THE *Official*

RASPBERRY PI HANDBOOK

2025



Olga **Fortune Teller**



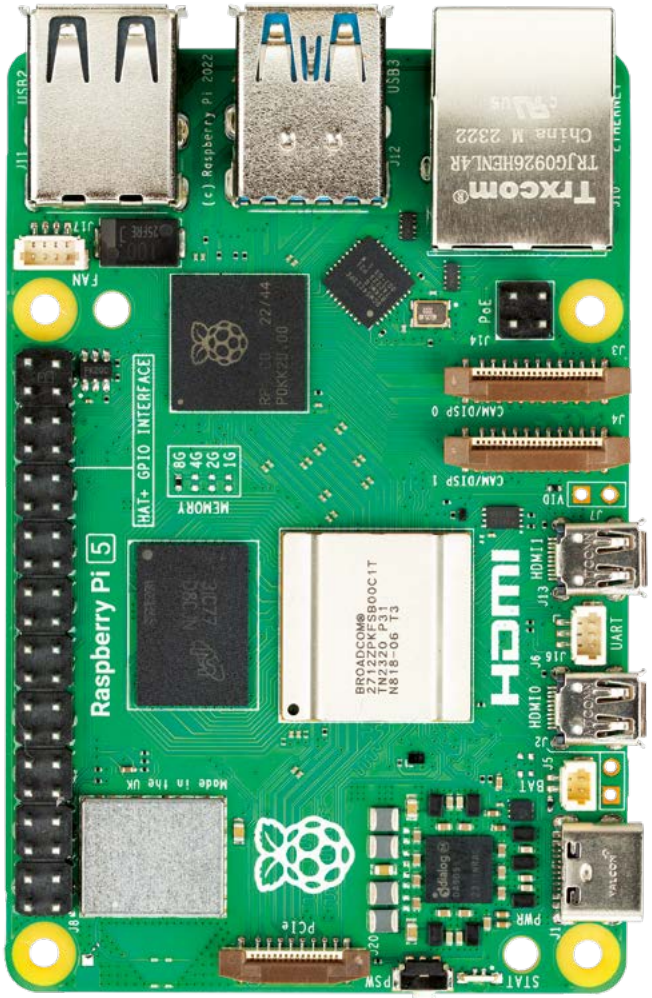
Astrophotography



Exoskeleton



Blue **ROV2**





Emulate a **BBC Micro**



CinePi **XL**



Python **Robots**

200 pages of Amazing Makes!

Including **RASPBERRY PI PICO 2**



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

SFERALABS

Strato Pi Max XL and XS



Powered by Raspberry Pi

Modular Industrial Server

Powered by the Raspberry Pi CM4 and RP2040

- Up to 4 expansion slots
- Dual SD, eMMC and SSD storage options
- 2 x Ethernet ports
- 2 x USB ports with power control
- WiFi & BLE
- DIN rail mounting

Open Technology for the Industrial World

Long Term Availability, High Reliability and Support

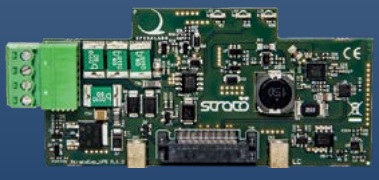
X2-Series Expansion Boards



Isolated CAN and Dual RS-485



Industrial Analog Current or Voltage Outputs



UPS



RS-232 and Isolated RS-485

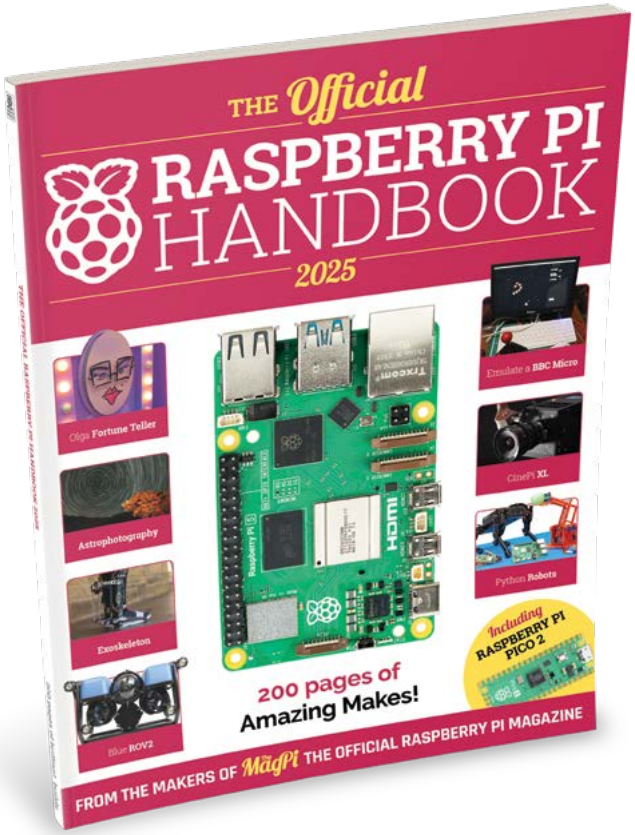


Isolated Industrial Analog Current, Voltage and Temperature Inputs



Isolated Industrial Digital I/O

www.sferalabs.cc



WELCOME!

It's been a huge year for Raspberry Pi. Raspberry Pi 5 has turned roughly a year old, Raspberry Pi Holdings is now listed on the London Stock Exchange – a big milestone that we're proud to have reached – and, as I write this, the powerful new Raspberry Pi Pico 2 (along with the all-important RP2350 chip that it's built around) has just been released. In the community, events are truly back in full swing, with dedicated calendars on the Raspberry Pi website and in the pages of The MagPi, and with all of you still making incredible things using various Raspberry Pi devices. It's all very exciting.

This is reflected in the articles you'll find in this book! Once again, it was extremely difficult to choose what to squeeze into these pages, but we've managed to include an incredible range of recent projects and tutorials, as well as a very cool, in-depth interview with Raspberry Pi engineers about the making of Pico 2 and RP2350.

I think there's something for everyone in the book this year – and maybe even something that will expand your horizons, too. Happy making!

Rob Zwetsloot

FIND US ONLINE magpi.cc **GET IN TOUCH** magpi@raspberrypi.com

The MagPi

Available on the App Store

GET IT ON Google Play

CC BY NC SA

EDITORIAL
Editor: **Lucy Hattersley**
Features Editor: **Rob Zwetsloot**
Contributors: **David Crookes, PJ Evans, Rosie Hattersley, Phil King, Nicola King, KG Orphanides, Danny Staple, Ashley Whittaker**

DESIGN
Critical Media: **criticalmedia.co.uk**
Designers: **Lee Allen, Sam Ribbits**
Illustrator: **Sam Alder**
Raspberry Pi Designers: **Sara Parodi, Natalie Turner, Jack Willis**

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave, London, EC1A 9PT | **+44 (0)207 429 4000**

MAGAZINE SUBSCRIPTIONS
Unit 6, The Enterprise Centre, Kelvin Lane, Manor Royal, Crawley, West Sussex, RH10 9PE | **+44 (0)207 429 4000**
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

PUBLISHING
Publishing Director: **Brian Jepson**
brian.jepson@raspberrypi.com

Advertising: **Charlotte Milligan**
charlotte.milligan@raspberrypi.com
Tel: +44 (0)7725 368887

Director of Communications: **Helen Lynn**
CEO: **Eben Upton**

recycle
When you have finished with this magazine, please recycle it.

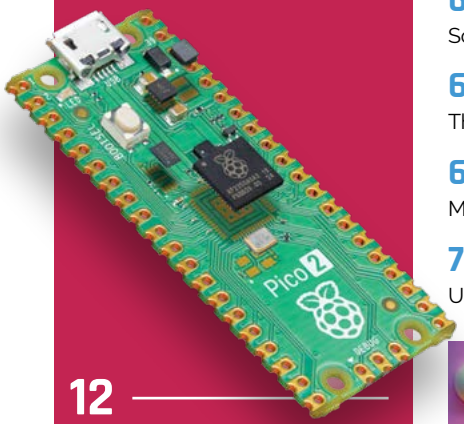
This bookazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

This official product is published by Raspberry Pi Ltd, Maurice Wilkes Building, Cambridge, CB4 0DS. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 978-1-912047-42-0.

Contents



06 Raspberry Pi Starter Guide



12 Raspberry Pi Pico 2

20 Raspberry Pi Adventure

Project Showcases

34 Exoskeleton
Raspberry Pi-powered walking

38 Team Pinball
These steel ball runs have a Pico twist

42 Odyssey Lights
Giant interactive lights at Blackpool

46 Tank driving simulator
An old trainer with new technology

50 Doom on a LEGO brick
A boomer shooter knee deep in the studs

52 Bugg.xyz acoustic monitoring
Conservationists take soundscapes of nature

56 Refurbished vending machine
Dispensing snacks thanks to Raspberry Pi

60 Antarctic Pi Cam
Scientific analysis from home to the pole

64 Olga the Fortune Teller
This AI-powered seer might reveal your future

68 Mona Lisa Fluid Painting
Microfluidic biosensors meet Da Vinci

72 Pipstrelle bat detector
Using echo location in reverse to identify bats





76 Badgercam
Keep an unobtrusive eye out for badgers

80 Roktrack
Robotic lawn care in rural Japan

84 CHIP bipedal companion robot
This walking robot could be your pal

88 Bristol Braille Canute 360
A custom, multi-line Braille reader

92 ScreenDress
This 3D-printed dress has its eyes on you

94 Retro Pi World Radio
An old-school radio with a global tuner

98 BlueROV2 R4
Dive under the sea remotely and safely

100 CinePI XL
A high-quality Raspberry Pi cinema camera

104 Volcano-monitoring UV PiCam
A Raspberry Pi-powered volcanic alert system

108 PeggyBoard
A DIY smart climbing wall built in a garage

112 Mini Observatory
Greenwich observatory on your kitchen table



Maker Guides

118 Astrophotography with Raspberry Pi
See the stars better than ever before

124 Python robots
Build dynamic robots with arms and legs

136 ML-powered audio transcriber
A must for any interviewer

140 DIY Home Automation
How to make your house smart

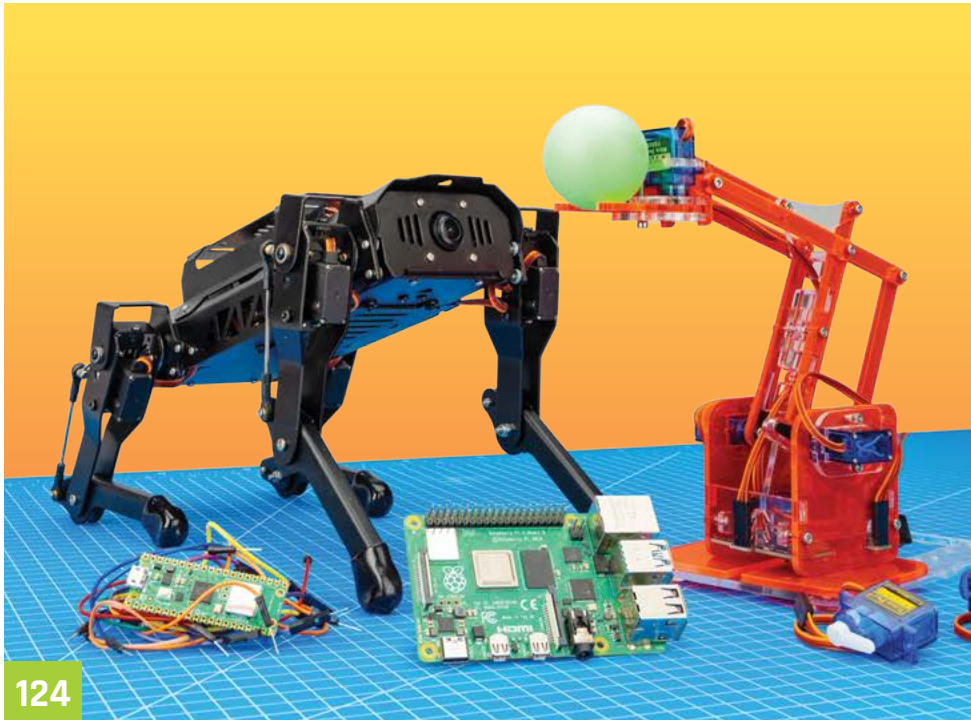
148 Emulate a BBC Micro on Raspberry Pi 400
Retro computing meets Raspberry Pi

152 Next Gen Retro Gaming on Raspberry Pi 5
Take it back to the 2000s

160 Build a universal media player
The last media centre you'll ever need

168 Arts & Crafts with Raspberry Pi
Get creative with a microcomputer





Reviews and Resources



176 Galactic Unicorn
A big screen for a tiny Pico

178 Badger 2040 W
An eInk badge for your next convention

179 Inky Impression 7.3"
A huge, multicoloured eInk display

180 PicoVision
Output video from your Pico

182 Vizy
A Raspberry Pi-powered AI camera

184 PiBoy XRS
The ultimate handheld gaming case kit

186 Argon NEO 5
The new NEO case for Raspberry Pi 5

188 Argon ONE V3 M.2 NVMe
Add superfast storage to Raspberry Pi

190 NVMe Base
Expand the PCIe port of Raspberry Pi 5

192 Puppy Pi
A state-of-the-art dogbot

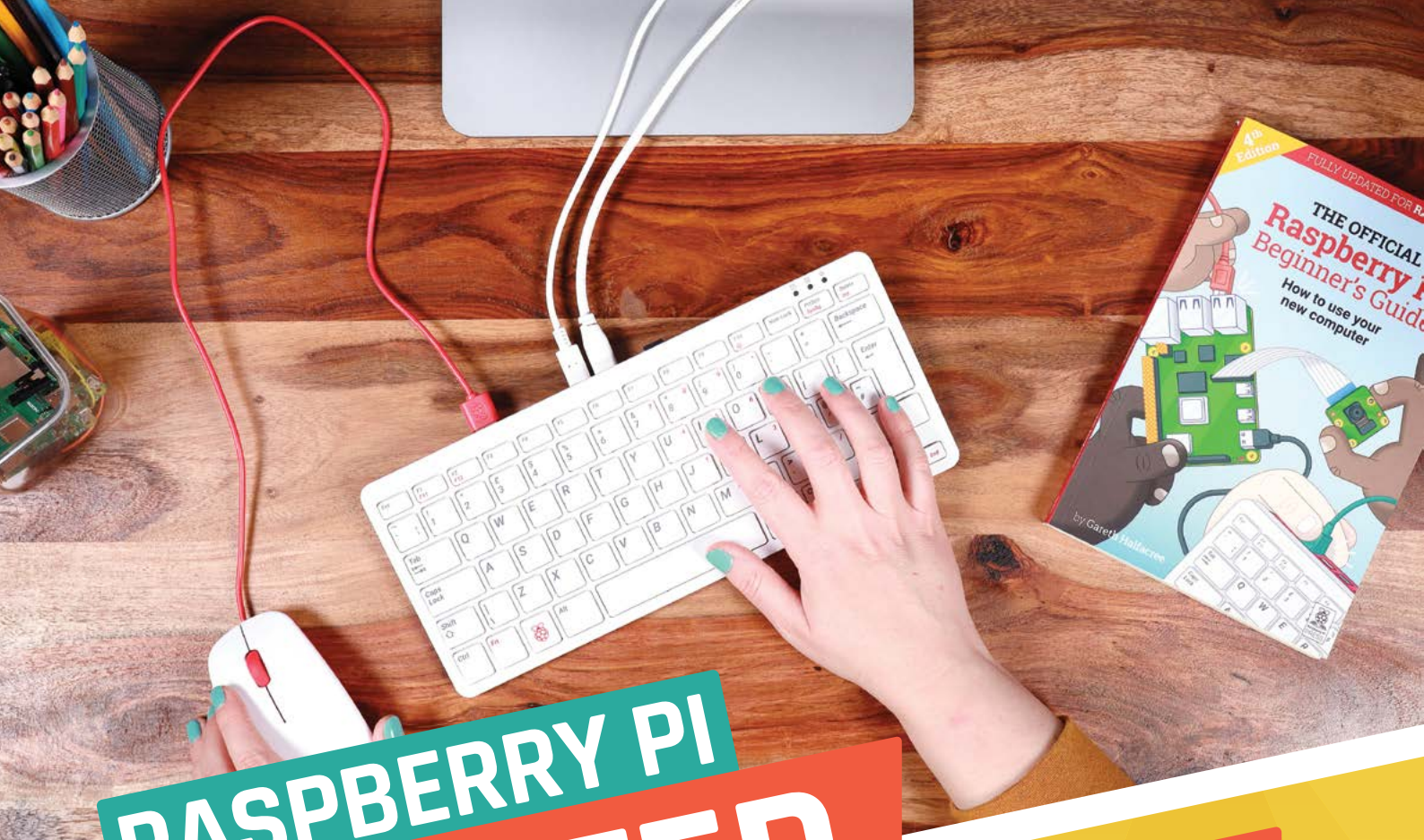
194 CM4 XGO-Lite robot dog
A powerful robot on four legs

195 Inventor HAT Mini
A tiny, all-in-one electronics playground

196 10 amazing cases
Keep your Raspberry Pi safe

198 10 amazing starter kits
Everything you need to start making

200 10 amazing Pico projects
Inspiration for the smallest Raspberry Pi



RASPBERRY PI STARTER GUIDE

SETTING UP RASPBERRY PI IS PRETTY STRAIGHTFORWARD, JUST FOLLOW THE ADVICE OF **ROSIE HATTERSLEY**

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

Getting started won't take long: just corral the extra bits and bobs you need on our checklist. Useful additions include some headphones or speakers if you're keen on using Raspberry Pi as a media centre, or a gamepad for use as a retro games console.

To get set up, simply use your pre-written microSD card (or use Raspberry Pi Imager to set up a card) and connect all the cables. This guide will lead you through each step. You'll find the Raspberry Pi OS, including coding programs and office software, all available to use. After that, the world of digital making with Raspberry Pi awaits you.

What you need

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

A Raspberry Pi

Whether you choose the new Raspberry Pi 5 or a Raspberry Pi 4, 400, 3B+, 3B, Zero or Zero 2 W (or an older model of Raspberry Pi), basic setup is the same. All Raspberry Pi computers run from a microSD card, use a USB power supply, and feature the same operating systems, programs, and games.

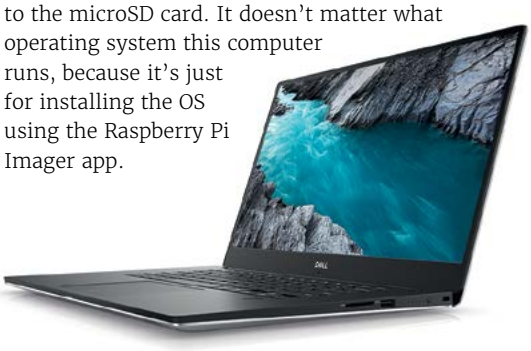


8GB microSD card

You'll need a microSD card with a capacity of 8GB or greater. Your Raspberry Pi uses it to store games, programs, and boot the operating system. Many Raspberry Pi computer kits come with a card pre-written with Raspberry Pi OS. If you want to reuse an old card, you'll need a card reader: either USB or a microSD to full-sized SD (pictured).

Windows/Linux PC or Mac computer

You'll need a computer to write Raspberry Pi OS to the microSD card. It doesn't matter what operating system this computer runs, because it's just for installing the OS using the Raspberry Pi Imager app.



USB keyboard

Like any computer, you need a means to enter web addresses, type commands, and otherwise control Raspberry Pi. The Raspberry Pi 400 comes with its own keyboard. Raspberry Pi sells an official Keyboard and Hub (**magpi.cc/keyboard**) for other models.





USB mouse

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



Power supply

Raspberry Pi 5, 4 and 400 need a USB Type-C power supply. Raspberry Pi sells power supplies (**magpi.cc/usbcpower**), which provide a reliable source of power. Raspberry Pi 1, 2, 3, and Zero models need a micro USB power supply (**magpi.cc/universalpower**).



Display and HDMI cable

A standard PC monitor is ideal, as the screen will be large enough to read comfortably. It needs to have an HDMI connection, as that's what's fitted on your Raspberry Pi board. Raspberry Pi 5, 4 and 400 can power two HDMI displays, but require a micro-HDMI to HDMI cable. Raspberry Pi 3B+ and 3A+ both use regular HDMI cables; Raspberry Pi Zero W needs a mini-HDMI to HDMI cable (or adapter).



USB hub

Raspberry Pi Zero and Model A boards have a single USB socket. To attach a keyboard and mouse (and other items), you should get a four-port USB hub (or use the official USB Keyboard and Hub which includes three ports). Instead of standard-size USB ports, Raspberry Pi Zero has a micro USB port (and usually comes with a micro USB to USB-A adapter).

SET UP RASPBERRY PI

Raspberry Pi 5 / 4 / 3B+ / 3 has plenty of connections, making it easy to set up

01 Hook up the keyboard

Connect a regular wired PC (or Mac) keyboard to one of the four larger USB-A sockets on a Raspberry Pi 5 / 4 / 3/3B+. It doesn't matter which USB-A socket you connect it to. It is possible to connect a Bluetooth keyboard, but it's much better to use a wired keyboard to start with.

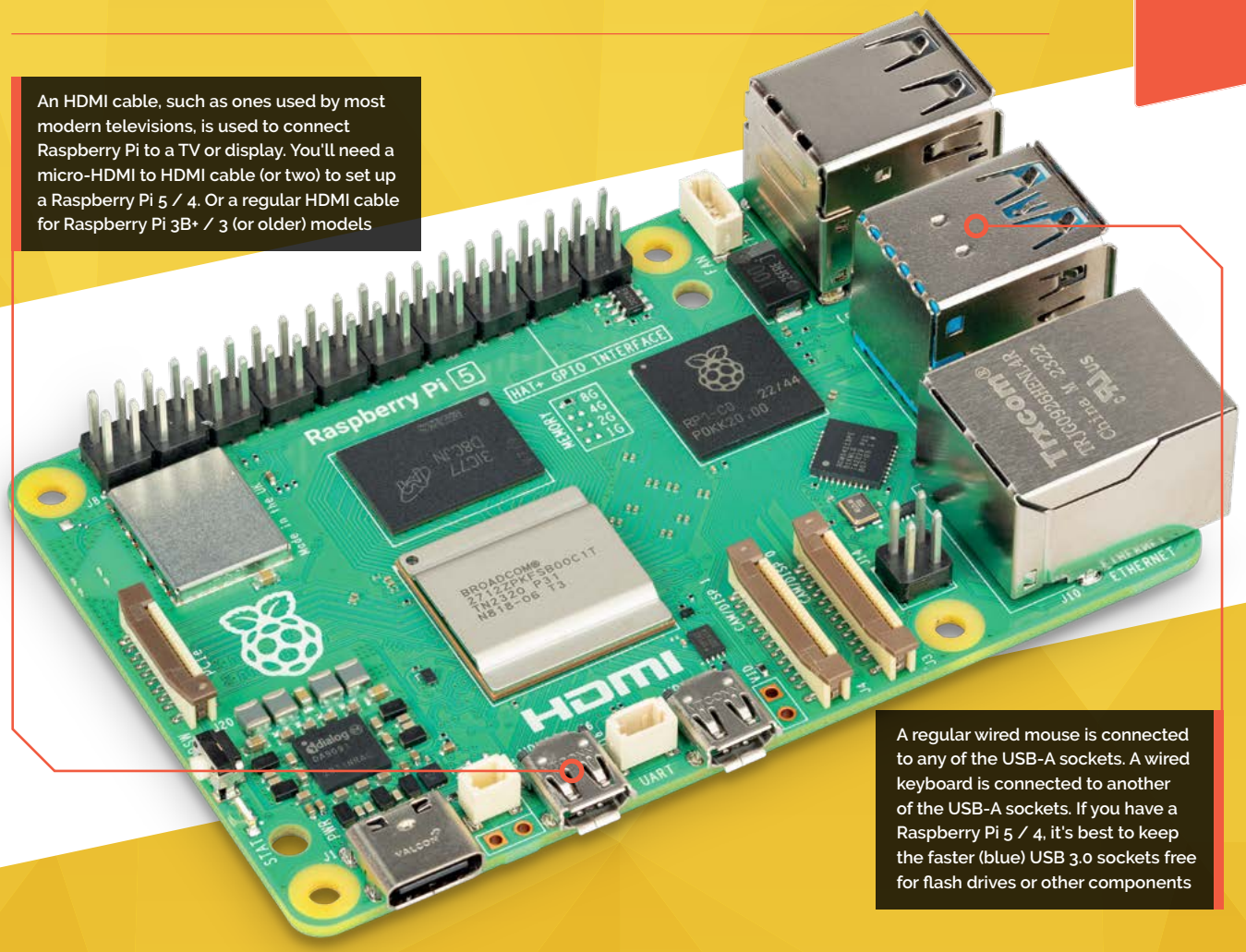
02 Connect a mouse

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

03 HDMI cable

Next, connect Raspberry Pi to your display using an HDMI cable. This will connect to one of the micro-HDMI sockets on the side of a Raspberry Pi 5 / 4, or full-size HDMI socket on a Raspberry Pi 3 / 3B+. Connect the other end of the HDMI cable to an HDMI monitor or television.

An HDMI cable, such as ones used by most modern televisions, is used to connect Raspberry Pi to a TV or display. You'll need a micro-HDMI to HDMI cable (or two) to set up a Raspberry Pi 5 / 4. Or a regular HDMI cable for Raspberry Pi 3B+ / 3 (or older) models



A regular wired mouse is connected to any of the USB-A sockets. A wired keyboard is connected to another of the USB-A sockets. If you have a Raspberry Pi 5 / 4, it's best to keep the faster (blue) USB 3.0 sockets free for flash drives or other components



SET UP RASPBERRY PI 400

Raspberry Pi 400 has its own keyboard – all you need to connect is the mouse and power

01 Connect a mouse

Connect a wired USB mouse to the white USB connection on the rear of Raspberry Pi 400. The two blue USB 3.0 connectors are faster and best reserved for external drives and other equipment that need the speed.

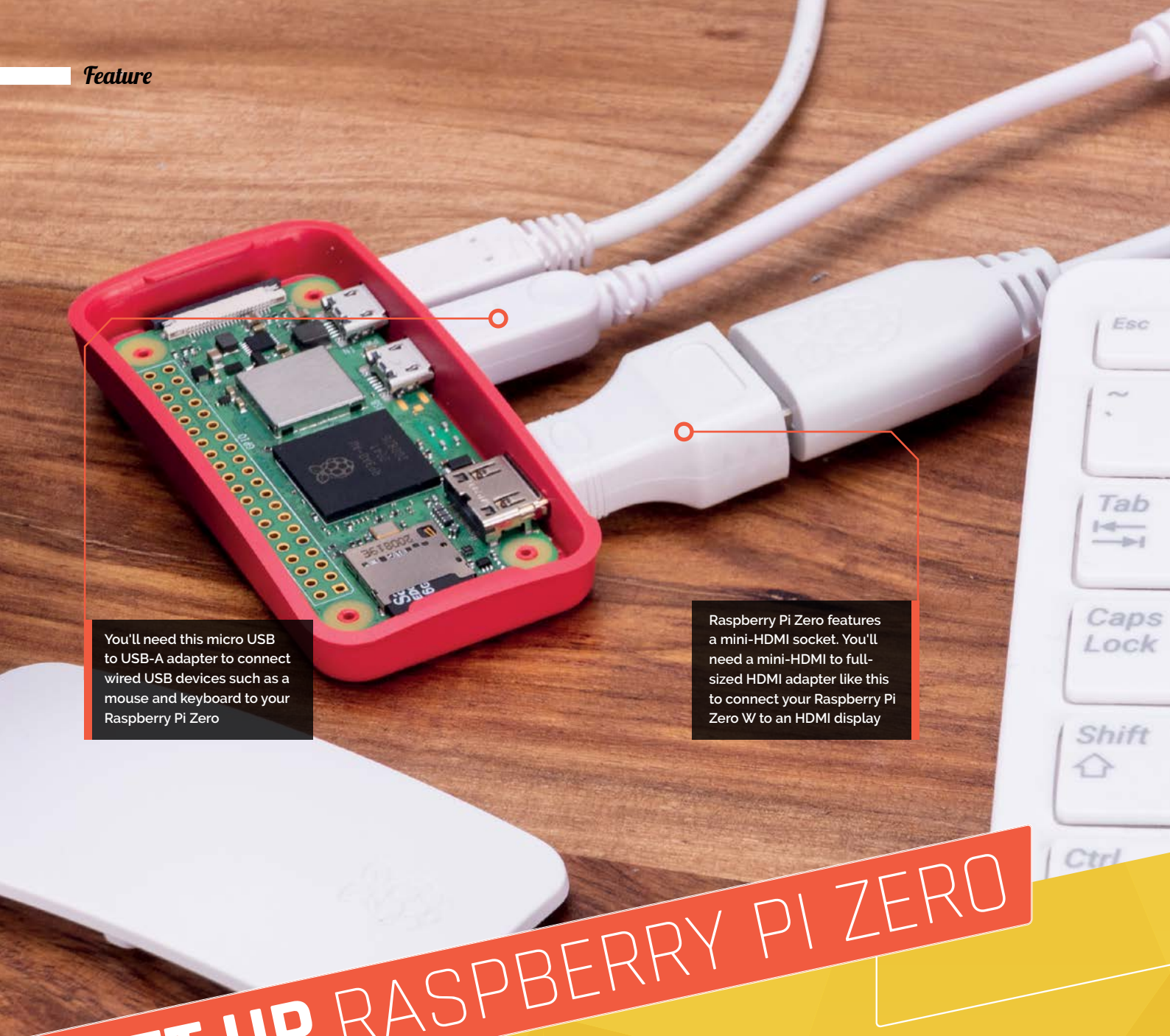
02 Attach the micro-HDMI cable

Next, connect a micro-HDMI cable to one of the micro-HDMI sockets on the rear of Raspberry Pi 400. The one next to the microSD card slot is the first one, but either connection should work. Connect the other end of the HDMI cable to an HDMI monitor or television.

03 The microSD

If you purchased a Raspberry Pi 400 Personal Computer Kit, the microSD card will come with Raspberry Pi OS pre-installed. All you need to do is connect the power and follow the welcome instructions. If you do not have a Raspberry Pi OS pre-installed microSD card, follow the instructions later in 'Set up the software'.





SET UP RASPBERRY PI ZERO

You'll need a couple of adapters to set up Raspberry Pi Zero / Zero 2 W

01 Get it connected

If you're setting up a smaller Raspberry Pi Zero, you'll need to use a micro USB to USB-A adapter cable to connect the keyboard to the smaller connection on the board. Raspberry Pi Zero models only have a single micro USB port for connecting devices, which means you'll need to either get a small USB hub or use an all-in-one mouse and keyboard.

02 Mouse and keyboard

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

03 More connections

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

First, insert your microSD card into Raspberry Pi

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

SET UP THE SOFTWARE

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

Now you've got all the pieces together, it's time to install an operating system on your Raspberry Pi so you can start using it. Raspberry Pi OS is the official software for Raspberry Pi, and the easiest way to set it up on your Raspberry Pi is to use Raspberry Pi Imager. See the 'You'll Need' box and get your kit together.



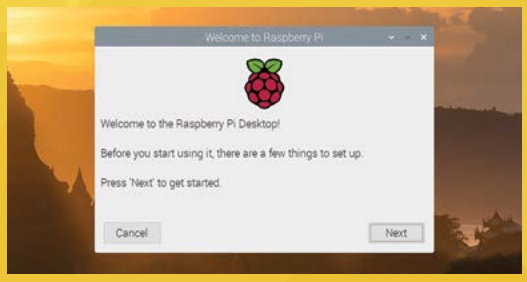
01 Download Imager

Raspberry Pi Imager is available for Windows, Mac, and Linux computers. You can also install it on Raspberry Pi computers, to make more microSD cards once you are up-and-running. Open a web browser on your computer and visit **magpi.cc/imager**. Once installed, open Imager and plug your microSD card into your computer.



02 Choose OS

Click on 'Choose OS' in Raspberry Pi Imager and select the recommended Raspberry Pi OS. Click 'Choose SD card' and select the microSD card you just inserted (it should say 8GB or the size of the card next to it). Click on 'Write'. Your computer will take a few minutes to download the Raspberry Pi OS files, copy them to the microSD card, and verify that the data has been copied correctly.



03 Set up Raspberry Pi

When Raspberry Pi Imager has finished verifying the software, you will get a notification window. Remove the microSD card and put it in your Raspberry Pi. Plug in your Raspberry Pi power supply and, after a few seconds, a blue screen will appear with 'Resizing Filesystem'. It will quickly vanish and be replaced by 'Welcome to Raspberry Pi'. Click on Next and follow the on-screen instructions to set up Raspberry Pi OS and start using your new computer.

Top Tip

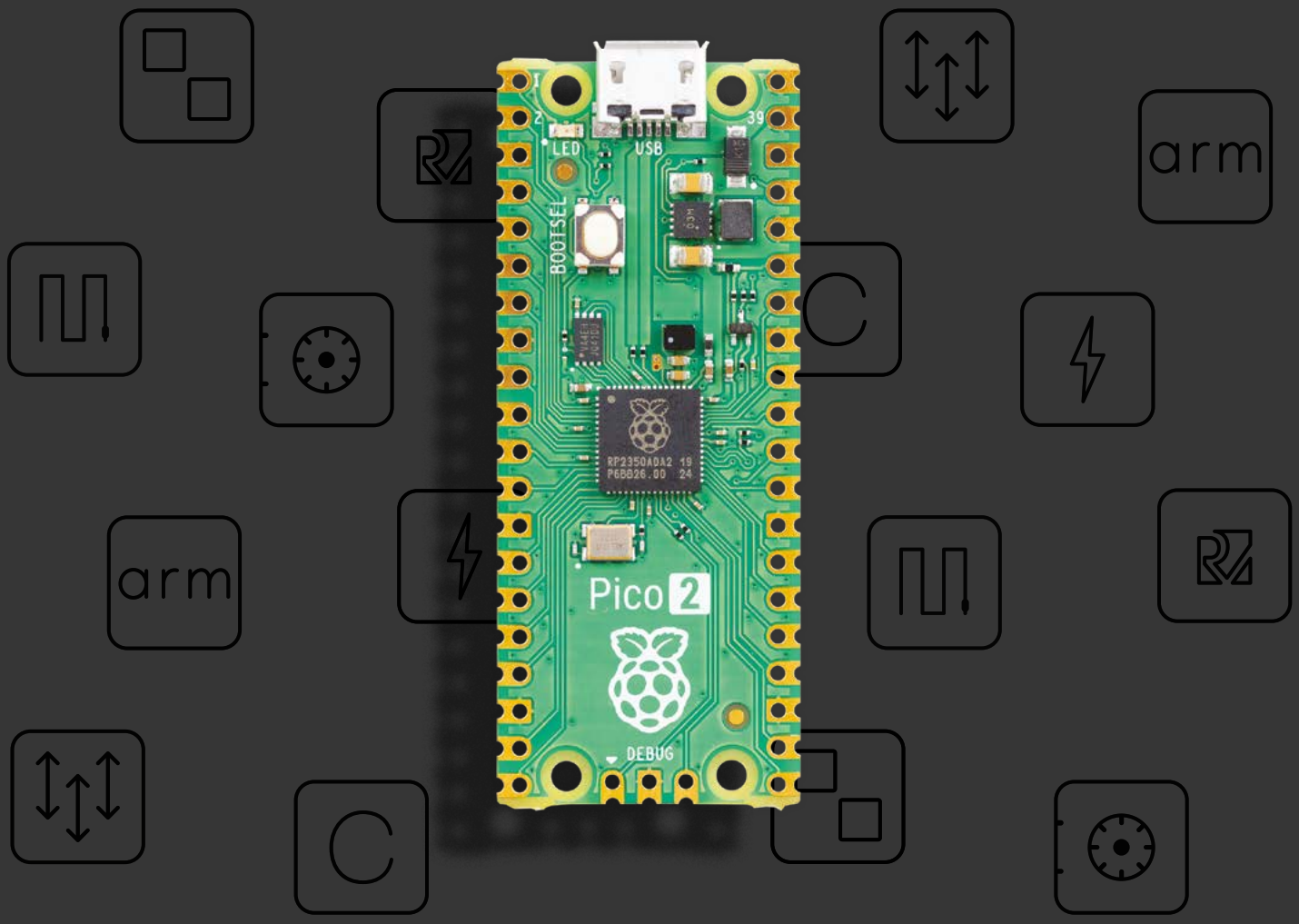
Advanced options

Click the cog to customise Raspberry Pi OS with options such as Wi-Fi passwords and usernames.

You'll Need

- A Windows/Linux PC or Apple Mac computer
- A microSD card (8GB or larger)
- A microSD to USB adapter (or a microSD to SD adapter and SD card slot on your computer)
- Raspberry Pi Imager **magpi.cc/imager**

Pico 2



Go behind the scenes of Raspberry Pi's tiny new device

Andrew Gregory

> Faster processors, more memory, greater power efficiency, and some industry-leading security features

It's here: the second generation of Raspberry Pi's Pico microcontroller board was released to an unsuspecting world on 8 August 2024. And what an upgrade it is: faster processors, more memory, more ways to get data in and out, greater power efficiency, and some industry-leading security features. We've been talking to the engineers behind this miniature marvel to find out how and why they made this device, and we've got a small insight in to the massive amounts of know-how it takes to produce an industry leading microcontroller board. Spoiler alert: it's a load of hard work.

What is a microcontroller board?

A microcontroller is a chip that contains a processor, interfacing options to connect it to other components, and some amount of RAM, to store the instructions that it needs to boot up properly. The difference between a microcontroller and a computer is that you can do multiple things on a computer; send emails, run graphics programs, browse the web, etc. With a microcontroller you program it to do one thing, and it does that one thing for the rest of its life — or until you program it to do something different. Microcontrollers are in every modern appliance, where a computer would be overkill, but there needs to be some form of programmed control.

The RP2350 microcontroller is at the heart of the Pico; take that chip, add some power management, flash memory, GPIO Pins and a USB port, and you've got a microcontroller board: an easy, affordable way to dip your toe into programming devices.

Pico 2: the same, but better



What's new in Pico 2

James Adams, CTO (Hardware), Raspberry Pi

At first glance, not much has changed, according to James Adams, CTO (Hardware) at Raspberry Pi. "It's got the same I/O… it's using the same form factor. It looks almost the same; the biggest visual change is that the silkscreen now says 'Pico 2'. Apart from the new RP2350 chip, we wanted everything to be as similar as possible to provide a drop-in experience for the user.

"Fundamentally, it's going to be a very similar experience to Pico, where developers will download the SDK and it will all look very, very similar in terms of building software, compiling software. Users will still plug the USB in and drag and drop their UF2 files on to it.

"There are two interesting things from the circuit perspective: one is that we've doubled the Flash memory, so you've got 4MB on Pico 2 vs 2MB on the original Pico. The other is to do with the way that power is supplied to the chip.

"RP2040 has an onboard linear regulator that takes 3.3V and regulates it down to 1.1V for the processor core. RP2350 has a switch-mode power supply on board, which does the same thing, but is more efficient.

