The MagPi

The official Raspberry Pi magazine

Build a
RASPBERRY PI
PORTABLE COMPUTER

Code on the go with a battery, touchscreen & keyboard

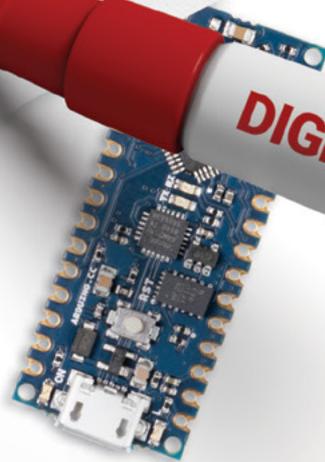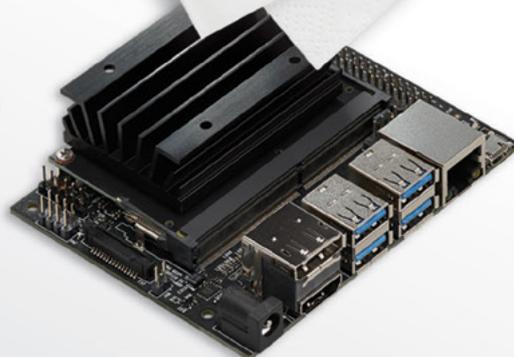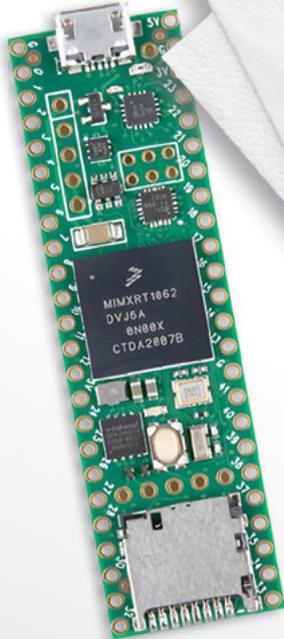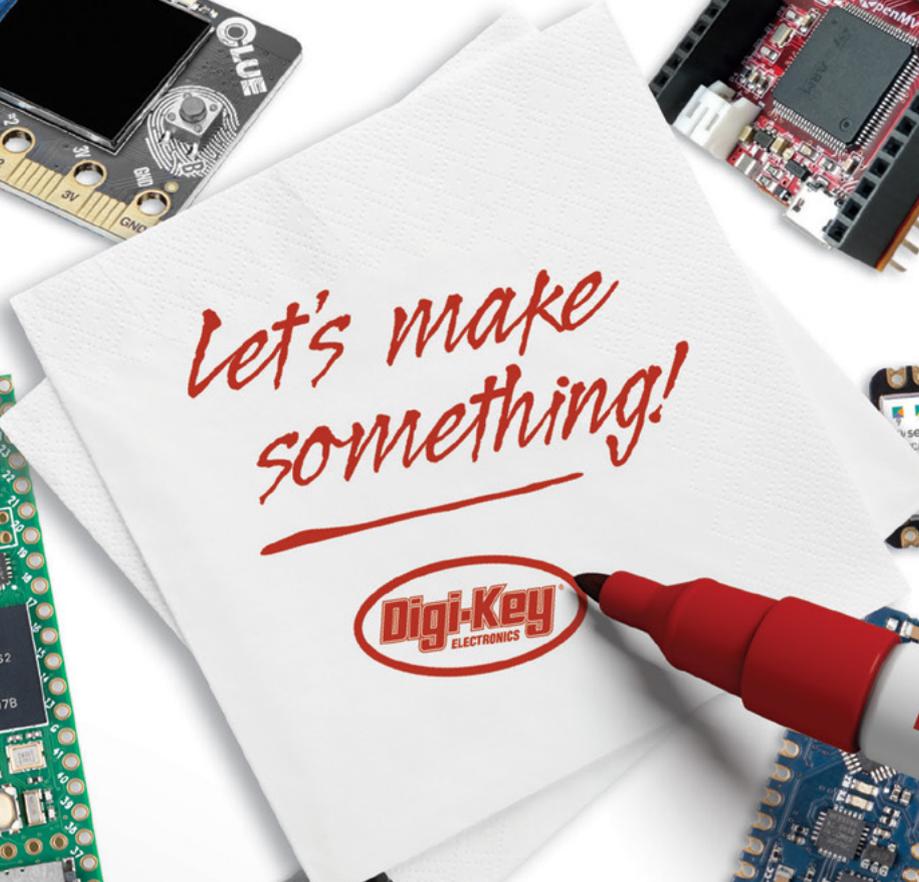Get legal ROMs for **classic consoles**

Hack a **haunted house!**

Using your Raspberry Pi

Contents

Raspberry Pi PRESS

GLOBAL DELIVERY
magpi.cc/store

SUDOKU ROBOT • AI AIR HOCKEY • DIY SCORECARD

WELCOME
to *The MagPi* 98

One of the true joys of the modern world, at least for me, is the ability to get stuff done on the move.
I love popping into the office, and the ability to code, write, and create in the park – or on a train, or while sitting in a coffee shop – has more than a little charm.

With its small footprint, quad-core CPU, and 8GB RAM, my Raspberry Pi 4 is a true pocket powerhouse. And there are now oodles of options for adding screens, keyboards, and trackpads for me to use Raspberry Pi on the move.

We've gathered together the full guide to portable computing with Raspberry Pi 4 (Page 34) and if you fancy building a Raspberry Pi 2Go portable computer (Page 38), then PJ has you covered.

Rob, on the other hand, loves the home life. And he's now getting ready for Halloween by bringing a haunted house to life with Raspberry Pi tech! Get a head start this year and make it one for the local kids to remember with our guide to building a Raspberry Pi haunted house (Page 68).

Whether you're sticking with home or getting out and about, this month's *The MagPi* magazine has you covered.

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy is editor of *The MagPi* and this month she has been exploring London and Cambridge.

@LucyHattersley

GET A
RASPBERRY ZERO W KIT
WITH A SUBSCRIPTION!
PAGE **32**

Contents

> Issue 98 > October 2020

34

26

16

SUDO

CubeSat

Tutorials

The Big Feature

Haunted House Hacks

Reviews

Community
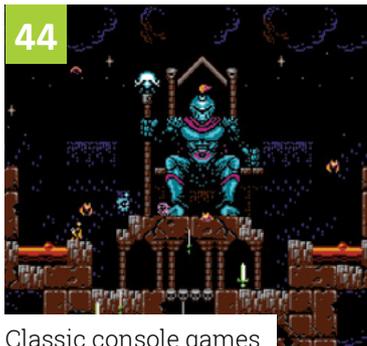
Classic console games

Tap-an-LED part 2

PiBoy DMG

Mister C interview

WIN 1 OF 3 **M.A.R.S. ROVER** ROBOT KITS!
95

Code

Design
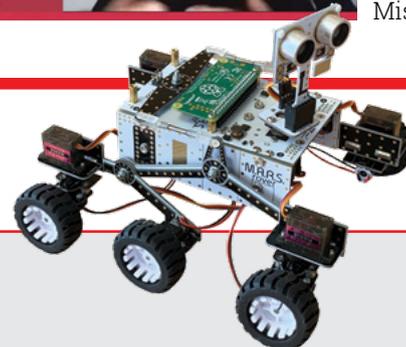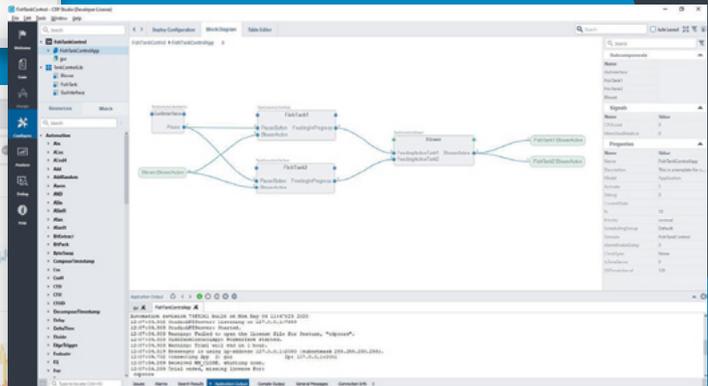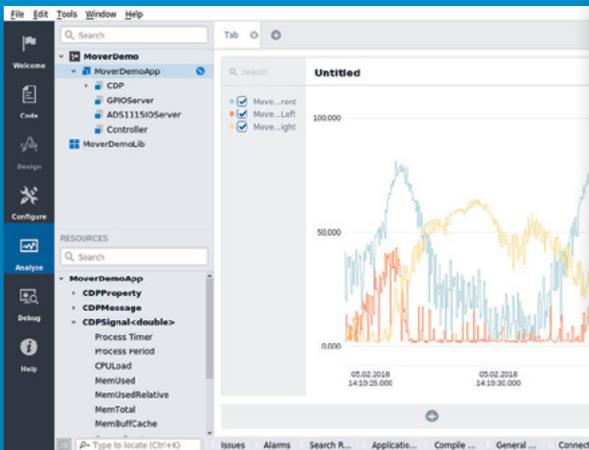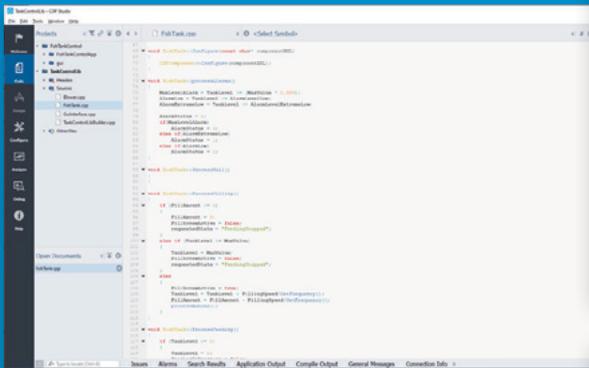
Configure

Analyze

CDP STUDIO: HOME PROJECTS
MADE EASY

Professional control system development tool

CDP Studio is a great software development tool for your home projects. Build systems for Raspberry Pi, use C++, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create user interfaces in the integrated designer. Code less and do more!

NOW FREE FOR NON-COMMERCIAL USE.

CDP Technologies AS
Hundsværgata 8, 6008 Ålesund, Norway
Tel: +47 990 80 900 · info@cdptech.com

www.cdpstudio.com

CDPStudio

Pi Wars 2021 lays
down double challenge

Pi Wars is taking a dual-pronged approach to Raspberry Pi battles, reports **Rosie Hattersley**

▲ The Sphero RVR is a popular robot kit

Pi Wars 2020 had to beat a hasty online retreat this year, but 2021 will see the celebrated Raspberry Pi robot challenge weekend make two feisty appearances.

Organisers Mike Horne and Tim Richardson from CamJam (Cambridge Raspberry Jam) have cunningly conjured up a Virtual Pi Wars Conference as well as a take-part-from-home series of Pi Wars challenges.

Mike explains that, rather than risk having to cancel an in-person version of Pi Wars 2021, they decided to turn a negative into a positive and hold two online events. The Virtual Pi Wars Conference will be on Saturday 23 January and will be a live-streamed event hosted by CamJam's Rob Bricheno. Attendees get an insight into what makes a successful Pi Wars robot, which may or may not help them in their endeavour to vanquish rivals during Pi Wars itself. Sign up at **piwars.org/2021-vc.**

Home advantage

Pi Wars @Home, meanwhile, will see teams first build their own challenge courses, then record their best attempt at completing it. The resulting

▶ The Pi Wars Mini Conference is returning in January 2021 after a successful debut

challenge attempts will be broadcast online over the weekend of 20–21 March.

Courses will have a 'home' theme and must be constructed from readily available household objects, with assembly instructions sent out well in advance to all teams who pay the £15 entry fee. Competitors can take part from anywhere in the world, since each team has to submit their challenge attempt videos at least 72 hours ahead of the competition weekend. Entry is open already and closes on 12 October. Teams that accepted and paid their entry fee for Pi Wars 2020 will be able to take part in the 2021 @Home event at no extra

▲ Audrey III – Amy Willis and her dad's robot tribute to Audrey II from *Little Shop of Horrors*, made its debut at this year's Virtual Pi Wars event

charge. You will still need to apply, but will be automatically accepted.

2020 re-vision

Despite having to become a virtual event in 2020, Pi Wars attracted plenty of online attention with more than 350 visitors watching the live-streamed

> ❝ Pi Wars @Home will see teams first build their own challenge courses, then record their best attempt at completing it ❞

competition at once. The Disaster Zone theme that was to have been the focus of Pi Wars 2020 may instead loom back into view in 2022. This will mean would-be competitors from 2020 will still be able to impress us all with their appropriately apocalyptic robots. For course building instructions and to sign up, go to **piwars.org/2021-vpw**.

▲ The makers of this Reliant Raspberry robot may have had Mr Bean's nemesis in mind

17000 ft

How do you get internet over three miles up the Himalayas? That's what the 17000 ft Foundation and Sujata Sahu had to figure out. **Rob Zwetsloot** reports

MAKER

Sujata Sahu

A veteran IT professional and educator, Sujata moved back to India and founded 17000 ft after visiting the remote Ladakh region.

17000ft.org

L iving in more urban areas of the UK, it can be easy to take for granted decent internet and mobile phone signal. In more remote areas of the country, internet can be a bit spotty but it's nothing compared with living up in a mountain.

"17000 ft Foundation is a not-for-profit organisation in India, set up to improve the lives of people settled in very remote mountainous hamlets, in areas that are inaccessible and isolated due to reasons of harsh mountainous terrain," explains its founder, Sujata Sahu. "17000 ft has its roots in high-altitude Ladakh, a region in the desolate cold desert of the Himalayan mountain region of India. Situated in altitudes upwards of

9300 ft and with temperatures dropping to -50°C in inhabited areas, this area is home to indigenous tribal communities settled across hundreds of tiny, scattered hamlets. These villages are remote, isolated, and suffer from bare minimum infrastructure and a centuries-old civilisation unwilling but driven to migrate to faraway cities in search of a better life. Ladakh has a population of just under 300,000 people living across 60,000 km² of harsh mountain terrain, whose sustenance and growth depends on the infrastructure, resources, and support provided by the government."

The local governments have built schools. However, they don't have enough resources or

▶ Ladakh is very much a desert, despite how high up and cold it is

A Raspberry Pi powers a local internet for the school

Tablet computers are provided that connect to a Raspberry Pi-powered network

Many students used to drop out or go away for school

Quick **FACTS**

> Ladakh is classed as a desert, a cold one at that

> 95% of schools in Ladakh exist in villages which can be inaccessible for six months a year

> Less than 20% have access to electricity for a few hours a day

> Less than 30% have access to a mobile signal, which is 2G

> Students were able to take tablets home while learning from home this year

Students have access to learning resources downloaded to the hub Raspberry Pi

▲ A huge number of students have already benefited from the program

in the cloud, with an app on a teacher's phone that will download data when it can and then update the installed Raspberry Pi in their school.

qualified teachers to be truly effective, resulting in a problem with students dropping out or having to be sent off to cities. 17000 ft's mission is to transform the education in these communities.

High-altitude Raspberry Pi

"The Foundation today works in over 200 remote government schools to upgrade school infrastructure, build the capacity of teachers, provide better resources for learning, thereby improving the quality of education for its children," says Sujata. "17000 ft Foundation has designed and implemented a unique solar-powered offline digital learning solution called the DigiLab, using Raspberry Pi, which brings the power of digital learning to areas which are truly off-grid and have neither electricity nor mobile connectivity, helping children to learn better, while also enabling the local administration to monitor performance remotely."

Each school is provided with solar power, Raspberry Pi computers to act as a local internet for the school, and tablets to connect to it. It serves as a 'last mile connectivity' from a remote school

> ❝ The system has now delivered over 60,000 hours of learning to students in remote villages ❞

Remote success

"The solution has now been implemented in 120 remote schools of Ladakh and is being considered to be implemented at scale to cover the entire region," adds Sujata. "It has now run successfully across three winters of Ladakh, withstanding even the harshest of −50℃ temperatures with no failure. In the first year of its implementation alone, 5000 students were enrolled, with over 93% being active. The system has now delivered over 60,000 hours of learning to students in remote villages and improved learning outcomes."

It's already helping to change education in the area during the winter. Many villages (and schools) can shut down for up to six months, and families who can't move away are usually left without a functioning school. 17000 ft has changed this.

"In the winter of 2018 and 2019, for the first time in a few decades, parents and community members from many of these hamlets decided to take advantage of their DigiLabs and opened them up for their children to learn despite the harsh winters and lack of teachers," Sujata explains. "Parents pooled in to provide basic heating facilities (a Bukhari – a wood- or dung-based stove with a long pipe chimney) to bring in some warmth and scheduled classes for the senior children, allowing them to learn at their own pace, with student data continuing to be recorded in Raspberry Pi and available for the teachers to assess when they got

back. The DigiLab Program, which has been made possible due to the presence of the Raspberry Pi Server, has solved a major problem that the Ladakhis have been facing for years!"

How can people help?

Sujata says, "17000 ft Foundation is a non-profit organisation and is dependent on donations and support from individuals and companies alike. This solution was developed by the organisation in a limited budget and was implemented successfully across over a hundred hamlets. Raspberry Pi has been a boon for this project, with its low cost and its computing capabilities which helped create this solution for such a remote area. However, the potential of Raspberry Pi is as yet untapped and the solution still needs upgrades to be able to scale to cover more schools and deliver enhanced functionality within the school. 17000 ft is very eager to help take this to other similar regions and cover more schools in Ladakh that still remain ignored. What we really need is funds and technical support to be able to reach the good of this solution to more children who are still out of the reach of Ed Tech and learning. We welcome contributions of any size to help us in this project."

For donations from outside India, write to **sujata.sahu@17000ft.org**. Indian citizens can donate through **17000ft.org/donate**. M

▲ Some of the village schools go unused in the winter

▼ Not all children stay in the villages year round

MAKER

Tyler McKay

Tyler is a 27-year-old software engineer based in New Jersey. He enjoys building things out of wood and metal and he runs a YouTube channel showcasing advanced Raspberry Pi projects.

magpi.cc/ tyandgig

⚠️

Warning! High voltage

Be careful when working with mains electricity, and make sure to ground and insulate projects using it.

magpi.cc/ electricalsafety

DIY Score Counter

Need to keep score? Then Tyler McKay's project is going to win over your heart, as **David Crookes** discovers

Some of the best ideas have sprung to mind during a few drinks with friends. But Tyler McKay's decision to create a device that could keep score came well before the first drink was poured.

With his roommate about to hold a beer-tasting party involving 40 people, Tyler wanted an eye-catching way of determining which of the eight brews being tested were thought to taste the best. "My friend wants to be a brewer and own his own brewery, so we thought it would be cool if we could score the beers live," he says.

The resulting project was a scoreboard that put two beers head-to-head in each round, allowing participants 15 minutes to cast their votes before moving to the next match-up. Tyler drew upon his previous experience creating an LED message board which wrote data to WS2812B LEDs from a Raspberry Pi.

"The main things I had to wrap my head around was understanding that the LEDs are individually addressable and the protocol to control them," Tyler explains. "It's helpful to understand what's going on behind the scenes so that it's easier to debug and fix something if it breaks."

Raising the bar

To construct the scoreboard, Tyler used pine plywood for the façade and red oak for the borders. "I used a circular saw to cut the plywood and a jig-saw to cut the rectangles for the two names and the timer sections," he reveals.

The LEDs were then lined up in a grid. "I drilled
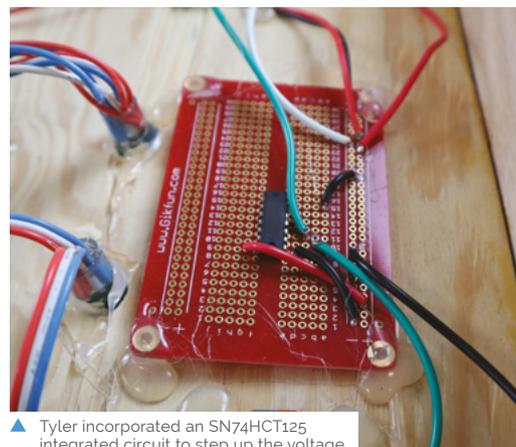
▲ Tyler incorporated an SN74HCT125 integrated circuit to step up the voltage

holes for the WS2812B LEDs and hot-glued them together," he continues. "For the electronics, I used an IC to step up the voltage from the signal that came from a Raspberry Pi Zero W computer. This was powered externally by a 5 V power supply."

> " **Participants would text the same service with the name of the beer they liked** "

The beers were seeded from one to eight and placed in a coding bracket. A web server was written in PHP for the first design, but this was switched to Ruby on Rails to make it easier for other makers to replicate. The match-ups, stored in an SQL database, were created via a text message containing the names of the beer and the time limit. Since the scoreboard is a set of seven connected message boards, the code assumes there is one continuous string of LEDs.

▲ The back end of the project shows the neat positioning of the LEDs connected to a Raspberry Pi Zero W

Tyler made use of the pure Python library, BiblioPixel, to manipulate the LED displays
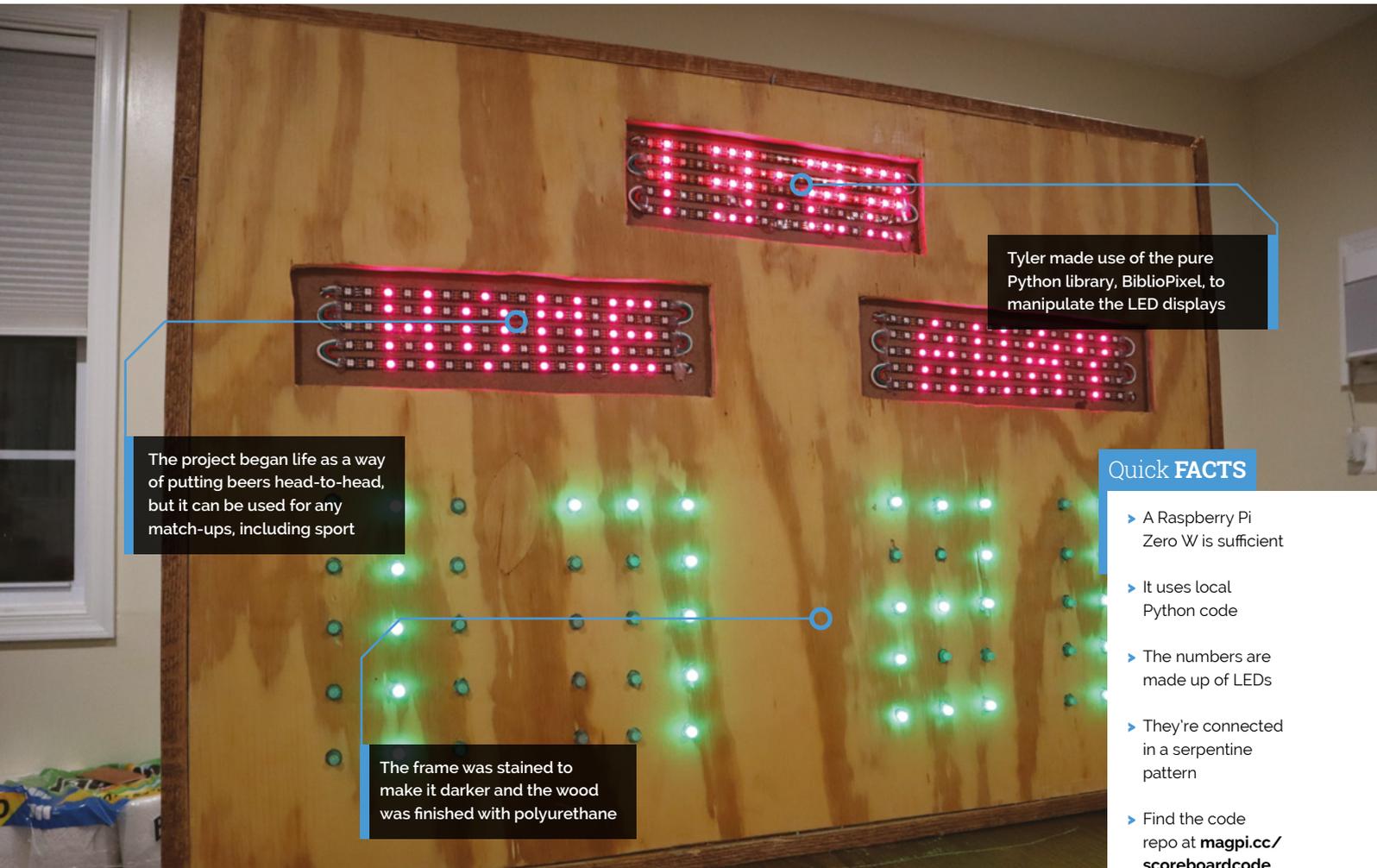
The project began life as a way of putting beers head-to-head, but it can be used for any match-ups, including sport

The frame was stained to make it darker and the wood was finished with polyurethane
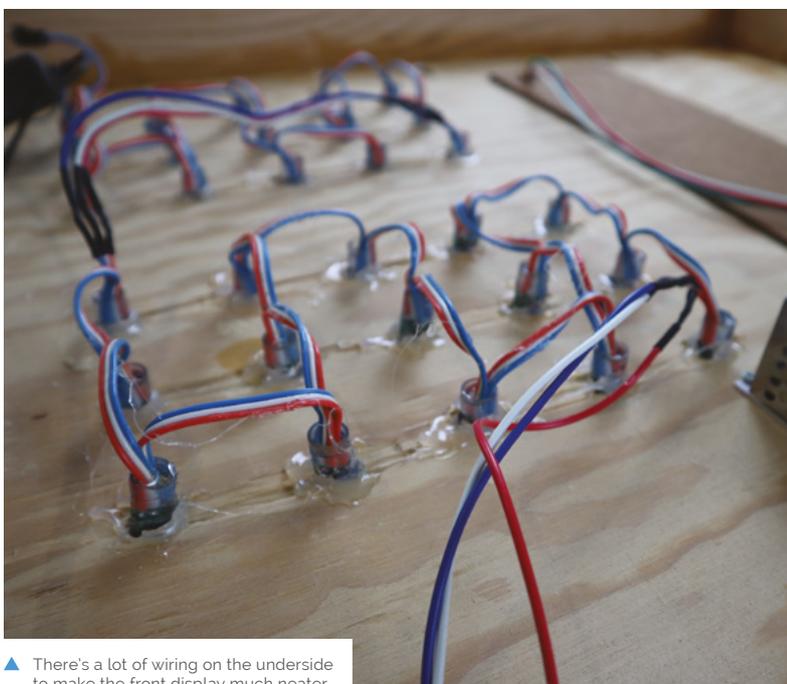
▲ There's a lot of wiring on the underside to make the front display much neater

Indexes determine the location of the start and end of each section, and each character is made up of five rows and three columns. But how do people use the scoreboard to place their votes?

"Participants would text the same service with the name of the beer they liked," explains Tyler. "They had the ability to change their decision until the end of the time limit, with each phone having one vote – enforced by checking the phone number."

Thirsty for more

With the party long gone and the favourite beer having been determined, Tyler has begun using the scoreboard to live-stream scores from sports games. He has also put a rendition of the code on GitHub so that it can be used more generically, and this has gone down well.

"The reception has been insane – I never realised so many people do this as a hobby and I've gotten messages from people telling me they are going to make their own. It's been a really cool experience and it makes me want to create something even cooler." And that, dear readers, is indeed a winning result. M

MAKER

Arijit Das

Arijit is a computer science engineering student based in India. He focuses on AI and robotics in a team known as Sparklers and often uses Raspberry Pi in his projects.

magpi.cc/arijitdas

SUDO: sudoku-solving robot

Fans of brain-training puzzles may be aghast, but there's a Raspberry Pi robot that can solve sudokus. **Rosie Hattersley** asks, why?

Keeping our minds and bodies active is reputedly key to a healthy life, with puzzles such as sudoku providing the right sort of mental stimulation. Much like chess, however, it seems machines are far more adept at them than humans. Computer science student Arijit Das has come up with SUDO, a Raspberry Pi-based robot that's a lightning-fast sudoku whizz.

Named after both the UNIX command and Japanese puzzle, SUDO began life as a companion to help Arijit with number-based puzzles. "I was trying to master sudoku puzzles and invested a lot of effort," he explains. As an experienced inventor and programmer, he soon realised there was no need to

solve sudokus manually. "I can build a robot to solve any complex sudoku within seconds," he reasoned.

Keeping track

Arijit has been experimenting with Raspberry Pi possibilities since 2017, when he began trying out IoT projects. These include a Spy-Dog live video surveillance bot, a drowsiness detection device, and a web-based GPS tracker. His SUDO

▲ Arijit demonstrates his sudoku-solving robot to fellow Raspberry Pi fans

▲ Not just a pretty face, this cute little robot is a whizz at solving sudokus

When Raspberry Pi's Camera Module detects a sudoku puzzle, SUDO displays the numbers it's detected on the screen, quickly followed by the solution

SUDO is powered by a Raspberry Pi 3B+ and has been carefully trained using machine learning to recognise numbers shown in different fonts

Arijit doesn't have a 3D printer, so he used the shell of a toy as SUDO's body

Quick **FACTS**

> SUDO is made from the shell of Arijit's childhood toy

> SUDO has become a popular selfie-magnet at maker faires

> Several kids have asked Arijit to make another version of SUDO

> ... one that can solve maths (homework)

> Implementing the puzzle-solving algorithm improved Arijit's own sudoku skills

▲ Arijit holds out a sudoku puzzle for the robot to scan and solve; at the moment, it needs to be held steady

robot runs on a Raspberry Pi 3 B+, but he says it could easily run on a Raspberry Pi Zero.

