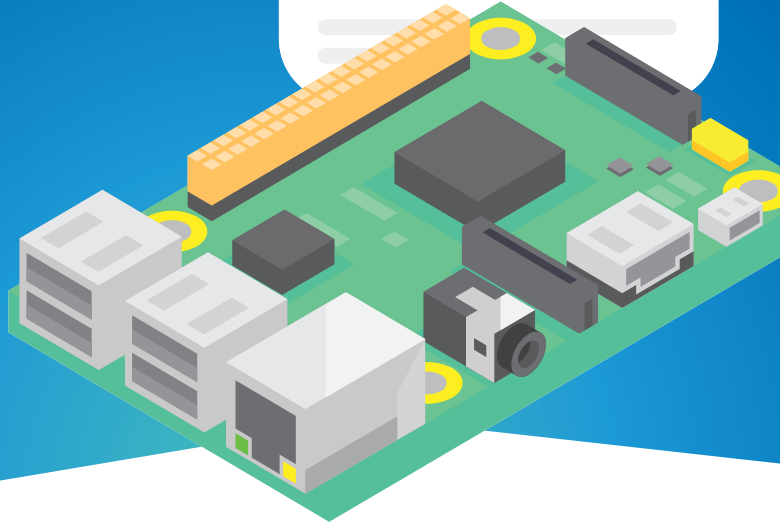
## Download the remote.itPi image to get started

The image and remote.it services are always free for non-commercial use. In support of businesses, all paid business class services are free until September 1, 2020 for new subscribers. Get the download at https://remote.it/downloads/#pi

## Easily set up secure access to any networked device

Access any device from anywhere – even over mobile networks (GC-NAT) – without opening ports and enable out-of-band management.

Any desktop computer or device using Windows, macOS or Linux

Easily create a gateway to a remote network with remote.itPi

Create a virtual private internet for your enterprise network

**Learn more at remote.it**

# Vineyard
## Kikushima

We've seen plenty of beer brewed with the help of Raspberry Pi, but now it's wine's time to shine. We paired **Rob Zwetsloot** with this full-bodied article

**MAKER**

**Kunio Kikushima**

An ex-employee of electronics manufacturers turned vine farmer, Kunio lives a modest lifestyle but wants to make great wine.

**W**e've covered several plant and garden automation projects in *The MagPi* before, and even a robot farm or two. However, we've not previously come across a vineyard with some IoT/automation abilities thanks to Raspberry Pi.

"We are now doing viniculture in Koshu city, Yamanashi Prefecture, and we aim to open a small winery in Katsunuma in the spring of this year," says Kunio Kikushima, owner of Vineyard Kikushima. "We also aim for eco-friendly wine without any agricultural chemicals where possible. We are now doing viniculture and vinification."

Kunio didn't have much experience with programming and developing, but a chance

> **"** We aim to open a small winery in Katsunuma in the spring of this year **"**

meeting with friend of *The MagPi*, Masafumi Ohta of the Japanese Raspberry Pi Users Group, enabled him to gain some knowledge



▲ The first prototype system set out in the field. Sensors hang out of the box to obtain an accurate reading

of Raspberry Pi and how it could be used to great effect in this instance.

### Solving a problem

Grapes can be prone to disease, especially in the relatively high temperatures and humidity of Vineyard Kikushima. While using agricultural chemicals and pesticides helps, they can alter the flavour of the wine, so Kunio needs a more exact method of applying them in the ideal weather conditions for maximum efficiency.

"I wanted to check the timing of high humidity [by] collecting temperature and humidity data

A Raspberry Pi monitors several sensors that Kunio can read at home

The whole system is solar-powered, with a car battery to store excess energy

貧農 IoT システム

Vineyard kiku shima

無料試飲

The electronics have some weather-proofing just in case of rain

## Quick **FACTS**

> The system is called Hinno IoT

> Similar products exist, but are very expensive

> Kunio had no prior coding skills

> He has four fields spread very far apart

> The use of chemicals is a big thing for small Japanese wineries

▲ A closer look at the electronics inside the sensor boxes

▶ The second box is near one of the electric fences used to keep animals out



automatically measured at regular intervals," he explains. "I want to check those data in real-time through the network, as the fields are scattered and far away from my office." This also means the setup requires solar and battery electric power.

> ❝ It is very easy to get Raspberry Pi at the store in Akihabara and online shops ❞

### Field tests

The system is currently in a trial phase, with one prototype being used for the field test, and another reserved for system development. The total cost is ¥30,000–¥40,000 [about £225–£300] including the solar power equipment. Adding 3G dongles to the system, the total expected cost will be ¥50,000–¥60,000 [about £375–£450].

Kunio hopes to add more sensors in the future and offer the system to other farmers in the region.

"It is very easy to get Raspberry Pi at the store in Akihabara and online shops," he says. "There are tons of various use cases I can see on the internet. I think I had the illusion that I could make [the vineyard-monitoring system] without any programming skills on Linux. But it is very fun for me to assemble by watching and imitating, and it could be applied to another use with the same platform if the automatic display and measurement of temperature and humidity in fields works well.

"I could make delicious wine with Raspberry Pi." M
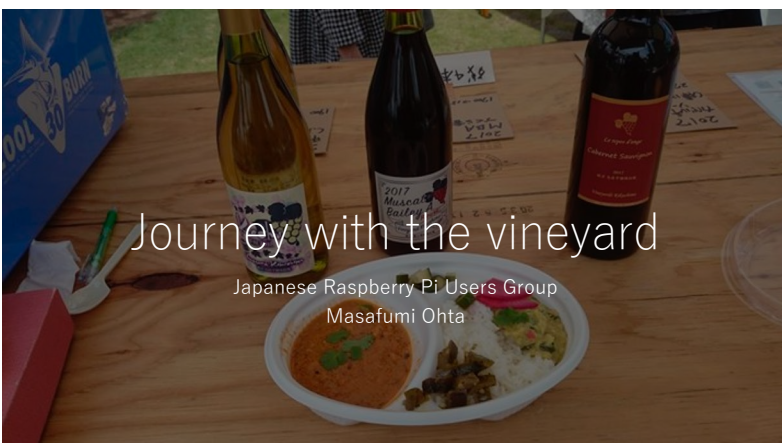
Journey with the vineyard
Japanese Raspberry Pi Users Group
Masafumi Ohta

## Raspberry Pi wine



**01** The grapes are monitored from the office using the Hinno IoT system to see what the atmospheric conditions are in the fields.



**02** A decision based on the monitoring tells Kunio whether or not to spray his crops. The fewer chemicals used, the better.



**03** The grapes are harvested and turned into small-batch wine. There are several types that are sold from Vineyard Kikushima.

# Hamster Feeder

A tiny Raspberry Pi Zero assisting a small rodent? What's not to love? **David Crookes** takes a look

**MAKER**

**Christopher Barnatt**

Christopher runs the ExplainingComputers YouTube channel, and also works as a futurist, keynote speaker, and freelance academic. He has produced over 60 videos featuring a Raspberry Pi.

**magpi.cc/excomp**

**C**hristopher Barnatt doesn't own a hamster. But that hasn't stopped him working on an automated feeding device to help anyone who does. His project allows a portion of food to be served at a suitable interval so that a hamster wouldn't gobble its whole supply at once. Chris reckons it could be recreated in just a few hours.

"I'd made a video about controlling servo motors using a Raspberry Pi and I wanted to follow that up with a simple, practical project that would put Raspberry Pi-controlled servos into action," he explains. To that end, the Hamster Feeder was created as a tutorial for his YouTube channel, ExplainingComputers. "It would just need a little refinement to be used to feed a real hamster," he readily admits.

## Air pods

Since his channel focuses on single-board computers, he decided to use a Raspberry Pi
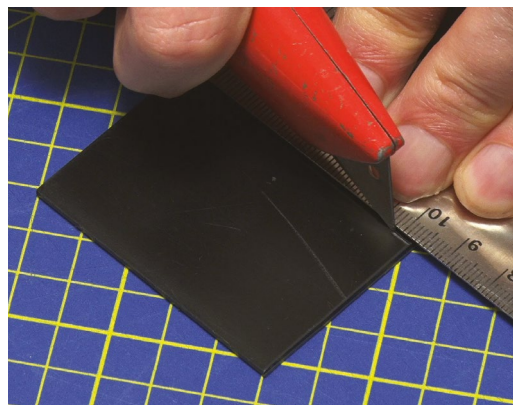


▲ Christopher made his own pods by cutting and pasting together five plasticard pieces for each

Zero running Raspbian for his project. He also handmade two little food pods, mounting a servo on the side of each one.

"The most difficult part of the project was progressing from the idea of 'let's use a servo as a catch to open a door at the base of a container', to actually having two of those little doors on two small containers suspended in the air," Chris says.
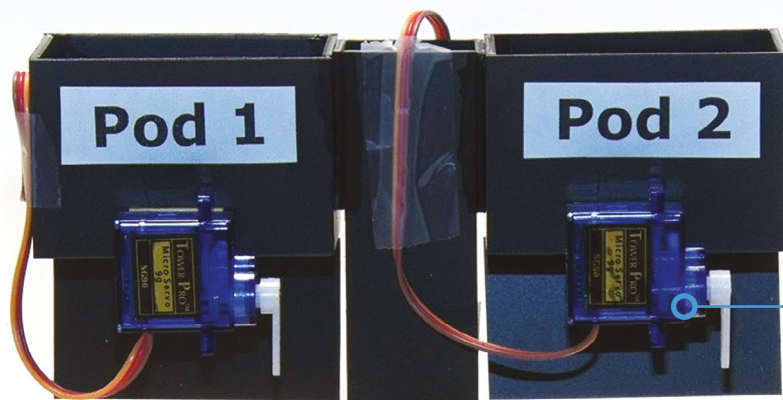
After the doors of the pods open and serve the nuts, the servos stop operating; this reduces wear and power drain

In a real-life situation, you'd keep Raspberry Pi Zero, servos, and wires out of reach of an inquisitive hamster

Portions of nuts are placed in the pods and, at a set time, they're allowed to drop to the floor

Pod 1

Pod 2
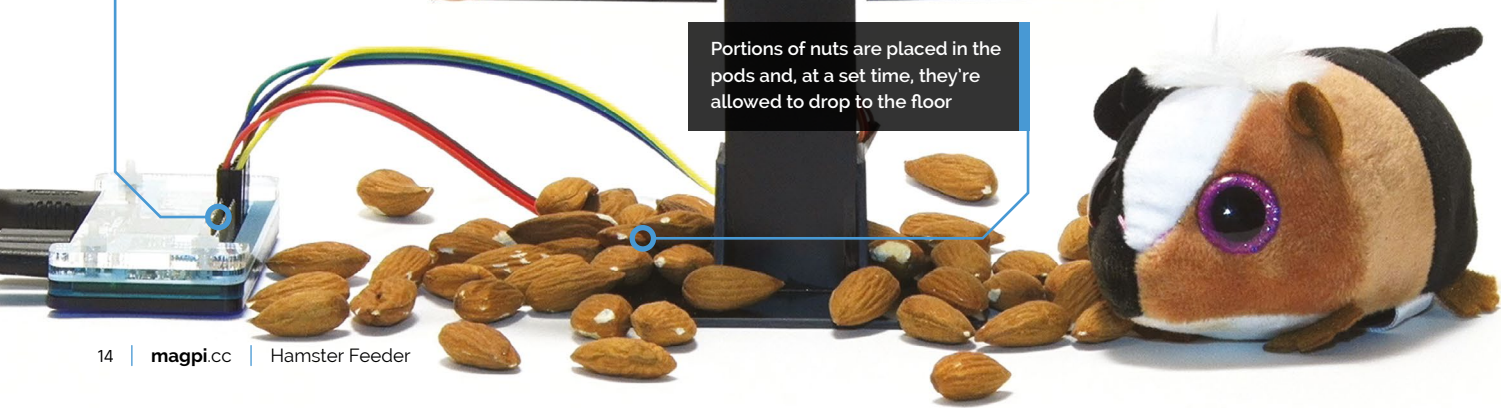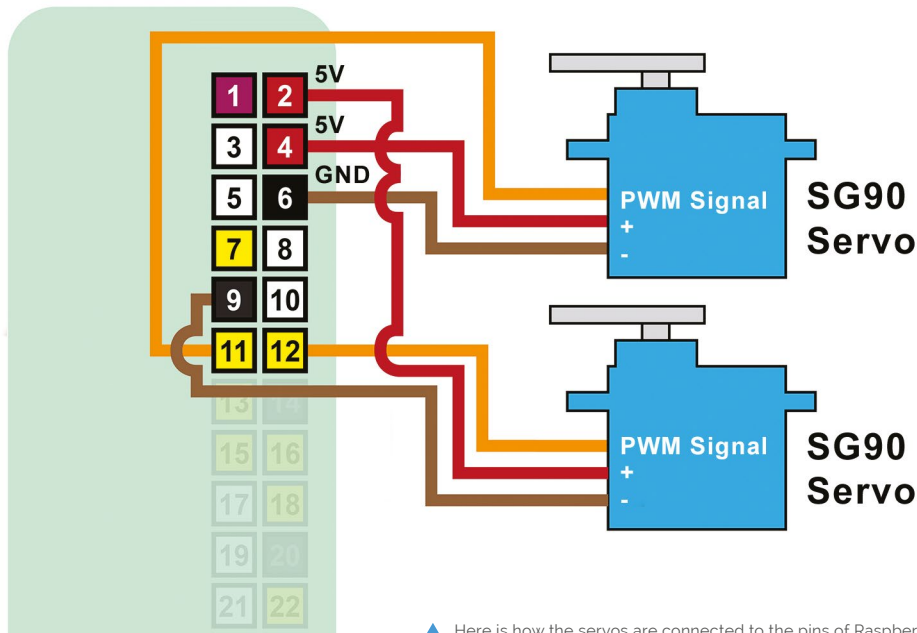
▲ Here is how the servos are connected to the pins of Raspberry Pi Zero

"I happened to have some plasticard sheet and solvent adhesive, but it was very much a situation of working out what I could design and build in a few hours, with the added complication of how to show the process in a video. I remember a very intense and frantic morning!"

## ❝ It would just need a little refinement to be used to feed a real hamster ❞

### Nutty coding

After connecting the servos to his Raspberry Pi Zero, Chris turned to Python and broke his code down into three sections. "The first loads in the GPIO, time, and datetime libraries, before setting up board numbering for the GPIO pins," he explains. "It then sets up pins 11 and 12 as GPIO outputs with software pulse-width modulation, or PWM, which is what is needed to control servos."

The second is dedicated to nut loading: "Here the user is invited to close (and hold shut) each pod bay door in turn, and press the **ENTER** key to allow the servo to move and function as a latch. To maintain a record of each door being closed, variables pod1 and pod2 are set to a value of 1 to indicate when their respective doors are shut."

Finally, there is a 'main while' loop – a control flow statement that can execute statements when a condition is true. "An if and an elif statement compare the current date and time with two hard-coded dates and times in order to check if a pod

door should be opened. They also see if a door is still shut by checking its pod1 or pod2 value, and setting this to 0 once a door is opened."

### Feeding frenzy

Of course, improvements could be made. Servos, Chris says, could swing the doors open and closed. "It would also need some chutes to direct the nuts into a hamster cage with the pods and servos entirely separated from a live animal outside of the cage," he continues. There's scope to regulate the number of nuts dispensed and to remove hard-coded dates and times in the code.

"I did think of using another servo to raise a flag saying 'lunchtime' every time a pod bay released some nuts," he adds. "This said, I'm not sure that most hamsters can read and which languages they are most familiar with." M
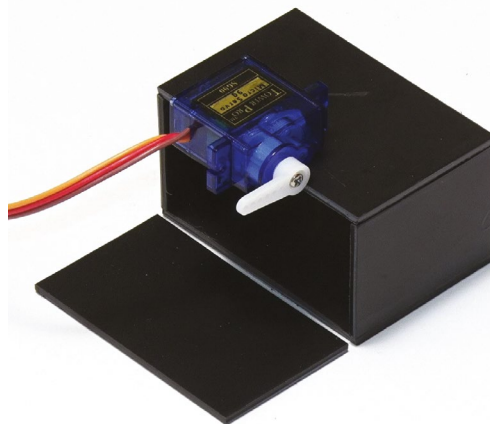


◀ As well as sticky-taping the lid to the rest of the box, double-sided tape is used to position the servo

# DropStitch

The joy of creating something wonderful – whether in code or another medium – can turn to despair without a means of rendering it. **Rosie Hattersley** applauds a new option

**MAKER**

## Margaret Low

Margaret is director of outreach at the University of Warwick's Manufacturing Group, making extensive use of Scratch and CAD in her efforts to widen participation in computing and technology.

**magpi.cc/ dropstitch**

▼ TurtleStitch uses a familiar block-based interface, so you can boost your coding skills while focusing on being creative

**C**reative crafting is all the rage and can be a fulfilling and mindful hobby that combines easily with technology. A workshop using one example, TurtleStitch, at last summer's Raspberry Pi-hosted Scratch Conference in Cambridge was packed. It's a popular visual means of using Snap! (**magpi.cc/snap**) and a great learning tool which will be made all the more usable thanks to the creation of DropStitch.

After being created in TurtleStitch (**turtlestitch.org**), designs are printed using a digital embroidery machine, such as the one Margaret Low from the University of Warwick brought along to the Scratch Conference.

Getting the designs on to her embroidery machine involved transferring them from a PC or smartphone to the device on a USB stick – a rather clunky option that meant the user had to be in the same location as the embroidery machine, or send it to someone to initiate the embroidery on their behalf. Margaret points out that frequently switching USB sticks can result in wear and tear on the embroidery machine USB port.

After discussing these issues with other TurtleStitch demonstrators, Margaret came up with the DropStitch project in order to solve this problem. "It enables devices that normally read files from a USB drive to receive files wirelessly using Bluetooth file transfer (or WiFi) – so
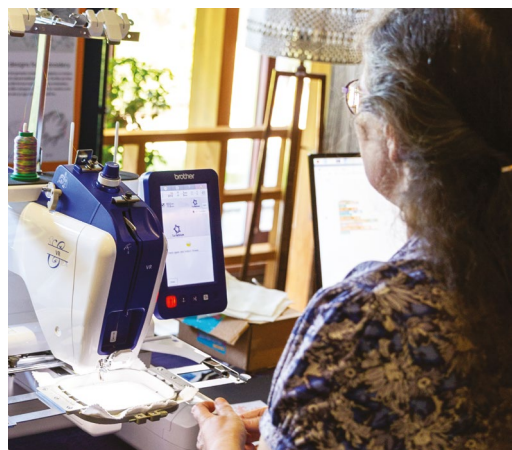


▲ Margaret demonstrated the DropStitch system at last summer's Scratch Conference in Cambridge

freeing them up from being tethered," she says. DropStitch works with TurtleStitch and other embroidery programs (those supported largely depend on the embroidery machine being used).

> ❝ The free stitch cards provided by TurtleStitch provide Snap! tutorials ❞

## Inspiring creative coding

Margaret uses the industrial embroidery machine in the university's engineering faculty's workshop, demonstrating TurtleStitch and coding to students and visitors, and potentially broadening the appeal of an under-represented STEM subject.

Because Margaret also regularly holds classes introducing schoolchildren to code using TurtleStitch, she knows only too well the issue of having to contend with a succession of USB sticks so that kids could print their designs. It's a time-consuming drag! However, she enthuses about the software's role in broadening the appeal of coding. Coding can be done on a tablet and laptop. The free stitch cards provided by TurtleStitch

▲ TurtleStitch designs use Snap!, a block-based programming language, to create some incredible geometric designs



Digital embroidery machines can stitch preloaded designs, but user-created or downloaded ones have to be loaded via USB

DropStitch uses a Raspberry Pi Zero W to add Bluetooth and wireless LAN to the embroidery machine

Once transferred to the embroidery machine, images to be stitched can be previewed and selected as usual from the touchscreen

provide Snap! tutorials that result in impressive geometric designs and lettering, so you can quickly go from being a beginner to producing impressive embroidery.

If you don't have access to an embroidery machine, you can send your completed design to someone who does – something that DropStitch makes appreciably easier.

## Smart thinking

Margaret explains: "The aim was to avoid some of the problems with USB memory sticks such as having sufficient USB memory sticks for the class, identifying files to be stitched, and the accidental overwrites of files. DropStitch is based on *The MagPi* article [by Dave Honess in issue 60], 'Make a Pi Zero W Smart USB flash drive' (**magpi.cc/usbflash**). DropStitch has some modifications [from the original] because Raspberry Pi Zero W and the embroidery machine can't both write to the same USB partition at the same time."

Essentially, it uses the embroidery machine's USB port to add either WiFi or Bluetooth connectivity and provides device discoverability and only needs a Raspberry Pi Zero W to do so. All the instructions to install and enable DropStitch are provided on the tool's webpages at **magpi.cc/dropstitch**.

Margaret is keen to help extend DropStitch's usefulness by making code available for others to customise: **magpi.cc/dropstitchcreate**. There's even a form on the DropStitch homepage to request support for a particular embroidery program. ◾

### Quick **FACTS**

> TurtleStitch uses Snap! a Scratch-inspired language

> The impressive embroidery tool has an international following

> TurtleStitch sees Snap!'s pen as an embroidery needle

> Margaret uses Scratch in school outreach programmes

> Warwick University provides lots of free TurtleStitch tutorials: **magpi.cc/ warwickTS**



▲ TurtleStitch designs created by Warwick University can be embroidered onto T-shirts or bags or used to make homewares. *Image: Helen Luckhurst*

# Hot Spotter

An autonomous drone that can detect and report smouldering fires. **Phil King** seeks it out

**MAKER**

## Jason McDonald

Jason has a degree in Applied Physics and numerous 'nanodegrees' for machine learning and AI. He is currently enrolled at the University of Montana in the welding technology program.
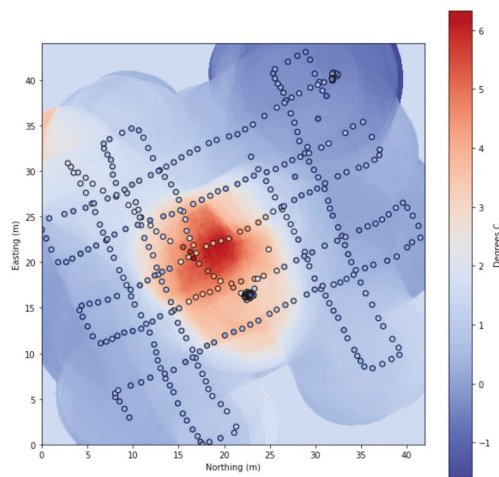
**magpi.cc/hotspotter**

⚠️

**Alert!**

Drone use is regulated in some territories (including the UK and USA).

**magpi.cc/CAAdrones**
**magpi.cc/FAAdrones**

**W**hile commercial operations use fleets of expensive drones to track the spread of wildfires, it's possible for a previously extinguished fire to relight. It's in the detection of these smouldering patches where a low-cost drone such as Jason McDonald's Hot Spotter could come in useful.

"I wanted to create something practical with the [NXP HoverGames drone] kit that filled a niche not already being targeted by the commercial drone industry and/or lower the cost of a current application," he reveals.

Made from off-the-shelf hardware, including a Raspberry Pi 3A+, the Hot Spotter can survey an area to create a heat map. "As the drone flies over an area, it records average temperatures of spots on the ground," explains Jason. "The size of the spot depends on how high the drone is flying, in the same way the spot size from a flashlight depends on how far away it is from a surface."

The software creates an imaginary grid of points, spaced about one metre apart, to be used for the heat map. When the drone takes a temperature reading of a spot, it's recorded for all the points contained within it. Then, when the heat map is generated, all the readings for a given point are averaged together. "This method helps to localise heat sources and distinguish between large warm regions versus hot spots."



▶ Preparing for take-off, the Hot Spotter drone is pictured in front of a charcoal fire used for testing purposes



▲ An example heat map generated by the drone's Raspberry Pi as it flew a cross-hatch pattern above the survey area

As the drone flies a cross-hatch pattern to survey the area, its on-board GPS is used to determine its precise position while a lidar module measures the distance to the ground. This results in more accurate heat mapping than when estimating the height, as well as opening up the possibility of terrain following and obstacle avoidance.

## Altitude testing

"For my tests, the drone flew about 20 metres above the ground," says Jason. "I would like to test at other altitudes. Since the size of the spot being measured increases with altitude, the drone can cover more area from higher altitudes at the expense of heat map fidelity. Ideally I would like to create a quick map from a high altitude followed by a higher fidelity map at a lower altitude for regions that look like potential hot spots."

The on-board Raspberry Pi 3A+ communicates with the drone's flight controller (using the MAVLink protocol) to receive information such its GPS position and orientation. "Raspberry Pi handles all of the calculations necessary to generate a heat map from sensor data recorded by the drone and from sensors attached directly to the GPIO using an I²C hub," says Jason. "Raspberry Pi can also command any autonomous

Connected to a Raspberry Pi 3A+, the thermal sensor on the drone's underside detects the ground temperature

A lidar sensor is used to measure the distance to the ground below for increased accuracy of the heat map

LiPo battery power is wired via the drone's power distribution board and a UBEC

functions such as setting a waypoint and returning to its launch point."

It sends the data in real-time – using a wireless serial connection via a 500 mW transmitter – to a ground station laptop via another Raspberry Pi used as a wireless hotspot for a laptop.

### Crash landing

It took Jason three months to build and program the Hot Spotter. However, after numerous successful test flights, disaster struck when the drone crash-landed due to a malfunctioning rotor. "It was a bummer to say the least," he says. "I have rebuilt the drone. Better than it was before. Better… stronger… faster."

> As the drone flies over an area, it records average temperatures of spots on the ground

While he thinks the drone's practicality is solid from a cost standpoint, its practicality 'in the wild' remains to be seen. "I would like to do further testing that includes real-world scenarios. There is a fire science laboratory (**firelab.org**) near me and if they find it useful, the project might have a future there." M

# DIY MIDI Door

Using a door as a musical instrument? **Nicola King** gets a handle on a different kind of harmony

**MAKER**

**Floyd Steinberg**

Freelance programmer, synthesizer enthusiast, nerd, hobby musician. Used Linux since the mid-nineties – has a kind of love/hate relationship with that OS.

magpi.cc/mididoor

**W**hen Floyd Steinberg received a Raspberry Pi electronics kit from his wife as a birthday gift, his first project idea involved turning one of their home's internal doors into a MIDI controller. "I finally had some time to read the manual and wanted to make something humorous to show on my YouTube channel," he recalls. "The DIY kit had some potentiometers and an 8-bit ADC [analogue to digital converter], so that seemed to be a good starting point."

The kit's manual included an example circuit for controlling the brightness of an LED by turning a potentiometer knob. So Floyd took that basic concept and added a switch and a MIDI interface to the setup to enable it to control the pitch of musical notes played.

### An a-door-able idea

After testing the circuit design with a breadboard setup, Floyd glued a potentiometer to the hinge of a door so that opening the latter to different degrees alters its voltage. Read by an ADC connected to a Raspberry Pi, this then determines the note played by a synthesizer whenever the door handle is pushed down to close a circuit.

> ❝ I wanted to make something humorous to show on my YouTube channel ❞

"A C program on Raspberry Pi reads the values picked up by the converter," he explains. "These values range between 0 and 256. On a synthesizer, all notes are numbered from 0 (that's C0, the lowest note) to 127 (G9, the highest note). So, my program just divides the converter's input by a certain factor and rounds those numbers and sends them [via MIDI] to the synthesizer. Pushing the door handle closes an electrical circuit which
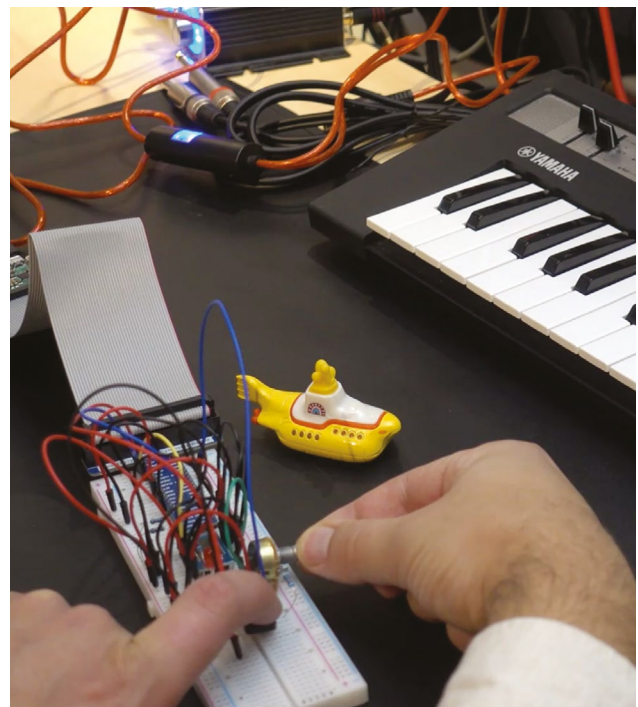


▲ Turning the potentiometer to test the note-playing system with a synth before mounting it on the door

makes Raspberry Pi send a 'note on' command to the synth, and releasing the handle sends the 'note off' command."

It's an ingenious use of Raspberry Pi, but not every step of the project was straightforward. For example, Floyd found that he had to spend some time finding the correct factor by which the numbers read by the ADC had to be divided. "Making the [door] movements too small wasn't an option, because the whole system isn't very precise. In the end, I defined some value ranges for the three or four notes which are used in the song [*Break on Through (to the Other Side)* by The Doors]."

In addition, some mechanical issues popped up, such as how to design a lever to turn the potentiometer's knob, as it had to be very tight so
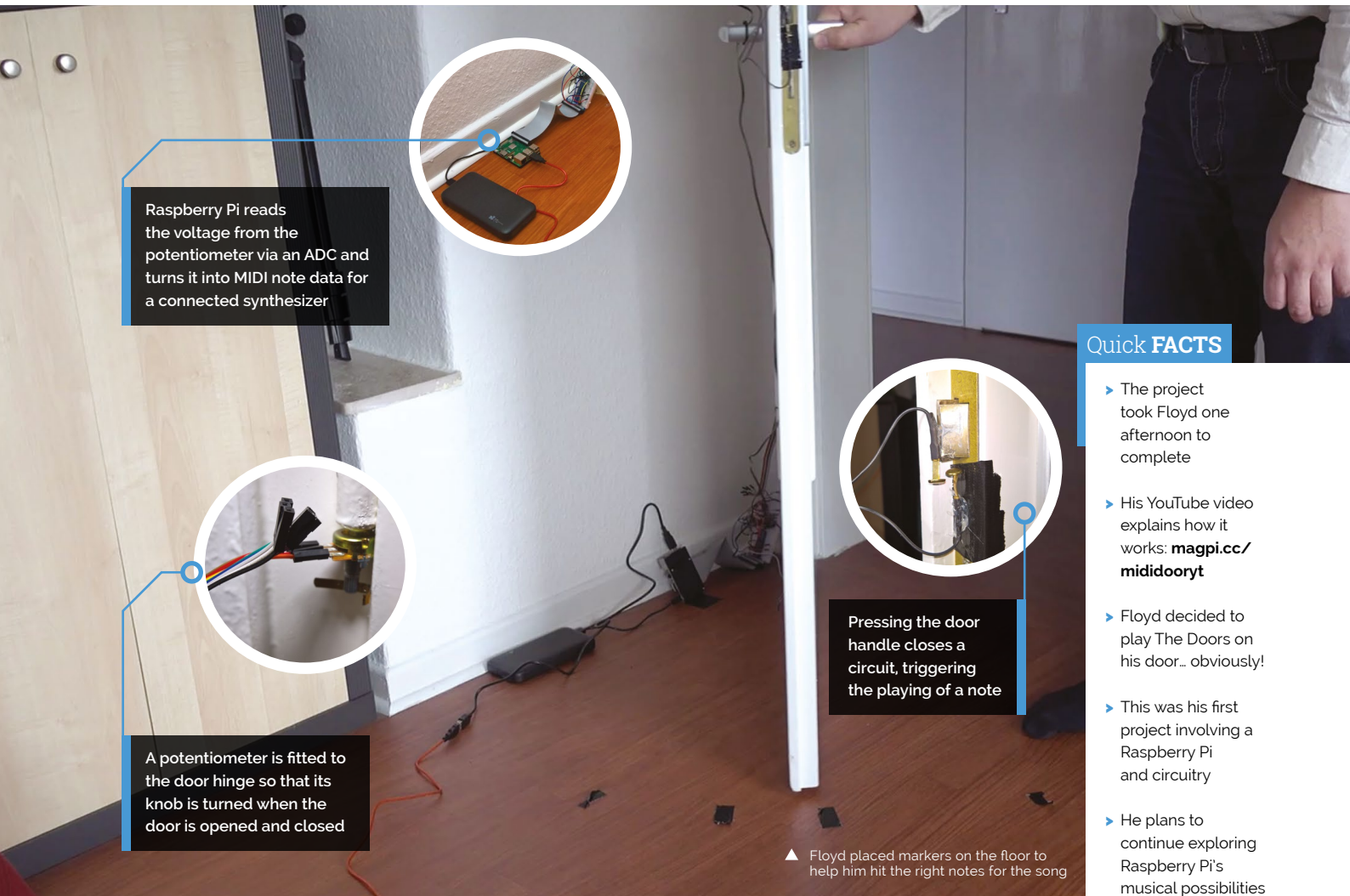
Raspberry Pi reads the voltage from the potentiometer via an ADC and turns it into MIDI note data for a connected synthesizer

A potentiometer is fitted to the door hinge so that its knob is turned when the door is opened and closed

Pressing the door handle closes a circuit, triggering the playing of a note

▲ Floyd placed markers on the floor to help him hit the right notes for the song

the door movement could be picked up precisely. He settled on a solid metal paper clip, with one end glued to the inner edge of the door and the other end attached to a notch in the potentiometer knob – after finding that it was catching on the door frame, Floyd solved the problem by simply bending the clip with a pair of pliers.