"One of the highlights of the RP2350 chip are the faster M33 processor cores, as opposed to the M0+. We also have RISC-V cores in there – that's the Hazard 3 as created by Luke Wren, who's one of our employees.

"We've got two M33 processor cores running at 150 megahertz, so they're roughly twice as fast as RP2040, though it will depend on your use case.

"You've got twice as much SRAM, which is going to help with any general stuff, but especially with running things that are quite SRAM heavy, like Micropython. The PIOs have had various improvements based on what we've learned from RP2040, to be more efficient and allow for more use cases. And you've got an extra one of them in RP2350.

"This is the chip we always wanted to build. RP2040 was the pathfinder to this chip. This is what we would have built as RP2040 if we'd had the capability – your second chip is always better, because you learn from the first one, and RP2350 on Pico 2 represents that. It's faster, it's got more peripherals. It's got the improved power states and security that were lacking in RP2040 and the original Pico. And so we're super pleased that we've now put together the chip that we always wanted to build in RP2040, but we had to cut a lot of features out because they take a long time to develop. It's the chip we always wanted, in the Pico we always wanted."

Pico 2 hardware specification

Form factor
21 mm × 51 mm

CPU
Dual Arm Cortex-M33 or dual RISC-V Hazard3 processors @ 150MHz

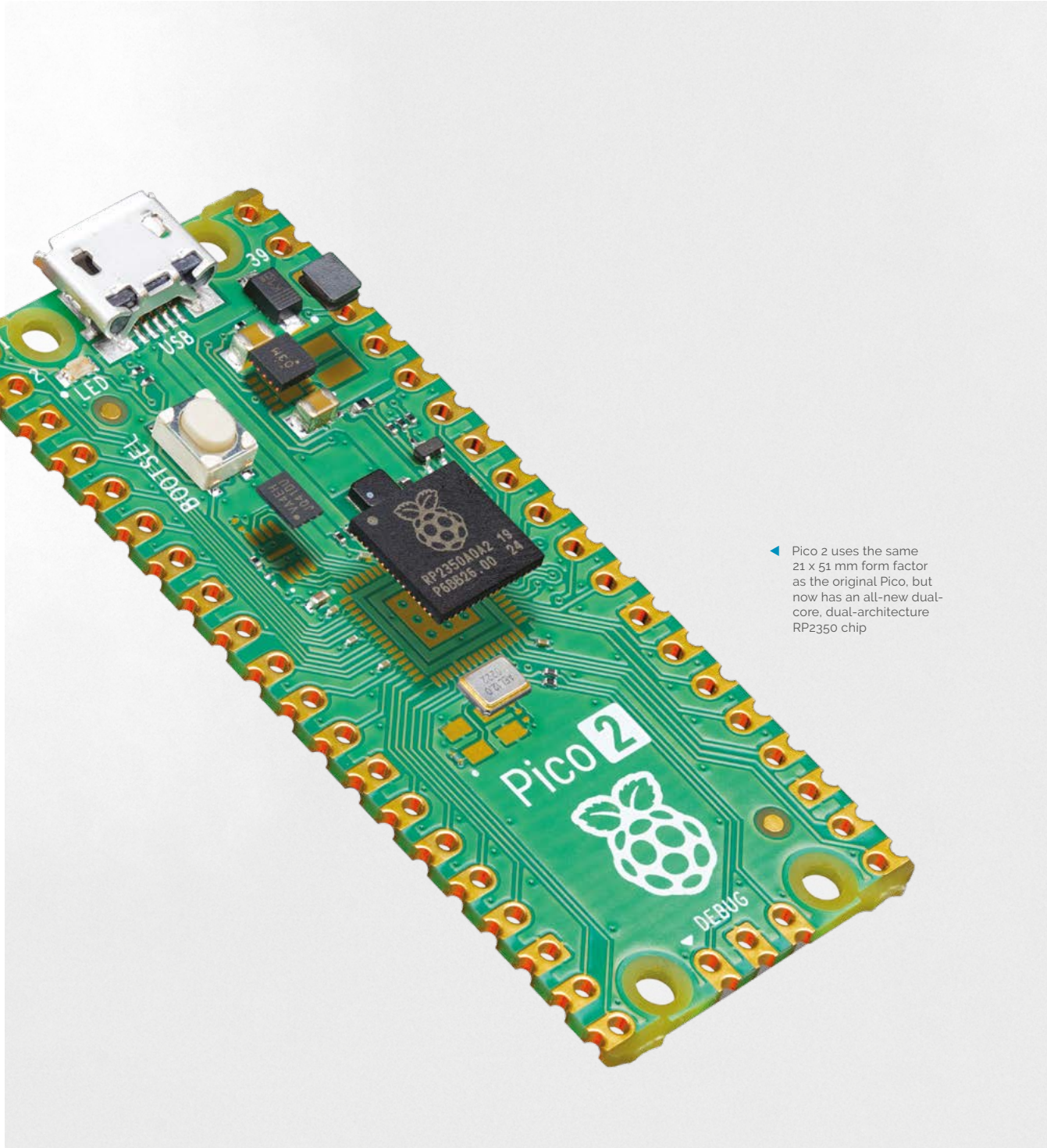
Memory
520 KB on-chip SRAM;
4 MB on-board QSPI flash

Interfacing
26 multi-purpose GPIO pins, including four that can be used for ADC

Peripherals
- 2 × UART
- 2 × SPI controllers
- 2 × I2C controllers
- 24 × PWM channels
- 1 × USB 1.1 controller and PHY, with host and device support
- 12 × PIO state machines

Input power
1.8–5.5V DC

Operating temperature
-20°C to +85°C



Pico 2 uses the same 21 x 51 mm form factor as the original Pico, but now has an all-new dual-core, dual-architecture RP2350 chip



RP2350

As we've heard from James Adams, the Pico 2 is physically virtually indistinguishable from the original Pico. What makes the difference is the brains of the board: the RP2350 chip.

> We can get down to less than a tenth of the power used by RP2040 in its low power state



RP2350 changes

Nick Francis,
ASIC Technical Director

"There are quite a few changes from RP2040 to RP2350. The main ones are security, higher performance through dual Cortex-M33 processors running at 150MHz, more memory and new OTP, and lower power modes.

On RP2040, in our low-power state, we'd get down to about 180 microamps in our dormant state with all clocks off. On RP2350 we can switch off power to a lot of the chip and get down to less than a tenth of that, which is going to be more useful for low-power applications, for example, where a device might spend a long time just waiting for input from a sensor.

"We've gone from a regulator to a switcher. Regulators are inherently not as efficient under load. So we've changed our core supply regulator – it's actually a combination device, so it has a small regulator for low-power modes up to 1 milliamp, when most of the chip is powered down and we're waiting to wake up. And then when we turn the core on we flip to a switch mode power supply which can provide up to 200 milliamps – more than RP2040, as the chip is bigger so has a higher peak current. The switcher is more efficient than the regulator used on RP2040.

"The PIO is still there; with some enhancements to that. We now have a third PIO block in there, so another four state machines. And we're still using the TSMC 40nm process. Re-using a process that you know saves a lot of time. When you're going to a new process node, you have to go and get a load of new IP and have to learn how to use it all.

So sticking to the same process meant that we can reuse some of this IP and knowledge from what we did on RP2040 and also RP1 [Raspberry Pi's first in-house chip, which was used on the Raspberry Pi 5]. So we've reused cell libraries, memories, ADC, PLL and USB PHY from RP2040.

So there's quite a lot that is the same, and we can just drop it in fairly easily and can then focus on changing the other logic and adding any new IP like the OTP or core supply regulator. We did review and update parts of the existing design, but there's an awful lot of new design in RP2350, and some higher level features like power changes or security touched a lot of the chip and has a wide impact on design and verification."



Software

Graham Sanderson,
Principal Software Engineer

"When you power up the chip, you have to run some software, but the program that the user installs, their firmware is stored in flash, so you have to run some code to be able to read flash before you can do anything else. That code is part of the boot ROM, named because it runs at boot, and it's stored in ROM.

"On RP2040 the boot path is fairly simple – there is a program in flash, you go look for it, and then you run it. The rest of the boot ROM space is taken up by things like floating point math support, a variety of other useful runtime APIs, and of course the UF2 bootloader that enables the user to drag and drop programs onto the Pico, mounted as a USB drive, and make them run.

"The RP2350 boot path supports a bunch of new functionality, support for RISC-V as well as Arm processors, and particularly completely new support for secure boot on Arm. This requires us to verify that the program stored in flash is trusted to run on the RP2350, by verifying a cryptographic signature. Additionally, we have hardened the boot code with the goal of making it impossible to run any user code that is not correctly signed, even in the hands of an attacker.

"The RP2350 boot ROM also supports dividing the flash into multiple partitions so that you can keep multiple copies of a binary, or keep shared data or resources separate from the main program. Our focus, as ever, has to make things powerful yet simple; therefore you can set up secure boot and have two (A/B) partitions, but still just drag and drop a UF2 to update your software. Dropping the UF2 will automatically target the partition that isn't currently in use, before switching at the next boot, thus avoiding situations where you have only half written the program. If your new program version is not correctly signed, the old version will continue to boot. Support for A/B partitions makes it much easier for user code to over-the-air updates, for example, reading a new version of itself from a web server, but now it can write that new copy to an unused area of flash, rather than worrying about updating the part of flash it itself is running from!

"Let's not forget about the Raspberry Pi Pico SDK either. This has had a lot of enhancements, bug fixes and new features, and of course, now supports both RP2040 and RP2350, and both Arm and RISC-V. Nonetheless, most people should just only need to recompile their RP2040 program for RP2350 with minor, if any, changes. I can't wait to see what people do with the new chip."

Where does the name come from?

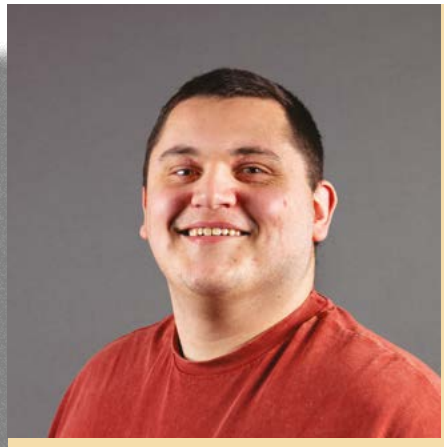
R P 2 3 5 0

- **R P** – RP stands for Raspberry Pi – obviously!
- **2** – The first number, 2, is the number of cores.
- **3** – The second number, 3, is loosely the type of processor – on RP2040 it was an M0+; on RP2350 it's an ARM Cortex-M33.
- **5** – The 5 is the log to base 2 of the RAM divided by 16 kilobytes. RP2350 has 520kb of memory, essentially double what we had on 2040, so that goes from 4 to 5.
- **0** – The last zero is the size of the on-chip non-volatile storage. On the RP2350 there isn't any – however, in the near future there will be variants on the market with on-board storage, and you'll see that reflected in the chip names.

RISC-V

RISC-V is an open-source chip instruction set. The first RISC-V chips were released in 2014, and since then it's been steadily growing in popularity, particularly in academic circles – the open source nature of RISC-V makes it a perfect place to start if you want to study how processors work.

The Hazard 3 RISC-V cores on the RP2350 were designed by Raspberry Pi's own Luke Wren in his spare time – and as they're open source, you can download the design files yourself and start poking around in the very same chip that will eventually be in use on millions of units out in the wild. As Eben Upton puts it: "In adding Hazard3 to RP2350, we're aiming to give software developers a chance to experiment with the RISC-V architecture in a stable, well-supported environment, and to popularise Hazard3 as a clean, open core, suitable for verbatim use in other devices, or as a basis for further development."



Luke Wren

Principal Engineer, Chip Team

"I've been doing logic design in my spare time since I was a student. It's highly addictive, and I think it's more accurate to say I'm a hobbyist who works in chip design than a chip designer with a hobby! It's an open-source processor design that anyone can put in their chip and use to run RISC-V code anywhere. You can also run it on an FPGA board, or run the simulator on your own machine. It's all built using open-source tools like yosys, nextpnr and gtkwave.

"The best way to get started is to get an FPGA board and just get hacking. Writing RTL [register transfer level] is a bit mind-bending at first – you can think of it like a C program where all of the statements execute at once, rather than sequentially – but that kick of seeing your own hardware come to life keeps you going. Start by blinking an LED, and keep going.

"Hazard3 is 100% my own design. It's a fork of Hazard5, the processor I designed for RISCBoy, my open-source competitor to the Game Boy Advance. Hazard5 is a five-stage pipeline – therefore having many hazards: data flow, control flow and structural – and a hazard is also a kind of 'risk', like the instruction set.

"Hazard5 was meant to run at the highest possible frequency on an iCE40 FPGA, so I could run the RISCBoy graphics core at a higher frequency too. Hazard3 on the other hand is a production-grade processor which delivers as much performance as possible in its small area envelope and within the range of frequencies I expect to see on microcontroller designs. It's a productionised version of Hazard5 with a shorter pipeline, hardware debug, and some security and memory protection features that people expect in real systems.

"From forking Hazard5 to having Hazard3 running CoreMark took less than a week. From that point until the first RP2350 tapeout was around two years, working on it on-and-off throughout. There is still ongoing maintenance work, and plans for future expansion – it will never be 'finished', just transition from development to stable releases.

"Before I started working on RISCBoy I had a project called Tarantula which was an eight-thread barrel processor implementing the Armv6-M instruction set, because that was the ISA I was most familiar with at the time, having written some Assembly during a summer internship. I abandoned the project because I realised I would never be able to share it with anybody, and I don't think I even have that source code any more.

"That experience changed how I looked at things from that point forward. When I decided I wanted to build a games console from scratch, including the processor, I looked around the instruction sets available at this point, this was around 2018, and there were a few interesting ones – Hitachi SuperH had just become much less legally restrictive – but RISC-V stood out as an instruction set where I could implement it fairly easily.

"The base instruction set is quite clean and simple, and you can add more complexity from a menu of extensions. I could share that with other people, and they could actually use it, and I could program using a real production-grade compiler like GCC or LLVM

"That was a long time ago, and RISC-V has come a long way since, both technically and as a community. There are other instruction sets that have become more open in the wake of RISC-V but I think it's clear where the momentum is. It's easy to criticise some of the technical decisions made in the base ISA – did we really need 31 link registers? – but the community is the most important thing in my eyes.

"I am excited about RISC-V because it lets you perform your mad-scientist architecture experiments on top of a clean and standard architecture. If you look at something like CHERI, which is a super-exciting development in the embedded security space, those folks have just gone and written a spec, and you can just go and implement it – no need to wait for it to be served on a plate.



The RP2350 has been designed to make it easy for third-party manufacturers to incorporate it into their products – such as the RP2350 Stamp by Solder Party

Security

James Adams, CTO (Hardware), Raspberry Pi

"What was missing from RP2040? Any kind of security, right? If you were building a product on top of RP2040, and you programmed your flash, there was no encryption in there – anyone could go and steal your firmware. If you build something on RP2040 there was no security whatsoever. And that was by design, because security is hard – we wanted to get a product out and see what people did with it.

"At the high level, you can do encrypted boot. You put encrypted firmware in the flash. It's got mitigations against the usual chip sort of fuzzing and glitching.

"It runs Arm's Trust Zone for microcontrollers. That's an interesting distinction – Arm always refers to Trust Zone, singular, but actually, there are two trust zones: there's the micro world trust zone, and then there's the big A-class Linux, Raspberry Pi stuff, and they're, both a little bit different.

"We run the microcontroller trust zone. You can hive off peripherals to be secure or insecure. So you basically have these two modes, you know, secure mode and insecure mode. And you have lots of different switches to allow you to partition the device into the secure bits and the non-secure bits. And that, along with, encryption, enables you to build a secure product.

"At a high level, you're kind of partitioning the chip into secure bits and insecure bits. And you can run secure bits of code and insecure bits of code, and they can talk to each other in prescribed ways that keep the whole thing secure. Just like any other microcontroller chip from STMicroelectronics and all these guys who have this security, the RP2350 does the same stuff, but we've also put quite a lot of effort into making it secure, right?

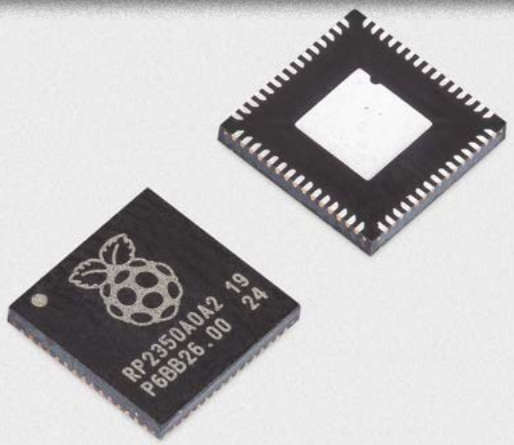
"[We've also been thinking] about how people could try and crack it, and putting mitigations in there. We have a special peripheral that means when the thing's actually booting from its boot code, it has to feed this peripheral with state [information] continuously. So as you're booting, you kind of capture the state, and you have to feed the peripheral with the right state, otherwise it resets the entire chip.

"What that means is, if someone's trying to glitch the chip, glitch the boot ROM to go and jump to a different set of instructions, that state becomes discontinuous, and the peripheral will immediately see that. Effectively, if you don't execute the instructions in the right way, it will immediately detect it and shut the chip down.

"And we've actually had a couple of consultants take a look at the chip and try and do all the standard glitching and resetting and fuzzing, and so far they've been unable to break this stuff.

"So that's one layer of mitigation; another one is glitch detection in the chip. So again, if it detects the voltage glitching, it will reset; we have a filter on the reset pin to try and stop people glitching it in the first place.

"There are several layers of mitigation for the security. And actually, it does seem, at least for the moment, that this is proving very difficult for the guys who tend to be able to crack these things to actually do it."



START YOUR

Raspberry Pi Adventure

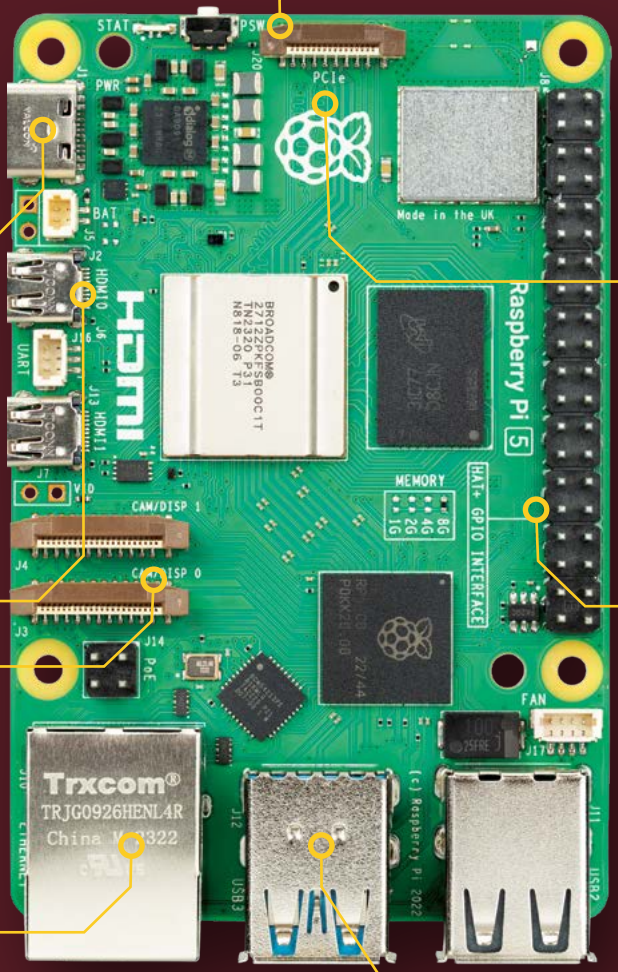
So you seek the power of Raspberry Pi, adventurer? Look no further, let us share ancient (and not-so-ancient) wisdom.

PJ Evans grabs his foam sword

We're not going to labour the Dungeons & Dragons metaphor (OK, maybe just a bit) but just like anyone setting out on an adventure, you may have little idea of how to start. If you did, it wouldn't be an adventure, and where's the fun in that? If you're completely new to Raspberry Pi, these pages will take you from the beginning and guide you through your first experiences with this little marvel. We'll cover using it as a desktop computer as well as the fun of physical computing and connecting your Raspberry Pi to the outside world. So, grab your knapsack of essentials (snacks) and let's go.

A tour of Raspberry Pi 5

So you've taken it out of the box. That's a good idea and a positive step forward, but what are you now looking at? Here are the essentials you need to know



Here is your USB-C power connector; without energy, this particular jewel is worthless

If you want to see what's going on, you'll need to connect a HDMI cable to either of these ports. You'll need a micro-HDMI to HDMI cable and yes, you can run two displays at 4K if you want!

These more delicate connectors allow you to add one or two Raspberry Pi cameras to your setup or the official 7-inch touchscreen

This small connector is a Raspberry Pi connector for PCIe, used for the powerful M.2 HAT+

SD cards act like a hard drive for your Raspberry Pi. You slot them in under here. The new on/off button sits alongside

The unique GPIO is one of the jewels in Raspberry Pi's crown. It provides a gateway to the outside world. Connect devices, LEDs, switches, sensors and more

Raspberry Pi 5 has great Wi-Fi, but you can also use wired Gigabit Ethernet for rock-solid network connectivity

Lots of USB ports, including super-fast USB 3.0 (the blue ones). This is where you connect your keyboard and mouse

Setting up your Raspberry Pi 5

Your first adventure is the most important! You can't take on quests without a working Raspberry Pi



MAKER

PJ Evans

When not on epic journeys and slaying dragons (actually befriending them), PJ is a writer, software engineer, tinkerer and level-15 wizard.

mastodon.social/@mrpjevans

You'll Need

- Another computer for writing the OS to a microSD card (16GB or more) **magpi.cc/microsdcard**
- Official power supply **magpi.cc/powersupply**
- USB keyboard and mouse **magpi.cc/keyboard magpi.cc/mouse**
- HDMI TV or monitor
- Micro-HDMI to HDMI cable **magpi.cc/microhdmi**

To the initiate, a Raspberry Pi can look a little intimidating. It lacks some of the friendliness of a traditional desktop. You've got a bare circuit board with little else, and that can seem a bit strange on your first encounter. Don't worry: everything can be explained and we can have you adventuring in no time. We'll get the software sorted out, the hardware plugged in, and get you on to the internet. Just remember to take your time, check everything twice, and learn as you go.

1 Acquire the spell book

OK, OK, we won't go overboard again on the D&D thing. What we mean is the first thing you need to do is prepare the operating system for your Raspberry Pi. An operating system (OS) is the foundation on which all applications run. But how to get it onto your Raspberry Pi? We need to get it written to an SD card that will then be inserted into the slot on the Raspberry Pi. Start by downloading Raspberry Pi Imager (**magpi.cc/imager**), which makes writing SD card images simple.

2 Incant the spell!

Sorry, we did it again. Never mind. Start Raspberry Pi Imager and make sure your microSD card is inserted into your computer, using an adapter if your machine does not have a microSD card slot. First, click on 'CHOOSE OS' and select the first option 'Raspberry Pi OS (64-bit)'. Now select 'CHOOSE STORAGE' and you should see your SD card (check the size to be sure). Now click 'Write'. You may need to enter your password at this point; that's normal. The image will be written to the microSD card and then verified to check everything went well. When prompted, you can remove the microSD card. We have a more in-depth guide to using Imager on page 40.

3 Assemble the hardware

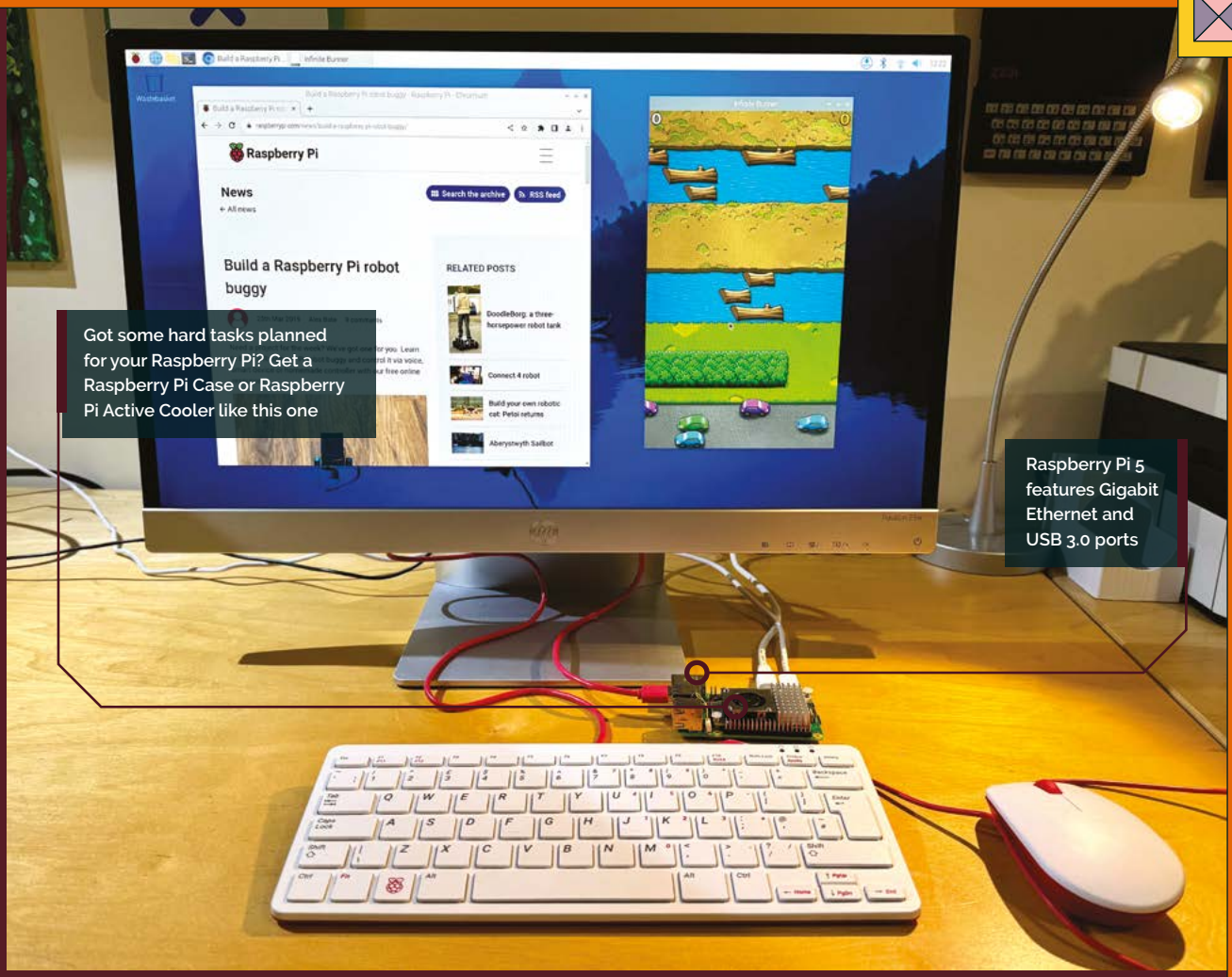
It's time to get everything plugged in. If you don't have a case, be very careful and only hold your Raspberry Pi by the edges. Do not place it on a metal surface or anything that might conduct electricity. Start by connecting the keyboard and mouse to the USB ports (it doesn't matter which). Now connect the micro-HDMI connector to either of the two available ports and the larger HDMI end to your monitor or TV. Finally, connect the unplugged USB-C power supply to your Raspberry Pi. Make sure your monitor is on and plug the power supply in.

4 Behold the life-giving spell!

Sorry, that's a slight overreaction. What you should see is a red & green flickering light on the Raspberry Pi as it comes to life. Your monitor should display some Raspberry Pi logos and a stream of text. Do not worry about this, it's nothing



▲ SD cards are very fragile. Insert the card gently into the slot with the gold contacts facing the underside of the board



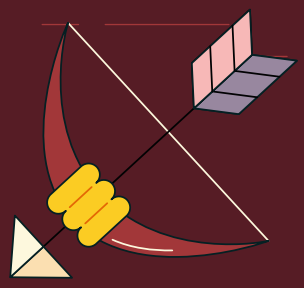
you need to read if all is well. The Raspberry Pi may boot once or twice during this initial period. It's setting everything up and ensuring that all the space on your SD card can be used. Shortly, you'll see the desktop appear for the first time. Congratulations, your Raspberry Pi is ready for use.

5 Wait, youngling! It must be fed!

Well, not fed exactly, but updated. Raspberry Pi software is constantly being updated, almost daily. Before doing anything, we recommend you ensure your software is absolutely on the latest version. To do this, watch out for a symbol in the top-right corner of the screen with a downward-pointing arrow. This is an indicator that software updates are available. Give it a click and follow the prompts to make sure the system is in tip-top shape. Whenever you see this symbol appear, we recommend you update as soon as possible.

6 Commune with the other beings

We mean networking. The best option is always wired. If that works for you, just connect an Ethernet cable between your Raspberry Pi and your internet router. Within a few seconds, you'll be good to go. If you prefer Wi-Fi, click the pair of arrows in the top-right of the screen and you can configure your connection there. You'll need your SSID (Wi-Fi network name) and the password. Once set, open Chromium (a bare-bones version of Chrome) and you can set sail across the internet. What now? Have a look at our side-quests!



Accessories

It's dangerous to go alone, take these!

As amazing as a Raspberry Pi 5 is, on its own, it won't do much. At the very least you need power. After that, there is a seemingly endless range of accessories so you can customise to your heart's content and build a perfect project. Considering its small size, you have a surprisingly wide range of connectors to make use of including USB 3.0, the versatile GPIO header, and for the first time, PCI Express. The choice can be overwhelming, so here are some of the most important accessories you should consider for your Raspberry Pi adventure.

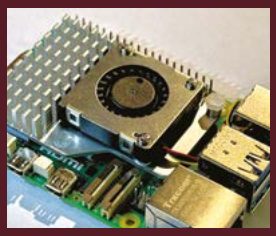


Protect your treasure



Your Raspberry Pi is going to need some protection from the elements, so a case should be your number one purchase. A change in the layout of Raspberry Pi 5 means that new cases are needed, with an official case available (magpi.cc/case) and both Pimoroni and Argon having their own versions as well.

Beware the dragon's breath!



Recent Raspberry Pi models run faster than their older siblings. A Raspberry Pi 5 can work quite safely without any kind of assisted cooling provided, but if you want to get the most speed from it, cooling is useful. There is now the powerful new Raspberry Pi Active Cooler.

Encode the ancient runes



Want to use your Raspberry Pi as a desktop machine? You're going to need a keyboard and mouse. Luckily, the vast majority of USB keyboards and mice will work instantly with Raspberry Pi 5 and you can even use wireless and Bluetooth devices with ease. The official pairing is a great choice: magpi.cc/keyboard.

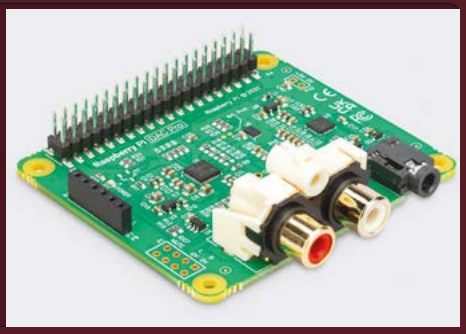
Wield the staff of power



Up for some gaming? Then get a decent controller. From inexpensive USB controllers to Bluetooth powerhouses, there's a wide range compatible with Raspberry Pi. We like 8BiTDo's range, which is perfect for retro gaming and the full AAA experience using services like Xbox Games Pass (magpi.cc/pro2).

Hear the songs of the people

Get a DAC or 'digital analogue converter'. Raspberry Pi has found a solid niche in performant audio playback. These add-ons (known as 'HATs') plug into the GPIO header and allow your Raspberry Pi to produce audiophile-quality audio. Pair with an OS like LibreELEC and you have your own media centre: magpi.cc/dacpro.



Illuminate the darkness

Raspberry Pi 5 can be more than a desktop computer – it can interact with the world around you by connecting electronics to the GPIO pins. Play with LEDs! If you feel a little unsure about this, then there are some great kits to get you started. We're huge fans of the CamJam series, starting with the EduKit #1: magpi.cc/edukit.



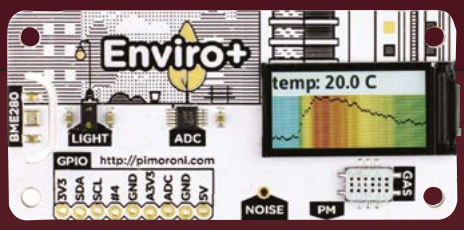
Capture souls

Whether it's setting up a webcam or building your own camera, photography has always been well supported by Raspberry Pi. The new model adds an extra slot for supporting multiple cameras. You can choose the HQ Camera board with mounts for different lenses, or the third generation of the standard Camera Module, now with auto-focus.



Don your wizard's hat

Everyone looks good in a hat, especially Raspberry Pi! HATs (Hardware Attached on Top) are boards that extend Raspberry Pi's functionality. If you can imagine it, there's probably a HAT that does it: displays, air monitoring, GPS, radio, climate, and more. A great one to start off with is Pimoroni's Enviro HAT for monitoring temperature and humidity: magpi.cc/enviro.



Enchant your memory

If you're looking for high-speed I/O and the microSD card is just not doing it for you, Raspberry Pi 5 is the first model to support the M.2 standard for SSD (solid-state drives) using its new Raspberry Pi interface for PCIe. The result is a significant increase in speed: magpi.cc/m2hatplus.



10 Side quests for the rookie adventurer

Looking for inspiration? You have found your treasure. Consider these 10 quests you can undertake to truly begin to master the Raspberry Pi. All are within your reach, noble traveller…

Learn the Desktop



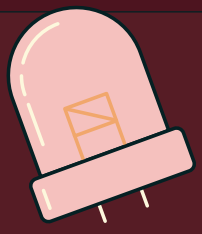
If you're completely new to Raspberry Pi, it's important to spend time learning the desktop environment. Raspberry Pi OS comes with some great software. Not only the usual office-type apps but programming environments, media players, and much more. You can add all of these from the Recommended Software app.

Write a game in Scratch

If you're new to programming, Scratch is the perfect introduction. It gives you a gentle introduction to the key concepts by using drag-and-drop 'blocks' to assemble code. It's a great learning tool, but most of all fun. It's already installed, so why not have a go at making a game! magpi.cc/gettingstartedscratch



Blink an LED



It's hard to explain just how satisfying this first project is. Making an LED blink on and off is commonly everyone's first Raspberry Pi GPIO project and a gateway to greater things. You'll need a few electronic components (see the CamJam EduKit #1 mentioned earlier), then follow the guide at magpi.cc/physcomputing.

Build a robot

For the more ambitious, let your Raspberry Pi hit the road by mounting it on a robotic chassis. Learn about motors, servos, and sensors. Robotics is a hobby that never stops growing. One day you could enter Pi Wars, the fun competition for Raspberry Pi robots (piwars.org). First, start here: magpi.cc/buildbuggy.



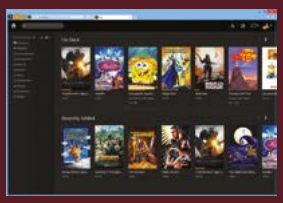
Retro gaming



As tiny as Raspberry Pi is, it is still an order of magnitude more powerful than its predecessors of the 1980s and '90s.
A thriving community has built up around emulating the classic consoles and home computers of that era. Check out the Recalbox project: recalbox.com.

Media centre

Why stop at music? Make your Raspberry Pi the centre of your media universe. LibreELEC (libreelec.tv) is a whole operating system based on Kodi. With it, your Raspberry Pi can be used to listen to music, view photos, and watch HD films. For streaming to various devices, check out Plex (plex.com).



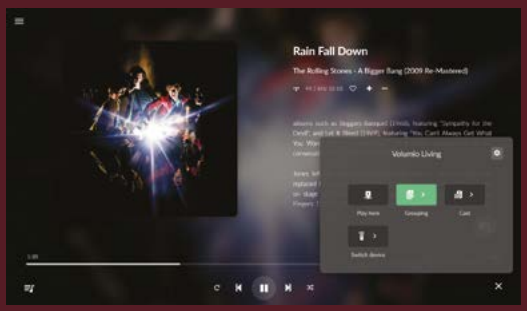
Home automation

Another popular use of our little computer is in home automation. The popular platform Home Assistant (home-assistant.io) is compatible with hundreds of smart home devices, including Philips Hue, Nest, Ring, and more. Create automations with scripting or build your own smart devices.



Audio player

If you want your Raspberry Pi to produce the best sounds around, get a DAC (digital to analogue converter) HAT to produce beautiful music.
Pair it with software like Volumio to get a web-controllable playback interface. Here are lots of other options: magpi.cc/mediaplayers.



Wildlife camera



Why not take a closer view of our feathered friends and other wildlife by setting up a Raspberry Pi wildlife camera? There are several ruggedised enclosures available that are suitable for life in the great outdoors. Add a camera and motion sensor and you're good to go: magpi.cc/wildlifekate.

Magic mirror



Another favourite of Raspberry Pi's community, the magic mirror takes a clear frame and adds a film of semi-transparent mirrored material. Place a screen behind it showing white-on-black images and the information looks to be hovering in thin air! Show calendar events, weather, news headlines, and more: magpi.cc/buildmagicmirror.

Ancient wisdom of the elders

We asked experienced Raspberry Pi users what they wished they had known when they bought their first one. Here's our favourites...

How to gain remote access

VNC allows you to see and control your desktop on a remote computer. You need to use SSH to connect your Raspberry Pi and enable VNC in Configuration. You can access Raspberry Pi OS desktop with TigerVNC (tigervnc.org). See The MagPi issue #136 (magpi.cc/136) for the in-detail solution!

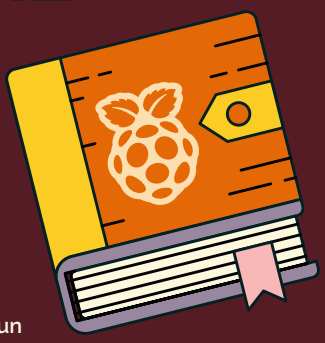
Back up your OS

You can use SD Card Copier in Raspberry Pi OS to backup your Raspberry Pi to a second SD card, external disk drive, or PC. Then reset Raspberry Pi OS whenever you want.



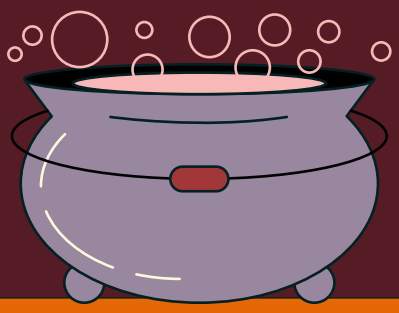
How to get a history of commands

When using the terminal, you often have to enter long commands. Press **TAB** to autocomplete commands and file names, use the **UP** arrow to cycle through your history; enter 'history' to view everything, and **CTRL+R** allows you to search history and rerun a command with one keypress.



Making is moreish

You won't believe how moreish making can be, how much time it can take up, and how many new friends you will make along the way. Yes, Raspberry Pi enthusiasm can be incurable.