For the sudoku challenge, the robot required "an eye to see the sudoku, a brain to solve the sudoku, and a medium to communicate the solution", says Arijit. He soon realised that Raspberry Pi's Camera Module could act as the eye, its processor as the brain, and a display and speaker as the communication medium.

As long as SUDO is able to detect the puzzle placed in front of its camera, it's able to solve complex sudokus in a few seconds. The caveat, says Arijit, is that the robot finds it tricky to read the puzzle grid in poor light. When it recognises a grid, SUDO states: "I have detected a sudoku," quickly followed by "I have solved the sudoku successfully." SUDO then displays the solution on its touchscreen display.

Puzzling it out

Arijit is working on overcoming the challenge of low light, which can cause SUDO to misread numbers on a sudoku grid. He has put in many hours getting SUDO to recognise different fonts. "Accuracy was the ultimate factor, as misinterpreting numbers would give wrong results," he says.

Build a sudoku robot

01 Repurpose or 3D-print a case that will house Raspberry Pi and hold a small touchscreen display in place. Affix the screen to the case.
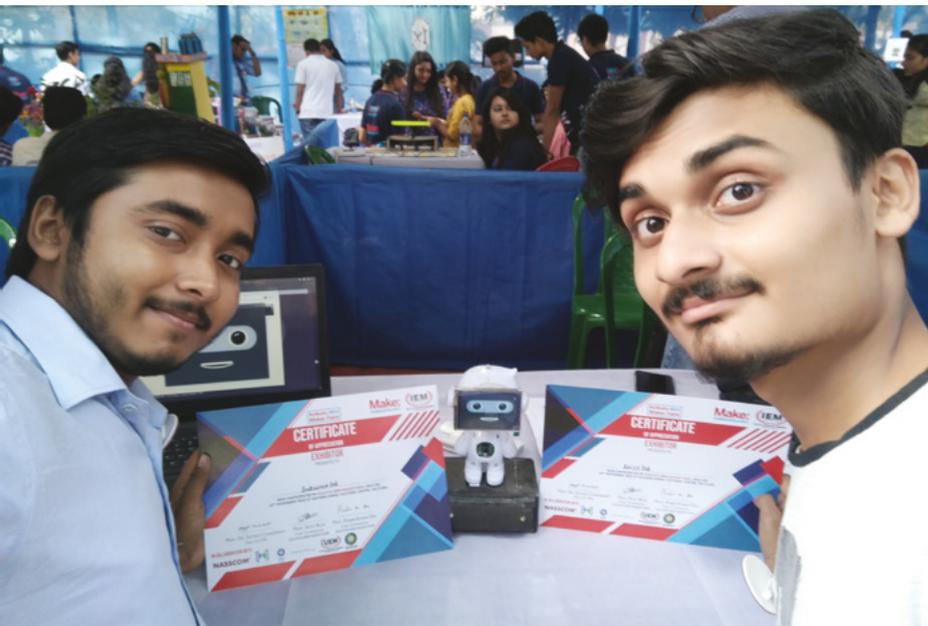
This required training the robot using machine learning algorithms to recognise all those fonts. Arijit used OpenCV (**magpi.cc/opencv**) to train SUDO. He is currently looking at ways to get his robot to follow and track a piece of paper held in front of its camera, since people often unintentionally move the paper containing the sudoku puzzle while SUDO is trying to decode it.

▲ Arijit (right) with a certificate of appreciation for the project from a Maker Faire

02 Create a hole for a Camera Module and position it in place. Use jumper wires to connect the screen to your Raspberry Pi.

> 🙾 SUDO is a Raspberry Pi-based robot that's a lightning-fast sudoku whizz 🙿

Having gained experience of machine learning when training Raspberry Pi to recognise vehicle number-plates for another project, Arijit knew which existing algorithms to use to get SUDO to solve the puzzles. He adapted them to work with Raspberry Pi and his own machine learning data set. He used open-source Python code to control the robot. The robot's appealing case was originally that of one of his childhood toys. The fully operational robot has been shown off at Maker Faires and tech events, attracting selfie-takers and kids who plead with Arijit to make them a robot to solve maths problems. SUDO even had a human challenger. Arijit proudly relates what happened: "Before the guy wrote a single digit on the paper, SUDO solved [the sudoku puzzle] and showed the solution".

Find instructions on how to build your own SUDO robot on GitHub: **magpi.cc/sudorobot**. 🅼

03 Place an IR sensor on top of the touchscreen and connect it to Raspberry Pi, then screw or clip the rear of the robot shell in place. Use sudoku grids to help train your robot to recognise numerals.

Air Hockey Robot

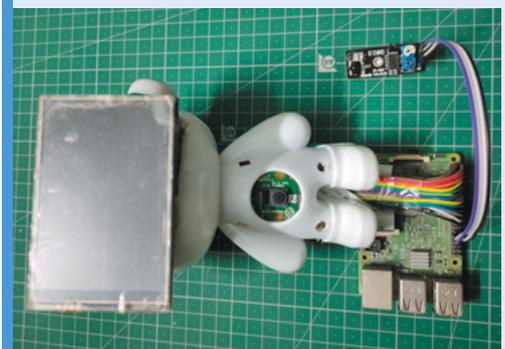Fancy pitting your wits against a robotic air hockey defender? This one is clever, very clever. **Nicola King** hones her sharpshootin' skills

MAKER

Ondřej Sláma

Ondřej studied mechatronic engineering and applied computer science. He enjoys challenging engineering and coding contests, and working on interesting projects in a team of like-minded people.

magpi.cc/ airhockeyrobot

MAKER

Dominik Jasek

Dominik is enthusiastic for modern technologies applied in DIY projects, and studied maths and applied computer science.

magpi.cc/ airhockeyrobot

When Ondřej Sláma and Dominik Jasek, two students from the Czech Republic studying at Brno University of Technology, needed to write their course thesis, they chose a rather engaging project – an Air Hockey Robot controlled by Raspberry Pi.

"I always dreamed of building some human vs machine project," Dominik tells us. "At the university, I got an idea to build an air hockey robot. What inspired me the most in designing the robot was definitely a project from JJRobots. I chose good parts from that and replaced bad ones and hopefully improved it. Luckily for me, Ondřej joined me very soon and took care of the Raspberry Pi. We made a great duo."

Out of thin air

The pair built the whole project from scratch, taking about a year. "A lot of work had to be done before even thinking about the actual implementation of the robot controlling the other side of the table," reveals Ondřej. "It would be hard to pick the most difficult element. We had to overcome a lot of challenges, including electrical wiring of all the chosen hardware, robot movement control algorithms, computer vision, game strategy algorithms, user interface etc."
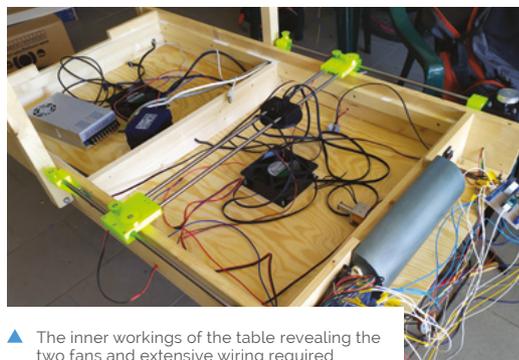
▲ The inner workings of the table revealing the two fans and extensive wiring required

After designing the table in Fusion 360, it was constructed from spruce and plywood with an Alubond playing surface. To ensure smooth gliding of the puck, a square mesh of 920 holes was drilled into the game board, enabling air to flow through from two fans located under the table.

As for the mechanical aspect, the pair opted for an 'H-bot' design to move the robot's paddle. Held in a 3D-printed housing, the paddle is moved around using a pulley and belt system, with two stepper motors controlled by an Arduino Micro. "[The H-bot design] is really the best solution for this problem as both steppers are stationary," explains Dominik.

Look and learn

The processing power for the robot's optical puck recognition and AI strategy is provided by a Raspberry Pi 4. It is connected to a Camera Module V1 mounted in the overhead part of the frame, along with LED strips to ensure good lighting. With the camera capturing frames at around 80 fps, OpenCV is used to recognise the bright green puck so its position can be determined.

For the robot's strategy, Ondřej and Dominik originally planned to use machine learning. That proved a step too far, however, given all the other fine-tuning issues that they faced in making the project. "Using machine learning was the plan from the beginning," says Ondřej. "But, trust me. We tried. We tried a lot to make it work. But it was literally impossible to implement, given how hard it is to train an agent in such a complex state space with even more complex action space."

Instead, they manually programmed four types of strategies with slightly different algorithms – you can find the code on GitHub: **magpi.cc/airhockeycode.**

The project's Raspberry Pi is also connected to a touchscreen with a GUI made using Kivy.

A Camera Module positioned on the top frame scans the table to locate the position of the bright green puck using OpenCV on Raspberry Pi
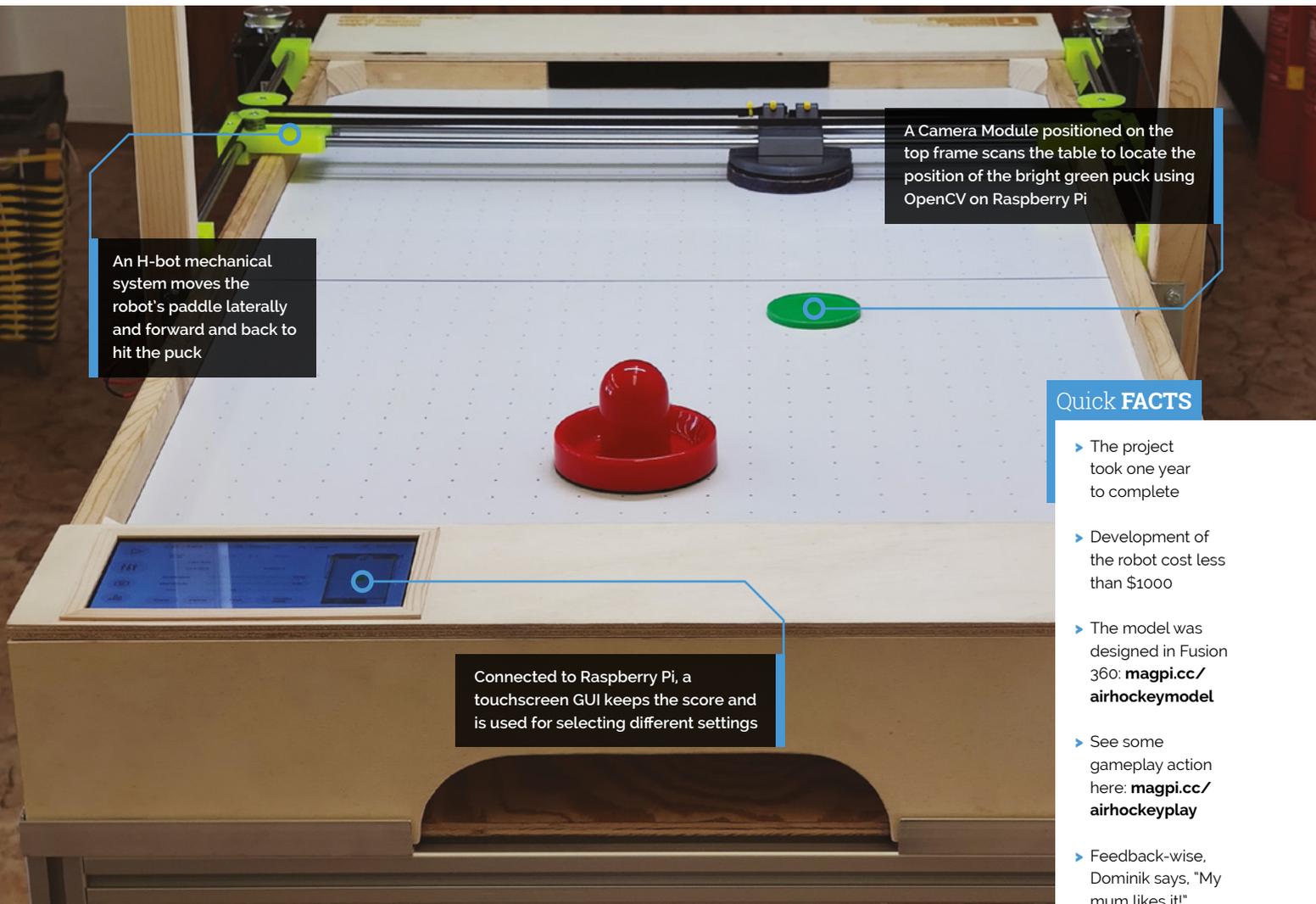
An H-bot mechanical system moves the robot's paddle laterally and forward and back to hit the puck

Connected to Raspberry Pi, a touchscreen GUI keeps the score and is used for selecting different settings

Quick **FACTS**

> The project took one year to complete

> Development of the robot cost less than $1000

> The model was designed in Fusion 360: **magpi.cc/airhockeymodel**

> See some gameplay action here: **magpi.cc/airhockeyplay**

> Feedback-wise, Dominik says, "My mum likes it!"

❝ We had to overcome a lot of challenges, including electrical wiring of all the chosen hardware ❞

Apart from the purposes of setting up the game and keeping score, this makes it possible to "set all kinds of parameters ranging from camera properties/calibration and motor speeds to the type of strategy," explains Ondřej.

The Air Hockey Robot was a very complicated and time-consuming project, but the result is indeed a brilliant piece of engineering and programming, where only the quick-witted can win. So, how often have they actually beaten the robot? "30-40% of the time. More at the beginning when things were not tuned out," says Dominik. "But, it got harder and harder. Especially for the not so good players that we are." M
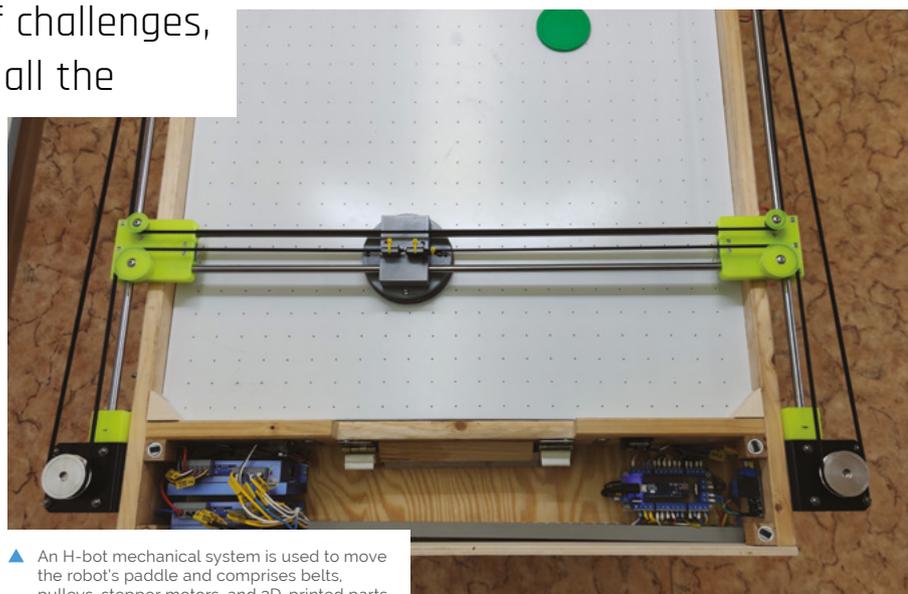
▲ An H-bot mechanical system is used to move the robot's paddle and comprises belts, pulleys, stepper motors, and 3D-printed parts

Weather Station

Çağan Çelik is making sure he won't caught in the rain without an umbrella. **David Crookes** checks the forecast

MAKER

Çağan Çelik

Çağan Çelik is a full-time hardware editor on a Turkish technology magazine. He has also been making tutorial and DIY videos on his YouTube channel about hobby electronics and programming in his spare time.

magpi.cc/ airstationyt

When fingers began touching the keyboard to write about this project, the sun was shining and the room was sweltering. By the time this article was complete, the streets outside were being pounded by rain, lightning lit the sky, and thunder was battering eardrums. If anyone needs a weather station, it's us.

As luck would have it, Çağan Çelik has created one. In fact, he's developed two – the first being five years ago. "The original was made using an Arduino Uno when I just starting to learn electronics and programming," he recalls. "Where I used to live, the weather was often unpredictable and I wanted to know the forecast so I could dress accordingly just by glancing at a screen."

That version had outdoor sensors hooked up to the Arduino using long cables and the values were displayed on a small Nokia 5110 screen. "It was a neat project, but I hit the limitations of Arduino," Çağan continues. "I couldn't add functions to it because of the 32kB of code memory, and I wanted to include an early warning system when the air pressure suddenly dropped, which mostly means it's going to rain or snow."

Cue Raspberry Pi, with the advantage of being able to show the results on a large and easily readable screen. "I could use fancy graphics if I wanted and also be far less conservative about my programming thanks to being able to use a massive microSD card as a storage medium. With a lot of CPU muscle in the background, it was a win for bringing my project alive again."

Inside and outside

To start with, Çağan wanted to expand the functionality of the weather station by adding wind speed to the temperature, humidity, and air pressure sensors of the original. He also wanted to present the information more clearly.

"The most important thing was separating the inside and outside modules because, in the previous iteration, the inside module – the Arduino and the screen – had to be placed near the window so that the cables would reach the exterior sensors," he explains. By using wireless communication, he could dispense with wires. "I was very familiar with NRF24L01 transceiver modules, so I picked this."
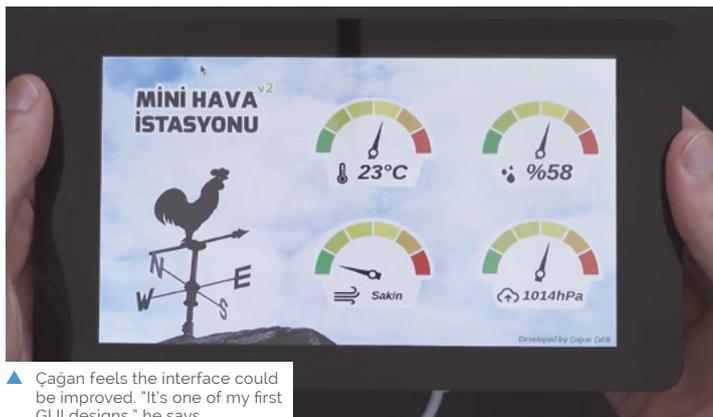
▲ Çağan feels the interface could be improved. "It's one of my first GUI designs," he says
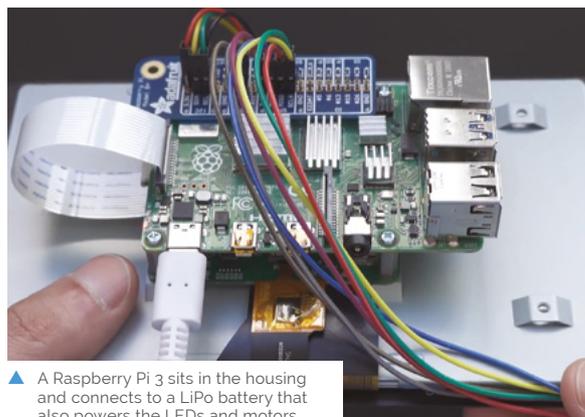
▲ A Raspberry Pi 3 sits in the housing and connects to a LiPo battery that also powers the LEDs and motors

A basic 12 V 140 mm computer case fan roughly measures the air speeds – attached spoons catch the air!
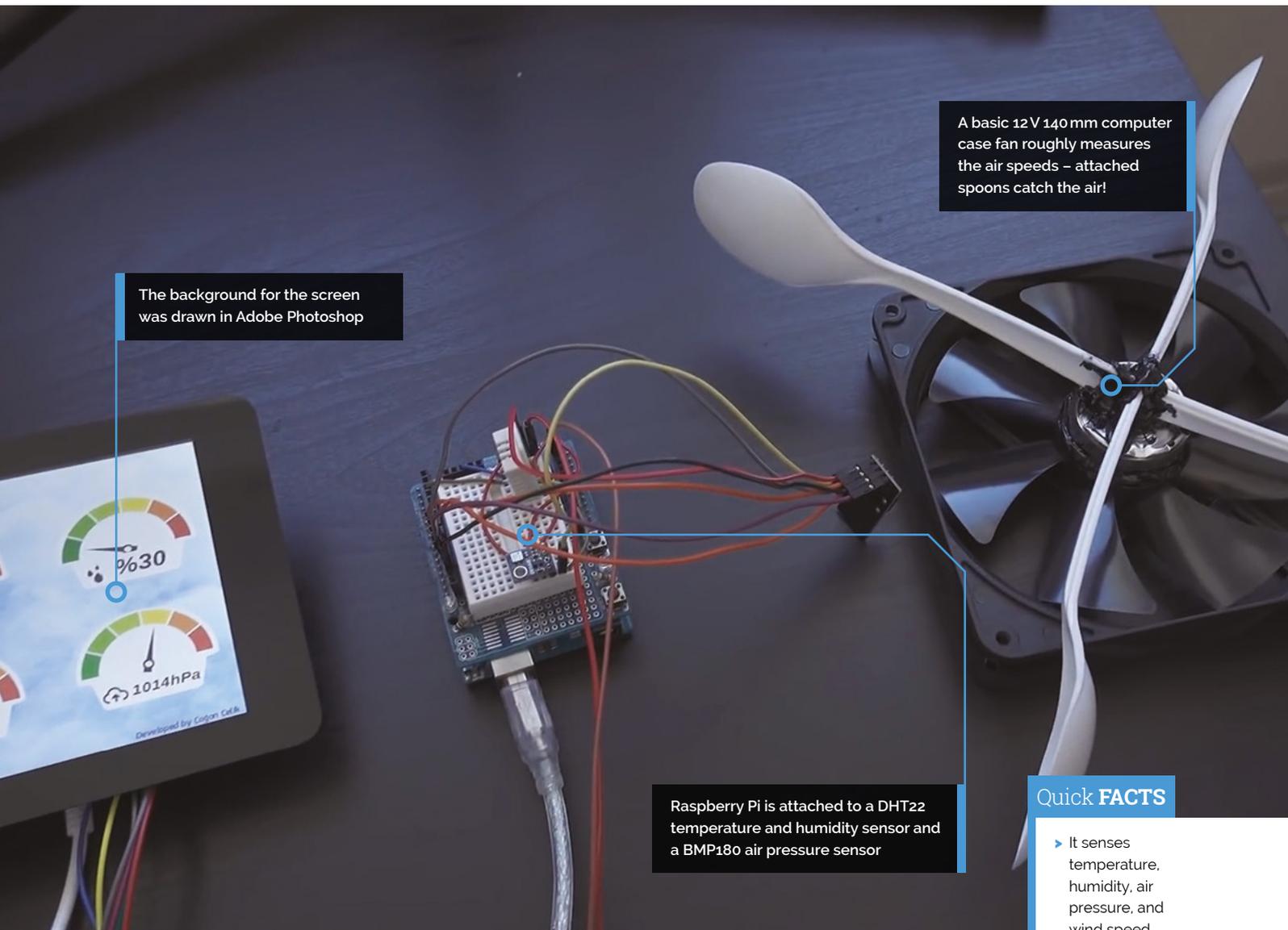
The background for the screen was drawn in Adobe Photoshop

Raspberry Pi is attached to a DHT22 temperature and humidity sensor and a BMP180 air pressure sensor

Quick **FACTS**

> It senses temperature, humidity, air pressure, and wind speed

> Sensor information is sent wirelessly

> Data is presented on a 7-inch touchscreen

> Have fun creating your own user interface

> Watch Çağan's video for the schematics

Even so, Çağan had never used this transceiver module with Raspberry Pi before. He wanted the latter to communicate with an exterior module based on an Arduino Uno, so he headed online to research and troubleshoot his way through and eventually got the two to talk to each other.

Sensor fun

The Arduino Uno is used to gather information from the sensors at set intervals before transmitting the data wirelessly to a Raspberry Pi 4 located inside his house. This is attached to a 7-inch official Raspberry Pi touchscreen running a clean user interface that he created using the GUI framework Tkinter.

"It was a new thing for me, although I realised it's fun once you grasp the basics," he says. "I learned how to use transparent PNG files with

> **"** I wanted to know the forecast so I could dress accordingly just by glancing at a screen **"**

the Tkinter interface and how to resize and manipulate them."

Now that the weather station is up and running, Çağan doesn't get caught by the ever-changing weather any more, but he admits he could easily get snowed under with extra work. "Maybe I could show the time and date, the wind direction, and the temperature inside using another sensor," he suggests. "But I'm pleased at the good feedback I've had so far." M

© Sally Mitchell

MAKER

FoAM Kernow

The Sonic Kayaks project is led by Amber Griffiths and David Griffiths at the non-profit organisation FoAM Kernow in Cornwall, in close collaboration with sound artist Kaffe Matthews, environmental science researchers James Duffy and Jo Garrett, and pro-kayaker Claire Eatock.

fo.am/studios/ kernow

Sonic Kayak

While gathering valuable environmental data, this waterborne sensor system plays ambient sounds. **Phil King** grabs his paddle

Part environmental sensing system, part musical instrument, the Sonic Kayak (magpi.cc/sonickayak) has been developed and fine-tuned over the past four years by the FoAM Kernow studio, run by Dave Griffiths and Dr Amber Griffiths. After helping out on Sonic Bikes, a sound-art project started by Kaffe Matthews, they thought about bringing the bikes to Cornwall, "but it's a dangerous place for cycling with very few cycle paths and narrow roads," Amber tells us, "so we started talking about making a version for boats."

With a background in wildlife conservation genetics, she thought it would be good to add some environmental sensors and try using it to gather underwater data. "So it's gone from being a pure sound-art project on bikes, to something that has both sound-art and scientific purpose on kayaks. Kaffe's now working on putting some of the environmental sensors onto the Sonic Bikes, so it's going full circle."

▲ One of the first attempts at mapping the underwater temperature data. Some test data can be found at **magpi.cc/kayaktestdata**

Each Sonic Kayak kit is equipped with four environmental sensors. Water temperature is measured using a digital thermometer, water cloudiness using a homemade sensor consisting of an LED and an LDR in a dark tube (the amount of light that hits the LDR is a measure of how cloudy the water is), underwater sound using a hydrophone, and above-water air pollution using a laser dust sensor. There's also a GPS unit to record the kayak's geographical co-ordinates, which enables mapping of the environmental data gathered.

Water music

Sounds based on the data are played through a waterproof Bluetooth speaker to the kayaker. "This does two things," explains Amber. "Firstly, it gives people a real connection to the underwater world that is otherwise very hard to understand. Secondly, if you are interested in gathering a particular type of data, for example pollution coming from a large boat, if you can hear when you are entering a polluted area then it allows you to follow that data and collect exactly what you need."

The system has also proven useful for people with visual impairments. "We've been working with an accessible kayak club to develop the system for navigation purposes, to allow people to kayak independently even if they have little or no sight."

▼ A diagram showing how all the components are connected. Raspberry Pi also handles all the data and audio processing
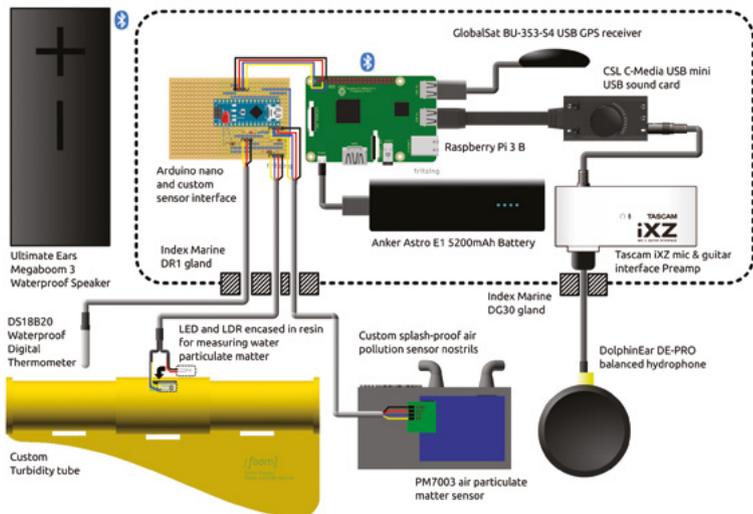
Connected to the kayak with Velcro straps, a homemade turbidity sensor comprises a tube with an LED and LDR sensor at either end to measure how cloudy the water is

A DolphinEar DE-PRO balanced hydrophone is attached to a fishing weight to keep it underwater. The signal is sent to Raspberry Pi via a pre-amp and USB sound card

Waterproofing the electronics, including a Raspberry Pi 3B, was one of the most challenging parts of the project

❝ This type of data is really hard to get, particularly for places close to cliffs or in tidal estuaries ❞

The kit's electronics – including a Raspberry Pi 3B and Arduino Nano – are housed in a Bopla Bocube plastic enclosure that fits into the rear of the kayak. The combination of electronics with salt water has proved problematic previously. "One of our earliest Sonic Kayak events was the British Science Festival in 2016, and on the second day the sea was very choppy and both kayaks capsized with the systems on them," recalls Amber. "We watched all the electronics break apart and wash up on the beach, then almost instantly corrode due to the salt water, which was rather painful! Since then, we've got very good with embedding electronics in resin, and using rubber seals for the sensor cables."

Valuable data

While the data gathered so far by the project is proof-of-principle, it may well be of interest to professional researchers. "This type of data is really hard to get, particularly for places close to cliffs or in tidal estuaries, where it's difficult to take large research boats," says Amber. "Our system also means you can get very fine-scale data, as opposed to the broader scale sea temperature data you get from satellites, or the data you can get from attaching sensors to buoys."

However, Amber says they're more interested in getting it into the hands of people who want to use it to map their local environments. "We've started working on ways to visualise the data from the Sonic Kayaks on maps, and would love to make a portal where people can upload the data from their kayaking trips and have it visualised automatically." M

▲ Out on the water around Falmouth Harbour, gathering environmental data from industrial areas, places where people live on houseboats, where there might be farm run-off etc.