## Scores on the doors

Unsurprisingly, viewers of Floyd's YouTube channel have found the idea to be very innovative. "I think most people thought it was really interesting, but in a weird way," he says. "The kids loved it for sure. The whole system is suited for slow music (think ambient music, meditation and so on, haha – meditating on a door handle)."

So, could we see some kind of door orchestra in the near future? "While that would make for a spectacular video, I think I'll use the ultrasonic distance sensor for determining the note's pitch next, creating a variation of a theremin." M

▲ Floyd is a keen musician, but this is the first time he's used a door to play music!

# RIoT Brick

Long-distance running and a love of intrepid devices led one creator to embark on a project demonstrating that 'true geekery and nature can co-exist'. **Rosie Hattersley** tries to keep up

**MAKER**

### Alan Peaty

Cloud architecture engineer Alan enjoys IoT tinkering and inspiring his kids. He says: "Technology is here to stay. We should *all* get to learn about it."

**magpi.cc/ rosieredrobot**

**M**embers of his local running club are no longer fazed when ultra-runner Alan Peaty cruises past with an unidentified extra item or three attached to his person. While some fitness fans go for the ultra-sleek look to minimise air resistance and extra weight, Alan regularly takes the opposite approach and brings a little something extra along on his treks on the Cotswold Way.

Cloud computer engineer Alan regularly blogs about his Internet of Things inventions at **magpi.cc/rosieredrobot** and has previously graced these pages with his Countdown-playing robot (see **magpi.cc/countdown**). "Building intrepid devices I can take along with me outdoors is the perfect way for me to combine my interests in one fell swoop," he says. "I have been known to go running with 3D-printed devices attached to my legs (although understandably only at night) to gather data on my leg movement."



▲ The Brecon Beacons, where Alan tested his project to track his progress during a gruelling running race

Last September, Alan put his movement-sensing project to the test in a gruelling endurance race across the Brecon Beacons in Wales. He created the 'RIoT Brick' (Rosie Internet of Things Brick) so his family – some as far away as Australia – could remotely track his progress. Hooking himself up in this way also enabled Alan to monitor basic environmental conditions during what would be "a very, very long day!"

> ❝ I have been known to go running with 3D-printed devices attached to my legs ❞

### Sensitive subject

The RIoT Brick is a Raspberry Pi Zero-based sensing station housed in a crudely 3D-printed box. Alan cheerfully explains that he prefers to

▼ The RIoT Brick's live data feed
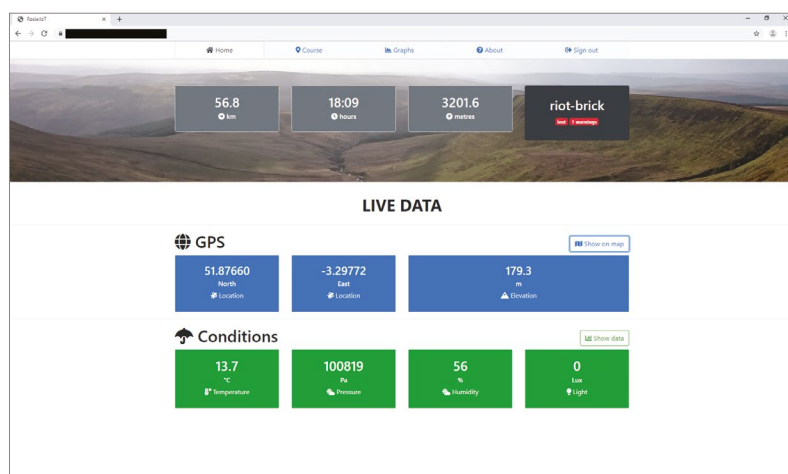
The RIOT Brick is encased in a 3D-printed shell to protect the Raspberry Pi Zero inside safe over the 58-mile trail

The rugged antenna helps the RIoT Brick communicate with other transceivers (maintaining a signal and internet connectivity proved tricky for this project)

The LCD can be used to provide GPS co-ordinates, but was switched off to preserve battery power during the 10 Peaks Challenge. Those following Alan's progress via the web still got detailed data!

R["I⊙T"]

## Quick **FACTS**

> The RIoT Brick project cost less than £50 all-in

> Remote tracker data exchange was a particular challenge

> There was no way to check whether the transceiver was faulty

> It didn't matter because before the race began...

> ...the aerial broke off in Alan's car boot

◀ The rig securely attached to Alan's ultra-running backpack

## Create a RIoT

The RIoT Brick consists of a Raspberry Pi Zero, battery pack, transceiver and sensor modules, plus related AWS scripts. Three Python scripts manage incoming data and check for an active internet connection.
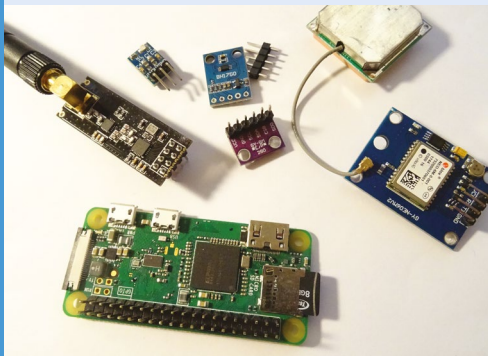


**01** Connect your Raspberry Pi Zero to an nRF24L01+ transceiver. This will receive the data from the other sensors. You also need a u-blox NEO-6 GPS receiver to obtain GPS readings.
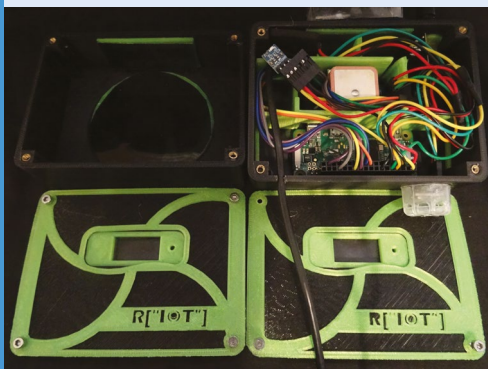


**02** Temperature, pressure, humidity, and light readings are provided by BME280 and BH1750 sensors. Data from all the sensors and transceivers is stored locally in an SQLite database on Raspberry Pi Zero.
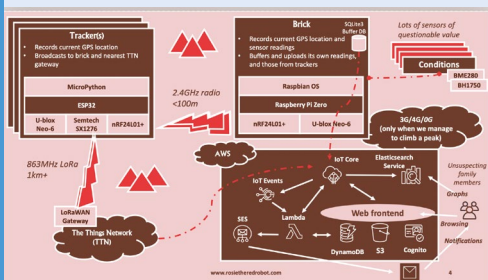


**03** This useful schematic shows the data flow and how to assemble the RIoT Brick. Full instructions are at **magpi.cc/riot10peaks**.
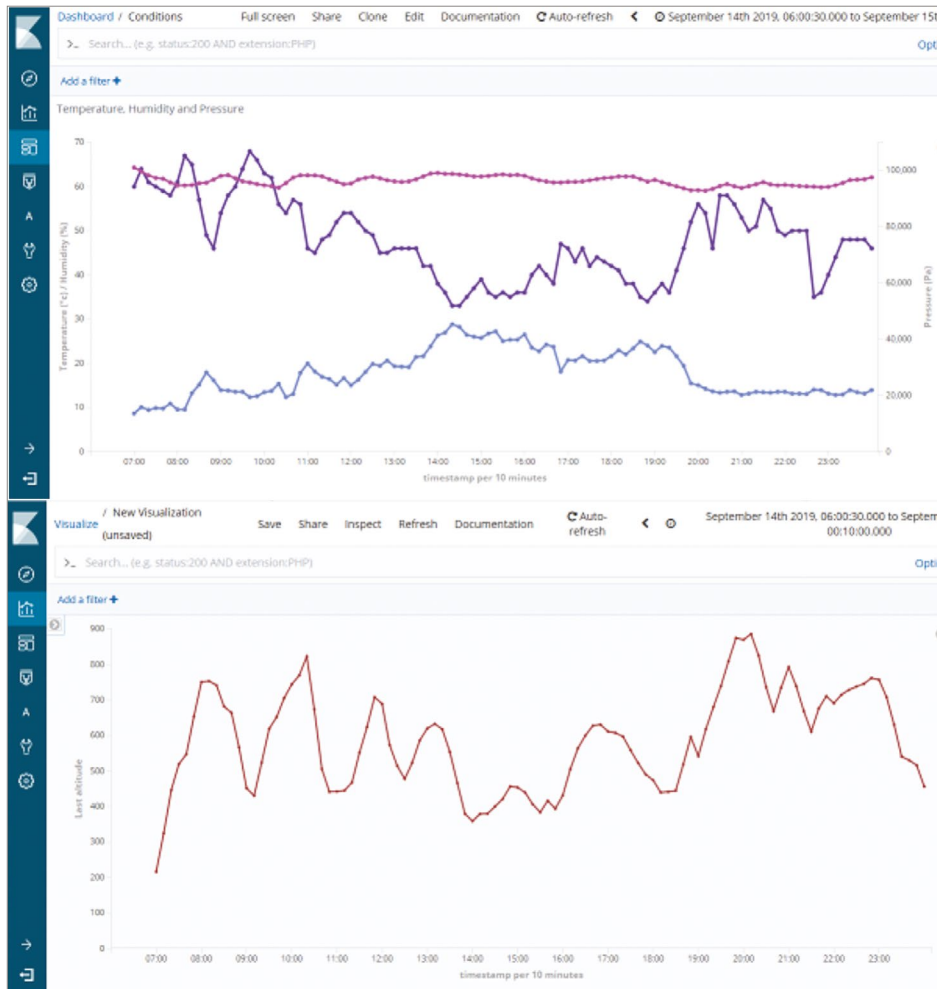


adapt existing code and hardware, and tries to stay well away from designing and printing 'stuff' for his projects. In this instance, the Brick case was needed to protect the Raspberry Pi Zero and sensing modules from many miles of jostling on his 58-mile run across ten peaks.

The box houses a nRF24L01+ transceiver and u-blox NEO-6 GPS receiver, along with BME280 temperature/pressure/humidity and BH1750 light sensors. Alan credits the Raspberry Pi and Linux community for the ease with which he was able to get these devices up and running. "I chose a Raspberry Pi Zero as the central component as I needed a flexible Linux platform that was sufficiently powerful to run multiple applications concurrently," he says.

The abundance of community-developed open-source libraries available for the modules and sensors he planned to connect to it was another big draw. He made use of an SQLite database and Python SDK Amazon Web Services. "IoT Core,
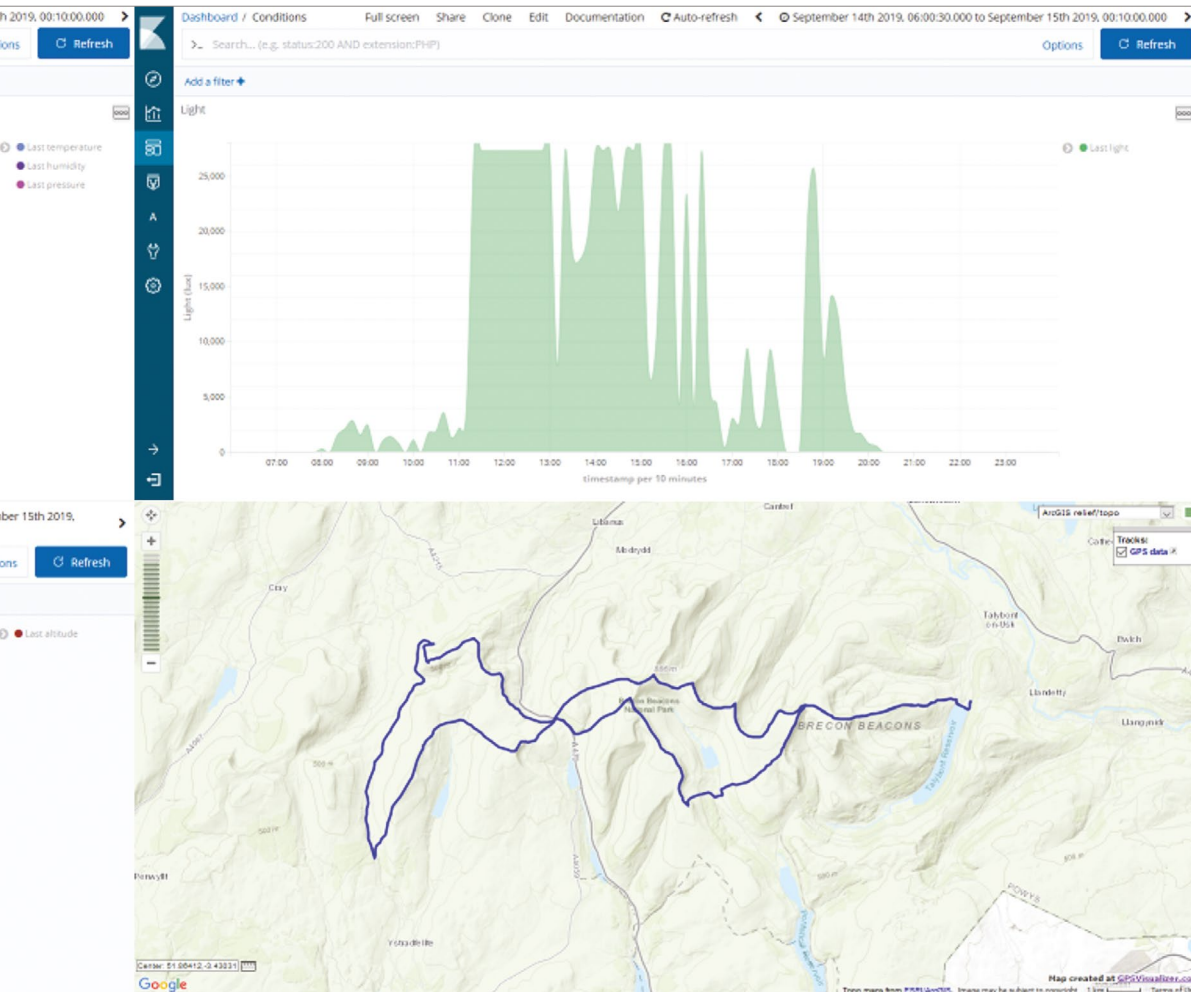
�◀ Data from four sensors in the RIoT Brick captured details of Alan's 10 Peaks race

▼ After 18 whole hours of running, Alan and his Raspberry Pi 'Brick' earned their 10 Peaks medal

## " I was able to raise a few pounds for Alzheimer's Research UK as well "

IoT Events, DynamoDB, S3, Cognito, Simple Email Service, Lambda, and Elasticsearch Service are used to collect, process, and present this information back to my supposedly interested family members", he explains.

### Bricks and pieces

"During the event itself, there were several onlookers understandably baffled by the strange bright device attached to the side of my backpack," says Alan. "A small number were even brave enough to ask! The reaction on Twitter has also been great, and I was able to raise a few pounds for Alzheimer's Research UK as well." **M**

## JOIN FOR 12 MONTHS AND GET A

# FREE Raspberry Pi Zero W Starter Kit

## WITH YOUR SUBSCRIPTION

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

WORTH
£20

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

Offer subject to change or withdrawal at any time
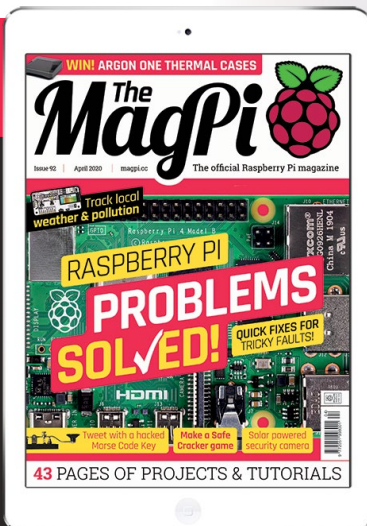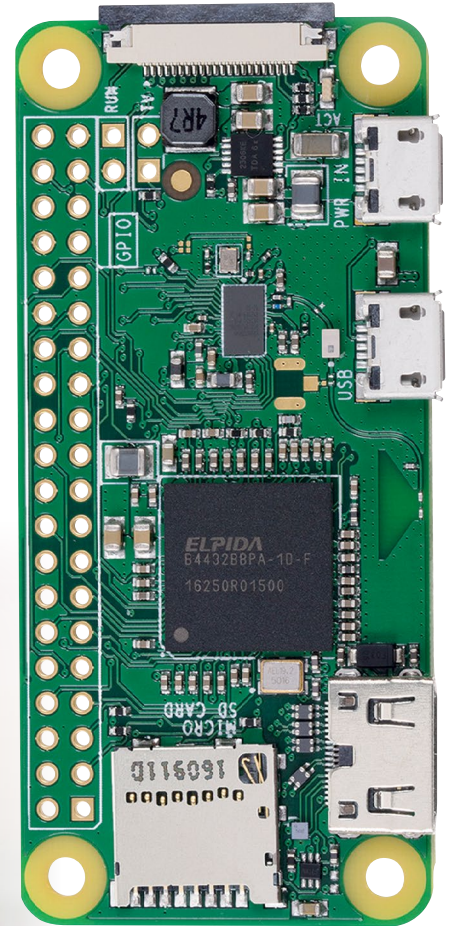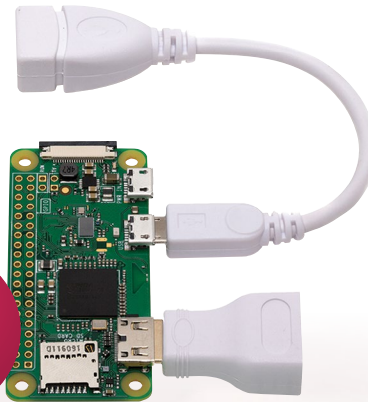
# HOW TO
# WORK FROM HOME
## WITH **RASPBERRY PI**

If you find yourself working – or learning, or simply socialising – from home, Raspberry Pi can help with everything from collaborative productivity to video conferencing

## You'll Need

> **Raspberry Pi, keyboard, mouse, and monitor or TV**

> Network connection (preferably wired)

> **Raspberry Pi Camera Module or USB webcam**

> USB headset, or USB sound card with analogue headset, or USB microphone

**01**  **Install the camera**

If you're using a USB webcam, you can simply insert it into a USB port on Raspberry Pi. If you're using a Raspberry Pi Camera Module, you'll need to unpack it, then find the 'CAMERA' port on the top of Raspberry Pi – it's just between the second micro-HDMI port and the 3.5 mm AV port. Pinch the shorter sides of the port's tab with your nails and pull it gently upwards. With Raspberry Pi positioned so the HDMI ports are at the bottom, insert one end of the camera's ribbon cable into the port so the shiny metal contacts are facing the HDMI port. Hold the cable in place, and gently push the tab back home again.

If the Camera Module doesn't have the ribbon cable connected, repeat the process for the connector on its underside, making sure the contacts are facing downwards towards the module. Finally, remove the blue plastic film from the camera lens.
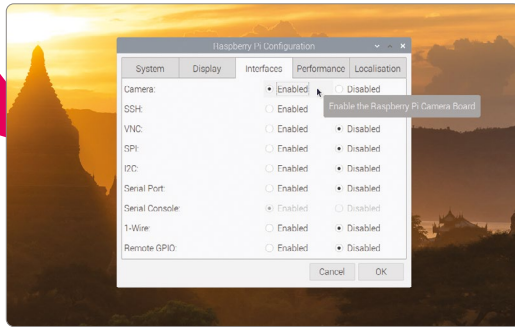
## 02 Enable Camera Module access

Before you can use your Raspberry Pi Camera Module, you need to enable it in Raspbian. If you're using a USB webcam, you can skip this step. Otherwise, click on the raspberry menu icon in Raspbian, choose Preferences, then click on Raspberry Pi Configuration. When the tool loads, click on the Interfaces tab, then click on the 'Enabled' radio button next to Camera. Click OK, and let Raspberry Pi reboot to load your new settings. If you forget this step, Raspberry Pi won't be able to communicate with the Camera Module.

## 03 Set up your microphone

If you're using a USB webcam, it may come with a microphone built-in; otherwise, you'll need to connect a USB headset, a USB microphone and separate speakers, or a USB sound card with analogue microphone and speakers to Raspberry Pi. Plug the webcam into one of Raspberry Pi's USB 2.0 ports, furthest away from the Ethernet connector and marked with black plastic inners.

Right-click on the speaker icon at the top-right of the Raspbian desktop and choose Audio Inputs. Find your microphone or headset in the list, then click it to set it as the default input. If you're using your TV or monitor's speakers, you're done; if you're using a headset or separate speakers, right-click on the speaker icon and choose your device from the Audio Outputs menu as well.

## 04 Set access permissions

Click on the Internet icon next to the raspberry menu to load the Chromium web browser. Click in the address box and type **hangouts.google.com**. When the page loads, click 'Sign In' and enter your Google account details; if you don't already have a Google account, you can sign up for one free of charge.

When you've signed in, click Video Call. You'll be prompted to allow Google Hangouts to access both your microphone and your camera. Click Allow on the prompt that appears. If you Deny access, nobody in the video chat will be able to see or hear you!

## 05 Invite friends or join a chat

You can invite friends to your video chat by writing their email address in the Invite People box, or copying the link and sending it via another messaging service. They don't need their own Raspberry Pi to participate – you can use Google Hangouts from a laptop, desktop, smartphone, or tablet. If someone has sent you a link to their video chat, open the message on Raspberry Pi and simply click the link to join automatically.

You can click the microphone or video icons at the bottom of the window to temporarily disable the microphone or camera; click the red handset icon to leave the call. You can click the three dots at the top-right to access more features, including switching the chat to full-screen view and sharing your screen – which will allow guests to see what you're doing on Raspberry Pi, including any applications or documents you have open.

> " Friends don't need their own Raspberry Pi to participate in a Hangout "

## 06 Adjust microphone volume

If your microphone is too quiet, you'll need to adjust the volume. Click the Terminal icon at the upper-left of the screen, then type **alsamixer** followed by the **ENTER** key. This loads an audio mixing tool; when it opens, press **F4** to switch to the Capture tab and use the up-arrow and down-arrow keys on the keyboard to increase or decrease the volume. Try small adjustments at first; setting the capture volume too high can cause the audio to 'clip', making you harder to hear. When finished, press **CTRL+C** to exit AlsaMixer, then click the X at the top-right of the Terminal to close it.

### Safety first!

Seeing friends and family over video chat is a great way to stay in touch, but always be aware of internet safety. For children, this means asking a parent's permission before using any video chat services – and if you see anything that you don't expect to see, or that upsets you, be sure to tell your parent or guardian.
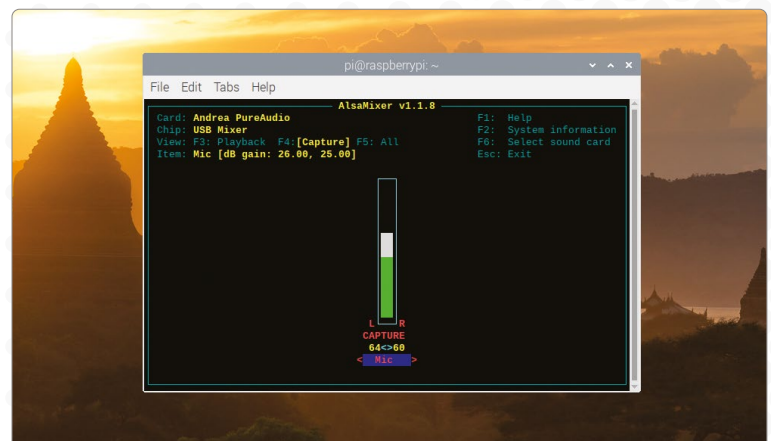
# WORK ONLINE WITH YOUR TEAM

*Just because you're not shoulder-to-shoulder with colleagues doesn't mean you can't collaborate, thanks to these online tools*
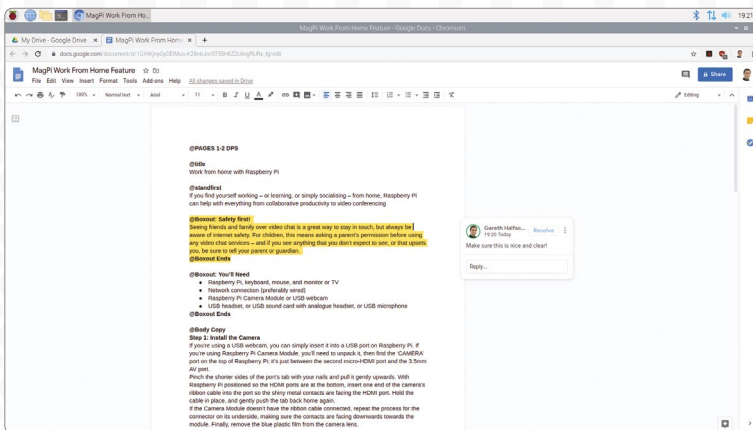


## Google Docs

Google Docs is a suite of online productivity tools linked to the Google Drive cloud storage platform, all accessible directly from your browser. Open the browser and go to **drive.google.com**, then sign in with your Google account – or sign up for a new account if you don't already have one – for 15GB of free storage plus access to the word processor Google Docs, spreadsheet Google Sheets, presentation tool Google Slides, and more. Connect with colleagues and friends to share files or entire folders, and collaborate within documents with simultaneous multi-user editing, comments, and change suggestions.
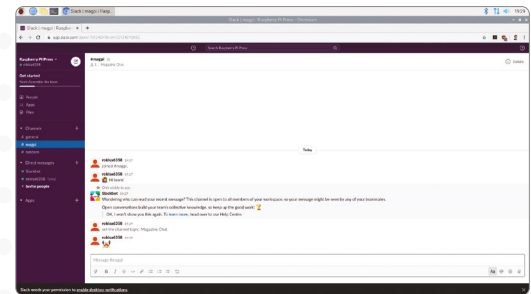


## Slack

Designed for business, Slack is a text-based instant messaging tool with support for file transfer, rich text, images, video, and more. Slack allows for easy collaboration in Teams, which are then split into multiple channels or rooms – some for casual conversation, others for more focused discussion. If your colleagues or friends already have a Slack team set up, ask them to send you an invite; if not, you can head to **app.slack.com** and set one up yourself for free.
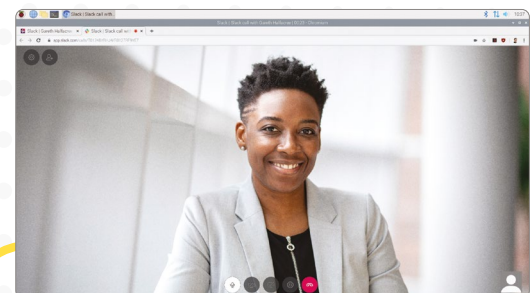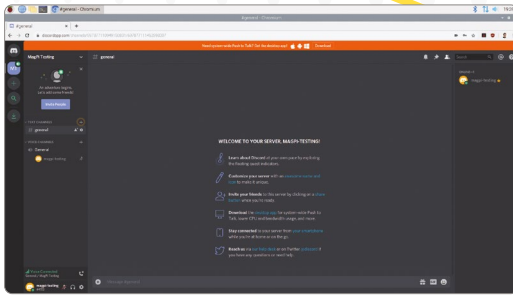
## Discord

Built more for casual use, Discord offers live chat functionality. While the dedicated Discord app includes voice chat support, this is not yet supported on Raspberry Pi – but you can still use text chat by opening the browser, going to **discord.com**, and choosing the 'Open Discord in your browser' option. Choose a username, read and agree to the terms of service, then enter an email address and password to set up your own free Discord server. Alternatively, if you know someone on Discord already, ask them to send you an invitation to access their server.
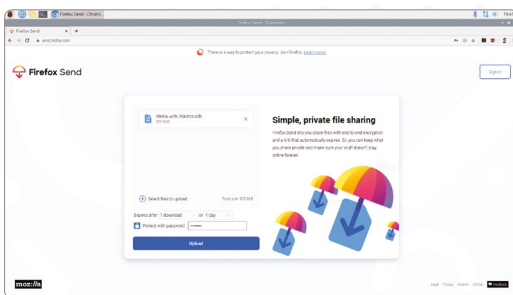


## Firefox Send

If you need to send a document, image, or any other type of file to someone who isn't on Google Drive, you can use Firefox Send – even if you're not using the Firefox browser. All files transferred via Firefox Send are encrypted, and can be protected with an optional password, and are automatically deleted after a set number of downloads or length of time. Simply open the browser and go to **send.firefox.com**; you can send files up to 1GB without an account, or sign up for a free Firefox account to increase the limit to 2.5GB.
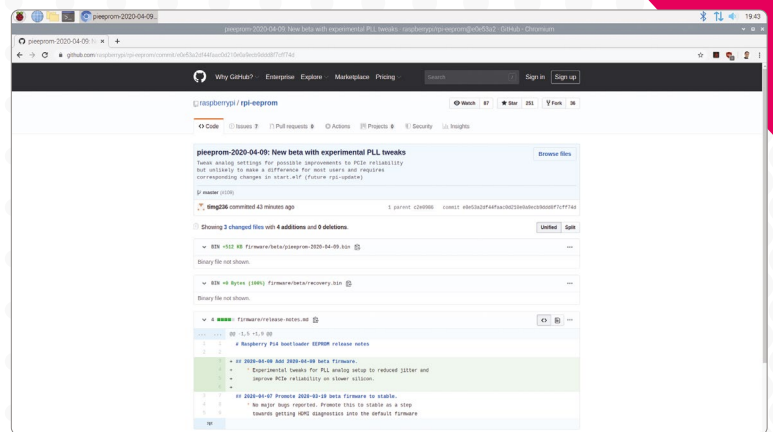


## GitHub

For programmers, GitHub is a lifesaver. Based around the Git version control system, GitHub lets teams work on a project regardless of distance using repositories of source code and supporting files. Each programmer can have a local copy of the program files, work on them independently, then submit the changes for inclusion in the master copy – complete with the ability to handle conflicting changes. Better still, GitHub offers additional collaboration tools including issue tracking. Open the browser and go to **github.com** to sign up, or sign in if you have an existing account, and follow the getting started guide on the site.