Not to use a cheap power adapter

This is one we sadly hear again and again. Always use a certified power supply. A phone charger is not the same thing, even if it has the same connector.

WIZnet

Meet the new products

Powered by
Raspberry Pi

EVB-Pico2
Family boards

IoT Network Accelerator

ioNIC
Co-Processor



wiznet.io

For reselling & bulk purchases,
please contact direct@wiznet.io

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W



+



Subscriber benefits

- **FREE Delivery**
  Get it fast and for FREE
- **Exclusive Offers**
  Great gifts, offers, and discounts
- **Great Savings**
  Save up to 23% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- £10 (UK only)

Subscribe for 6 Months

- Free Pico W
- 6 issues of The MagPi
- £30 (UK) $43 (USA)
  €43 (EU) £45 (Rest of World)

Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

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

Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a six-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

WORTH **$6**



- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.





Buy now: magpi.cc/subscribe

SUBSCRIBE on app stores

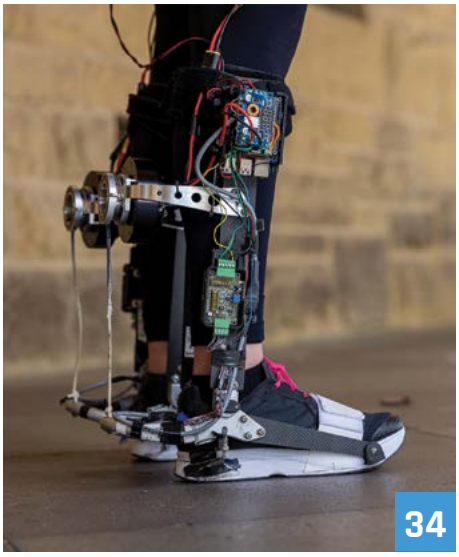
From **£2.29**

Available on the App Store

GET IT ON Google Play

Project Showcases







34 Exoskeleton
Raspberry Pi-powered walking

38 Team Pinball
These steel ball runs have a Pico twist

42 Odyssey Lights
Giant interactive lights at Blackpool

46 Tank driving simulator
An old trainer with new technology

50 Doom on a LEGO brick
A boomer shooter knee deep in the studs

52 Bugg.xyz acoustic monitoring
Conservationists take soundscapes of nature

56 Refurbished vending machine
Dispensing snacks thanks to Raspberry Pi

60 Antarctic Pi Cam
Scientific analysis from home to the pole

64 Olga the Fortune Teller
This AI-powered seer might reveal your future

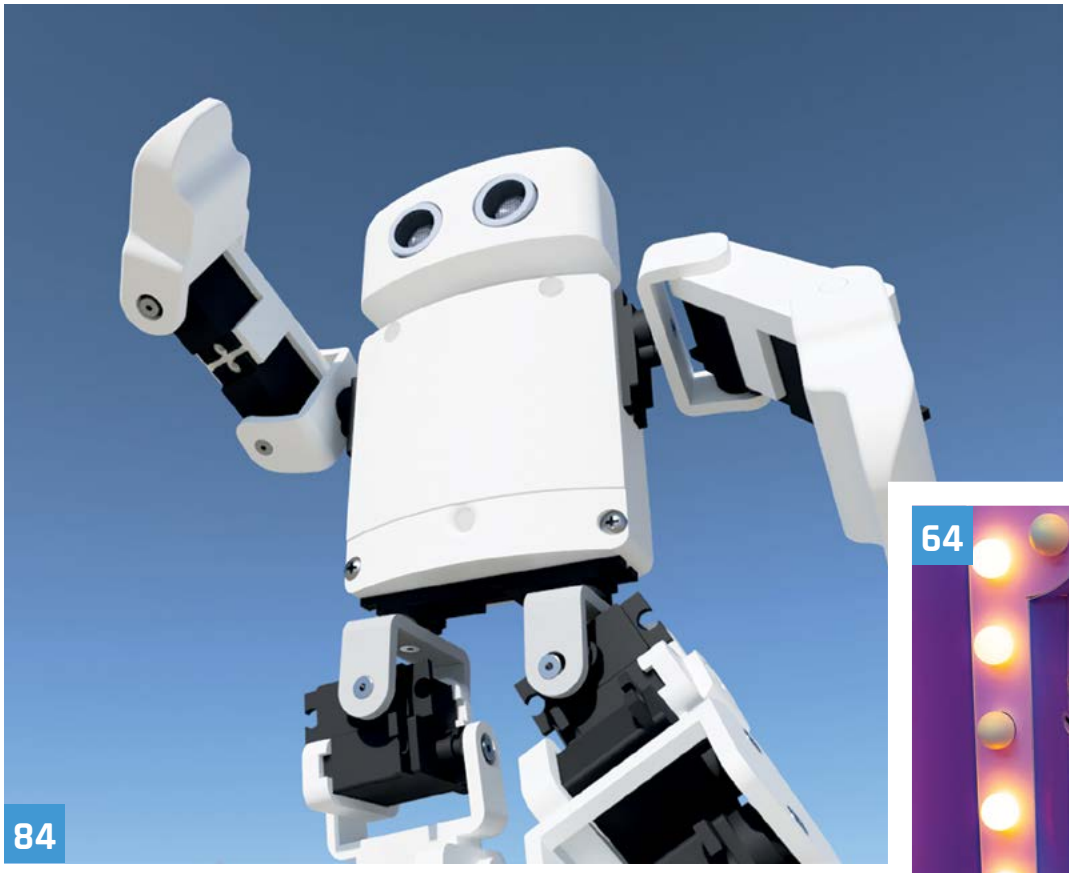
68 Mona Lisa Fluid Painting
Microfluidic biosensors meet Da Vinci

72 Pipstrelle bat detector
Using echo location in reverse to identify bats

76 Badgercam
Keep an unobtrusive eye out for badgers

80 Roktrack
Robotic lawn care in rural Japan

84 CHIP bipedal companion robot
This walking robot could be your pal





88 Bristol Braille Canute 360
A custom, multi-line Braille reader

92 ScreenDress
This 3D-printed dress has its eyes on you

94 Retro Pi World Radio
An old-school radio with a global tuner

98 BlueROV2 R4
Dive under the sea remotely and safely

100 CinePI XL
A high-quality Raspberry Pi cinema camera

104 Volcano-monitoring UV PiCam
A Raspberry Pi-powered volcanic alert system

108 PeggyBoard
A DIY smart climbing wall built in a garage

112 Mini Observatory
Greenwich observatory on your kitchen table



Exoskeleton

This groundbreaking device aids walking while understanding that people move in different ways, as **David Crookes** explains

MAKER

Stanford Biomechatronics Laboratory

Stanford Biomechatronics Laboratory develops wearable robots to improve efficiency, speed, and balance while walking and running.

magpi.cc/biomechstan

Raspberry Pi always has the potential to put a spring in our step but, in the case of this project, it's doing so in a literal sense. A research group at Stanford University has designed a robotic exoskeleton – an external frame that aims to support the body – with the aim of helping people with mobility impairments to walk and run more quickly. It's worn over the lower leg like a boot, and it has the potential to be revolutionary.

According to the team, exoskeletons of this nature, while proving effective in reducing the energy required to walk, tend to be expensive. They are also usually tethered which, as you can imagine, would be restrictive. The Stanford Biomechatronics Laboratory's exoskeleton, however, is neither of those things. And much of that is down to the incorporation of a Raspberry Pi 4 computer linked up to a host of sensors, a motor, and a motor driver.

"We selected Raspberry Pi specifically because it is low-cost, and quite capable for integrating many sensors and running multiple threads of code at the same time," Patrick Slade tells *The MagPi*. "The Raspberry Pi is also really nice because we can write code in Python, which is much easier to work with when validating a complex device than typical microcontrollers that might run Arduino or C."

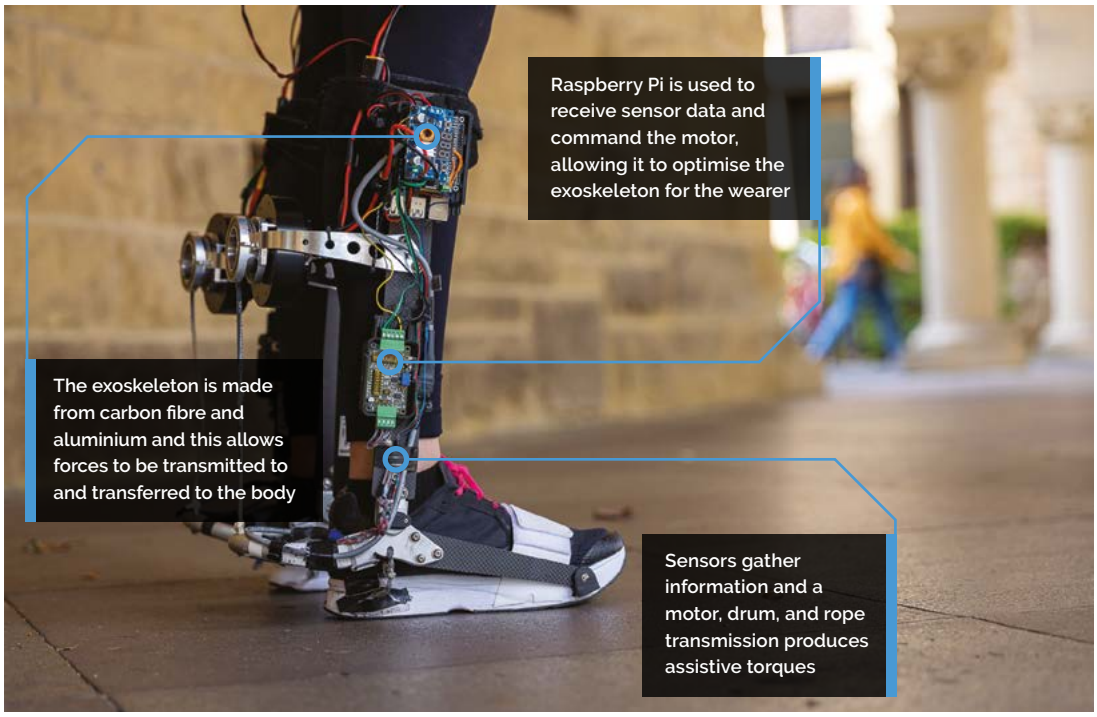
Gathering momentum

Patrick worked on the exoskeleton as a PhD student and he's now a postdoctoral scholar at Stanford. The project is led by associate professor of mechanical engineering Steve Collins and the work has, so far, performed well in extensive testing. The key to this exoskeleton is that it's personalised to suit different ways that humans walk. "It resulted in exceptional improvements in walking speed and energy economy," Steve says.

For many years, the team has been using exoskeleton emulators to train a machine-learning-based model. It has involved volunteers being hooked up to expensive devices so that motion and energy expenditure data can be collected and this has enabled the boot to work out the best way of assisting an individual.

> "We selected Raspberry Pi specifically because it is low-cost, and quite capable for integrating many sensors"

With the untethered exoskeleton, a wearer simply needs to walk around for about an hour in order for the system to figure out a pattern, then adjust and create a suitable aid. To do this, Raspberry Pi 4 reads sensor data and performs real-time control and optimisation at 200Hz, the

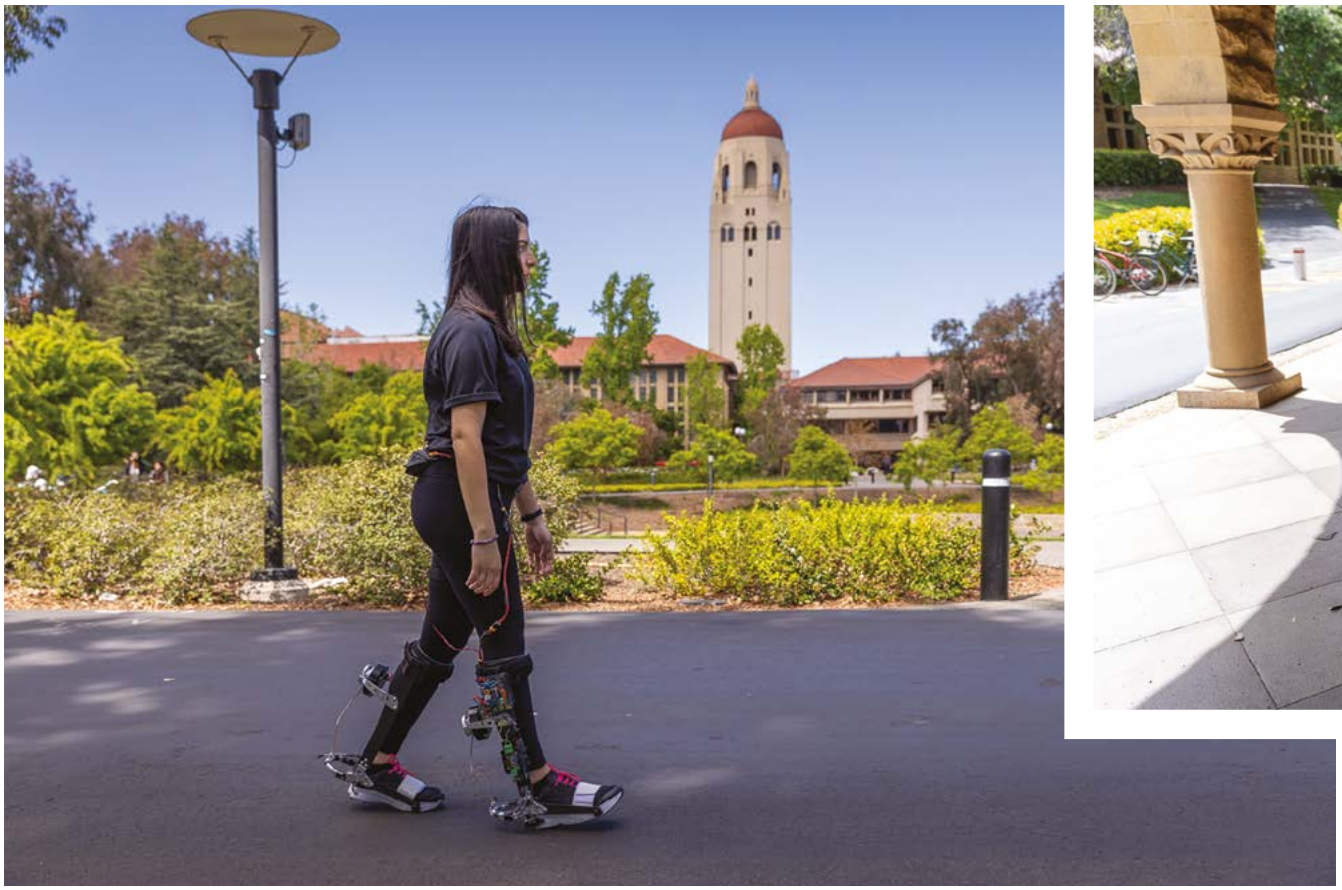




Quick **FACTS**

- It's a successful real-world exoskeleton
- The device will help people to walk
- Raspberry Pi is a key component...
- ...and it's helping to lower the costs
- Initial research suggests it's already successful

The exoskeleton has been extensively tested outside to see how much quicker people are able to walk when wearing it



One important aspect of the exoskeleton is that it is untethered and self-contained

It's a bulky device and the electronics are on view, but this is still very much a work in progress





You can read more about the project in a scientific article published in Nature, at **magpi.cc/exoarticle**

team says, and the benefits can be felt within the first 15 minutes.

Moving forward

So far, the results are heading in the right direction. People are not only walking nine percent faster, they are expending 17 percent less energy too. The researchers say the difference is equivalent to "taking off a 30-pound backpack." What's more, it's a greater improvement to walking than any other exoskeleton made before.

> "It resulted in exceptional improvements in walking speed and energy economy"

As you can imagine, this is sure to have major benefits for older people and those who are experiencing mobility decline. "It will allow people to lead more active, independent, and meaningful lives," says Patrick. Of course, much more work is needed and, to bring it to market, the research group will be seeking commercial partners, but the exoskeleton could soon be working alongside traditional aids such as wheelchairs and walkers at some point in the near future.

Perfecting a walk



01 First of all, wearers don the exoskeleton and place the battery pack on their waist. The battery weighs 0.3 kg and powers the device for 30 minutes.



02 When the wearer begins to walk, sensors record the ankle angle and velocity so that the assistive pattern can be analysed and personalised.



03 Torque is applied at the ankle and this, researchers say, replaces some of the function of the calf muscle. The device gives a push off just before the toes leave the ground.

Team Pinball

The thrill of pinball, plus the chance to plot a brand-new game, came together nicely with the help of Raspberry Pi. **Rosie Hattersley** hears how







MAKER

János Kiss, Otília Pasaréti, and Romain Fontaine

Team Pinball was founded in 2016 by pinball enthusiasts and open-source advocates János, Otília, and Romain.

teampinball.com

Team Pinball is the UK's only pinball design-and-build company. Based in Wales – pretty near to the Raspberry Pi factory, by coincidence rather than design – the trio behind Team Pinball come from Hungary and France, and are firm believers in all things open-source. Having seen the success of RetroPie gaming and how well it uses the capabilities of Raspberry Pi, the founders set about designing a new generation of pinball machines.

"Raspberry Pi is a powerful tool that has already found its place in the video game community with RetroPie. We wanted to do the same and create new, fun games with the Raspberry Pi," explains Romain Fontaine. When the team sat down to decide on the theme and title for their pinball machine, it became clear they were all fond of games from the 1980s era. Their retro game, The Mafia, "brings the player back to Chicago and the 1930s Prohibition Era with gangsters, casinos, and of course, a bank to rob!"

> "The pinball table brings the player back to Chicago and the 1930s Prohibition Era with gangsters, casinos, and a bank to rob"

Pi-eyed

Team Pinball originally designed their game for Raspberry Pi 2, noting its compact size and price and that it had "everything we needed", including being able to output audio, drive the screen, and control the machine through its GPIO," says Romain. Having been updated and upgraded in the five years since Mafia Pinball launched to some acclaim in the pinball enthusiast arena, The Mafia is now based on a Raspberry Pi 3 that runs game code on top of a modified Linux image optimised for fast boot and low latency. There are two separate sets of software. One runs the game logic that reads the switches located in the machine and drives the solenoids and light show; the other plays music and audio effects, and also renders the score and animations on the LCD screen.



Their friend Attila Szabo created the striking graphics, while Balint Ats's simulation helped check the gameplay worked as it should

Starting from scratch

Team Pinball designed the game and cabinet themselves, basing it on their own sketches, cutting playfields, wiring harnesses, and the electronic controller. The game's pinball layout is their own design too, while most of the pinball-specific parts, such as mechanisms, are standard and can be bought on pinball websites. As far as possible, parts were locally sourced: the wooden

Raspberry Pi's SPI bus powers the pinball machines' two lamps and two WS2811 RGB LED chains

Raspberry Pi 3B scans the switches every 3 ms, processes the game logic, and drives the solenoids through shift registers and IRL540 MOSFETs

Quick **FACTS**

- The Mafia Pinball game took two years to develop
- The first prototype machine cost around £10,000
- A virtual simulator was used to check the layout and gameplay
- Issues included metric-manufactured European parts and imperial pinball parts...
- ... and a 32-tonne truck delivering them all to their rural office



▲ Final assembly complete, Romain applies the Team Pinball logo





▲ Team Pinball commissioned Balint Ats to create the distinctive graphics and decals for the game

◀ Flippers, lights, and other standard pinball parts provide a familiar gaming experience

cabinet was made by a Welsh company, with printing and glass panels also done in Wales. "Our office is only 30 minutes away from Raspberry Pi's factory," says Romain, so they can also count that as a locally sourced part.

Team Pinball designed their own controller board, called Rboard, specifically for Raspberry Pi that is compatible with direct switches, as well as a switch matrix with up to 100 switch inputs. "It can drive more than 200 LEDs and 48 solenoids. This is more than previous pinball controllers such as the WPC, while using one tenth of the board space thanks to Raspberry Pi," says Romain. "Several revisions" to the Rboard added a watchdog circuit to protect the hardware, capacitors, "and a regulator to maintain a perfect 5V for Raspberry Pi." The latest version includes a DAC for improved audio quality. Video needed to be in H.264 format, so they had to find alternative rendering techniques.

Manufacturing was measured in a combination of metric for bespoke parts and imperial for standard pinball parts, which added to the challenge – not least because some of the manufacturers had never worked on a pinball machine before. When the 32-tonne lorry

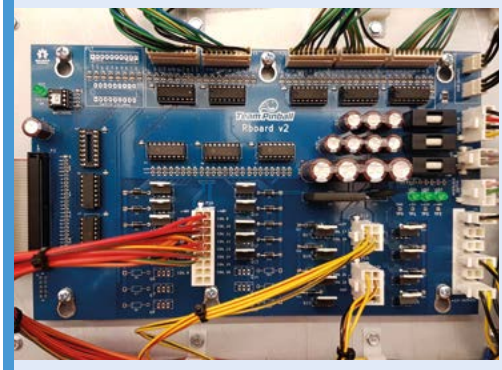


▲ With commissions to create machines for others, each machine is handmade by Team Pinball

delivering everything to Team Pinball's HQ arrived, turning down the narrow Welsh lane was a feat too far, so Romain and his colleagues had to carry the cabinets for the last part of their journey in the pouring rain. Nonetheless, everything was assembled and comprehensively checked before Mafia Pinball eventually made its much-anticipated debut.

The game has been a big hit in the pinball community, gaining numerous plaudits and orders. Their machines have shipped globally, to USA, Canada, Sweden, France, Austria, and even Australia. A Canadian company also uses the Team Pinball system for their game design. Better yet, Romain says Team Pinball "absolutely!" has more pinball machines in the pipeline. "More pinball projects, all powered with Raspberry Pi, CM4, and Raspberry Pi Pico!"

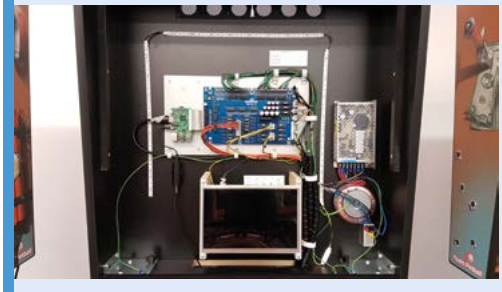
Get pinning



01 If you want to make your own pinball machine, there are plenty of options. Team Pinball's controller is open-source, and can be found on GitHub: **magpi.cc/rboardgit**. Romain also suggests **pinballmakers.com** as a good place to start.



02 A good knowledge of Python is useful for coding your own pinball machine. Alternatively, you could use code from **missionpinball.org** to either run software on an old pinball machine or for a brand-new one.



03 Low latency to ensure rapid flipper response is vital. Along with a lightweight Linux image, Team Pinball uses C/C++ "to communicate with the GPIO pins for the best performance."

Odyssey Lights

Blackpool's world-famous illuminations provided the setting for an interactive project that features Raspberry Pi. **Rosie Hattersley** gets enlightened



MAKER

Lorraine Underwood/ Jack Irving/ Sam Nugent / Lightworks NugentLightworks

Lorraine Underwood is a senior teaching associate and PhD student at University of Lancaster; Sam Nugent is a Computer Science student; Jack Irving is an installation artist; Lightworks make Blackpool Illuminations.

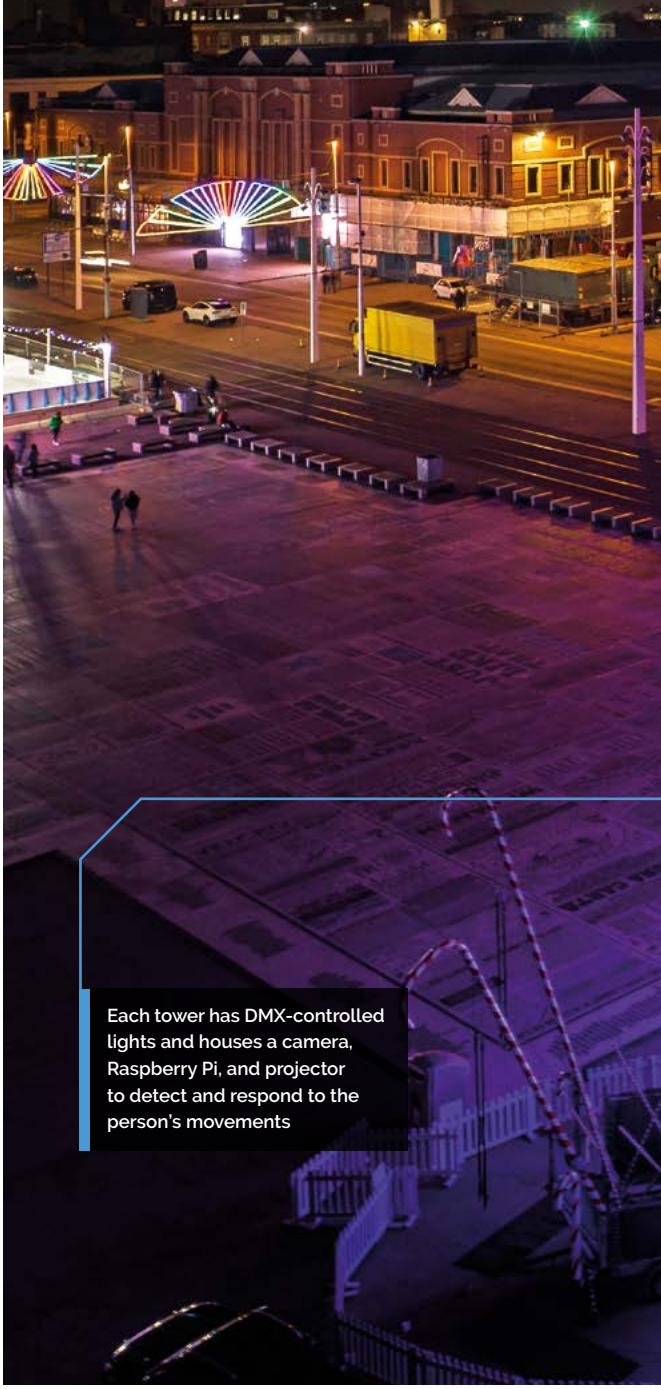
lorraine.mcunderwood.org

Wintertime illuminated walks and interactive art shows have proved extremely popular over the past few years. Blackpool's Odyssey lights combine the two, adding to the appeal of the renowned annual Blackpool Illuminations with impressive large-scale installations that respond to and even follow visitors as they explore the brightly lit site.

Odyssey is the brainchild of Lightworks, which creates and manages the existing Blackpool Illuminations attraction, Blackpool-born costume designer Jack Irving **(jackirving.co.uk)**, and Lancaster University's School of Computing and Communications. The research team there led by Joe Finney, Professor of Computer Science, includes Raspberry Pi enthusiast and *The MagPi* regular Lorraine Underwood, a Senior Teaching Associate of Creative and Applied Computing. They had worked with Lightworks before to produce interactive displays, "but this is our first proper huge-scale project together," she explains.

Enormous and ambitious

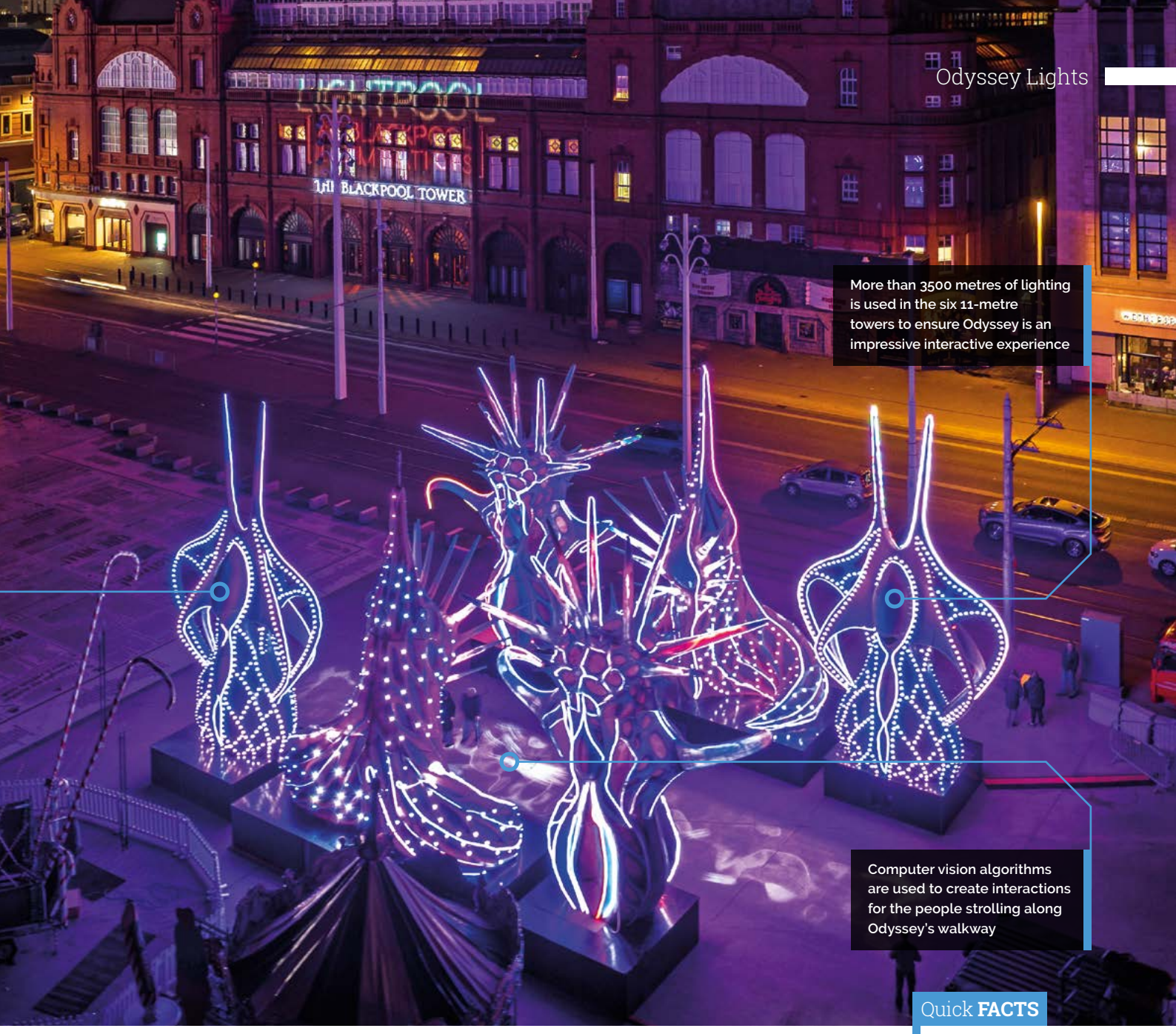
"All three organisations use, and create, cutting-edge technology for the public, and were keen to create something new and exciting for the public to enjoy," says Lorraine. Jack specialises in creative productions that fuse technology and fashion. Odyssey hoped to harness this to "engulf the world in the most fantastical spectacle." They began by co-designing and storyboarding, resulting in a vision for a series of 11-metre-tall towers featuring lasers, lights, speakers, and fog machines, each housing a projector and a camera. The six towers would project a ten-minute interactive show with different scenes containing alien sea creatures along an 18-metre path.



Each tower has DMX-controlled lights and houses a camera, Raspberry Pi, and projector to detect and respond to the person's movements

Cameras detect people along the path while each of the fantastical creatures behaves differently around them. "Some creatures would avoid people; some of them would follow you; others would swim together or appear when they detect a person nearby," says Lorraine.

"All the designs came from Jack Irving's unique style and imagination. It amazed me that



More than 3500 metres of lighting is used in the six 11-metre towers to ensure Odyssey is an impressive interactive experience

Computer vision algorithms are used to create interactions for the people strolling along Odyssey's walkway

Huge automated drills were used to carve each creature's shape from polystyrene blocks, before sculptural elements were refined



Quick **FACTS**

- Odyssey plans began in 2019, but were delayed by Covid
- Each tower houses a Panasonic 12,000 lumen projector, Raspberry Pi, and camera
- The interactive displays required 10,000 lines of code
- There are 2400 LED nodes and over 3500 metres of LED lighting
- All of the lights are individually controllable

▲ The sculptures tower above participants walking along the installation's 18-metre walkway



▶ Odyssey being assembled at the Lightworks factory

Lightworks were able to create his designs in fibreglass at such a height," she adds.

Technical challenges

"The Odyssey team needed a unique set of technologies for this project," explains Lorraine. "While the towers are huge, there's limited space inside them for equipment alongside all of the LED strips." They needed a small camera with strong video streaming capabilities, but it was also vital to minimise latency so the creatures would react in real-time to the audience.

Lorraine is the maker in the Lancaster University team and often uses Raspberry Pi in her projects, so she was confident in recommending it for the Odyssey. It transpired that Lightworks also knew Raspberry Pi well: some of the Blackpool Illuminations are controlled by Raspberry Pi too. Although accustomed to large-scale projects, testing the full Odyssey setup was challenging. They took over two lecture theatres at the University of Lancaster for the entire summer, but this had a ceiling only 3m high. The Lightworks building,

meanwhile, is tall enough to accommodate trucks, but the skylights weren't ideal for trialling something intended to be seen at night. "The cameras were calibrated using QR codes on the ground, which are difficult to see in sunshine." There was a final obstacle once Odyssey was unveiled. Several of the cameras fell victim to the "unforgiving" Blackpool wind and weather, necessitating a switch to USB cameras with sturdier protection. The results, however, are stunning.

> Wherever you look while standing inside Odyssey, something amazing is happening

"Every time you walk through Odyssey, different creatures appear and act differently," explains Lorraine. "We wanted to create an intense experience in every sense. The colossal 11-metre-high towers, the sound system, smoke machines, and lasers would complement the interactive projection show on the ground. Wherever you look while standing inside Odyssey, something amazing is happening."

The team insisted on making Odyssey adaptable – the setup can be easily configured to use fewer cameras and projectors – while plans are afoot to make it easy to load different graphics and scenes into the projection show, ensuring many more marvels to come.



The scale of the creatures becomes clear when laid out on the Lightworks factory floor

Interactive light show



01 There are six cameras with viewing angles that overlap the six projections. The cameras take photos continuously and feed images to computer vision algorithms which inform the content of the projection.



02 Since the idea is to respond to the participant, constantly updating the person's position and movements is critical to Odyssey appearing to mimic them. The calibration process was rather complex.



03 OpenCV homography matrices calibrate each camera to its adjacent projector. The calibration information is then translated between the camera coordinates and projector coordinates. Working out how it works was a challenge!

Tank Driving Simulator

A Swiss tank museum is home to an historic training vehicle that has been upgraded with a Raspberry Pi. **Rosie Hattersley** finds out why







MAKER

Gerold Handschin, Michael Salathé, & René Demarmels

The museum team, including electronics specialist and technician Gerold, were determined to salvage the 50-year-old FASIP.

magpi.cc/fasip

Although it has become known by its German moniker of FASIP (shorthand for Panzerfahrsimulator), the vehicle was actually developed in France in the 1970s, explains Gerold Handschin, one of the trio from the tank museum who set about saving the unlikely survivor. "A total of eight tracks were used to train prospective tank drivers, initially for the Centurion tank and later for the Panzer 68 and Leopard 2," Gerold relates. The trainee driver sat in a replica driver's cab, with a screen displaying the terrain he needed to negotiate. Hydraulics under the cab replicated the tank's movements, while a camera on a trolley travelled the length of the 12 m-long detailed terrain model. The combination of responsiveness to the driver's steering movements and on-screen display created a realistic driving experience that predated arcade games by several years. In addition to the large terrain model and hydraulic system, various computers were required to operate the system.

Ultimate upcyclers

In 2004, the Military Museum saved one of the FASIP systems in Thun from being scrapped, and rebuilt it at the museum site in the village of Full. But, after several years, various defects in the antiquated computers and in other electronic components had led to its apparent end. The Full museum's dedicated helpers, Gerold Handschin, Michael Salathé and René Demarmels, could not resign themselves to this. Instead, they began a thorough analysis of how the FASIP system worked, identified defects, and gradually replaced any components that failed. Their aim was to replace each item "as faithfully as possible."

Sadly, the original 1970s MITRA-125 computer which controlled almost every aspect, including responding to the displacement sensors and controlling the simulator's movements, lamp, and displays, was defective. This central control unit clearly needed to be replaced. "Fortunately, the program in operation on the MITRA-125 was available as a hard copy. So we decided to replace the old computer with a new system," says Gerold. The only issue: it was all in French. Scanning in the several-thousand-page document and using text recognition software helped, but in some cases the type had faded so much that the characters could not be recognised. "We sometimes had to approach the correct parameter values by trial and error," says Gerold, but new C-based simulator software for Raspberry Pi was eventually written.

Restoration challenges

The decision to use Raspberry Pi 3B+ as the new central control computer was down to its GPIO, its size, and the price. Raspberry Pi was fitted onto a new adapter board via its interface slot, replacing the interface board for the MITRA-125. The original power supply units were in poor condition and voltages were no longer stable, so modern switching power supplies were fitted. "The new adapter board contains multiplexers to connect all the signals needed for the Raspberry Pi GPIO," explains Gerold. It was also used to adjust the 3.3 V voltage of the GPIO to the 5 V TTL logic of the existing systems.



The FASIP tank simulator challenges the trainee driver to master eight different terrains, beamed to the cab from a camera trained on the scenic model layout

Originally running on custom software written for the MITRA-125 computer that powered it in the 1970s, FASIP now runs code written in C

This Raspberry Pi-controlled setup emulates the bumps and turns the driver encounters thanks to cameras and sensors that follow the tank simulator's route

Quick **FACTS**

- The team had four custom circuit boards made...
- ...just in case the tank simulator needed replacements in the future
- Overall, the project cost 820 Swiss francs, including spare parts
- The original instruction manual was in French
- So, they scanned in the text and used a translation tool

▲ View from inside the FASIP tank simulator control cab with live terrain updates as the driver navigates

▶ Sensors and camera hover above the test track, feeding back to the FASIP tank sim control deck





▲ The team looking over the scale model that the tank 'drives' around

> A realistic driving experience that predated arcade games

The IO plug-in unit's original boards, XERUDI and XUCI, were replaced by the new board and Raspberry Pi too, but it was a challenge to work out the correct timing when driving the components of the interface plug-in unit. The overhaul of the simulator was completed at the end of July 2020. Since then, it has been possible to drive the Panzer 68 simulator through the grounds at the Swiss Military Museum in Full by appointment.

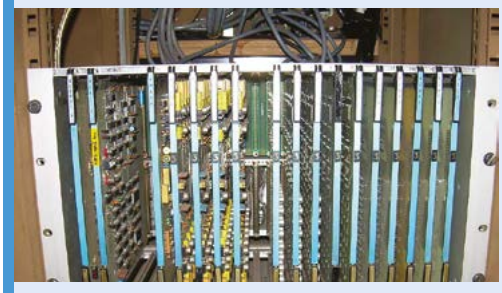
Tank training



01 The 1970s MITRA-125 custom code and hardware were replaced with a bespoke circuit board to control the cameras and respond to feedback from them and the track sensors.