CubeSat
dual-redundant
flight computer

The industrial version of Raspberry Pi, Compute Module lends
itself well to space travel, as **Rosie Hattersley** discovers

MAKER

Alex Pirie

University of Surrey
final-year student
Alex is taking an
MEng in Electronic
Engineering with
Space Systems and
worked with Surrey
Space Centre for
this project.

magpi.cc/obcviv

Raspberry Pi is no stranger to space travel. Each year, kids taking part in Astro Pi missions send code to the International Space Station where two Raspberry Pi computers duly conduct the experiments they demand. Raspberry Pi's Compute Module has also earned its space explorer badge: it's a popular component for CubeSat small form factor satellites. "CubeSats represent a low-cost opportunity that allows individuals and universities to gain experience designing and building spacecraft", explains Alex Pirie.

The University of Surrey student got the chance to work on a live CubeSat project with Surrey Space Centre, drawing on his undergraduate dissertation research. The AAReST (autonomous assembly of a reconfigurable space telescope) mission that he worked on takes a modular approach to setting up a large-scale telescope in space. Its dual-redundant flight computer is equipped with two Compute Modules. Alex worked on its PCB (printed circuit board).
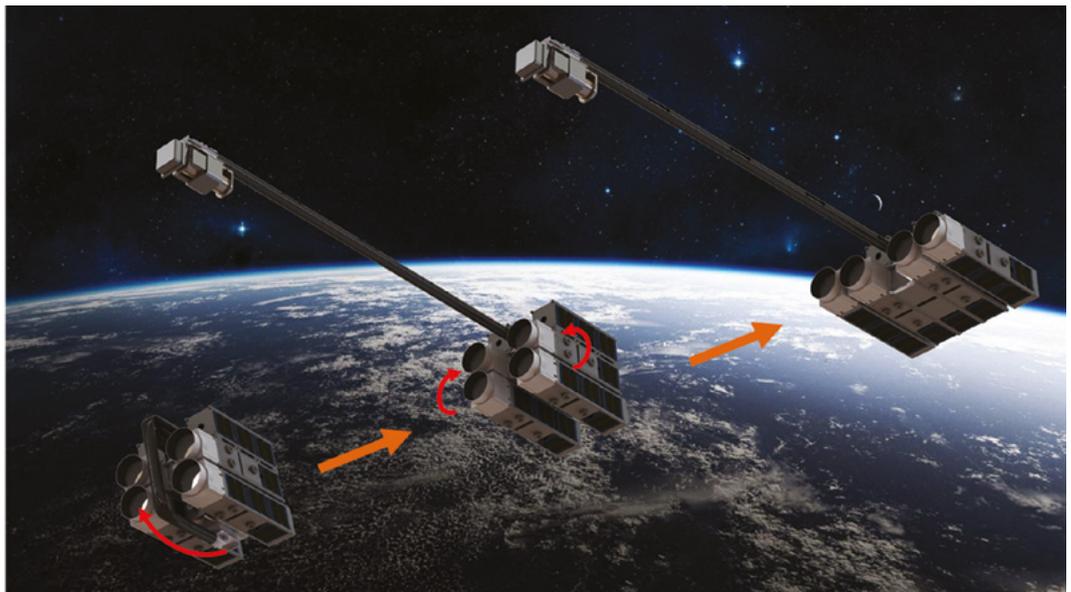
▶ AAReST concept
showing how each
CubeSat docks to
form part of the
modular satellite.
*Credit: AAReST
collaboration*
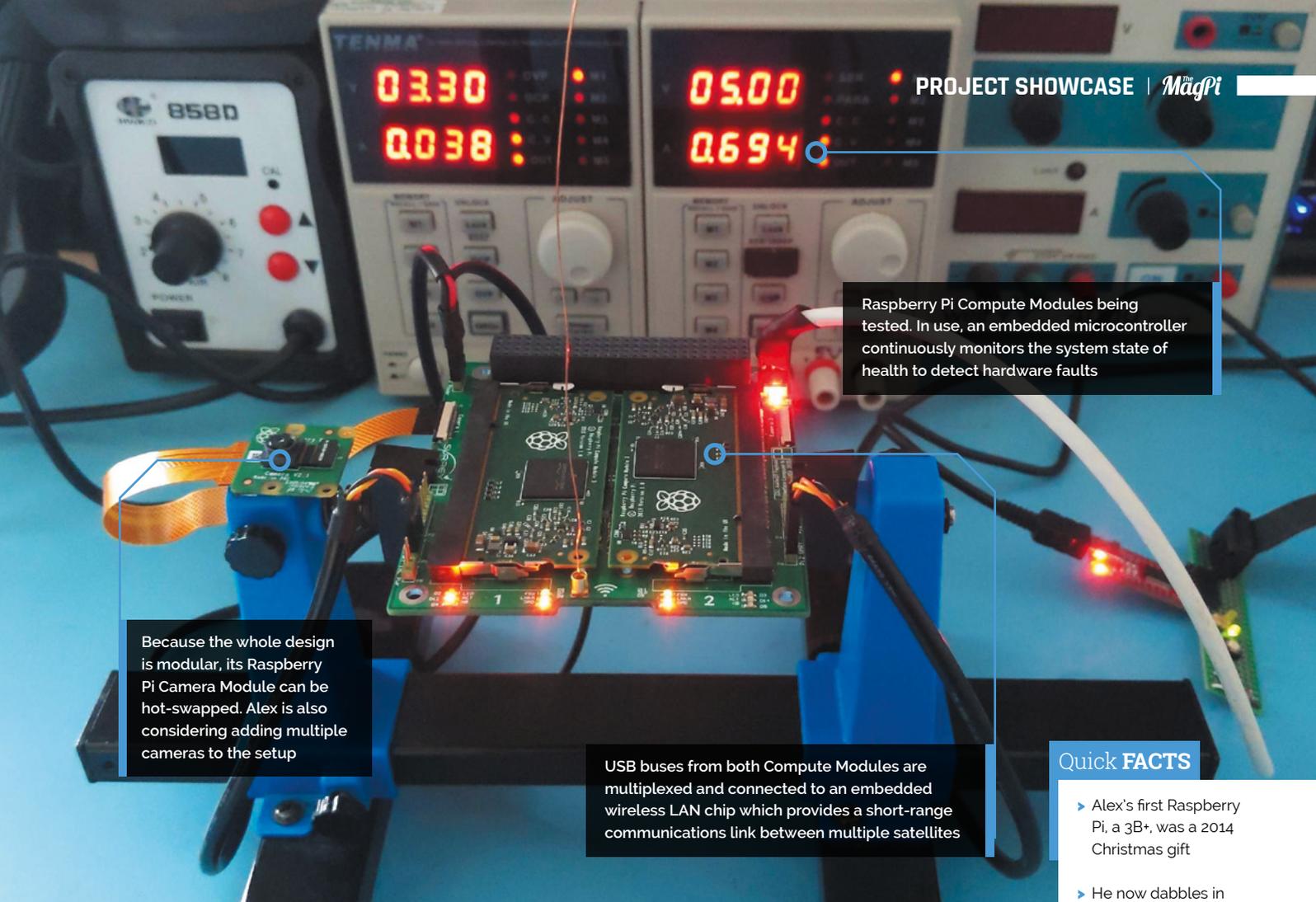
Raspberry Pi Compute Modules being tested. In use, an embedded microcontroller continuously monitors the system state of health to detect hardware faults

Because the whole design is modular, its Raspberry Pi Camera Module can be hot-swapped. Alex is also considering adding multiple cameras to the setup

USB buses from both Compute Modules are multiplexed and connected to an embedded wireless LAN chip which provides a short-range communications link between multiple satellites

Quick **FACTS**

> Alex's first Raspberry Pi, a 3B+, was a 2014 Christmas gift

> He now dabbles in robotics as well as space tech

> His university dissertation opened the door to the CubeSat project

> CubeSats aren't just dummy spacecraft...

> NASA's Marco mission sent a CubeSat to Mars
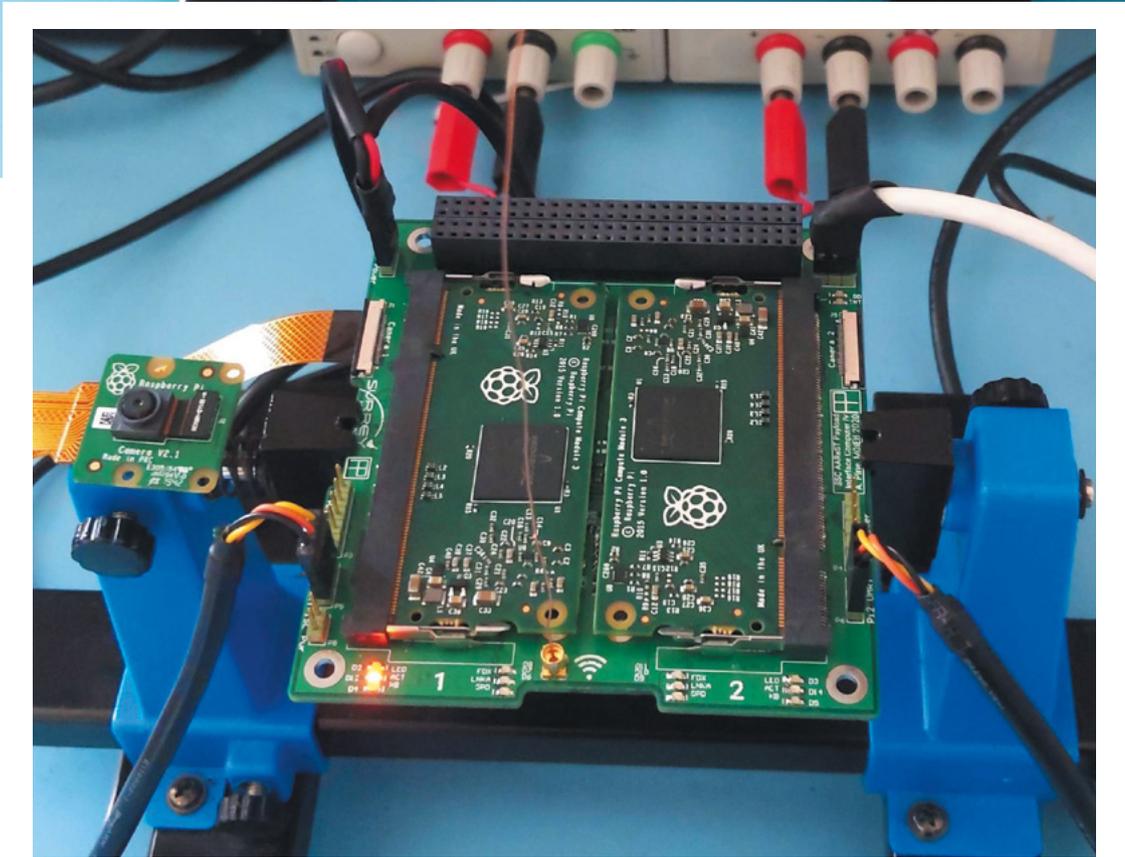
◀ The flight computer's capabilities are centred around its dual-redundant architecture: two independent, mirrored Raspberry Pi systems that communicate and synchronise data at a high speed

CubeSat design insights

A CubeSat uses readily available parts, making it possible for enthusiasts to send their own satellite into orbit.
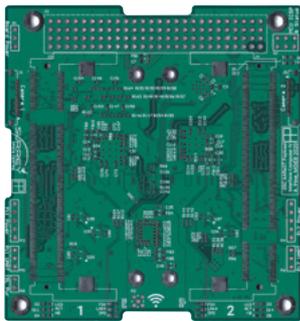
01 Use PCB prototyping software to design your CubeSat. Alex suggests developing the PCB schematics and layout concurrently, to ensure the design fits the available area.

02 The greatest challenge is fitting all the circuitry onto a circuit board not much bigger than a coaster. Check that the on-board LAN and cross-switched cameras fit, since these are critical.
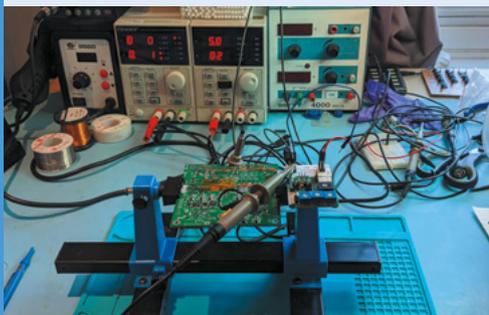
03 3D-print your design and assemble the parts. Thoroughly test your design, starting with the Compute Modules and low-level monitors, before adding USB wireless LAN and the cameras.
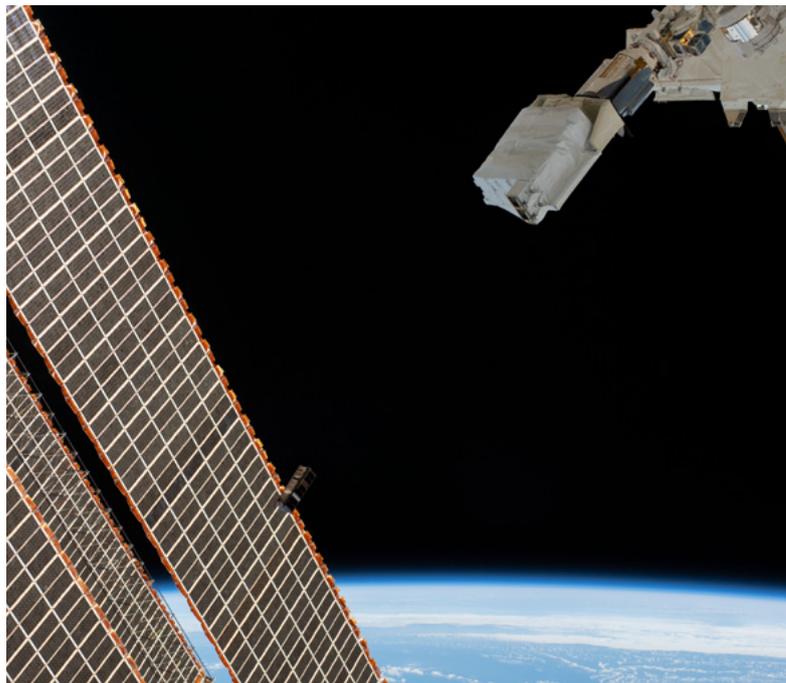
▲ A CubeSat from Singapore is ejected from the Japanese Small Satellite Orbital Deployer past the International Space Station's solar arrays. *Credit: NASA*

The project began in 2014, soon after the launch of the original Raspberry Pi Compute Module. This made a CubeSat board with redundant devices viable. "Redundancy is a common way to increase the reliability of space hardware. It has been used in the computers of Mars Rovers and the Space Shuttle," Alex notes. "The idea is that if one system fails or becomes unresponsive, another can take over and continue its task." This project was covered on the Raspberry Pi blog back in 2015 (**magpi.cc/cmsats**).

❝ **A low-cost opportunity that allows individuals and universities to gain experience designing and building spacecraft** ❞

Rendezvous in space

Alex has been working on its fourth iteration. AAReST (**magpi.cc/aarest**) will use "a WiFi inter-satellite link to allow multiple spacecraft to rendezvous," he explains. "Imagine flying into space one day and being able to connect to a satellite from your phone to conduct maintenance activities."

Raspberry Pi works well because of its low cost and compact size. It's a good alternative to the significant expense of a 'radiation-hardened' device that can withstand the extreme temperature of near space. "Raspberry Pi has many attractive
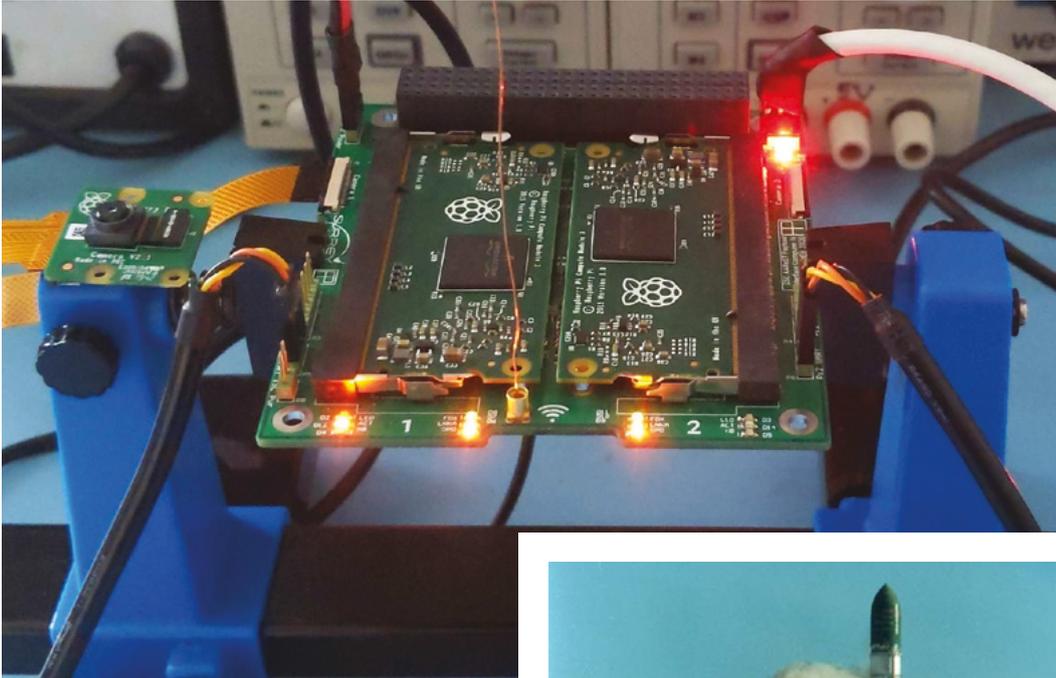
characteristics for a COTS (commercial, off the shelf) flight computer," says Alex. "It is low-cost and compact, more powerful than comparable microcontrollers, and has excellent support from the electronics and hobbyist community.

From an engineering perspective, he says "its self-contained processor system can be integrated into an application without needing to worry about hard-to-implement supporting components.

"The Compute Module's edge connector gives access to the GPIO, USB, and camera interfaces, providing the flexibility to directly integrate a Raspberry Pi into another application."

Built-in redundancy

Alex began with a partially completed blueprint from a previous student, but says this was largely a "clean sheet project". Having been tasked with adding a cross-switched Raspberry Pi camera function and improving the system monitoring and control, he realised an integrated USB WiFi chipset would be beneficial. The Compute Modules run identical images of Raspberry Pi OS Lite (32-bit). In the future, custom software code will run the dual-redundant code, possibly using Docker.

"I needed a stable software platform for running basic benchmarks and interacting with the system. Both Compute Modules run identical images of Raspbian [aka Raspberry Pi OS] Buster Lite, with modified device tree blobs to initialise

all required GPIO and buses on the right pins. During testing, pigpio (**magpi.cc/pigpio**) was used to access the functions of I²C and UART buses."

Alex explains, "The CubeSat flight computer includes an MSP430 microcontroller to detect hardware failures and switch to the backup systems as soon as possible, to avoid permanent hardware damage. If a radiation hit caused a latch-up fault in the active Raspberry Pi, it would create a short-circuit path between the supply and ground." The CubeSat would then switch to the other on-board Raspberry Pi. **M**

AI Noise Sensor

Smarter and more useful than traditional recordings, the AI Noise Sensor actually knows what is making a sound. **Rob Zwetsloot** lends an ear

MAKER

Ryan Vincent

Ryan is a 16-year-old student from London who decided to create a smarter way of determining sound.

magpi.cc/ainoise

Image recognition is a big part of AI. So is voice recognition. Determining different sounds, though, is something you don't see often, but it's something 16 year-old Ryan Vincent decided to look into due to real-life events.

"Noisy motorcycles and a neighbour's barking dog got me started on investigating noise recorders," Ryan tells us. "They all had a limitation in that they measured only noise intensity, and not really the type of sound."

From a legal perspective, traditional audio recorders can no longer be used due to data protection. To Ryan, building a device that identifies sound sources was the next step.

Machine-learned sound

"I built an AI Noise Sensor and Classifier on a Raspberry Pi using machine learning," Ryan explains. "It has a trained convolutional neural network that recognises noise – and several Python programs to monitor and classify noise, and plot infographics about the intensity and type of noise in a time-series fashion."

The system is mostly software: it uses a small USB microphone instead of any high-fidelity hardware, which does keep the price down.

"My device will make it easy for environmental managers to pinpoint sources of noise pollution, and effectively take action, including prosecuting offenders," Ryan says. "Also my low-cost and

▲ Ryan has made a video to explain how the AI Noise Sensor project came about and how it works

portable device will make it easier for the general public to use it in a much more widespread manner than other noise recording devices."

According to Ryan, the finished device can accurately identify sounds it's been trained on over 99% of the time, and has an 82% accuracy rate on sounds it was unfamiliar with.

Building a smart listener

Using a Raspberry Pi Zero was a way of making it small and portable so it can be plugged in anywhere, according to Ryan: "It is also powerful enough to run a neural network prediction program. Also, it has wireless LAN connectivity so that environmental managers can access infographics abut time-distribution of types of noise."

FLOWCHART OF PROGRAMS
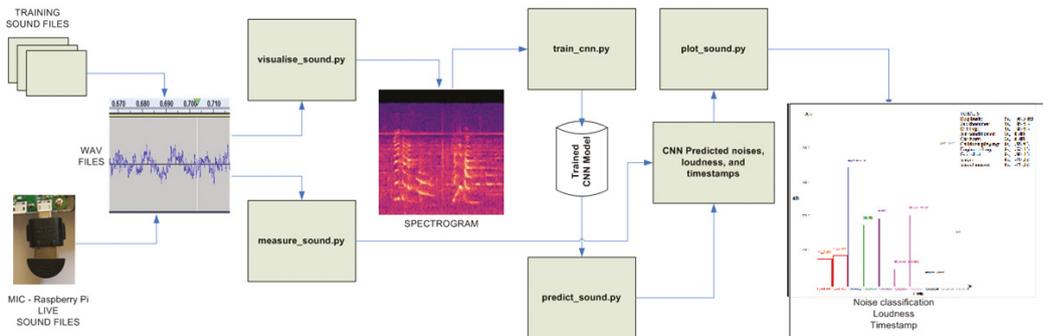
▲ The flowchart explains how the data is analysed by the neural network

NOISE POLLUTION MEASUREMENT & CLASSIFICATION USING MACHINE LEARNING ON RASPBERRY PI

FLOWCHART OF PROGRAMS
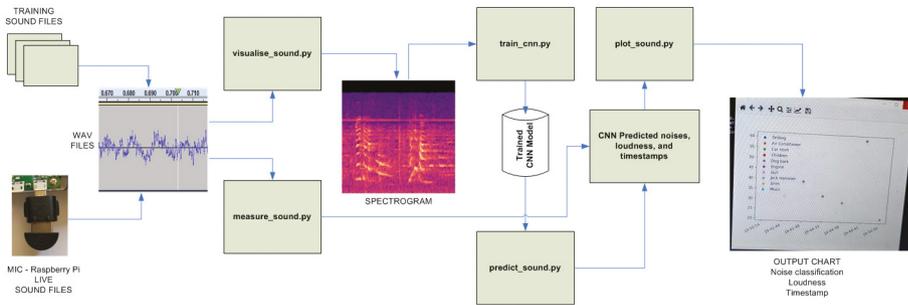
MODEL of the CONVOLUTIONAL NEURAL NETWORK

▲ For a seemingly small and simple device, a lot of the software is fairly complex
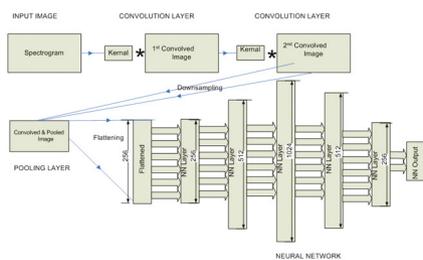
FULLY CONTAINED COMPACT DEVICE
- **Raspberry Pi Zero**
- **SD card loaded with CNN model and programs**
- **Microphone**
- **Power supply**

The system works by recording four seconds of sounds on a continuous loop. Each four seconds is then analysed for loudness and fed into the AI to try to identify the sound type. Some of the data is plotted into various charts for the user to look over as well.

The project has since won an award from Apps for Good (**appsforgood.org**), although Ryan has not finished working on it.

> ❝ My device will make it easy for environmental managers to pinpoint sources of noise pollution ❞

"I am planning a swarm of AI noise classifiers configured to work together, so that they can be installed at various parts of a building – hospitals, schools, elderly homes, offices," he mentions. "Each AI Noise classifier device could upload 'result data' to a central database within the building's WiFi. Web software can access contents of the database and plot charts for each device. It can also average the results across devices."

Ryan has managed to effectively engineer a device that usually retails for £200–£500, with a bill of materials of about £25. And it works better than expensive rivals. **M**

A Raspberry Pi Zero W is powerful enough for the AI software, and the wireless LAN makes it accessible remotely
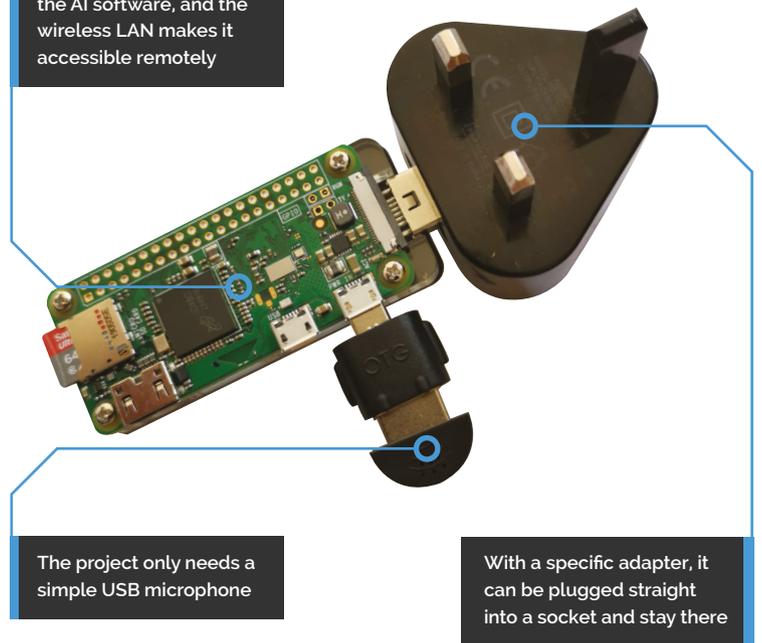
The project only needs a simple USB microphone

With a specific adapter, it can be plugged straight into a socket and stay there

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero W Starter Kit

**WITH YOUR FIRST
12-MONTH SUBSCRIPTION**

Subscribe in print today and you'll receive all this:

WORTH
£20

▷ Raspberry Pi Zero W

▷ Raspberry Pi Zero W case with three covers

▷ USB and HDMI converter cables

▷ Camera Module connector

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.
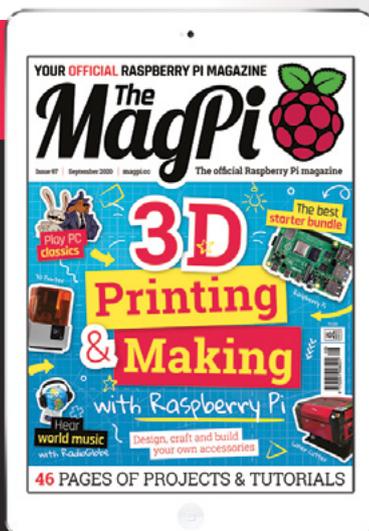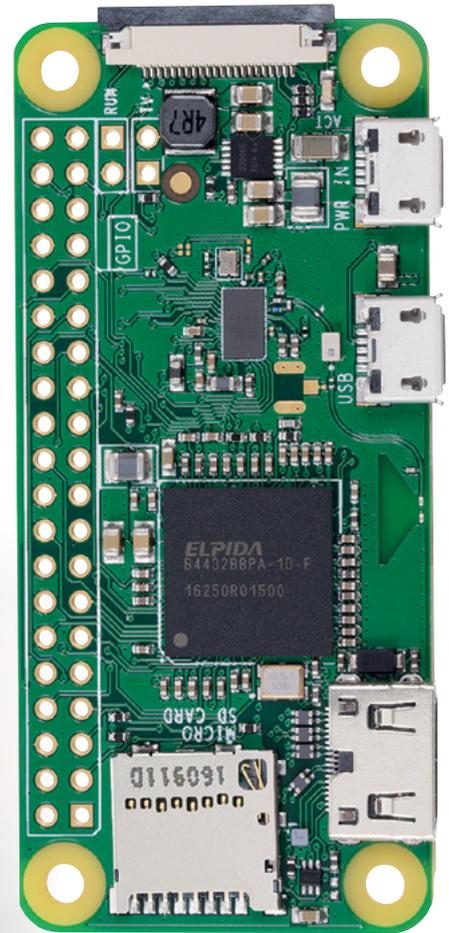
Build a Raspberry Pi Portable computer

Take your Raspberry Pi out for a walk and code on-the-go with our guide to portable computing. By **Lucy Hattersley**

Use Raspberry Pi 4 on-the-go

A touchscreen display is a good idea. It enables you to interact with Raspberry Pi OS and use the device like a tablet or touchscreen laptop

With its faster 1.5GHz quad-core CPU and up to 8GB RAM, Raspberry Pi 4 makes for an ideal mobile computing device

pi-top's Bluetooth Keyboard ($44.95) contains a keyboard and trackpad which enables you to control Raspberry Pi like a laptop

We may have mentioned this before, but team Raspberry Pi Press are quite a mobile bunch. We're based in Cambridge but live in London, Bournemouth, New York, original York, and a host of places between.