## Watch your tabs

If you find Raspberry Pi is running slowly, check how many tabs you have open in your browser. Keeping lots of tabs open in the background is a great way to quickly switch between tasks, but consumes memory – and some sites, particularly messaging tools, consume processing power even when they're only running in the background.

# FREE OFFICE SOFTWARE

*Working alone? This software will get you as productive as possible without ever needing to leave your Raspberry Pi*
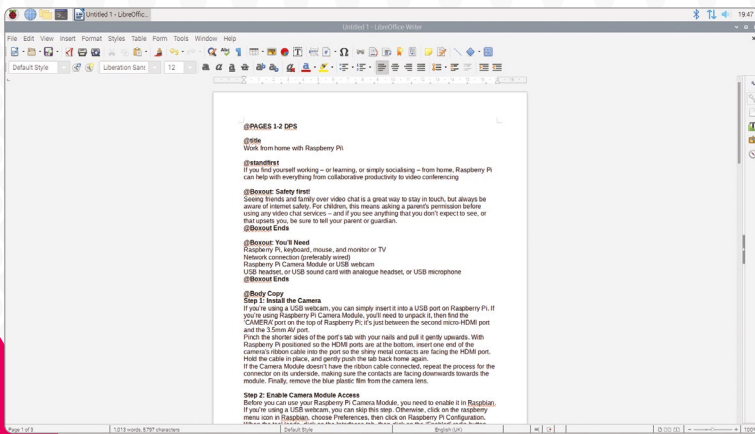


## LibreOffice Writer

Equivalent in functionality to Microsoft Word, LibreOffice Writer is a fully-featured rich-text editing program. You can write new documents and open existing ones in a variety of formats – including Microsoft Word – and insert graphics, spreadsheets, tables, and more. You can even export your files in Portable Document Format (PDF) for commercial printing, or choose from a range of templates.

LibreOffice Writer is found in the raspberry menu under Office.

## Check your Raspbian version

Raspbian is available in three versions: Raspbian Lite, Raspbian with Desktop, and Raspbian with Desktop and Recommended Software. Only the latter version includes additional software like LibreOffice; if your version doesn't include it, use the Recommended Software tool in the raspberry menu under Preferences to install it.

If you're using the Raspberry Pi Imager to write a microSD card, be sure to choose Raspbian (other) then Raspbian Full to get all the recommended software included.
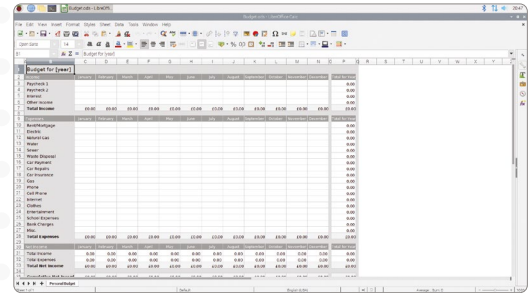


## LibreOffice Calc

A spreadsheet application equivalent to Microsoft Excel, LibreOffice Writer is a great tool for everything from budgeting to complex mathematical modelling. Point-and-click to create new formulae or write them by hand, and the whole spreadsheet will update automatically when things change. A variety of chart options offer visualisation, and can be exported separately from the spreadsheet as a whole.

LibreOffice Calc is found in the raspberry menu under Office.
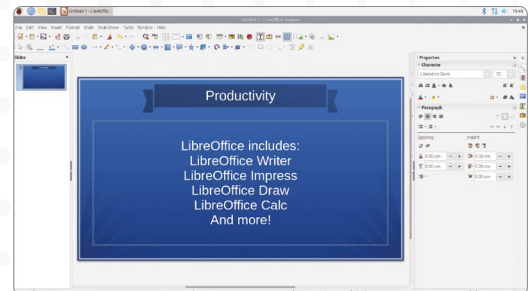


## LibreOffice Impress

Presentations are easy to produce with LibreOffice Impress, a tool equivalent to and compatible with Microsoft PowerPoint. Templates offer an easy way to add a splash of colour, or you can use the built-in tools to design your own slide aesthetics. As you'd expect, a range of animations and other presentation features are included.

LibreOffice Impress is found in the raspberry menu under Office.
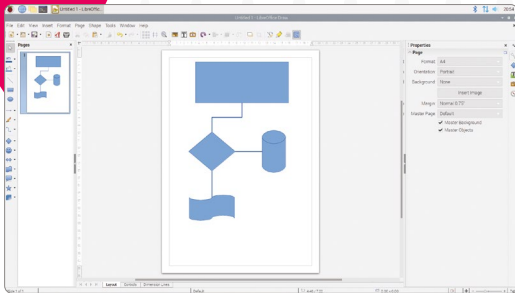
### LibreOffice Draw

Designed for technical drawings, flowcharts, and the like, LibreOffice Draw includes tools for quickly producing vector illustrations. Draw content manually or choose from a range of predefined shapes, icons, lines, and arrow types, then save or export the file directly – or copy it for pasting into LibreOffice Writer or another application.

LibreOffice Impress is found in the raspberry menu under Office.

> " **LibreOffice Draw includes tools for quickly producing vector illustrations** "
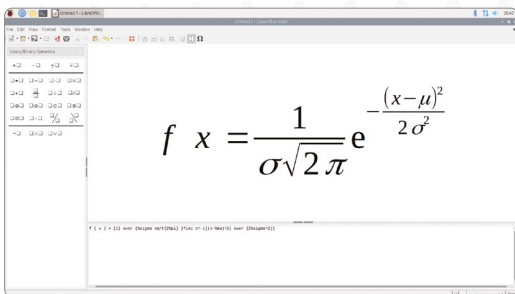


### LibreOffice Math

If you work with mathematical or scientific formulae, LibreOffice Math aims to make your life as simple as possible. Accessible as a standalone application or within other LibreOffice packages, Math offers simple formula editing with elements including fractions, integrals, and matrices. Finished formulae can then be embedded in other documents.

LibreOffice Math is found in the raspberry menu under Office.
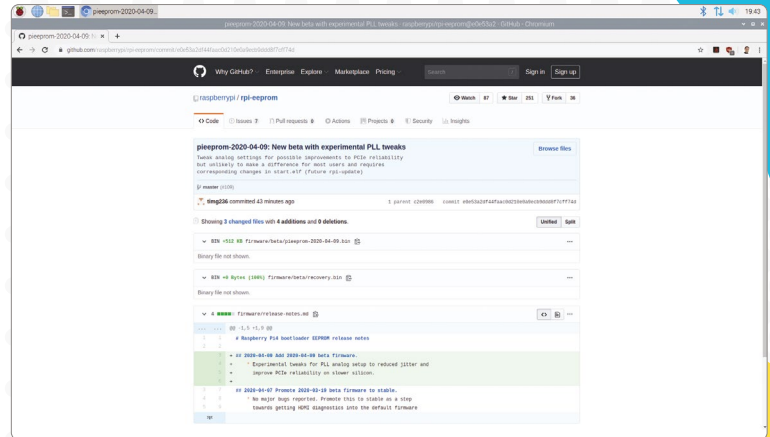


### The GNU Image Manipulation Program

Designed as an open-source alternative to Adobe Photoshop, the GNU Image Manipulation Program (GIMP) is a bitmap graphics editor. Load images from a digital camera, a USB flash drive, or downloaded from the internet to perform a wide range of editing tasks, from simple cropping and rotating to complex image enhancement and distortion effects. Save the edited images in GIMP's internal format while you're working, then export them in a range of formats for sharing or publication.

To install GIMP, click the raspberry menu icon, choose Preferences, then Add/Remove Software. In the search bar at the top-left, type 'GIMP' followed by **ENTER**, then find 'The GNU Image Manipulation Program' and click the checkbox to the left of it. Finally, press OK and enter your password when prompted. Once installed, GIMP will be available from the raspberry menu under Graphics. **M**

## Use USB storage

You can save your work to the microSD card, but for transferring it between machines an external storage device is more convenient. Simply plug a USB flash drive into a USB 3.0 port on Raspberry Pi and it will automatically appear as a device in the File Browser and application menus. You can also use USB SSDs and even hard drives, though more power-hungry storage devices may need a powered USB hub to work correctly.

# Raspberry Pi
# QuickStart Guide

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

**C**ongratulations 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 started, simply download the Raspberry Pi Imager tool, ready to install an operating system on your microSD (or SD) card. This guide will lead you 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 or SD card, use a USB power supply, and feature the same operating systems, programs, and games.

## Top Tip 👍

### Where to buy

You can buy the parts you need from the Raspberry Pi store – **magpi.cc/retail-store** – or online from official resellers. Find links to online sellers at **magpi.cc/products.**
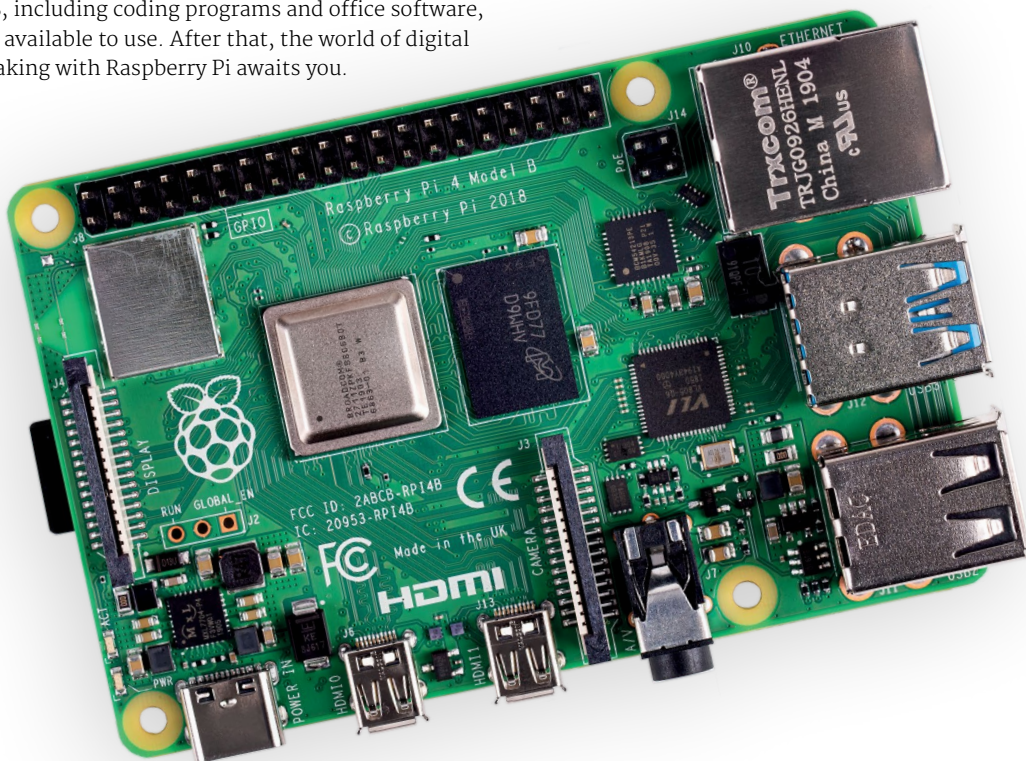
## 8GB microSD card

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 Ubuntu Linux PC, or an Apple Mac computer, to run the Raspberry Pi Imager tool in order to download and write an operating system onto a microSD (or SD) card for your Raspberry Pi.
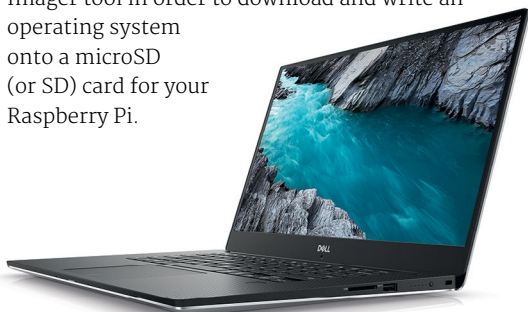
## USB keyboard

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**).
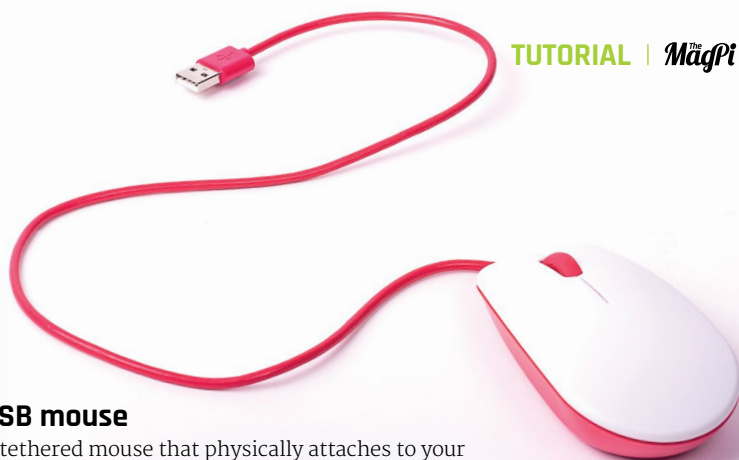
## USB mouse

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.

An 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.
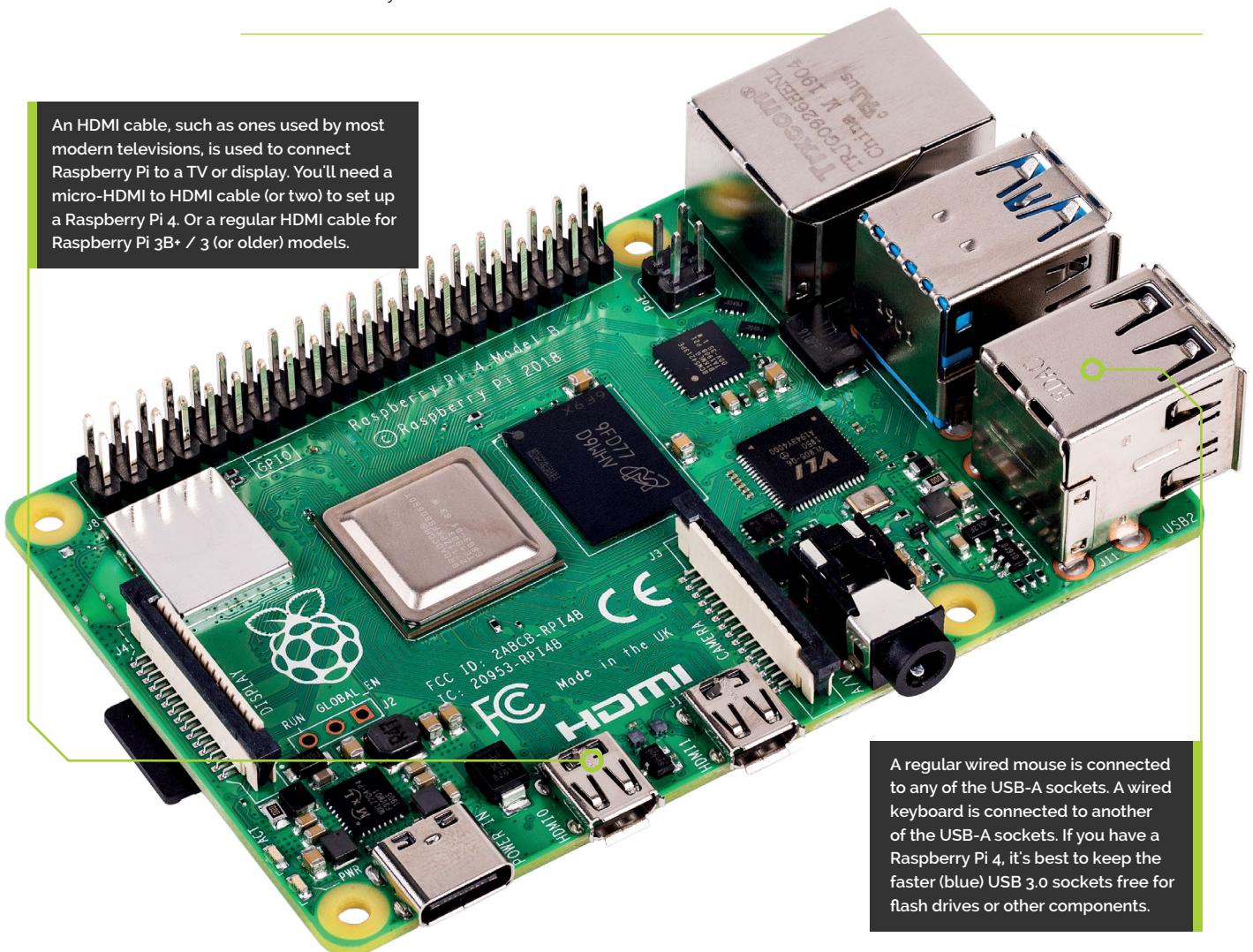
A regular wired mouse is connected
to any of the USB-A sockets. A wired
keyboard is connected to another
of the USB-A sockets. If you have a
Raspberry Pi 4, it's best to keep the
faster (blue) USB 3.0 sockets free for
flash drives or other components.

Make sure you connect your power supply to the micro USB port labelled 'PWR IN'

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
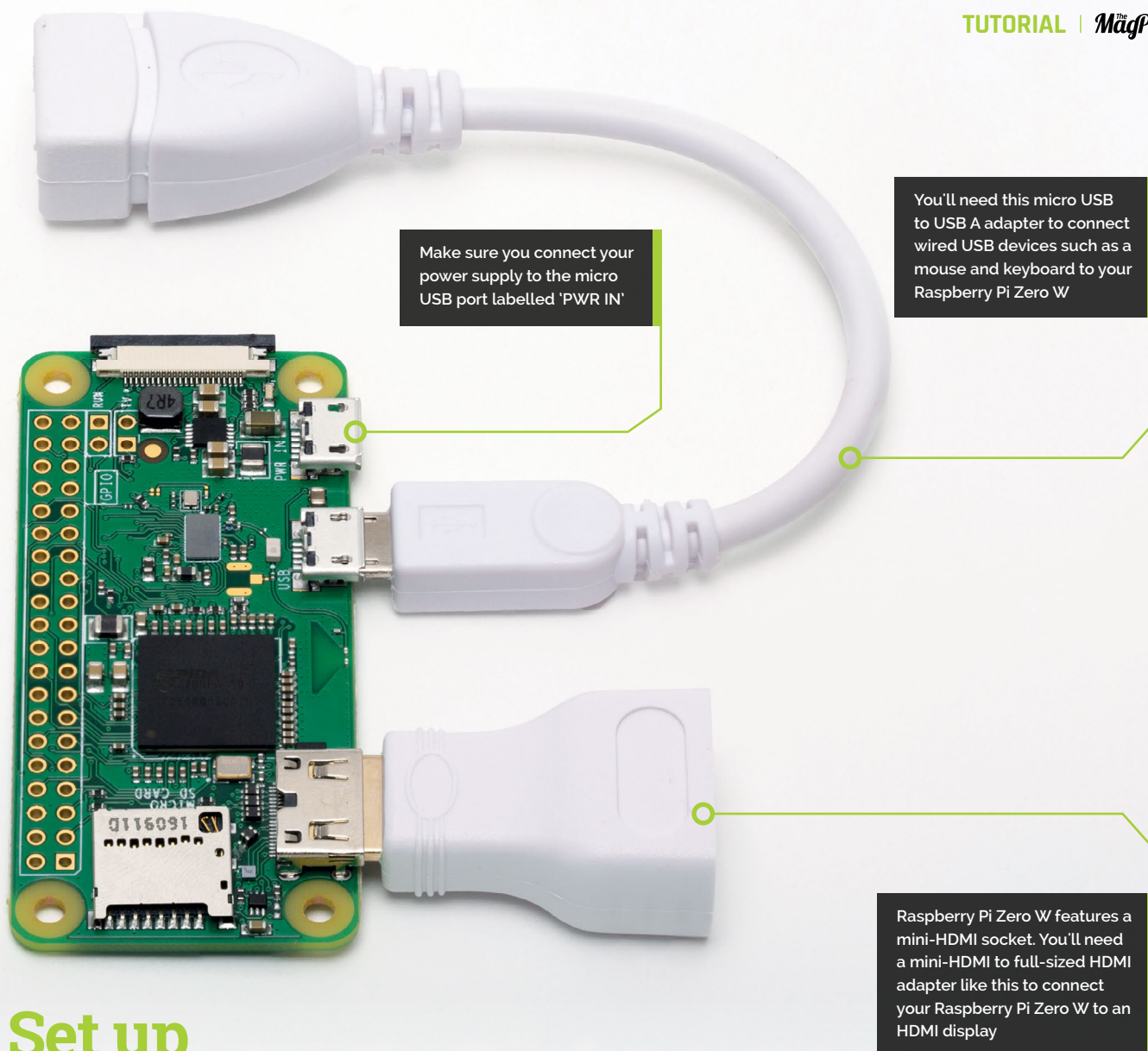
# Set up Raspberry Pi Zero

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 the new Raspberry Pi Imager
tool to install Raspbian OS on
your microSD card



**N**ow 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 the new Raspberry Pi Imager tool (**magpi.cc/imager**).

This method supersedes installing your OS via NOOBS, although the latter is still available from **magpi.cc/downloadnoobs**. Pre-installed NOOBS microSD cards are also on sale. See the 'Using NOOBS' box for instructions.
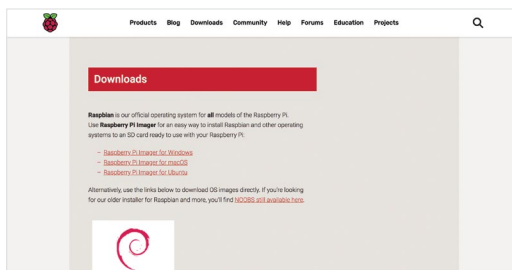


## You'll Need

- 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)
- Raspberry Pi Imager **magpi.cc/imager**

**01** **Download Raspberry Pi Imager**

Download the Raspberry Pi Imager application from **magpi.cc/imager**. This utility is available for Windows, macOS, and Ubuntu Linux computers, so choose the relevant version for your system.

On macOS, double-click the downloaded DMG file. You may need to change your Security & Privacy setting to allow apps downloaded from 'App Store and identified developers' to enable it to run. You can then just drag the Raspberry Pi Imager icon into the Applications folder.

On a Windows PC, double-click the downloaded EXE file. When prompted, select the 'Yes' button to enable it to run. Then click the 'Install' button to start the installation.

**02** **Write the OS to the microSD card**

Attach your microSD card to your PC or Mac computer: you will need a microSD card USB adapter unless you have a card reader built in. Note that the card doesn't need to be pre-formatted.

Launch the Raspberry Pi Imager application. Click the 'Choose OS' button to select which operating system you would like to install. The top option is standard Raspbian – if you'd prefer the slimmed-down Raspbian Lite, or Raspbian Full (with all the recommended software pre-installed), select 'Raspbian (other)'. There are also options to install LibreELEC (choose the version for your Raspberry Pi model) and Ubuntu Core or Server.

Note: If you want to install a different OS, such as Lakka, simply download the image from the relevant website, then select the 'Use Custom' option in Raspberry Pi Imager.

With an OS selected, click the 'Choose SD card' button and select your microSD card (typically there will be just the one option).

Finally, click the 'Write' button and wait while the utility writes the selected OS to your card and then verifies it. When complete, you may remove the microSD card.

**03** **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.

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 Power up

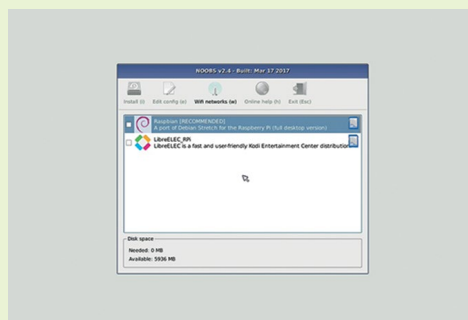Plug in your Raspberry Pi power supply and, after a few seconds, the screen should come on. The Raspbian OS (or alternative OS that you have installed to the microSD card) will then boot up.
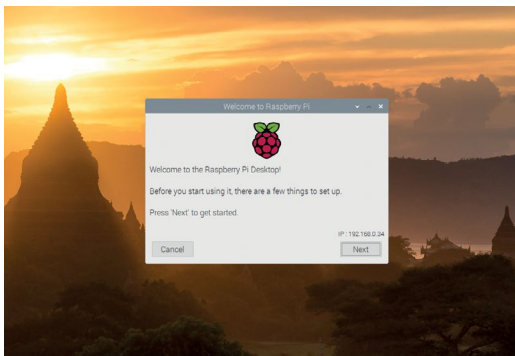
## 05 Get online

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

## Using NOOBS

After downloading the NOOBS file to your computer, double-click to unzip it. Then copy all the files from the NOOBS folder to your attached micro SD card. Once done, eject the card and insert it into your Raspberry Pi.

Plug in Raspberry Pi's 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: 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.

The installation process takes a little while; when it is complete, a message confirming the successful installation appears. Your Raspberry Pi will prompt you to click OK, after which it will reboot and load the Raspbian OS.

# Get started with the
# High Quality Camera

Find out how to set up the new Raspberry Pi camera, connect it, and take your first shots and videos

**MAKER**

**Phil King**

Phil is Raspberry Pi Press's books editor, as well as being a sub-editor and contributor for *The MagPi*.

**magpi.cc**

**T**he new Raspberry Pi High Quality Camera (HQ Camera for short) can capture higher-resolution images than the standard Camera Module. Unlike the latter, it doesn't have a lens already attached. Instead, it can be used with any standard C- or CS-mount lens; 6 mm and 16 mm lenses are available to purchase with the camera to help you get started.

In this tutorial, we'll show you how to attach a lens to the HQ Camera and adjust its focus and aperture settings. We'll then use the supplied ribbon cable to connect it to Raspberry Pi, enable it in Raspbian, and enter some commands in a Terminal window to start capturing photos and video.
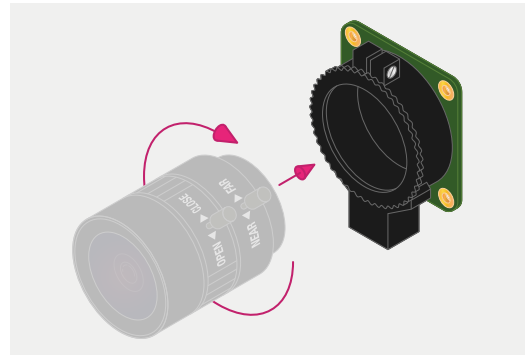
## Using the 6 mm CS-mount lens

A low-cost 6 mm lens is available for the HQ Camera. This lens is suitable for basic photography. It can also be used for macro photography because it can focus objects at very short distances.

### You'll Need

> Raspberry Pi High Quality Camera and lens
**magpi.cc/hqcamera**



**01** **Fit the lens to the camera**
The lens is a CS-mount device, so it has a short back focus and does not need the C-CS adapter that comes with the HQ Camera. Rotate the lens clockwise all the way into the back focus adjustment ring.
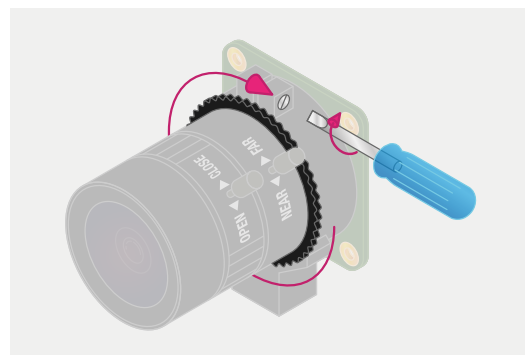


**02** **Back focus adjustment ring**
The back focus adjustment ring should be screwed in fully for the shortest possible back-focal length. Tighten the back focus lock screw to

make sure it does not move out of this position when adjusting the aperture or focus.



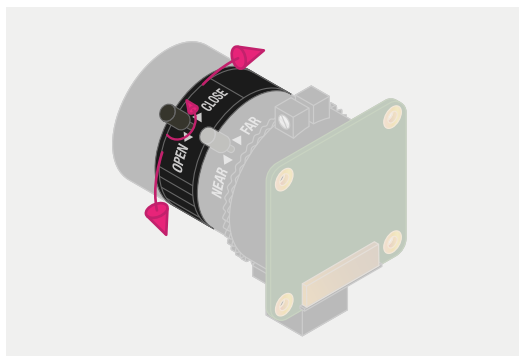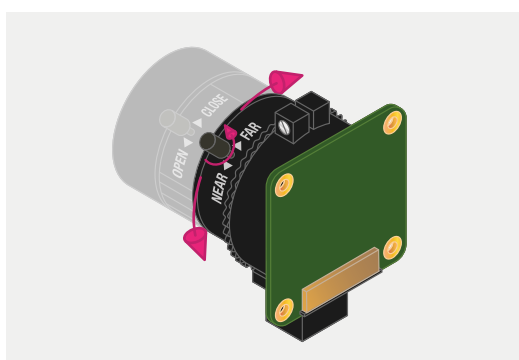### 03 Adjust aperture

To adjust the aperture, hold the camera with the lens facing away from you. Turn the middle ring while holding the outer ring, furthest from the camera, steady. Turn clockwise to close the aperture and reduce image brightness. Turn anti-clockwise to open the aperture. Once you are happy with the light level, tighten the screw on the side of the lens to lock the aperture ring.