02 The corroded old network and IOs needed to be replaced with components capable of delivering a constant voltage.



03 The original tank sim rig needed a complete overhaul, custom circuit board and adapters, plus reprogramming in C.

Doom on a LEGO brick

If you have an appreciation of both LEGO and 1990s video games, read on. **Nicola King** digs out her glasses to admire a minuscule make



MAKER

James Brown

James lives in Wellington, New Zealand, has a background in games and graphics programming, and creates interactive exhibitions.

magpi.cc/ancientj

We've seen portable games projects before, but nothing quite this small! Maker James Brown took on the huge challenge of embedding a playable game of the classic first-person shooter Doom in a LEGO brick, powered by Pico's RP2040 chip and featuring a tiny OLED screen.

James makes clear that it was an iterative process over many months. The first version (**magpi.cc/firstlegobrick**) was "very simple and not particularly well-planned. I figured I'd make a tiny computer, attach a 0.42″ OLED which just (only just!) fits in the face of a LEGO slope brick, and somehow encapsulate it in resin to look like the original. So, I drew up a PCB as big as I could go without it sticking out, populated it with the smallest microcontroller I could find in stock at JLCPCB (**jlcpcb.com**), and sent it off to be manufactured."

After the project garnered a positive response, he sought to make it accessible to more people. The question was, how could he create a version that could support a USB connection and be reprogrammed interactively? "RP2040 ticked all the boxes," he shares, "and (crucially!) was available."

3D construction

James ended up with a 3D PCB structure, taking what he'd learnt from the first iteration and improving it: "By soldering together a complete frame of PCBs, I get a very precisely machined brick shape, contacts for the battery, and a USB port tucked away inside."

The frame consists of four panels, with the front one containing the driving forces including the RP2040, a 4MB flash chip, crystal oscillator, and 0.42″ SPI OLED screen. "Four lugs on the front panel provide the physical mounting points," explains James, "as well as the electrical connections to the rest of the panels – two for power, two for data."

With 9V contact strips on the side panels, along with a voltage regulator and some diodes, the rear panel has just enough room for a micro USB socket. The other three panels have castellated edges to provide "the mechanical and electrical connection, and can be designed as a single flat circuit which is later chopped up to make the 3D frame."

Once James had assembled the circuit, he put it in a silicone mould to be encapsulated with epoxy resin. The biggest headache was getting the resin layer over the screen just right: "It needs to be very thin (around 0.1mm), both in order to fit in the space between the outer surface and the studs of the bricks below, but also to be clearly visible."

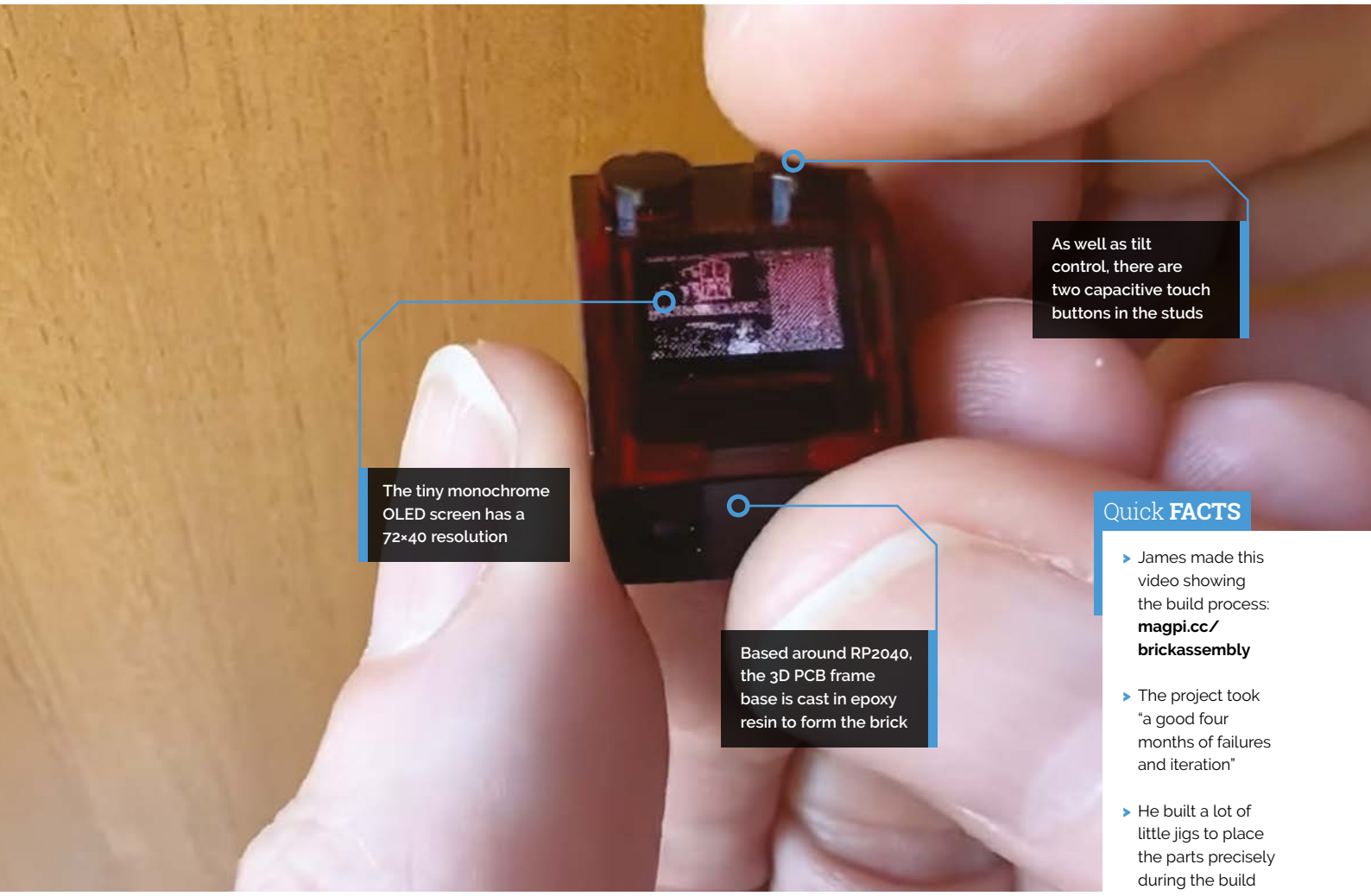
Doom's day

Sparked by questions from social media as to whether his LEGO creation could run Doom, James investigated the possibility but had initial issues running video on the screen. After some tinkering, he "took advantage of the RP2040's second core to update the screen fast enough to create greyscale images (**magpi.cc/ugrey**)." With Doom now a practical proposition, James looked to Graham Sanderson's "amazing" work on a Doom port, did a "horrible quick hack" and it now runs beautifully, ticking off "another entry on 'things that run Doom'."

> "The biggest headache was getting the resin layer over the screen just right"

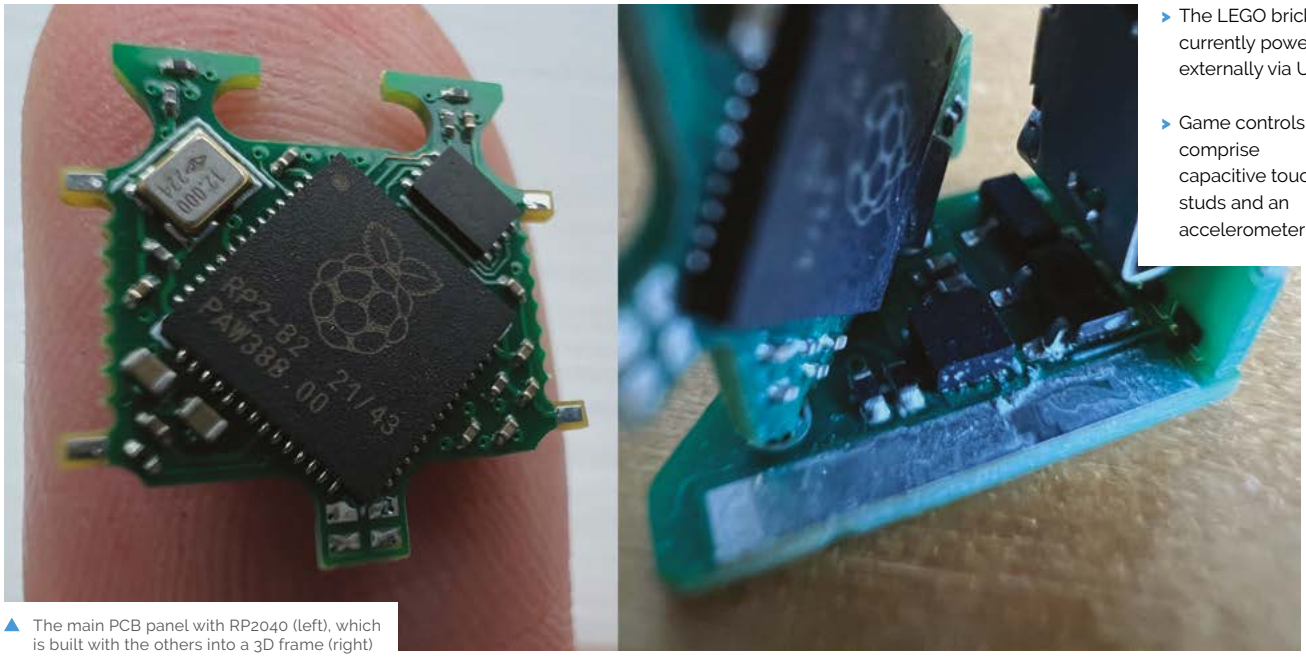
Seeking to improve the pocket-sized project further, he's now working on adding an internal battery. "I found some rechargeable cells which could fit, but only if I remove the USB port, and I'm still looking for a neat solution for this."

As well as making a test board to try out the battery and charging circuit, James has copy-pasted the LEGO brick's computer module onto it "to make a wearable ring (**magpi.cc/doomring**) which can run all the same code as the bricks."



Quick **FACTS**

- James made this video showing the build process: **magpi.cc/brickassembly**
- The project took "a good four months of failures and iteration"
- He built a lot of little jigs to place the parts precisely during the build
- The LEGO brick is currently powered externally via USB
- Game controls comprise capacitive touch studs and an accelerometer



▲ The main PCB panel with RP2040 (left), which is built with the others into a 3D frame (right)

Bugg.xyz acoustic monitoring

Soundscapes can provide a valuable insight into the health of a habitat. A Raspberry Pi can help conservationists listen in, explains **Rosie Hattersley**



MAKER

Sarab Sethi

Dr Sethi is a post-doctoral research fellow in Cambridge University's Department of Plant Sciences.

bugg.xyz

The louder the dawn chorus and the cacophony of wildlife, the better the locale and, by extension, life for the creatures that dwell there. This is a simplified take on the importance of acoustic monitoring, as conducted by Bugg.xyz using a Raspberry Pi–based system.

Bugg wildlife traps record the sounds of the forest, and are AI-trained to listen to the quietest creatures, as well as those that broadcast their whereabouts to all and sundry. In fact, it's more of an acoustic census, creating detailed records of the particular soundscapes of contrasting habitats, noting changes that may indicate deforestation or a natural disaster leading to depleted biodiversity. Dr Sarab Sethi, an environment research scientist at Imperial College London, developed the Bugg acoustic monitoring system, which is now being used to record the sounds of forests from Norway to Taiwan, as well as Bali and Borneo.

Sound and vision

Sarab created his Bugg kit having read about field monitoring devices built using off-the-shelf components and based around a Raspberry Pi and which featured a microphone, a USB audio card, a plug-in internet dongle, and a weather sensor. Suitably inspired, Sarab set about designing "a continuous, long-term fully autonomous acoustic monitoring device" that could send details back to the lab for analysis and which was robust enough to be placed and remotely monitored in a variety of habitats. With price as well as the hardiness of the components to consider, Raspberry Pi was an obvious choice: "The bigger questions weren't so much about whether a Raspberry Pi would work with it, but more around the field logistics of mounting one in a forest to withstand temperature swings, humidity swings, and long duration deployment."

Helpfully, Sarab's PhD supervisor had set up SAFE (Stability of Altered Forest Ecosystems),



The team in Norway retrieves the Bugg boxes each winter, since there's little wildlife activity then and deep snow hampers recording



Bugg's injection-moulded case provides robust weather-proofing and can be securely lashed to a tree in even the most exposed environment

Acoustic monitoring using a microphone, attached to Raspberry Pi 3B or above, helps further our understanding of humans' impact

Solar charging is sufficient to power Raspberry Pi, the microphone, and 3G/4G mobile module so sounds can be sent and processed as soon as they're recorded

Quick **FACTS**

- Weatherproofing for the first Bugg monitoring kit was rustic…
- A gardener's dry-box was used to protect cables
- Later versions are 3D-printed and injection-moulded
- DC power correction compensates for erratic electricity supplies
- Solar panels power the devices throughout all seasons bar winter



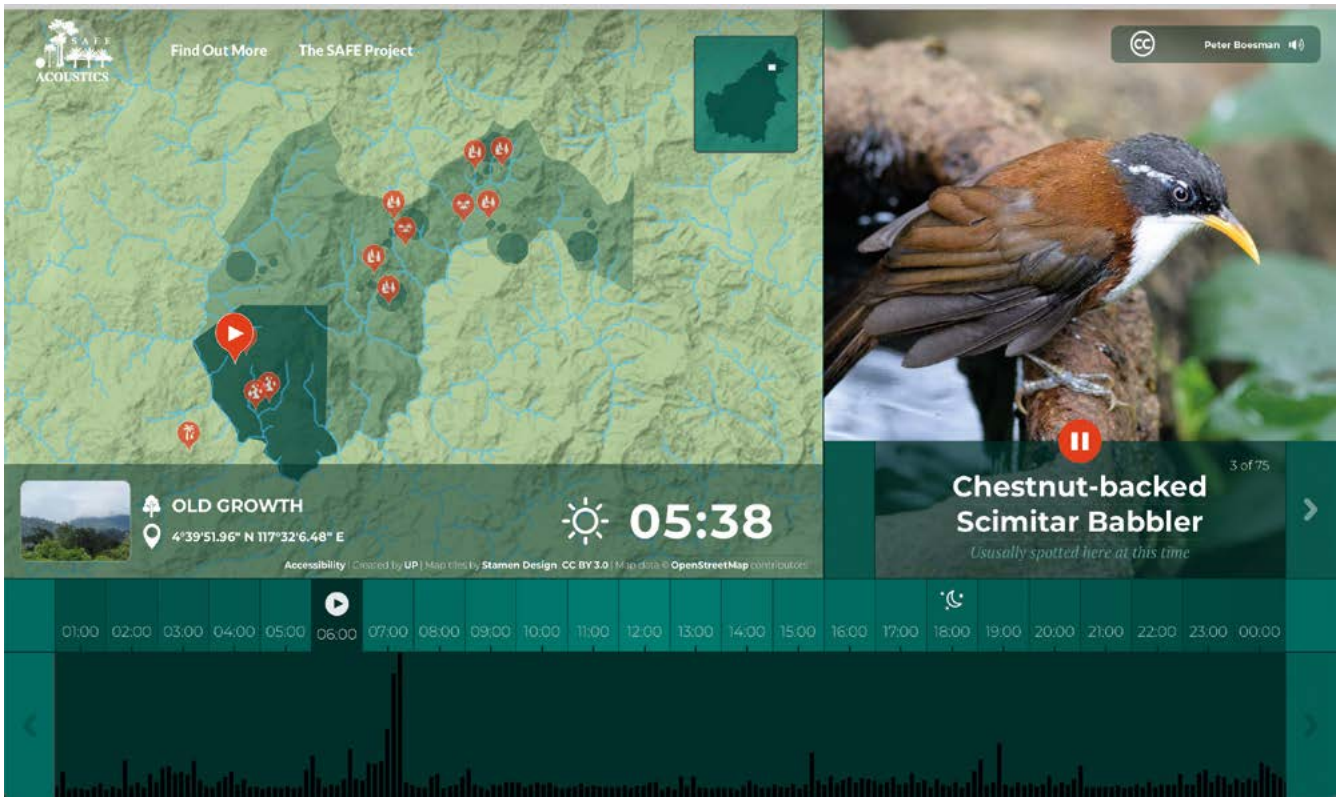
Sarab [rear] and colleague install Bugg boxes in the Borneo rainforest, which is subject to man-made degradation that affects both wildlife and the forest itself

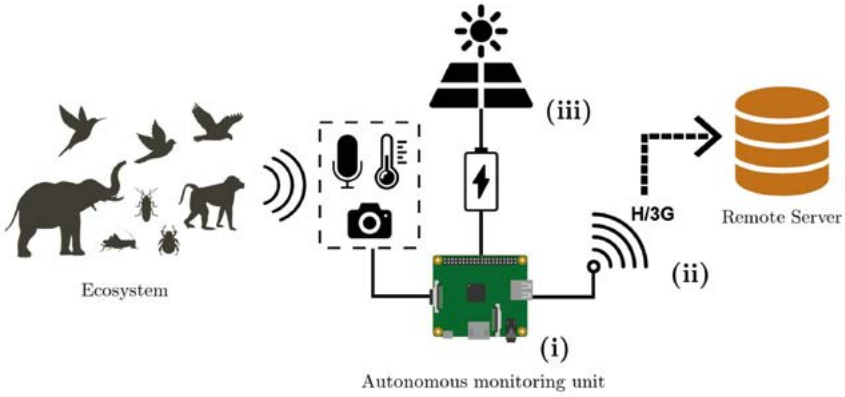
Field recordings of insects, invertebrates, and birds can be viewed and heard on the SAFE Acoustic site, along with maps showing habitat health variations

a rainforest degradation monitoring project in Borneo (**acoustics.safeproject.net**), which formed the ideal proving ground for the Bugg acoustic monitoring system.

"Whenever we're setting up an ecological experiment, we're looking at gradients," says Sarab. The SAFE Project questions how habitat degradation in Bali and tropical forest scenarios affect biodiversity. Samples are taken from pristine virgin untouched forests to lightly logged and heavily logged forests and then ultimately, the oil part monocultures. "You can also look at machine learning embeddings of the audio or fingerprint, go and survey ten sites then figure out which soundscapes correspond to what types of biodiversity and what type of habitat quality." By going to ten more sites, or visiting the same site on ten different days, Sarab can use Bugg to infer "just based on that minute of audio, what the habitat quality or the biodiversity or stability of that system is."

In the event, Bugg more than held its own against the various moth-trapping and bug-detecting traps, and neatly complemented studies into how forest fragmentation and forest degradation affect biodiversity. The





▲ This schematic shows how data is continuously captured from an ecosystem and uploaded to a remote server

results were published in the journal Methods in Ecology and Evolution, plus a website that details how to go about building your own device (**magpi.cc/ecosystemmonitoring**). Unfortunately, acoustic monitoring sometimes reveals a lack of wildlife and, what appears from photos to be healthy rainforest, is actually an empty forest.

Anyone home?

The open-source Bugg kits are now used for multiple monitoring tasks. While vegetation sampling involves taking samples and measuring trees, Sarab's Bugg monitor takes a more holistic approach, "using soundscapes as a whole to infer habitat health." Once the microphone in one of the Bugg devices detects sounds, they are transmitted over a mobile network for machine

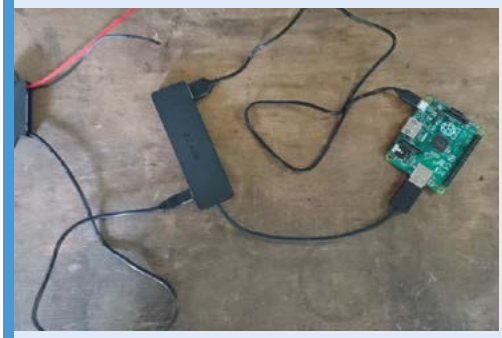
> A rainforest degradation monitoring project in Borneo formed the ideal proving ground for the Bugg acoustic monitoring system

learning-based identification and analysis by scientific researchers. Bugg is generally used to pick up the sounds of particular insects or bird species, so the Raspberry Pi 3-based kit can be trained to recognise and almost instantly identify what's making each sound, be it migratory bird flocks over Arctic Norway or ants in an Indonesian rainforest. Some projects look at a particular habitat in great detail over time, as is the case in Borneo, with four distinct gradients of degraded forest, while in Norway there's a focus on multiple sites with the aim to get a detailed overview of bird life and habitat choice across the whole country (**thesoundofnorway.com**).

Get bugging



01 Sarab shares the project setup and materials information on his website (**magpi.cc/ecosystemmonitoring**). You'll need a Raspberry Pi, a microphone, USB audio card and a Camera Module, a waterproof case to house it, and a means of powering everything.



02 Install the software image provided on the GitHub link to a microSD card of at least 32GB capacity. Connect the microphone, camera, audio card, and other components to Raspberry Pi.



03 A gardener's dry-box makes a decent waterproof housing. Tape the microphone to the outside of one end, and run cables for the power adapter (if using) to the other. Snap the lid into place and lash the box to a secure post or tree, out of harm's way.

Refurbished Vending Machine

A tech fan's plan to give something back to his community led to a Raspberry Pi-based dispenser, learns **Rosie Hattersley**



MAKER

Fady Faheem

17-year-old Fady is a "tech enthusiast, programmer and perfectionist" whose love for computing began when he learned Java at school.

magpi.cc/vendingmachinegit

If you can think of it, "you can probably do it on a Raspberry Pi," believes 17-year-old Fady Faheem, who saw a broken vending machine for sale online and decided to give it a tech upgrade. Fady wanted to use his tech chops to benefit his community in Fort Worth, Texas, and the refurbished vending machine seemed an ideal gift to his local church – "the best place I knew" – which could do some micro fundraising too. The project has kick-started his career: "If I didn't do this, I wouldn't be where I am today, working as a software engineer at 17."

Feeling blessed

Task one was the discover whether the vending machine was fixable. Buying it for so little, "knowing it was broken and the motherboard fried, I wasn't sure what I was walking into," reveals Fady.

Even so, having learned Java and programming since his freshman year, he felt sure he had the power to accomplish it. He became interested in Raspberry Pi because it can be used for "just about any project", noting the range of different-sized models, "different ways to utilise it, low energy, and overall simplicity."

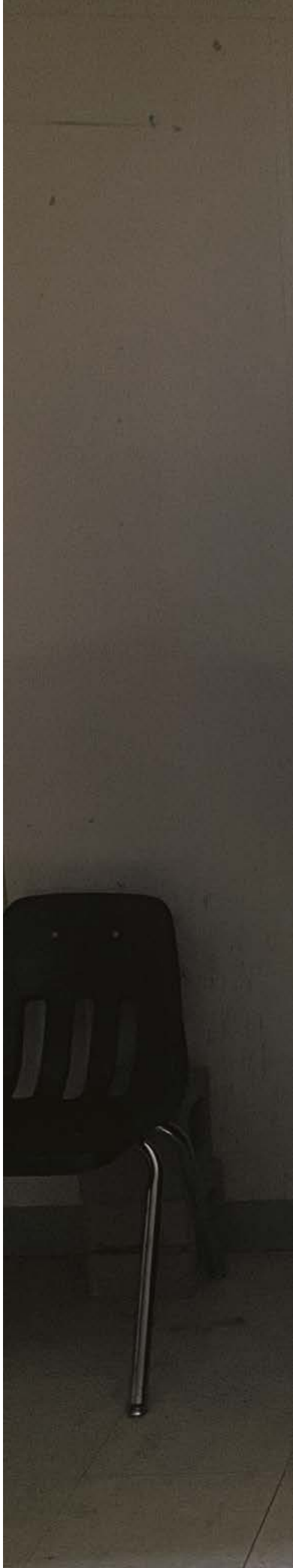
Raspberry Pi would become the heart of his vending machine, connecting everything: communicating with the dollar bill acceptor, sending data to relays to activate them, and handling power between all devices. There was little to no space to put a machine or full-size computer behind the vending machine screen, so the diminutive Raspberry Pi was "the ultimate solution to all my problems," he says. "I had a small corner to run the hardware. If it could fit in my hand, I knew it would fit in the machine." Fady's refurbished vending machine originally used a 3B+, but he recently upgraded

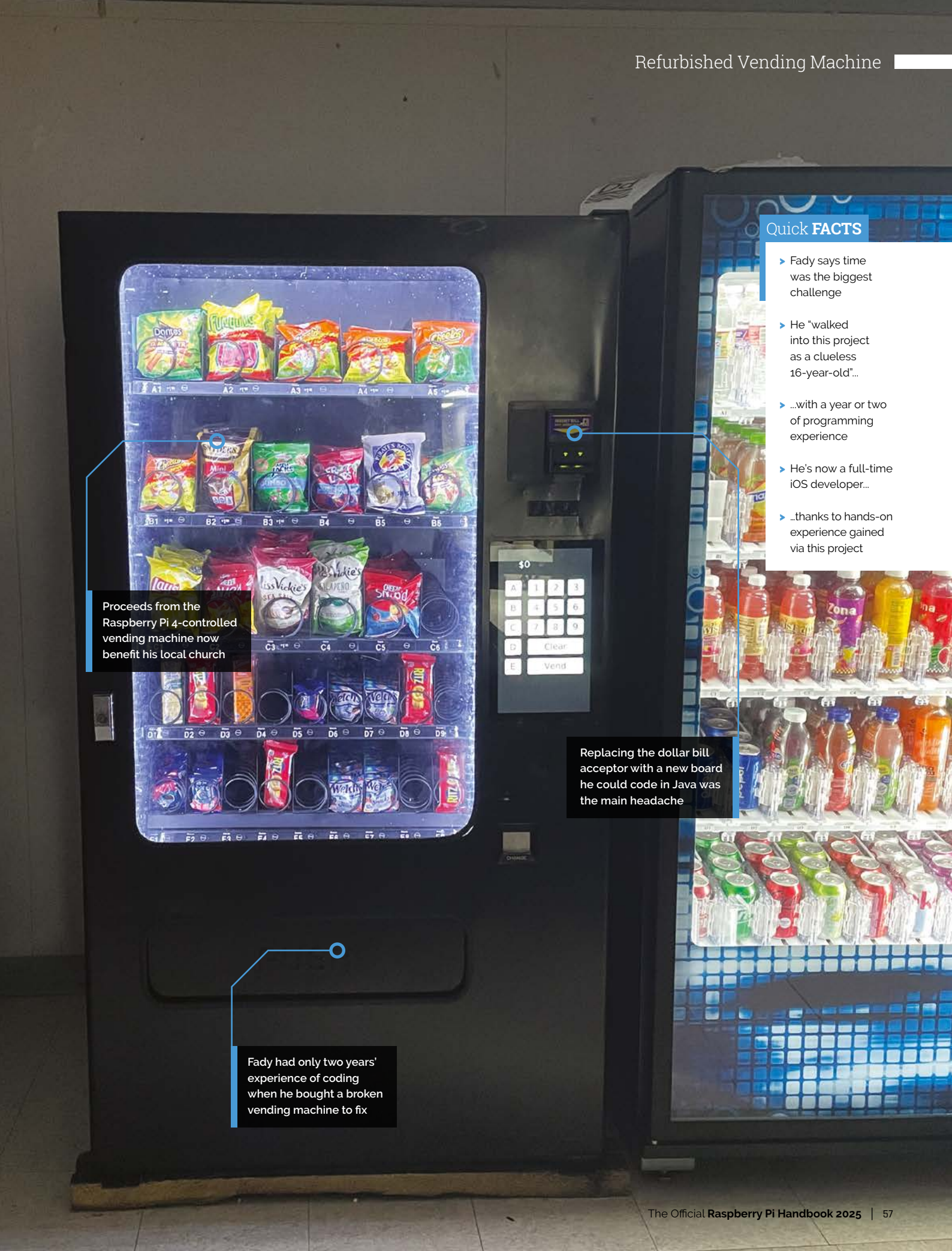


Warning!
Electrical Safety

Please be careful when working with electrical projects around the home. Especially if they involve mains electricity.

magpi.cc/electricalsafety





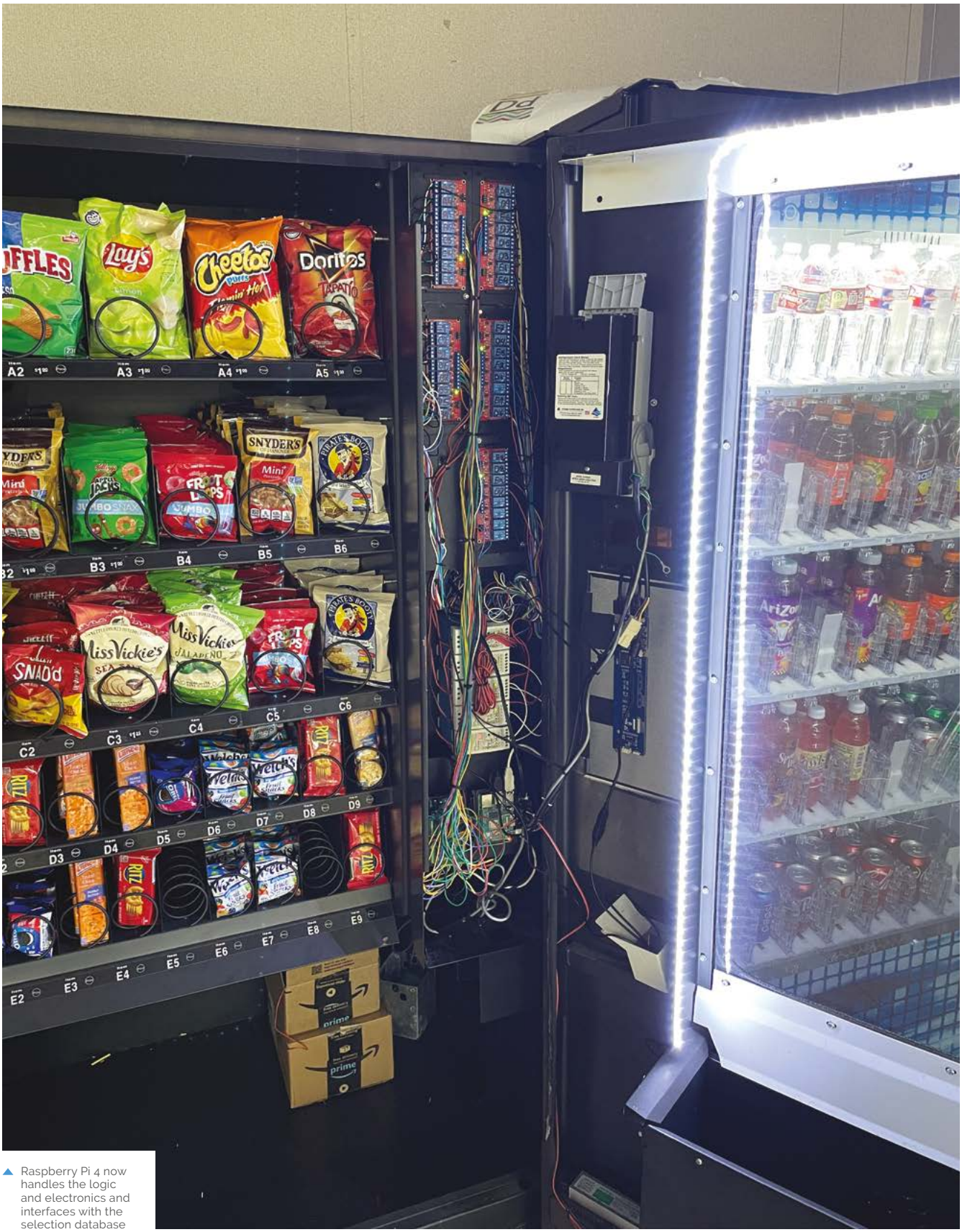
Quick **FACTS**

- Fady says time was the biggest challenge
- He "walked into this project as a clueless 16-year-old"...
- ...with a year or two of programming experience
- He's now a full-time iOS developer...
- ...thanks to hands-on experience gained via this project

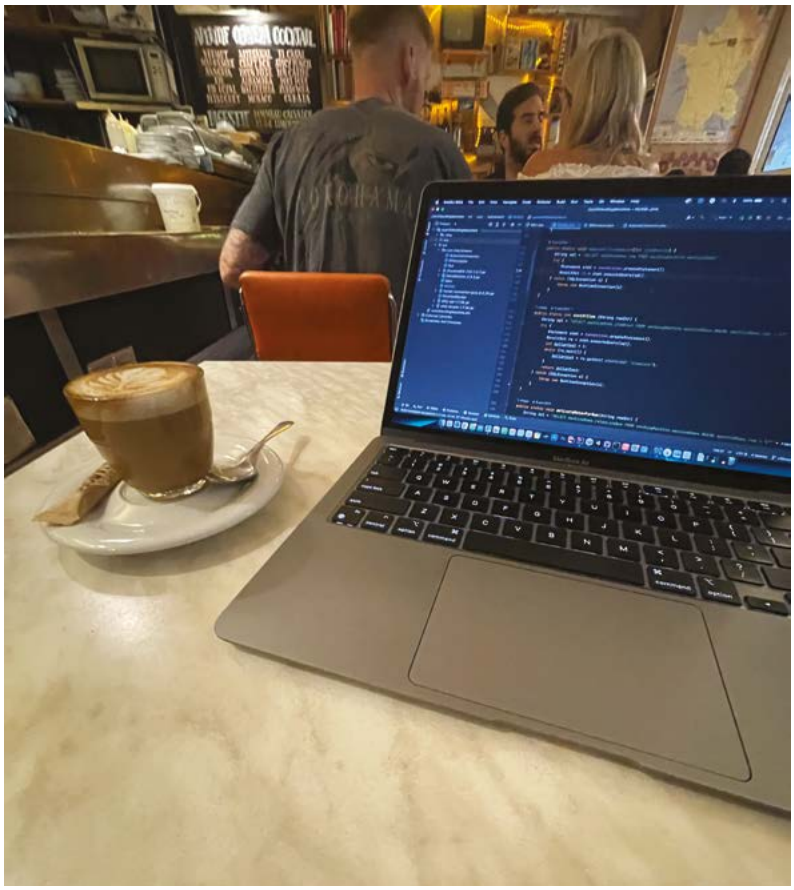
Proceeds from the Raspberry Pi 4-controlled vending machine now benefit his local church

Replacing the dollar bill acceptor with a new board he could code in Java was the main headache

Fady had only two years' experience of coding when he bought a broken vending machine to fix



▲ Raspberry Pi 4 now handles the logic and electronics and interfaces with the selection database



▲ Fady put his love of Java to great use by providing a welcome refreshments machine for fellow members of his church

it to Raspberry Pi 4 for performance on start-up, should he ever need to pull the plug and restart.

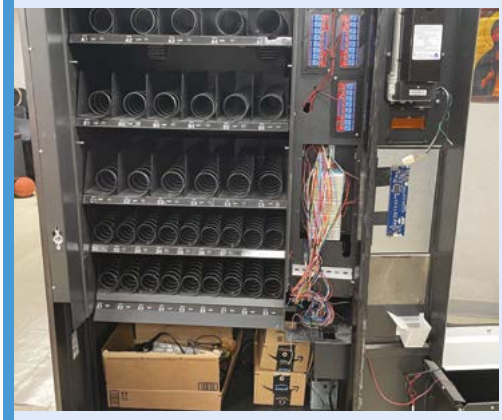
Money makers

Fady began to tackle the project by researching how other people were able to use a specific combination of dollar bill acceptor and a circuit board to run Java. For the first version, Fady used a Pyramid Apex 5400, but later switched to the Apex 7400 bill acceptor, resulting in better support and handling through Java. Vending machine selections are handled by the MariaDB database and Raspberry Pi using a number of libraries Fady found on GitHub. Meanwhile an Arduino connected to Raspberry Pi via the vending machine's printer cable, and accessed using Java, handles the signals to the relays and the relays themselves.

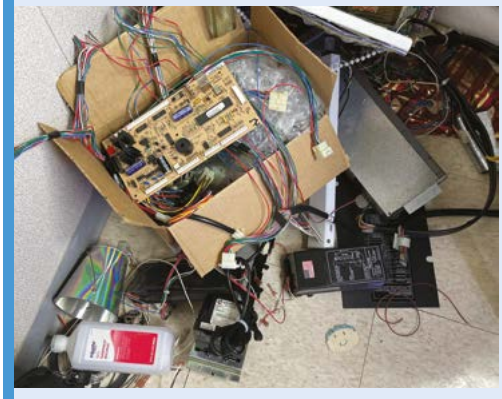
> "If I didn't do this, I wouldn't be where I am today, working as a software engineer at 17"

"At the end of the day, all the work that went into it was designed and 3D-printed by me," he says proudly.

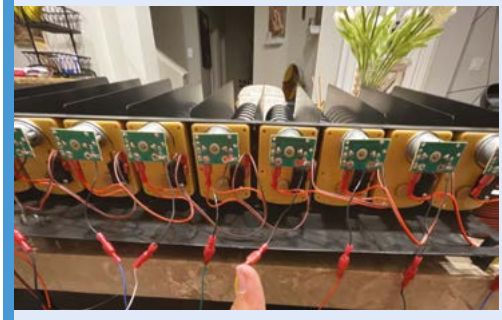
Back in service



01 Having received the machine he bought online, Fady removed the control panels to reveal outdated, underpowered coin and note mechanisms.



02 Fady installed a replacement bill acceptor board that was sufficiently powerful for his needs and that worked smoothly with the Java code, database, and libraries he chose.



03 At the 'heart' of the vending machine a Raspberry Pi 4 controls everything, alongside an Arduino connected to relays to fulfil customer selections.

Antarctic Picam

A simple waterproof camera by a school student is being used to look under the cracks in the Antarctic. **Rob Zwetsloot** takes a peek



MAKER

Oliver & Autun Purser

Oliver is a student who knows Autun and his work from a stay in Germany. He was inspired by an article in *The MagPi*. Autun is co-ordinator for the scientific use of the deep sea robots and camera systems for the Alfred Wegener Institute of Polar and Marine Research, Bremerhaven, Germany.

awi.de

Way back in issue 74 (magpi.cc/74), we wrote about PiCam Marine, little monitoring devices for cold water coral deployed by submarine. We like to think the projects we publish in the magazine can act as inspiration, and this one did…

"A couple of years ago, we started designing and using Raspberry Pi-based camera systems for under-sea research work," Dr Autun Parser tells us. He's a research scientist for the Alfred Wegener Institute of Polar and Marine Research. "These cheap, programmable platforms allow us to place many cameras on the sea-floor, for the same costs as one 'off the shelf' marine camera. We had an article on these cameras in *The MagPi* magazine and Oliver contacted us, a student then at the International School Bremen, Germany, who wanted to design and program his own camera."

"[I really] wanted to see what laid in the river outside of my house," Oliver explains. "After I first deployed the camera, I told Autun, who said that he could take it with him to Antarctica on his next mission." With the camera tested, Autun stood by his word, took it on the RV Polarstern, a research icebreaker ship, and deployed it during two helicopter missions.

Plumbing the depths

The solution for the water-tight camera ended up being nice and simple.

"Oliver made the camera housing out of robust plastic, and used an interesting plumbing seal to close one end," Autun says. "You put a Raspberry Pi in, turn on the power, and turn a nut to close the housing… the nut presses two plastic pieces together to expand a rubber seal – very effective, quick, and robust. The front of the housing is a simple acrylic sheet, against which a standard Raspberry Pi Camera Module 2 was placed. This was connected to a Raspberry Pi 3, attached to a wood mounting block. Below the block, a simple USB power bank was placed to power the system."