We have spent a lot of time on the road with Raspberry Pi computers, and discovering the 'best portable Raspberry Pi' solution is something we've taken a keen personal interest in.

It turns out that there are lots of different ways to use a Raspberry Pi on the go. All of them are more fun than lugging around a boring laptop. With Raspberry Pi, you can code, hack, make, and discover electronics, robotics, and computing wherever you are. Raspberry Pi as a computer is ideally suited for mobile computing, but the single-board computer requires a lot of components to make it mobile-ready.

The key components of Raspberry Pi (Broadcom architecture and ARM processor) are, in many ways, similar to those found in modern smartphones. As such, they have low energy usage and are capable of running powerful Linux applications.

Unlike a smartphone, Raspberry Pi is designed to be set up manually. It's a distinctly bring-your-own keyboard, mouse, and monitor affair. To take things mobile you'll also need to add power, typically in the form of a power bank battery.

This is where that low power requirement comes in handy. For the Raspberry Pi 4, you'll need a power adapter that outputs at least 5 volts and 3 amps. Later in this feature, PJ Evans uses a Charmast power bank (£17, **magpi.cc/powerbank**).

Once you have mobile power sorted, it's time to think about input and output. Lugging around a desktop monitor, full-sized keyboard, and mouse would be a good way to keep fit, but a terrible way to work outdoors.
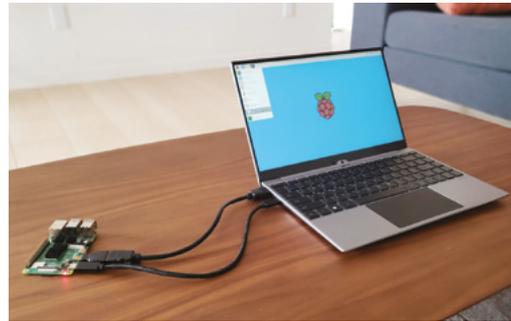
◀ Laptop docks are typically used to extend mobile phone displays, but they make a great laptop extender for Raspberry Pi 4

Fortunately, there are many small screen solutions for Raspberry Pi, including the official 7-inch Raspberry Pi Touch Display (£58, **magpi.cc/touchscreen**) and OKDO's smaller 5-inch 720×1280 HDMI IPS LCD Display (£35, **magpi.cc/lcddisplay**) There are also tablet-style cases like RasPad 3 ($186, **raspad.com**).

❝ We have spent a lot of time on the road with Raspberry Pi ❞

For our Raspberry Pi 2Go build, starting on page 38, the SmartiPi Touch 2 comes into play (**magpi.cc/smartipi2**).

For the keyboard and mouse, there are many Bluetooth micro keyboards available. In Raspberry Pi 2Go we use Pimoroni's Mini Wireless Bluetooth Keyboard with Touchpad (£18, **magpi.cc/minikeyboard**). The Pi Hut has a larger Full Sized Keyboard with Trackpad (£36, **magpi.cc/wirelesskeyboard**).

We also built a smaller micro laptop computer in *The MagPi* issue 74 (**magpi.cc/74**). That laptop used an even smaller PiTFT 3.5-inch screen, Rii miniature keyboard, and a LiPo battery.

▼ Raspberry Pi touchscreen displays are used to create powerful tablet devices
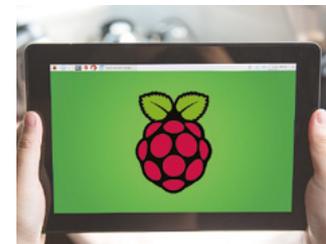
USB-C and iPad Pro or Chromebook

The USB-C connection on Raspberry Pi 4 can be used to share both power and data. A relatively simple configuration process (**magpi.cc/ipadpro**) enables the device to appear as a network connection in iPad Pro. Use Screens (**magpi.cc/screensios**) and Termius (**magpi.cc/termius**) to remotely access Raspberry Pi OS.

Ben Hardill (**magpi.cc/hardill**) has a script for iPad Pro and assures us that one for other laptop computers and Chromebooks is on the way. Look out for an updated version soon.

Popular laptop options

RasPad 3

£141/$186

RasPad (**magpi.cc/raspa**d) designs cases and screens that turn Raspberry Pi 4 into a tablet device. The latest RasPad 3 works with Raspberry Pi 4 and contains a 10.1-inch touchscreen, speaker, and battery. We reviewed an earlier version of RasPad in *The MagPi* issue 69 (**magpi.cc/69**).

NexDock Touch

£205/$269 + shipping

Laptop docks, such as those made by NexDock (**magpi.cc/nexdock2**) are terminals with a keyboard, screen, and trackpad. They use a separate device (such as your Raspberry Pi 4 or an Android phone) as the brains. NexDock Touch is currently in pre-order (**magpi.cc/nexdocktouch**).

pi-top 4 DIY Edition

£94/$100

The new pi-top 4 DIY Edition (**magpi.cc/pitop4diy**) encloses Raspberry Pi 4 with a battery, four buttons, a small OLED information display, and custom expansion options. An optional pi-top Bluetooth Keyboard (£42/$45) and FHD Touch Display (£179/$190) make this a neat-looking solution to using Raspberry Pi on the move.

The Raspberry Pi 2Go build is larger, and more ambitious and (frankly) more usable. It's also easier to build, so we hope you have a lot of fun with it.

Get your kit on

If building your own Raspberry Pi 2Go computer sounds a little daunting, or if you'd prefer a ready-to-go Raspberry Pi laptop computer, then take a look at one of the kits available.

The biggest name in Raspberry Pi laptops is pi-top, which has been building portable Raspberry Pi solutions from the start. Older pi-top devices are striking lime-green laptops with a Raspberry Pi inside. They're still available from reseller OKDO (**okdo.com**), but the pi-top 2 and pi-top 3 models use older Raspberry Pi computers that aren't as fast as the new Raspberry Pi 4.

> **" There's joy to be found using Raspberry Pi as a portable computer "**

The newer pi-top 4 model has a very different approach. It's a case with a Raspberry Pi 4 inside. Within the case is a five-hour internal battery and a fan cooling system. A 128×64 OLED mini screen and four buttons on the case provide information on battery life, CPU usage, and networking.

A new pi-top 4 DIY Edition (£94/$100) brings the device to a wider consumer market at a much more affordable price (you will need to bring your own Raspberry Pi 4 to the mix, but this enables you to switch to the latest Raspberry Pi 4 with 8GB RAM if you want all the power on-the-go).

pi-top also announced a new Bluetooth Keyboard ($45) and FHD Touch Display ($190), a sleek-looking 11-inch touchscreen tablet-style display for Raspberry Pi 4. We haven't tested either the new screen or keyboard yet, as they've just been

▶ pi-top is back with a new laptop solution based on its pi-top 4 DIY Edition. The package combines a smart-looking touchscreen with a Bluetooth keyboard and the pi-top 4 case

announced at press time. But the combined set of pi-top 4, Bluetooth Keyboard, and FHD Touch Display is a professional-looking and appealing option. Expect a full review soon.

Another option for a pre-built Raspberry Pi laptop is CrowPi2 (£231/$300, **magpi.cc/crowpi2review**). Reviewed in *The MagPi* #97, CrowPi2 is fairly expensive, and you need to add your own battery. The presence of various electronic learning components also adds a considerable amount of bulk to the laptop. But if you're looking to learn on-the-go, it's a great mobile education station.

Third ways

There is another option to using Raspberry Pi on the move that has gained a lot of traction. Rather than build a Raspberry Pi laptop, you can use a screen and keyboard device with it.

Devices known as 'laptop docks' or 'lapdocks' are designed to connect to Android mobile phones and provide a full screen and keyboard interface. Swap out the Android phone for a Raspberry Pi and you have a screen, keyboard, and computer – if not all in one device then at least all in one place.

We tested the NexDock 2 (**magpi.cc/nexdock2**) and found it a useful way to access Raspberry Pi on the go. The NexDock Touch is on the way (**magpi.cc/nexdocktouch**) and, like pi-top 4's keyboard and screen, adds a touchscreen. It's available for pre-order, at $269 + $40 shipping to the UK (£231 plus import tax and VAT). We haven't tested it yet, but it'll be interesting to compare it to pi-top's marginally better-value offering.

One final solution that we're keen on at the moment is using Raspberry Pi 4's USB-C connection to share data and power with a compatible device. We recently connected a Raspberry Pi 4 to an iPad

Pro with this method (**magpi.cc/ipadpro**) and have plans to extend the tutorial to Google Chromebook and other laptops with USB-C connections. A static IP address enables you to use VNC (**magpi.cc/vnc**) or SSH (**magpi.cc/ssh**) to connect directly with Raspberry Pi and control it.

Of course, this requires you to own a Chromebook or iPad Pro, and the cost of each is much higher than any other solution here. But if you already have one of these, this is a neat solution.

Whatever method you choose, there's joy to be found using Raspberry Pi as a portable computer. Raspberry Pi 4 is a great Linux desktop machine, capable of powering through all your regular work tasks. On top of that, Raspberry Pi is a fantastic coding, electronics, and learning tool – Raspberry Pi OS is much more inspirational than the spinning wheel of social media, websites, and mundane office work that most laptops endure on a daily basis.

Take your Raspberry Pi to the coffee shop, or on your commute, or even into the office and you'll find a world of education and electronics hackery comes with you.

◀ Raspberry Pi makes a great portable work machine, and you can use it to learn electronics and coding

▼ A Raspberry Pi tablet machine is much more adept at coding than a typical tablet device

Build a mini laptop

A different type of laptop build can be found in *The MagPi* issue 74 (**magpi.cc/74**). The build instructions can also be found on our website (**magpi.cc/buildalaptop**). It combines a PiTFT 3.5-inch screen with a Rii miniature keyboard and PowerBoost battery. It's a much smaller version of the laptop in this feature and uses Raspberry Pi Zero W.

Make a
Raspberry Pi 2Go
portable workstation

Ready to get up and go? Build a portable Raspberry Pi workstation and you'll be able to make on the move

MAKER

PJ Evans

PJ is a writer, software engineer, and is never knowingly more than one metre away from a Raspberry Pi.

@mrpjevans

Despite its size, this Bluetooth keyboard is surprisingly easy to type on and features a multi-gesture trackpad

Using the official Raspberry Pi touchscreen makes this project a bit fancier than the average laptop

There's a great solution to portable computing in the form of Raspberry Pi and a few accessories. It's every bit as good as a regular laptop for work, and even better for learning code and electronics.

No laptop can easily recreate the power of Raspberry Pi's GPIO and the Raspberry Pi OS with which you can control it. If you want sensors, LEDs, motors, and more on the move, you need a Raspberry Pi setup you can take anywhere.

So, instead of a laptop build, we're going to make a portable maker's kit: the Raspberry Pi 2Go.

01 Prepare Raspberry Pi

As with most of our tutorials, we start by making sure our Raspberry Pi 4 is prepared and ready for use. For our project it's especially important that we do this first, as access to the microSD card will be blocked once the case is assembled. To make the most of the touchscreen, install the 'full' Raspberry Pi OS with recommended software. Make sure you then go through the usual steps of connecting to wireless LAN and updating all the software to the latest version. Additionally, make sure the camera port is enabled by clicking the Raspberry Pi logo, then Settings > Raspberry Pi Configuration > Interfaces.

02 Get on the case

Next, we start to assemble the case. We've chosen the very clever SmartiPi Touch 2 case which not only mounts the touchscreen and Raspberry Pi 4 in a neat side-by-side configuration, but also has space for an optional Raspberry Pi Camera Module. The SmartiPi Touch 2 case comes with a large number of accessories and configuration options, so have a look on their website for full instructions on how to place and secure the screen and optionally the camera. Finally, mount your Raspberry Pi 4 and connect the ribbon cables as instructed.

03 Keeping cool

The SmartiPi Touch 2 case thoughtfully comes with a fan assembly that can sit above Raspberry Pi 4. Heat from the CPU has been a hot topic in the past few months, so you may wish to take advantage of this option. SmartiPi recommends that the fan is not required if you have updated Raspberry Pi 4 to the latest firmware. Open a Terminal window and enter:

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

Whether you would still like to use the fan is completely your call. We opted to use the 'plain' cover, removing the press-out panel so the GPIO header is exposed.

You'll Need

> SmartiPi Touch 2
> **magpi.cc/smartipi2**

> Bluetooth keyboard
> **magpi.cc/ minikeyboard**

> Camera Module
> **magpi.cc/ cameramodule**

> Mini Black HAT Hack3r
> **magpi.cc/hack3r**

> Charmast 10,400 mAh power bank
> **magpi.cc/powerbank**

> BUBM carry case
> **magpi.cc/bubm**

> CamJam EduKit
> **magpi.cc/edukit**

> Iron-on patch
> **magpi.cc/patch**

▲ Assembly took about 30 minutes. Long ribbon cables are provided so the computer and camera can sit alongside the display PCB

04 A stand that delivers

You have some options when it comes to making the SmartiPi Touch 2 stable. The base's hinges slot into the hinges on the main body, enabling you to tilt the assembly to your preferred angle. This also works well for 'flattening' our Raspberry Pi 2Go for transport. If this appeals to you, make sure the base's hinges are facing towards you so the base folds backwards, flush with the screen. Another option is to use the mounting hinges supplied to connect to a custom base, or rotate the base to give you a neat little workspace (although it may be a little less stable without support!).

Top Tip 👍

Am I covered?

SmartiPi (**smarticase.com**) sells additional covers to make space for full-size HATs, and even custom circuitry.

> ❝ You have some options when it comes to making the SmartiPi Touch 2 stable ❞

05 The perfect pair

It's now time to test your setup. Connect a regular USB keyboard and mouse to check that your Raspberry Pi 4 is booting and that the screen is working and rotated correctly (it should be unless you've changed settings in **boot/config.txt** already). Once you're happy, you can pair the Bluetooth keyboard. If you've got the same model we've used, switch the keyboard on, then press **FN+C** together until the blue light flashes. Now, on Raspberry Pi OS Desktop, click the Bluetooth icon,

then 'Add Device'. When the keyboard appears, select it and click Pair, following the on-screen instructions. Now your keyboard will be working and you can disconnect the USB keyboard.

06 Time for a little hacking

If we want a portable Raspberry Pi kit for physical computing, then we need access to the GPIO pins. As the header is now on the back of the screen, we'll need to extend it in some way. Luckily, Pimoroni has the perfect solution with the Mini Black HAT Hack3r. This is a board with two GPIO breakouts and a ribbon cable that slots into Raspberry Pi 4's GPIO. Now you've got two GPIO headers to play with that can sit in front of the SmartiPi Touch 2 case. Installation is simple enough, but does involve some careful cable bending and must be done with everything switched off.

07 Ready for your close-up?

If you've gone down the camera option, so you can videoconference with some of your Raspberry Pi making friends, you may notice a problem. The camera has been mounted sideways, so that means you're sideways too. To resolve this

and get yourself the right way up, open up a Terminal window and enter this:

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

Find the final line, which reads `exit 0`, and on the line *above*, add the following:

```
v4l2-ctl --set-ctrl=rotate=90
```

Now exit Nano with **CTRL+X** followed by **Y**. Reboot to get your camera working correctly.

---

08 **You have the power**

As the case is assembled and the Bluetooth keyboard is working, you're probably down to just one single wire: the power supply. For better portability, we can use a battery pack to power the Raspberry Pi 4, screen, and camera. We chose the Charmast unit for many reasons: its build quality, compact size, 10,000 mAh capacity, but most of all its ability to output 3 amps of power over USB-C. This is essential if you want to run a Raspberry Pi 4 and accessories without voltage drops. The power bank connects via a short cable to the SmartiPi Touch 2's custom USB-C power lead which directly powers both Raspberry Pi 4 and touchscreen.

---

09 **Sound investments**

Our build doesn't have any sound output, but if this is something you need, then you have some options. You can connect the 3.5mm AV jack on Raspberry Pi to an active device such as a battery-powered portable speaker. Alternatively, a more permanent solution would be to get a small amplifier such as Adafruit's MAX98357A (**magpi.cc/max98357a**) and a 3W speaker. A fun option is to add a Pimoroni Pirate Audio Speaker (**magpi.cc/pirateaudio**) to one of the GPIO headers

of the Mini Black HAT Hack3r, for decent-sounding audio with an OLED screen.

---

10 **Package up**

We've now got all of our parts ready. To avoid carrying them around loose, we'll need a case – the form factor favoured by case manufacturer BUBM is perfect for the SmartiPi Touch 2. Our chosen case has two compartments: a deep space with Velcro dividers and a shallower space for cables with a zip-pouch and elastic restraints. Providing the SmartiPi Touch 2 base is mounted pointing backwards, you can fold it flat and place it in the deep area with the keyboard on top. The Mini Black HAT Hack3r and power bank fit easily into the shallower area.

---

11 **Add goodies**

Now the fun part: adding goodies to augment your portable Raspberry Pi 4. If you want to do physical computing on the go, the first item we would recommend is a small breadboard. Then you need something to plug into that breadboard. If you're starting out in electronics, a CamJam EduKit #1 is an inexpensive kit that gives you some first 'supplies' as well as worksheets for your first physical computing adventure. For a finishing touch, we also got an Adafruit Raspberry Pi logo iron-on patch and adhered it to our case. Now we can make a project with Raspberry Pi anywhere.

---

12 **Alternatives**

This project can be customised in various ways. Go larger with a bigger screen – or smaller with a HAT-based touchscreen, as in the mini-laptop project we made in *The MagPi* #74 (**magpi.cc/74**). You can also add to the kit with HATs such as a Pimoroni Enviro (**magpi.cc/enviro**) or Breakout Garden (**magpi.cc/breakoutgarden**) which supports up to six different sensors/displays at the same time. If that's a bit much, Pi Supply's Traffic HAT (**magpi.cc/traffichat**) gives you three giant LEDs, a button, and a buzzer in a single package – ideal for beginners. Get creative on the move. M
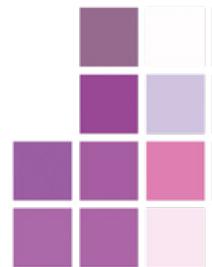
Power armour
for your Raspberry Pi 4

Mini OLED screen & programmable buttons

Integrated 1-watt loudspeaker

Integrated heat sink & centrifugal cooling fan

5-hour internal battery

UPS-style soft shutdown system prevents SD card corruption

USB-C Power Delivery specification at 15V for faster battery charging

Full access to Raspberry Pi's GPIO pins & ports

Run project?

pi-top

*

The ultimate Raspberry Pi case to go anywhere and do anything!

pi-top [4] DIY Edition is the only Raspberry Pi 4 case with an inbuilt battery and power management system. It docks to a family of 'plates' to connect to our expanding range of sensors, motors and more.

By installing your Raspberry Pi 4 into **pi-top [4] DIY Edition**, it is upgraded into the ultimate mobile hands-on computing platform. Build anything from rovers to drones: the only limit is your imagination!

pi-top [4] DIY Edition - from £94.95 / $99.95

---

pi-top

Raspberry Pi made simple, robust and modular.

Play classic console games legally on Raspberry Pi

Discover a range of ways to buy and source classic games legally for Raspberry Pi

MAKER

K.G. Orphanides

K.G. is a writer, maker of odd games and software preservation enthusiast. They will fight anyone who claims that piracy is the only thing emulation's good for.

@KGOrphanides

Console emulation has been firmly in the mainstream in recent years. However, hobbyist emulation and DIY consoles run the risk of involving you with illegal copyrighted content.

But you don't have to be a bootlegger to build your own home multi-console emulation with Raspberry Pi and RetroPie.

Emulators themselves are strictly legal, and we've talked in the past about the wide range of homebrew and legal ROM images available (**magpi.cc/legalroms**).

In this tutorial we're going to look at a much broader range of legal console ROMs. Some can be purchased legally, while others have been developed and are distributed for free.

So let's set up a RetroPie console and play some classic games.

▲ Ecco the Dolphin is just one of the classic games in the Mega Drive Classics collections

Thriving scene

Sega's Mega Drive Classics collections include ROM images of the games that you run on any emulator you like, and brand new commercial releases for Sega and other platforms are thriving, as are active homebrew scenes bringing innovative new games to console formats that went out of production over 25 years ago.

Sega is incredibly supportive of its emulation community, and is happy to just sell you some classic Mega Drive ROMs, included in the Sega Mega Drive Classics collections for Windows, macOS, and Linux. You can buy 50 classic Mega Drive games on Steam (**magpi.cc/segaclassics**), either individually or as a pack.

Once bought, to find the ROMs, open the title's Steam Library page, clear the gear icon on the right, select properties Properties, select the Local files tab, and then click Browse local files. You'll find all the ROMs in the clearly labelled uncompressed ROMs directory. Rename all files with '.68K' and '.SGD' extensions to '.bin' and copy them over to Raspberry Pi using a USB stick or via its Samba share (**magpi.cc/samba**).

Buy new classics

If you're after new games for classic systems, **itch.io** should be your first port of call. The Nintendo Entertainment System is the most popular 8-bit console for modern developers, while the Mega Drive has won the hearts of 16-bit devs. Games are

RETROPIE SETUP

If you want to install extra emulators or update drives, you'll find all that in the RetroPie Setup menu

Connect to or disconnect from a wifi network and configure wifi settings.

You'll want to enable wireless LAN to easily copy over games, but RetroPie also includes more sophisticated networking features including online multiplayer game hosting

⚏ OPTIONS ⚏ MENU ⓑ BACK ⓐ LAUNCH ✛ SYSTEM ✛ CHOOSE

---

also available for the 8-bit Sega Master System and 16-bit Super Nintendo Entertainment System.

We've made itch.io collections for each of those platforms, going out of our way to avoid unauthorised ports and ROM hacks. These include both commercial and freeware games, plus a couple of open-source titles.

That's not the only place that you'll find releases for those platforms. In the tutorial, we download Mystery World Dizzy by the Oliver Twins. It's a wonderful example of a lost NES game that was recovered by its creators and released as freeware to the fan community, but it's also rare to find Nintendo games from that era re-released with their creators' blessing. And unlike Sega, Nintendo doesn't look fondly on ROM hacks, fan games, and the like.

On the homebrew side of things, projects such as Retrobrews (**retrobrews.github.io**) and sites like **vintageisthenewold.com** and **indieretronews.com** compile collections and lists of homemade games for classic consoles, but watch out for the odd unauthorised port slipping into their catalogues.

There's a small but lively industry releasing cartridges for retro consoles, and a number of developers and publishers make the ROM files available online, either for free or a small price. Among these are RetroSouls (**retrosouls.net**), the team behind Old Towers, and Miniplanets publisher Playonretro (**playonretro.itch.io**).

◄ If you buy Sega's Mega Drive Classics collection on Steam, you'll get emulator-friendly ROM files for 50 games, including Golden Axe, Ecco the Dolphin, and Sonic the Hedgehog

❝ Sega is incredibly supportive of its emulation community ❞

Eight modern Mega Drive games

Here are some of the best new Mega Drive games:

Tanglewood – magpi.cc/tanglewood
Miniplanets – magpi.cc/miniplanets
Devwill Too – magpi.cc/devwilltoo
Arkagis Revolution – magpi.cc/arkagis
L'Abbaye des Morts – magpi.cc/labbaye
Old Towers – magpi.cc/oldtowers
Irena: Genesis Metal Fury (demo) – magpi.cc/irena
Cave Story MD – magpi.cc/cavestory
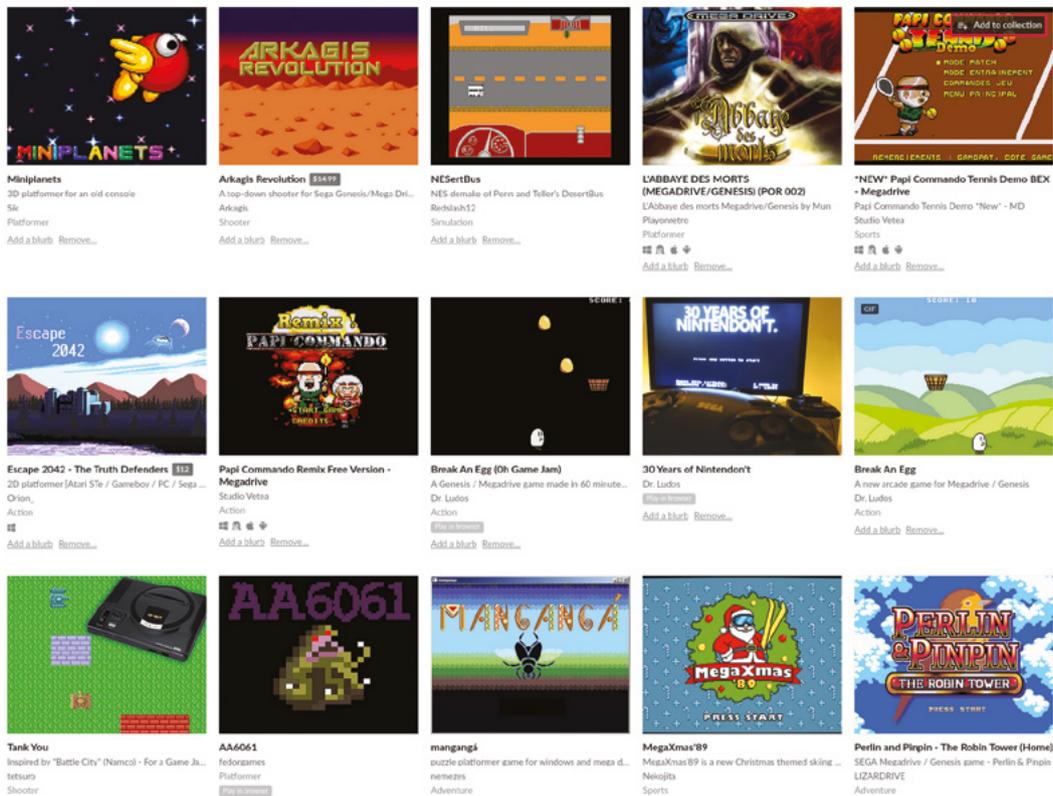
New developers publish games for classic consoles on popular indie platform itch.io

Top Tip 👍

Handheld paradise

If you'd rather build a handheld console, then that's a very viable prospect using a chassis such as the Retroflag GPi Case or the Waveshare Game HAT.

01 Image your RetroPie drive

Download the Raspberry Pi Imager for your operating system from **magpi.cc/imager**. Insert a microSD card – 8GB will be fine if you plan to limit yourself to 8- and 16-bit games, but if you want to emulate more powerful consoles in future, a 32GB card is a good investment.

Run Raspberry Pi Imager and pick RetroPie from the choose operating system list. You want the most powerful Raspberry Pi you can lay your hands on, but a Raspberry Pi Zero will do the trick if you stick to emulating relatively early systems, and is great for embedded console projects.

Choose your microSD card, click Write, and give the software permission to overwrite any data on the microSD card. Wait for the image to be downloaded and flashed.

Itch.io console games

Use these links to find new games for classic consoles on itch.io:

Mega Drive games: magpi.cc/itchmd
NES games: magpi.cc/itchnes
Master System games: magpi.cc/itchms
SNES games: magpi.cc/itchsnes

02 Plug it in, baby

Insert the microSD card and connect Raspberry Pi to a keyboard, mouse, and monitor. If you've got controllers or joysticks, plug them in before you power up.

After the image boots, you'll be prompted to assign your gamepad's buttons, if you have one. Trigger buttons on some controllers – notably Xbox 360 compatible gamepads – may not register when pressed. Press and hold any other button to skip configuring them for now. If you make a mistake, you'll be able to go back and correct it when you get to the end of the configuration list.

03 Fix your triggers (optional)

If the triggers are unresponsive on your Xbox 360 compatible controller, you should update the xpad driver. Go to RetroPie configuration and select RetroPie Setup. From the ncurses menu, select Manage Packages > Manage Driver Packages > 847 Xpad Driver, then Update.

Exit back to the main EmulationStation interface and open the Menu. You may find that this has been remapped from Start to the Right Trigger button after the update. Scroll down and select Configure Input.
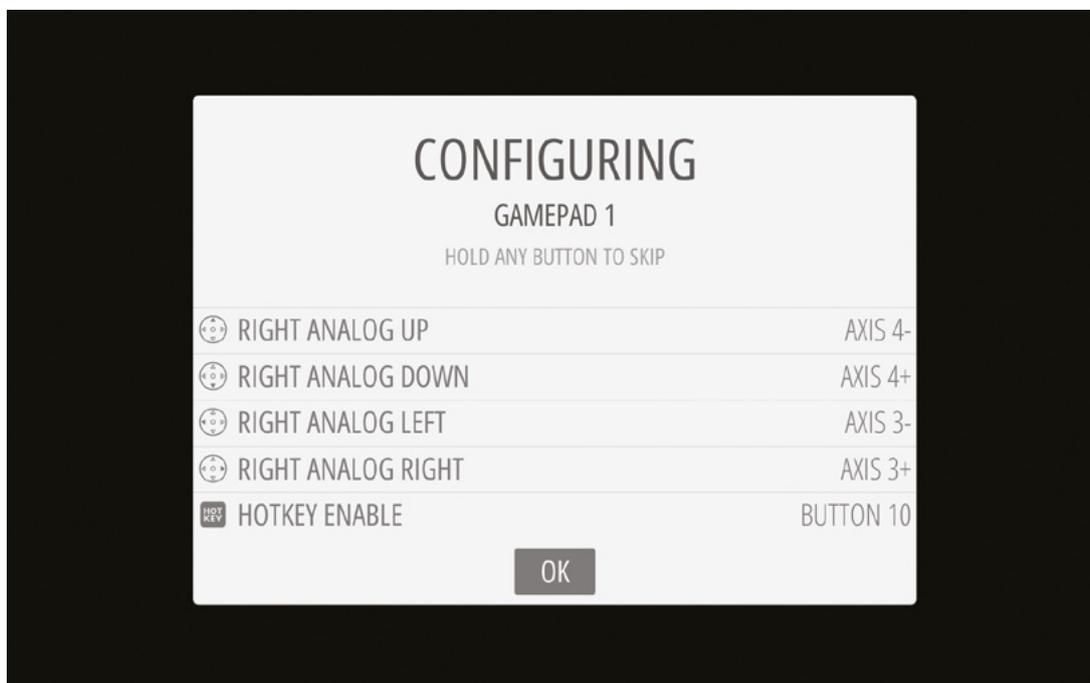
04 Set me up

With your controller configured, you'll be taken to the main interface. You won't see any emulators on offer until you've copied over games for them to play. Press A on RetroPie to enter the config menu.