### 04 Adjust focus

To adjust the focus, hold the camera with the lens facing away from you. Hold the outer two rings of the lens; this is easier if the aperture is locked as described above. Turn the camera and the inner ring anti-clockwise relative to the two outer rings to focus on a nearby object. Turn them clockwise to focus on a distant object. You may find you need to adjust the aperture again after this.
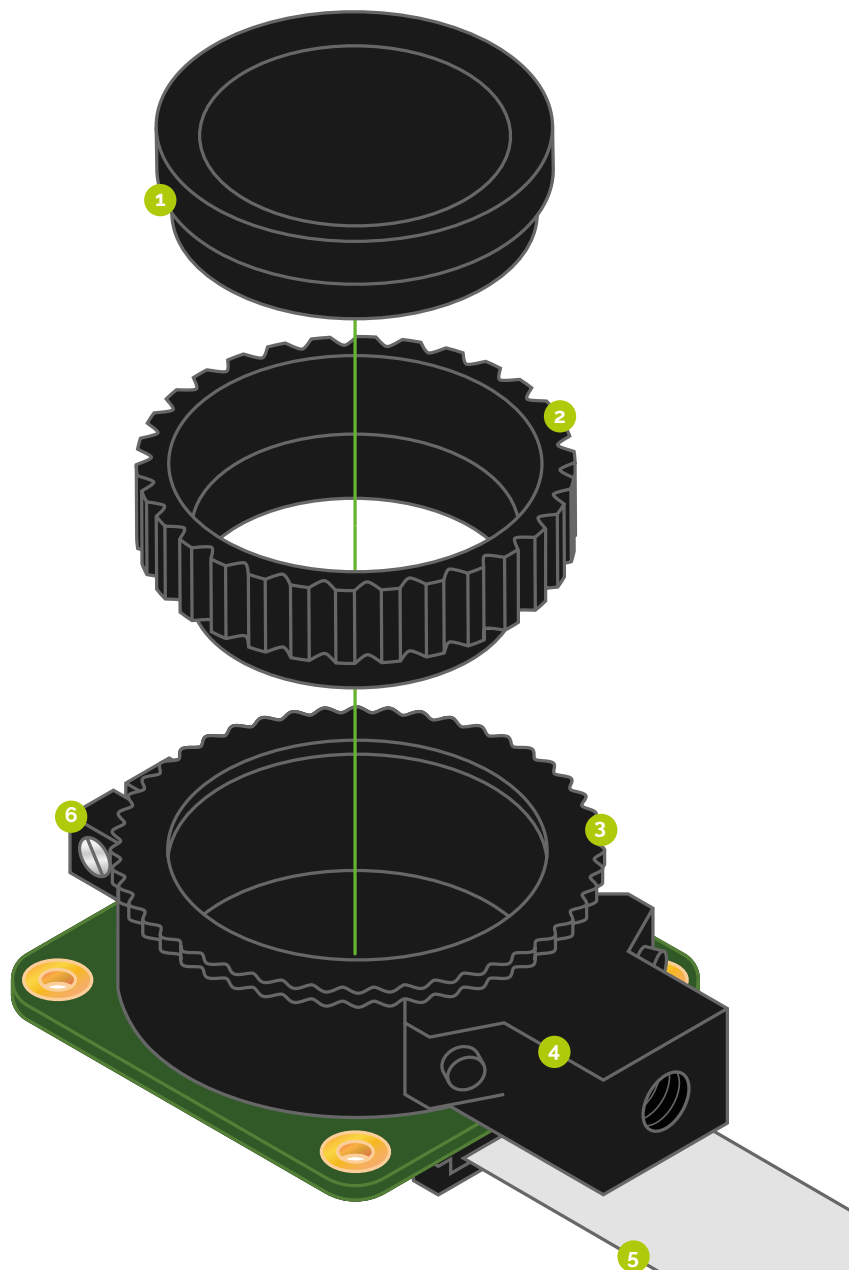


1. The dust cap should be used when a lens is not attached, as the camera sensor is sensitive to dust

2. Supplied with the camera, the C-CS adapter should be used when attaching a C-mount lens

3. The back focus adjustment ring can be used to alter the focus when using a fixed-focus lens, or to change the focal range of an attached adjustable-focus lens

4. This enables you to mount the camera on a standard tripod

5. The camera is supplied with a 20 cm ribbon cable to connect it to Raspberry Pi's Camera port

6. Tighten this screw to lock the back focus adjustment ring in position

# Using the 16 mm C-mount lens

The 16 mm lens provides a higher-quality image than the 6 mm lens. It has a narrow angle of view which is more suited to viewing distant objects.
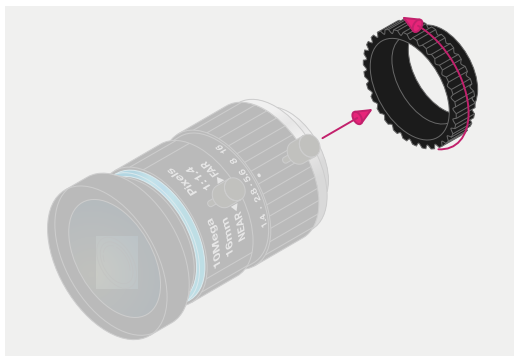


## 01 Fit the C-CS adapter

Ensure the C-CS adapter that comes with the HQ Camera is fitted to the lens. The lens is a C-mount device, so it has a longer back focus than the 6 mm lens and therefore requires the adapter.
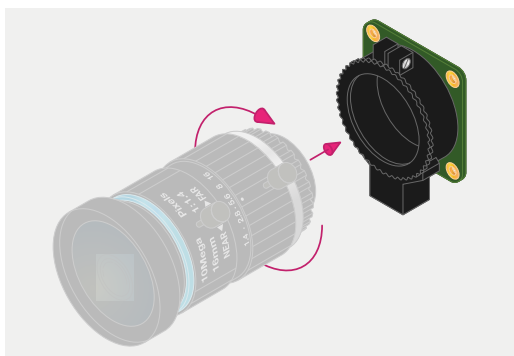


## 02 Fit the lens to the camera

Rotate the lens and C-CS adapter clockwise all the way into the back focus adjustment ring.

## Top Tip 👍

### Longer cable

The HQ Camera is supplied with a standard 20 cm ribbon cable for connection to Raspberry Pi. However, longer camera cables are available from the usual online retailers.



## 03 Back focus adjustment ring

The back focus adjustment ring should be screwed in fully. Tighten the back focus lock screw to make sure it does not move out of this position when adjusting the aperture or focus.

> ❝ Tighten the back focus lock screw to make sure it does not move out of this position ❞
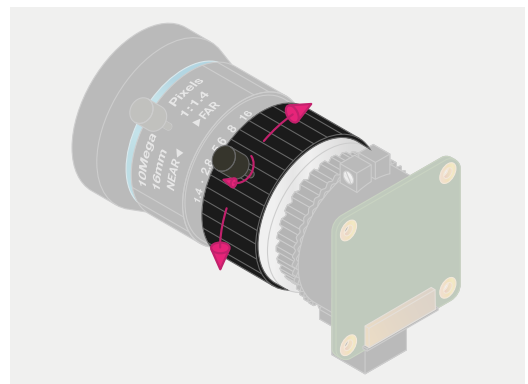


## 04 Adjust aperture

To adjust the aperture, hold the camera with the lens facing away from you. Turn the inner ring, closest to the camera, while holding the camera steady. Turn clockwise to close the aperture and reduce image brightness. Turn anti-clockwise to open the aperture. Once you are happy with the light level, tighten the screw on the side of the lens to lock the aperture ring into position.

## 05 Adjust focus

To adjust the focus, hold the camera with the lens facing away from you. Turn the focus ring, labelled 'NEAR ◄► FAR', anti-clockwise to focus on a nearby object. Turn it clockwise to focus on a distant object. You may find you need to adjust the aperture again after this.

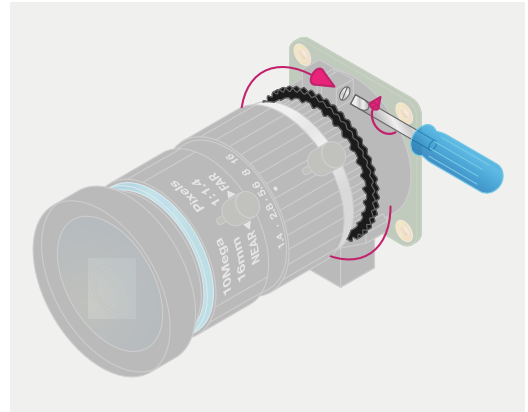## Connecting and using the camera

With your HQ Camera and mounted lens ready, it's time to connect it to your Raspberry Pi and start capturing some images.

## 01 Connect ribbon cable to camera

On the bottom of the HQ Camera board, you'll find a black plastic flap (**Figure 1**). Carefully pull the sticking-out edges until the flap pulls part-way out. Slide the ribbon cable, with the silver edges downwards and the blue plastic facing upwards, under the flap you just pulled out, then push the flap gently back into place with a click; it doesn't matter which end of the cable you use. If the cable is installed properly, it will be straight and won't come out if you give it a gentle tug; if not, pull the flap out and try again.

## 02 Connect cable to Raspberry Pi

Find the Camera port on Raspberry Pi and pull the plastic flap gently upwards. With Raspberry Pi positioned so the HDMI port is facing you, slide the ribbon cable in so the silver edges are to your left and the blue plastic to your right, then gently push the flap back into place. As before, if the cable is installed properly, it'll be straight and won't come out if you give it a gentle tug; if not, pull the plastic flap out and reinsert the cable.
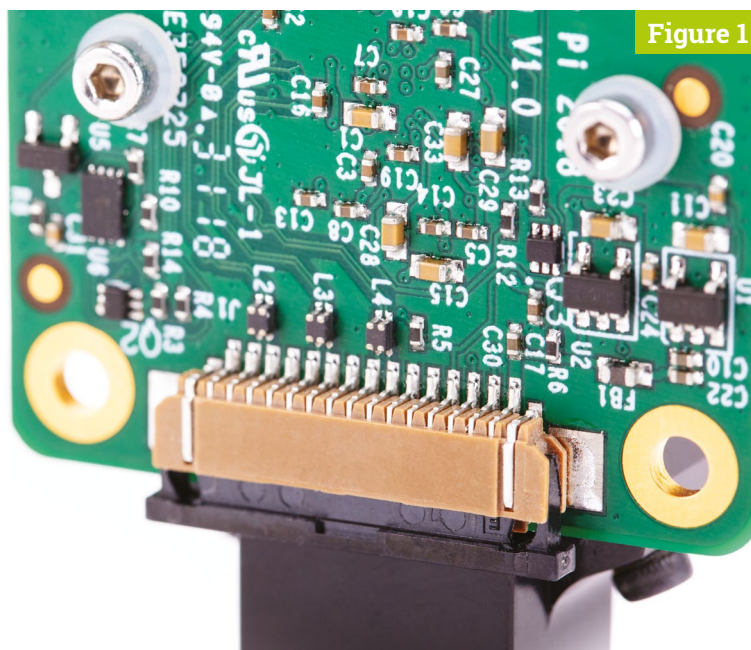


Figure 1

If using a Raspberry Pi Zero, its Camera port is found on the edge of the board. However, as it's a smaller size than the regular one on other Raspberry Pi models, you'll need a camera adapter cable to use it (**magpi.cc/zerocamcable**).

## 03 Enable the camera

Connect the power supply back to Raspberry Pi and let it load Raspbian. Before you can use the camera, you'll need to tell Raspberry Pi it has one connected: in the Raspbian menu, select Preferences, then Raspberry Pi Configuration. When the tool has loaded, click the Interfaces tab, find the Camera entry in the list, and click on the round radio button to the left of 'Enabled' to switch it on (**Figure 2**, overleaf). Click OK, and the tool will prompt you to reboot your Raspberry Pi. Do so and your camera will be ready to use.

## 04 Test the camera

To confirm that your camera is correctly installed, you can use the raspistill tool. This, along with raspivid for videos, is designed to capture images from the camera using Raspberry Pi's command-line interface (CLI). In the Raspbian menu, select Accessories, then Terminal. A black window with green and blue writing in it will

### Top Tip 👍

Permanent preview

To keep the preview window showing on screen, then take a photo whenever the **ENTER** key is pressed, use this command: `raspistill -t 0 -k -o my_pics%02d.jpg`. Press **CTRL+C** to exit.

appear: this is the Terminal, which allows you to access the command-line interface.

To take a test shot, type the following into the Terminal:

```
raspistill -o test.jpg
```

As soon as you hit **ENTER**, you'll see a large picture of what the camera sees appear on-screen (**Figure 3**). This is called the live preview and, unless you tell raspistill otherwise, it will last for five seconds. After those five seconds are up, the camera will capture a single still picture and save it in your home folder under the name **test.jpg**. If you want to capture another, type the same command again – but make sure to change the output file name, after the -o, or you'll save over the top of your first picture.
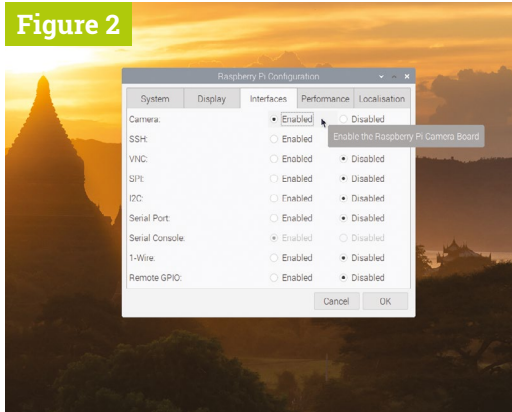


Figure 2

### 05 More advanced commands

The raspistill command has a list of options so long that it borders on the intimidating. Have no fear, though: you won't need to learn them all, but there are a few that might be useful to you, such as:

```
raspistill -t 15000 -o newpic.jpg
```

The -t option changes the delay before the picture is taken, from the default five seconds to whatever time you give it in milliseconds – in this case, you have a full 15 seconds to get your shot arranged perfectly after you press **ENTER**.

### 06 Capture video

For shooting video, raspivid is what you need. Try it out with this Terminal command:

```
raspivid -t 10000 -o testvideo.h264
```

This records a ten-second video (10,000 milliseconds) at the default 1920 × 1080 resolution. You can also shoot slow-mo video at 640 × 480 by using:
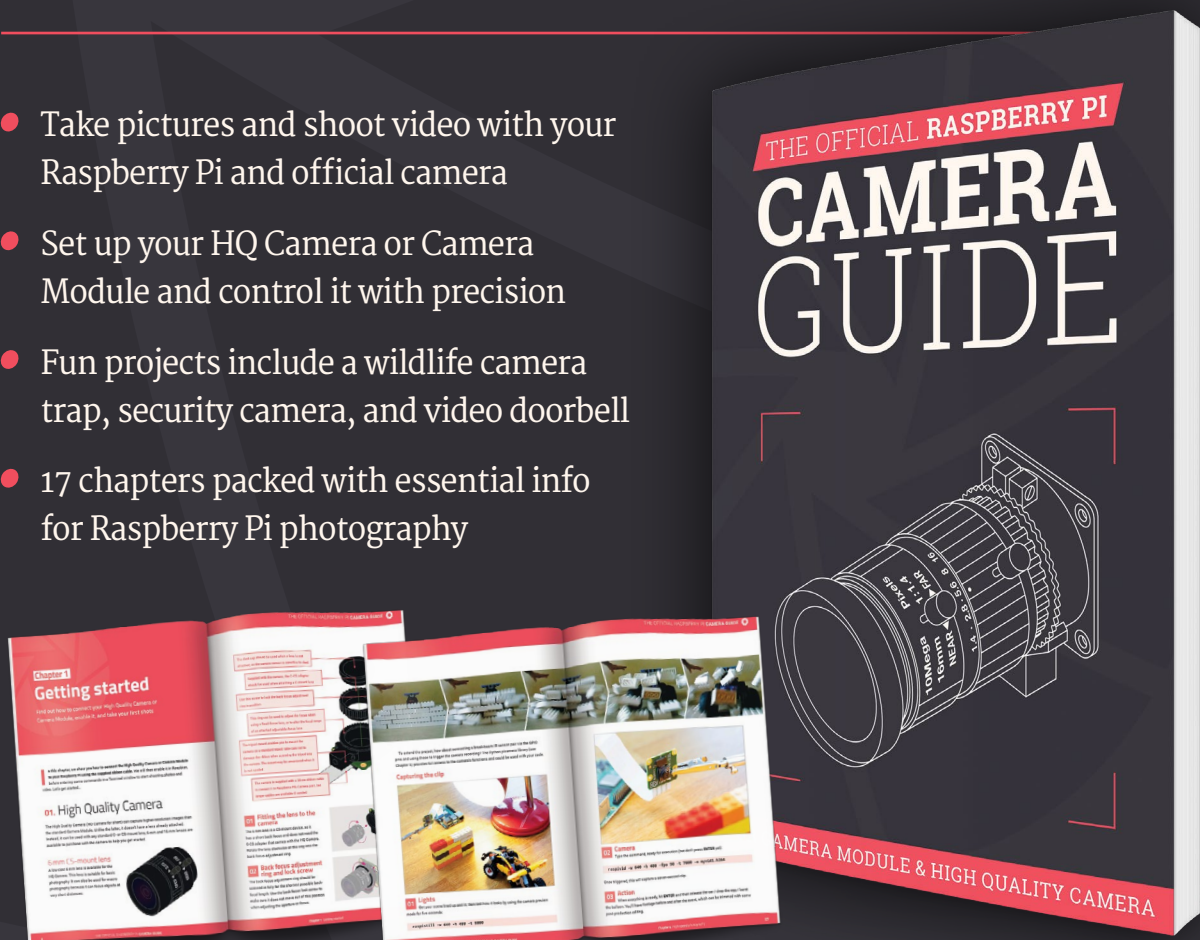
```
raspivid -w 640 -h 480  -fps 90 -t 10000 -o test90fps.h264
```

You can use VLC to play the videos back. This application is pre-installed in the Raspbian 'Full' version. If it's not, you can use the Recommended Software tool to install it. M



Figure 3

**MAKER**

**PJ Evans**

PJ is a writer and software engineer. His Smol Robot would probably be fuelled by coffee and mild panic.

**@mrpjevans**

**Part 04:**

# Create modules for
## your magic mirror

Can't find the perfect add-on for your magic mirror? Why not make your own? In this tutorial we'll create a fun module for MagicMirror²

**A**s we've covered in the previous magic mirror tutorials, there are hundreds of modules out there to display custom information on your MagicMirror² display, including stock prices, the weather, or even **your favourite Pokémon.** So what to do when you can't find what you're looking for? Write your own of course! The module system in MagicMirror² lets anyone add custom code to their mirror with powerful control over the display. You can even interact with other modules. As a simple introduction, we're going to build a simple module that displays the latest cute little robot from the **@smolrobots** Twitter account.

### 01 Set up your mirror

To make sure we don't get confused with anything else, we recommend removing everything from your display except the clock – which we'll use to check that everything is working, as MagicMirror² does its best to keep calm and carry on if a module has a problem. Logging in as the 'pi' user, first make a copy of the config file:

```
cp ~/MagicMirror/config/config.js
~/MagicMirror/config/config-backup.js
```

Now have a look at the file:

```
nano ~/MagicMirror/config/config.js
```

You will see JSON-formatted text. Find the 'modules' section and remove everything so it looks like this:

```
modules: [
        {
```

```
            module: "clock",
            position: "top_left"
        }
    ]
```

Press **CTRL+X** followed by **Y** to save. Now restart the software to make sure everything has taken effect:

```
pm2 restart MagicMirror
```

### 02 Choose how to code

If you can't access your mirror with a display and keyboard (as it's busy being a magic mirror), you'll need to find a way of coding on your Raspberry Pi. You can SSH in and use nano or vi if you like. If you have a capable SFTP client, you can also use this to access files in your home directory and then work on them on your 'regular' computer using a JavaScript code editor such as Visual Studio Code. MagicMirror² is written using JavaScript, so Python editors such as Thonny cannot be used.

### 03 How to create a module

Each MagicMirror² module is a JavaScript file and the code inside is called a class. The class has several built-in methods (like functions) that we don't see and the server calls them to get the information it needs to create the display. What we need to do is replace (or 'override') one or more of the methods to get the output we want. You'll find all modules in the **~/MagicMirror/modules** directory. The built-in displays such as the clock are, unsurprisingly, in 'default'. Create a folder alongside default called **MMM-Hello-World**

### You'll Need

> A magic mirror running MagicMirror²

> JavaScript code editor such as Visual Studio Code
**magpi.cc/ visualstudio**

> 3D-printed case (optional)
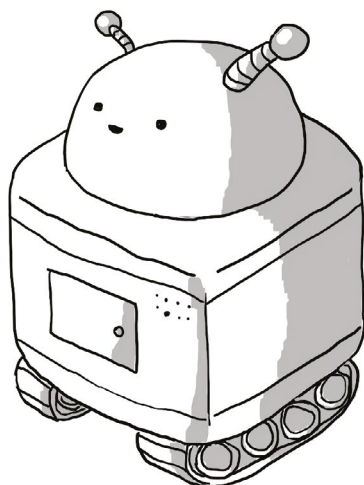
▲ Thomas Heasman-Hunt's Smol Robots are a favourite in the Raspberry Pi community. He even designed the Pi Wars robot!

(MMM = Magic Mirror Module). Inside the folder, create a blank file called **MMM-Hello-World.js**.

## 04 Hello World

As you've probably guessed, we're going to create a simple 'Hello World' module. Enter the code in **listing1.js** (or download it from **magpi.cc/mm4git**) using your code editor. Use nano if you wish, but a proper editor will do all kinds of highlighting and syntax checking which makes it much easier to fix typos and bugs. The code overrides the 'getDom' method, which is called when the server wants us to return some content. Here we create a HTML element, 'div', and add the text to it that is defined in the defaults section. We then return that HTML object to the server to place on the display.

> ❝ A proper editor will do all kinds of highlighting and syntax checking ❞

## 05 Try it!

To actually get our message to display on the screen, we have to ask MagicMirror² to load the module. Let's go back to the config file:

```
nano ~/MagicMirror/config/config.js
```

Find the 'modules' block, as we did previously, and modify if so it looks like this:

```
modules: [
        {
            module: "clock",
```



When developing modules, keep the clock enabled so you can quickly see if something has frozen

We're going to display the latest small robot, but we'll have to invert the image first

```
            position: "top_left"
        },
        {
            module: "MMM-Hello-World",
            position: "lower_third"
        }
    ]
```

Save the file and restart MagicMirror² as in the earlier step. You should now see 'Hello World!' displayed at the bottom of the mirror. You've created your first module! You can also specify a custom message in **config.js** like this:

```
{
    module: "MMM-Hello-World",
    position: "lower_third",
    config: {
        text: "MagPi rocks!"
    }
}
```

This overrides anything in 'defaults' in your module.

## Top Tip 👍

### What's this 'Dom' business?

The 'Dom' in 'getDom' and 'updateDom' refers to the Document Object Model (DOM), which is how the browser structures the webpage you are viewing.

## Top Tip 👍

### Learn JavaScript

New to JavaScript? We've got a guide to learning this powerful language in this very issue.

**06** ### Create the Smol Robots module

Let's get a bit more advanced. Thomas Heasman-Hunt draws cute little robots a few times a week and posts them on Twitter. Wouldn't it be nice to be greeted by a little robot in the morning when we looked in the mirror? Luckily, Tom Panton has created an API that lets us fetch the very latest robot, so we can use that to get the robot image and display it. Create a directory under modules:

```
mkdir ~/MagicMirror/modules/MMM-Smol-Robots
```

Now create a matching JavaScript file:

```
touch ~/MagicMirror/modules/MMM-Smol-Robots/
MMM-Smol-Robots.js
```

**07** ### Accessing the API

Open the file you created in the previous step in your favourite editor and enter the code in **listing2.js**. This is JavaScript, so be careful with the curly brackets and semicolons. Again, we're overriding 'getDom' but this time we're using the built-in 'fetch' function to get information from the Smol Robots API. If you want to see what we're getting, enter the following URL in a web browser:

**https://mrpjevans.com/smol/api/latest/1**

What is returned is a block of JSON-formatted data about the latest robot. You should be able to find its name, description and, most importantly, the location of the image file.

**08** ### Get a robot!

Save your code and edit the config file again to replace the 'Hello world' module. Your config file's modules block should look like this:

```
modules: [
        {
                module: "clock",
                position: "top_left"
        },
        {
                module: "MMM-Smol-Robots",
                position: "lower_third"
        }
    ]
```

Restart MagicMirror[2] as before. Within a few seconds, a delightful robot will grace your screen. The code has accessed the API, located the URL to the image, and created a block of HTML that calls the image in at a reasonable size (feel free to tweak this). The HTML object generated is then returned to the server, which places it in the desired position.

> ❝ This is JavaScript, so be careful with the curly brackets and semicolons ❞

**09** ### Let's be negative

But wait! There's a problem. Smol Robots are always black ink on white, which doesn't work well with our mirror. Sure, if you're happy with it displayed this way, that's up to you, but if we could invert the image, it would look much better.

Thankfully, the powerful world of CSS (Cascading Style Sheets) can do just that. Find the line that reads:

```
img.style = "width: 300px;";
```

Now change it to read like this:

```
img.style = "width: 300px; filter:
invert(1);";
```

'Filter' is a powerful style rule that can perform all manner of tricks on images. In this case, we ask it to

# listing1.js

> Language: **JavaScript**

```
001.  Module.register("MMM-Hello-World",{
002.          // Default module config.
003.          defaults: {
004.                  text: "Hello World!"
005.          },
006.
007.          // Override dom generator.
008.          getDom: function() {
009.                  var wrapper = document.
      createElement("div");
010.                  wrapper.innerHTML = this.config.text;
011.                  return wrapper;
012.          }
013.  });
```

invert all the colours. Save the file and reload your magic mirror. Ah, that's much nicer: our robot now seems to float on the screen.

## 10 More robots!

Our robot is displaying, but what about when a new robot is unleashed from the Smol Robots laboratory? Typically this happens a few times a week, so a daily check should suffice. We can do this by setting up a timer function to refresh the module. Right after the 'defaults' block, add **listing3.js**. This override is called once when the server is started. We use it to set up a timer that will call updateDom (which in turn triggers getDom) once a day. You can change this value in the defaults section or specifying a value in **config.js**. Now your robots will appear like magic.

## 11 Publishing modules

If you've got an idea for a module, and now have enough knowledge to get started, great! When it's done, why not share it with the community? The main page for third-party modules is **magpi.cc/mirrormodules**. We recommend hosting your module on a service like GitHub. Then you can edit this webpage and add your module to the list. Make sure to read (and follow) the guidelines and be prepared to offer support.

## 12 Experiment

This has been only a brief introduction to the world of MagicMirror[2] modules. There is a wealth of information in the official documentation (**magpi.cc/mm2doc**), where you can learn about the other tricks possible. As well as communicating with the outside world, as we have done here, you can communicate with other modules. The weather module can tell you the temperature, or alert you when there's new data. The calendar broadcasts an alert when it's time for an appointment and you can react to that. We hope you get planning and would love to see the results. 🅼

*Many thanks to Thomas Heasman-Hunt for letting us borrow their robots. Check out their latest book at **magpi.cc/smallrobotsbook.** Also, thank-you to Tom Panton for provision of the API.*

## listing2.js

> Language: **JavaScript**

```
001.  Module.register("MMM-Smol-Robots", {
002.    // Default module config.
003.    defaults: {
004.      feed: "https://mrpjevans.com/smol/api/latest/1",
005.      updateInterval: 3600 * 24,
006.      fadeSpeed: 4000
007.    },
008.
009.    // Override dom generator.
010.    getDom: async function() {
011.
012.        Log.info("Refreshing Smol Robot");
013.
014.      // Get the latest Smol Robot
015.      const response = await fetch(this.config.feed);
016.      const bots = await response.json();
017.      const bot = bots[0];
018.
019.      // Wrap the image URL in an IMG element
020.      const img = document.createElement("img");
021.      img.src = bot.image.url;
022.      img.style = "width: 300px;";
023.
024.      // Add title and description
025.      const title = document.createElement("div");
026.      title.innerHTML = bot.name.full;
027.      const description = document.createElement("small");
028.      description.innerHTML = bot.description;
029.
030.      // Create a wrapper element and return
031.      const wrapper = document.createElement("div");
032.      wrapper.appendChild(img);
033.      wrapper.appendChild(title);
034.      wrapper.appendChild(description);
035.      return wrapper;
036.    }
037.  });
```

## listing3.js

> Language: **JavaScript**

```
001.  start: function() {
002.    Log.info("Starting module: " + this.name);
003.
004.    var self = this;
005.
006.    // Schedule update timer.
007.    setInterval(function() {
008.      self.updateDom(self.config.fadeSpeed);
009.    }, this.config.updateInterval);
010.  },
```

Part 01

# Get started with
# Java and electronics

Let's get started with Java 11 with Raspberry Pi and some basic components

**D**id you know that Java is still the most popular programming language in the world, according to the Tiobe Index (magpi.cc/tiobe)? And Raspberry Pi with Raspbian OS is the ideal tool to get started with Java, whether the Java language is new to you or you want to use your existing Java skills and learn to work with hardware components.

In this tutorial, we will guide you through the different steps to get started with Java and electronics with Raspberry Pi.

### 01 Prepare the Raspberry Pi

A simple basis to start with: we're using Raspberry Pi with the most recent Raspbian Full OS, screen, and keyboard. An internet connection will be needed to get additional tools on the board later.

When you start from a new and empty microSD card, use the new Raspberry Pi Imager tool from **magpi.cc/imager**. Make sure you select Raspbian (other) > Raspbian Full as the OS.

When done, put the microSD card in your Raspberry Pi, start it up, and go through all the steps to configure your language, keyboard layout, and wireless LAN connection.

## You'll Need

> Raspberry Pi 2, 3, 3A+, 3B+, or 4

> Raspbian Full version

> LED

> 2 × 330 Ω resistor, push-button, breadboard and wires

> Optional T-Cobbler connection between Raspberry Pi and breadboard

### 02 Test the Java installation

In the release notes of Raspbian (**magpi.cc/releasenotes**) you can see that since version 2019-06-20, Java OpenJDK 11 is included:

```
2019-06-20:
* Based on Debian Buster
* Oracle Java 7 and 8 replaced with OpenJDK 11
```

The JDK (Java Development Kit) is a software package containing everything to develop, compile, and run Java applications with extra developer tools. And because this JDK is already part of Raspbian, we can immediately run our first

Java command to check the installed version. Open a Terminal and type:

```
java -version
```

And you will see:

```
openjdk version "11.0.6" 2020-01-14
OpenJDK Runtime Environment (build
11.0.6+10-post-Raspbian-1deb10u1)
OpenJDK Server VM (build 11.0.6+10-post-
Raspbian-1deb10u1, mixed mode)
```

Because we are starting the Java executable with the option `-version`, the output shows the version and additional information about the installed version of Java.

### 03 Our very first Java application

From Java version 11, you can run simple Java code without the need to compile it, in the same way as you can run a Python or Bash script.

Traditionally, we test this with a 'Hello World' application. Let's move to the home directory and create a new file by editing with the nano text editor.

```
cd /home/pi
nano HelloWorld.java
```

In this file we add code from the **HelloWorld.java** listing, including the `main` method. This is the minimal required code for any Java application.
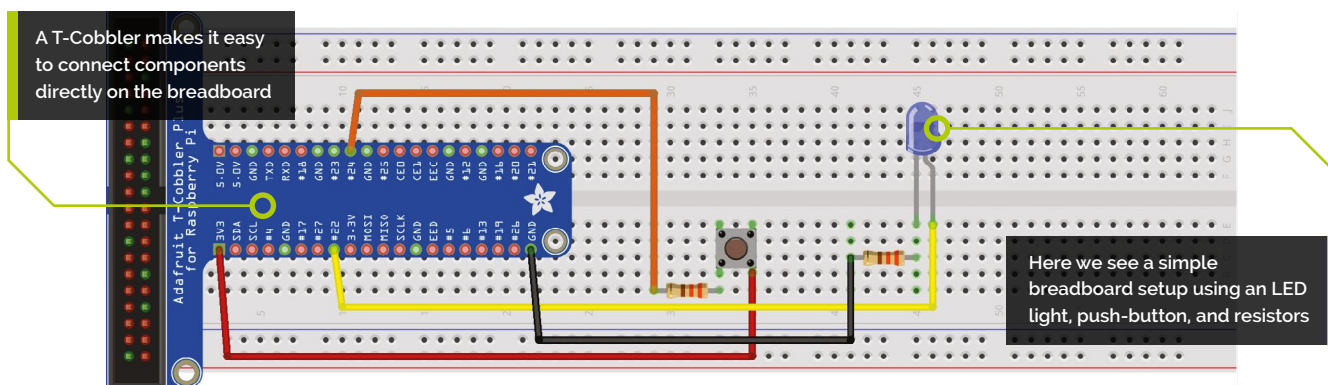
Save this code by exiting from nano with **CTRL+X** and pressing **Y**. Press **ENTER** to confirm the file name. To execute the code, we call Java with the name of the file we just created as an argument:

```
cd /home/pi
java HelloWorld.java
```

The program will run and output the following text on the command line:

A T-Cobbler makes it easy to connect components directly on the breadboard

Here we see a simple breadboard setup using an LED light, push-button, and resistors

```
Hello World
```

And there is our first working Java 11 code on Raspberry Pi.

**04** **Variables and looping in Java code**

Let's write a basic Java program to learn how variables are used and arguments can be given. Make an empty new file called **HelloLoop.java**.

```
cd /home/pi
nano HelloLoop.java
```

Now enter the code from the **HelloLoop.java** listing. This example should not contain any surprises if you've already used another programming language. Some variables are defined and when startup arguments are provided, the application will behave differently.

Save the file and start it without and with one or two number arguments…

```
java HelloLoop.java
```

The code will return:

```
Hello loop: 1, calculated: 1
Hello loop: 2, calculated: 2
Hello loop: 3, calculated: 3
```

Now try:

```
java HelloLoop.java 7
```

And then:

```
java HelloLoop.java 2 4
```

…to get different output. Now is the perfect time to do some experimenting. Can you pass a third string argument to be used instead of 'Hello loop'? Or change the output to two separate lines?

## HelloWorld.java

**DOWNLOAD THE FULL CODE:**

magpi.cc/github

> Language: **Java**

```java
001.  public class HelloWorld {
002.      public static void main (String[] args) {
003.          System.out.println("Hello World");
004.      }
005.  }
```

❝ You can now run simple Java code without the need to compile it ❞

## HelloLoop.java

> Language: **Java**

```java
001.  public class HelloLoop {
002.      public static void main (String[] args) {
003.          String message = "Hello loop";
004.          int loopCounter = 1;
005.          int multiplier = 1;
006.          int numberOfLoops = 3;
007.
008.          if (args.length >= 1) {
009.              multiplier = Integer.valueOf(args[0]);
010.          }
011.
012.          if (args.length >= 2) {
013.              numberOfLoops = Integer.valueOf(args[1]);
014.          }
015.
016.          while (loopCounter <= numberOfLoops) {
017.              int calculated = multiplier * loopCounter;
018.              System.out.println(message + ": " + loopCounter +
      ", calculated: " + calculated);
019.              loopCounter++;
020.          }
021.      }
022.  }
```

### 05 Connect hardware

Let's add hardware to use the full power of Raspberry Pi. Tip: check **pinout.xyz** for more info about the GPIO pin numbers and a clear overview.