PVC pipe with a clear acrylic window which the camera can peer out of. It's waterproof thanks to a simple seal at the opposite end

Oliver's original version also had a little bait container to try and lure fish closer. "The program is written entirely in Python, simply takes a 30-minute recording, and after the 30 minutes, it automatically shuts off," Oliver elaborates. According to Autun, this 30-minute length of the

Raspberry Pi is mounted onto a simple wooden base which can be easily slid into the waterproof housing

30 minutes of footage is taken by the Camera Module 2, which is able to see just fine through the acrylic window

Quick **FACTS**

- Antarctica holds 70% of the world's fresh water
- The water in Antarctica can get as low as -2°C
- Oliver wants to take his camera to other bodies of water near him to test it more
- The RV Polarstern ship is over 40 years old…
- …and means 'pole star' in German

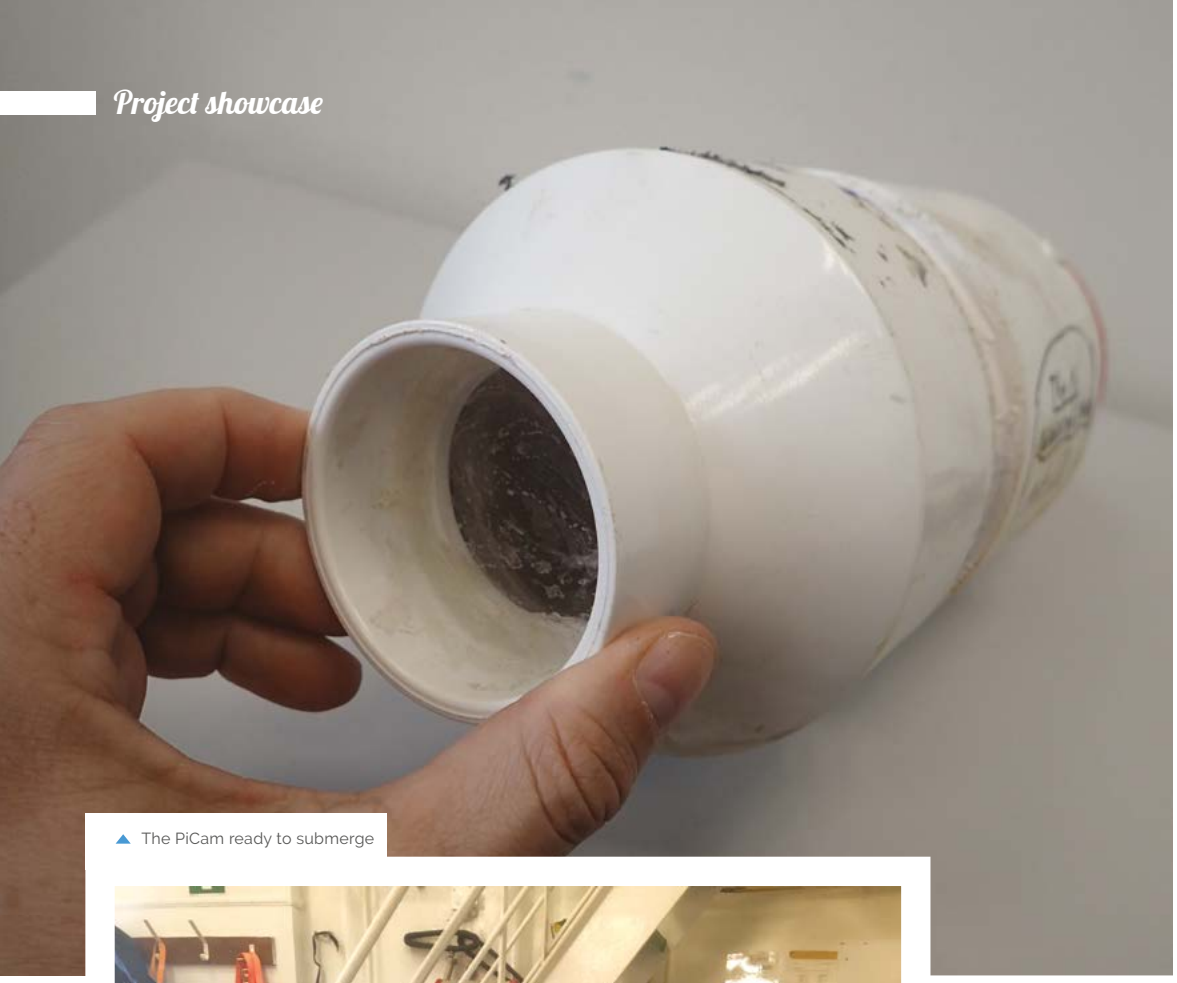


▲ The seal uses a very simple plumbing system

script was enough for what they had planned: lower from a helicopter, let the video record, and then retrieve it. It was deployed five times during the mission, mainly through ice cracks. "We even slid it from the helicopter down the flank of a large iceberg and down onto a shelf of ice 10m or so below the waterline," Autun mentions.

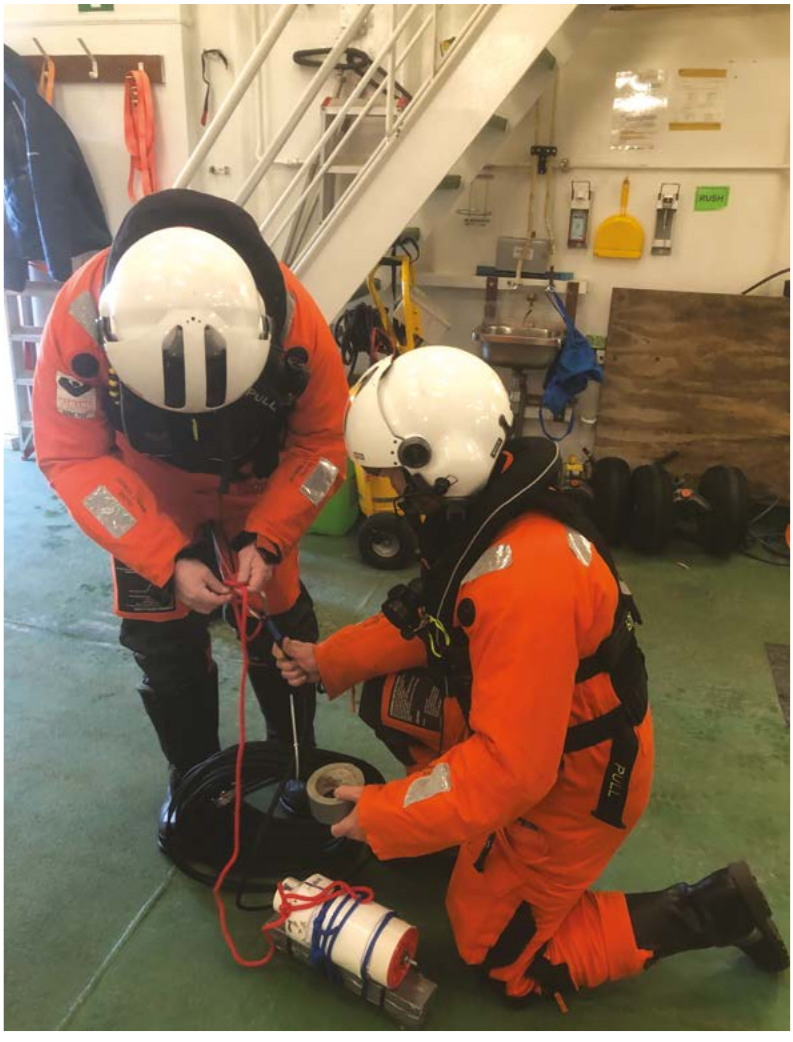
Icy depths

"The results from the camera have been really good," Olive tells us. "It's able to capture decent

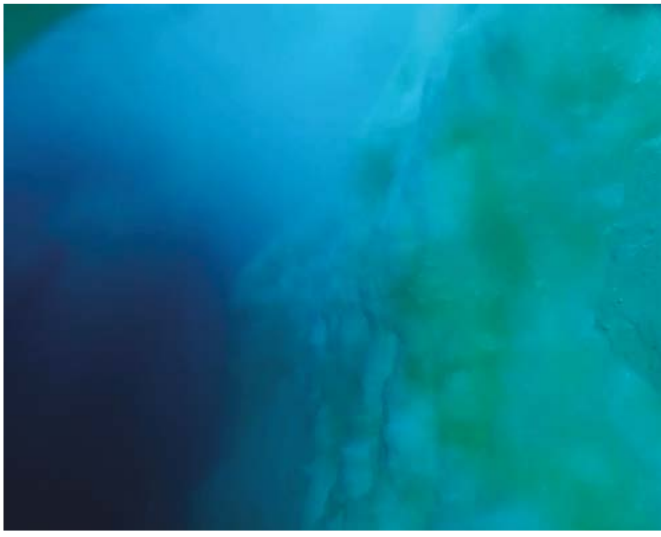


▲ The PiCam ready to submerge





▲ The PiCam was used in helicopter missions



▲ Algae grows on ice, apparently

footage, and I can clearly make out everything that it captures. Raspberry Pi is relatively low on power usage, and the battery has enough capacity that, theoretically, the camera could run for a couple of hours."

Autun sent us some amazing images from under the ice to prove it, and is planning to use similar cameras for a future trip on the RV Polarstern, although this time up to the Arctic and the North Pole.

"Within our working group, we are very happy to support the development of environmental



▲ View from under the ice

monitoring devices and sensors for use in polar regions, both above and below the ice, with a Raspberry Pi base," Autun reveals. "We travel

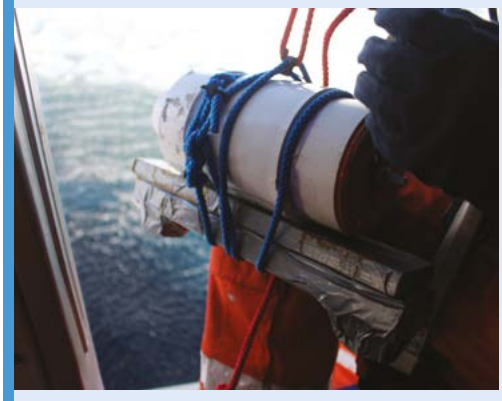
> "The nut presses two plastic pieces together to expand a rubber seal - very effective, quick, and robust"

every year to the Arctic and/or Antarctic and we can gladly discuss taking your devices with us for testing and scientific use – the scientific community can definitely benefit from the development of systems and coding by enthusiastic makers. You can email [me] at **autun.purser@awi.de** if you have an idea of something you would like to see above or below the polar ice!"

Getting cold snaps



01 Getting to the Antarctic takes a special kind of ship – in this case, an icebreaker. Load it up with people and helicopters and you're ready to go.



02 The camera needs to be started and then deployed from the helicopter into ice cracks. Due to the length of the video capture being 30 minutes, there's plenty of time to set up and go.



03 Scrubbing through the video once the camera has been retrieved helps with scientific study of the icy waters near the poles.

Olga The Fortune Teller

Admiring glances across an office led to a creative partnership involving several Raspberry Pi models and some charming AI, learns **Rosie Hattersley**



MAKER

Kakapo Labs: India, Will Lahr; AlterD: Will Lucking, Claire Richards

AlterD creates retail displays for clients and shares office space with interactive design agency Kakapo Labs, which routinely uses Raspberry Pi in its creations.

magpi.cc/kakapolabs

Last issue, we brought you details of the Love Machine ChatGPT project which dished out compliments and sometimes sweets based on cutesy WhatsApp exchanges. Raspberry Pi staffers and *The MagPi* magazine readers weren't the only ones to fall for its AI charms. A company that shares office space with Love Machine makers Kakapo Labs were equally entranced, and soon made a move, proposing a coupling up and a glow-up! The result: Olga The Fortune Teller, a shiny new ChatGPT model designed to appeal to the Insta-savvy influencer keen to check out both the latest skincare options and receive reassuring messages about their future fame and follower count.

Altered images

Olga The Fortune Teller was designed in partnership with AlterD, a company that designs displays for retailers. Both companies are based at Hone (**hone.london**), a creative space in East London. The team from AlterD was on the look out "for something visually attention-grabbing", saw Kakapo's Love Machine, got talking to its makers, and things went from there.

With a retail display trade show coming up imminently, AlterD chose a cardboard-based design to show off the material's possibilities. "It was important that it didn't look too specifically like any particular brand, so they went for a fun, quirky, dream-like look. They wanted to add an interactive element to encourage people to interact with the stand and start conversations," Kakapo Labs founder Will Lahr explains.

As with Love Machine, mobile connectivity was vital, so they chose a Pico W: "simple, low-cost and small but, at the same time, [it] can run a full





Olga's beauty secrets involve lashings of perspex, cushioning cardboard, and colour-changing mood lighting, plus some Pico W-controlled eyebrow raising

Once you've checked your visage in the helpfully supplied make-up mirror, scan the QR code below it to engage Olga in a WhatsApp chat

Olga gets ChatGPT to foretell your fortune, sending it to a cloud server for Pico W to download and deliver as printed ticket

Quick **FACTS**

- Olga cost around £160 to build
- It was originally going to be a visual guessing game
- Instead, she became a 'director of divination'
- She features ChatGPT's new chat Completion model
- It can tell the fortunes of multiple WhatsApp users



▲ Olga's gorgeous face and expressive eyebrows were laser-cut

wireless stack which is easy to use." With MicroPython on the Pico W, things work painlessly and the development time is short. It's used here to communicate with a Zebra thermal printer via an RS232 connection and prompt it to print out a stand visitor's fortune ticket. ChatGPT is instructed to: 'Pretend you are a fortune teller and tell the person you are talking to – who is called $NAME – their fortune in a poem of three or four short lines. Make sure their fortune is vague but positive! Include a lucky item (retail related).'

A concurrent request is made to DALL-E (an image generator made by the same people as ChatGPT for a black and white cartoon image of a fortune teller. Fortune text and graphic are imported and encoded as an image that can be recognised by ZPL, the language used by the printer. Laggy performance processing and outputting the images led the Kakapo and AlterD team to switch from the unreliable existing WebSocket delivery setup. The "much better" system they implemented saves the ticket image online and just uses WebSocket to send Pico W the URL in order to download and print it.

Fortune favours

Olga herself is, of course, a large part of the appeal. She has a face cut from a sheet of perspex and some extremely expressive eyebrows thanks to a Pico W, micro servos, and some black thread that help them wiggle. She also imparts her wisdom via a fancy new means of communicating (due to changes to ChatGPT that OpenAI recently implemented), but it's still based on WhatsApp and Twilio, like Olga's sibling, the Love Machine (see *The MagPi* 130, **magpi.cc/130**).

The brief from AlterD was for a playful interactive display showing off the possibilities for retail settings. However, Will points out that creating something very quickly and efficiently using several Pico Ws (which, being connected



▶ Colour-changing LEDs make the booth design more eye-catching and hide Olga's cardboard outline



> Olga herself is, of course, a large part of the appeal. She has a face cut from a sheet of perspex and some extremely expressive eyebrows

wirelessly, avoids the need for long cable runs), making use of simple web services has all sorts of potential. "We've been exploring more possibilities for interaction through a shop window – using a phone as a device to let someone outside the window play with something inside the window," he says, and is up for collaborating with other partners for more fun, playful projects.

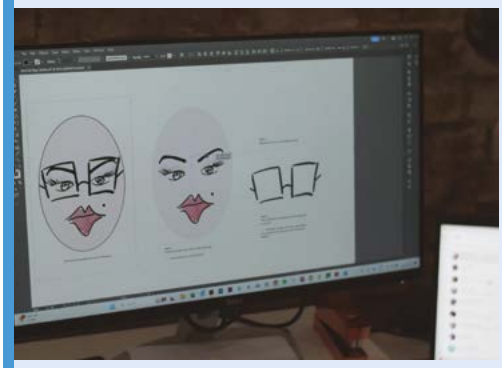
"Olga may not be able to predict the lottery or tell us when the cost of living crisis will end, but what she can do is offer a glimpse into the future of interactive retail experiences."

▲ Olga can hypnotise even the most casual passer-by into having their fortune told…

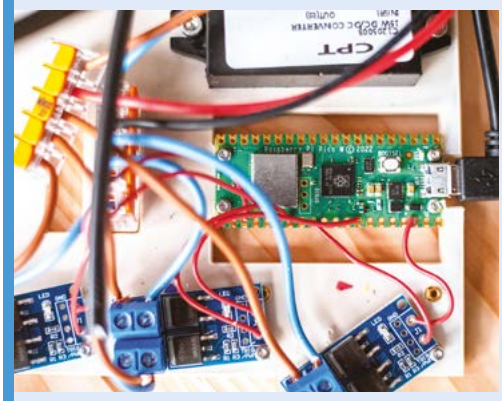


◀ After a brief WhatsApp chat, the booth prints out a pronouncement

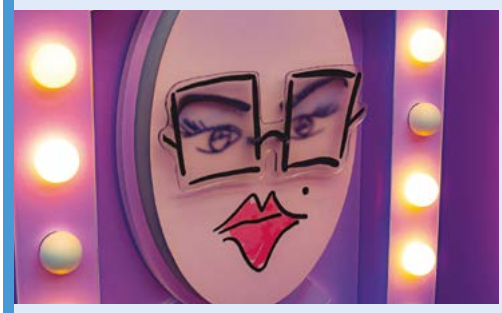
Characterful interaction



01 Olga's look is very much down to you. Create the fortune teller's face with or without a computer using items such as cardboard and Perspex, adding micro servos and Pico W if you'd like to include changing facial expressions.



02 Pico W controls the LED lights around Olga's make-up mirror, as well as the servo motors creating her facial expressions, while another connects to the Zebra printer to produce the all-important fortune readings.



03 Use the instruction in the main article to get ChatGPT to tell a fortune. Connect everything to a web server and thermal printer to print them out on demand.

Mona Lisa Fluid Painting

A creative way of demonstrating the capabilities of microfluidic biosensors, captured on a Raspberry Pi HQ Camera, caught the eye of **Rosie Hattersley**



MAKER

Yuksel Temiz

Microsystem engineer Yuksel impressed people so much with his Raspberry Pi Lego microscope, he has since set up a company developing 3D microscopes: **microqubic.com**.

magpi.cc/monalisa

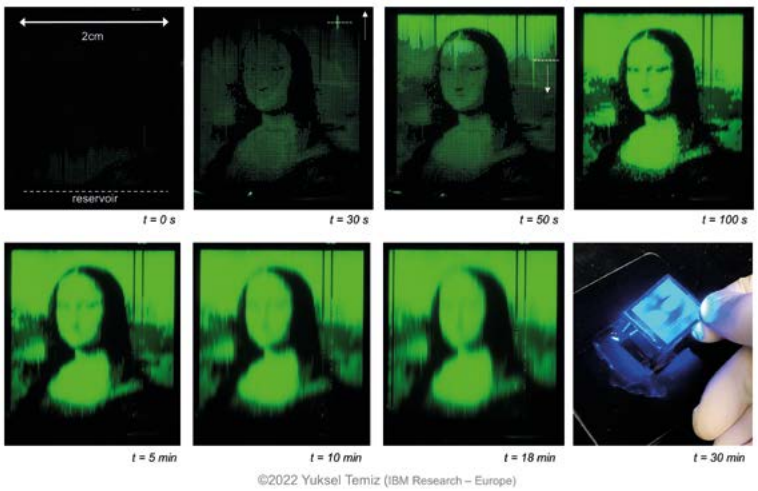
Big, brash ideas and enormous displays gain attention, whereas tiny but significant products can be far harder to convey. After helping develop a crucial microfluidic-based biosensor system for use in microscopic research at IBM Zurich (**magpi.cc/precisiondiagnostics**), microsystem engineer, and keen YouTuber, Yuksel Temiz and Polytechnique Montréal researchers hit on the idea of using an iconic artwork to show off its possibilities.

The Mona Lisa Fluid Painting project showcases a new microfluidic architecture that allows local and highly parallel dissolution of biochemical reagents in volumes as low as two nanolitres per spot. The image was created from just five microlitres of water 'painted' on a 2cm square microfluidic chip. The microfluidic architecture has potential applications in healthcare diagnostics and life sciences, "opening up new possibilities in studying kinetics and interactions of chemicals in nanolitre volumes and in high-throughput."

Yuksel spent nearly ten years working at IBM's labs in Zurich, and says, "In our research, we always needed compact optical, electrical, and mechanical peripherals to read out and control our biosensors. For me, Raspberry Pi has always been the easiest way to implement a minimum viable product, especially for imaging applications."

When it came to showing off the incredibly tiny droplets, Raspberry Pi soon became the obvious route to take, enabling them to create an impactful showcase to go alongside a press release about their work without investing too much time and money. "Raspberry Pi allowed me to test and optimise the setup very quickly with minimal investment," enthuses Yuksel.

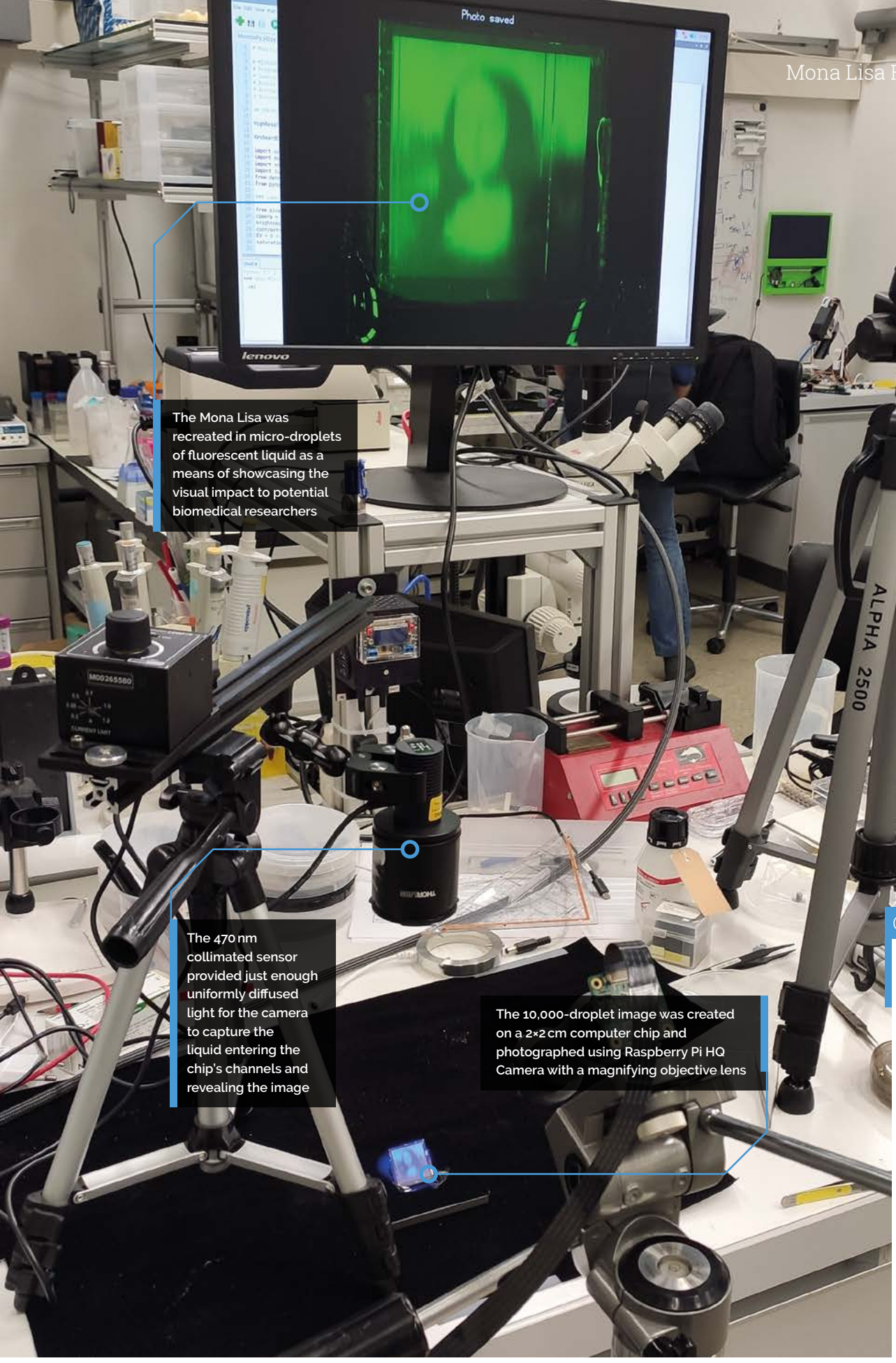
"Coding is just a hobby for me. But the thing I like about Raspberry Pi is you don't need to be a professional coder to build functional systems."



Stills captured using Raspberry Pi HQ Camera showing the Mona Lisa image being created

Camera conundrum

More than 10,000 droplets of fluorescent dye in eight different shades were spotted and dried inside microfluidic channels to generate the Mona Lisa image. Once the channels for the droplets had been engraved, the top of the chip was sealed and liquid applied – a process that took just under two minutes. It was important that the dyes were



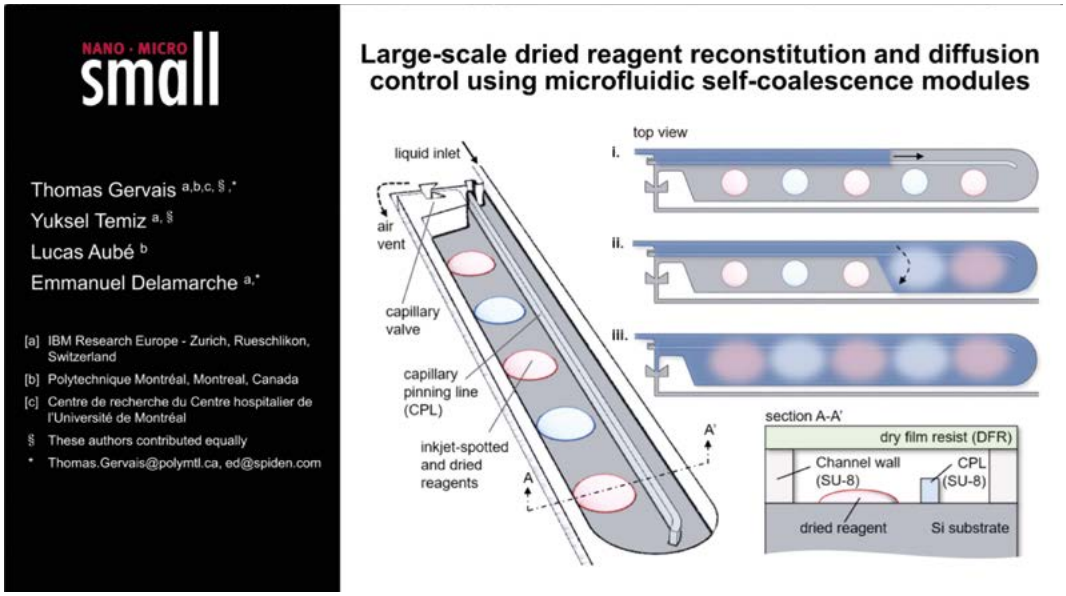
The Mona Lisa was recreated in micro-droplets of fluorescent liquid as a means of showcasing the visual impact to potential biomedical researchers

The 470 nm collimated sensor provided just enough uniformly diffused light for the camera to capture the liquid entering the chip's channels and revealing the image

The 10,000-droplet image was created on a 2×2 cm computer chip and photographed using Raspberry Pi HQ Camera with a magnifying objective lens

Quick **FACTS**

- The 2×2 cm image contains more than 10,000 droplets
- Fluorescent dye droplets were dried inside microfluidic channels
- The microfluidic droplets provide eight levels of shading
- They sit in microfluidic channels a hair's width wide
- The image lasts only a few hours before dissolving



Yuksel and colleagues presented a guide to using the microfluidic sensor modules as part of their research paper

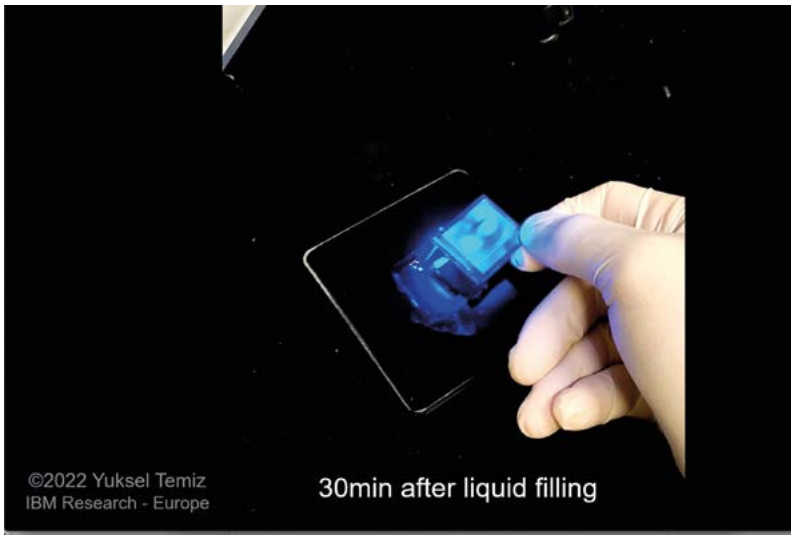
> The thing I like about Raspberry Pi is you don't need to be a professional coder to build functional systems

dissolved locally without dragging them. Having painstakingly planned the image, it then faded out after a few hours as the droplets diffused. However, the fluorescence image was captured for posterity, having been filmed using a custom-built setup based on Raspberry Pi's HQ Camera. Given the impermanence of the image, it was critical that the camera captured everything first time, as there wouldn't be a chance to reset and repeat things if they went wrong.

Yuksel explains the difficulty of videoing the water filling the channels and revealing the portrait: the imaging area was too large for the field-of-view of their high-end fluorescence microscopes, while the channels and individual droplets of fluorescent dye were too small for an imaging setup with an SLR camera to be considered. "Image stitching was not an option either because the liquid filling was too fast for the

The YouTube video shows the image being created in real-time





▲ The image lasted just long enough for Raspberry Pi HQ camera to capture it, then began diffusing so the colours merged

motorised stage of our fluorescence microscope." He realised an imaging setup using Raspberry Pi's HQ Camera would be ideal because he could easily attach a C-mount magnifying objective lens and fix the camera to a tripod.

Building blocks

Fortunately, Yuksel already had a number of Raspberry Pis in his lab, and had gained plenty of experience of microscopy while designing and building a very impressive Raspberry Pi-based LEGO microscope a couple of years previously: **magpi.cc/hqmicroscope**.

For the Mona Lisa water painting project, he custom-designed and 3D-printed an adapter for the emission filter to be placed in front of the objective lens for fluorescence imaging. Illuminating the sample was tricky, says Yuksel, because the area was too large for light sources used in classical microscopes, and it had required a specific wavelength for the fluorescent dye. Fortunately, he had a collimated 470 nm LED light from an earlier project, which was able to uniformly illuminate the sample.

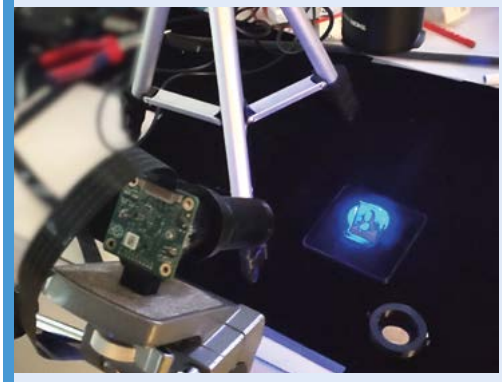
Photography-wise, Yuksel had also cut his teeth on a prior Raspberry Pi project: a stop-motion project showing a 3D model boat he'd made in action: **magpi.cc/stopmotionboat**.

Nonetheless, he was taking no chances, and put together a standby chip just in case. After all, the chip couldn't be reused once the liquid was applied, so there was just one chance to capture the image-creation process. Thankfully, everything went without a hitch, thanks to his meticulous planning.

Microscopic art



01 Microscopic channels to hold 10,000 drops of dried fluorescent chemicals were etched onto a 2×2 cm chip. Eight shades were used to create the Mona Lisa image.



02 A Raspberry Pi HQ Camera was mounted on a tripod, a microscope filter attached, and the whole setup trained on the microscopic chip. Many camera recording checks were made because there would be no second chances.



03 Filming began as liquid was slowly but steadily introduced in a controlled manner, where it dissolved the spotted dyes. The effect was visible for a few hours before diffusion caused the colours to bleed into each other and the spectacle was over.

Pipistrelle Bat Detector

Watching wildlife can be highly rewarding. This bat detector puts Pi Zero to good effect, discerning species by their calls. By **Rosie Hattersley**



MAKER

Phil Atkin

Phil began coding in 1980 and often uses it for high-end audio and acoustic projects.

omenie.com

What bats do in the shadows is quite mysterious, given the challenge of watching the speedy creatures as they flit silently by at dusk. However, their barely-audible-to-humans ultrasonic squeaks can tell those in the know plenty about their movements, moods, and motivations.

Unfortunately, bat detectors cost anywhere from £120 up to £1500+, and devices in the sub-£200 price range are technologically rudimentary, warns Pipistrelle Bat Detector maker Philip Atkin. "They operate like a transistor radio, requiring the user to twiddle a knob to tune into a specific frequency (i.e. a specific bat species), which can lead to frustration as you can miss bats because you are not tuned into the right species." As the Bat Conservation Trust shows, there are many more species even here in the UK than you might realise: **magpi.cc/ukbats**.

Phil set about designing a method of detecting bats that did not fall foul of frequency issues. His several decades of coding experience and, in particular, his expertise in music synthesis, proved ideal when it came to designing a low-cost device based around Raspberry Pi Pico.

Tuning in

The first challenge was working out whether there were any bats around. Bats echolocate using ultrasonic frequencies, well beyond human range, but they can't be heard and are hard to see: "uniquely for mammals, you need technology to detect and study bats," says Phil. His approach was to have Pico's sensors scan all the frequencies and seek out the strongest ultrasonic signal. It took him just three weeks to design a Pico-based bat detector that included an operational amp (one which amplifies weak signals), an ultrasonic microphone, a button, and enough software to detect bats and perform speech synthesis.

However, the working prototype board was "an ugly mess and too fragile to take out on surveys," says Phil, who then turned it into both a printed circuit board and an ultrasonic recorder, adding the ability to record 384kHz/16-bit WAV files to an SD card. This extended the project's usefulness and meant he could move on to perfecting the surprisingly challenging ultrasonic recording features. Not one to shy away from a technical challenge, Phil chose 128-bit FFTs (fast Fourier transforms) to ensure even the highest frequency bats could be detected. Pico offers exceptional functionality for its size. "Its rich feature set and programmable GPIOs meant that I needed to add the bare minimum of hardware to the design beyond the Pico," Phil comments "while offering a combination of low cost, low power consumption and the ability to handle 100% duty cycles when processing 128bit FFTs." This efficiency means Pipistrelle can be used as a passive recorder for four or five nights, "sleeping during the day, listening during the dark, and triggering recordings whenever bat candidate sounds are heard." These, he likens to music.

Using AudioMoth as a real time clock, the Pipistrelle device can automatically detect bats while unattended, as well as flash when one is detected during a bat walk

Pico offers high-sensitivity recording and 'speech-based' species recognition, recording each bat's beautiful calls and speech

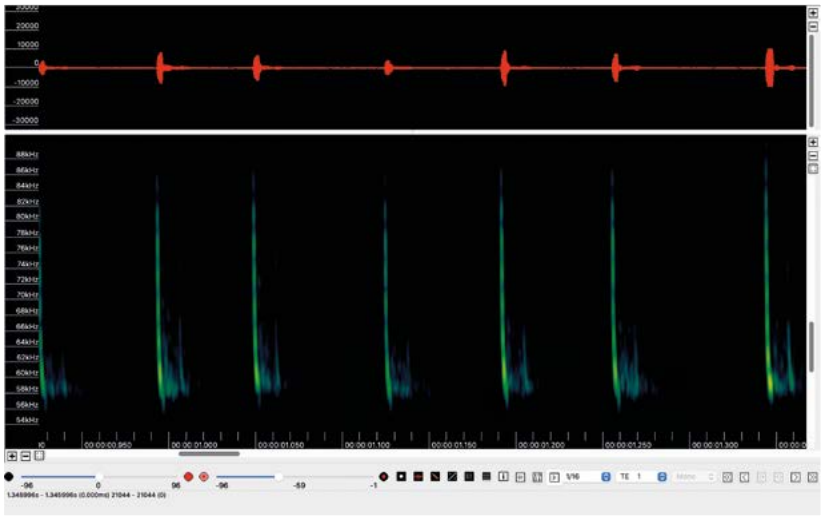
This Pico-based self-assembly Pipistrelle bat detector kit can detect echolocation bat calls across multiple frequencies

Quick **FACTS**

- Phil's interest in bat detecting began in church
- The Bats in Churches project used expensive kit
- So he learned CAD design to create a better option
- Using skills from his high-end *Doctor Who* audio project
- Which also used Raspberry Pi Pico of course: **magpi.cc/drwhopico**



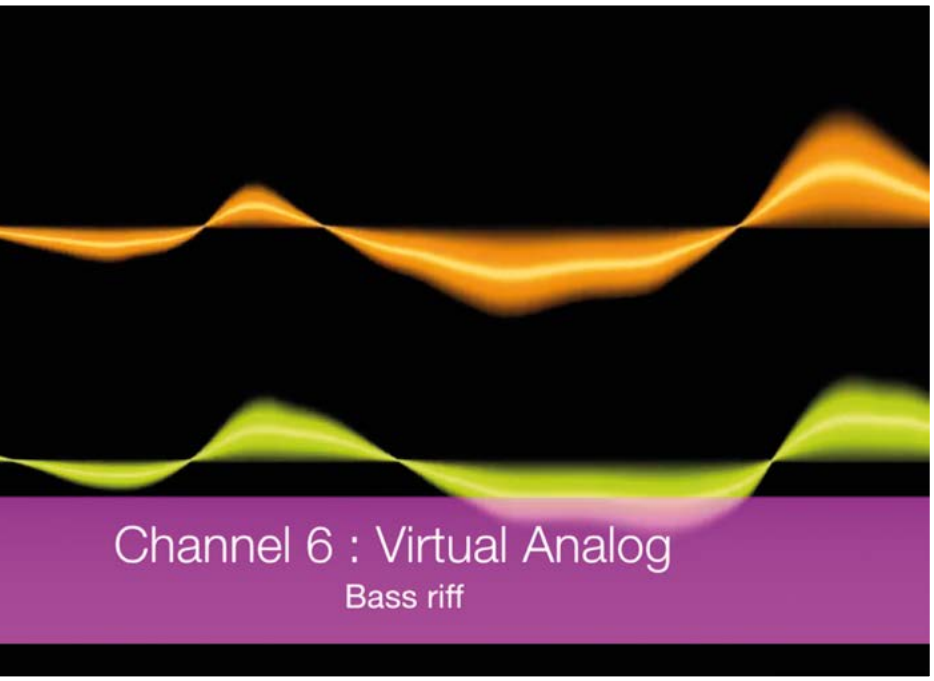
▲ The Bat Conservation Trust helps you identify bat species. This beauty is a Lesser Horseshoe



▲ A soprano pipistrelle bat captured in sonogram form by Pipistrelle Bat Detector



▲ Churchyards are great for finding bats



▲ Phil's previous project used a Pico to recreate the *Doctor Who* theme tune

"To hear the bats' true calls – the bird-like whistles, the peeps, chirps, and high-pitched screams of the Horseshoes – is remarkable."