Select WiFi. Press OK at the following menu to go on to connect to a wireless network. Choose from the network list and enter the network key. Select Exit to return to the main EmulationStation config menu.

Some 1920×1080 displays will show a black border. If this is the case, select raspi-config. Go to Advanced Options, then Overscan – this will get rid of the black border. Select No to disable overscan compensation. You'll need to reboot for this to take effect.

05 Get some ROMs

Before we go any further, you'll need some games to run on RetroPie's suite of emulators. For our first NES ROM, we'll grab the Oliver Twins' Mystery World Dizzy. Go to **yolkfolk.com/mwd** and click Download. To test Mega Drive emulation, go to **arkagis.com** and click 'Download trial version'

> ❝ It's easiest to download ROMs on another computer and copy them across ❞

to take Arkagis Revolution's great rotating field navigation for a spin.

It's easiest to download ROMs on another computer and copy them across your local network to RetroPie's Samba share at **retropie.local** using your file manager. Each console has its own subdirectory under the **roms** directory. Windows users should ensure that network discovery is enabled.

Four modern NES games

These new NES games are excellent examples of modern retro game development:

Micro Mages – magpi.cc/micromages
From Below – magpi.cc/frombelow
Wolfling – magpi.cc/wolfling
Legends of Owlia – magpi.cc/owlia

▶ Micro Mages is a commercial modern NES game with fantastic graphics and tight single- and multiplayer gameplay for up to four people

Top Tip 👍

What's a ROM?

ROM (read-only memory) files are data images of a non-rewritable storage medium, usually a game cartridge or – more rarely – computer or console firmware.

06 Time to play

Back on Raspberry Pi, restart EmulationStation: press Start on your controller, select Quit, then Restart System. Restart the interface every time you add games to force it to re-check its ROM directories.

If you have a keyboard connected, it's quicker to press and hold **F4** to quit to the command line, then type `exit` to restart EmulationStation.

As you scroll to the left or right, you should see logos for the NES and Mega Drive. Press A to enter the menu, then press A while highlighting the game you want to play. Right and left directional controls switch between different consoles.

Direct download ROMs

Although it's easiest to copy ROMs over from another computer, you can just download them at the command line of your RetroPie box if you have the URL. Press and hold **F4** to exit to the command terminal. You can download the ROM files directly to their directories using wget:

```
wget -P /home/pi/RetroPie/roms/nes/ http://yolkfolk.com/
flash/mwdidd.nes
```

Restart EmulationStation by typing `exit` at the command prompt. If you'd rather just download all your files to a single location and move them later, the Midnight Commander file manager accessible from the Configuration menu makes this fairly hassle-free too.

▲ Old Towers is a new homebrew Mega Drive game, available as a digital download or even as a cartridge!

07 Shortcuts, mods, and fixes

Remember the Hotkey you defined during controller configuration? You'll be using that a great deal, as it serves as a mode switch for controller shortcuts. You'll find more info at **magpi.cc/hotkeys**, but these are the most useful:

Hotkey + Start – quit the game
Hotkey + Right Shoulder – Save
Hotkey + Left Shoulder – Load
Hotkey + B – Reset
Hotkey + X – Open quick menu for save states, screenshots, recording and similar

If you don't get any audio from Raspberry Pi 4, make sure the HDMI lead connecting your monitor is plugged into the HDMI 0 port, nearest to the power connector. **M**
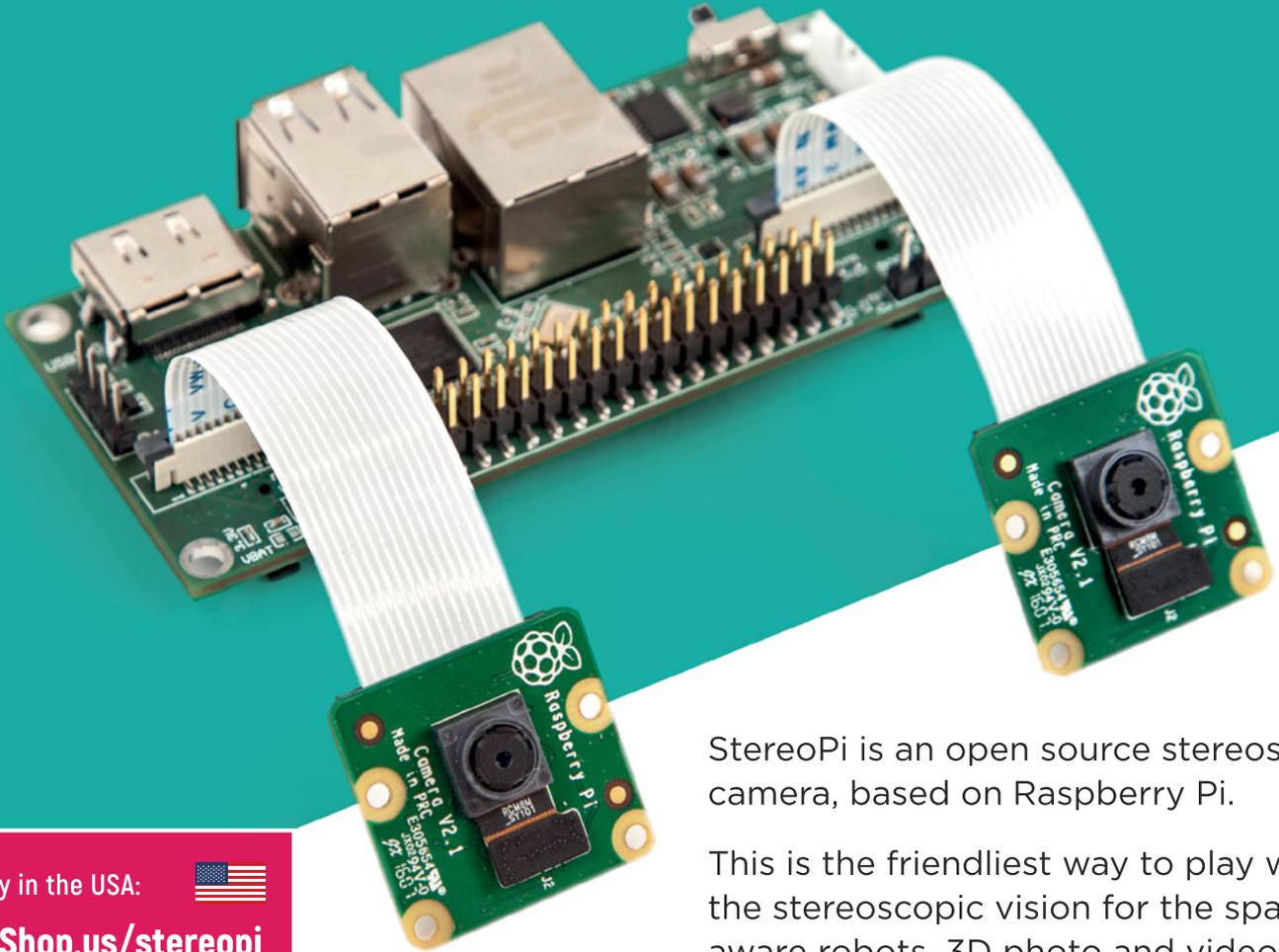
Do you know
HOW ROBOTS SEE?

StereoPi is an open source stereoscopic camera, based on Raspberry Pi.

This is the friendliest way to play with the stereoscopic vision for the spatially aware robots, 3D photo and video!

Buy in the USA:
PiShop.us/stereopi

Buy in Canada:
BuyaPi.ca/stereopi

RASPBERRY PI INSIDE

Raspbian
STOCK RASPBIAN SUPPORT

OPEN SOURCE

CROWDFUNDED PROJECT

LinuxGizmos.com
"The StereoPi can capture, save, livestream, and process real-time stereoscopic video and images for robotics, AR/VR, computer vision, drone instrumentation, and panoramic video."

MickMake
"With it you can do things like, stream stereoscopic 3D video to YouTube, build real-time depth maps using OpenCV, create panoramics using Hugin and even a 3rd person view of real life. Cool."

Raspberry Pi Blog
"There are some excellent community efforts too, of which our current favourite is this nifty dual camera board."

Hackster News
"You can hook this up to YouTube, to Oculus Go, you can use it with OpenCV.. I cannot wait to start messing around with these because it's basically a dream come true."

Elettronica In LinuxGizmos.com Geeky Gadgets OPENELECTRONICS electronics-lab

High Quality Camera:
Make a Minecraft photo booth

Part 06

Create a photo booth in Minecraft that takes photos of the real world. What will you see on your travels?

Not only is Minecraft Pi great fun to play around with, you can also use Python programming to manipulate the Minecraft world and create various structures within it. Going beyond this, you can even have it interact with the real world. In this instalment, we'll be getting Minecraft to trigger the Camera Module or HQ Camera with code whenever the player enters a virtual photo booth.

The first thing you need to do is import the Minecraft API (application programming interface). This enables you to connect to Minecraft and program it with Python. You also need to import picamera's PiCamera class to control the camera, and the time module to add a small delay between taking each photo.

The Official Raspberry Pi Camera Guide

For further tutorials on how to use your HQ Camera or Camera Module, take a look at our new book, *The Official Raspberry Pi Camera Guide*. Its 132 pages are packed with essential info and a range of exciting projects. **magpi.cc/cameraguide**

❝ You need to be able to move your player inside and onto the trigger block ❞

Open Minecraft from the applications menu (if it's not present under Games, install it via the Recommended Software tool), then enter an existing world or create a new one. Move the Minecraft window to one side of the screen. You'll need to use the **TAB** key to take your mouse's focus away from the Minecraft window to move it. This will be needed later when you switch between the Minecraft and Python windows.

Open Thonny from the applications menu. This will open up the code editor which you'll use to write the photo booth program.

Enter the code from **listing1.py**, or download it. Save with **CTRL+S** and run the program with **F5**.

You should see the message 'Find the photo booth' appear in the Minecraft world. This is the first part of the code. Now stop the program running by pressing **CTRL+C**.
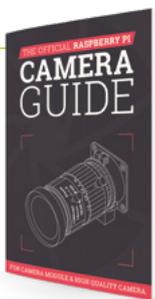
Camera tests

Next, we'll make sure the camera is set up. We've set the camera to show a two-second preview, so that you can strike your pose and smile before the picture is taken. The image is stored as a file called **selfie.jpg** in your home directory (**/home/pi**).

Now, you need to create a photo booth in the Minecraft environment. This is done manually, and the booth can be built wherever you want to locate it. Using any block type, build your photo booth. It can be any shape you like, but it should have at least one block width of free space inside so that the player can enter.

Once you have created your photo booth, you need to be able to move your player inside and onto the trigger block. This is the block that the player stands on to run the function that you wrote in the first step, which will then trigger the camera. In the Minecraft environment, your position is given in reference to the x, y, and z axes. Look at the top-right of the window and you'll see the x, y, and z co-ordinates of your player – for example, 10.5, 9.0, -44.3. Assuming you are still in the photo booth, then these are also the x, y, and z co-ordinates of the trigger block in your booth.

Walk into your photo booth

Note down all three co-ordinates of your camera trigger block. When you're playing Minecraft, your program will need to verify that you are inside the photo booth. If you are, then it will trigger the `take_the_pic` function and take a picture with the

Construct your photo booth however you wish; just make sure the code knows where it lives

Steve is your 'shutter' in the Minecraft world: move him to the booth to take a photo

Place the booth anywhere in your world. Give it a special room in your house, or use it as a trap to see if someone is in your world

camera. To do this, Minecraft needs to know where you are in the world.

To find your position, you use the code `x, y, z = mc.player.getPos()`. This saves the x, y, and z position of your player into the variables `x`, `y`, and `z`. You can then use `print(x)` to print the x value, or `print(x, y, z)` to see them all if you wish, by adding it to the code. Now you know the position of the player, you can test to see if they're in the photo booth.

At this point we have a photo booth, the co-ordinates of the trigger block, and code to control the camera and take a picture. The next part of the code is to test whether the program knows when you're in the photo booth. To do this, we create a loop which checks if your player's co-ordinates match the trigger block co-ordinates. If they do, then you're standing in the photo booth. For this, we use a simple `if` statement, which is known as a conditional.

Change the `if` line in the code to ensure the co-ordinates you enter are those of your photo booth. Save and run your code to test it: walk into your photo booth and you should see the message 'You are in the photo booth!' in the Minecraft window.

You will note that the `if` statement checks if the x value is greater than or equal to 10.5: this is to ensure that it picks up the block, as it could have a value of 10.6. Remember to replace the x, y, and z values with those from your photo booth. After the message is printed, the same preview and camera snap will happen as before the `while` loop. The loop then resets itself so you can enter it again and take another photo! M

listing1.py

> Language: **Python 3**

```python
001.  from mcpi.minecraft import Minecraft
002.  from picamera import PiCamera
003.  from time import sleep
004.
005.  mc = Minecraft.create()
006.  camera = PiCamera()
007.
008.  mc.postToChat("Find the photo booth")
009.
010.  camera.start_preview()
011.  sleep(2)
012.  camera.capture('/home/pi/selfie.jpg')
013.  camera.stop_preview()
014.
015.  while True:
016.      x, y, z = mc.player.getPos()
017.
018.      sleep(3)
019.
020.      if x >= 10.5 and y == 9.0 and z == -44.3:
021.          mc.postToChat("You are in the photo booth!")
022.          sleep(1)
023.          mc.postToChat("Smile!")
024.          sleep(1)
025.          camera.start_preview()
026.          sleep(2)
027.          camera.capture('/home/pi/selfie.jpg')
028.          camera.stop_preview()
029.
030.      sleep(3)
```

Part 03

Build the ultimate
home music system

You've built your music system and enhanced it with
streaming services, now let's fill your home with sound

MAKER

PJ Evans

PJ is a writer,
software engineer,
and now has a
house with too
many speakers.

@mrpjevans

In the previous two tutorials, we've built
and configured an all-singing, all-dancing,
tricked-out music system using a Raspberry
Pi computer and Mopidy, the Python-based
extendable music server. We can listen to our
music in beautiful FLAC quality and stream from
the worlds of Spotify, SoundCloud, and YouTube.
There's one amazing feature that systems such as
Sonos offer that we haven't yet covered: multi-
room audio. Not just playing music in different
rooms, but also playing in perfect sync. In this final
part, we'll do just that and learn a few new Mopidy
tricks along the way.

01 **Going multi-room**

Sure, playback from our Mopidy setup is
great, but you don't want to be carrying that setup
around the house. With Snapcast we can play music
anywhere in perfect sync, so you can wander around
your home without interruption. This clever piece of
open-source software sends out audio in 'frames',
each one with a time code attached. Any device
that's part of the stream matches the frame's time
code to its own internal clock to ensure playback
happens at the same time, providing in-sync audio.
The only downside is a short delay in starting
playback as everything syncs up.

You'll Need

> Mopidy Music
System
magpi.cc/96

> Hi-Berry DAC+Zero
**magpi.cc/
hifiberrydaczero**

> Case (optional)
**magpi.cc/
hifiberrycase**

Hiding inside are a
Raspberry Pi Zero W and
HiFiBerry DAC+ Zero digital-
to-analogue converter

Connect this box to an
amplifier or active speakers for
synchronous whole-house audio

02 Snap to it

Before we can add Snapcast clients, our original Mopidy needs to become one itself, so it can keep in sync with everything else. We start by installing the Snapcast client and server on the same machine (it's effectively streaming to itself). Enter the following on the command line to download the client:

```
get https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
  sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

If you get an error here, run this to fix it:

```
sudo apt -f install
```

This automated install sets everything up and will restart the service on reboot.

03 Snappy service

We have the client running on your Mopidy system, but nothing to serve music. So, now install the server:
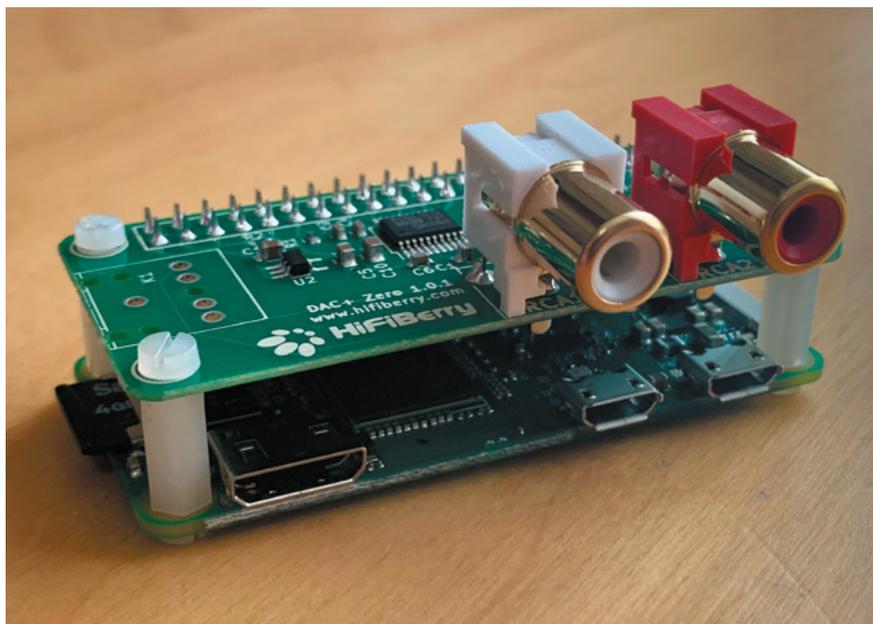
```
get https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
  sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

This will also restart on boot. The client will automatically find the server as its local. Now tell Mopidy to send its audio stream to the Snapcast server instead of the DAC.

```
sudo nano /etc/mopidy/mopidy.conf
```


▲ The addition of this purpose-built case not only protects your project, but also adds a bit of class

▲ The HiFiBerry DAC is an entry-level digital-to-analogue converter that provides much higher quality than the standard audio-out from a Raspberry Pi

Change…

```
[audio]
output = autoaudiosink
```

…to (the second part on one line):

```
[audio]
output = audioresample ! audioconvert !
audio/x-raw,rate=48000,channels=2,format=S16
LE ! wavenc ! filesink location=/tmp/snapfifo
```

Now restart Mopidy:

```
sudo systemctl restart mopidy
```

04 Set me up

Iris comes with full control over the Snapcast system. After restarting Mopidy, go to the Iris interface and into Settings. You'll see a Snapcast icon. Click on it and then click 'Enable'. You should see a 'Connected' message appear. There will also be a group which represents your local Mopidy setup. Click the group to rename it to something memorable. Before proceeding further, make sure that playback still works. It'll take a second or two longer to start as Snapcast syncs up, but should otherwise be unaffected. If it works well, your local Mopidy setup is complete.

Top Tip 👍

Choose your DAC

There are a wide range of digital audio convertors out there catering for every budget, from £12 to thousands – choose wisely.
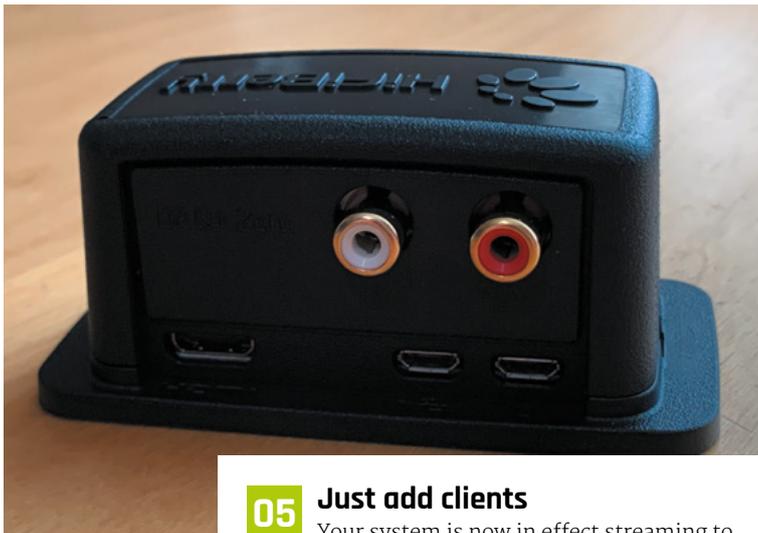
▲ The case includes access to the HDMI and USB ports, so you can connect a touchscreen monitor and use it as a Mopidy control centre

▶ You can use Iris on any web device to control which group is listening to which stream

05 Just add clients

Your system is now in effect streaming to itself, which means it can play in sync with other devices, so let's add one. We're using the HiFiBerry DAC+Zero, a great DAC for a small price. Start by connecting the DAC to a Raspberry Pi Zero W, ideally using standoffs to ensure a secure fit. Install Raspberry Pi OS Lite on this device as we're going to be running it headless. Using raspi-config, make sure you've configured WiFi and set a suitable hostname. Now update everything to the latest version using:

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

06 Make some noise

To enable the DAC+ Zero, obtain command-line access to your Raspberry Pi Zero W (using SSH or connect up a monitor and keyboard) and edit the main configuration file:

```
sudo nano /boot/config.txt
```

Near the end of the file, find the line reading:

```
dtparam=audio=on
```

Delete it (or comment out with a #) and add:

```
dtoverlay=hifiberry-dac
```

Save (**CTRL+X**) and reboot your computer. Once back up, connect the phono sockets on the DAC to an amplifier and test the output:

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

If everything is working well, a nice person will say "front centre".

Warning! Copyright

Where you live may or may not allow the copying of data from a CD. Check here for more info:

**magpi.cc/ripping**

07 Join the band

To get streaming working, we now repeat the process for installing the Snapcast client. Start by downloading and installing the client:

```
wget https://github.com/badaix/
snapcast/releases/download/v0.20.0/
snapclient_0.20.0-1_armhf.deb
  sudo dpkg -i snapclient_0.20.0-1_armhf.deb
```

If you see errors at the end of the process with the line 'dependency problems – leaving unconfigured', run the following command:

```
sudo apt -f install
```

This will detect and install all the dependencies required and then finish configuration. Snapcast will be configured to automatically start on boot.
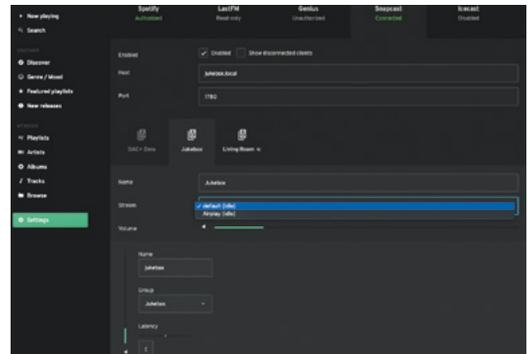
08 Configure and test

The Snapcast client is now running, but we need to tell it where the server is. Edit the configuration file:

```
sudo nano /etc/default/snapclient
```

Find the line that reads `SNAPCLIENT_OPTS=""` and add your Mopidy server hostname as follows:

```
SNAPCLIENT_OPTS="--host jukebox.local"
```

Replace 'jukebox.local' with whatever you named your server. Save and exit, then restart the client:

```
sudo systemctl restart snapclient
```

In a web browser, open up Iris on your main server and go to Settings, then click on Snapcast. You should see a new group (something like 'Group 8ec'); that's your device. You can rename it if you wish.

09 **Play, tweak, repeat**

Now try playing something from Mopidy with your new client hooked up to an amplifier or active speaker. Everything should be in sync. If not, you can adjust latency under your group settings to fine-tune the playback. At the bottom of the screen, you can click the speaker icon to control which devices are playing and set their individual volume levels.

You can add as many Snapcast clients as you like. It's a great use for an older Raspberry Pi and you don't have to use a DAC, you can just use the standard audio/video jack to an active speaker. We built another client using Pimoroni's Speaker pHAT and a battery for portable tunes.
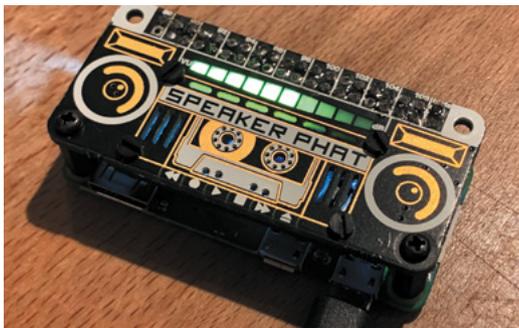
10 **Add some Apple**

Snapcast supports multiple streams, allowing clients to switch between them. We can add Apple AirPlay 2 support as a Snapcast stream that runs alongside Mopidy. We can't just install a package, though: we have to build and install Shairport Sync with the following commands:

```
cd
sudo apt install build-essential git
xmltoman autoconf automake libtool libpopt-
dev libconfig-dev libasound2-dev avahi-daemon
libavahi-client-dev libssl-dev libsoxr-dev

git clone https://github.com/mikebrady/
shairport-sync.git

autoreconf -fi

./configure --with-stdout --with-avahi
--with-ssl=openssl --with-metadata

make

sudo make install
```
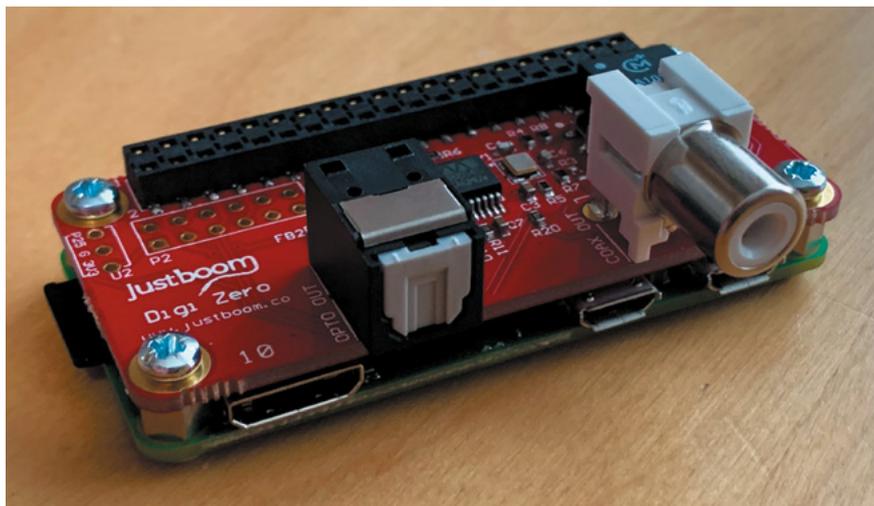
11 **Multistream!**

The final step is to configure Snapcast to enable Shairport Sync as a stream. Open up the Snapcast configuration file as follows:

```
sudo nano /etc/snapserver.conf
```

In the section starting `[stream]` you'll see an entry starting `[stream] =`. Directly under there, add the following line:

```
stream = airplay:///shairport-sync?name=
Airplay&devicename=Jukebox
```

(You can change the names to anything you like.) Now restart everything:

```
sudo systemctl restart snapserver.service
sudo systemctl restart mopidy.service
```

In Iris, go to the Snapcast settings and change the default stream to AirPlay. You can now push audio from iOS devices and Macs to your music system or any other Snapcast client.

12 **Going further**

If you've played along and built this setup, you now have a pretty sweet audio player setup. The great thing about this project is the wider range of budgets and platforms for which it caters. You can reuse older hardware, upcycle speakers, and turn just about anything with a processor into a streaming client. You can also get HATs with pure digital S/PDIF or coaxial output so you can use an amplifier's DAC if you prefer. How about using a USB audio capture device to stream audio around the house from your record player? As ever, it's over to you. **M**

▲ If your home audio setup features coax or S/PDIF digital interfaces, this JustBoom adapter is perfect and can be set up identically to the DAC+ Zero

◀ You can have a lot of fun with Snapcast. We've used a Pimoroni Speaker pHAT here for the ultimate in portable tunes

Top Tip 👍

DAC's not for me

If you're on a tight budget, you don't need a DAC at all. The 3.5 mm AV jack found on most Raspberry Pi computers will suffice for smaller projects.

Set up a LAMP server
with Mythic Beasts

Get your own Raspberry Pi 4 web server and learn to set up a LAMP server to host your own WordPress website

MAKER

Lucy Hattersley

Lucy is the editor of *The MagPi* and enjoys working in Raspberry Pi OS and building websites. This tutorial means she can do both at once.

magpi.cc

In this tutorial, we will look at setting up a web server to deliver our website and web services. You can experiment with this software bundle at home, but we're going to work in a live Mythic Beasts environment.

Mythic Beasts is an ISP (internet service provider) that uses Raspberry Pi to create low-cost, but powerful, Linux servers. The firm recently improved speeds using Raspberry Pi 4 computers. Raspberry Pi 4 is a powerful computer in the world of web servers, with its quad-core processor and 4GB RAM.

One big benefit for us is that the Raspberry Pi OS environment is pre-installed. This makes it easy for us to figure out how to set up a good stack of software.

In this tutorial, we're going to provision a Raspberry Pi 4 and install a LAMP (Linux, Apache, MySQL, and PHP/Perl/Python) software suite. LAMP (**magpi.cc/lamp**) is a common software bundle that provides a good basis for all kinds of online activity.