We start with an LED and 330 Ω resistor connected to the physical pin 15 (= BCM 22). We will also use a push-button and 330 Ω resistor connected between 3.3V and physical pin 18 (= BCM 24).

To check if we've connected the LED in the correct direction, we can temporary change the cable connected to pin 15 to a 3.3V pin. If the LED doesn't turn on, we need to swap it.

### 06 Test hardware in Terminal

To test the connections, we will use `raspi-gpio` in a Terminal, which allows you to configure BCM 22 as an output pin and turn the LED on (high = dh) or off (low = dl):

```
raspi-gpio set 22 op
raspi-gpio set 22 dh
raspi-gpio set 22 dl
```

Similarly, you can also read the button state connected to BCM 24 after configuring it as an input pin (level 1 = pressed, 0 = not pressed):

```
raspi-gpio set 24 ip
raspi-gpio get 24
GPIO 24: level=0 fsel=0 func=INPUT pull=DOWN
raspi-gpio get 24
GPIO 24: level=1 fsel=0 func=INPUT pull=DOWN
```

### 07 Full HelloGpio.java code

Now let's run the full code provided in the code box of this article! Start with a new empty file or a copy and call it **HelloGpio.java**.

```
cd /home/pi
cp HelloWorld.java HelloGpio.java
nano HelloGpio.java
```

Now enter the code from the **HelloGPio.java** listing. The first two `import` lines are needed to tell our application we need some extra components.

```
import java.io.IOException;
import java.util.Scanner;
```

The lines which start with '//' are comments and not used by Java. They are added here to explain the code. In the `main` method, we initialise BCM 22 to be an out-pin and toggle it between 'on' and 'off', with an interval of one second. At each interval, we

also read the button state from BCM 24 and output a text if it is pressed. All this is done in a `while` loop with a sleep. As Java requires us to handle possible exceptions, there is some extra try/catch code.

We use the Java runtime executor to run the exact same commands we used before in the Terminal. This functionality is provided in a separate method called `execute`, which returns the result of the command. This way, we can check the state of the button.

Running this code can be done in the same way, without compiling:

```
java HelloGpio.java
```

This will output:

```
Changing LED to ON
Changing LED to OFF
Changing LED to ON
Button is pressed
Changing LED to OFF
Button is pressed
Changing LED to ON
Changing LED to OFF
```

## ❝ We use the Java runtime executor to run the same commands as before ❞

### 08 Using Visual Studio Code

Working in nano or another text editor is a good start, but an IDE (integrated development environment) makes it easier to write and run the code. Visual Studio Code is the perfect match and you can use it for a lot of different programming languages by installing extensions.

There is a small problem with Visual Studio Code on Raspberry Pi, as it's only available in a 64-bit version on the Microsoft website. But with this script, we can still get an earlier 32-bit version:

```
cd /home/pi/
wget https://github.com/stevedesmond-ca/vscode-arm/releases/download/1.28.2/vscode-1.28.2.deb
sudo dpkg -i vscode-1.28.2.deb
```

Visual Studio Code will be added to Raspbian's Programming submenu, and its built-in terminal allows you to start your program from there.

When you open a Java file for the first time in VSC, it will immediately ask you to install the 'Java Extension Pack'. This is an extension from Microsoft which combines multiple other extensions to make it easy to write Java in Visual

Studio Code. When installed, you'll need to reload the application.

## 09 Remote development

A second option with Visual Studio Code is to install it on your PC and use remote development. This allows you to work on your PC and use its full power, but with files located on a Raspberry Pi.

You'll need to enable SSH on your Raspberry Pi. But first, make sure you change your password form the default 'raspberry', because this is a security risk. In a Terminal, enter:

```
passwd
```

Enter your current password, then enter and re-enter a personalised password. Now start the configuration tool:

```
sudo raspi-config
```

Go to Interfaces and enable SSH. And while we are in the Terminal, type `hostname -I`. This returns the IP address of the board. In our case it was 192.168.0.213.

In Visual Studio Code on your PC, go to the Extensions menu and search for Remote Development, then click Install. When this is done, hit **F1** for the 'Search Everywhere' pop-up on top and search for Remote-SSH and hit **ENTER**. Make a new connection with the login, password, and IP address of your Raspberry Pi.

If you didn't change the username, the SSH connection command will be `ssh pi@192.168.0.213` (use your IP address here that you found with `hostname -I`). You will also need to fill in the password you configured before.

Give VSC some time to install the necessary files on the board so it can be used for remote development. You will see on the lower left of the IDE that it's busy or ready. Now you can open a file or directory from Raspberry Pi on your PC in VSC.

For a more in-depth explanation, check out this Visual Studio Code blog post: **magpi.cc/vscremote**.

## 10 Experiment and have fun

Can you use Visual Studio Code and combine the HelloLoop and HelloGpio code to define the speed of the blinking LED and the number of loops with startup arguments? Can you change the order of the code and the `if` statements so that the LED only blinks when the button is pressed? Or end the loop when the button is pressed five times? Just a few ideas to get you to work… M

# HelloGpio.java

> Language: **Java**

```java
001.  import java.io.IOException;
002.  import java.util.Scanner;
003.
004.  public class HelloGpio {
005.
006.      // Main method required by Java, this is called at the start
      of the application
007.      public static void main (String[] args) {
008.          try {
009.              // Initialize GPIO BCM 22 as output pin
010.              execute("raspi-gpio set 22 op");
011.              // Initialize GPIO BCM 24 as input pin
012.              execute("raspi-gpio set 24 ip");
013.              // Initialize the integer loop counter as 1
014.              int loopCounter = 1;
015.              // Initialize the boolean for the led state to true
016.              boolean ledOn = true;
017.
018.              // Loop six times
019.              while (loopCounter <= 6) {
020.                  // Change the LED state
021.                  if (ledOn) {
022.                      System.out.println("Changing LED to ON");
023.                      execute("raspi-gpio set 22 dh");
024.                  } else {
025.                      System.out.println("Changing LED to OFF");
026.                      execute("raspi-gpio set 22 dl");
027.                  }
028.
029.                  // Check the state of the button
030.                  if (execute("raspi-gpio get 24").
      contains("level=1")) {
031.                      System.out.println("Button is pressed");
032.                  }
033.
034.                  // Change the state for the LED for the next loop
035.                  ledOn = !ledOn;
036.
037.                  // Wait a second
038.                  Thread.sleep(1000);
039.
040.                  // Increment the loop counter with one
041.                  loopCounter++;
042.              }
043.          } catch (InterruptedException ex) {
044.              System.err.println("Exception from Thread: " +
      ex.getMessage());
045.          }
046.      }
047.
048.      // Separate methode to execute the given command and return
      the result string.
049.      public static String execute(String cmd) {
050.          try {
051.              Scanner s = new Scanner(Runtime.getRuntime().
      exec(cmd).getInputStream()).useDelimiter("\\A");
052.              return s.hasNext() ? s.next() : "";
053.          } catch (IOException ex) {
054.              System.err.println("Exception from Runtime: " +
      ex.getMessage());
055.              return "";
056.          }
057.      }
058.  }
```

Part 02

# Rotary encoders:
# Polar Etch-A-Sketch

Use rotary encoders to create a special Etch-A-Sketch device

**MAKER**

**Mike Cook**

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

**magpi.cc/mikecook**

**L**ast month, we explored how to read a rotary encoder with Raspberry Pi and created a Simon Safe Crack game with it. Now, let's see how to use two encoders to make a unique Etch-A-Sketch, and explore the world of polar co-ordinate functions.

### 01 What are polar co-ordinates?

When we plot a graph, we use points defined by how far along and how far up; these are called Cartesian co-ordinates. **Figure 1** shows a line connecting the point four units along and three units up and the origin, which is no units along and no units up. This will produce a line which is 5 units long and has an angle of 30°, marked with θ (theta). Instead of defining that point in Cartesian co-ordinates, we can simply use r and θ: how far out and how far round. These are know as polar co-ordinates.

### You'll Need

▶ Two KY-040 rotary switches
**magpi.cc/rotary**

▶ 3 mm MDF or plywood for the box
**magpi.cc/3mmdf**

▶ Two 40 mm knobs
**magpi.cc/40mmknobs**

### 02 Why bother with polar co-ordinates?

Well, polar co-ordinates offer a different view on things. For example, the sensitivity of a



▲ **Figure 2** Using a bent solder tag to push out the sides and stop them bending

microphone is directional; what better way to show that graphically than with polar co-ordinates? It is instantly more understandable than showing the same thing in Cartesian co-ordinates. Functions can also be expressed in polar co-ordinates, and you are able to make many more pleasing patterns than you could using Cartesian ones. So, we thought it would be fun to make a polar Etch-A-Sketch, using two rotary encoders – one for the radius r, and the other for the angle θ.

### 03 Construction

Basically, this is similar to last month's single encoder design, but this time we have two encoder switches. As they are designed to be used with one in each hand, we spaced them 200 mm apart in a 280×80 mm box, 24 mm deep. The construction was the same as last month's box, only we added an extra pair of spacers in the middle. We found that when gluing the sides to the base, the sides bowed in a bit, so we used a solder tag bent up at the end to push the sides back into place – see **Figure 2**.



▲ **Figure 1** Two types of graphing co-ordinate systems

| | | |
|---|---|---|
| X = 4, Y = 3 | === | 5, 30° |
| (4, 3) | | (r, θ) |
| Cartesian | is the same point as | Polar |

The left knob acts as the θ (angle) control

The right knob acts as the r (radius) control
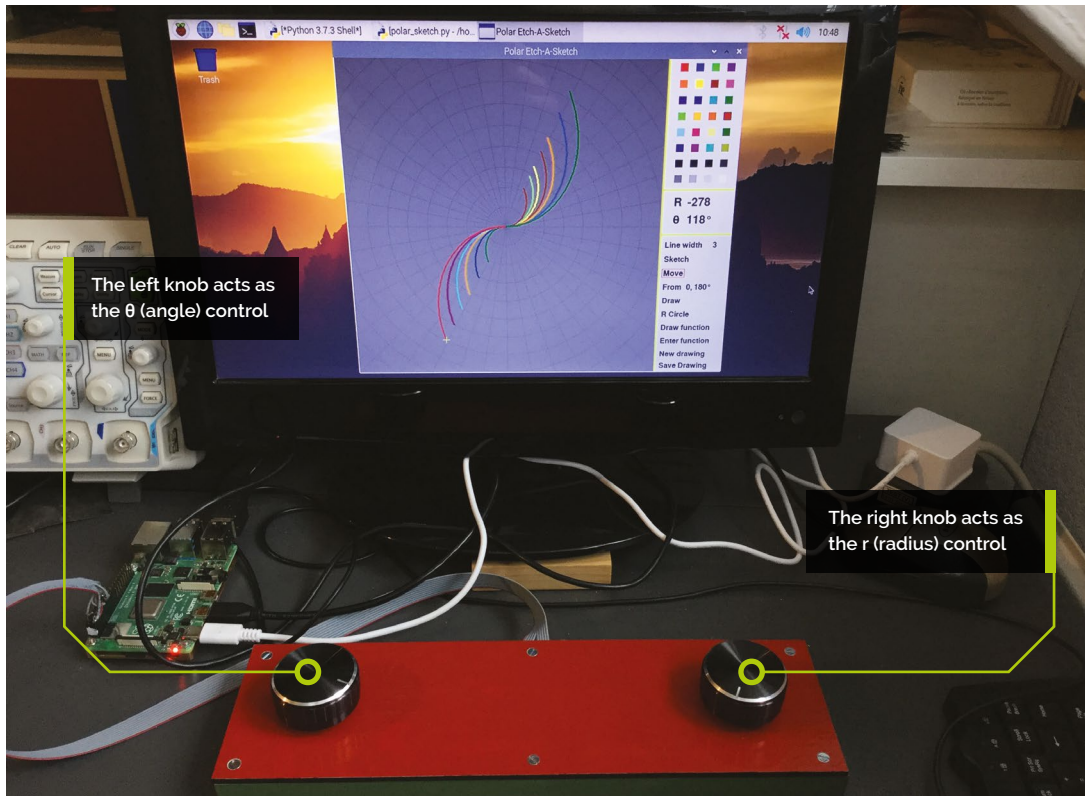
## Top Tip 👍

Ribbon cable placement

It is best to place the ribbon exit slot one side of the central spacer – not in the middle where we put it.

## 04 Wiring

The two switches were wired up to Raspberry Pi's GPIO pins via a short 20-way connector, attached to the rightmost 20 pins using the connections shown in **Figure 3**. The left-hand encoder was wired up to the same pins we used last month. We also added the two I²C lines for something else we might do with this box in the future. To make the wiring easer, we first mounted the switches on the lid and wrote, in pencil, the signal names beside each switch – see **Figure 4**.

> ❝ The left-hand encoder was wired up to the same pins we used last month ❞

## 05 The grid

Our grid was devised mainly to make things easier for the calculations, and fit in with what the arctan2 function returns. This is a function required when converting between Cartesian and polar co-ordinates. However, like all computer trigonometry functions, it works in radians and returns an angle between π and −π radians. We wanted to use degrees between 0 and 360, so we ended up with the grid shown in **Figure 5** (overleaf). This is a little unconventional because
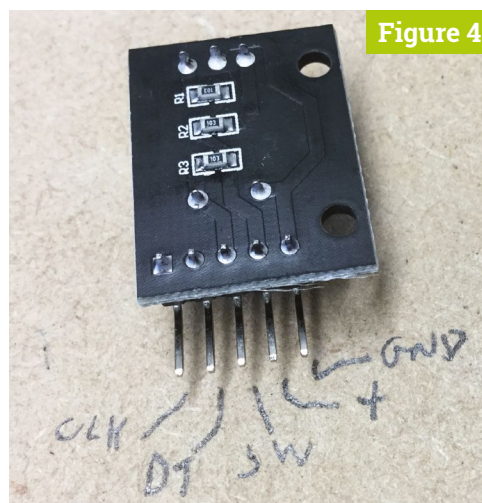


**Figure 3**



**Figure 4**

▲ **Figure 3** Wiring diagram of the Etch-A-Sketch

◄ **Figure 4** Make wiring easy by drawing on the reverse of the lid

θ increases with a clockwise motion while r is defined by the drawing window size, instead of being normalised to 1.0.

### 06 Software

You can find the full software listing on out GitHub page: **magpi.cc/pibakery**. It is written under the Pygame framework and the screen is shown in **Figure 6**. The right-hand control panel is a bitmap drawn only once, as is the grid. Over that is plotted another bitmap, which is the drawing itself, with a transparent background, and a very small bitmap is used for the cursor. Finally, various current variables are overlaid on the control panel. This makes for an efficient updating of the screen. One surprise was that a sleep delay, before events were checked, actually made the system more responsive.

### 07 Operation

The software starts in the sketch mode. Just like a real Etch-A-Sketch, moving the knobs will trace the path on the screen, while the r and θ values are updated to reflect the plot position. Unlike a real Etch-A-Sketch, if you click on the word 'Move' in the control panel, a box is drawn round it to show you are in the move mode, and the cursor will not leave a trail. This allows drawings to be made in separate parts of the

▼ **Figure 5** The grid dimensions of the Etch-A-Sketch software
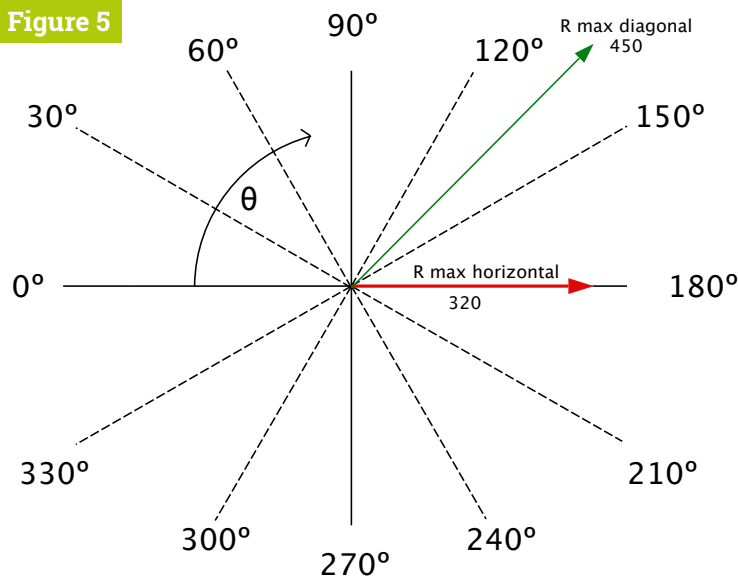
**Figure 5**
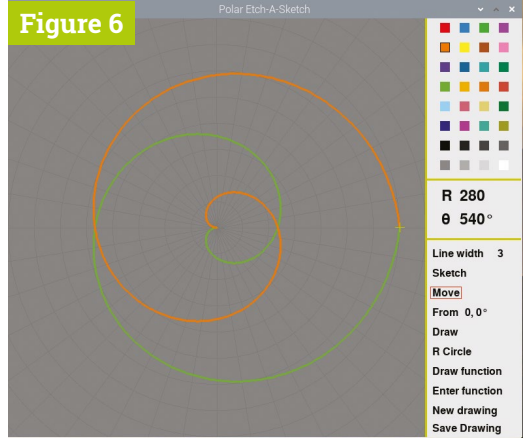
▲ **Figure 6** Screen dump of the user interface

screen. Pressing the rotary encoder's built-in button adds or subtracts 360° from the current θ value.

### 08 More controls

Clicking on 'From' transfers the current r and θ values to this point, and then clicking on 'Draw' will draw a line to the current r and θ values from this stored point. Drawing between two points in polar space is ambiguous, as there are always two ways to go: clockwise and anti-clockwise. In this software, when start and end angles are between 0° and 360°, the path chosen is the shortest one. Shown in **Figure 7** (overleaf), we feel this is the most intuitive way it should behave. Clicking on 'R Circle' will draw a full circle centred on the origin at the current value of r.

> ❝ Drawing between two points in polar space is ambiguous ❞

### 09 Yet more controls

If you want to wipe the screen, click on 'New Drawing'. To save what you have drawn as a PNG file with a transparent background, click 'Save drawing' for a save dialog box. Clicking on 'Draw Function' draws a predefined function whose starting value is R = r sin(3). This is one of the so-called petal functions shown in **Figure 8**. Interestingly, when the number N is odd you get N petals, but when N is even you get 2N petals.
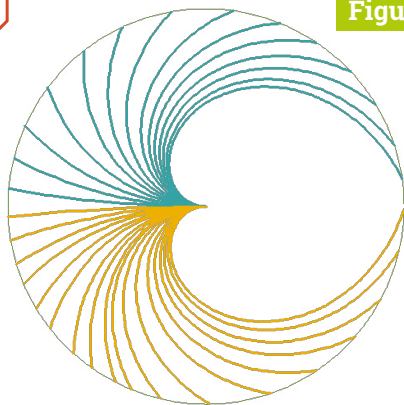
**Figure 7**



▲ **Figure 7** Drawing in polar space always takes the shortest route

## Top Tip 👍

**Controlling function plotting**

The start and end angle of a function plot can be changed using the **s**, **S**, **E**, and **e** keys.

### 10 Changing the function

You can change this built-in function by using the 'Enter function' option. When you use this, you need to enter the formula you want to plot as it would appear in a line of code, not in mathematical format. Therefore the built-in function looks like **r \* (maths.sin((th + cTh) \* 3))**, where **r** is the current r value, **cTh** is the current θ value, and **th** ranges from 0º to 360º. This is placed into a string through the Python console; you must click on this to make the console active and type in your formula.

▼ **Figure 8** The petal function from N=2 to N=9

**Figure 8**

### 11 Things to try

Here are some functions in mathematical format, and how you should enter them.

The petal function
$R = r * \sin(\theta * N)$
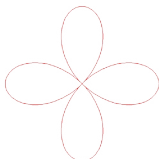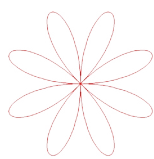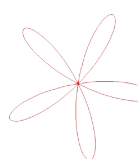


N = 2     N = 3     N = 4     N = 5
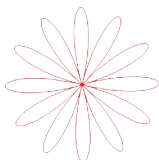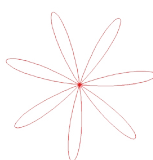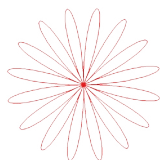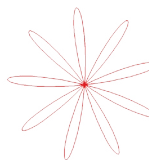
N = 6     N = 7     N = 8     N = 9
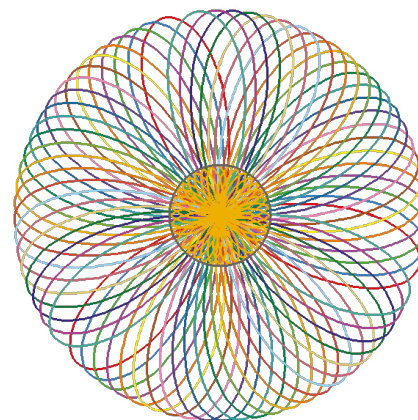


▲ Repeated drawing of a petal function at different angles, the centre being a smaller version of the outside

| Function | Program line |
|---|---|
| $R = 1 - \cos(\theta)$ | r – r*maths.cos(th) |
| $R = 1 + \sin(\theta)$ | r – r*maths.sin(th) |
| $R = 2 + 4\cos(\theta)$ | 2*r + 4*r*maths.cos(th) |
| $R = 1 + 0.1\sin(10\theta)$ | r + 0.1*r * maths.sin(10*th) |
| $R = \sin(\theta/2)$ | r*maths.sin(cTh + th/2) |
| $R = \sin(\theta/4)$ | r*maths.sin(cTh + th/4) |
| $R = \sin(\theta) + \sin^3(5\theta/2)$ | r*maths.sin(th) + r*maths.sin(5*th /2)**3 |
| $R = \theta.\sin(\theta)$ | r*th*maths.sin(th) |
| $R = 1 + 4\cos(5\theta)$ | r + r*4.maths.cos (5*(th + cTh)) |
| $R = 1/(\sqrt{\theta})$ | r / maths.sqrt(th + cTh) |

### 12 Drawing colours

When it came to the selection of drawing colours, we found it best to use a predefined palette, rather than take time trying to design our own. We used some of the many palettes defined in the library of colour palettes called Palettable, formerly known as brewer2mpl. Installing it is simple – in a Terminal window, type:

```
sudo pip3 install palettable
```

We used a choice of 32 drawing colours consisting of four palettes with eight colours in each. They are installed in the **loadPallet** function, and are easily changed if you want to use other ranges of colours. 🅼

Design

Code
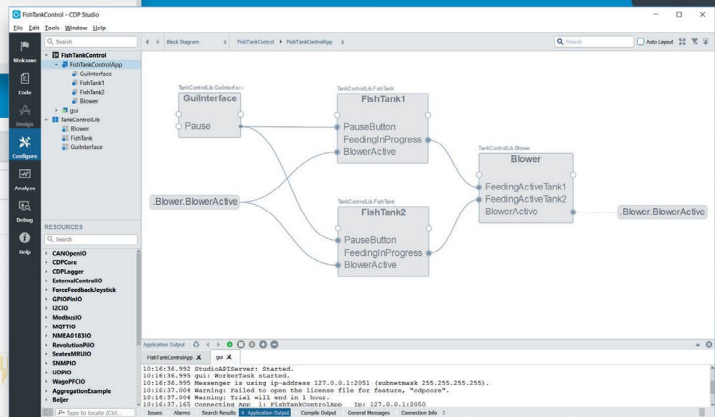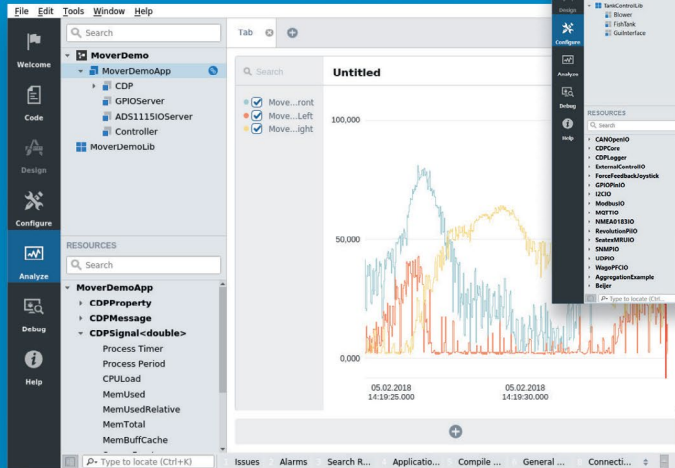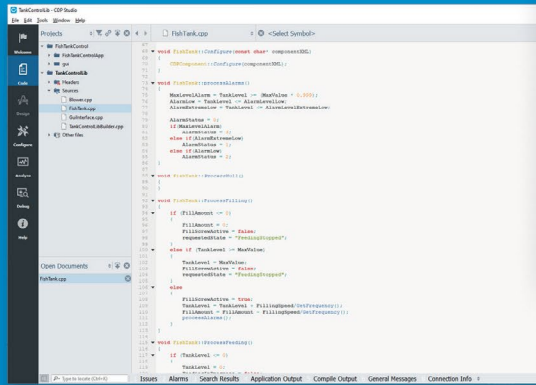
Analyze

Configure

# NEW GRAPHICAL EDITOR!
# CDP STUDIO
## 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.

CDPStudio

# Make a Sense HAT
# rainbow display

Bright colours put a smile on most peoples' faces. Join the rainbow parade with a light display, says **Rosie Hattersley**

**MAKER**

### Rosie Hattersley

Rosie has been writing about how to use tech longer than the concept of Googling has existed.

**@rosiehattersley**

**Y**ou've probably noticed splashes of colour appearing all over the place, particularly in people's windows as they share joyful rainbow drawings and artwork. You can join in with this trend using a Raspberry Pi to create your own amazing electronic display. All you need is a Sense HAT and just a few lines of Python code.

By following our step-by-step guide, you will discover how to light up rainbow and heart images on the Sense HAT's LED matrix, as well as showing a custom scrolling message. If you don't have a Sense HAT, you can still try out the code using the Sense HAT Emulator in Raspbian.

## 01 Why use a Sense HAT?

The Sense HAT sits on top of your Raspberry Pi and adds the ability to sense and report details about the world around it. It can measure noise, temperature, humidity, and pressure, for example. The Sense HAT can show readings on an 8×8 LED matrix, but first needs to be instructed, using Python code, what sort of data it should look for. The Sense HAT's visual display can also be programmed to show specific details including simple images. In this tutorial we'll look at how to control the LED matrix. Don't worry if you don't have a Sense HAT as you can use the Sense HAT Emulator and try out the code in Raspbian.
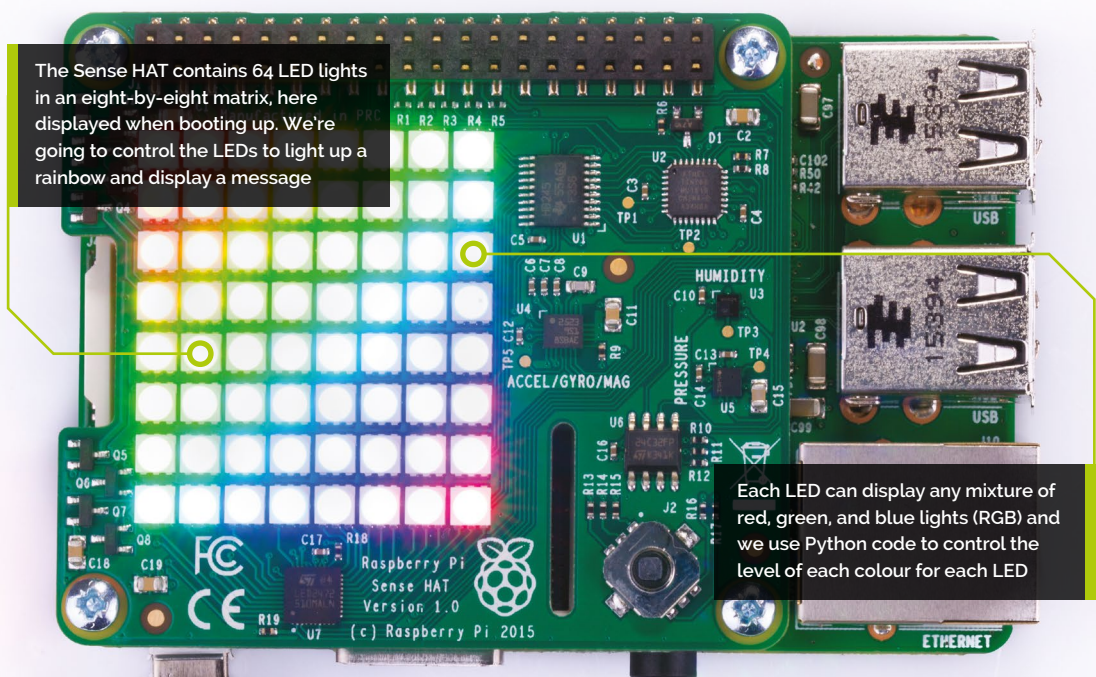


The Sense HAT contains 64 LED lights in an eight-by-eight matrix, here displayed when booting up. We're going to control the LEDs to light up a rainbow and display a message

Each LED can display any mixture of red, green, and blue lights (RGB) and we use Python code to control the level of each colour for each LED

## You'll Need

> Sense HAT (or the Sense HAT Emulator in Raspbian)

## 02 Attach the Sense HAT

Shut down your Raspberry Pi (if it isn't already) before attaching the Sense HAT to it. Hold the Sense HAT above your Raspberry Pi and line up the yellow holes at each corner with the corresponding ones on Raspberry Pi; make sure the header on Sense HAT lines up with the GPIO pins on Raspberry Pi. The white LED matrix should be at the opposite end of your Raspberry Pi from the USB ports. Gently push the Sense HAT onto Raspberry Pi's GPIO pins and then screw the two boards together with standoffs.

Now power up your Raspberry Pi as usual.

## 03 Open Thonny

We're going to use a program called Thonny to instruct our Sense HAT and tell it what to do. When Raspbian loads, select Programming from the top-left raspberry menu, then choose Thonny Python IDE. Click the New icon to open a new, untitled window. We need to get our program to recognise the Sense HAT module. To do this, type these two lines of code into the Thonny window:

```
from sense_hat import SenseHat
sense = SenseHat()
```

Click the Save icon and name your file **rainbow.py**.

## 04 Say something

From now on, Thonny will know to use the Sense HAT whenever you type 'sense.' followed by a '.' and a command. Let's get the Sense HAT to say hello to us. Add this line of code to line 4 in Thonny:

```
sense.show_message("Hello Rosie")
```

Of course, you can use your own name. Click Run and the letters should scroll across the LED display. If you get an error in the Shell at the bottom of the Thonny window, check your code carefully against the **sense_hello.py** listing. Every letter has to match.