Double duty

Two years on from his original prototype, there are now three models: BatWalk, a detector to take on bat walks; PippyG ultrasonic recorder; and Pipistrelle itself, which offers both functions.

> "Uniquely for mammals, you need technology to detect and study bats"

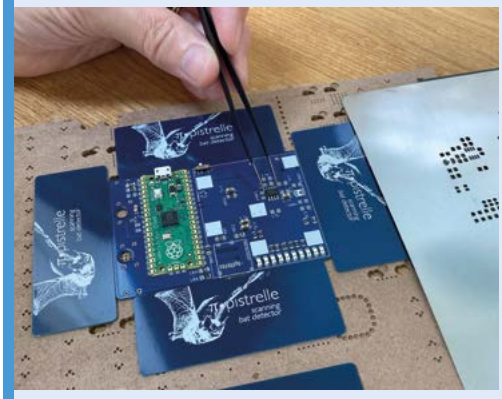
The recorder can either be used manually for on-demand recording, or set to record overnight whenever a bat call is heard. "Overnight recording lets me shut down more of the Pico to get cleaner recordings," says Phil, an audio purist keen to banish even the slightest operational sounds of the electronics caused by the need to write to the SD card. Each of these can be bought via his Omenie website, and integrated into your bat detection project. A full bill of materials and instructions are provided.

Though he continues to tinker with the audio, Phil finally feels the project is mature enough for someone to potentially create an Android version of it (since the software is open-source) and sees its further potential for studying other ultrasonic creatures, listing cetacean research, since dolphins use ultrasound, along with small mammals and insects.

Is a bat about?



01 Build a solder stencil rig and carefully tape it up, ready to fit the Pipistrelle circuit board once you've snapped them off their cardboard backing.



02 Align the Pipistrelle PCB parts around the microphone pads and SD card and attach Raspberry Pi Pico using solder paste. Connect and power up to check Pico's connections.



03 Add a battery pack for portable bat detecting, using sticky pads to help secure it. Happy batwalking!

Badgercam

What do you do when you want to observe wildlife at a distance with little interference? Dig out a spare Raspberry Pi, as **Rob Zwetsloot** finds out



MAKER
Philip Mill

A sound and AV engineer, sound recordist, and wildlife enthusiast.

▲ A small, yet powerful, wind turbine and solar panel charge the current version



Warning! Observing Animals

Do not bother animals in the wild more than you have to. Any observation posts should not disturb wildlife.

It seems like we've been covering wildlife camera traps a lot recently but, to be fair, they're all very cool. Philip Mill, a sound engineer with a love of wildlife, actually got in contact with us at the start of 2023 about his current project. Perhaps it's inspired the last few months of articles?

"Originally the project was created – and I generally referred to it as – 'Badgercam', but the current iteration is not at the badger location, so now it's just Solar and Wind-powered sound recorder (with image capture)," Philip tells us. We think Badgercam rolls off the tongue a bit better though. "[This was] to address the problem and help with identifying the species or source of sounds; making sure it was the badger I recorded; and also to keep my footprint at the setts low by having a discreet recorder, and a way to see into the life of badgers without walking through their environment. They are very sensitive animals and I wanted to only visit once, or a limited amount of times!"

While he initially wanted to use a Raspberry Pi Zero, stock shortages at the time resulted in him using a spare Raspberry Pi 3B+. "Raspberry Pi... had many appealing factors, including size, adaptability, and the option to connect it to a network for remote access via VNC," Philip explains. "And the [ability to add] cameras, PIR sensors, and USB inputs or an audio HAT made Raspberry Pi a logical choice. I always wanted to make a project on one and I thought what better application than this: to watch and observe badgers from my living room!"

Better camera trap

Raspberry Pi makes things easier, especially when you have an idea in mind.

However, there can always be a few snags with the parts surrounding it. "The building process, in terms of the hardware, has been a long and

Remote energy sources, such as solar and wind, can be used to keep the system operational

An IP camera takes photos, with a PIR motion sensor being used to activate it

It's important to make the smallest impact on the badgers' environment, so camouflaging is essential

frustrating (but fun!) learning process," Philip admits. "I've tried many, many combinations of different hardware, Python libraries, cameras, and housing designs to get to the point I am at now. This project is constantly growing and, despite all

> This project is constantly growing and, despite all of the challenges, I look forward to the next ones!

of the challenges, I look forward to the next ones!" The process has involved a lot of different box sizes, batteries, and solar panels, partly designed around the original idea of using a Raspberry Pi Zero.

The current version uses an 180W solar panel and a 400W wind turbine charging a 12V car

Quick **FACTS**

- The audio equipment used is a Zoom H1 Sound Record and Micbooster Clippy Stereo Mics
- A Tapo IP Camera C310 was the original camera used
- Philip's Dad, Robert, helped him with the coding
- The spare Raspberry Pi came from a project Philip funded
- Setts can have hundreds of metres of tunnels, and house 15 or more badgers



▲ Philip went through many variations of the build



A camera and stereo mics record the environment



While maybe not his intended target, at least Philip knows his system works

battery to power the whole system, which has some serious sound equipment added, along with an IP camera and a wireless router. This is clearly a bit more than the few amps that a 5V Raspberry Pi usually needs.

Witness the badger

So far, all the frustrations have been worth it.

"It works fairly well; it's come along way and I am really happy with the results right now; but there is much room for improvement," Philip tells us. "Also I am still learning things about efficiency. For example, the recent challenges have been designs for the housing that holds most of the recording equipment, Wi-Fi, Raspberry Pi, etc. At one point the box got to a temperature of 64°C, and although Raspberry Pi keeps working OK, either with a fan connected to the board or a heat sink, the other parts such as the Wi-Fi stop working at such temperatures and the unit switches off. Although this isn't such an issue as it doesn't affect the recording process."

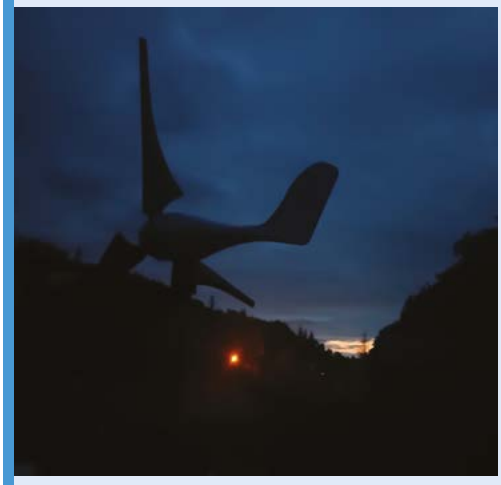
The next iteration is already being planned along with some fixes, such as RF interference from the audio recording equipment, and replacing the power-hungry IP camera.

"Recently, and I am very happy about this, I finally got my hands on a Raspberry Pi Zero [with a Raspberry Pi Camera Module NoIR attached]," Philip says. "This was the original idea for the project. Now, I am looking at options for audio and have a few HATs and some ideas for other ways to get better-quality audio. The code will most likely stay the same, but I am also considering the option of removing the PIR sensor and having the image capture by detecting motion changes based on a still image, which then will trigger the recording process. I'm excited for the next step and eventually deploying this at the badger sett; fingers crossed!"

Setting up Badgercam



01 Find a safe spot for the device, and make sure it doesn't impact on the badgers in the wild. Cameras and microphones are good passive sensors for this too.



02 Badgers are nocturnal, so you'll have to be very patient and wait until night to see some badgers live. Sometimes, waiting until the following morning works best.



03 Not quite a badger, but a night creature nonetheless. Tweak settings as you go to make sure you get stoatin' photos.

Roktrack – Pylon-guided mower

Here's a robot that takes grass and weed cutting to the next level... literally. **Nicola King** wonders at this weed assassin



MAKER

Yuta Suito

A software engineer, Yuta previously worked in Tokyo, but now lives in a very mountainous area. He wanted to create an outdoor tool to improve people's lives.

hackaday.io/yutasuito

Living in a mountainous region of Japan has proved inspiring for maker Yuta Suito. The local area has many rice terraces but, as he explains, "With the ageing of the farmers, this beautiful landscape will disappear in about ten years. Many farmers say that the hardest part of maintaining the terraced fields is weeding." Determined to try and help, Yuta wanted to "solve this problem by providing mobility (with no need to bury boundary wires underground) and intuitive operation (in reality, many elderly people cannot use smartphones or computers)."

The result is the lightweight, solar-powered Roktrack pylon-guided mower, and it's perfect for rough terrain. Prompted by a road construction site, Yuta noticed that "by simply enclosing the area with traffic cones, the people inside would understand the scope of the work. I was inspired by the idea that computers could do the same thing." Applying this logic to his project, Yuta's mower is able to mow a range of coned-off areas, with a 200 m² area, for example, taking around two hours. Not only that, but it's equipped with a speaker, so "if you are in the immediate vicinity, an audible voice will tell you when the mowing is done!"



Traffic cones act as a visual guide for Roktrack, helping it to work its way around an identified area

The word in weeding

With software written by Yuta, Roktrack is programmed to head towards a pylon (or cone) which it recognises through a camera image, and it then starts working its way towards the next one. Equipped with two blades, it cuts backs grass and weeds as it goes, leaving them on the ground. It's rain-proof and has eccentric front wheels which allow it to traverse less hospitable terrain than you'd find in your average garden. In addition, object detection means that two mowers operating in the same area won't collide.

Yuta used a Raspberry Pi 3 Model A+ to control Roktrack for temperature and power consumption reasons. "I wanted to cool the Raspberry Pi only with the fan provided on [Roktrack], not with a dedicated fan for Raspberry Pi. In addition, since this unit uses only one LiFePO4 (4S, 12.6 V) power supply for the control and drive systems, it was necessary to step down the power supply for Raspberry Pi."

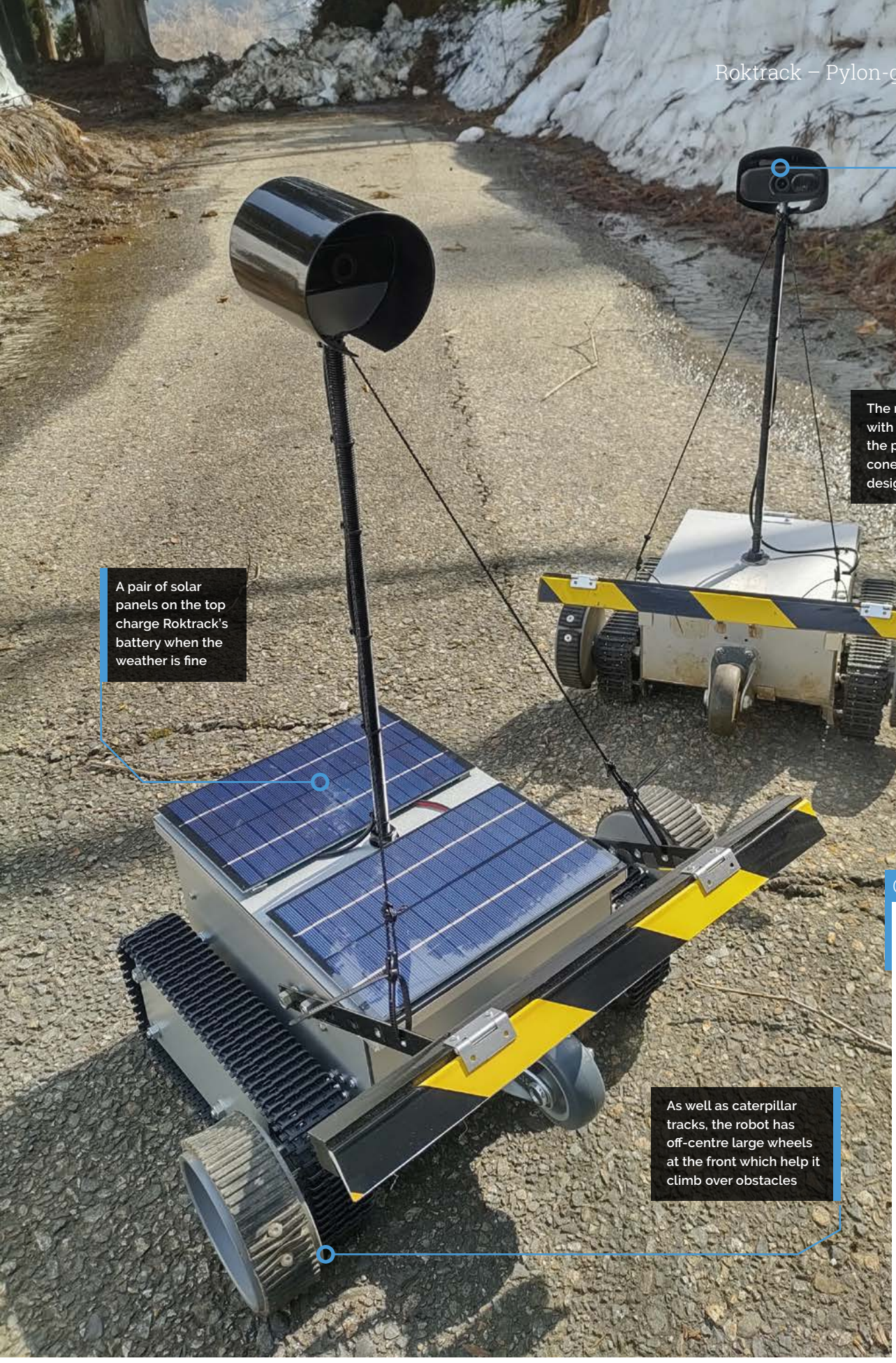
To start mowing, the user just presses the power button – or they can use an Android app, where communication between the robot and the app relies on Bluetooth Low Energy (BLE). Impressively, the app can instruct the mower to make an emergency stop, switch mode instructions, and, as Yuta details, "the coordinates



Warning!
Mower safety

Fast-spinning lawnmower blades are very dangerous and can also eject debris at high speed.

magpi.cc/mowersafety



The robot is equipped with a camera to spot the pylons (traffic cones) marking the designated area

A pair of solar panels on the top charge Roktrack's battery when the weather is fine

As well as caterpillar tracks, the robot has off-centre large wheels at the front which help it climb over obstacles

Quick **FACTS**

- Yuta has spent around two years on this project…
- …with more than one year spent building software using image recognition
- The raw materials for the current prototype cost around £275…
- …with trial and error leading to a total cost of around £1650
- Yuta's also considering creating a winter snow removal system



▲ An underside view of the robot shows its two rotary blades which enable pebbles to be ejected

of the past mowing point are recorded by the Android [device]'s GPS, and the current vegetation (weed growth) can be remotely sensed using Sentinel satellite images."

Finding solutions

Testing Roktrack was challenging, says Yuta. "It was very difficult to test the system until I could get it to work stably." He once took his eyes off of it while testing, and the robot tumbled down a cliff, but his conscientious hard work, testing thoroughly, and solving problems "one by one" now means, "I just leave it on and have a cup of tea in the house, and the weeds are cut." Yuta is currently developing what he calls "parallel operation with multiple units," and, by the end of the year, he hopes to experiment with up to six mowers working on several hundred square metres at a time.

The building of a robotic mower like this is probably not a suitable beginner project: for one thing the creation of the chassis gave Yuta some issues initially, but then he discovered a sheet metal service (**PCBWay.com**), which was able to send him "precisely machined" aluminium sheet for his project, and this solved a number of problems. "I don't think it would be easy to make it from scratch, as it would require a lot of tools and the number of parts is very large," he notes. "However, you can find the 2DCAD [file] of the body on Hackaday (**magpi.cc/roktrack**). If you use an aluminium sheet metal service, you can reduce the difficulty of the work."

A life changer

Pleasingly, Yuta's project has brought people together. "The field robots are very eye-catching and provide an opportunity to talk to many people, including passers-by," he notes. More than that,



▲ A couple of Roktrack robots in action on tricky terrain



Yuta is now ramping up production of his robot mowers to help out local farmers

> "I just leave it on, and have a cup of tea in the house, and the weeds are cut"

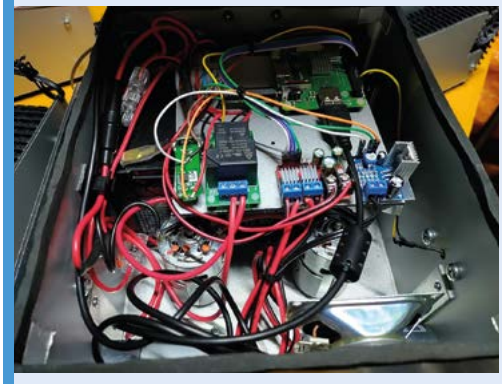
the project appears to have struck a chord with people and, when presented to the Japanese maker community, it became a "hot topic." Interviews followed and his local government has agreed to support the project, and five additional units will now be built to mow vacant lots in the community. "Together with the two I have built so far, I will have a total of seven units. I feel that this project is definitely starting to change my life," declares Yuta.

Japan is not the only part of the world with rice terraces and the inherent issues that rice farmers are facing, as other parts of Asia, where rice-eating is an important part of the culture, face potentially the same problems. Happily, Yuta has received well-deserved interest from potential investors, including "venture companies, mower manufacturers, and agricultural corporations. It would be great if this project becomes a business." Ingenuity and resourcefulness at its best – we wish him well!

Grass me up



01 Yuta's casing needed to be strong to weather the terrain, so aluminium sheet metal was chosen. He then installed motors, fans, and switches.



02 Yuta converted Raspberry Pi OS on his 3A+ to read-only. He also allocated it more memory so that it could handle the task (**magpi.cc/roktrackreadonly**). The mowing program is automatically started by systemd, and starts mowing on its own.



03 Yuta attached solar panels to the top of the fully assembled robot with double-sided tape. The battery can be charged from the panels. Under solar power, Roktrack can operate for up to an hour a day, if the weather is good.

Chip Bipedal Companion Robot

Chip is accomplished robot maker Kev's take on a cute companion. **Rosie Hattersley** hears his origin story



MAKER
Kevin McAleer

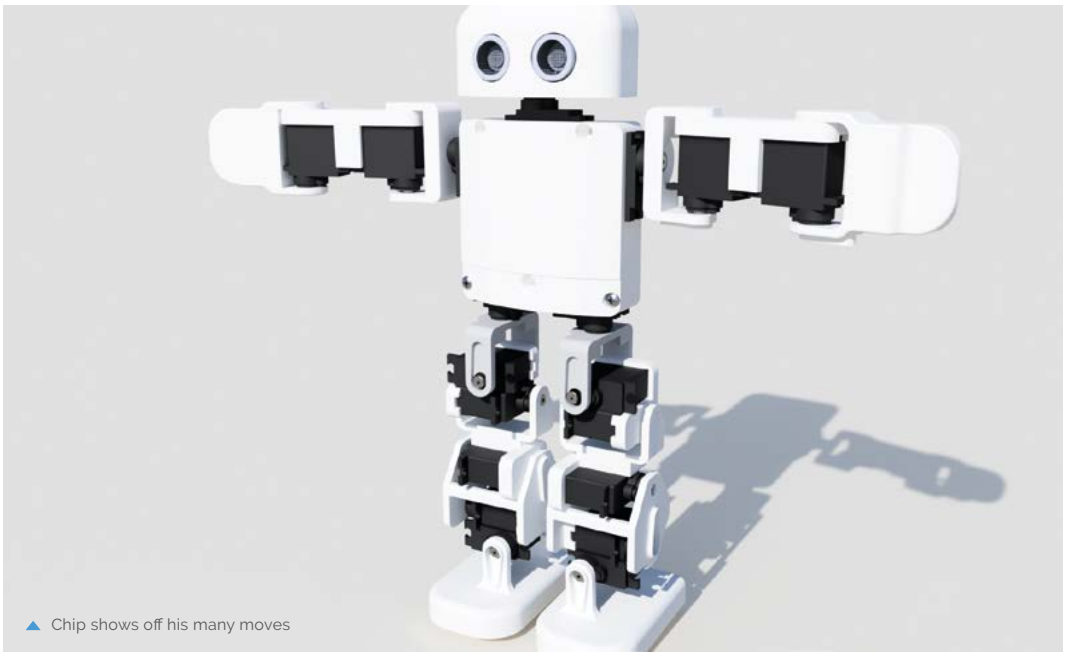
Kevin McAleer is a master of robotics, wowing us nearly weekly with a new robot.

kevsrobots.com

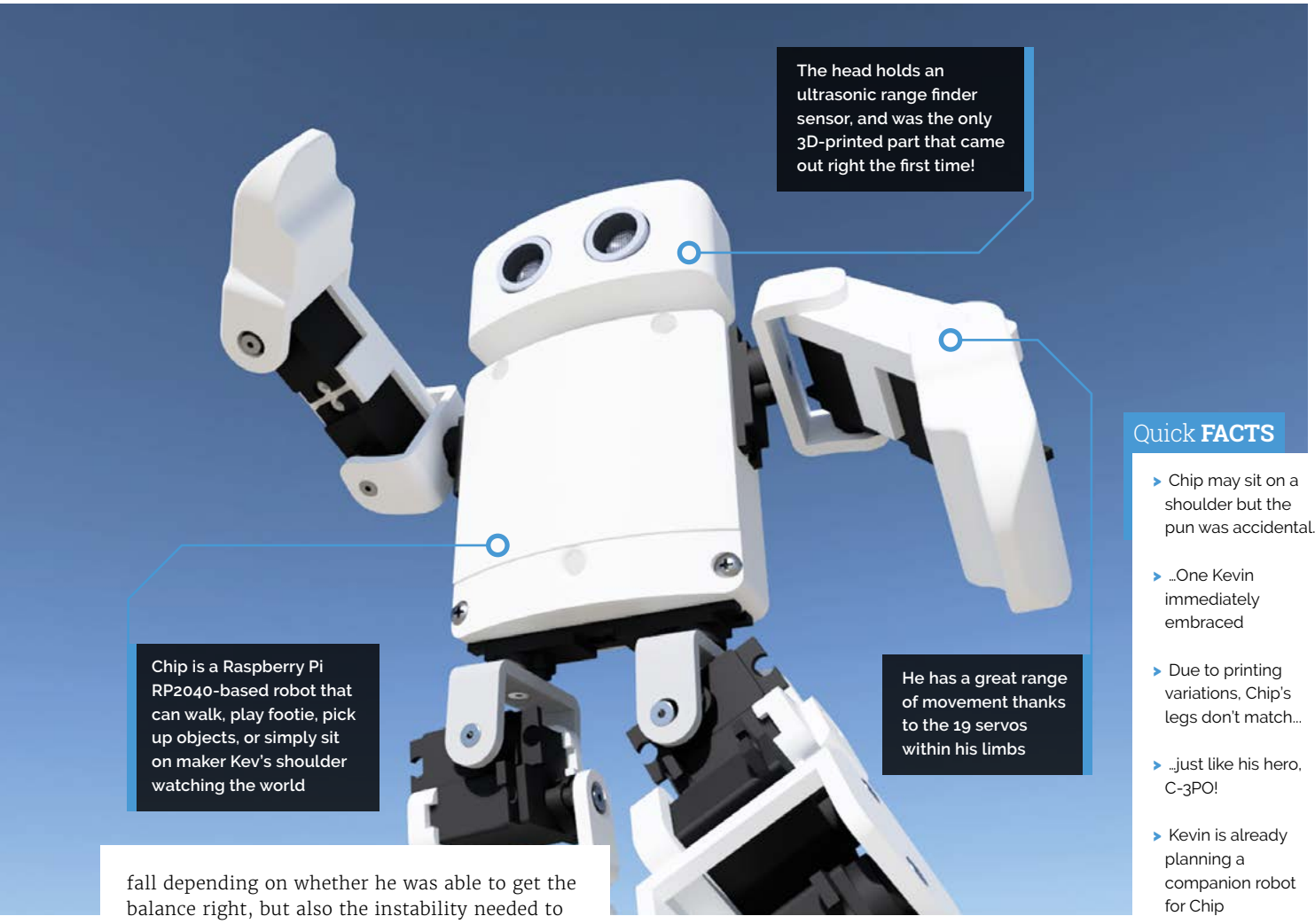
Many readers will be familiar with maker Kevin McAleer's endearing robots and other inventions, either through reading *The MagPi* or enjoying his wonderful YouTube broadcasts which enthuse about all things tech. Earlier this year, he designed and built a Pico-based companion named Bubo 2T. This second companion, Chip (**magpi.cc/chip**), is so-named because it will sit on its maker's shoulder (insert pun here). Whereas Bubo was a social media maven, tweeting out missives, Chip is more of a 'cute interactive humanoid pal'. He's got a Raspberry Pi RP2040 board inside in the form of a Pimoroni Servo 2040, chosen for its 18 servo sockets, and senses the world through an ultrasonic range finder.

Walk the walk

Kevin wanted to create a small bipedal robot so he could develop his expertise with walking and gestures. He relished the idea of designing something that would quite literally stand or



▲ Chip shows off his many moves

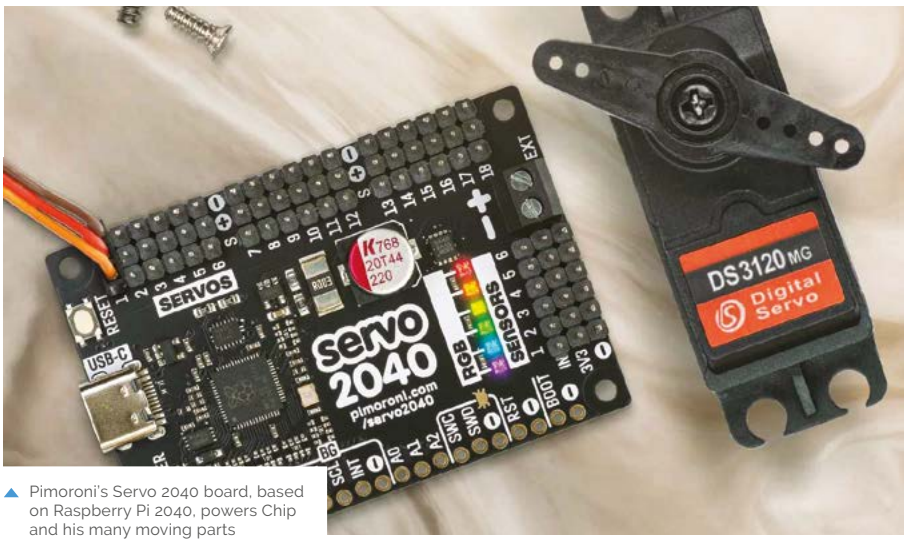


Quick **FACTS**

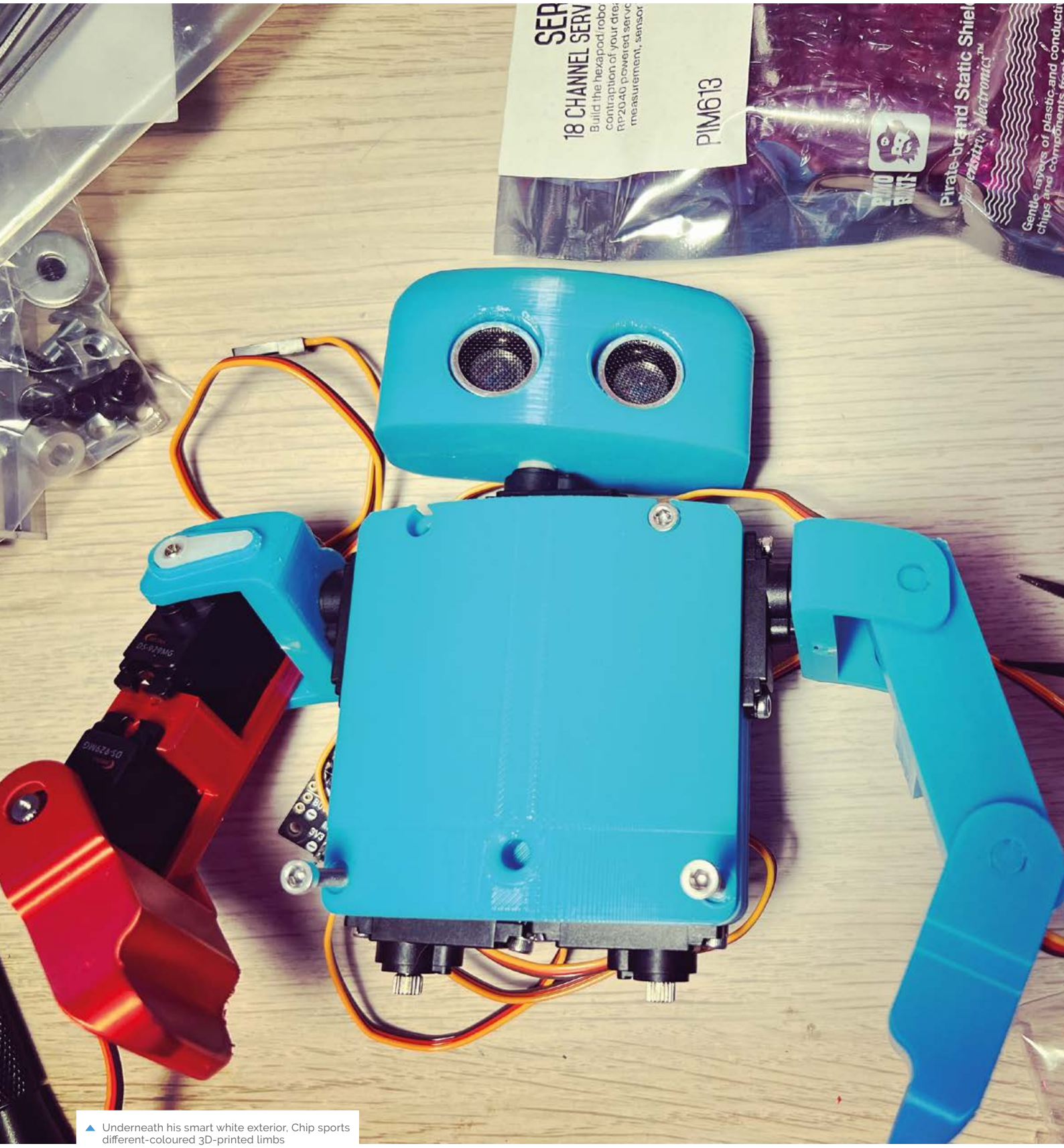
- Chip may sit on a shoulder but the pun was accidental...
- ...One Kevin immediately embraced
- Due to printing variations, Chip's legs don't match...
- ...just like his hero, C-3PO!
- Kevin is already planning a companion robot for Chip

fall depending on whether he was able to get the balance right, but also the instability needed to allow his electronic biped to walk. He envisaged a robot similar in looks to Plen (the first open-source, 3D-printable robot), with nods to Robosen (which makes the likes of Optimus Prime and Buzz Lightyear robots) and EZ-Robot, a self-assembly robot for educational use. Walking and smoothly extending the limbs were key, and meant Kevin would need to consider the robot's aesthetic design and be vigilant about 3D-printing tolerances.

The ambitious robot design will provide "18 degrees of freedom" proclaimed Kevin part-way through the build, "and move in 18 different ways." The hands, arms and shoulders, hips, upper leg, knee, lower leg, shin, and foot will all articulate, while the neck will be able to rotate. All of this necessitates a board with 18 servo sockets, hence Kevin's choice of Pimoroni's Servo 2040 board based on the RP2040 chip.



▲ Pimoroni's Servo 2040 board, based on Raspberry Pi 2040, powers Chip and his many moving parts



▲ Underneath his smart white exterior, Chip sports different-coloured 3D-printed limbs



A clear advantage for Kevin was being able to program everything in MicroPython, his first choice of programming language due to its simplicity. The straightforward code is pretty important as Kevin aims to put together a tutorial for Chip which all ages will be able to follow. He also rates Pimoroni's documentation and support, and says "Pimoroni includes some great servo classes that have servo easing to make motion really buttery smooth." Even so, he needed to adjust the tolerance (distance between the two arms between each servo) so the servos would move freely.

Rock solid foundations

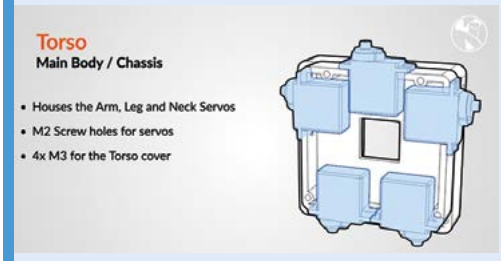
Kevin says using the RP2040-powered Pimoroni Servo 2040 board meant he could get straight to the interesting part of this project, designing the parts and constructing them. The Raspberry Pi RP2040 chip at its heart "has plenty of power and flexibility to enable me to add extra sensors, such as an IMU (inertia measurement unit), so the robot can balance and keep itself upright." However, it didn't all go quite to plan: "I burned out a couple of servos making this project; the joints were too stiff, and I cranked up the amperage to see if they could move it better, and magic smoke appeared."

> "Seeing a little human-like robot walking around is too much fun!"

Despite his experience in 3D printing and robot design, Kevin found he had to print several versions of each part to get the fit just right. "Designing the model in 3D doesn't always highlight the issues you find after 3D-printing the model," he explains. He was able to reduce the printing time as he has two 3D printers, but one uses red filament, so the pieces then didn't match, necessitating a complete body spray to give Chip the intended look.

We think Chip looks great and will be keeping an eye on Kevin's blog to see how the self-balancing feature based on the PID algorithm (**magpi.cc/pidtheory**) comes along. Having splurged on 19 servos costing £9 each, Kevin admits Chip is an expensive build, but loved bringing him to life: "Seeing a little human-like robot walking around is too much fun!"

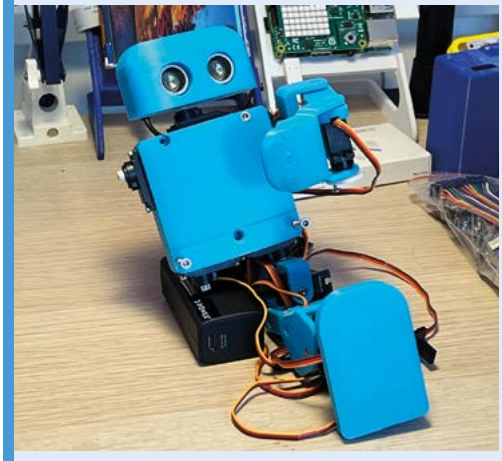
Construct a companion



01 Kevin details the build for Chip (Cute Humanoid Interactive Pal) here: **magpi.cc/chipgit**. You will need to 3D-print the parts; the STL files are at **magpi.cc/chip**.



02 The 19 servos are attached to a Pimoroni RP2040 Servo board via its headers, while attaching an IMU sensor helps keep Chip surroundings-aware and, therefore, upright.



03 The head unit's ultrasonic distance sensor doubles as eyes. The fully assembled Chip has 18 different degrees of freedom, but loves to gaze at the world from his maker's shoulder.

Bristol Braille Canute 360

Braille readers bring literacy to non- and partially sighted readers. **Rosie Hattersley** hears about a breakthrough Raspberry Pi version



MAKER

Ed Rogers

Bristol Braille MD Ed Rogers has used Raspberry Pi professionally and for his own interest since it first launched, and loves the community and support it provides.

bristolbraille.org

Screen fatigue is all too real for millions of smartphone and laptop info-seekers, but the details we get to read oh-so-casually are something most of us take for granted. Not so for blind and partially sighted readers. Text-to-speech engines do a great job of narrating on-screen information, but it's no substitute for being able to read what you wish to at your own pace.

Community interest organisation Bristol Braille set about changing this with its Canute 360 multi-line reader which has a Raspberry Pi at its heart. This Raspberry Pi Zero-based device is the world's first multi-line Braille reader and is capable of generating whole paragraphs of text at a time – a huge leap forward compared to the few words at a time Braillists are often accustomed to. Canute has 40 cells on each of its nine lines of text. It can read Braille Ready Files and Portable Embossed Files, so the world of reading now spans texts from Bookshare, the RNIB Library, and any other Braille library in the world. Excitingly, it can also open up the world of gaming.

Keeping it local

Founder and MD of Bristol Braille Technology Ed Rogers has been working on "refreshable Braille technology" since 2008 and co-invented the Canute multiline Braille display. He is also a co-founder and trustee of the Braillists Foundation, a community of more than 1500 people who help each other with projects to promote Braille use. Ed drew on this community's expertise when designing the Canute 360 to ensure it would meet their needs and expectations. Having closely followed Raspberry Pi's success for many years, he was convinced from the outset that it would be the ideal component for the embryonic Braille reader. "We didn't want to distract ourselves by

The brand-new Console model uses Raspberry Pi 400 and adds a BRLTTY screen reader in which Raspberry Pi converts Terminal window text to nine lines of Braille

Bristol Braille's Canute 360 is the world's first multi-line Braille reader, and is controlled by Raspberry Pi Zero



The large display and coloured text massively improve the legibility

Quick **FACTS**

- The Canute 360 contains 360 Braille dots
- It's the size of a coffee table book
- It can read any Braille-ready text file
- The new Console model has a built-in high-contrast screen
- So, both Braille and low-vision reading are supported





▲ The Canute Console Braille reader, based on Raspberry Pi, produces musical notation, charts, graphs, tables, and diagrams

reinventing the wheel, or having to support our own OS," so the dedicated OS and thriving Raspberry Pi community were vital factors.

Bristol Braille started with a standalone, "functional" book-reading prototype, says Ed, and the 14 major revisions over an eight-year development cycle each involved Raspberry Pi and Python. The folks at Raspberry Pi loved the concept, and Ed says individuals "bent over backwards" in a bid to ensure Canute made it to market. This is refreshing for a small company working in a supply chain that likes scale. While some components for the Canute are made in places such as China and India, the circuit boards are made by Philtronics in Wales, and the final build is done in-house in Bristol and Exeter. "By taking advantage of the proximity, we are able to be much more hands-on," says Ed.

Ed describes Raspberry Pi Zero as Canute's 'brains'. It runs Raspberry Pi OS, along with the Braille custom user interface (**magpi.cc/canuteui**). The interface was Bristol Braille's biggest design hurdle, especially since the final product needed to be relatively affordable. Although it has a £2000 price tag, many are destined for libraries and educational institutes and, thus, will have multiple users.



▲ Bristol Braille took their Braille readers on tour to garner feedback to ensure the new device would match everyone's needs



Raspberry Pi Zero is connected to media ports on the main circuit board, and is able to read the BRF (Braille Ready Files), interpret them as Braille, and send them to an Atmel chip on the main board to convert into dot patterns. Each Braille cell is made up of six dots which can either be up or down. There is virtually no margin for error, and a stuck dot can change the entire meaning of a word.