In this test, we'll use it to set up a WordPress website similar to the one we use online. But you could also use LAMP to deliver any kind of website.

You'll Need

> Mythic Beasts account

> Provisioned Raspberry Pi 4 server

> Computer to remotely access the server (Raspberry Pi 4)

01 **Sign up with Mythic Beasts**
Start by ordering a Raspberry Pi 4 from Mythic Beasts.Open your web browser and visit **magpi.cc/mythicbeastsorder**.

We're going to use a faster Raspberry Pi 4, to get the extra 4GB RAM for our LAMP server. There's a monthly charge of £7.25 for a Raspberry Pi 4 web server, which is an excellent deal for a dedicated computer with network storage (our look at non-Raspberry Pi services showed around £27 per month for a Linux machine with four cores).

You can test out the service with a Raspberry Pi 3 for only £5.25 per month (this only has 1GB RAM).

We're going to stick with the default 10GB storage, although the 'Disk space' slider at the bottom is used to select up to 250GB (at 2p per GB).

Click the green Order Now button.

Enter a service name and tick the 'I agree to the terms and conditions' box. The other fields are optional. Click Confirm to provision (activate) your Raspberry Pi. Enter your payment details or click Raise Invoice.

Wait for Raspberry Pi to be provisioned and the operating system to be installed. Don't close the web window.

> " LAMP provides a good basis for all kinds of online activity "
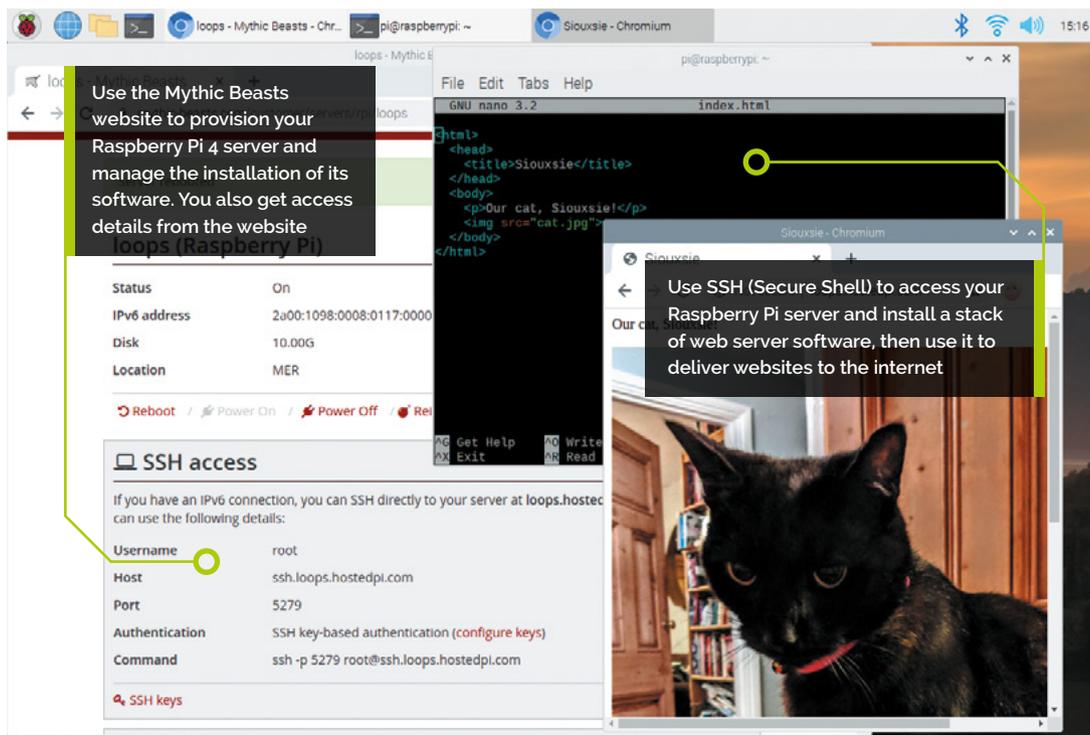
02 **Generate an SSH key**
When provisioning has completed, click 'Configure your server' (if you browse away, then you will find the server at **mythic-beasts.com/customer/servers/rpi**).

Mythic Beasts uses SSH keys to provide secure access between your local computer and the remote Raspberry Pi server. If SSH Keygen is new to you, then take a look at passwordless SSH access on the Raspberry Pi documentation (**magpi.cc/sshkeygen**).

First, check whether there are already keys on the local computer. Open a Terminal window and enter:

```
ls ~/.ssh
```

If you see files named **id_rsa.pub** or **id_dsa.pub** then you have keys set up already, so you can

Use the Mythic Beasts website to provision your Raspberry Pi 4 server and manage the installation of its software. You also get access details from the website

Use SSH (Secure Shell) to access your Raspberry Pi server and install a stack of web server software, then use it to deliver websites to the internet

skip the 'Generate new SSH keys' step below and head to Step 3. To generate new SSH keys, enter the following:

```
ssh-keygen
```

Upon entering this command, you will be asked where to save the key. We suggest saving it in the default location (**~/.ssh/id_rsa**) by pressing **ENTER**.

You will also be asked to enter a passphrase, which is optional. The passphrase is used to encrypt the private SSH key so that if someone else copied the key, they could not impersonate you to gain access. If you choose to use a passphrase, type it here and press **ENTER**, then type it again when prompted. Leave the field empty for no passphrase.

03 Add the key SSH

Take a look a the content of the key:

```
cat ~/.ssh/id_rsa.pub
```

The output will start with 'ssh-rsa' and end with your hostname 'pi@raspberrypi'.

Use your mouse to select all the output of the SSH key, then right-click and choose Copy. Now head back to the browser and click 'configure keys'. Right-click on the large 'Keys' text field and choose Paste to enter the SSH key. Click 'Save changes'.
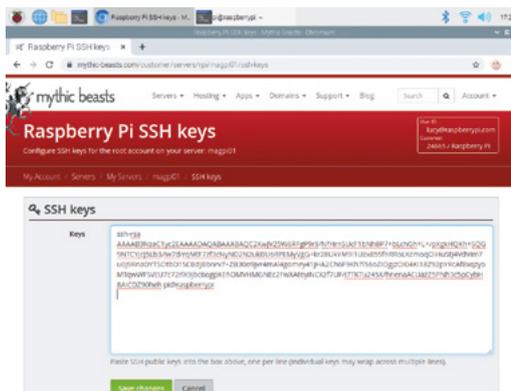
▲ A secure SSH key generated on your local computer is shared with the remote computer to provide access

04 SSH access

Scroll down the server window and take a look at the details in SSH access. Here you will see your Username, Host, Port, and Authentication information. You will use this information to connect to your Raspberry Pi server. There is also a Command section, with the Terminal command used to connect directly. Ours looks like this:

```
ssh -p 5274 root@ssh.magpi01.hostedpi.com
```

Copy and paste the command into the Terminal and press **ENTER**.

Respond 'yes' at the 'continue connecting' prompt and your SSH key will be added to your list of 'known hosts'.

Top Tip 👍

Network unreachable?

If you get 'Network Unreachable' when trying to connect via SSH, click on Reboot in the 'Manage your Raspberry Pi' webpage to restart your Raspberry Pi server.

If you created one, you'll need to add the SSH key password when prompted. When logged in, the command prompt will change to root@raspberrypi. You are logged in as 'root' and in the root user's home directory '/root'.

05 Install a web server

We're going to use the Apache web server (**magpi.cc/apache**), which you can install with the following commands:

```
apt update
apt upgrade -y
apt install apache2 -y
```

Note that you don't need to use 'sudo' as you are the root user.

Open a web browser and visit the following URL (replacing '**magpi01**' with the name of your own hosted server: **magpi01.hostedpi.com**.)

This will display the Apache2 Debian Default webpage with an 'it works!' message.

This page is an HTML file located on your remote Raspberry Pi, at **/var/www/html/index.html**.

Navigate to this directory in the Terminal and have a look at what's inside:

```
cd /var/www/html
ls -al

cat index.html
```
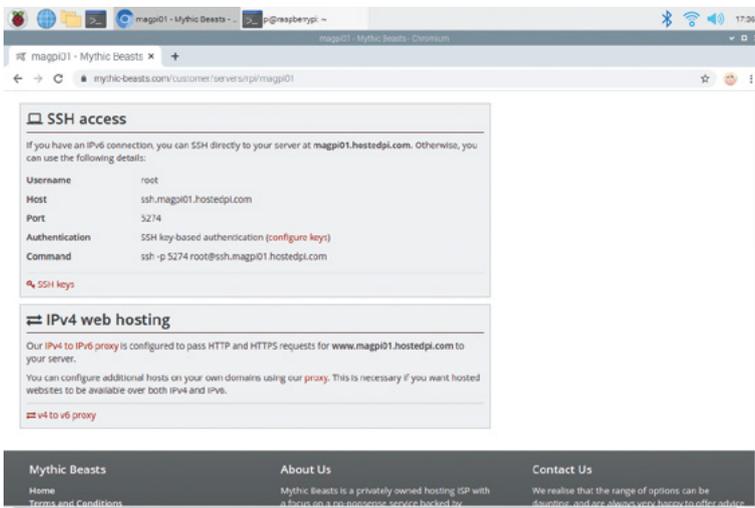

▲ Once your Raspberry Pi 4 server is provisioned, you'll be able to use SSH to access it

06 Install PHP

We now have the 'L' and 'A' of our LAMP server: Linux and Apache. The M and P come next: MySQL and PHP.

MySQL is a database system, while PHP is a programming language. You'll need both to run most content management systems, such as WordPress.

Now is the ideal time to add the software. You can install PHP with the following command:

```
apt install php -y
```

And MySQL with this command:

```
apt install mariadb-server php-mysql -y
```

Now restart your Apache server to ensure both services are running.

```
service apache2 restart
```

❝ Send files directly from your local computer using secure copy (scp) ❞

07 Upload content remotely

You don't need to be logged in to your Raspberry Pi server to edit content in the **html** directory. Close the connection with:
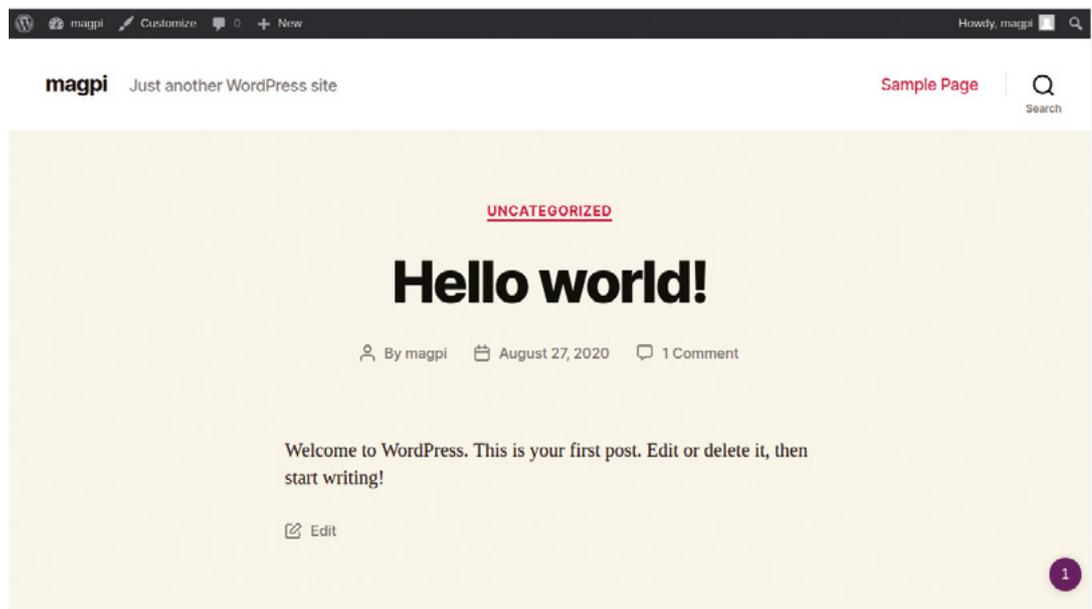
```
exit
```

You can send files directly to the **html** directory from your local computer using secure copy (scp).

First, we're going to get a photo (of our cat) and name it **cat.jpg**. Then create an **index.html** file.

```
nano index.html
```

And enter this basic HTML code:

```
<html>
  <head>
    <title>Siouxsie</title>
  </head>
  <body>
    <p>Our cat, Siouxsie!</p>
    <img src="cat.jpg">
  </body>
</html>
```

A LAMP (Linux, Apache, MySQL, and PHP) server can be used to run a WordPress-based website

Copy the two files directly to the **html** directory on your remote Raspberry Pi server:

```
scp -P 5274index.html cat.jpg root@ssh.
magpi01.hostedpi.com:/var/www/html/
```

…making sure to replace the '5274' and 'magpi01' parts with the port and hostname of your Raspberry Pi server. Press **F5** to refresh your web browser and view the new webpage design.

08 Test out PHP

Let's test that PHP is working, and also take a look at the index page for our website. Create the file **index.php**:

```
cd /var/www/html
nano index.php
```

Put some PHP content in it:

```
<?php echo "Hello, World!"; ?>
```

Save the file and close with **CTRL+O** and **CTRL+X**.
Now get rid of the **index.html** file (because it takes precedence over the **index.php** file):

```
rm index.html
```

Reload your website in the web browser and you will see 'Hello, World!'. This page isn't dynamic, but it's created with PHP code.

09 Install WordPress

Now we're going to head back into our remote server and do something a little more detailed. We're going to set up WordPress, a popular CMS (content management system). This makes use of our PHP and MySQL database and is a great option if you're looking for a powerful website with a minimum of coding.

Log back into your remote Raspberry Pi server.

```
ssh -p 5274 root@ssh.magpi01.hostedpi.com
```

Go to your **html** folder and get rid of all the content.

```
cd /var/www/html/
rm *
```

Now download the latest version of WordPress:

```
wget http://wordpress.org/latest.tar.gz
```

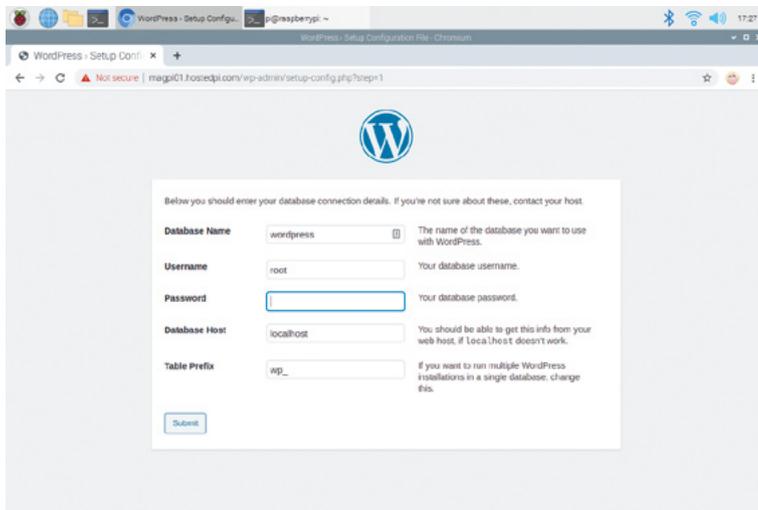If you need to install wget, use `apt install wget`.
Next, extract the WordPress tarball to get at the WordPress files:

```
tar xzf latest.tar.gz
```

Move the contents of the extracted **wordpress** directory to the current directory.

```
mv wordpress/* .
```

Tidy up by removing the tarball and the now-empty **wordpress** directory:

▲ The WordPress login page running on a Raspberry Pi web server using PHP and HTML

```
rm -rf wordpress latest.tar.gz
```

Use `ls` to view the contents of a WordPress directory inside your **html** directory. It will include a **new index.php** file along with various HTML and PHP files.

---

10 **Set up your WordPress database**
Although you have the WordPress files, you can't use your website just yet. First, you need to set up the MySQL database. Enter this command:

```
mysql_secure_installation
```

As this is the first time running MariaDB, there is no password, so just press **ENTER**.

Type in **Y** and press **ENTER** to 'Set root password?'. Type in a password and press **ENTER**. Make sure you note this root password, as you will need it shortly.

You'll be asked four questions. Answer 'y' to each one: Remove anonymous users, Disallow root login remotely, Remove test database and access to it, and Reload privilege tables now.

When complete, you will see the message 'All done!' and 'Thanks for using MariaDB!'.

Top Tip 👍

Learn SQL

If you want to learn SQL and database design, take a look at some of the resources on *The MagPi* website (**magpi.cc/learnsql**).

---

11 **Create a database**
Now that the database is installed, you need to create a database for WordPress:

Run mysql in the Terminal window:

```
mysql -u root -p
```

Enter the root password you created. You should start up the MariaDB monitor and see this command prompt:

```
MariaDB [(none)]>
```

Enter this command:

```
create database wordpress;
```

If this has been successful, you should see this:

```
Query OK, 1 row affected (0.03 sec)
```

Now enter these two commands:

```
  GRANT ALL PRIVILEGES ON wordpress.*
TO 'root'@'localhost' IDENTIFIED BY
'YOURPASSWORD';
  FLUSH PRIVILEGES;
```

Exit the MariaDB monitor using **CTRL+D**.

---

12 **Open WordPress**
Open the web browser on Raspberry Pi and head to your website: **magpi01.hostedpi.com**.

You should see a WordPress setup page; click on Let's Go.

You will see 'unable to write to wp-config.php file'. Select all the code inside the window and right-click to copy. Now switch to the Terminal window on your remote Raspberry Pi and enter:

```
nano wp-config.php
```

Paste the code into the **wp-config.php** file in Nano, then save and exit (**CTRL+O, CTRL+X**).

Switch back to your web browser, and click 'Run the installation'.

Fill out the information fields and click the Install WordPress button, then log in using the account you just created. **M**

Run your Raspberry Pi code automatically

Turning a script into a service

Ben Everard

🐦 @ben_everard

Ben is a former sysadmin who's spent longer than he cares to admit fiddling with running services on Linux boxes.

So, you've written some code for your Raspberry Pi that runs your project. You need to run it every time your system starts, but how do you do this? Perhaps the easiest method is adding a line to the **/etc/rc.local** file. This is a Bash script that runs each time the computer boots up. It's quick and easy, but there are a few drawbacks to it, namely:

- If your script outputs any information or errors, where do these errors go?
- If your script crashes, how do you know, and how do you restart it?
- Can you stop or reload your script easily?

Fortunately, the Linux operating system used by Raspberry Pi computers has a mechanism for managing things in the background: systemd. This is a service layer that's widely used to manage all sorts of bits of essential software on your Raspberry Pi or other Linux computer, from databases to window managers. It provides a way of controlling software that runs in the background. Let's take a look at what this means in practice.

Above ◈
You can easily access the status and output of your script from the command line, either via a screen or a remote session (such as SSH)

Systemd can handle a huge range of things, but we'll be looking at 'services', which is the name used for software that runs in the background because it typically provides a service.

Our really simple script will simply use Python to output the phrase 'hello world' every second. This is, admittedly not the most useful script, but it'll show the technique of running our code automatically and capturing the output.

This can be done with the following code:

```
#!/usr/bin/python

import time
import sys

while True:
        print "hello world"
        sys.stdout.flush()
        time.sleep(1)
```

There are a couple of unusual bits in this. The first is the first line – this is known as a shebang, and it's used to tell the operating system what program to use to run the script. This is the complete path to the binary for Python. We got this on Raspberry Pi OS. It might be a little different on other Linux distros. If you're unsure, you can run the command **which python** and it will tell you.

The second unusual thing is the call to **sys.stdout.flush()**. This isn't strictly necessary, and it just tells the operating system to make sure all the output has made its way through the output buffers and on to its destination. We found we had a large lag between the script running and output making its way to the logs if we didn't use it. In practice, you may find that you can live without this in your programs (or just include it once in a main loop).

We saved this in a file called **hello-service.py** in **/home/pi**.

Finally, we need to make this file executable by running the command:

```
chmod a+x /home/pi/hello-service.py
```

RUN THE SCRIPT

That's our script ready. Let's now get it running as a service. In order to let systemd know what we want to do with our simple service, we need to create a unit file. There's a lot that can go into a unit file, but for basic usage, they can be quite simple. Ours will be as follows:

```
[Unit]
Description=A service to say hello world
After=systemd-user-sessions.service

[Service]
Type=simple
ExecStart=/home/pi/hello-service.py
```

This tells us a bit about our script. The `After` section tells systemd when we want our script to start – in this case, after the `systemd-user-sessions` (which is one of the system services that starts every boot).

> There's no fussing around trying to **find PIDs of processes in order to stop or restart them**

`Type=simple` (as opposed to `forking`) tells systemd that the command will continue to run in the session it was started.

The final line tells systemd what command to run.

Save this unit file as **hello.service** in the **home/pi/** directory, then copy it to the **systemd** directory with:

```
sudo cp /home/pi/hello.service /etc/systemd/system
```

That's everything set up and ready to go. We just need to let systemd know what we want to do.

You can start your service with:

```
systemctl start hello.service
```

…see its status with:

```
systemctl status hello.service
```

…and stop it with:

```
systemctl stop hello.service
```

It's this management of the running code that makes this method of running code much easier. There's no fussing around trying to find PIDs of processes in order to stop or restart them. What's more, once you start a service, systemd will monitor it, and if it crashes for some reason, it will attempt to restart it. Obviously, this isn't a reason to create code that's unstable, but it is an extra line of defence if your code has to run all day by itself.

If you want to see the output of the service, you can use `journalctl`:

```
journalctl -u hello -e
```

What we've done so far will get the service up and running, but it won't automatically start it every time you boot the system. For that, you need to enable the service with:

```
systemctl enable hello.service
```

With this done, you can restart your machine and it will automatically start. If you want to stop it starting automatically, you can disable it with:

```
systemctl disable hello.service
```

There's far more to systemd than we've looked at here, but with these basics, you can get your code up and running and make sure it's looked after properly as it whirs away in the background. □

Above ◈
Systemd service files are straightforward and easy ways of encapsulating the information needed to run your script

Left ◣
Flushing the output means you always have the latest information

HackSpace

This tutorial is from HackSpace magazine. Each issue includes a huge variety of maker projects inside and outside of the sphere of Raspberry Pi, and also has amazing tutorials. Find out more at **hsmag.cc**

Tap-an-LED:
the software

Detect touches, create animations, and compose musical sequences with your Tap-an-LED hardware and some Python code

Part 02

Mike Cook

MAKER

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**

You'll Need

> The hardware from *The MagPi* #97 **magpi.cc/97**

Tap-an-LED brings you a touch interface on top of a 16×8 multicoloured LED matrix, for emulating a Monome, generating new user interfaces, and playing games. Last month we showed you how to build the hardware; this month we see how to match a tap to an LED in software and some of the things you can do with it.

01 **A touch of class**

The hardware we built last month will light up the LED matrix, and also register a touch, or tap, on the screen. Now we need to bring the two things together so that we can detect a tap and alter the LED in the matrix that corresponds to the tap position. We are going to do this by using two Python classes: Caltap and Stmpe610. The Caltap (calibrated tap) class is the one that the user will see, and interact with, whereas the Stmpe610 class only gets called by Caltap. This structure is shown in **Figure 1**.
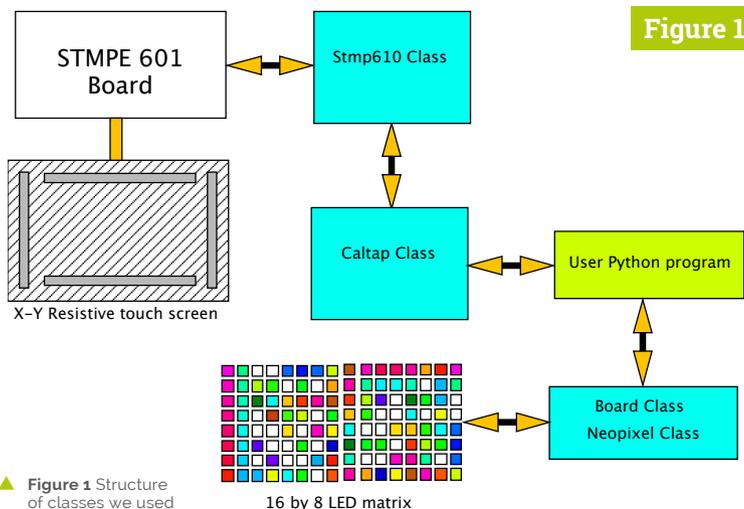
Figure 2

getPos() returns a tuple

(X position, Y position, LED number, valid reading)

0 to 15   0 to 7   0 to 127   True or False

getRaw() returns a tuple

(X position, Y position, touch pressure, valid reading)

0 to X of corner2 – X of corner1   0 to 255   True or False
From A/D reading 0 to 4095

0 to Y of corner2 – Y of corner1
0 at bottom left corner
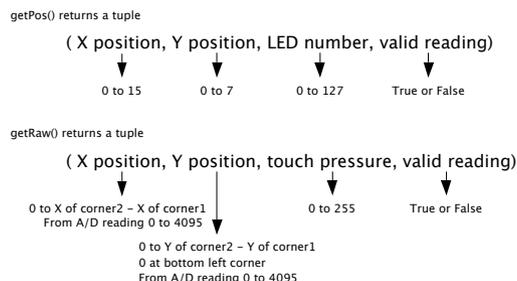From A/D reading 0 to 4095

▲ **Figure 2** Returned values from Caltap class methods
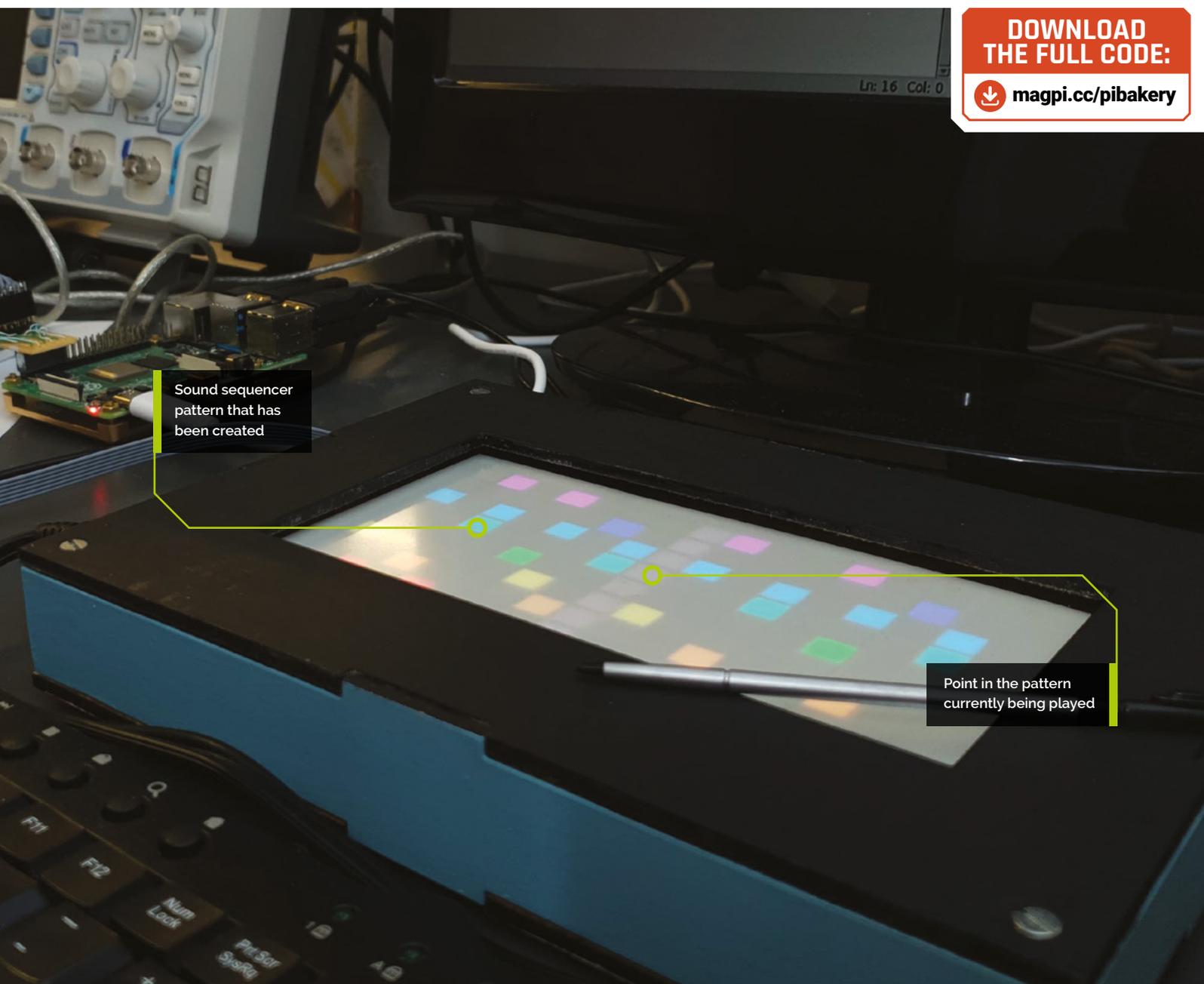
02 **The Stmpe610 class**

The touch controller chip, the STMPE610, is complex and requires a lot of setting up. After studying the code that Adafruit has produced, we thought the chip needed a bit of a different setup, so we wrote the Stmpe610 class. One big problem we came across is that the chip would register a touch, but then, if the touch was quick or light, it would not be able to take a reading. This meant either returning the wrong co-ordinates for the tap or hanging in a loop waiting for another tap. Neither of which is very satisfactory.