## 05 Choose your colours

We are going to get the Sense HAT to light up a rainbow and display a heart. We do this with the `set_pixel()` function.
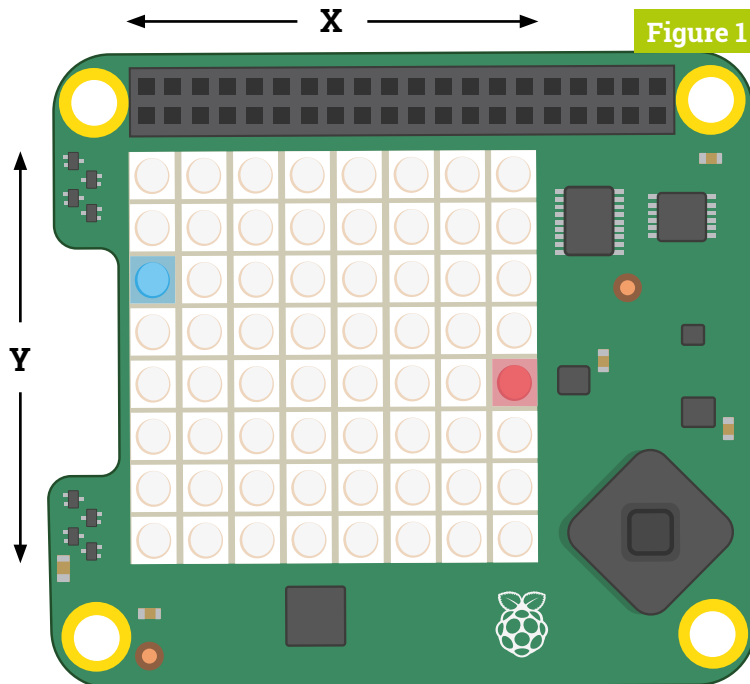


▲ **Figure 1** Sense HAT uses a co-ordinate system to locate each LED on the matrix. The blue light is at (0, 2) and the red light is at (7, 4)

### sense_hello.py

**DOWNLOAD THE FULL CODE:**
**magpi.cc/github**

> Language: **Python 3**

```
001.  from sense_hat import SenseHat
002.  sense = SenseHat()
003.
004.  sense.show_message("Hello Rosie")
```

> *Click Run and the letters should scroll across the LED display*

### sense_pixels.py

> Language: **Python 3**

```
001.  from sense_hat import SenseHat
002.  sense = SenseHat()
003.
004.  sense.clear()
005.  sense.set_pixel(7, 4, 255, 0, 0)
006.  sense.set_pixel(0, 2, 0, 0, 255)
```

We need to tell **set_pixel()** which LEDs we want to light up, using x and y variables to correlate to the axes of the Sense HAT's 8×8 grid of LEDs – see **Figure 1**. We also need to tell **set_pixel()** the colour using a three-digit code that matches the RGB (red, green, blue) value for each light. We're also going to start with **sense.clear()**, which clears any currently lit up LEDs. The 8×8 LED display is numbered 0 to 7 in both the x (left to right) and y (top to bottom) axes. Let's make the top-left LED red. Delete the **show_message()** line from your code and enter:

```
sense.clear()
sense.set_pixel(7, 4, 255, 0, 0)
```

The 7, 4, locates the pixel in the last column, and four rows down, and then the RGB value for red is 255, 0, 0 (which is 255 red, 0 blue, 0 green).

## sense_heart.py

> Language: **Python 3**

```
001.  from sense_hat import SenseHat
002.
003.  sense = SenseHat()
004.
005.  r = (255, 0, 0)      # red
006.  o = (255, 128, 0)    # orange
007.  y = (255, 255, 0)    # yellow
008.  g = (0, 255, 0)      # green
009.  c = (0, 255, 255)    # cyan
010.  b = (0, 0, 255)      # blue
011.  p = (255, 0, 255)    # purple
012.  n = (255, 128, 128)  # pink
013.  w =(255, 255, 255)   # white
014.  k = (0, 0, 0)        # blank
015.
016.  heart = [
017.      k, r, r, k, k, r, r, k,
018.      r, r, r, r, r, r, r, r,
019.      r, r, r, r, r, r, r, r,
020.      r, r, r, r, r, r, r, r,
021.      r, r, r, r, r, r, r, r,
022.      k, r, r, r, r, r, r, k,
023.      k, k, r, r, r, r, k, k,
024.      k, k, k, r, r, k, k, k
025.      ]
026.
027.  sense.set_pixels(heart)
```
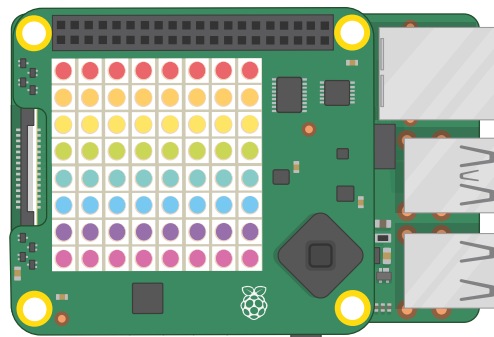


▲ We light up each row of the matrix a different colour of the rainbow using loops

Now let's add a third line, to light up another pixel in blue:

```
sense.set_pixel(0, 2, 0, 0, 255)
```

Check your code against the **sense_pixels.py** listing. Click Run to see your two dots light up. To add more colours, repeat this step choosing different shades and specifying different locations on the LED matrix.

## 🟩 It is much easier to create a set of variables for each three-number value 🟩

### 06 Colour variables

Working with the RGB values soon becomes frustrating. So it is much easier to create a set of variables for each three-number value. You can then use the easy-to-remember variable whenever you need that colour.

```
r = (255, 0, 0)      # red
o = (255, 128, 0)    # orange
y = (255, 255, 0)    # yellow
g = (0, 255, 0)      # green
c = (0, 255, 255)    # cyan
b = (0, 0, 255)      # blue
p = (255, 0, 255)    # purple
n = (255, 128, 128)  # pink
w = (255, 255, 255)  # white
k = (0, 0, 0)        # blank
```

Then you can just use each letter for a colour: 'r' for red and 'p' for purple, and so on. The # is a comment: the words after it don't do anything to the code (they are just so you can quickly see which letter is which colour).
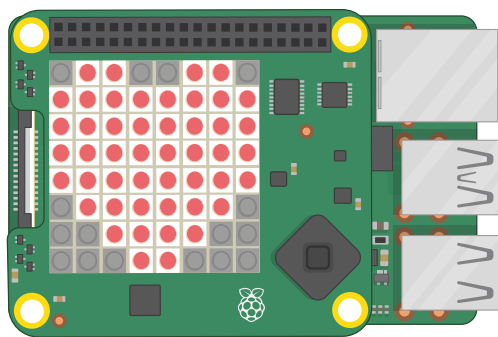
▲ Light up a pattern of RED leds on the Sense HAT matrix to form a heart.

## 07 Make a heart

Now we've got our colours, let's make a heart. The code in **sense_heat.py** enables us to draw a heart using 'r' letters for red lights, and 'k' for blank. This line of code then draws the heart:

```
sense.set_pixels(heart)
```

Check your code against the **sense_heart.py** listing. Click Run and you will see a heart.

## 08 Draw a rainbow

Creating a rainbow is a little tricker. We could just draw out each line, like a heart. But we want ours to build up line by line. For this we'll need to light up each pixel in a row, then pause. On the second row (underneath `from sense_hat import SenseHat`), add this line of code:

```
from time import sleep
```

We now have the `sleep()` function to slow down our program. To add a three-second pause before new pixels are lit up, you would type:

```
sleep(3)
```

You can change the number of seconds by changing the number in brackets.

## 09 Create blocks of colour

To turn an entire row on the LED matrix red, you could type:

```
for x in range(8):
    sense.set_pixel (x, 0, r)
```

## sense_rainbow.py

> Language: **Python 3**

```
001.  from sense_hat import SenseHat
002.  from time import sleep
003.
004.  sense = SenseHat()
005.
006.  r = (255, 0, 0)      # red
007.  o = (255, 128, 0)    # orange
008.  y = (255, 255, 0)    # yellow
009.  g = (0, 255, 0)      # green
010.  c = (0, 255, 255)    # cyan
011.  b = (0, 0, 255)      # blue
012.  p = (255, 0, 255)    # purple
013.  n = (255, 128, 128)  # pink
014.  w =(255, 255, 255)   # white
015.  k = (0, 0, 0)        # blank
016.
017.  rainbow = [r, o, y, g, c, b, p, n]
018.
019.  sense.clear()
020.
021.  for y in range(8):
022.      colour = rainbow[y]
023.      for x in range(8):
024.          sense.set_pixel(x, y, colour)
025.
026.      sleep(1)
```

Press **TAB** to indent the second line. This loops over each LED in the first row (x, 0), replacing x with the numbers 0–7.

We could repeat this, to set the second line to yellow, with this command:

```
for x in range(8):
    sense.set_pixel (x, 1, y)
```

…And repeat this step until you have eight different coloured lines of LEDs. But this seems a bit clunky.

## 10 Make a rainbow on command

We think it's neater to create two nested `for` loops for the x and y columns. Enter this code:

```
for y in range(8):
    colour = rainbow[y]
    for x in range(8):
        sense.set_pixel(x, y, colour)

    sleep(1)
```

### Top Tip 👍

**Sense HAT Emulator**

You can try out this tutorial using the Sense HAT Emulator in Raspbian. To install it, search for 'sense' in the Recommended Software tool. Note: If using this Emulator, you'll need to replace `from sense_hat` with `from sense_emu` at the top of your Python code (but not if using the online Sense HAT emulator).

# rainbow.py

> Language: **Python 3**

```python
001.  from sense_hat import SenseHat
002.  from time import sleep
003.
004.  sense = SenseHat()
005.
006.  r = (255, 0, 0)      # red
007.  o = (255, 128, 0)    # orange
008.  y = (255, 255, 0)    # yellow
009.  g = (0, 255, 0)      # green
010.  c = (0, 255, 255)    # cyan
011.  b = (0, 0, 255)      # blue
012.  p = (255, 0, 255)    # purple
013.  n = (255, 128, 128)  # pink
014.  w =(255, 255, 255)   # white
015.  k = (0, 0, 0)        # blank
016.
017.  rainbow = [r, o, y, g, c, b, p, n]
018.
019.  heart = [
020.         k, r, r, k, k, r, r, k,
021.         r, r, r, r, r, r, r, r,
022.         r, r, r, r, r, r, r, r,
023.         r, r, r, r, r, r, r, r,
024.         r, r, r, r, r, r, r, r,
025.         k, r, r, r, r, r, r, k,
026.         k, k, r, r, r, r, k, k,
027.         k, k, k, r, r, k, k, k
028.         ]
029.
030.  while True:
031.      sense.clear()
032.
033.      sense.set_pixels(heart)
034.      sleep(3)
035.
036.      for y in range(8):
037.          colour = rainbow[y]
038.          for x in range(8):
039.              sense.set_pixel(x, y, colour)
040.
041.          sleep(1)
042.      sleep(3)
043.
044.      sense.show_message("THANK YOU NHS!", text_colour =
      w, back_colour = b)
045.      sleep(3)
```

Take a look at the **sense_rainbow.py** listing to see how the code should look. Click Run and a rainbow will build up line by line.

**11 Write a message**

Writing messages in Python to show on your Sense HAT is really straightforward. You just need to decide on a few words to say and type them into a `sense.show_message()` command, as we did right at the start.

You can easily specify the text and background colours, too. Choose a contrasting colour for the background. To use your colours in a message, type: `sense.show_message("text here" text_colour = ` , followed by the colour value you chose for the text, `back_colour = ` , followed by the colour value you chose for the background.

Since we have already defined several colours, we can refer to any of them by name in our code. If you haven't already consulted our rainbow colour list, take a look at the **sense_rainbow.py** listing and add them to your code.

For instance, type:

```python
sense.show_message("THANK YOU NHS!",
text_colour = w, back_colour = b)
```

Click Run and your rainbow, followed by your message, should appear on your Sense HAT's display.

**12 Bring it together**

We're now going to take the three things we have created – the heart, the rainbow, and the text message – and bring them together in an infinite loop. This will run forever (or at least until we click the Stop button).

```python
while True:
```

All the code indented underneath the `while True:` line will replay as a loop until you press Stop. Inside it we will put our code for the heart, rainbow, and text message.

By now, you will be able to see the potential of making patterns to display on your Sense HAT. You can experiment by making the colours chase each other around the LED matrix and by altering how long each colour appears.

Enter all the code from **rainbow.py** and press Run to see the final message. M
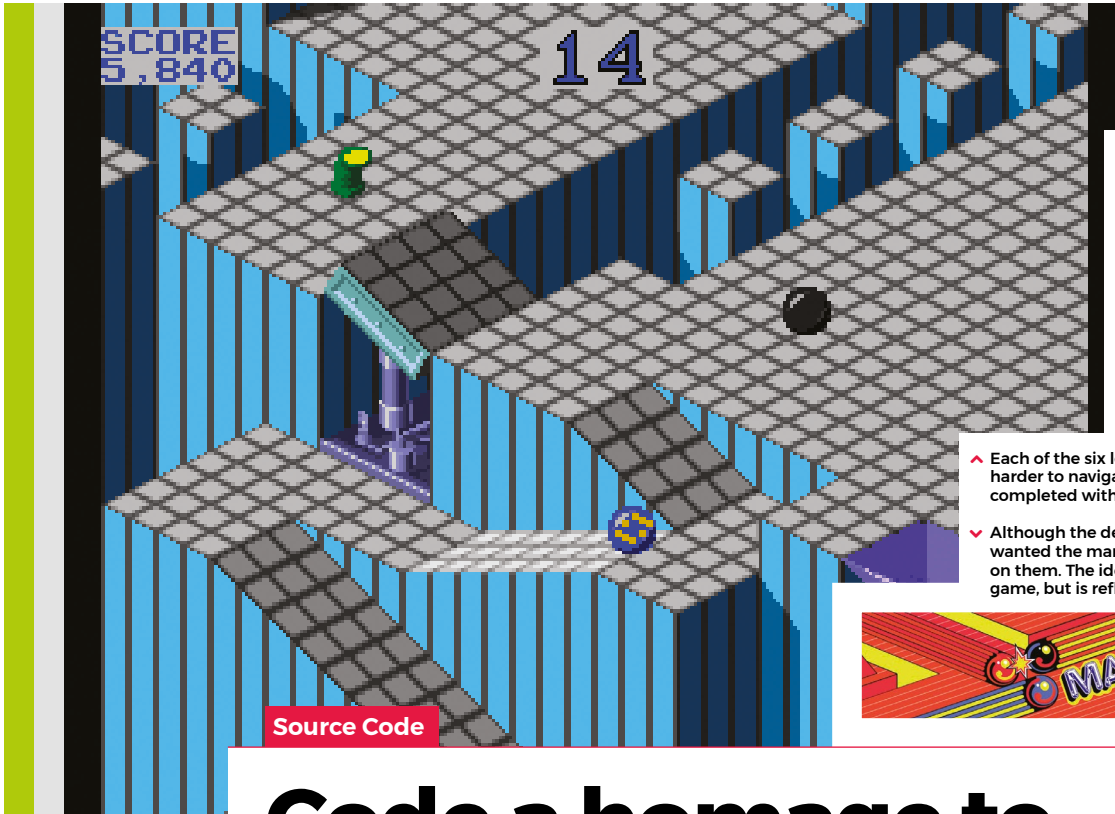
**Source Code**

# Code a homage to Marble Madness

Code the map and movement basics of the innovative marble-rolling arcade game

**AUTHOR**
**MARK VANSTONE**

itting arcades in 1984, Atari's *Marble Madness* presented a rather different control mechanism than other games of the time. The original arcade cabinet provided players with a trackball controller rather than a conventional joystick, and the aim was to guide a marble through a three-dimensional course in the fastest possible time. This meant that a player could change the angle and speed of the marble as it rolled and avoid various obstacles and baddies.

During development, designer Mark Cerny had to shelve numerous ideas for *Marble Madness*, since the hardware just wasn't able to achieve the level of detail and interaction he wanted. The groundbreaking 3D display was one idea that made it through to the finished game: its prerendered, ray-traced isometric levels.

*Marble Madness* was the first game to use Atari's System 1 upgradeable hardware

platform, and also boasted the first use of an FM sound chip produced by Yamaha to create its distinctive stereo music. The game was popular in arcades to start with, but interest appeared to drop off after a few months – something Cerny attributed to the fact that the game didn't take long to play.

> ## "The ball physics are calculated from the grey-shaded heightmap"

*Marble Madness*'s popularity endured in the home market, though, with ports made for most computers and consoles of the time – although inevitably, most of these didn't support the original's trackball controls.

For our version of *Marble Madness*, we're going to use a combination of a rendered background and a heightmap in Pygame Zero, and write some simple physics code to simulate the marble rolling over the terrain's

flats and slopes. We can produce the background graphic using a 3D modelling program such as Blender. The camera needs to be set to Orthographic to get the forced perspective look we're after. The angle of the camera is also important, in that we need an X rotation of 54.7 degrees and a Y rotation of 45 degrees to get the lines of the terrain correct. The heightmap can be derived from an overhead view of the terrain, but you'll probably want to draw the heights of the blocks in a drawing package such as GIMP to give you precise colour values on the map.

The ball rolling physics are calculated from the grey-shaded heightmap graphic. We've left a debug mode in the code; by changing the debug variable to `True`, you can see how the marble moves over the terrain from the overhead viewpoint of the heightmap. The player can move the marble left and right with the arrow keys – on a level surface it will gradually slow down if no keys are pressed. If the marble is on a gradient

# Rolling marbles in Python

You'll need to install Pygame Zero – see **wfmag.cc/pgzero**. To run the code, **cd** to its directory, then **pgzrun marble.py**

```python
# Marble Madness
from pygame import image

HEIGHT = 570
WIDTH = 600
gameState = 0
marble = Actor('marble', center=(300, 45))
marbleh = Actor('marbleh', center=(300, 60))
marble.dir = marble.speed = 0
heightmap = image.load('images/height45.png')
# set debug variable below to True for debug mode
debug = False

def draw():
    if(debug):
        screen.blit("height45", (0, 0))
        marbleh.draw()
    else:
        screen.blit("map", (0, 0))
        if gameState == 0:
            marble.draw()
        else:
            if gameState == 2:
                screen.draw.text("YOU WIN!", center = (300,
300), owidth=0.5, ocolor=(255,255,255), color=(0,0,255) ,
fontsize=80)
                marble.draw()
            else:
                screen.draw.text("GAME OVER!", center = (300,
300), owidth=0.5, ocolor=(255,255,255), color=(0,0,255) ,
fontsize=80)
        screen.blit("overlay", (0, 0))

def update():
    if gameState == 0:
        if keyboard.left:
            marble.dir = max(marble.dir-0.1,-1)
            marble.speed = min(1,marble.speed + 0.1)
        if keyboard.right:
            marble.dir = min(marble.dir+0.1,1)
            marble.speed = min(1,marble.speed + 0.1)
        moveMarble()
        marble.speed = max(0,marble.speed - 0.01)

def moveMarble():
    global gameState
    ccol = getHeight(marbleh.x,marbleh.y)
    lcol = getHeight(marbleh.x-10,marbleh.y+10)
    rcol = getHeight(marbleh.x+10,marbleh.y+10)
    if ccol.r == 0:
        gameState = 1
    if (lcol.r < ccol.r or rcol.r < ccol.r):
        marble.y += marble.speed
        marble.speed += 0.03
    marbleh.x += marble.speed*marble.dir
    marbleh.y += marble.speed
    marble.x = marbleh.x
    marble.y = (marbleh.y*0.6)+((255-ccol.r)*1.25)
    marble.angle = marble.angle + marble.speed*marble.dir*-10
    if marble.angle > 0 : marble.angle = -50
    if marble.angle < -50 : marble.angle = 0
    if marbleh.y > 610: gameState = 2

def getHeight(x,y):
    return heightmap.get_at((int(x),int(y)))
```

on the heightmap, it will increase speed in the direction of the gradient. If the marble hits a section of black on the heightmap, it falls out of play and we stop the game.

That takes care of the movement of the marble in two dimensions, but now we have to translate this to the rendered background's terrain. The way we do this is to translate the Y co-ordinate of the marble as if the landscape was all at the same level – we multiply it by 0.6 – and then move it down the screen according to the heightmap data, which in this case moves the marble down 1.25 pixels for each shade of colour. We can use an overlay for items the marble always rolls behind, such as the finish flag. And with that, we have the basics of a *Marble Madness* level. Ⓦ



⌃ **In our sample level, you can control the movement of the marble using the left and right arrow keys**
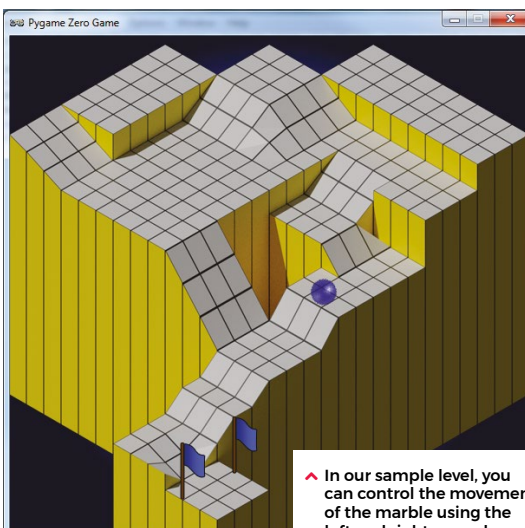
## Wireframe

This tutorial first appeared in Wireframe, our sister magazine that lifts the lid on the world of video games. Every issue includes tutorials and in-depth interviews, along with news and reviews of the latest indie and triple-A games.

To find out more, visit their website at **wfmag.cc**.

Check out their subscription offers at **wfmag.cc/subscribe**.

# AT HOME WITH THE
# INTERNET
# OF THINGS

## Upgrade your home – or just have some fun – with these amazing IoT projects

**W**hen we think of projects for the home, we often think of IoT and/or home automation. These concepts tend to interconnect, so we decided to have a look at instances of both types with ten amazing projects. You can also combine these with some of the recycling projects from our top ten list on page 80. For now, though, grab your Raspberry Pi and let's get building.

### What is the Internet of Things?

We'll be the first to admit that the term 'Internet of Things' (IoT) is not very descriptive. The concept is defined many ways, but usually involves an internet-connected device that has some level of automation or analysing that does not require human interaction. It can involve human interaction, though. Or be set up for you to then walk away and never think about it again.

Many features of IoT devices can be found in smart devices – such as smart lighting, smart fridges, smart heating, etc. Don't get too bogged down in descriptions and definitions like this, though – in the case of this feature, let's just have a bit of internet-connected fun!

### Getting started with IoT

Want to find out some of the basics of using the Internet of Things with a Raspberry Pi? Check out this blog from Raspberry Pi on some ways you can get started: **magpi.cc/iotstart**

# Controlling electrical sockets with
## Energenie Pi-mote

**magpi.cc/pimote**

**A core of IoT is being able to control when electronic devices get turned on.** Energenie is a wireless-connected mains plug socket adapter that you can control via radio waves. The Pi-mote is an add-on for Raspberry Pi that lets it control these Energenie devices – and, with a bit of code, automate them. A lot of simple yet cool IoT devices can be created this way, and Gordon Hollingworth, CTO of Raspberry Pi, introduces the concept in this blog piece.

▶ Amy Mather demonstrates using Pi-mote to Rick-roll the office

# Judd Treat Machine

**magpi.cc/juddtreat**

**We're sure many of you share your homes with a beloved pet, like Judd here. Judd likes treats (don't we all?), so his owner created a machine that dispenses treats for him after it receives an email.** It will take a picture and return it to you with a 'personal' thank-you from Judd for the treat. Wonderful. It uses several 3D-printed parts and some Arduino bits as well, but it's a very fun project.

▲ Judd is a two-year-old Viszla who loves his treats

# Temperature & Humidity Network Monitor

**magpi.cc/temphumid**

Weather monitoring is a great and simple way to use a Raspberry Pi – we even showed you how it can be used on a grander scale in *The MagPi* issue 92. This one is a bit smaller, but is no less powerful as it allows you to keep track of the atmosphere in your home, and outside as well with a bit of weatherproofing. It's an older tutorial, but not much has changed since those days.
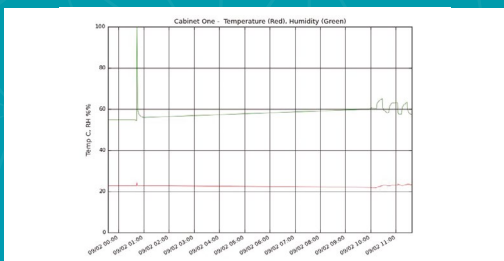


▲ View a graph of the temperature and humidity levels in your home

# Raspberry Pi 3 Baby Monitor

**magpi.cc/babycam**

Straight from the pages of our sibling publication, HackSpace magazine, this baby monitor is pretty high-tech. As well as streaming video, it has LED night-lights and a web interface. It's made up of several 3D-printed parts, as well as recycled bits and pieces from around the house. You can make yours a little less obtrusive as well if you wish.



▲ Keep an eye on the young ones with this baby monitor
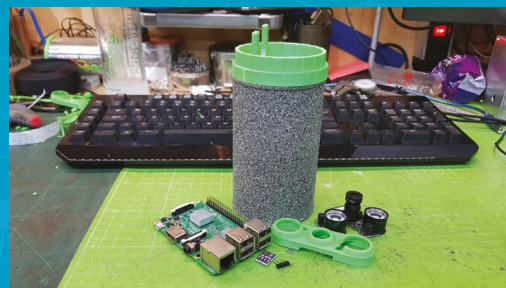
# Internet Doorbell

**magpi.cc/iotdoorbell**

**This is a classic IoT project – an internet-connected doorbell. Sure, there are plenty of commercial offerings, but making one yourself is part of the fun.** This version by Ahmad Khattab will call you, text you, and even stream video straight to your phone. The whole lot only costs about £17 as well, although you do have to supply your own Raspberry Pi. A Raspberry Pi NoIR Camera Module is also a good trick for making sure late-night visitors can be seen.
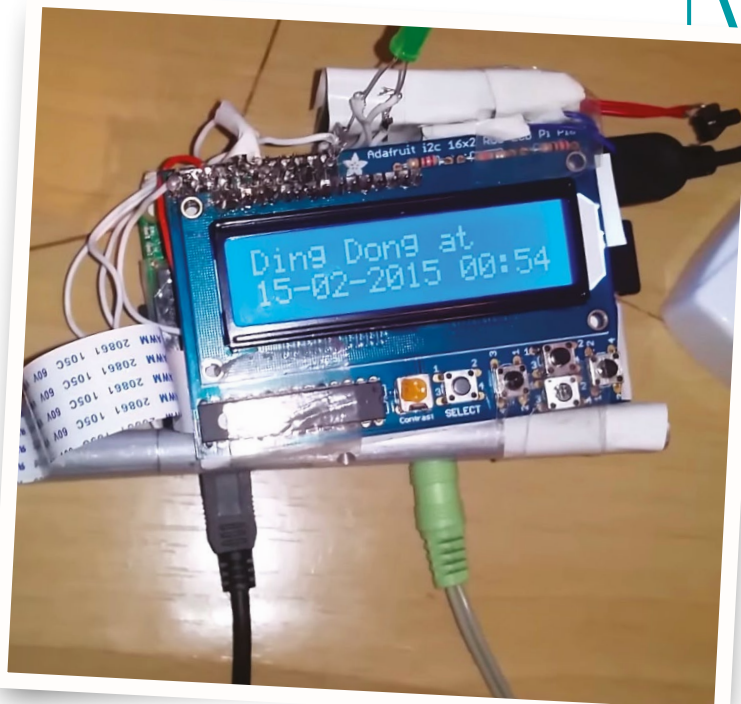
▲ Build your own internet-connected doorbell

# PiHue

**magpi.cc/pihue**

**Controlling lights gets easier and easier as companies get on board with the trend.** A couple of years ago, we had Tim Richardson of CamJam and Pi Wars fame write about this project. It uses a Pimoroni Touch pHAT to control the colour of the lights and is generally very fun. All it requires is the right hardware and a bit of code trickery to get it working.

▲ Change the colour of your lights with the touch of a button

# Internet-connected Shop Stereo

**Internet radios are an amazing way to listen to your favourite radio stations wherever you are!** They're also prime IoT project material, much like this cool-looking, laser-cut stereo. The creator made it due to the amount of sawdust he was creating in his shop, thinking it may not be conducive to the continued operation of his laptop. This is actually a fairly simple project in terms of electronics, and you can create any kind of casing for it you wish.





▲ A cool casing for this internet radio

# Automated Plant Watering

**Plant-watering projects are fun, and this makes great use of IoT by allowing you interact with it via a web browser on your local network – with a Raspberry Pi controlling the pump.** We appreciate how, besides the necessary components, there's a lot of recycling of stuff from around the house. We may have to employ this for our balcony flower beds we keep forgetting about…



▲ Keep your plants watered without lifting a finger

# Internet Monitor

**A simple project with only one use – however, considering that we'd need this at least once or twice a day, we think it's a very worthwhile and useful piece.** It's pretty involved as well, with a very well-thought-out circuit system so that the giant traffic lights can work properly with Raspberry Pi. The laser-cut wooden box is also very cool, although like other projects on here, you can always try a different case.
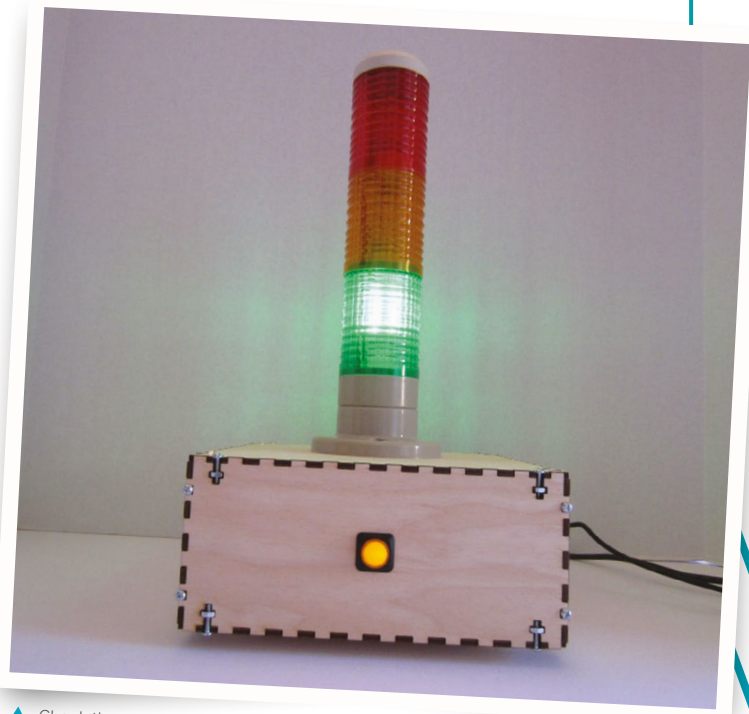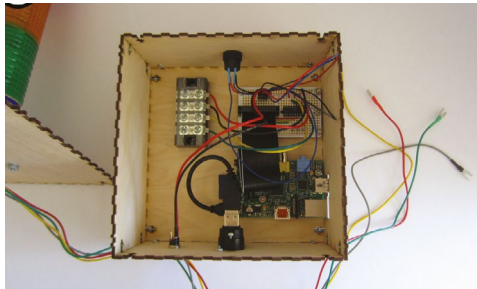




▲ Check the status of the internet with a traffic light

# Casio Pi Portable CCTV Monitor

**CCTV setups are actually pretty easy to do with Raspberry Pi, but Martin Mander has created a far more interesting way to view it by upcycling an old portable TV to display the picture from any security cameras.** The whole thing is very custom, although it does use the original LCD display that came with the 1997 device. While you may not have a portable TV to hand, you can always find other things and use this project for inspiration!