▲ The Canute 360 electronic Braille reader has ports including USB, HDMI, and SD card

Incredible touch

Bristol Braille's Canute 360 has been enthusiastically embraced by the Braillist community, with several hundred already sold. This summer, the team embarked on a roadshow demonstrating the 360 and its successor, the Canute Console. This is based around Raspberry Pi 400, meaning greater processing power as well as an integrated keyboard. A hardware add-on that includes a Canute 360, it adds a BRLTTY screen-reader. Ed says the Console enables them to do "far more exciting things with tactile graphics, using Raspberry Pi 400 to interpret sports matches as tactile animated pitches [and] to draw maps of cities in a top-down GTA 2 fashion."

> The circuit boards are made by Philtronics in Wales, and the final build done in-house in Bristol and Exeter

This recently launched model has a technical graphics focus, and is aimed at computer science students and those familiar with using Linux or willing to learn it. Every pre-existing multi-line Linux terminal application shows up in digital Braille for the first time. The new device has generated much excitement, with the first pre-orders currently being delivered. Watch the demo video at **magpi.cc/canuteconsole** showing a football match replay.

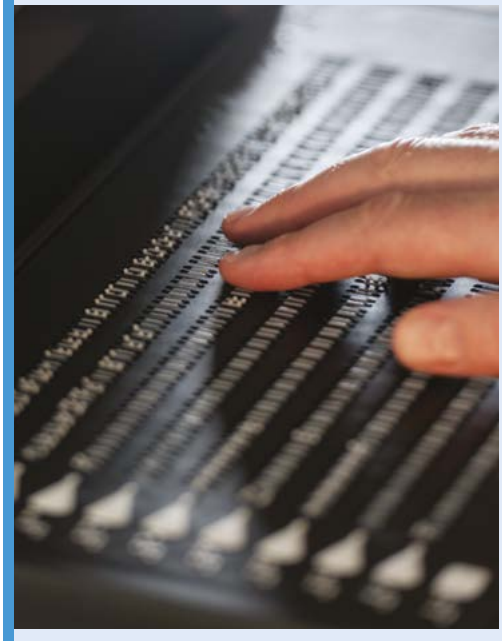
Brilliant Braille!



01 The Canute 360 comes pre-assembled, but there are GitHub setup instructions for the user interface at **magpi.cc/canutegit** so you can emulate it.



02 The Canute and Console readers can use transcription software, such as Duxbury DBT or Braille Blaster, on any digital text file. This can be imported via USB and read by either reader.



03 An Atmel chip on the main circuit board communicates with Raspberry Pi and converts text files into nine lines of Braille text.

ScreenDress

We'd definitely say yes to this dress – a stunning creation by Anouk Wipprecht that is proving to be an eye-opener, as **David Crookes** discovers



MAKER

Anouk Wipprecht

Anouk Wipprecht is a Dutch FashionTech designer and innovator who works and travels between San Francisco, Los Angeles, Miami, and Amsterdam.

anoukwipprecht.nl

It's fair to say that clothing fashion and computing don't always go hand-in-hand. But bring amazing talent such as Anouk Wipprecht on board and all eyes are definitely going to be on the wearer – as this striking ScreenDress shows to great effect.

As well as fitting snugly around the body, the 3D-printed dress includes a set of circular LCD screens, each of which looks like an eye. Connected to an EEG sensor – a four-channel BCI headset developed by neurotechnology company g.tec – they change according to how the wearer feels by reading brainwaves and working out a person's cognitive load.

To that end, the screens can show signs of stress, fatigue, and frustration. The more a person subconsciously feels their mental load is increasing, the wider the iris and pupils dilate. By making changes in real-time, Anouk, a Dutch FashionTech designer, says the dress is able show a direct correlation between a person's actions and their brain's reaction. One thing's for sure, it's certainly eye-catching.

> "The 3D-printed dress includes a set of circular LCD screens"

Dress to impress

ScreenDress was created from scratch. As well as the headset – called the Unicorn Headband – the dress was designed in Onshape (**magpi.cc/onshape**), a free, cloud-based CAD app. Anouk said it helped her to understand the look and feel of the possible embedded LEDs and how light reflected back on the body and space around it. "I used various Onshape design capabilities including mixed modelling, generative design, render studio, and in-context design to reference hardware elements," she adds.

The dress was outputted to an HP Jet Fusion 5420W 3D printer using a form of nylon. "This material is ideal for engineering-grade, white, quality functional production parts," Anouk continues, pointing to the components being light – an important factor when being worn.

Behind each of the HyperPixel 2.1 round hi-res screens that display the eyes are Raspberry Pi Zero 2 W computers. "There are six of them," Anouk says. They take information from the sensor that has been analysed in real-time and determine how the animated eyes appear on the screen.

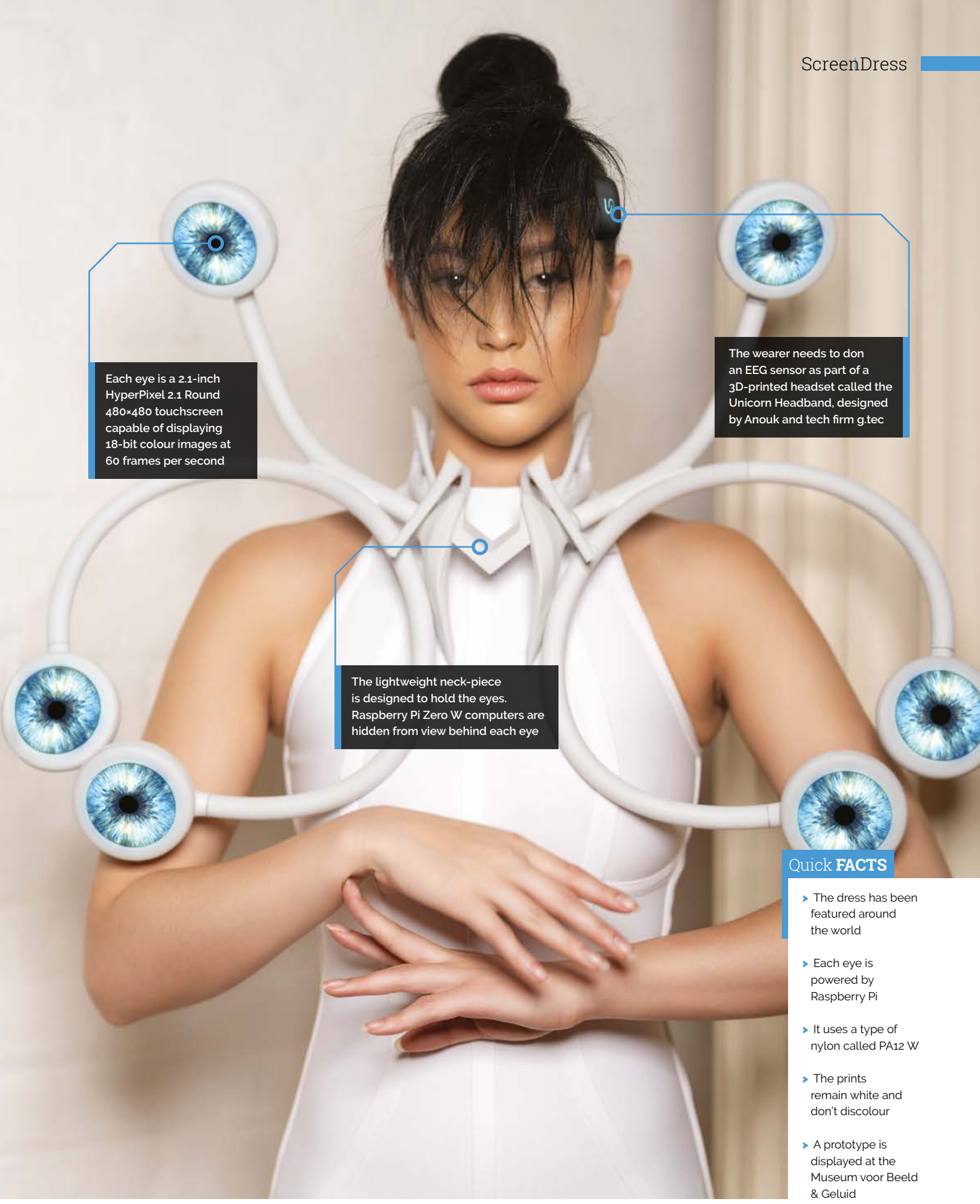
Got the look

To ensure the ScreenDress is in-tune with the wearer's brain, anyone wearing the clothing, which has the eyes flaring out from a sculpted neck-piece, has to undergo a two-minute training session. Machine learning then begins to understand the wearer's mental workload so that the eyes work as accurately as possible.

"Our fortunes turned when Raspberry Pi Zero 2 W, equipped with a quad-core Cortex A53 processor, was released," Anouk says. "This upgrade significantly enhanced the processing capabilities [over Raspberry Pi Zero W], allowing us to manage data seamlessly and achieve the desired performance."

ScreenDress is certainly cool and it turned a lot of heads when it was first put on show at the ARS Electronica Festival in Linz, Austria in September before being presented in Budapest, Hungary and Eindhoven, Netherlands. It has also posed a lot of questions.

"What does it mean when we can connect technological-expressive garments to our bodies, body signals, and even emotions? What dialogues can we trigger? This is what I am exploring with designs like these," Anouk explains. It also means that Raspberry Pi computers can be seen as a fashion accessory, potentially inspiring many other makers.



Quick **FACTS**

- The dress has been featured around the world
- Each eye is powered by Raspberry Pi
- It uses a type of nylon called PA12 W
- The prints remain white and don't discolour
- A prototype is displayed at the Museum voor Beeld & Geluid

Retro Pi World Radio

An old valve radio provides the ideal housing for this nostalgic Raspberry Pi build, finds **Rosie Hattersley**



MAKER

Raju Subramanian

Software engineer Raju loves tinkering with tech, assembling classic characters from LEGO and helping kids get to grips with code

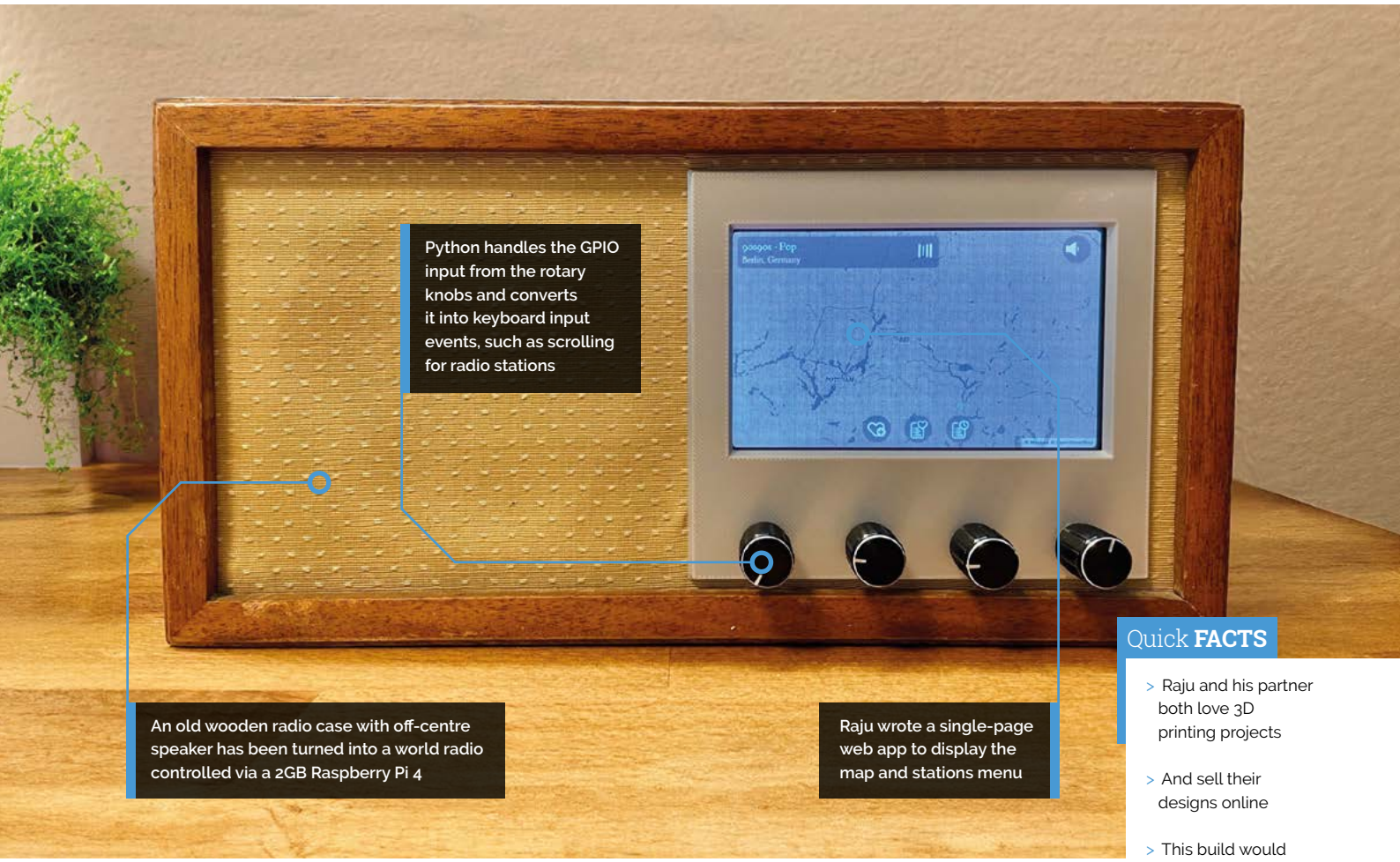
magpi.cc/worldradio

Most of the maker-interviewees who feature in The MagPi's project showcase pages have a deep love of technology and, often, a career in computing. Retro Pi World Radio creator Raju Salubramanian is just such a fellow, but he is also on a mission to teach the next generation to code: something that exactly chimes with why Raspberry Pi exists in the first place.

Raju describes himself as an AFOL (adult fan of LEGO) and, as well as building his own Optimus Prime, Hulkbuster and other complex kits, he has used the brightly-coloured bricks in his mission to teach his kids and their friends how to code. He does this as a 'coach' for the FIRST LEGO League (**firstlegoleague.org**) in his home town of San Jose, California. He is almost as big a fan of Raspberry Pi as he is of LEGO, with multiple projects under



He posted a video of his project on YouTube showing station selection



Quick **FACTS**

- Raju and his partner both love 3D printing projects
- And sell their designs online
- This build would cost roughly $275 if if you made it from scratch
- But Raju was able to repurpose many components he already owned
- Next, he's going to tackle a Raspberry Pi LEGO Robotics project

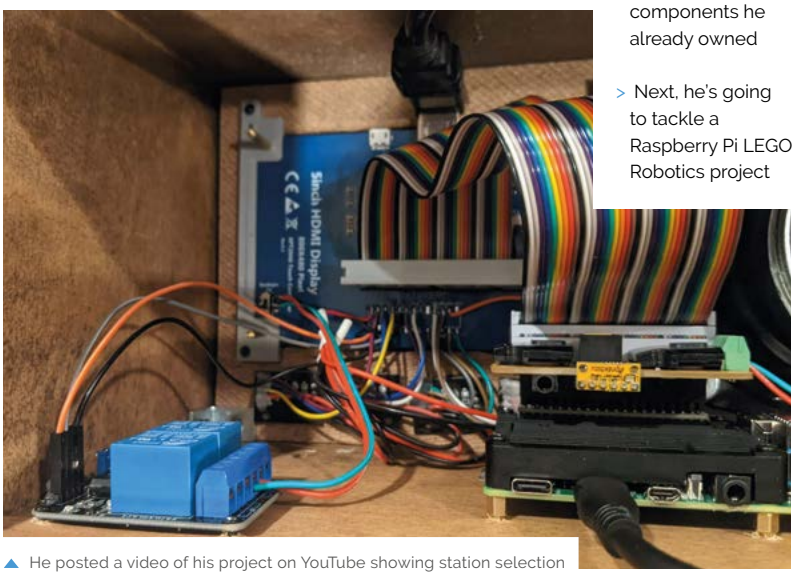
> "Raju wanted something that would force us to slow down in our current fast-paced digital world"

his belt and six Raspberry Pi boards in use around his home (not all disclosed to his partner!).

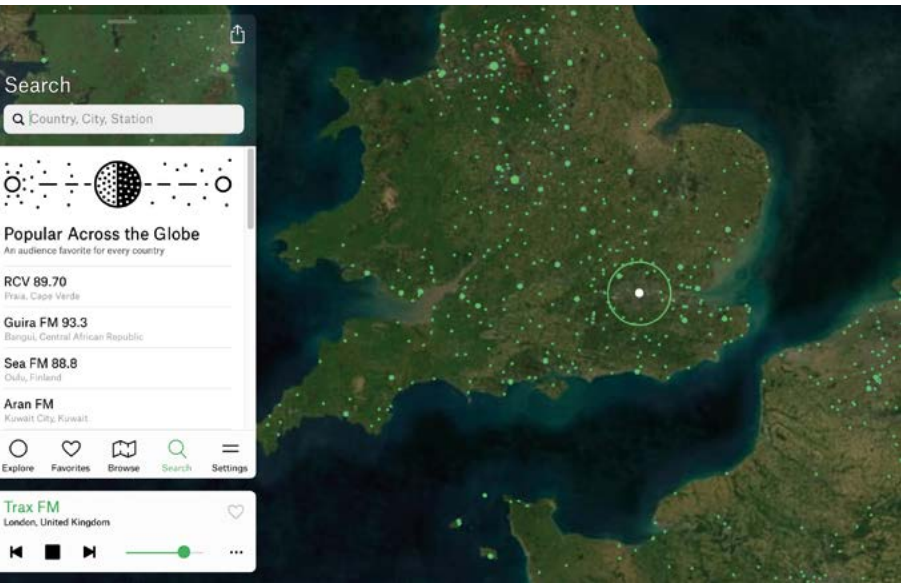
Despite his thoroughly modern career as a software engineer in Silicon Valley and a CV that lists VMware and Google among his credits, Raju has a fondness for the old days. "I have always loved to tinker and repurpose old hardware to build something new", he says, favouring Raspberry Pi boards because they are "perfect" for the projects he likes to work on since "they are so easy to setup and get started with".

A nod to nostalgia

For the retro radio build Raju wanted something that would force us to slow down in our current fast-paced digital world. He says the analogue controls of the radio are in some way the opposite of how we interact with our devices today with touch and voice. The project was also meant to



▲ He posted a video of his project on YouTube showing station selection



▲ Radio.Garden provides a user-friendly global station selector

remind Raju of his childhood of using analogue knobs to change stations. His lengthy search for the ideal old radio eventually led to him choosing one with a speaker to one side and enough space to cut out a window and fit a new 5in screen. His Retro World Radio can tune in to a vast collection of internet streaming radio stations from around the globe, and features a vintage map to browse and find radio streams.

Keep on improving

Raju had seen other retro radio upcycling projects online, and was particularly keen on creating one similar to this internet radio project: **magpi.cc/internetradio**. He used a 2GB Raspberry Pi 4 because he "needed the extra horsepower of Raspberry Pi 4 to get experience with the map display". He bought the main components from online stores such as Adafruit.

Sourcing a suitable radio and writing the software interface were his main challenges,

> Retro World Radio can tune in to a vast collection of internet streaming radio stations

since backend engineering is his professional forte. "I enjoyed the experience of writing all the Javascript code for the display and I am still thinking of ways to improve the overall aesthetic. Raju designed the new parts himself including a new front face for the radio, using CAD software to design and create some



▲ The searchable Radio Garden site serves up stations from around the globe



3D-printed parts. He also wrote software for the Retro World Radio, making liberal use of existing SDKs and open source or free online libraries.

He began with a development platform using a breadboard, which allowed him to test the software. The first version had three rotary knobs mapped to Raspberry Pi's GPIO pins, but Raju later decided to use four knobs. These pull in internet streams from Radio.Garden and can be chosen by location as well as genre.

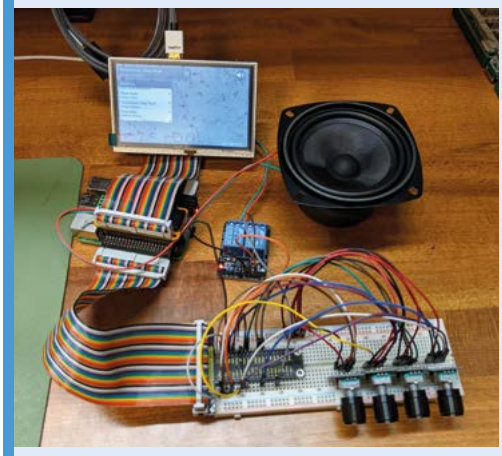
Having completed the radio project, Raju is now gearing up to combine his love of tinkering with tech with LEGO, and has recently acquired a Raspberry Pi Zero W.

▲ Raju bought an old radio from eBay for its off-centre speaker and warm acoustics

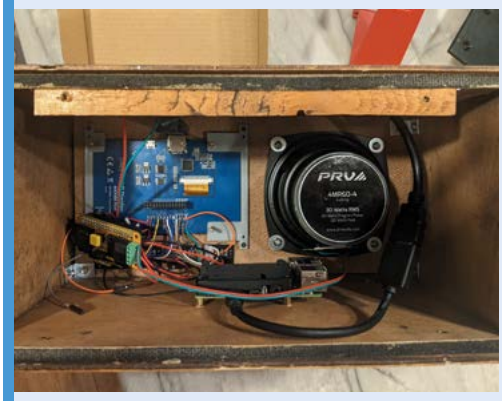


▼ He used the remaining cabinet space for the new radio menu

Rig up a radio



01 Having sourced a lovely vintage radio through eBay, Raju replaced the old backing board and fitted new rotary encoders and fitted a Raspberry Pi and AudioDAC.



02 He used the preconfigured version of Raspberry Pi OS Lite he found on Comitup to launch an automatic wireless connection: **magpi.cc/comitup**



03 NodeJS controls backend functions and serves up static images for the display. Raju's build details are at **magpi.cc/worldradio**.

BlueROV2 R4

This Raspberry Pi-powered submarine ROV safely explores the depths. By **Ashley Whittaker**



MAKER

Rusty

Rusty started out with a solar-powered surfboard trying to autonomously head from LA to Hawaii and Blue Robotics grew to develop hundreds of low-cost marine components.

bluerobotics.com

BlueROV2 is an underwater ROV that surveys the darkest depths – well, the really very gloomy depths anyway – with the help of a Raspberry Pi 4 wearing its bespoke Navigator Flight Controller as a HAT.

What is an underwater ROV?

If you're not sure what an underwater ROV is, the short answer is: a Remotely Operated Vehicle you chuck in the sea. But if you're our favourite kind of geek and wish to feast on a much longer answer, Blue Robotics has shared a short history and explanation of how they work, including the excellent tidbit that the first ever ROV was called the Poodle (**magpi.cc/whatisrov**).

BlueROV2 is billed as "the world's most affordable high-performance underwater ROV." While $3,950 seems like a lot, this ROV buying guide quotes prices up to $40,000 (**magpi.cc/rovbuying**). We like to think our affordable Raspberry Pi brain had a lot to do with bringing that cost down

What's it made of?

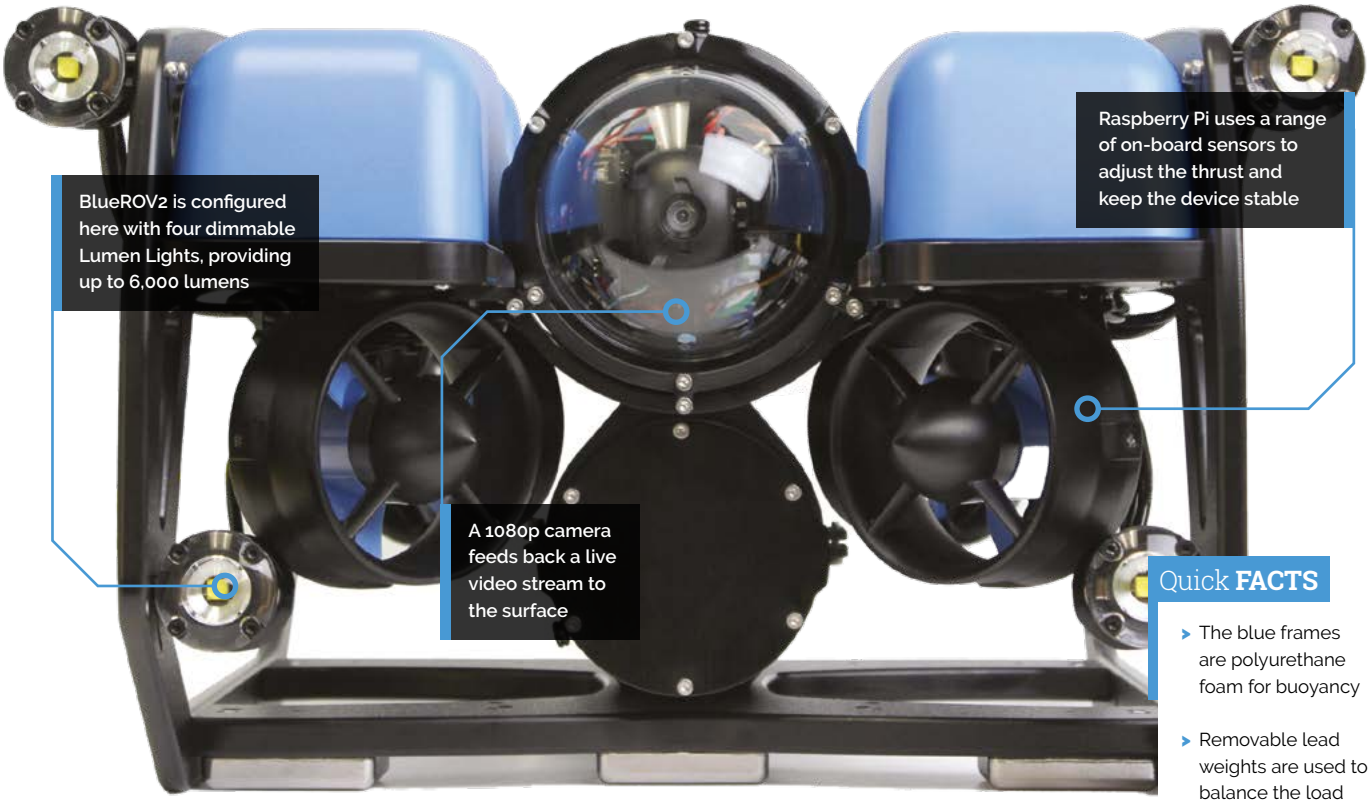
The device measures 18 × 13.3 × 10 inches and can capture live 1080p HD video, thanks to a wide-angle low-light camera mounted to a tilt mechanism on the front. An on-board gyroscope, accelerometer, and magnetometer, along with pressure, depth, temperature, voltage, and leak detection sensors, make sure everything is functioning and navigating as it should.

> "Splash out on the aluminium option and you're safe to dive up to 300 metres"

If you choose a ROV with an acrylic enclosure, you can get to a depth of 100 metres, but splash out on the aluminium option and you're safe to

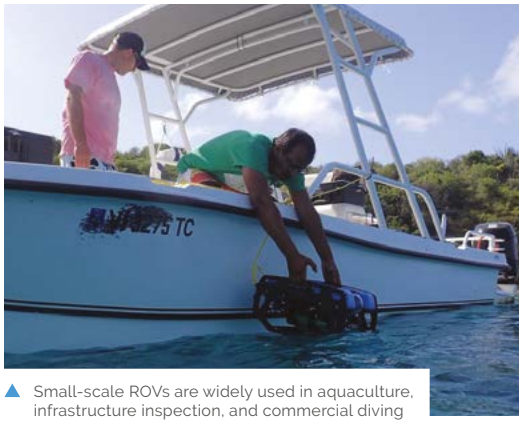


▲ ROVs are often used by police and fire departments for search and rescue missions, looking for missing people, ships, cars, and any other sunken objects



Quick **FACTS**

- The blue frames are polyurethane foam for buoyancy
- Removable lead weights are used to balance the load
- There are six to eight thrusters to control movement
- Raspberry Pi and the Navigator HAT control stability
- It runs BlueOS, a custom operating system



▲ Small-scale ROVs are widely used in aquaculture, infrastructure inspection, and commercial diving

dive up to 300 metres. As far as we can tell, that's deep enough to explore right down to the sea bed over much of the North Sea and almost all of the Irish Sea. Feel free to Google other bodies of water yourself.

What does its Raspberry Pi do?

The BlueROV2 is controlled by the Navigator Flight Controller, which is a HAT designed for Raspberry Pi 4. Raspberry Pi runs Blue Robotics' bespoke BlueOS software and handles all of the processing and computing requirements inside the ROV (**magpi.cc/navigatorhat**).

All of those sensors are provided by the Navigator, and Raspberry Pi 4 communicates what's happening back to base on dry land via BlueOS.

While we do think BlueROV2 is cool and all, we're still too traumatised by the shark chase at the beginning of *The Little Mermaid* to watch any of the video it captures. M



▲ A standard Xbox controller can be used to control BlueROV2

CinePI XL

CinePI is an open-source cinema camera that lends itself to DIY design, including larger sensors, learns **Rosie Hattersley**



MAKER

Csaba Nagy

Csaba's extensive interests cover technical and artistic domains including programming, hardware development, 3D printing, photography, and filmmaking.

github.com/CinePI

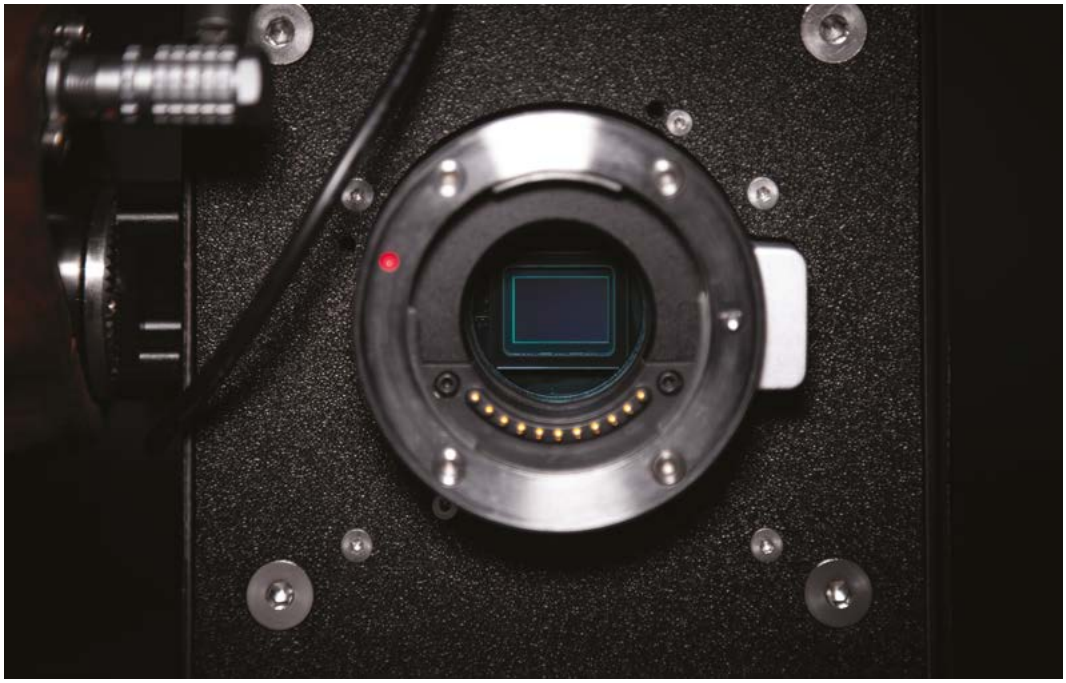
Canadian creative maker Csaba Nagy spent hours during his university course immersed in the world of camera sensors and computer vision, nicely complementing his degree in computer science.

Raspberry Pi's ability to connect with so many other devices, plus the possibilities of the camera module, piqued his interest. He subsequently developed CinePI – an open-source project that was "a direct result of the intersections between my passions for programming and filmmaking" – to add cinema camera features to Raspberry Pi. In early 2023 Csaba launched version 2, based around a Raspberry Pi 4 and 12.3 megapixel HQ Camera (**magpi.cc/CinePI**). Now he's back, with CinePI XL.

Big ambitions

Csaba says that CinePI was always intended to push the possibilities of Raspberry Pi as a video camera with features found in cinema cameras, but in an open-source device so developers could integrate it into their own designs. CinePI V2 can record 2K RAW Cinema DNG video at frame rates up to 50 fps with 12-bit colour depth that far surpasses the 8-bit or 10-bit usual video fare. Csaba showcased its abilities, using CinePI V2 to film more than half of his *Timekeeper* short film this summer: **magpi.cc/timekeeper**.

Having completed CinePI V2 in early 2023, it wasn't long before Csaba began work on developing this latest version, with a Compute Module. He has been inspired all along by the



Csaba is a big fan of Blackmagic Design's camera housings

A 2.8 in WaveShare DSI screen also distinguishes CinePI XL from the HyperPixel 4.0 display on CinePI V2

The Blackmagic camera box encases a Raspberry Pi Compute Module – and CM4IO board necessitated by the sensor's need for a MIPI CSI-2 link with four data lanes

Behind CinePI XL's machined metal lens turret is a OneInchEye sensor that can capture 12-bit video footage and photos at 20.3MP, even in low light conditions



CinePI XL is much larger than its CinePI V2 stablemate

Quick **FACTS**

- Csaba learned about the OneInchEye sensor and its maker…
- …on a Raspberry Pi forum thread: **magpi.cc/overclockcsi2**
- Meanwhile, Will had discovered Csaba's Discord server…
- …where he posted more details about the impressive sensor
- And Csaba realised it would be ideal for CinePI XL



▲ As intended, CinePI has inspired many takes on how such a Raspberry Pi cinema camera might look

cameras made by Blackmagic Design. "They were one of the very first manufacturers to offer RAW video capabilities at affordable prices; democratising tools used for filmmaking." This means cameras cost a couple of thousand dollars rather than many thousands. If, as Csaba intends, you were to build your own CinePI camera, it would cost far less again, and can largely be built with off-the-shelf components and sensors from the likes of Adafruit. He estimates CinePI V2 costs $250–$300 while CinePI XL in the same or similar configuration would be closer to $400–$600. "A steeper cost, but still very affordable in comparison to the types of camera it would aim to compete with on the market."

> "People have been using CinePI, modifying and customising it for their unique needs"

Whereas CinePI V2 was a compact form factor handheld camera, the XL model is noticeably larger. That's because it features a specially developed sensor, OneInchEye, designed by Will Whang (**willwhang.dev**). This larger sensor enables greater image quality and higher frame rates than the Raspberry Pi HQ Camera and also necessitated a switch to Compute Module because it required a MIPI CSI-2 link with four data lanes. Csaba decided to embrace the possibilities this offered in terms of additional sensors and design a camera that mimicked the feature set and capabilities of real professional cinema cameras. Advantages included a larger battery, bigger display, more mounting points for accessories, and a larger physical lens mount.

Community spirit

This summer Csaba saw membership of his Discord server grow from 300 to 1000 members, "It's been amazing to see the ways people have been using CinePI, modifying and customising it





▲ A side view of the CinePI XL camera

for their unique needs. Exactly what I was hoping for when I started the project." With the launch of Raspberry Pi 5, Csaba expects to see even more interest. "Perhaps in the future, with the efforts of talented individuals from the community or established companies already in this space, we can see the production of larger sensor modules that will work on the Raspberry Pi platform," he suggested to CineD: **magpi.cc/cined**.

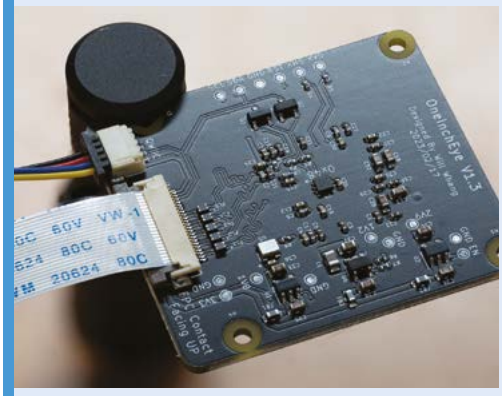


▼ *Timekeeper* was shot largely using CinePI 20.3MP cameras

Extraordinary scenes



01 Csaba's lens turret was repurposed from another camera. He encourages others to design their own CinePI. You will need a Raspberry Pi Compute Module, CM4IO board, connectors, a screen, and battery.



02 CinePI XL uses a OneInchEye sensor. Code for it is built on top of Raspberry Pi OS 64-bit Lite using libcamera-apps / libcamera. Setup instructions for the original CinePI are at **github.com/CinePI**.



03 Csaba details the core build elements at **magpi.cc/CinePIbuild** and will add CinePI XL instructions in due course. Camera design resources are at **github.com/schoolpost**.

Volcano-monitoring Ultraviolet PiCam

Volcanic activity is dramatic but often devastating for nearby communities. **Rosie Hattersley** finds out about a Raspberry Pi-based alert system



MAKER

Tom Wilkes

Dr Tom Wilkes is a lecturer in volcanology at the University of Sheffield, where he has used Raspberry Pi for several monitoring projects.

volcanotech.org

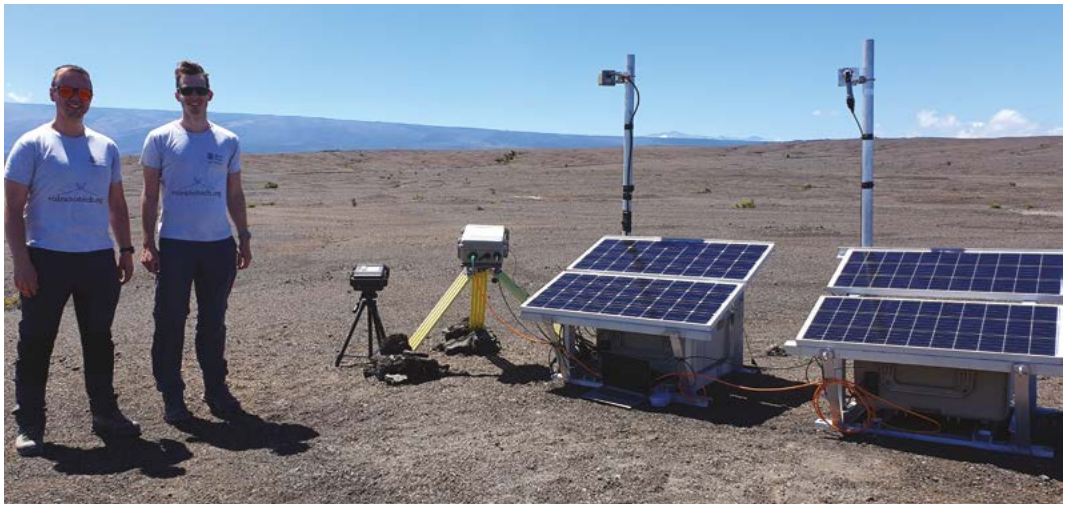
Iceland is currently on standby for imminent eruptions, while Sicily's Mount Etna gave off stunning emissions late one summer. However, most volcanic activity takes place in lesser-known, less touristy and often poorer nations, where preparations for a big blowout are far harder to facilitate. Furthermore, their remote locations can make installing equipment to track seismic activity an expensive challenge.