03 **Solving the problem**

We did try using the capability of the board to signal an interrupt and use that to get a reading as soon as a tap was detected – not when the user program asked to see if a touch had been detected. But unfortunately, the problem still remained. In the end, we opted for the Caltap class returning not only the data, but also a flag that showed if the data was valid. This meant that the final application had to make the decision of what to do when a touch was detected but no measurement was possible.

Figure 1

STMPE 601 Board

Stmp610 Class

X–Y Resistive touch screen

Caltap Class

User Python program

Board Class
Neopixel Class

▲ **Figure 1** Structure of classes we used

16 by 8 LED matrix

Sound sequencer pattern that has been created

Point in the pattern currently being played

04 Timeout

We did this by first clearing the FIFO (first in first out) buffer and then waiting until a new measurement had been made. But in order not to hang, waiting for a reading that might never come until the screen was touched again, we implemented a timeout. Normally, a reading will be produced before the timeout occurs, but in case it didn't, a timeout set to 40 ms caused an invalid flag to be returned (**Figure 2**). Both the **caltap.py** and **stmpe610.py** files, along with all the examples discussed here, can be sourced from our GitHub page (**magpi.cc/pibakery**).

05 Calibrating

The **caltap.py** code must have its own calibration data from your hardware, not ours. So you need another program to directly call the Stmpe610 class and get five calibration points needed; this is done with the **led_cal.py** program. When this is run, it will first ask you to tap on the furthest corners of the touchscreen you can access; a red LED arrow points to where (**Figure 3**, overleaf). Then you have to click on the centre of three lit pixels. Finally, copy five lines from the Python console and paste them into the **caltap.py** file, replacing those from `corner1...` to `led120...`.

Top Tip 👍

Generating colour sets

We used palettes generated with our project from *The MagPi* #94 (**magpi.cc/94**).

Figure 3

06 Let the tapping begin

Now we are ready to put the system to the test. Note that whatever program you use, the two class files must be in the same directory. Start off by loading in the **tap-a-sketch.py** file; as the name implies, tapping on an unlit LED position will light it, and tapping on a lit LED will turn it off. At each tap, the colour of the LED to light up is incremented through a palette of colours defined in the program (**Figure 4**). You can customise these by adding, removing, or changing colours in the code.

Top Tip 👍

Seeing unlit pixel positions

We used a very dim white colour for the background to make the position of unlit pixels stand out.

07 A small quirk

Now, there is a small quirk in the Adafruit NeoPixel Python library. While it is easy to set the colour of an LED with:

```
pixels[number] = (128, 128, 0)
```

…and get the colour of a specific pixel with:

```
colour = pixels[number]
```

…the catch is that the colour is set using a tuple, but is returned as a list. So a line like:

```
if pixels[number] == (0, 0, 0):
```

…will never be True. Instead, you must use:

```
if  pixels[number] == [0, 0, 0]:
```

Which means if you are looking for a specific colour, you can't take it directly from a palette entry.

08 Animation

The **tap-an-animation.py** program is a Pygame extension of this basic idea. The patterns tapped out on the LEDs will also appear on the screen, as seen in **Figure 5**. You can tap out a number of frames' worth of patterns and animate them in a loop. The 'Next frame' box automatically adds a new frame to the sequence if you have reached the current limit. Each new frame starts off with the pattern set to the previous one. You can shift the whole frame in

any direction to make animations easy. While playing the loop, the + and – keys will alter the speed.

" Each note can be set to a different channel "

09 Saving the animation

You can save any animation in one of two formats, LED (.led) or JPG. The LED format is our own custom format which simply consists of a single file containing all the frames preceded by a frame count, useful for returning to previous work. The JPG format saves each individual frame as a separate JPG image of your computer's display, in a single directory with each one post-fixed with a frame number. You can then use a utility like ImageMagick to convert them into an animated GIF for use on social media or other websites.

10 Sound sequencer

We are always interested in music applications here at the Bakery, so we have produced two sound sequencers. The first, **sequencer1.py**, is a self-contained program playing back prerecorded sound samples. It starts by asking you to type in a speed in BPM (beats per minute), then tap a square to give you a note. The sound samples are in a separate directory called **Marimba**, and playback is through the normal audio system. This makes use of the 'mixer' component of Pygame; we were surprised to learn that you could use this without having to start up the video component of Pygame.

11 Sound sequencer using MIDI

The **sequencer_midi.py** file is a full Pygame-driven sequencer. You need to have a MIDI interface lead plugged into a USB port and the other end connected to a sound-generating module. The program starts off with all sounds set to piano, and the notes in the key of C major. Each row on the LED matrix corresponds to one note and one colour; you can tap in any 16-step sequence you want. Once you hit the Run control, the sequence will be played and a faint white bar will sweep across the LEDs. You can alter the sequence as it plays, making for hours of fun.

Figure 4

▲ **Figure 4** The LEDs from the tap-a-sketch program

12 Sequencer controls

All value changes are made by clicking on the + or – icons next to the displayed value. Each note can be set to a different channel, and as each channel is assigned its own instrument sound, when you change the channel you also change the instrument shown in the voice column. The pitch of each note can be changed, as can the strike velocity. Each line of the sequencer can be individually muted by clicking on the rectangles under the Mute column. Finally, the tempo or BPM can be changed by the controls in the bottom-left corner (**Figure 6**).

13 Next month

Next month, in the third and final part, we will see how – with the addition of a small Arduino – we can connect MIDI devices direct to the USB socket of a laptop, to interact with DAWs (digital audio workstations) like Ableton Live and work up some LED visual displays of music. **M**

◄ **Figure 5** Screenshot of the tap-an-animation program

▼ **Figure 6** Screenshot of the sequencer_midi program

Figure 5

Figure 6
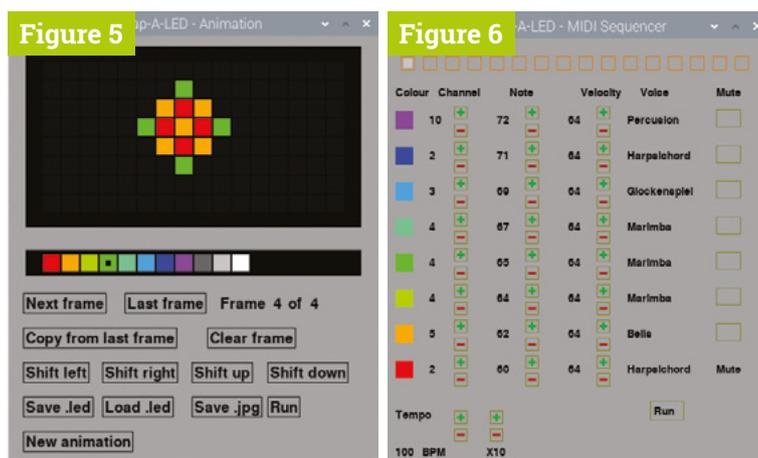
Haunted House Hacks

Spookify your home in time for Halloween with **Rob Zwetsloot** and these terror-ific projects!

Dressing up for Halloween is so last year. As in, that was the focus of our Halloween article in 2019's *The MagPi* #86 (magpi.cc/86). Check it out for cool costume ideas. This year we're looking at dressing up something else: your home.

Haunted houses are a mainstay attraction at amusement and theme parks around the world, and for a very good reason – they're fun! Sometimes they're cheesy, sometimes they're scary, but the best ones are a bit of both, we think.

From decorations to pranks, here are some great projects for fully transforming your home for the witching hour.

Frightful façades

Make your house horrifying with these cursed outdoor projects

RASPBERRY PI HALLOWEEN LIGHTS AND MUSIC SHOW

This simple idea of Raspberry Pi-controlled lights has been taken to an extreme by having it control an entire show in front of Noel's house. It's been adapted from a Christmas show he created a few years before this that didn't use a Raspberry Pi. It activates several lights and plays music when motion is detected at the front of the house – like a spooky floodlight!

It does connect to some relays, so only attempt this if you're good with electrics and know the right circuit breaker.

magpi.cc/hallolightsmusic

SCARE FACTOR 4/10

▲ As well as all the lighting and other effects, we like the general decorations that have gone up to add to the atmosphere

RASPBERRY PI HAUNTED HOUSE

This project is a bit of a mixture of indoors and outdoors, with a doorbell on the house activating a series of spooky effects like a creaking door, 'malfunctioning' porch lights, and finally a big old monster mash in the garage.

We talked to its creator Stewart Watkiss about it a few years ago and he revealed how he used a PiFace HAT to interface with home automation techniques to create the scary show, although it can be made much easier these days thanks to Energenie. Our favourite part, though, is still the *Home Alone*-esque monster party that caps it off.

magpi.cc/hauntedhouse

SCARE FACTOR 7/10

▲ Do you dare press this dastardly doorbell?

▲ It's a very nice-looking build as well

EYE OF SAURON

The dreaded dark lord Sauron from *Lord of the Rings* watched over Middle-earth in the form of a giant flaming eye atop his black tower, Barad-dûr. Mike Christian's version sits on top of a shed in Saratoga, CA.

It makes use of the Snake Eyes Bonnet from Adafruit, with some code modifications and projecting onto a bigger eye. Throw in some cool lights and copper wires and you get a nice little effect, much like that from the films.

magpi.cc/sauron

SCARE FACTOR 8/10

▲ Atop the shed with some extra light effects, it looks very scary

POPLAWSKI HOLIDAY LIGHTS

Would you like to control someone else's holiday decorations? The Poplawskis set up a live stream at Christmas and Halloween for people to control over the internet. Check out their Facebook page for info about when the new set goes up! **magpi.cc/poplawski**

Dreadful decorations

Fill your home with eccentric electronics
to celebrate the season

▲ A simple circuit, but a nice and spooky scare

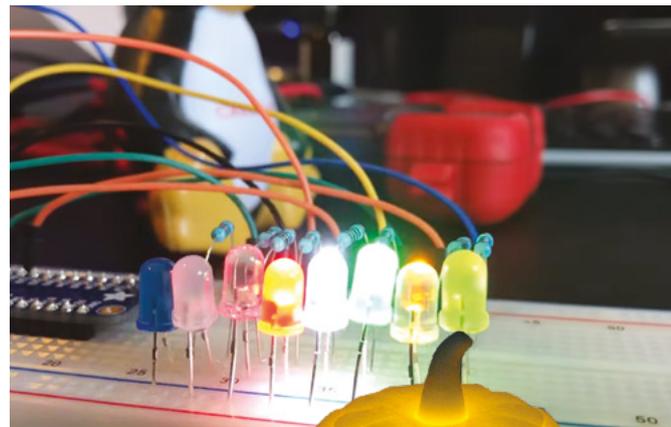▲ In the dead of night, there are just… Halloween-themed lights

SPOOKY HALLOWEEN MUSIC-LIGHT SHOW

This one is nice and easy – a series of LEDs, some music, and some syncing of the two. You can make the music as scary as you desire (maker's choice), and the lightshowpi software allows for easy syncing of the lights to the music as well.

Need some ideas of what to install it into? How about an indoor pumpkin, or a plastic skull? Maybe behind some books or DVD cases to create a spooky party for ants.

magpi.cc/halloshow

SCARE FACTOR **2/10**

▼ A simple-looking breadboard setup that can be extra spooky with the right placement

▲ Another hidden motion sensor; we're sensing a theme here

AUTOMATED HALLOWEEN DISPLAY

Much like some of the outdoor spooks, this indoor scare is a motion-activated light and sound show using battery-operated lights and devices! Creator Paul Smith decided to add some cool lightning effects to the show as well, with sounds and strobe lights.

It's all powered by Python and even has some optional bits and pieces for a rumble motor to simulate a 'monster'. What kind of monster, we're unsure of.

magpi.cc/autohallo

SCARE FACTOR **6/10**

▲ We love the eyes and scary sounds in this version that seem to follow you around

RASPBERRY PI-POWERED JACK-O'-LANTERN

A classic indoor Halloween decoration (and outdoor, according to American movies) is the humble Jack-o'-lantern. While you could carve your own for this kind of project (and we've seen many people do so), this version uses a pre-cut, 3D-printed pumpkin.

If you want to put one outside as well, we highly recommend you add some waterproofing or put it under a porch of some kind, especially if you live in the UK. If you don't fancy spooky eyes on a display, you can always do something different with LEDs, like the other project on this page.

magpi.cc/pilantern

▲ The original 3D print lit with a single source is still fairly scary

SCARE FACTOR **5/10**

Halloween upgrades

Turn these Raspberry Pi projects into something a bit more ghoulish

GPIO MUSIC BOX

This project helps you create what is basically a Raspberry Pi synth with buttons and such. It's a pretty nice and simple project that also allows you to add your own specific sounds. Like thunder, wolf howls, and cackling witches perhaps?

magpi.cc/gpiomusic

ULTRASONIC THEREMIN

If you don't know what a theremin is, you'll definitely know once you hear one. It's popularly used to make spooky and chilling sci-fi sound effects (think films from the 1950s and *Doctor Who* openings) – perfect for a scary space theme if you want one!

magpi.cc/theremin

SPINNING FLOWER WHEEL

Not just something you find in Sonic the Hedgehog, spinning flowers are a fun way to work with motors on Raspberry Pi. They can even be motion-activated if you like, and you can easily change the flowers to skulls or pumpkins or… bats? Do bats spin? They do now.

magpi.cc/spinningflower

WHOOPI CUSHION

This prank from Raspberry Pi is one of our favourites because it's a) a pretty ingenious build and b) a whoopee cushion. How could this be changed to be scary? Well, what if someone screamed when they sat down? Now you're thinking with poltergeists.

magpi.cc/whoopi

Automated apparitions

Create petrifying pranks that will surely scare any unwitting victims

▲ A monster trying to get out of the room is one of the many scares
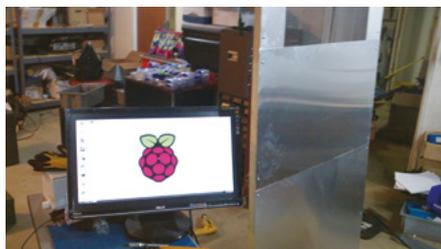
◀ Unfortunately, we'll have to ruin some of the magic to show the door frame

SCARY DOOR

You're unlikely to trick someone already in your house with a random door that has appeared out of nowhere, but while they're investigating they'll get the scare of their life. This door was created as a 'sequel' to a Scary Porch, and has a big monitor where a window might be in the door. There's also an array of air-pistons just behind the door to make it sound like someone is trying to get out.

There are various videos that can play on the door screen, and they're randomised so any viewers won't know what to expect. This one also uses relays, so be careful.

magpi.cc/scarydoor

SCARE FACTOR 10/10
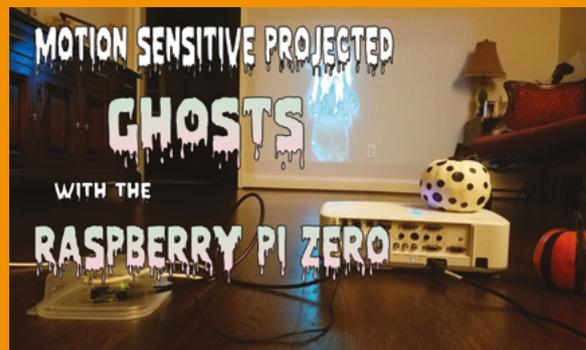
▲ It's a relatively simple setup – you just need a wall!

MOTION-SENSITIVE PROJECTED GHOSTS

We quite like this one, and it has multiple applications as well. In this case, though, ghostly pranks at Halloween are definitely a perfect use case. Much like the other projects we've talked about in this article, it relies on motion detection to display a random video of a ghostly presence. It makes use of stock footage, although if you're handy with a green screen and Adobe After Effects, you could easily make your own.

You could even use this outside your house if you have the means. Try not to scare everyone in your neighbourhood, though.

magpi.cc/motionghost

SCARE FACTOR 8/10

▲ Sometimes a friendly ghost will wander across the wall… sometimes they won't be so friendly

HALLOWEEN MAGIC MIRROR

The magic mirror is one of the most iconic cool projects that you can create: a great piece of IoT and making that relies of the strengths of Raspberry Pi. So, let's turn it into a scary prank as well.

This version acts as a regular mirror, but we've seen versions that add scary photos and video to the traditional smart mirror during Halloween time. Just make sure the mirror is installed in a place without many breakable ornaments.

magpi.cc/hallomirror

SCARE FACTOR 9/10

▲ Imagine walking past a mirror only for a spooky spook to pop out at you?

▲ The insides have been heavily modified to allow for the scare

HAUNTED JACK-IN-THE-BOX

This prank is pretty special. A Jack-in-the-box that you can set on a shelf somewhere that will wind itself and release a Jack 'very violently', according to creator Sean Hodgins. It uses OpenCV and a Raspberry Pi Camera Module to check that you're actually looking at, or have seen, the box. This then runs the script which does some random winding of the crank, with a spectacular jump scare finish.

magpi.cc/hauntedbox

▲ This is actually not too bad a tutorial on building a frame as well

POSSESSED PORTRAIT

This very modern take on a haunted portrait is a little too effective if you ask us. It's a similar technique to the haunted mirror, but this one has its own different effect. Studying a new painting in a house that seems… slightly off? Has it been printed? No, it's just a screen and suddenly the portrait is looking at you.

There have recently been updates to this project to allow for recording any of your victims, so you can well and truly embarrass them at a later date.

magpi.cc/portrait

▲ A very normal portrait. It's just weird to see one here!

SCARE FACTOR 8/10

SCARE FACTOR 9/10

◀ The pant-wettingly scary bit

THE *Official*

RASPBERRY PI
PROJECTS BOOK

VOLUME 5

**200
PAGES**
OF IDEAS

PiBoy **DMG**

SPECS

CONTROLS:
Ten buttons, eight-way D-pad, analogue joystick

POWER:
4500 mAh LiPo battery

DISPLAY:
3.5-inch LCD DPI 640×480

EXTRAS:
On/off button, HDMI out

Is this seemingly ultimate handheld worth the price? **Rob Zwetsloot** investigates

A Raspberry Pi retro gaming case based around a Game Boy design is not exactly uncommon, and you might have turned the page here and been a bit sceptical. We were too when it landed on our desk; however, we were pleasantly surprised because that's the only feature of the PiBoy DMG that it shares with other similar devices.

Even on paper it has some interesting features – sure, it has a ton of buttons and a battery etc. (read the specs box for all that), but it also has an active cooling fan, an analogue joystick, and even a brightness control wheel for the screen – something very reminiscent of the contrast control on the original Game Boy.

Usually, a lot of these kits can feel very cheap and rough, using standard 3D-printed parts for everything that can feel uncomfortable and flimsy and don't really have the nicest aesthetic. The PiBoy feels more like the real deal: the main case is sturdy, the buttons are nice to use, and even the analogue stick has a little click-down thing. Unfortunately, like a lot of Game Boy form factor builds, the 'shoulder' buttons on the rear are a bit fiddly. With six face buttons, though, you're probably set for playing any games up until the Mega Drive / SNES era.

Pocket emulation

Speaking of playing games, the software on the PiBoy is a slightly modified version of RetroPie,

▶ The PiBoy DMG takes its design cues from the classic Nintendo Game Boy

▲ If you really fancy connecting a portable console to Ethernet, you can do that!

▲ Different, simple adapters need to be used depending on which Raspberry Pi you install

with specific Experimental Pi splash screens and branding to the startup. Thanks to this, you're only really limited by your Raspberry Pi choice, with Raspberry Pi Zero, Raspberry Pi 3/3B+, and Raspberry Pi 4 supported.

Because of this, the kind of games you'd be running on RetroPie systems run as smoothly as you'd expect. The LCD screen outputs at a fairly reduced resolution anyway, which reduces some of the load. With the fan on the rear of the PiBoy, we didn't find it getting too hot with a Raspberry Pi 4 in it, although the whining of the fan is slightly unnerving for a handheld and sounds like a CD. You can play the PiBoy in any position you wish without scratching anything, thankfully.

❝ The LCD screen outputs at a fairly reduced resolution ❞

The various adapters and such for the PiBoy allow for all the output and input options of the installed Raspberry Pi to be accessible. As well as USB sticks which can be used for storage, and easy access to the microSD card, you can even plug in headphones and use a (regular size) HDMI cable to plug it into your TV. Use the available USB ports for some USB controllers and you have a very portable plug-and-play box.

◀ The battery case is the same as the original, albeit this one uses a much more powerful battery

Amazingly, it also has a special Steam Link function. You'll likely be connected to wireless LAN on the PiBoy and if you have a decent connection, it's amazing to play some games in your hands in your own home.

It's a pretty fantastic piece of kit, and we think it earns its price tag. Just don't rely on the shoulder buttons. **M**

Verdict

An incredible portable retro gaming build, this has just about everything you'd want from a Raspberry Pi-based Game Boy clone.

10/10

Pioneer600

This Raspberry Pi expansion board adds a range of features and professional sensor support. **Lucy Hattersley** senses if it's any good

SPECS

SCREEN
0.96-inch OLED display (128×64 resolution)

SENSOR:
BMP280 pressure sensor

I/O:
USB to UART, IR receiver (LFN0039K), 1-Wire interface, sensor interface, 8-bit analogue to-digital converter, PCF8591 port expander

OTHER:
DS3231 real-time clock, buzzer

Pioneer600 is an expansion board from Waveshare international, a Chinese manufacturer with a wide range of Raspberry Pi products. This unit found its way to us via The Pi Hut, which is reselling it in the UK.

The Pioneer600 is a blue expansion board that clicks onto the GPIO pins of Raspberry Pi and adds a range of additional features.

First of these features we noticed was a five-way joystick and 0.96-inch OLED display. This is backed up by a dual-LED and buzzer. On closer inspection, we discovered the on-board LFN0038K to enable remote control. A DS3231 real-time clock is also included (you'll need to supply your own battery).

There is a built-in BMP280 for measuring air pressure and temperature, but it's the available ports for external sensors that are the key attraction: a PCF8591, 8-bit resolution, screw terminal interface, a 1-Wire device (a DS18B20 thermometer is included), a 4×4 pin sensor interface, and GPIO expansion.

There is also a micro USB to UART connection and cable supplied. UART is normally used to debug Raspberry Pi, but with debugging mode disabled, you can send data to and from Raspberry Pi and a connected PC.

One for the C crowd

The physical setup is easy enough. This isn't an official HAT (hardware attached on top) standard board, so manual installation of the various software library components is required (**magpi.cc/p600installation**).

> " A professional sensor board that will find a happy home in an industrial environment running C code to sense and report "

Sample programs for bcm2835, WiringPi, and Python can be found on the Pioneer600 wiki (**magpi.cc/pioneer600wiki**). We had more success with the bcm2835 and WiringPi sample code (both are in C) than Python, where we encountered a few errors.

Perhaps this is better than the other way around: Pioneer600 is a professional sensor board that will find a happy home in an industrial environment running C code to sense and report on its findings. Ultimately, we recommend it to C developers over Python coders. M

Verdict

Pioneer600 is packed with a lot of I/O and the OLED screen is a nice touch. We found the Python sample code cranky, but the WiringPi code runs just fine. Better for C developers.

7/10
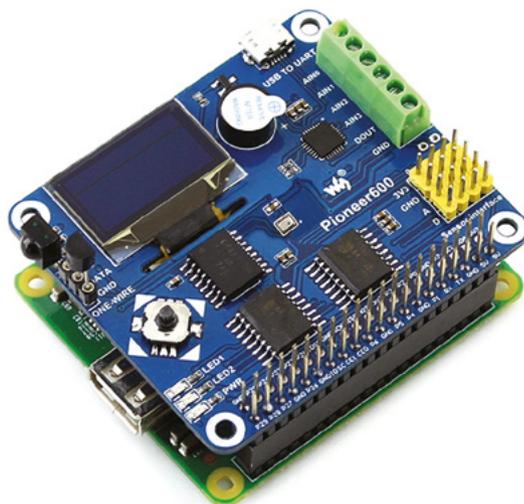
▲ It's compatible with Raspberry Pi 3 Model A+, pictured here, which is a good low-energy solution

Iono Pi Max combines an industrial server with a programmable controller in a single DIN-rail solution

Iono Pi Max is unique in its offering, as **no other solution** combines as much functionality with as many features in a single package

- Raspberry Pi CM 3/3+
- Switching Dual SD card
- RS-232 - RS-485
- CAN/CAN-FD
- 12V/24V UPS
- Digital/Analog I/O
- I2C - Wiegand - 1 Wire
- Relay outputs
- Power supply outputs
- Real Time Clock

- Hardware watchdog
- Secure element chip
- 32-bit micro-controller
- Low-power mode
- In-field FW/OS update support
- Wide-range power supply
- Power-controlled USB's
- Ethernet
- 6M DIN rail case
- CE/FCC/IC - RoHS compliant

POWERED BY Raspberry Pi

SFERALABS
www.sferalabs.cc

10 Amazing:

Raspberry Pi IoT projects

Use a Raspberry Pi to make your life just that extra bit smarter

One of the most popular uses of Raspberry Pi is IoT and home automation, and for good reason. Thanks to this, there are over eight years' worth of projects for replicating or getting inspiration to create the ultimate automated life. Here are some of our faves. **M**

▲ Rosie IoT Brick

Outdoor IoT

Allow friends and family to remotely track your marathon running. It can also act as a GPS for the runner if required.

magpi.cc/riot

▲ Smart doorbell

See who's ringing

Internet-connected doorbells are quite popular these days. Instead of buying an expensive system, build your own, better one.

magpi.cc/smartdoorbell

▶ PiRoomba

Robot vacuum enhanced

The Roomba vacuum cleaner is fantastic. Using a Raspberry Pi, though, you can supercharge it like Tim 'The Toolman' Taylor.

magpi.cc/piroomba
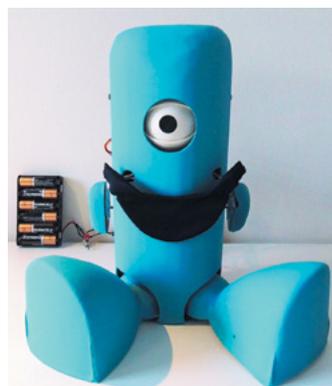
◀ Sleepbuddy

Robot babysitter

A social robot that can help looking after a child. It's not a full-on *Jetsons* maid, but it will help when you're trying to enjoy some telly.

magpi.cc/sleepbuddy

▲ Furlexa

Nineties throwback assistant

However you feel about Furbies, we're sure you'll definitely have stronger opinions about a smart Furby that can answer your questions. Put down the pitchforks.

magpi.cc/furlexa

▶ Smart home bulb

Clap on

Controlling your lights via the internet has never been easier, thanks to loads of companies now making smart bulbs you can hack.

magpi.cc/smartbulb

▶ MudPi

Automated gardening

Automated gardening is in vogue, and what better way than growing your own delicious vegetables (climate allowing) in your own garden with help from robots?

magpi.cc/mudpi
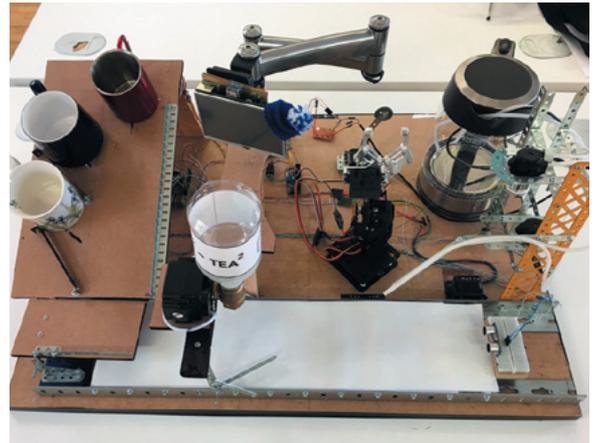
▲ Raspberry Pi AI Teasmade

Wake up to a brew

Looking less like a classic teasmade and more like a Rube Goldberg machine, this contraption will nevertheless make you a cuppa – possibly while playing *Powerhouse*.

magpi.cc/teasmade

▲ Magic mirror

Smart casual

A now popular Raspberry Pi project, smart/magic mirrors are an amazing way to make sure you're ready for the morning.

magicmirror.builders

▶ PiClock

Time and weather

A classic use of IoT is the weather. Turn a Raspberry Pi into something that powers a lovely clock that also displays the weather.

magpi.cc/piclock
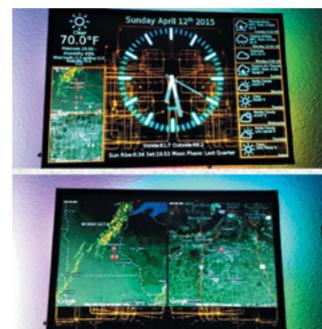
Learn beginner electronics with Raspberry Pi

Resources to help you get started in the world of electronics using Raspberry Pi. By **Phil King**

Simple Electronics with GPIO Zero

CREATOR

Phil King

Price:
£3 / $4

magpi.cc/ gpiozerobook

If you're not doing physical computing with Raspberry Pi, you're missing out on one of its great strengths. This compact 100-page Essentials guide from *The MagPi* takes you through the basics of connecting electronic components to Raspberry Pi and controlling them, before moving on to building more advanced projects.

The first few chapters explain how to connect common electronic components, such as LEDs and push-buttons, to Raspberry Pi's GPIO (general-purpose input/output) pins and then control them with programs written in Python. The GPIO Zero library makes it much more intuitive and easier than before, eliminating a lot of boilerplate setup code for the GPIO pins.