▲ Bringing an old pocket TV back to life

# ZX Spectrum
# Next Accelerated

Raspberry Pi Zero adds audio and realistic loading to this reimagined ZX Spectrum computer. By **Lucy Hattersley**
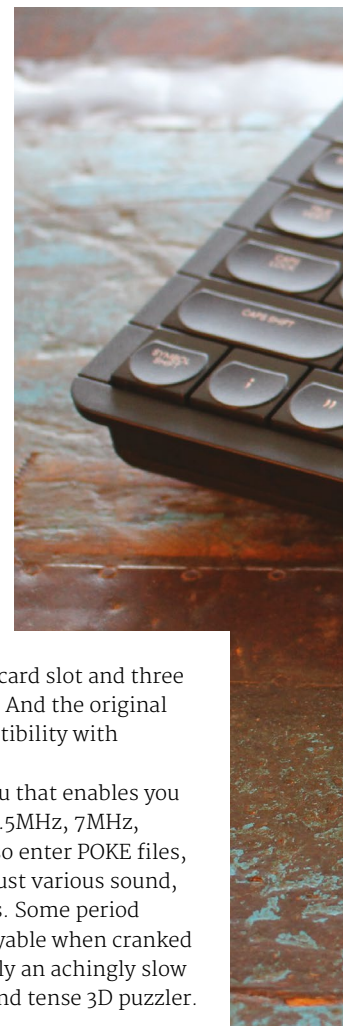
## SPECS

**PROCESSOR:**
Z80 on Xilinx Spartan-6 XC6SLX16 FPGA

**MEMORY:**
1MB RAM (expandable to 2MB internally)

**AUDIO:**
3 × AY-3-8912 audio chips with stereo output

**WIRELESS:**
ESP8266 WiFi module

**ACCELERATOR BOARD:**
Raspberry Pi Zero W

**F**ollowing several years of development, The ZX Spectrum Next Accelerated – a reimagining of the popular 1980s computer – has finally arrived. Part of the Spectrum's modernisation is a Raspberry Pi Zero acting as an accelerator, melding the power of a 1980s classic with the greatest computer of the modern age.

Inside the box you'll find the ZX Spectrum Next computer, a power supply, and a spiral-bound user manual which is pleasingly similar to the original but packed with information from BASIC programming to machine code.

The keyboard is a thing of beauty. The keys are responsive, although the layout is a bit weird after years of muscle memory bonded to PC.

It's packed with connections: HDMI and VGA for video out; 3.5mm ear and mic mini-jacks; PS/2 for keyboard and mouse; plus the mini HDMI and micro USB ports of Raspberry Pi Zero; and two 9B9 joystick ports (compatible with Kempston, Cursor, and ZX Interface 2 Protocols). To the left

of the device sits a full-size SD card slot and three buttons: Reset, Drive, and NMI. And the original Expansion port provides compatibility with classic hardware.

The NMI button opens a menu that enables you to flick between turbo modes: 3.5MHz, 7MHz, 14MHz, and 28MHz. You can also enter POKE files, browse memory banks, and adjust various sound, graphical, and memory settings. Some period games become wonderfully playable when cranked up to 28MHz: Sentinel, originally an achingly slow trudge, becomes a fast-paced and tense 3D puzzler.

## Z80 and beyond

The heart of the Spectrum Next is a Xilinx Spartan-6 XC6SLX16 FPGA (field-programmable gate array, **magpi.cc/spartan6**). FPGA isn't

▼ A range of modern and classic ports make for a versatile computer



DIGITAL VIDEO DEBUG | USB | DIGITAL VIDEO | EAR MIC | AUDIO OUT | RGB\VGA | KEYBO



▲ The keyboard, designed by Rick Dickinson, has come together perfectly

emulation: the programmable logic blocks create a perfect representation of the Z80 chip.

You can take the FPGA beyond the Z80 with processor cores. We turned our ZX Spectrum Next into a BBC Micro B and BBC Master using BeebFPGA (**magpi.cc/beebfpga**). Victor Trucco has made a range of Intel 8080 cores available, including MSX, NES, and Colecovision (**magpi.cc/specnextcores**).

A separate Anti-Brick core protect users from breaking the machine when messing around with cores, and can be used at any time to switch back to its original state.

## ❝ It's a great example of using the power of Raspberry Pi to add oomph to a project ❞

Alongside this sits a Raspberry Pi Zero, which enables you to load digital .tzx files as analogue cassette tape (screeches, loading screen, and all). It also brings SID (Sound Interface Device) support to the table, enabling better audio for games. There are plans afoot for Raspberry Pi Zero's micro USB port to act as a digital joystick port, and the mini HDMI output may be used down the line to add a second display. Beyond that, Raspberry Pi Zero adds a 1GHz CPU and 512MB of RAM to the hardware – plenty of extra headroom for ambitious game developers.
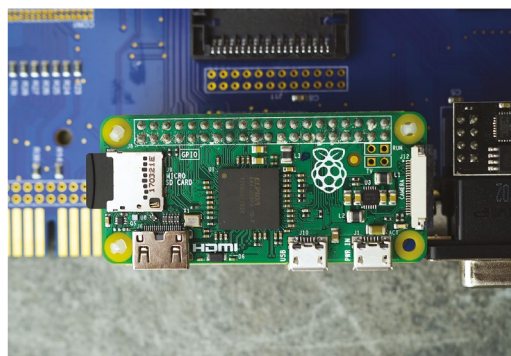


▲ Attached to the motherboard, a Raspberry Pi Zero acts as a sound system, enabling files to be loaded as if they were tapes

▲ Its classic design is very similar to that of the ZX Spectrum 128, but it packs far more power

We're impressed. From a design and build quality perspective, ZX Spectrum Next has achieved all we wanted from a new Spectrum. And it's a great example of using the power of Raspberry Pi to add oomph to a project. From a licensing and business perspective, managing to maintain this purity of focus while blending multiple open-source and proprietary software projects, all while juggling licensing owned by (to our count) 15 separate organisations including Amstrad/Sky, is seriously impressive stuff. Bravo, SpecNext, bravo! ◱

## Verdict

The ZX Spectrum Next is a lovely piece of kit. Well-designed and well-built: authentic to the original, and with technology that nods to the past while remaining functional and relevant in the modern age.

# 9/10

# Automation HAT **Mini**

## SPECS

**CONNECTIONS:**
3 × analogue inputs,
3 × buffered inputs,
3 × sinking outputs,
1 × relay

**TOLERANCE:**
Up to 24 V; up to 500 mA (total for sinking outputs), up to 2 A (relay)

**DISPLAY:**
0.96-inch IPS LCD

**BREAKOUT:**
Six-pin female header

Control your home automation projects with this tiny board. By **Phil King**

**S**uperseding Pimoroni's Automation pHAT, the Automation HAT Mini has the same Zero-size form factor (but can be used with any Raspberry Pi).** Although a little more expensive than its predecessor, it has the bonus of coming fully assembled and adds a tiny full-colour LCD which provides useful status info on connected inputs and outputs. Another nice addition is the six-pin female header on the underside, which has the same pinout as Pimoroni's extensive range of I²C mini breakouts (including numerous sensors), although you'll need an extender to connect one.

The board's main connections for inputs and outputs are 3.5 mm screw terminals. As well as a single relay, there are three analogue inputs, three buffered inputs, and three sinking outputs. All of these are tolerant to voltages up to 24 V – which is fine for controlling a plethora of household devices that typically have a 12 V or 24 V control board. But make sure you don't use the board to switch mains voltages!



▲ There are plenty of connections for inputs and outputs, and a tiny LCD can be used to show their status
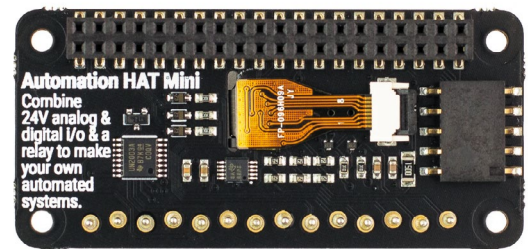


▲ The underside of the board, showing the female breakout header on the right

> ❝ The LCD provides useful status info on connected inputs and outputs ❞

### Inputs and outputs

Read via a 12-bit ADC, the analogue inputs have an accuracy of ±2%, which is fine for most purposes when you need to read a variable voltage. The digital inputs are used to tell if a connected device is on (signal above 3 V) or off (below 1 V).

For turning devices on and off, you have a choice of the three sinking outputs and the relay. For the former, you connect your device load on the ground side; a maximum of 500 mA can be sunk between the three outputs. With a tolerance of up to 2 A, the relay can be connected on the NC (normally closed) or NO (normally open) side, depending on whether your device spends more time switched on or off respectively.

Although created for the original Automation HAT/pHAT, Pimoroni's getting started guide (**magpi.cc/automationguide**) shows you how to use the board's Python library – instead of using the curl command, you'll need to clone it from GitHub (**magpi.cc/automationgh**) to get the up-to-date version. The relevant code examples show the status of inputs/outputs on the LCD, including bar graphs for analogue inputs.

If you're new to the world of home automation and its terminology, you should also check out Tanya Fish's excellent explainer on the Pimoroni blog: **magpi.cc/automationblog**. ◾

## Verdict

It's ideal for most home automation purposes, and we love the mini LCD, although if you need more relays you might prefer the full-size Automation HAT.

**9**/10

# CUSTOM PC

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

## THE MAGAZINE FOR
# PC HARDWARE ENTHUSIASTS

# ISSUE 201 OUT NOW

VISIT **CUSTOMPC.CO.UK** TO LEARN MORE

# 10 Best:

# Upcycling projects

Have some old tech lying around your home? Get inspired to recycle it...

**R**ecycling stuff in your house shouldn't just be limited to cardboard boxes and plastic bottles: it can also involve reusing tech that may be a little past its sell-by date. Here are ten amazing examples of reusing and upcycling some older tech. M



## ▲ Tomy Turnin' Turbo Dashboard Outrun Arcade

### Toy to life

Do you have any kids' toys lying around? Then maybe you can do something like Circuitbeard (aka Matt Brailsford): turn a toy steering wheel into a mini Out Run cabinet, complete with force feedback.

**magpi.cc/outrun**

## ▶ Cassette Pi

### Tapeless technology

A 'self-contained real-time notification scroller', according to upcycling extraordinaire Martin Mander – this Raspberry Pi Zero stuffed inside a clear cassette tape retrieves IoT connected notifications using IFTTT.

**magpi.cc/cassettepi**
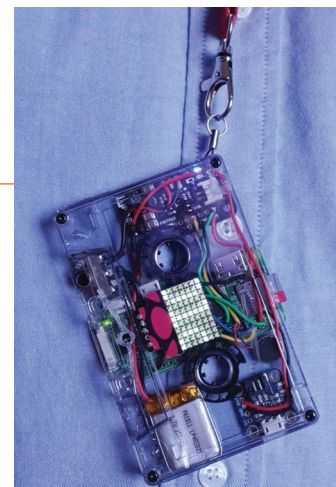


## ▼ RetroPie NES controller

### Console in a controller

We did this ourselves when Raspberry Pi Zero launched, albeit a bit more crudely. This version is much neater and allows you to use it with an original NES controller – if you have one spare, that is.

**magpi.cc/nespi**



## ◀ Victorian dresser magic mirror

### Antique wizardry

We featured this a long time ago in *The MagPi*, but we love the idea of taking the very cool magic mirror concept and putting in an antique dresser. An amazing melding of old and new.

**magpi.cc/magicdresser**

## ▲ Flash Cam

### Analogue to digital

This very retro camera has had its flash completely upgraded with a Raspberry Pi Zero, which allows it to take short video clips. The flash and camera still don't work, but it's got a great aesthetic.

**magpi.cc/flashcam**

## ▶ Thermal paper Polaroid

### Instant digital photography

Printing with thermal paper is something we see every now and then. However, fitting the entire system so neatly inside an old Polaroid camera is a feat in itself. Don't have an old Polaroid? Hack another film camera.

**magpi.cc/thermalpolaroid**



## ▼ Console Emulator TV



### Raspberry Pi Zero inside

Be careful with this one, as dismantling an old CRT TV can be hazardous. With caution and a good soldering iron, though, you can hack it so a Raspberry Pi Zero can play directly via the SCART/AV/Composite input. Perfect for older consoles.

**magpi.cc/emulatortv**

## ▶ Lego Raspberry PiBook

### Brick case

Lego cases used to be all the rage when Raspberry Pi was released, and this Lego laptop takes the concept to its logical conclusion. Lego is great for prototyping cases of all sizes and types, though.

**magpi.cc/pibook**





## ▲ ScrapPi

### Upcycled media hub

This media hub is more interesting than most – not just because it's housed in an old DVD player, but because it's able to read DVDs and other optical discs as well. A few hacks were involved to get this working, though.

**magpi.cc/scrappi**

## ▶ PiVidBox

### Upcycled media player

From last month's issue of *The MagPi*, if you don't fancy hacking an entire DVD player, you can always simplify the concept to play video files from old USB sticks and spare SD cards.

**magpi.cc/pividbox**
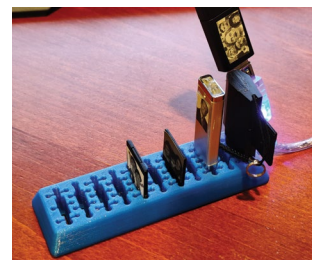
# Learn JavaScript
# with Raspberry Pi

JavaScript is one of the most popular languages out there. **PJ Evans** opens his code editor

## Beginner's Guide to Installing Node.js on a Raspberry Pi

**CREATOR**

**Dave Johnson**

Price:
Free

**magpi.cc/ installnodejs**

**JavaScript is a programming language.** To be useful, something has to run that code. JavaScript is most commonly used in web browsers, to make websites dynamic and easier to use. You can also run JavaScript on the command line, just like Python. The application that handles this is called Node.js. Unlike Python, Node.js is not installed in Raspbian by default and there are different versions for different hardware. Luckily, Dave Johnson has written an excellent guide to take you through the process.

What sets Dave's guide apart from others is how current it is. Node.js is a fast moving-platform with releases coming thick and fast. At time of writing, the guide is covering the very latest version of Node.js, 13. Sadly, although you can install Node.js from APT without any fuss, you will get a much older version and miss out on many performance enhancements for the Raspberry Pi range. Dave's guide explains how to add the NodeSource repository and get the latest and greatest Node.js versions on Raspbian. 𝕄

## Essential resources

Learning JavaScript and Node.js? You're going to need these…

### NPM
Node Package Manager is like Python's pip for Node. Literally thousands of libraries to enhance your project can be installed in a single command. Always start here to avoid reinventing the wheel.
**npmjs.com**

### MDN WEB DOCS
This free resource from Mozilla is like the Library of Alexandria for internet development, amongst which you'll find comprehensive documentation on all versions of JavaScript with examples and browser support information.
**magpi.cc/jsguide**

### STACK OVERFLOW
Every developer's little secret. A gigantic Q&A site for many subjects including Raspberry Pi, JavaScript, and Node.js. Chances are the answer to your question is here somewhere.
**magpi.cc/stackoverflowjs**

# NodeSchool

**CREATOR**

**NodeSchool**

Price:
Free

**nodeschool.io**

**NodeSchool is an astounding resource for learning not only JavaScript but many of the tools you need to get the most out of the experience.** The



tutorials available are installed as a series of command-line applications that allow you to create your code for real and then have it verified automatically. This means you are breaking out of the sandbox and coding for real.

Starting with the very basics of JavaScript, you soon progress to the Node.js environment and then on to more advanced topics such as data streaming. You can even build your own workshops! Furthermore, NodeSchool organises and runs workshops and regular groups all over the world, so beginners can get support and friends. A perfect resource for those who prefer to collaborate. M
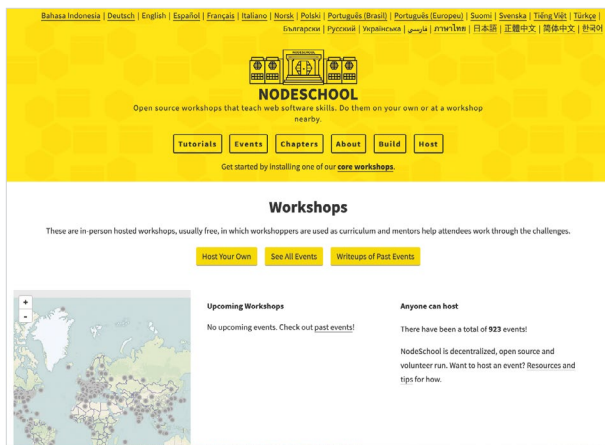
# Node.js in Action (2nd Edition)

**CREATOR**

**Alex Young, Bradley Meck, Mike Cantelon**

Price:
£18

**magpi.cc/nodeinaction**

**Manning Publications has been a supporter of Node.js since its inception in 2009.** This had been showing a bit with the original *Node.js in Motion* book getting a bit out of date as Node.js versions raced ahead. Now the book has been brought up to date to be in line with modern Node.js practices. While it doesn't cover the most recent developments (Node.js progresses so rapidly, it would be impractical), what the book excels at is giving a thorough grounding in how Node.js works under the bonnet, really helping the reader understand some of the key concepts that make Node.js different from browser-based JavaScript. There's also a video course available (you may recognise the presenter). M
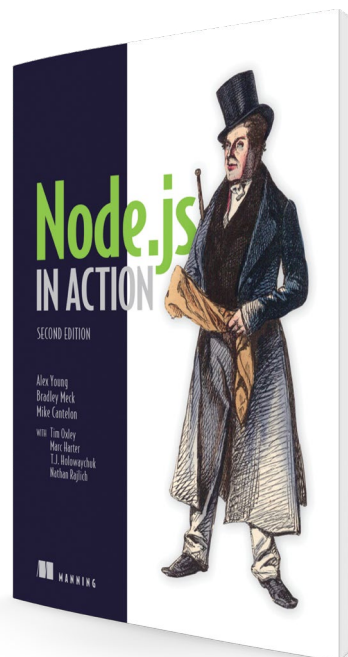


## Amazing frameworks

Frameworks allow your JavaScript code to do amazing things

### EXPRESS.JS
Think writing a web server from scratch is difficult? It certainly was until Express.js came along. This library and command-line tool can scaffold a full working server in seconds, perfect for web apps and APIs.
**expressjs.com**
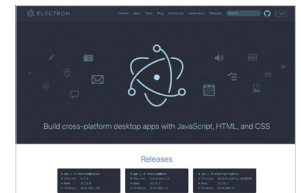


### ELECTRON
Electron takes the components of your web app, the JavaScript, HTML and CSS, and packages it all with a web browser to create a standalone application that can be compiled for many different platforms.
**electronjs.org**

### SEQUELIZE
If databases are your thing but you find SQL a bore, this object-relational mapping (ORM) framework makes creating, managing, and using databases a joy. Supports many platforms and integrates with Express.
**sequelize.org**

# 8 Bits and a Byte

The YouTube duo tell us about making amazing Raspberry Pi project videos

> Name **Dane and Nicole** | > Community role **YouTubers** | > Website **magpi.cc/8bitsbyte**

**Y**ou can find a huge variety of Raspberry Pi projects online. A lot of them are useful and cool, some of them great for beginners, and others are just extremely wacky. Dane and Nicole from 8 Bits and a Byte are on the wacky side.

"We build all kinds of stuff, including internet-controlled robots, wearables, talking appliances, data visualisations, and self-driving cows," they tell us. "On our YouTube channel we post montages of how we built it and a demonstration of the projects, whereas we share a step-by-step guide and the code on websites like Instructables and Hackster."

The name comes from a bit of a pun on bits and bytes, and they like to think of each other as both four bits half a byte. You can probably do the maths yourself on that.

**How did the channel start?**
The idea of the channel started when Dane finished his first Raspberry Pi hobby project. He loved the project, but he also wanted to take it apart to be able build something else: his next project. We thought it would be a shame to just take it apart and be done with it. Around the same time, Nicole had just finished a video editing course. We were able to borrow a camera and we thought we'd try to document the project, but at this point it was just for ourselves. The video wasn't too bad and we also realised we would be making more of these projects in the future. That's when we had the idea that it could be fun to share them with others.

From thereon it grew; we indeed built more things, although very infrequently at

▼ A classic Monty Python sketch made into a robot face that argues with you. Poorly
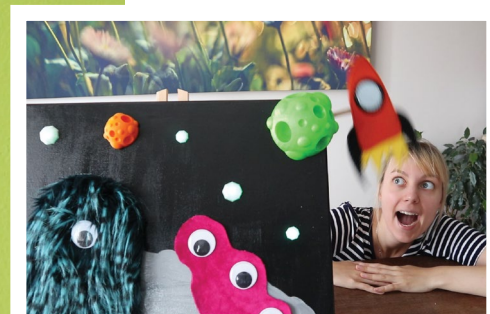




▲ This painting is both interactive *and* has space monsters, so it's definitely the greatest

▶ This hamster wheel has IoT capabilities because why not?

## 8-bit projects

### Ted the Talking Toaster

"It was great seeing the project come to life and the build was also very diverse, which made it particularly interesting."

**magpi.cc/tedtoaster**

### Mooomba the Cow Roomba

"We took the Mooomba to multiple events and somehow people just fell in love with her!"

**magpi.cc/mooomba**

### The Historic Voicebot

"[This] was made as a part of [Nicole's] final work / bachelor thesis, so there was a lot of research and design behind the concept."

**magpi.cc/voicebot**



first. Eventually we got our own camera, so we could also document the building process and not just film the finished result. We also started writing tutorials and sharing the code, which is not only nice for others but it has actually been really beneficial for us too. When you're working on something, it makes it very easy to track back and find how you did this same thing before. It feels a bit like cheating off your own previous work, but it's very helpful and we do it all the time.

**When did you learn about Raspberry Pi?**
Oh that's a tough question, we're not sure when we first heard about them. Dane came into contact with Raspberry Pi for the first time during his internship about four years ago, where he used one to build a prototype. Nicole heard about Raspberry Pi from Dane, who couldn't stop talking about them.

> **" Nicole heard about Raspberry Pi from Dane, who couldn't stop talking about them "**

**What was your first Raspberry Pi project?**
For both of us, the first Raspberry Pi projects were work-related prototypes. Whilst working with them, we realised Raspberry Pi [computers] were a lot of fun and that there was so much more that you could do with them. It didn't take long before we bought one to mess around with at home.

Dane's first hobby project with a Raspberry Pi was Tata, a cute and fluffy remote-controlled pet that could drive around and make all kinds of monster noises. Nicole's first Raspberry Pi-powered hobby project was a device that tracked how far our hamster Harold ran in his wheel. M

# This Month in
# Raspberry Pi

# #MonthOfMaking

You've been making and sharing – here are some of our faves!

**T**he **#MonthOfMaking is now over, and it was a lot of fun!** People from around the world shared their projects with us for #MonthOfMaking and also #MagPiMonday! Here are just some of the amazing things we saw… **M**

**01.** This Raspberry Pi laptop build seems to have just about everything! When will it be able to make tea?

**02.** This looks very handy for stargazing!

**03.** We always love to see some ham radio projects using a Raspberry Pi

**04.** We definitely think this new UI is cooler

**05.** Making sweet music with a lot of hacking

**06.** What clip would you use to wash your hands? We like the song from *OK K.O.! Let's Be Heroes*

**07.** A useful device for darkrooms (do people still have those?) or recording studios

**08.** The semaphore clock has been upgraded with nicer fonts. It looks even snazzier now

**09.** This is a very cool way to upcycle an old power supply!

**01**

**Martin Parker**
@Mr_MartinParker

Replying to @TheMagPi
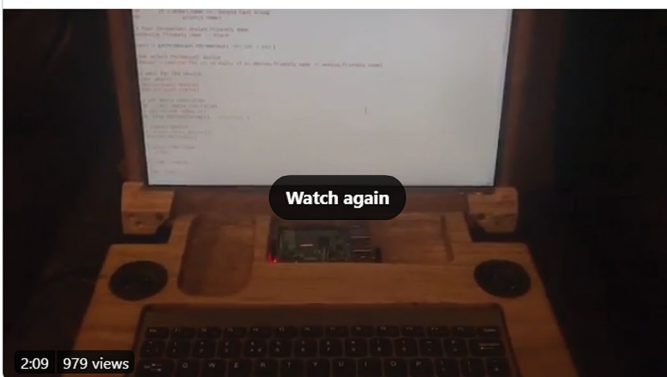
This is part of a project I'm starting for a security system and other controls using 2 Raspberry Pi a touchscreen and the pi's will communicate over my network to trigger relay for lighting and heating control.

**Martin Parker** @Mr_MartinParker · Mar 29
A project I've been thinking about for a while, this is a test for voice announcements on my google home mini's. This test is to sound on living room only as Oliver's in bed but can be set to all or a group of speakers.
See follow up comments for more information. @BIG_PESH
Show this thread

Watch again

2:09   979 views

**02**

**Edwin Wisse**
@edwinwisze

Replying to @TheMagPi

Last week I made this small map showing the planets in the sky over my location using a raspberry zero with an e-ink display. Now working on showing stars on the map.



**03**

**David Taylor**
@gm8arv

Replying to @TheMagPi

Updated a couple of #mmdvm #hamradio hotspots to the most recent Pi-Star 4.1. As this involved a Stretch to Buster upgrade, it meant a fresh microSD card format. One unit has an external antenna, the other's antenna is in the case. Yes, some dusting would be a "good idea"!



**06**

**James @raspjamberlin**
@monkeymademe

Replying to @TheMagPi

I made a hand washing timer for the kids using a sound bite from @npr show 'wait wait don't tell me'... ir Sensor, speaker phat and led shim from @pimoroni yes!!! I am going crazy locked up in the house!



**05**

**Blitz City DIY**
@BlitzCityDIY

Replying to @TheMagPi

I started working on the XyloPi again
twitter.com/blitzcitydiy/s...
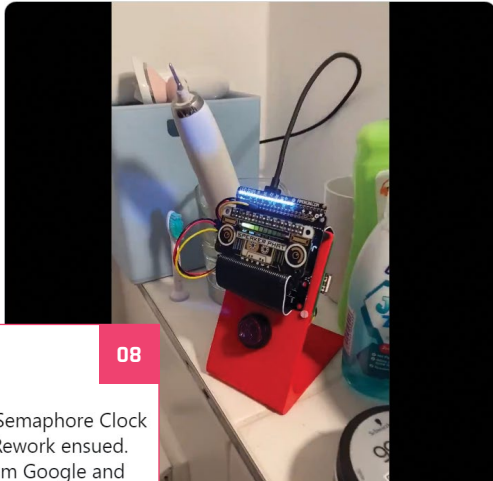
**Blitz City DIY** @BlitzCityDIY · Mar 22
Simple test to make sure everything is still noiding 🎹🔌



**04**

**GermanJMC**
@germanjmc

Replying to @TheMagPi

I just did an update to the web UI of my heater manager, adding CSS to the new one and now is cooler! (or warmer?)



**08**

**RaspberryPint**
@RaspberryPint

Replying to @TheMagPi

I made the foolish error of sharing my Semaphore Clock project with a friend who loves fonts. Rework ensued. Ended up with a Hammersmith One from Google and more space to add. Likely weather from the Met Office Datapoint API. GIF from before - before/after photos in reply.



**07**

**Jim Bennett**
@jimbobbennett

Replying to @TheMagPi

Wrapped up this one last week.

Would love to write it up for MagPi! Let me know if it's of interest.

**Jim Bennett** @jimbobbennett · Mar 19
🤔 One problem with #WFH when schools are closed is ensuring your family don't interrupt your meetings.

🚦 So I built a busy light with a @Raspberry_Pi Zero, @pimoroni 🎩 pHAT and Azure #IoTCentral and #LogicApp linked to my @Office365 calendar!
👇
github.com/jimbobbennett/...



0:08   1.2K views

**09**

**Gam3t3chElectronics**
@Gam3t3ch

Replying to @TheMagPi

My converted PC Power supply to Raspberry Pi Project case. With dual speed fan controller and fan voltage meter. Was a fun Sunday project will probably turn it into a nas server and add some light to it soon. Build video coming soon.

# Pi Wars Turkey

Pi Wars Prime may have had to go online, but the Turkish version managed to go ahead

**W**ho doesn't love a bit of friendly robot battling? The organisers of Pi Wars Turkey got in contact with us, asking if we knew about them. We did! So they showed us some excellent photos from the event. M



**01.** Prep work and coding are an important skill for building a Pi Wars competitor

**02.** The attendee count was huge! We love the variety of robots on display at the front as well

**03.** This obstacle course has some classic hazards, such as the turntable and some gravel

**04.** Pi Noon is popular everywhere

**05.** As usual, robots had to take part in multiple, varied events
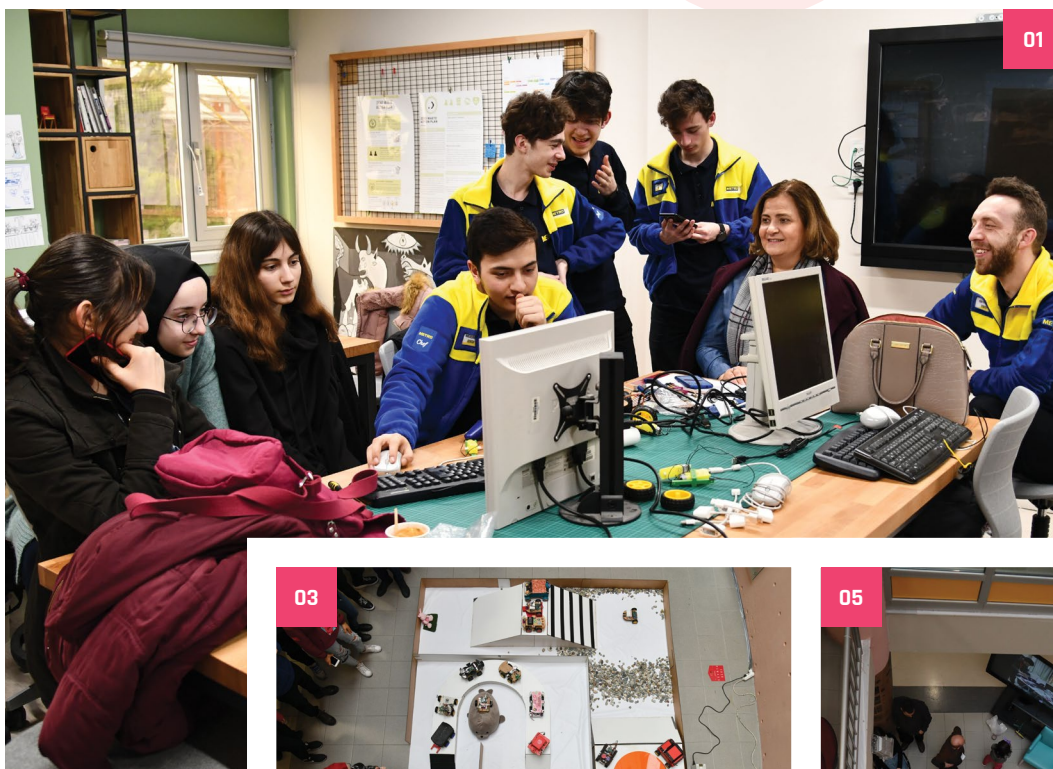
# Crowdfund **this!** Raspberry Pi projects you can crowdfund this month
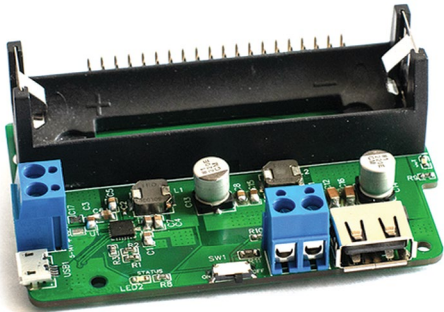


## Power Pi

If your Raspberry Pi is doing critical work that would not react well to a power outage, blip, or surge, then having an uninterruptible power supply is a great way to add some level of security to its continued operation. This one could provide up to an hour of extra power.

▶ **kck.st/2JTul0F**



## Raspberry Pi NFC/ Audio Skirt

This cool little add-on gives Raspberry Pi a couple of extra features: an NFC/RFID reader, which is very cool, and two 3.2 watt amplifiers for enhanced sound! We've seen a couple of projects that meld both NFC and audio, so it could be great for that!