Fun projects to make include a motion-sensing alarm, laser tripwire, internet radio, LED thermometer, and basic robot. Clear wiring diagrams are provided for each project, along with photos and all the code required.
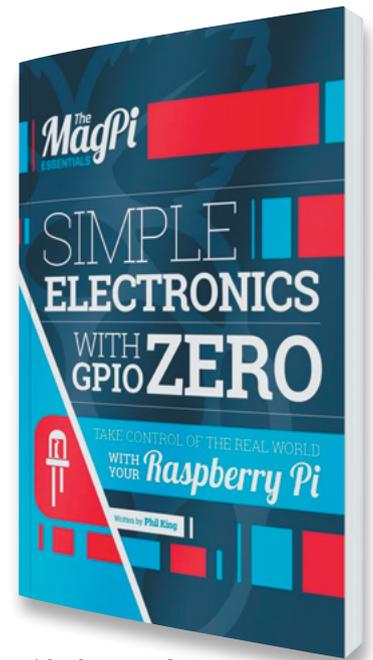
Like all *The MagPi* Essentials books, *Simple Electronics with GPIO Zero* is available to buy in print and digitally, or as a free download under the Creative Commons licence.

Books to read

Essential reading for learning electronics with Raspberry Pi
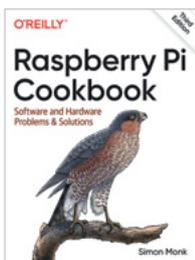
RASPBERRY PI COOKBOOK

This 608-page tome by Simon Monk features several chapters dedicated to connecting electronics such as LEDs, push-buttons, and sensors. Simon has also written an *Electronics Cookbook* for Raspberry Pi and Arduino.
magpi.cc/rpicookbook

GET STARTED WITH RASPBERRY PI

As well as a cover-mounted Raspberry Pi 3A+ with case and 16GB microSD NOOBS card, this 116-page book features an electronics starter guide along with coding tips and other helpful tutorials.
magpi.cc/getstartedbook

20 EASY RASPBERRY PI PROJECTS

By Rui and Sara Santos, this is a beginner-friendly collection of electronics projects. Making use of LEDs, buttons, potentiometers, sensors and more, projects include a weather station and WiFi RC robot.
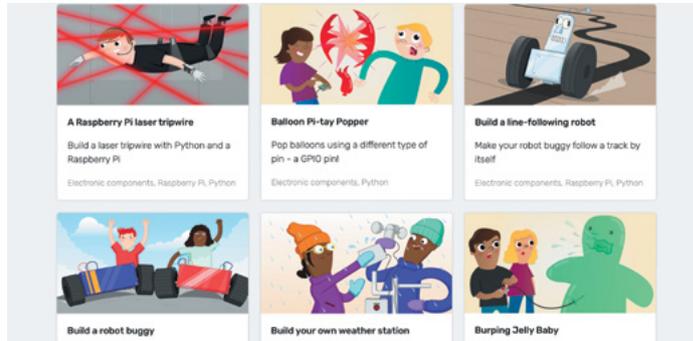**magpi.cc/20easy**

Raspberry Pi Projects

CREATOR

Raspberry Pi Foundation

Price:
Free

**magpi.cc/
electronicprojects**

The Raspberry Pi Foundation's learning resources website features a wide range of electronics projects to get you started, mainly based on Raspberry Pi but also a few using micro:bit, Crumble, and other boards.

To help you begin, there are guides to getting started with physical computing using both Python and Scratch – the Raspberry Pi version of Scratch features special extension blocks for controlling electronics (as well as the Sense HAT board).

Fun Raspberry Pi projects include the classic Burping Jelly Baby, Fart Detector, Robot Buggy, Grandpa Scarer, Spinning Flower Wheel, Lights Out game, and Ultrasonic Theremin. Each project features step-by-step illustrated instructions, along with a list of all the hardware you'll need. For educators, many of the resources cover elements from the Raspberry Pi Digital Making Curriculum.

Pi My Life Up

CREATOR

Pi My Life Up

Price:
Free

pimylifeup.com

This popular website features a range of Raspberry Pi electronics (and other) projects. There are also helpful guides to the GPIO pinout, configuring I²C and SPI protocols, and technical aspects such as understanding Ohm's law, identifying resistors, and creating a voltage divider.

The Raspberry Pi electronics section features a good array of projects demonstrating how to use hook up and control a variety of components such as an analogue-to-digital converter, RFID chip, LED strip, real-time clock, accelerometer, and sensors for ultrasonic distance, temperature, humidity, light, and motion. Each detailed step-by-step tutorial is illustrated with plenty of photos, wiring/circuit diagrams, and sometimes a video demonstration. Example code listings are also supplied, making this a great place to get started with electronics on your Raspberry Pi. 🄼

Electronics kits

Get some useful components in a handy kit

CAMJAM EDUKIT #1

A great starter kit in a tin for just £5, it features LEDs, a button, buzzer, breadboard, and wires. Step-by-step guides show you how to use them in simple projects.
magpi.cc/camjamkit

CAMJAM EDUKIT #2

Moving on from the basics, this kit helps you get to grips with sensors including a PIR, LDR, and immersible temperature probe. Projects include a burglar alarm.
magpi.cc/camjamkit2

MONKMAKES PROJECT BOX 1

Put together by noted technology author Simon Monk, this kit includes LEDs, a thermistor, phototransistor, switches, and more – everything you need to make the ten projects detailed on instruction cards.
magpi.cc/projectbox1

Mark Calleja, aka Mister C

The new master of live streams talks to us about how the Raspberry Pi Foundation has been helping to teach kids at home

> ⟩ Name **Mark Calleja** | ⟩ Community role **Educator**
> ⟩ Occupation **Learning Manager** | ⟩ Twitter **@M1st3r_C**

The Raspberry Pi Foundation is an educational charity that provides a lot of free teacher training, guides, and more for computing and digital making. With a lot of kids having to learn from home this year, the Foundation folks decided to try something new: a weekly Digital Making at Home stream hosted by Mister C, aka Mark Calleja of the Raspberry Pi Foundation.

"As a response to the early days [of young people being home] here in the UK, the Foundation decided that in the absence of being able to attend school, it would be beneficial to young people if we could provide them with some educational content that they could engage with at home," Mark explains. "Back when we started, the Foundation threw together a team of people to answer the question on what streaming and video learning content would look like and we all just sort of made the thing happen. We were filming in bedrooms and basements initially, with whatever equipment we had at home."

The streams have evolved greatly since then to be a lot more streamlined and useful, and the Foundation also provides pre-recorded videos for students to follow along with.

> ❝ We all just sort of made the thing happen ❞

How did you first learn about Raspberry Pi and related programs?

It was way back in 2014, when I was still teaching Year 3. I used to run a code club for my school and I was on the second ever Picademy, held at the old Raspberry Pi offices up on the hill in Cambridge. Since then, I've taught literally thousands of people with a Raspberry Pi – from six- and seven-year-olds taking their first steps in coding, to brainstorming ways for postgraduate marine biologists to maintain lab conditions, to the thousands of young people who run their code each year on the International Space Station

▼ *The MagPi* Features Editor Rob was on the live stream to talk about game development

▲ Mark and other members of the Foundation receiving awards from the Scouts

as part of our Astro Pi project! I've been working with (and for) Raspberry Pi for quite a while, and now each week I get to host a live stream that is broadcast to the whole world, aimed at giving young people the opportunity to do some coding at home wherever they are. I'm very lucky.

What has the reaction been to the streams?

Really positive! We've had loads of guests from all over the world come onto the stream to discuss what they do, or do some coding with us – we've had people from all over the world as well as a few young people from Ireland and the UK. Now that we're running at a later time each week, we're able to get a lot more young people from the US to drop by for some practice in programming, both on and off camera. We've had environmental scientists,

▲ Christina Foust from the North American Raspberry Pi team often hosts alongside Mark

editors, inventors, and iconoclasts on the stream, as well as representatives from the European Space Agency and even one of the Founders of Raspberry Pi, Eben Upton. We're taking a lot of what we're learning and using it to help run online clubs, as well as training sessions for groups like the Scouts.

Any upcoming plans you can share?

This whole month on Digital Making at Home is about Self Care – we're looking at ways you can create with technology that are therapeutic and stress-relieving during back-to-school madness. Have a look at **rpf.io/home** for more videos and info! 𝙈

This Month in
Raspberry Pi

MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made. This month we received a huge amount of excellent entries, so please enjoy three whole pages of community projects, and remember to follow along at the hashtag #MagPiMonday! 🅼

01. It's only black and white, but we love the look of this Retro Player

02. PiMowBot, previously featured in *The MagPi*, is getting upgrades

03. It's amazing what you can do with only a limited number of pixels

04. The HAT really makes this project from Brian look lovely

05. We're not sure what's more impressive – the detailing or automated home-brewing

06. Definitely do not try this at home, but you can at least admire making this with a Raspberry Pi

07. A classic and very useful IoT setup with a Raspberry Pi

08. The 'Blooming Pi Matrix' is a far prettier voice assistant

09. 3D-printed cases and accessories continue to be great

10. We hope this robot uses complete voice control for all the challenges

01

TeCoEd
@Dan_Aldred

Replying to @TheMagPi

I built a Retro Player. Displays a range of short films and the @TENETFilm trailer. OLED screen and @Raspberry_Pi 3B. More build details here. tecoed.co.uk/retro-player.h...
#MagPiMonday

0:26   1.2K views

02

TGD
@TGD_Consulting

Replying to @TheMagPi

I tweaked the remote control of PiMowBot to use a virtual joystick (nippleJS) from @YoannM on touch devices for a better navigation for lefties & righties.

Don't miss PiMowBot's showcase @TheMagPi issue #96.

03

geekinchief
@geekinchief

Replying to @TheMagPi

My son and I programmed a platformer game to run on the @pimoroni Unicorn Hat HD.

04
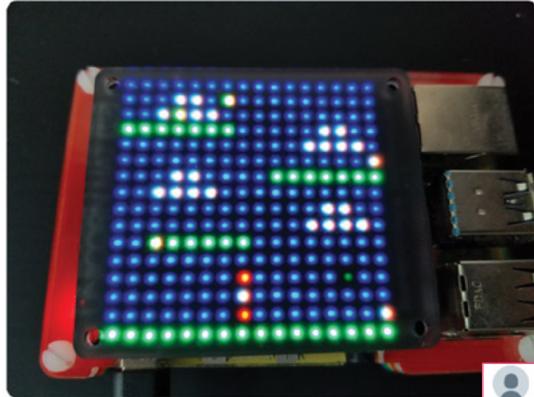
Brian Corteil
@CannonFodder

Replying to @TheMagPi

created a remote display for our Raspberry Pi weather station temperature using MQTT to sent and receive values. The display is showing the external temperature when the LEDs are on. Hat is a @pimoroni rainbow hat

0:06 1.9K views

05

mabraidot
@mabraidot

Replying to @TheMagPi

I did an automated brewing beer machine.
photos.app.goo.gl/Ha9v7VJRAD3Qqo...

06

Mochilero.V
@VMochilero

Replying to @TheMagPi

Optical lens +HD301D1 laser projection module + Pi zero= Retina projection unit

Translate Tweet

07

Ash Smith
@DrAshSmith

Replying to @TheMagPi

My project this week ... a cabinet and cheap LED strip lights from a popular Swedish furniture manufacturer now controllable from Home Assistant using a @Raspberry_Pi #MagPiMonday

08

Gam3t3chElectronics
@Gam3t3ch

Replying to @TheMagPi

I finished up my b.p.m. project with the @MATRIX_Creator voice running alexa on it. Making my google home look like a brick. Full project is up one the @E14Community here
element14.com/community/comm...

09

Allen Heard
@AllenHeard

Replying to @TheMagPi

Maaaaaybe

10

Ciprian Manea
@ovipic

Replying to @TheMagPi

Testing voice control options on our @unpi_ong #RVR bot for #PiWars #unPi

11

PenguinTutor
@penguintutor

Replying to @TheMagPi

I've written up details about my latest Raspberry Pi robot based around the T200 tank chassis, with some added 3D printed parts.

penguintutor.com/projects/robot...

#MagPiMonday #Robot
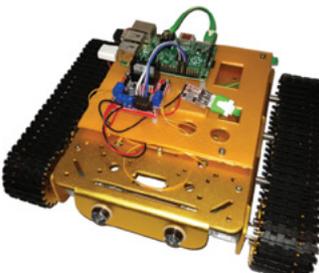
13

Stewart Watkiss
@stewartwatkiss

Replying to @TheMagPi

I've created a 3D printed attachment for the STS-Pi robot for a ultrasonic distance sensor.

Used to add collision avoidance (well sort of).

penguintutor.com/projects/robot...
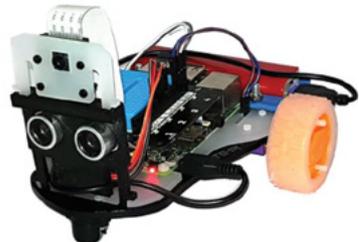
12

8 Bits and a Byte
@8BitsandaByte

Replying to @TheMagPi

An Interactive Storytelling Radio!

15

GaryHomewood
@GaryHomewood

Replying to @TheMagPi

I had fun with eInk generative art

#MagPiMonday

GaryHomewood @GaryHomewood · Aug 19
Monochrome eInk generative art now with a webpage refresh. Tentative artist's finger clearly visible.
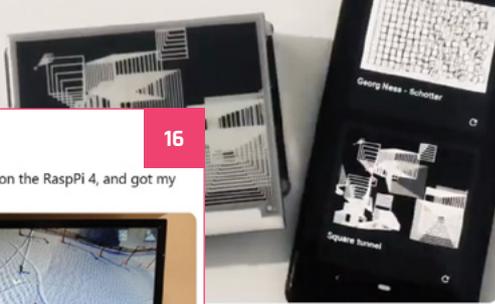
14

TGD
@TGD_Consulting

Replying to @TheMagPi

I did some brainstorming about the next model of PiMowBot. This might be the size of Raspberry #PiZero based PiMowBot XXL...

#MagPiMonday #Make #3Dprinted #IoT #smart #robotics #lawnmower based on #RaspberryPi #PiZero & #Rebol

TGD @TGD_Consulting · Aug 21
When size does matter. For comparison, this might be the solar panel of PiMowBot's big brother. Coming soon. Available next year.

#Make #3Dprinted #IoT #smart #robotics #lawnmower with #RaspberryPi & #Rebol. All 3D models @Cults3D (bit.ly/2YxRoYw).

16

Michael Field
@field_hamster

Replying to @TheMagPi

Got Android 10 running on the RaspPi 4, and got my fave game running.
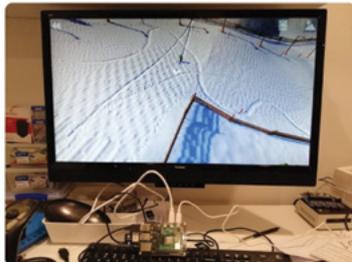
11. This robot looks like serious business

12. 8 Bits and a Byte are back with another fun little project

13. The STS-Pi is a nice, inexpensive little robot kit, so this is a very cool addition to it

14. That is a huge solar panel! We love the PiMowBot upgrades

15. Make sure to watch the full video of this – it's stunning! **magpi.cc/6pictb**

16. We're always really impressed when we see Android running on a Raspberry Pi – it's usually a lot of hassle to do

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month

HAT RACK

A 'DIN Rail Mount for Raspberry Pi HATs', which in layman's terms means you can connect multiple HATs to one Raspberry Pi. We're seeing a little trend of these at the moment, and they're a great idea to be honest. Always looks a bit suspect when you have random wires connected to GPIO pins via a HAT.

▶ **kck.st/2YVlZNP**

Game Gear / Raspberry Pi Mod Kit

This we really like. A full modification kit for the classic battery-guzzling Sega Game Gear to make it into a RetroPie machine. You can add extra buttons as well, so you're not limited to the Game Gear's three. It replaces the power with the ability to add a rechargeable lithium battery, and the setup is a lot more efficient than the old Game Gear anyway.

▶ **kck.st/2Q22j60**

CROWDFUNDING A PROJECT?

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

POSSIBLE EVENT CANCELLATIONS

Please follow local public health advice and take decisions on whether to cancel clubs or events in consultation with the venues that host them.

Raspberry Jam
Event Calendar

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

01. Hackaday U: Linux + Electronics: A Raspberry Pi Course
- 📅 **Thursday 1 October**
- 📍 **Online**
- ▶ **magpi.cc/laeklg**

This course is for those who had wanted to go from Arduino to a much more powerful Raspberry Pi.

02. Houston Raspberry Pi Meetup
- 📅 **Wednesday 7 October**
- 📍 **Online**
- ▶ **magpi.cc/7eqkqj**

A community of coders, artists, educators, and engineers interested in project-based learning with Raspberry Pi.

03. Fashion Womenpreneur Startup Hackathon
- 📅 **Wednesday 7 October**
- 📍 **Online**
- ▶ **magpi.cc/vhrrze**

Tips on how to combine fashion and technology for your own business idea.

04. West Sound CoderDojo
- 📅 **Saturday 10 October**
- 📍 **Online**
- ▶ **magpi.cc/5qvcdo**

Ninjas, parents, mentors, and all are invited to join West Sound CoderDojo for online biweekly sessions.

05. Medway Makers Tech Meetup
- 📅 **Sunday 18 October**
- 📍 **Online**
- ▶ **magpi.cc/4agosj**

Workshops in building cool things with Arduino, how to code for beginners, how to use a Raspberry Pi, etc.

06. Growing Radio Amateurs
- 📅 **Wednesday 21 October**
- 📍 **Online**
- ▶ **magpi.cc/amse3w**

Every month, the local SCRA club (Ham Radio) is holding beginner electronics classes.

07. BCG Platinion Hackathon 2020
- 📅 **Friday 23 October**
- 📍 **Online**
- ▶ **magpi.cc/4bd3br**

Several cities, one goal: coding digital strategies that shape the digital world of tomorrow, unlock digital potential, and ignite change!

08. Virtual Maker Meetup
- 📅 **Tuesday 27 October**
- 📍 **Online**
- ▶ **magpi.cc/qlloxd**

Join other makers for a virtual meet-up to share ideas, inspire and encourage each other, and highlight the things made.

FULL CALENDAR

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

rpf.io/jam

FIND OUT ABOUT JAMS

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

jam@raspberrypi.org

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

Raspberry Jam advice:
Fundraising

"**In order to avoid charging for tickets, we always hold a raffle and raise between £60 and £200, and everyone's happy whether they win or not.** You can use Twitter to ask for swag from the community, to give away as prizes. It's so nice to be able to run community events which are free to attend."

Grace Owolade-Coombes – South London Raspberry Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **rpf.io/guidebook**
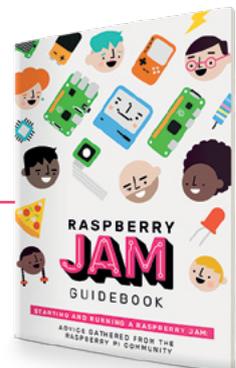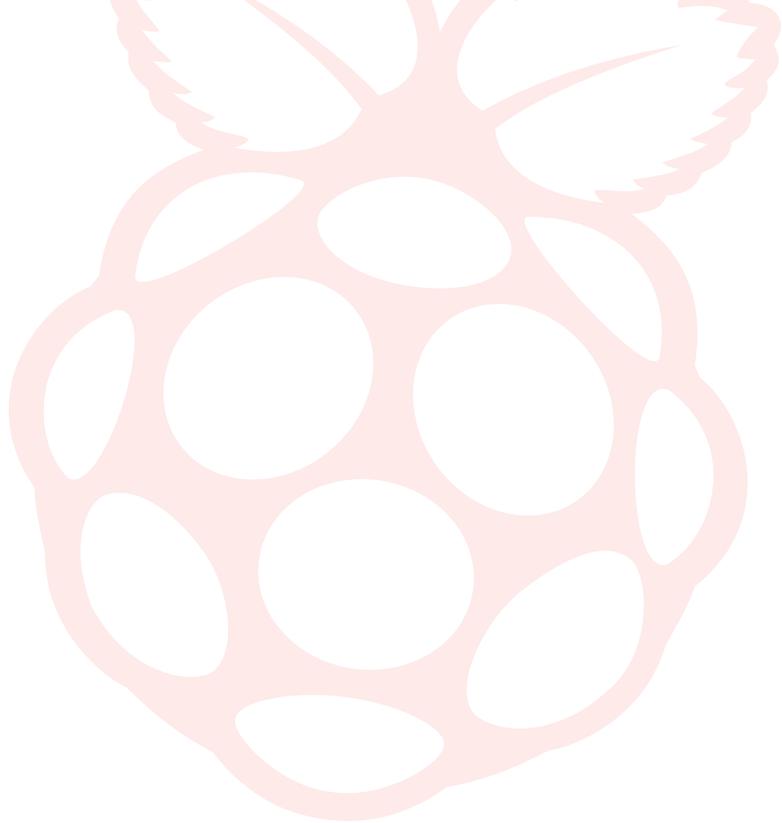
Your
Letters
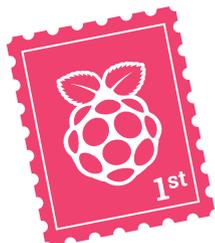
▼ This issue was all about Astro Pi – physical versions came with a poster and sew-on mission badge!

Creative Commons

I am currently studying a computer science degree and I need to design a website which has some articles. I selected Raspberry Pi as the topic for my website as I do a lot of work with them. I have at least five Raspberry Pi [computers] and try to incorporate them wherever I can.

I was wondering if I could use some text from your articles on my site? I will of course reference you and the writer on the site. This site will also not be publicly accessible. It will only be visible to my tutors.

Ben via email

This is perfectly fine to do – *The MagPi* is licensed under Creative Commons and this means the content is free to use and remake as you desire, as long as you say where the content originally came from. You're not allowed to make money with it though. So no ads! Either way, under these licensing conditions, you can actually set the website public and it would be just fine.

Good luck with your degree.

▶ Learn more about the way we license the magazine at **magpi.cc/about**
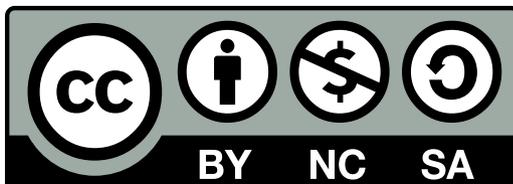
Errant

The link 'View the contents pages' on the issue 47 webpage incorrectly points to the table of contents for issue 46.

Is it possible to obtain the correct address?

Marcus via email

This has been fixed now, but if you ever see any weird links on *The MagPi* website, be sure to let us know via an email or social media so we can get it fixed! Long-time followers will know we moved the website recently and some of the older content may still have some hidden bugs.

▼ Check out our Haunted House feature for some inspiration

Battery life

I'm looking at making a portable Raspberry Pi that runs off a battery – I wanted to know what kind of battery you recommend and how long it will run a Raspberry Pi for?

Laura via twitter

This is unfortunately not a simple answer. A Raspberry Pi 4 will generally consume more power than a Raspberry Pi Zero, but how much power consumed depends on what you're doing with it. In the portable computer feature earlier in the issue, we recommended a specific power bank; however, that may be overkill for some projects, as the size of the power bank may be a factor.

All you can really do is make sure you can get the largest capacity power bank in the correct size for whatever you're building, and make sure it supplies 3 A for a Raspberry Pi 4.

Future frights

Will you be sharing people's Halloween projects this year? Where should I send them? I've got some ideas I was wanting to make and thought now would be a great time to use my free time and get creating!

Joan via Facebook

Absolutely, we love sharing what people have made anyway, and we especially love the creativity of people during Halloween. We'll try to feature as many as we can in This Month in Raspberry Pi over the next couple of issues.

To submit stuff, just email us or send pictures over social media! Video can work as well, although we can't quite print those (yet).

Contact us!

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

WIN ONE OF THREE
M.A.R.S.
ROVER ROBOT KITS!

Last issue (**magpi.cc/97**), we reviewed the M.A.R.S. Rover kit and liked it a lot! Now is your chance to win one!

❝ The 4tronix M.A.R.S. Rover is loosely based on the Curiosity and Mars 2020 rovers from NASA/JPL. It uses the same rocker arm, bogey, and differential arm mechanism. ❞
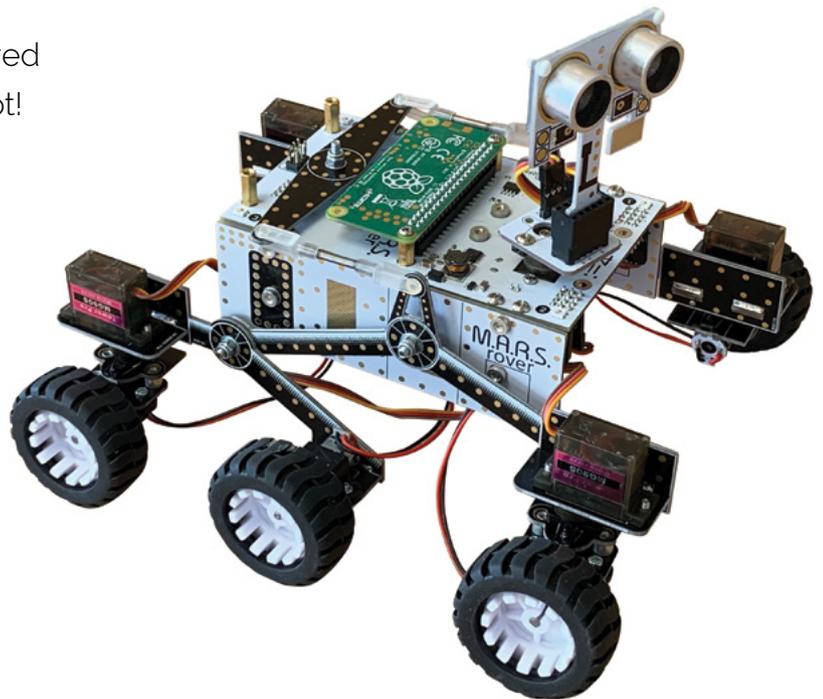
Head here to enter: magpi.cc/win | **Learn more: magpi.cc/marsrover**

Terms & Conditions

RASPBERRY PI
BUYERS' GUIDE

THE MAGPI #99
ON SALE 29 OCTOBER

Plus!
Play with Raspberry Pi Guitar

Build a smart BBQ

Discover the Mars Clock

DON'T MISS OUT!
magpi.cc/subscribe

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

Coding on-the-go

Running around with computers is something
Lucy Hattersley has been doing for years

don't know about anybody else, but things seem to have taken a turn for the better. The weather, which was a bit over-the-top last month and no mistake, has chilled out and now it's a lovely day.

I've been getting out and about, heading to Raspberry Pi Towers in Cambridge, working from a variety of coffee shops, and patiently waiting for the library to reopen.

Of course, there's still time for things to go a bit pear-shaped but right here, right now, everything seems to be going OK. So I'm going to roll with it and enjoy the rest of autumn.

I hope you've enjoyed this edition of *The MagPi*: every edition is made with a lot of love, and more than a little luck.

We make *The MagPi* to inspire folks to make incredible things with Raspberry Pi. Sometimes things just line up, and it seemed particularly fortunate that I've been testing out a lot of Raspberry Pi laptop and portable computing devices. At the same time, PJ was working on a portable Raspberry Pi 2Go build.

I'm still not wholly sure which is my favourite mobile option. For a while I enjoyed the pi-top, then the NexDock. Recently I've been using an iPad Pro with the USB-C data connection, but it feels a little incongruous and I'd prefer something direct. I'm looking forward to pi-top's new screen and keyboard offering.

❝ I feel my life is richer for having a Raspberry Pi in it ❞

One thing for sure is that using Raspberry Pi on the move is far better for me than any other laptop or tablet.

Science to go

Because Raspberry Pi OS is built on top of Debian, I have real control over my computer.

On top of that, the GPIO pin access enables me to experiment with hardware and electronics in ways that are unthinkable with a smartphone or tablet.

Understanding how computers work is an ongoing passion of mine. These things that we put so much

faith into, and use to organise so much of our lives, shouldn't be a mystery.

Lately, I've been following the nand2tetris course (**nand2tetris.org**) and using Raspberry Pi to explore creating a computer from virtually nothing. It's tough going at times but is a rewarding experience. Hopefully, I'll make it to the end and be able to make a Tetris console of my own. I'm hoping to put a screen and buttons on a Raspberry Pi and make my own Game Boy-style machine.
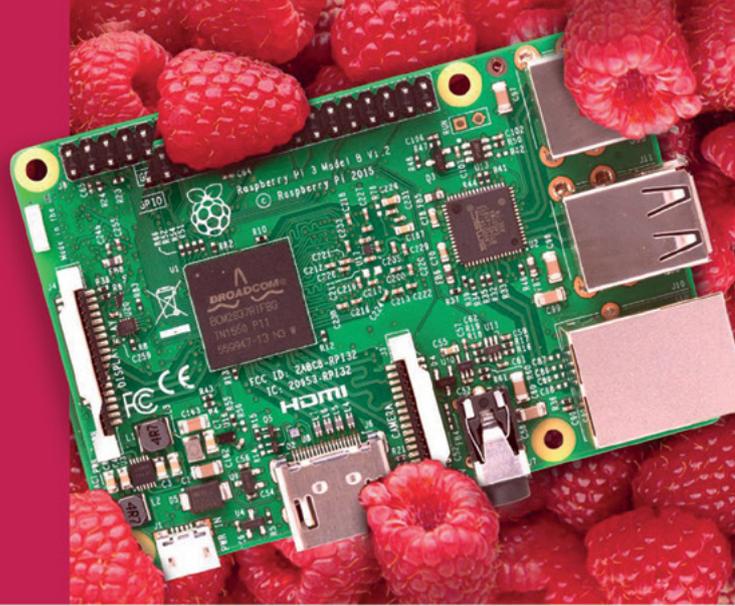
Whatever the outcome, I feel my life is richer for having a Raspberry Pi in it, and one that I can carry around and use wherever I am. ▥

AUTHOR

Lucy Hattersley

Lucy is editor of *The MagPi* and has a very bouncy new pair of trainers, along with a myriad of mobile Raspberry Pi options.

magpi.cc