▶ **kck.st/2ViLRk0**

**CROWDFUNDING A PROJECT?**
If you've launched a Raspberry Pi-related project, let us know! **magpi@raspberrypi.org**
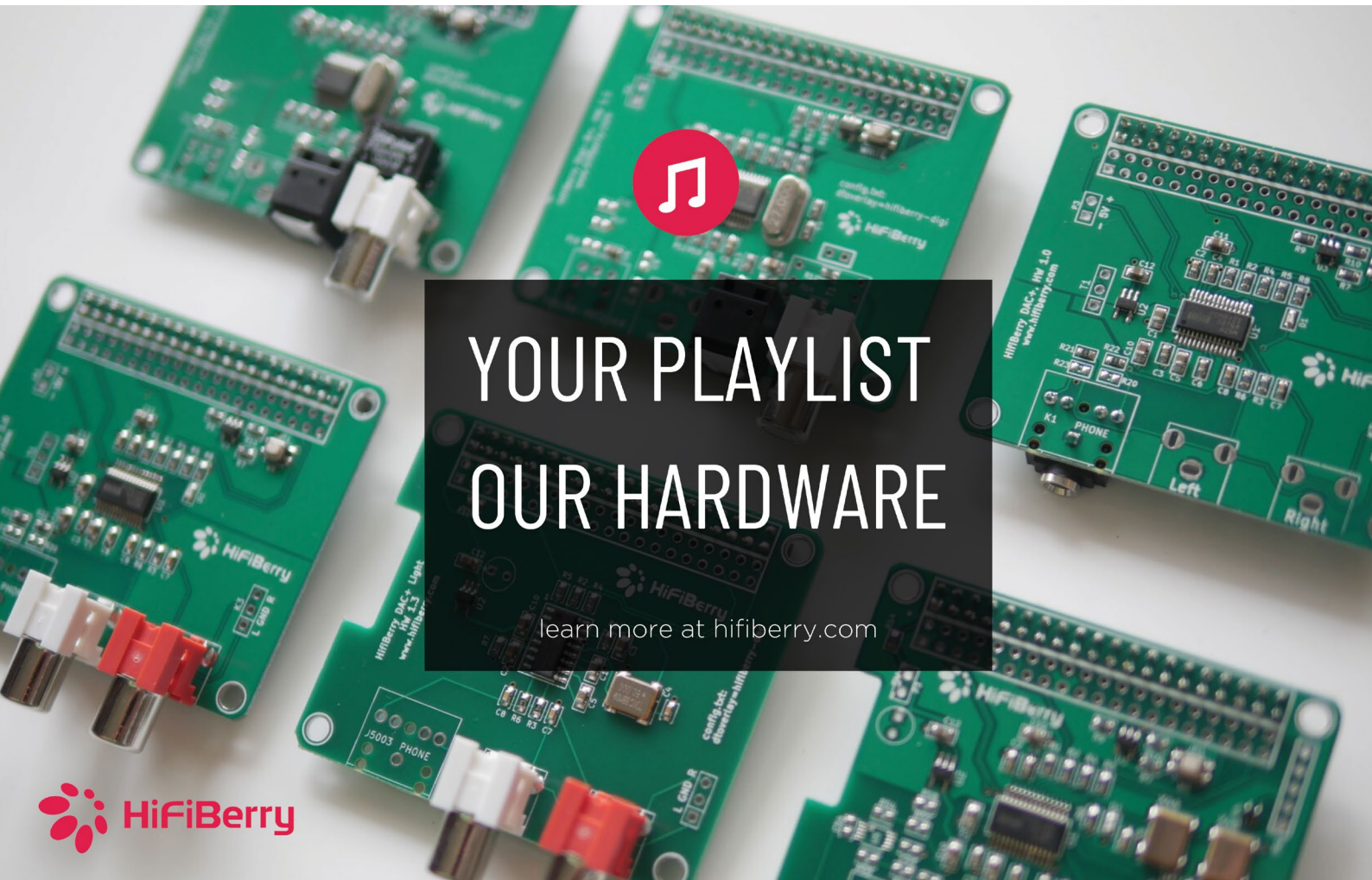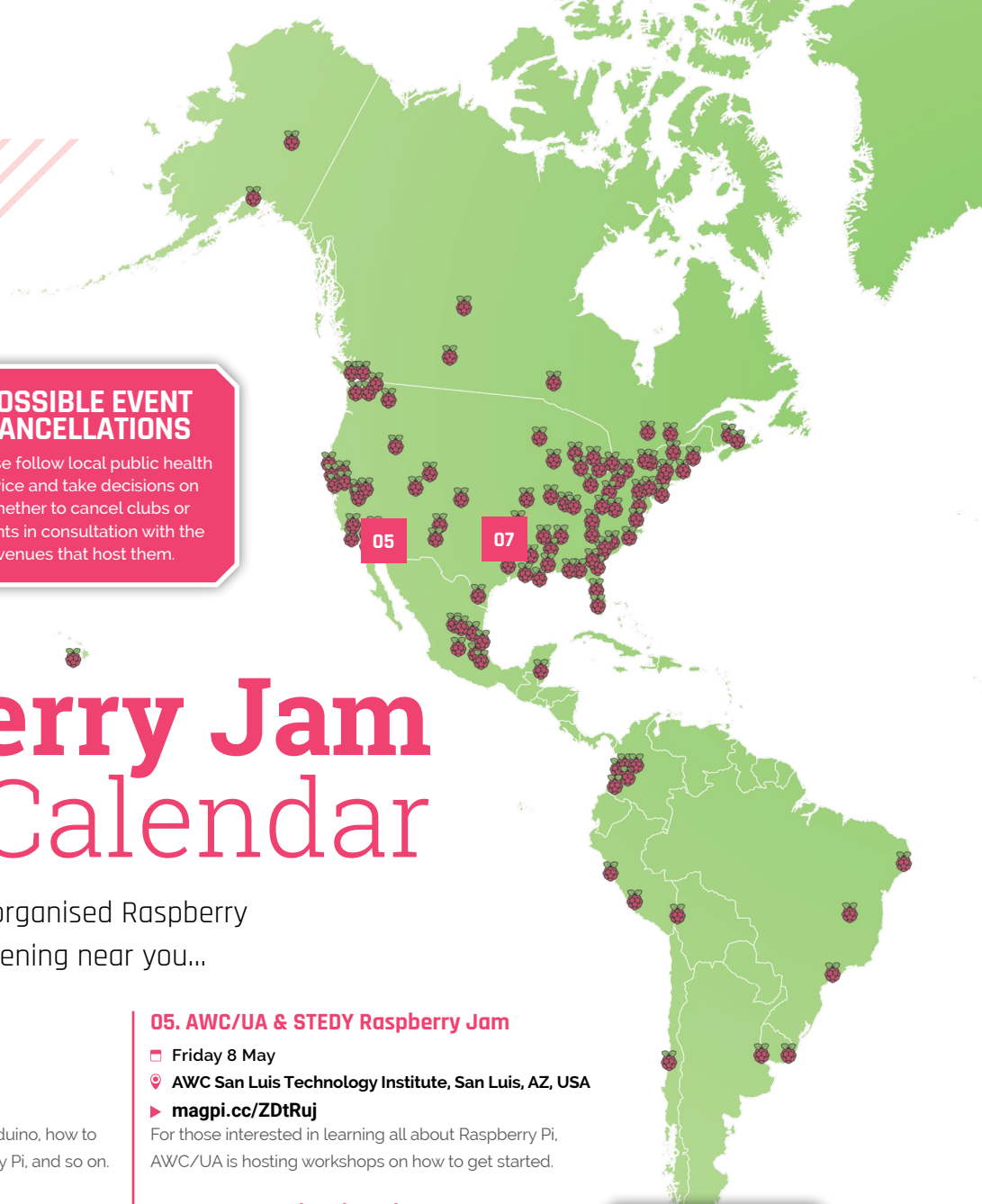
# Raspberry Jam
## Event Calendar

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

**05**

**07**

### 01. Medway Makers Tech Meetup
📅 Sunday 3 May
📍 Online
▶ magpi.cc/OPTwTU
Workshops on building cool things with Arduino, how to code for beginners, how to use a Raspberry Pi, and so on.

### 02. Free Robotics Sunday Virtual Lesson
📅 Sunday 3 May
📍 Online
▶ magpi.cc/sgixR9
A virtual lesson from this free after-school robotics workshop that normally meets on Wednesdays/ Thursdays near Roppongi, Tokyo.

### 03. ONLINE: Music Creation Basics with Sonic Pi
📅 Tuesday 5 May
📍 Online
▶ magpi.cc/cM3fNs
Learn the basics of Sonic Pi, a free, cross-platform code-based music creation and performance tool!

### 04. Houston Raspberry Pi Meetup
📅 Wednesday 6 May
📍 Online
▶ magpi.cc/qfSx5y
A community of coders, artists, educators, and engineers interested in project-based learning.

### 05. AWC/UA & STEDY Raspberry Jam
📅 Friday 8 May
📍 AWC San Luis Technology Institute, San Luis, AZ, USA
▶ magpi.cc/ZDtRuj
For those interested in learning all about Raspberry Pi, AWC/UA is hosting workshops on how to get started.

### 06. West Sound CoderDojo on Zoom!
📅 Saturday 9 May
📍 Online
▶ magpi.cc/jQ16dX
Ninjas, parents, mentors, and all are invited to join West Sound CoderDojo for bi-weekly sessions of coding and fun.

### 07. Dallas Young Makers Club
📅 Saturday 23 May
📍 J. Erik Jonsson Central Library, Dallas, TX, USA
▶ dallasyoungmakers.org
Free mentor-led, hands-on projects for kids with Raspberry Pi, Lego Mindstorm robots, and more.

### 08. Moon Hack 2020
📅 Saturday 30 May
📍 Online
▶ magpi.cc/TNZg8m
A world-record-breaking online event for kids who love to code. A fun way to learn, with no experience necessary.

**FULL CALENDAR**

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

**rpf.io/jam**
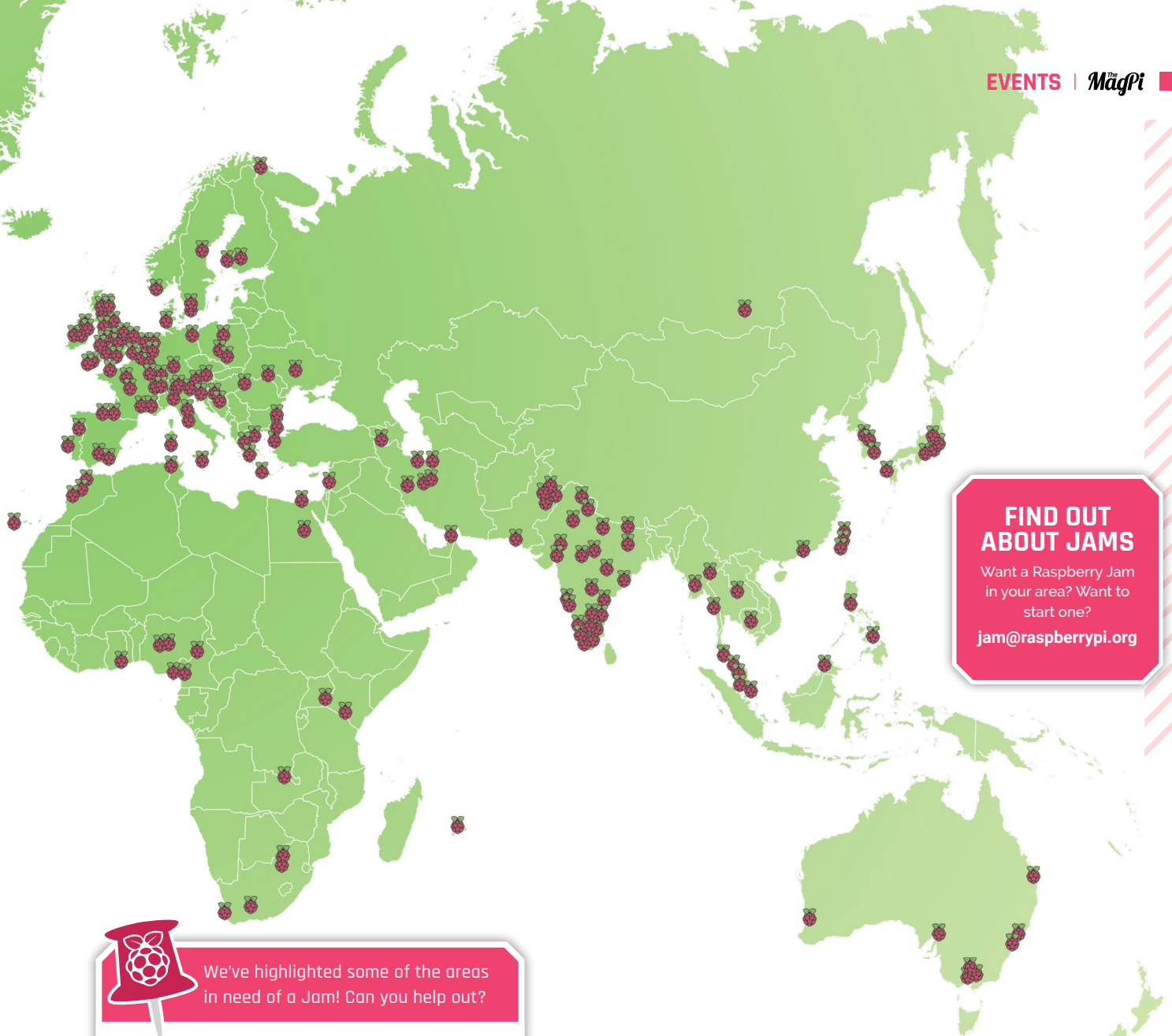
**FIND OUT ABOUT JAMS**

Want a Raspberry Jam in your area? Want to start one?

**jam@raspberrypi.org**

We've highlighted some of the areas in need of a Jam! Can you help out?
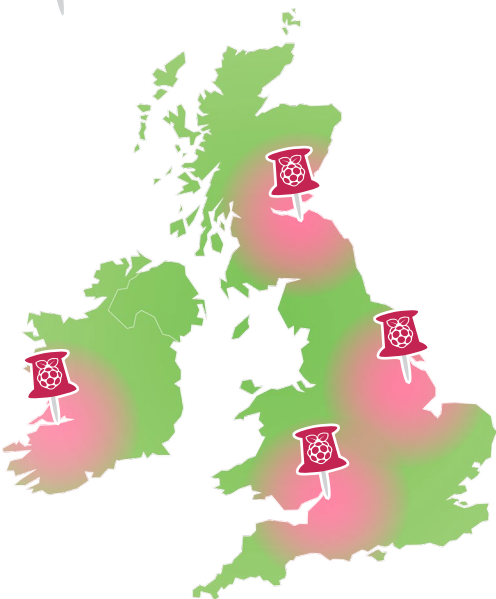
## Raspberry Jam advice:
# Activities

**"S**omething like Astro Pi is a fantastic thing to get involved in if you want to really give children some focus and **direction.** The idea of having your code run on the International Space Station is a real motivator for young kids."

**Dave Honess – ESA**

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

**RASPBERRY JAM GUIDEBOOK**

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

# Your Letters
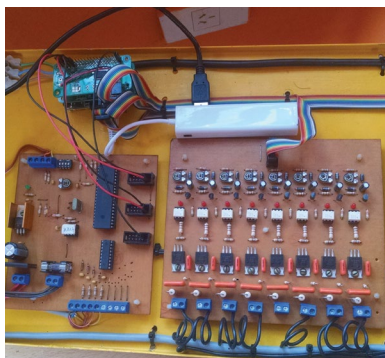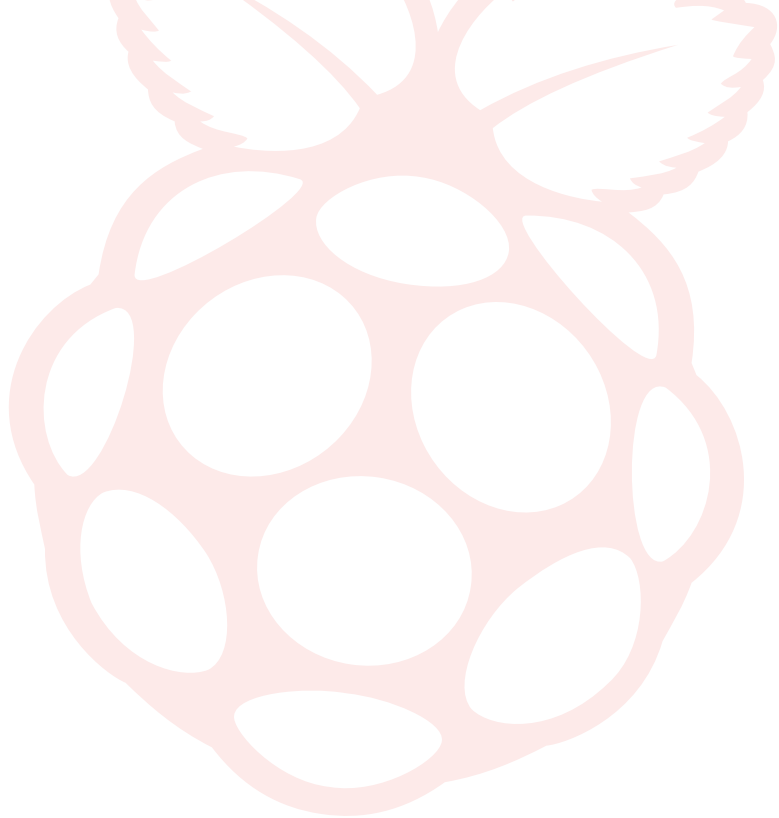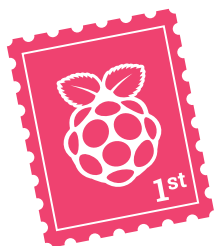


▲ El Carrillon will last for another several years thanks to the upgrades we covered in issue 92

## Bells, bells, bells

**What a big surprise to see this here, I was looking for news about Raspberry Pi software and see this story. Immediately when I saw the Carillon, I was shocked! It was my father and an electric engineer who worked on the original project and built the Carillon. Also, I was 13 at that time and was involved too, loading the code with the notes and songs in a Commodore 64. The program sends the signals to the parallel port of the machine and a custom interface activates the switch of the bells. My father's name was Carlos Maria Morelli. He passed away in August 2014, but it is really good to see his project is still alive and working!**

**JPMorelli** via the blog

It's fantastic to hear from you about this! We featured El Carrillon in issue 91 of the magazine as it had recently been updated to use a Raspberry Pi. To have a link to the original build this way is fantastic, and not something we regularly see! Thanks for getting in touch and telling us your story.

## Lens concerns

**Are there any ways to add proper lenses to a Raspberry Pi Camera Module? I have some nice lenses I'd like to use for some more advanced projects.**

**Karl** via Twitter

There are a few 3D-printable mounts on Thingiverse for specific lens makes (Canon, Nikkon, etc) and they seem to work pretty well. Alternatively, take a look at the new official HQ Camera that can be used with any C- or CS-mount lens.

▼ The new HQ Camera has support for interchangeable lenses and also features a higher-res 12.33MP sensor

▼ Videos and projects are available for those wanting to learn from home!
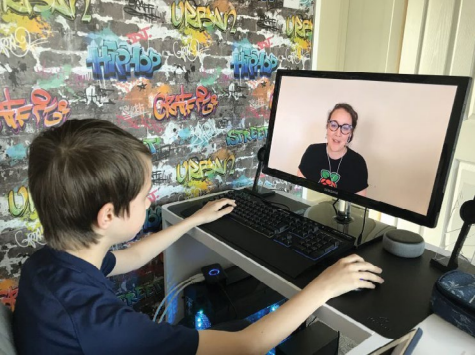
## Teaching at home

I know the Raspberry Pi Foundation has a lot of great project resources online, but I was wondering if there were any resources specifically to help teaching at home? Would love to help my kids with their computer learning a bit more!

**Holly** via Facebook

You're in luck – the Raspberry Pi Foundation has been recently creating resources for parents wanting to teach at home, and you can find the guide here: **magpi.cc/makingathome**.

This is part of a current series of 'digital making at home' blogs which have ideas and projects and such for doing fun stuff at home with kids!

Don't forget, you can also get Hello World, our free sister publication for educators, at **helloworld.cc**.

## Contact us!
> Twitter  **@TheMagPi**
> Facebook  **magpi.cc/facebook**
> Email  **magpi@raspberrypi.com**
> Online  **raspberrypi.org/forums**

## #MonthOfMaking again

I have made a project which I would like to share with you for the 'Month Of Making' and I was wondering if you would be interested in including it in your magazine.

My project, Plant Pi, is an automated self-watering system that detects the moisture level in the plant's soil. I have attached a write-up of my project:

"Our project is a fully automated plant watering system. It is designed to be used in your house or greenhouse to make life easier and to make the watering process more precise. With a smart system putting the right amount of water in the soil of a plant when it is needed. Ensuring that your plant will always be happy.

"To set it up, all you need to do is fill up the reservoir, put the moisture sensor in the soil, and place the tubing into the plant pot. After this, run the code and then you leave it there to do all the work to help your plant get watered. You can go to the website to see if it's watering the plant or not and look at the moisture levels in the soil. The system will also send you notifications when your plant is being watered."

I have also made a small YouTube clip you can watch it here: magpi.cc/plantpi.

I have written the code myself in Python. I used HTML for the website and I used the Pushover API for the push notifications.

**Cyrus** via email

Cyrus is twelve, and we love the project he's made here! Not every automated plant waterer has such a detailed website interface to go with it.

▼ We like the relative simplicity of this setup
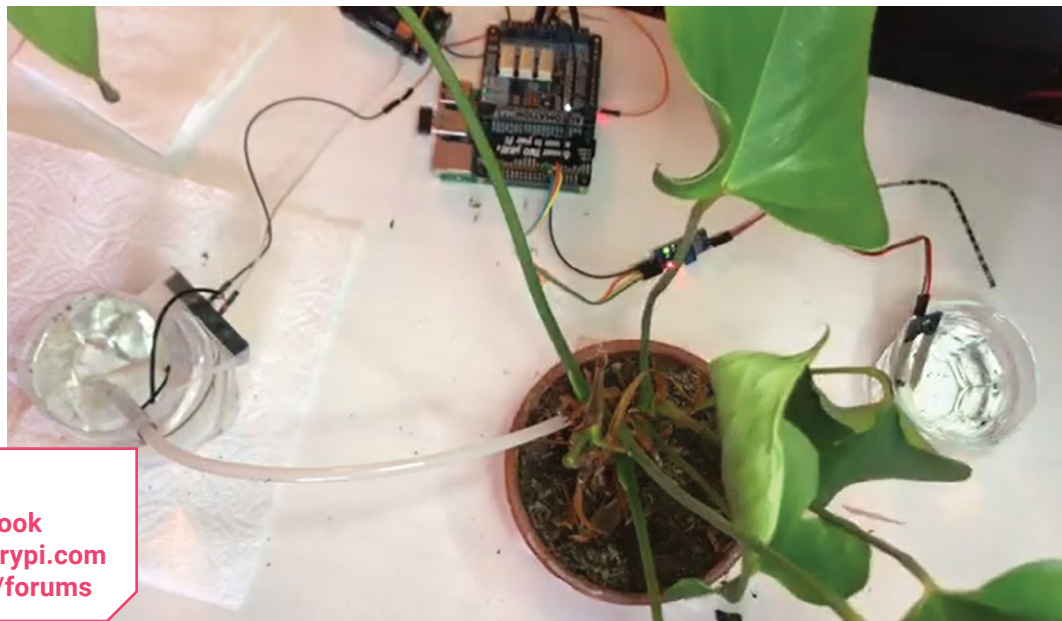
# WIN ONE OF FIVE HQ CAMERAS!

The new official Raspberry Pi camera supports up to 4K image capture, with changeable lenses for the best shot!

Five lucky competition winners will each receive an **HQ Camera with a 6 mm CS-mount lens** and our new **Official Raspberry Pi Camera Guide** 132-page book.
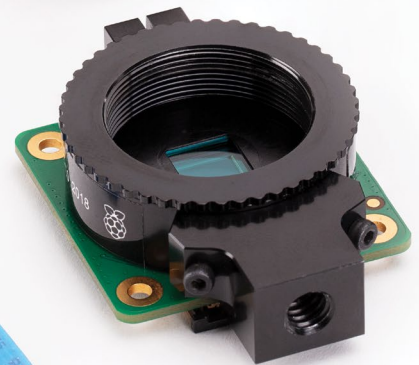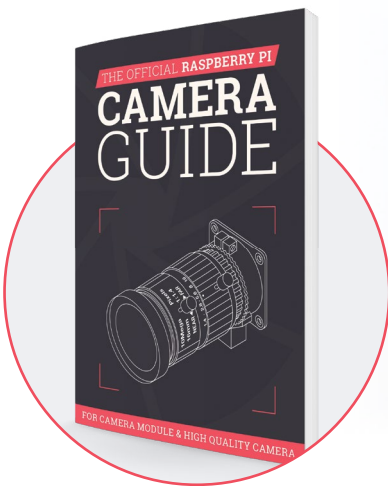
**Head here to enter:** magpi.cc/win │ **Learn more:** magpi.cc/cameraguide

## Terms & Conditions

# LEARN COMPUTING & ICT

## WITH RASPBERRY PI
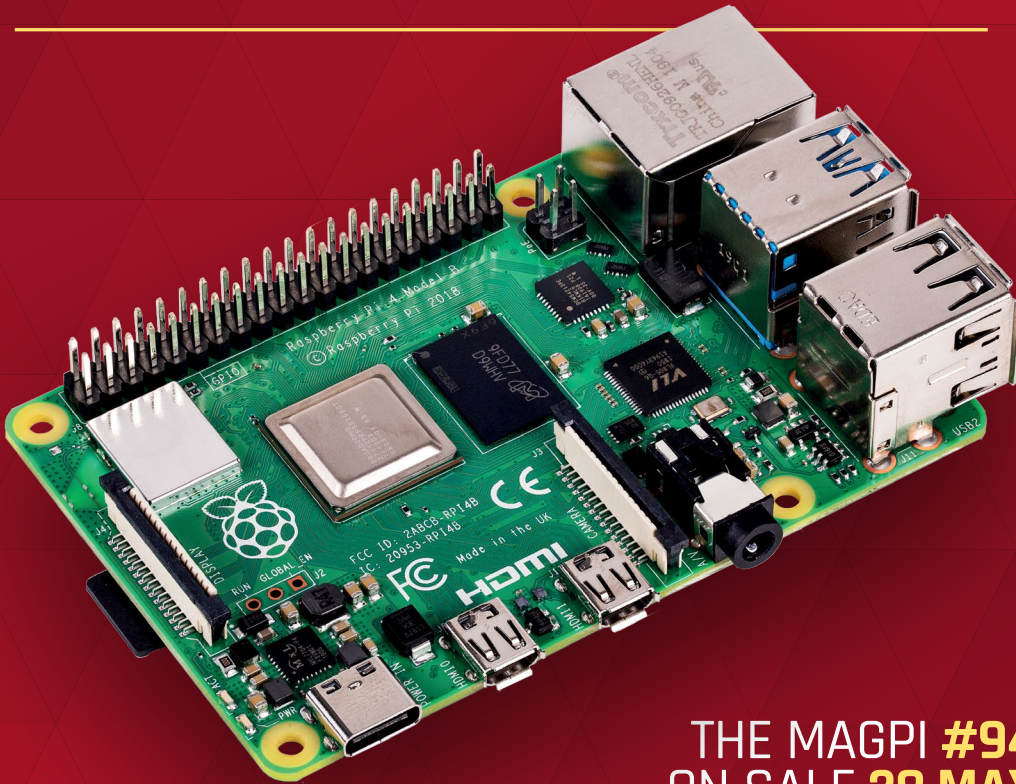
THE MAGPI **#94**
ON SALE **28 MAY**

## Plus!

Build a 3D camera

**Make a doorbell intercom**

Create an ISS tracker

**The top 10 robot kits**

### DON'T MISS OUT!

# magpi.cc/subscribe

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

# Let's get to work

Raspberry Pi isn't just fun and games. Put it to work and you'll be surprised. By **Lucy Hattersley**

**H**ave you ever tried working on a Raspberry Pi? It's a surprisingly pleasant experience. Sure, you might not be able to run Microsoft Word or Apple Mail, but Chromium, LibreOffice and Claws Mail are excellent, and free, alternatives. Plus you get beginner-friendly programming tools like Scratch and Thonny.

There's been a huge uptake in Raspberry Pi recently, and we think it's because folks want to explore low-cost computing.

Gareth Halfacree, one of our most trusted testers, tried out the Raspberry Pi 4 as a desktop solution when it first arrived, and it turns out to be a surprisingly versatile option. I'm glad we asked him to look at how Raspberry Pi 4 shapes up as a remote working option.

There may still be shortcomings: we couldn't get Zoom to work, although some would see this as a welcome bonus. But most tools – such as Google Docs, LibreOffice, GIMP, Slack, Skype, Discord, and GitHub – are great on Raspberry Pi.

Raspberry Pi as a desktop computer is a wonderful experience.

I often use it because it allows clarity of focus. Linux is where I get things done. And the Raspberry Pi 4 with 4GB RAM snaps and crackles along; there are no sluggish setbacks.

We're delighted that Raspberry Pi, 'the little computer that could', is now very much 'the big computer that does'.

computing tasks, but still has its heart and soul in programmable computing. You can't hack electronics to the GPIO pins of a tablet, no matter how touchable the interface is.

A Raspberry Pi can become whatever you want it to be – even if what you want it to be is a work

> ❝ A Raspberry Pi can become whatever you want it to be – even a work tool ❞

Still, we're all about hacking and making at heart. I've been having fun lately with the Raspberry Shake seismograph, building an outdoor light display, and working with robotics. I still dream of a Raspberry Pi drone project that doesn't take your hands off.

Like most readers of *The MagPi*, I'm on the lookout for new projects and things to build.

### Perfect machine

In many ways Raspberry Pi is the perfect machine for the modern age. It is powerful enough to do modern

tool. We're lucky, because our work is Raspberry Pi. And we get to spend our days making cool things and writing about them. And I get to do it all on the same computer.
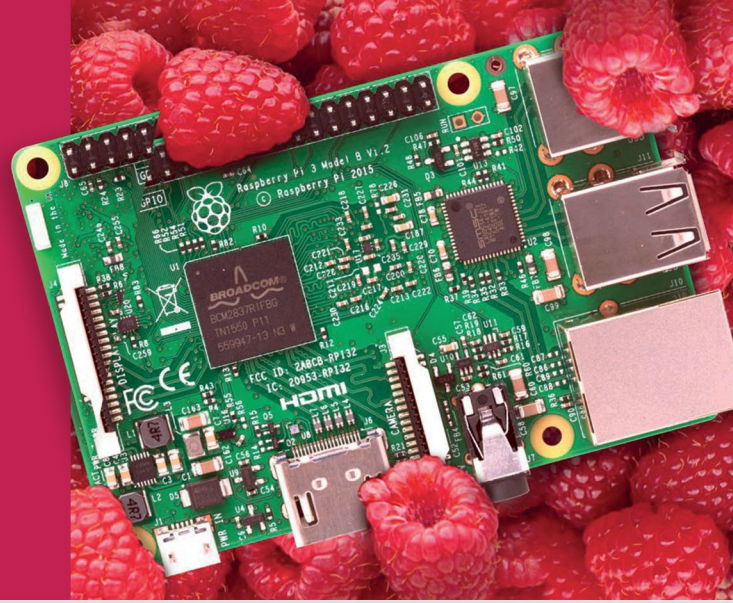
If you haven't tried working on Raspberry Pi then I hope this month's *The MagPi* magazine has managed to change your mind. Give it another shot. **M**

**AUTHOR**

### Lucy Hattersley

Editor of *The MagPi* magazine and expert at making wonky robots and burnt pizza.

**magpi**.cc
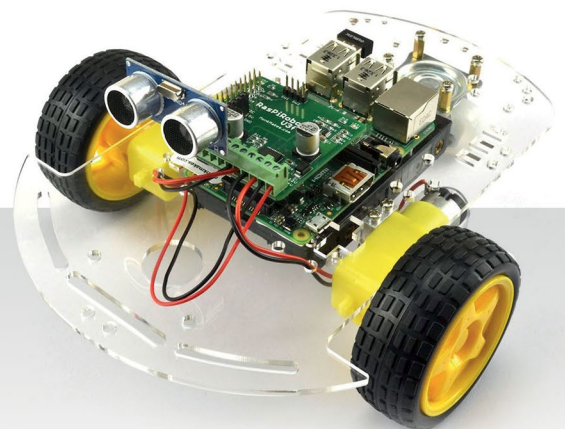
# PIPER

**INSPIRE FUTURE GENERATIONS
WITH PIPER COMPUTER KIT 2**

the award-winning computer kit for ages 8+