"UV cameras provide invaluable data for both monitoring and research," explains University of Sheffield volcanologist Dr Thomas Wilkes, "but funding in volcanology, particularly for monitoring organisations, is often limited, prohibiting their widespread adoption." For his PhD thesis, Tom set about designing low-cost ground-based remote sensing instruments using Raspberry Pi 3B+, having initially been attracted by the £20 Camera Module after a tip-off from a colleague (also called Tom). He read up on all things Raspberry Pi using our forums and online resources and was able to publish both his research paper and set up his first sulphur dioxide monitoring camera in 2016. Two images are taken – one in which UV light is absorbed and one where it isn't. Overlaying the images reveals where sulphur dioxide – a corrosive, acidic gas – is being emitted and in what quantity (**magpi.cc/so2**).

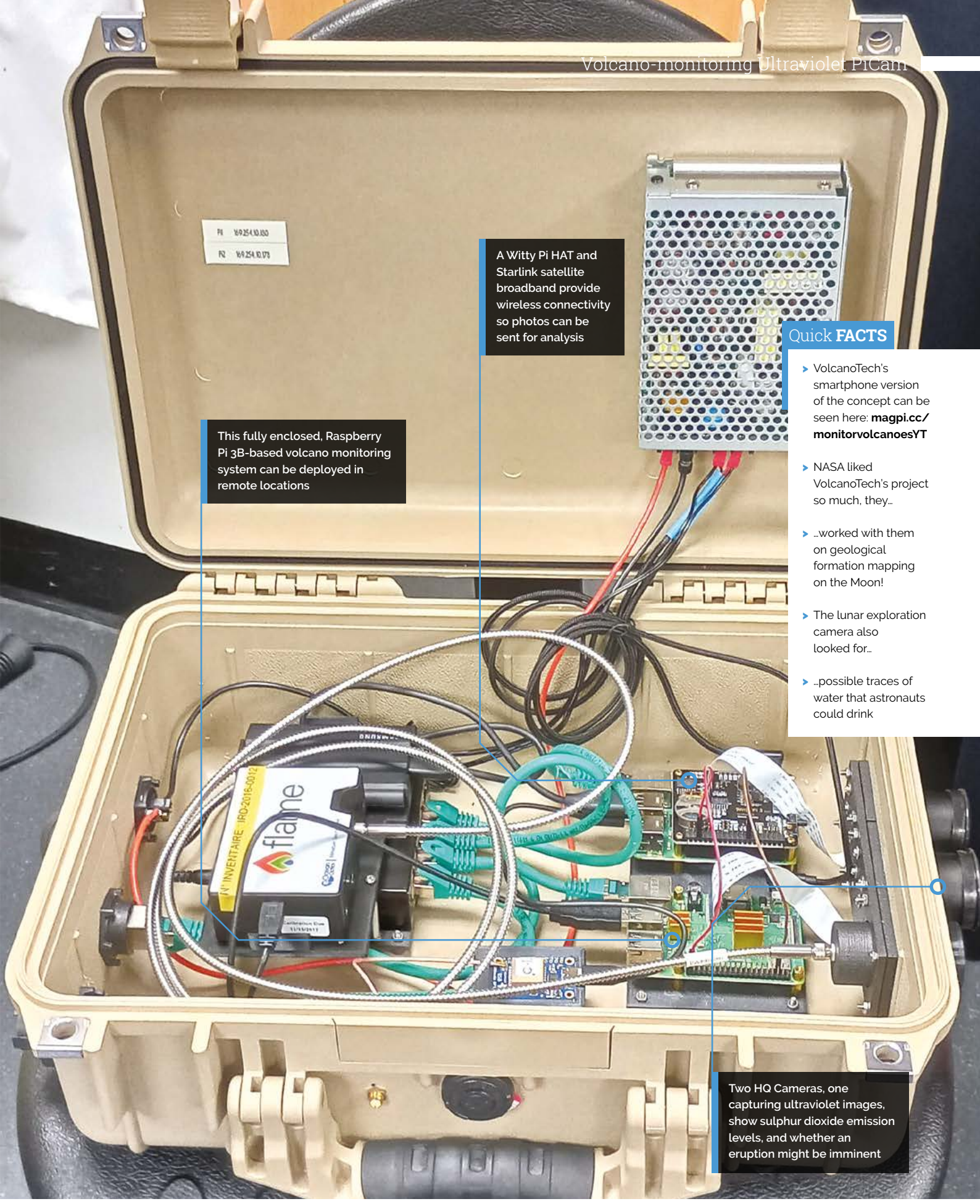
The two Toms continue to work together and are both members of Sheffield-based VolcanoTech (**volcanotech.org**). They aim to get more instruments onto volcanoes and acquire unprecedented long-term datasets that contribute to volcano research and monitoring. Designing and constructing low-cost scientific instrumentation is crucial for developing countries, where funding for equipment can be quite limited but where many of the world's active volcanoes are located.

Tell-tale signs

A volcano's degassing behaviour changes before an eruption, often increasing or decreasing. Since

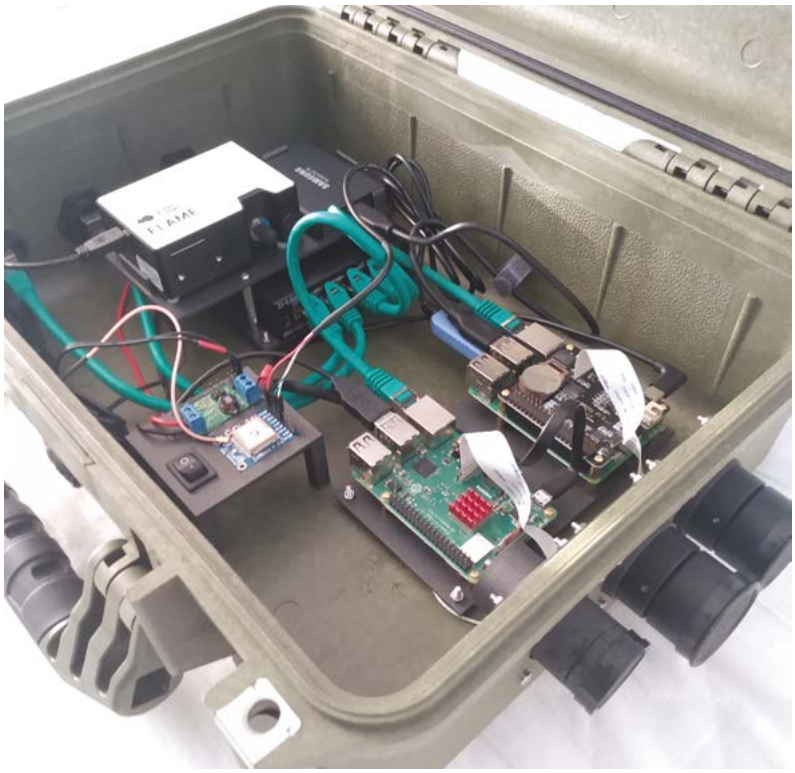


Tom Wilkes and colleague Tom Pering beside their UV camera rigs monitoring Kilauea volcano, Hawaii



Quick **FACTS**

- VolcanoTech's smartphone version of the concept can be seen here: **magpi.cc/monitorvolcanoesYT**
- NASA liked VolcanoTech's project so much, they...
- ...worked with them on geological formation mapping on the Moon!
- The lunar exploration camera also looked for...
- ...possible traces of water that astronauts could drink



▲ The weatherproof Peli case provides a robust housing and enough space inside to mount everything

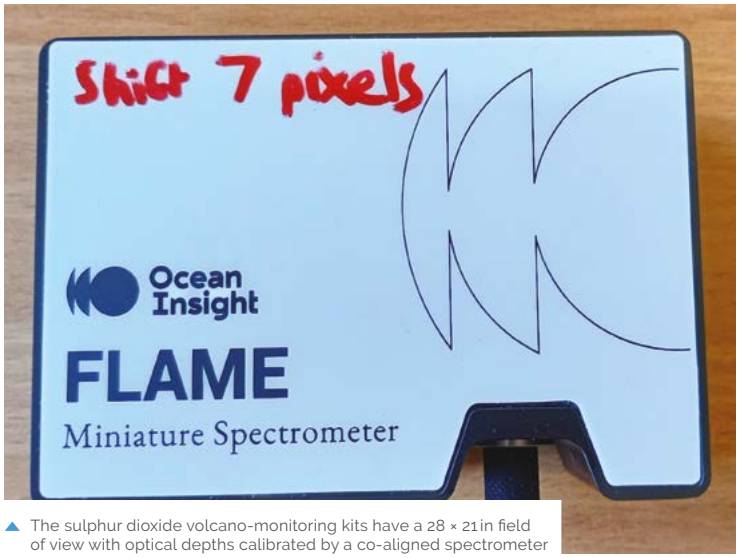
sulphur dioxide is the gas most prevalent at active volcano sites, Tom focused on monitoring this using a UV camera. There were already scientific-grade UV cameras in the field in some locations, but they cost tens of thousands of pounds, whereas Tom aimed to develop a UV camera "an order of magnitude cheaper" and much smaller, making it suitable for permanent installation and remote reporting. Monitoring volcanoes often involves tracking how sulphur dioxide diverges from a baseline and then investigating those changes. Tom's system uses the Beer-Lambert law of absorption, comparing the optical depth of plume pixels in an image to the background sky radiance (**magpi.cc/beerlambertlaw**).

Tom and his colleagues didn't know much about Raspberry Pi at the outset, "which made the detailed knowledge and wealth of experimentation by others invaluable when it came to asking specific questions or pulling ideas from pre-existing threads." A particularly fruitful discovery was a YouTube video by Les Wright (**magpi.cc/leslabYT**) which simplified the process of removing the Bayer filter from the camera lens to enhance its sensitivity to ultraviolet light. This worked well for a Raspberry Pi Camera V1.3, but Tom reports that he was unable to remove the Bayer filter in the same way for the newer Raspberry Pi Camera Module. The process in any case requires a fume cupboard and dangerous chemicals (see 'Warning' box). Tom originally made use of the UV camera in a spectrometer he used for sulphur dioxide monitoring. Although this worked well, he subsequently decided to focus on the camera-based system.

> "Tom aimed to develop a UV camera an order of magnitude cheaper"

How predictable

Having created an affordable alternative to the £10,000 scientific cameras, Tom has been working on modifying the camera design so they can be deployed permanently on volcanoes, without the need for human interaction. To date, only Stromboli in the Aeolian Islands, and Etna just to the south, have fixed monitoring. "We are now beginning to build high time-resolution sulphur dioxide emission rate datasets from several volcanoes, the likes of which are quite rare in volcanology," he says.

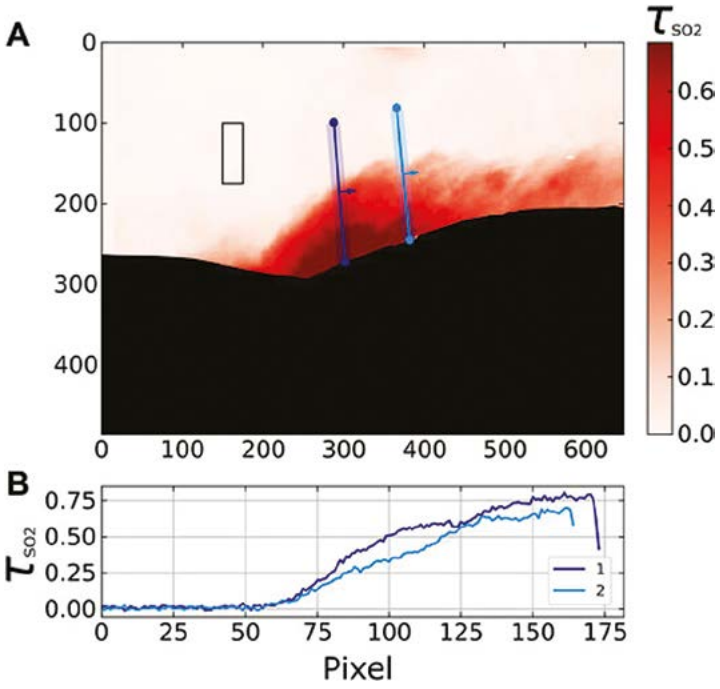


▲ The sulphur dioxide volcano-monitoring kits have a 28 × 21 in field of view with optical depths calibrated by a co-aligned spectrometer

Warning!
Strong chemicals

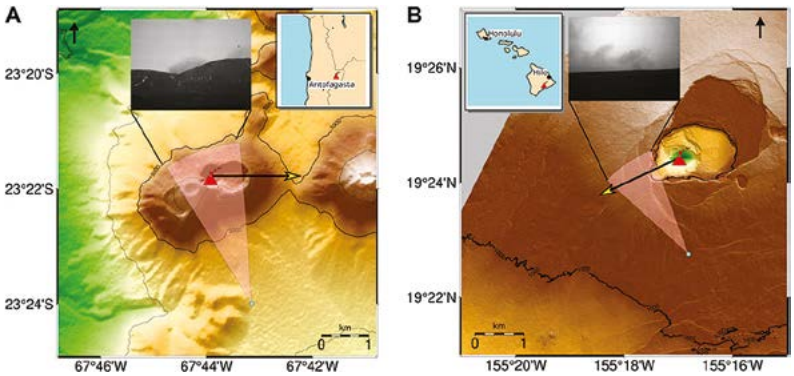
De-Bayering the sensor requires dangerous chemicals such as nitric acid and carcinogenic dichromates. It should not be done at home. It requires a fume cupboard and proper chemical disposal procedures. An alternative is to simply scratch the Bayer filter away with something like a piece of plastic – this can have a reasonably good result but may not clean the sensor up quite as nicely as the chemical procedure.

magpi.cc/chemicals



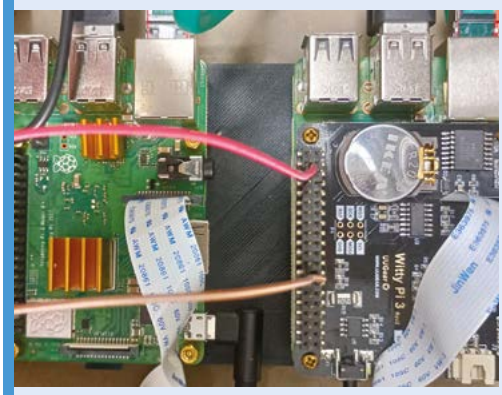
▲ Optical depth image captured at Lascar volcano near Buenos Aires with a control rectangle of clear sky

Getting the cameras installed on hazardous volcanoes is no mean feat! Using a Starlink satellite connection has helped them overcome the major issue of handling and processing the terabytes of data each camera acquires each year, as well as partially solving issues relating to debugging camera installations remotely. With cameras in the Amazon rainforest as well as the Atacama desert, temperature extremes and huge amounts of rainfall add to the challenges. However, once in place, they'll give volcanologists significantly improved monitoring capabilities that contribute to longer-term volcanological research – an increasingly critical aspect of their work "since a volcano can exhibit significant changes in activity over a wide range of time scales."

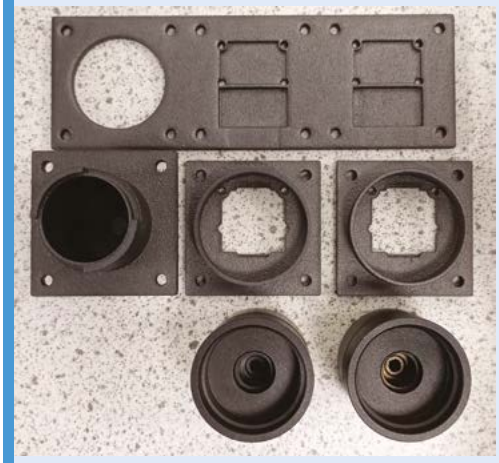


▼ Digital elevation models (DEMs) of Lascar and Kīlauea; the blue dots are camera positions

Glowing report



01 A modified Raspberry Pi camera is fitted with an ultraviolet transmissive optical system, designed in Zemax ray-tracing software. Bandpass filters compensate for changing wavelength transmission and light incidence angles.



02 The triplet mount was made from machined black anodised aluminium, resulting in improved precision and longevity over previous 3D-printed, non-triplet versions.



03 A UV spectrometer, WittyPi HAT, and network drive manage data and send it back to the volcano monitoring team for analysis.

PeggyBoard

Climbing walls are already pretty exhilarating but that didn't stop one maker from upgrading his with a Raspberry Pi. **Rob Zwetsloot** gets to grips with it.



MAKER

Pegor Karoglanian

An embedded software engineer who likes to take on a variety of projects in his downtime. His goal is to get others excited to make something.

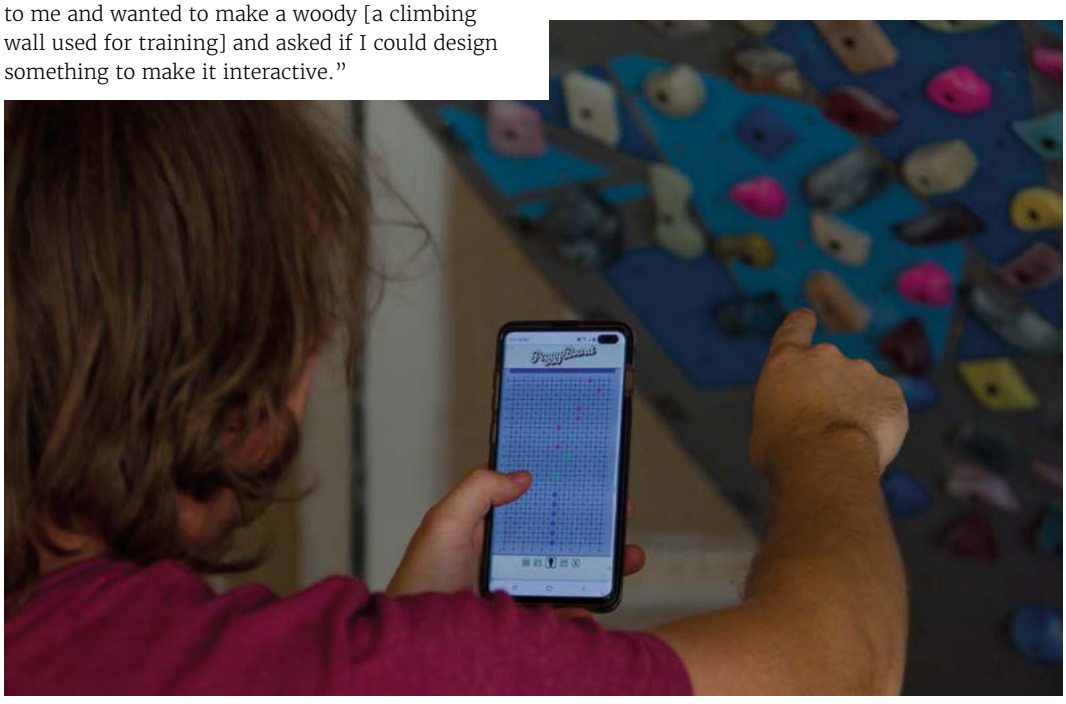
github.com/PegorK

During the various lockdowns at the start of COVID, people picked up various hobbies. Baking sourdough bread, setting up a new island in *Animal Crossing*, and getting some home exercise. Instead of deadlifts and HIIT, however, one maker helped a friend out with a special piece of equipment.

"[It's] called the PeggyBoard," Pegor Karoglanian tells us about his creation. "It's an interactive climbing wall powered by a Raspberry Pi that allows the user to set specified routes using a web app which also allows them to save routes or load old ones. It was originally developed when the COVID-19 pandemic first started and the local climbing gym had to shut its doors to follow quarantine protocols. My friend Liam reached out to me and wanted to make a woody [a climbing wall used for training] and asked if I could design something to make it interactive."

This is not a new idea – in fact Pegor mentions a pre-existing product called MoonBoard which he drew inspiration from – however making it becomes part of the fun. Also, it's significantly cheaper too.

> "The LEDs all light up properly, the web app is easy to use... and the database is very useful to go back to saved climbs"



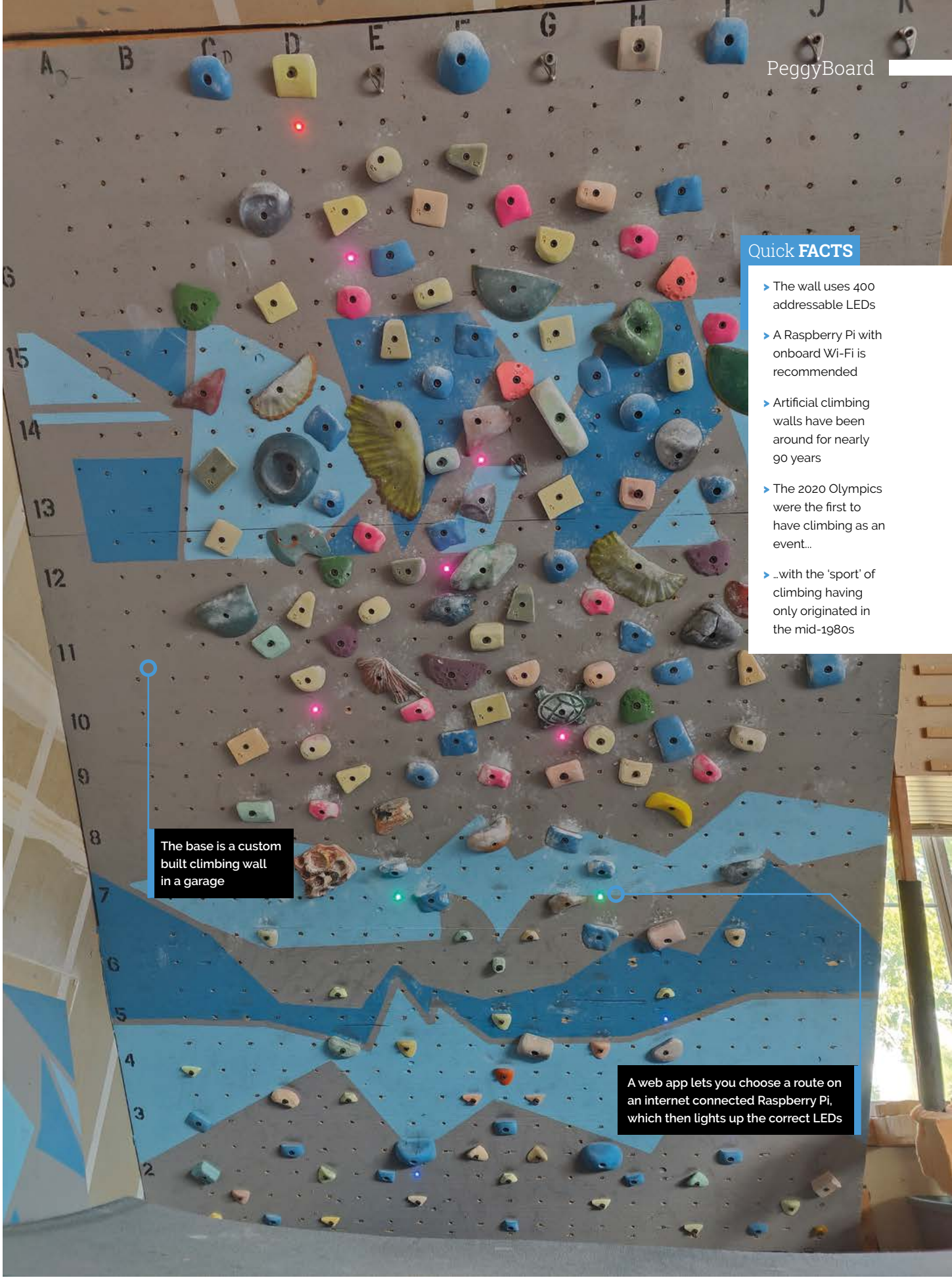


Warning!
Build & Climb Safely

If you plan to recreate this project, it should only be attempted by people well versed in building (and climbing safety)

magpi.cc/powertools

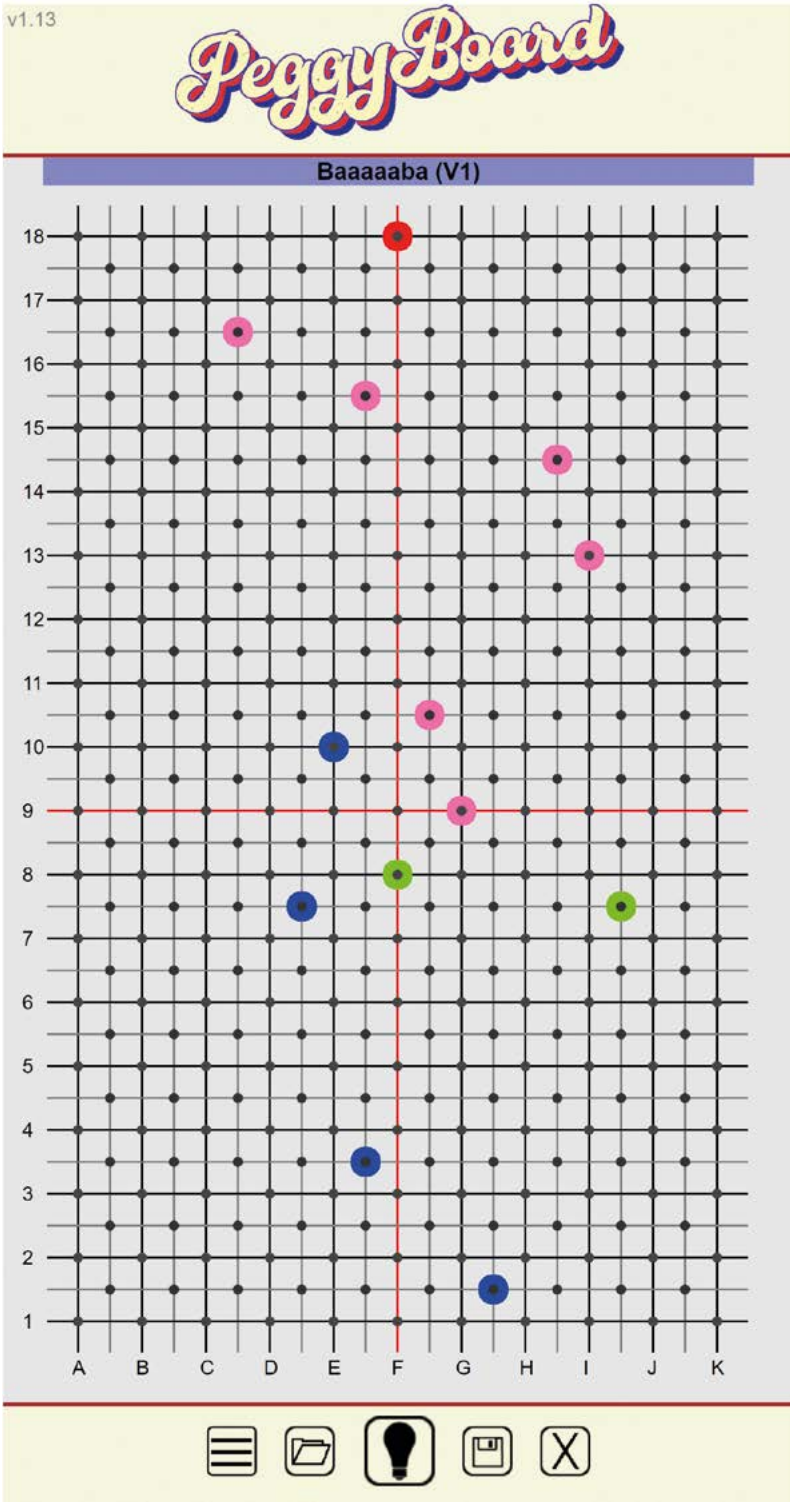
▶ Select your route on your phone to activate it on the wall. The web app remembers past routes and lets you create custom ones



Quick **FACTS**

- The wall uses 400 addressable LEDs
- A Raspberry Pi with onboard Wi-Fi is recommended
- Artificial climbing walls have been around for nearly 90 years
- The 2020 Olympics were the first to have climbing as an event...
- ...with the 'sport' of climbing having only originated in the mid-1980s

▼ The LED grid is also painted onto the wall so you know what holds you're selecting



LED walls

Building a high-tech climbing wall might seem daunting, but breaking it down into three parts makes it seem a little less intimidating.

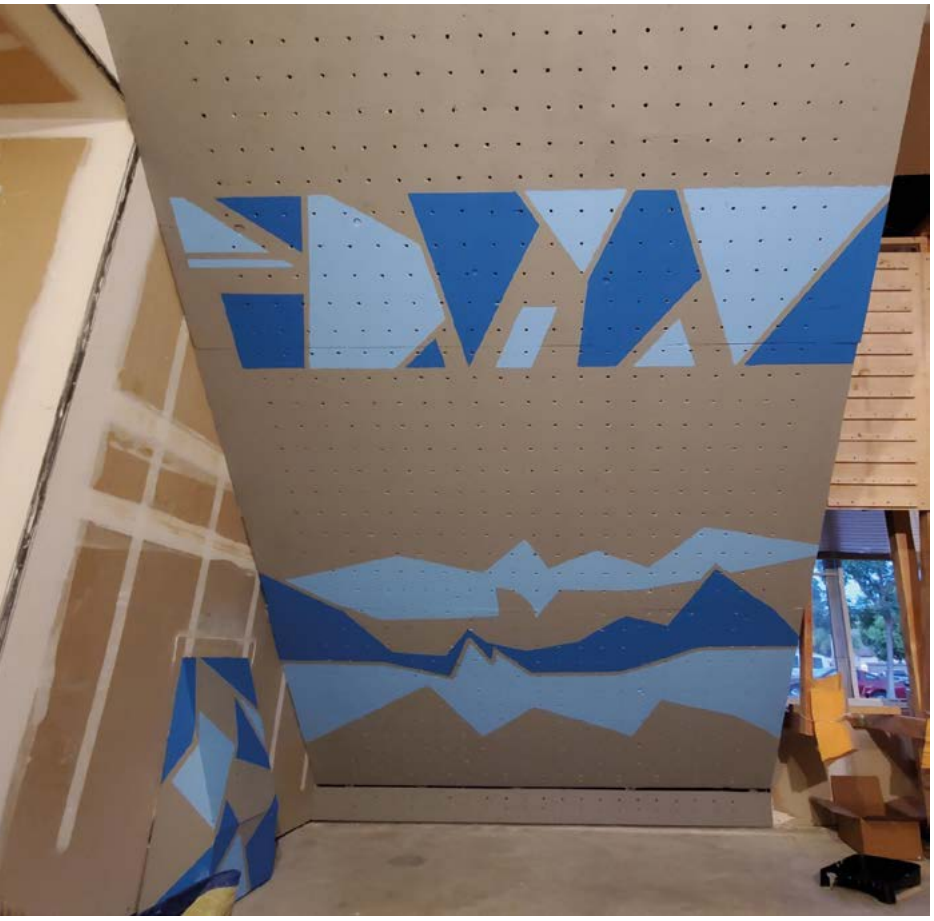
"The first being the building of the physical climbing wall which involved some framing and reinforcing the garage roof," explains Pegor. "The second was putting together the control panel which housed the Raspberry Pi, touchscreen, power supply, and level shifter. The last part was writing the code to make the whole thing interactive and functional. It's mostly written in JavaScript, PHP, shell scripts, HTML, CSS, and Python."

Reinforcing a roof still seems like a bit of a tricky process, but he makes it sound simple when it's broken down that way. It seems to have done the trick though.

"It works great," Pegor says. "The LEDs all light up properly, the web app is easy to use –outside of the occasional fat finger on the wrong hold – and the database is very useful to go back to saved climbs. The only issue we ran into was the occasional Wi-Fi connection drop, but that was due to router issues and the distance of the control panel."



▲ A new addition is a touchscreen control so you don't always need the webapp



▲ A climbing wall should look like one, including the art you'd commonly see in a climbing gym

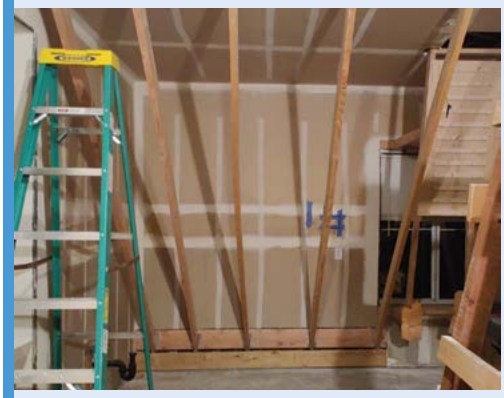
Wall of the future

It's a very interesting setup which would likely lead to good data collection about your climbs – great for people wanting to improve times, or see any mistakes they hadn't noticed. For now, though, this particular wall is largely complete.

"The touchscreen addition was the latest upgrade," Pegor tells us. "It didn't originally have one, and this made configuration a bit of a hassle on first setup or network changes. With that addition I also decided to clean up the code and make it open-source, and here we are. I don't have any future upgrade plans at this time. My friend has made a couple of suggestions for some new fun features, such as lighting up a hold momentarily so it's easier to see which hold you have set, but I haven't set time aside to implement them yet."

Hopefully the garage roof won't need further reinforcement for it.

Custom climb



01 Building a frame for a climbing wall is serious business. It needs to be sturdy enough to support itself and a person – in this case the ceiling and roof had to be reinforced.



02 Adding electronics means a lot of LEDs in this case – which you'll need to power and address specifically to illuminate the chosen route



03 A lot of coding went into PeggyBoard – routes, LED control, web services, and a web app, built using several kinds of technology

Mini Observatory

Finally having the time to focus on his hobbies led one maker to design and build a self-contained star-gazing setup, learns **Rosie Hattersley**



MAKER

Matt Hough

Matt Hough is an ardent Raspberry Pi aficionado and astronomy fan using his recent retirement to focus on ever more complex makes

shortbus.blog



Warning!
Stay indoors

This is a tabletop observatory for indoor use and the dome is not weatherproof. (Matt plans one that is). This version should not be used outside as the electrical components will be damaged

magpi.cc/electricalsafety

Matt Hough's retirement from the corporate IT world freed him up to return to hobbies that fascinated him many years ago. He studied computing in the 1980s, but had not tinkered much since, other than with software.

While astronomy had always been a big draw, Matt rarely had time to pursue it while working. When Raspberry Pi was launched in 2012, Matt was immediately drawn to it because "it was incredibly easy to make and experiment with". After familiarising himself with Python, Matt found himself buying several more Raspberry Pi computers Matt found himself appreciating their diminutive size too: his new hobby was portable! "I could pack a small kit and take it with me when travelling – so whenever I had downtime, I could tinker and try things". Several of the projects he created have been astronomy-themed, but creating a mechanised mini observatory and telescope based around Raspberry Pi 4 offered far more of a challenge – one that took two years of development and would see him create a dozen versions of his telescope before he felt it was ready to be shown off to the world. Once he did, Matt was "slightly stunned how much interest it generated". We think it's pretty impressive.

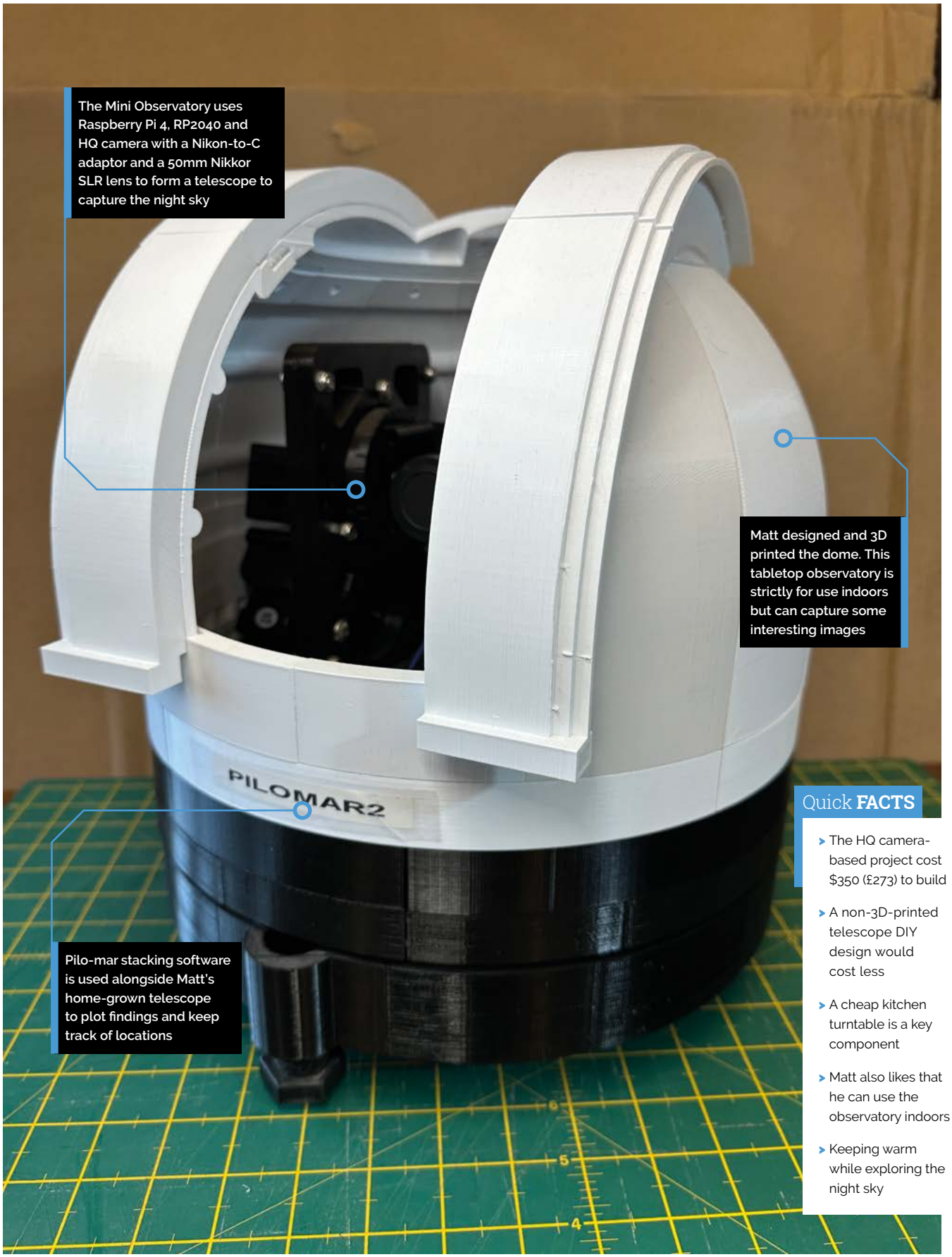
A familiar tale

Matt began with a series of traditional telescopes, but found them "a little frustrating to use in practice". It was hard to locate objects in the sky and most of what he could see was "often just a grey smudge". Digital telescopes were not that common and were expensive, but when Raspberry Pi launched the HQ camera sensor Matt wondered whether he could build a really simple digital telescope with the Raspberry Pi at the heart of it.

He had already built an Aurora clock that lets you know if there's a chance of seeing the Northern Lights, and a similar device using Raspberry Pi to track the International Space Station. With ten years of Raspberry Pi familiarity to draw upon, Matt was confident he could handle both high- and low-level functions and decided it would be the ideal basis for his own telescope and observatory design. "I knew there were Python packages available and I hoped to find pre-existing solutions to most of my needs". The initial plans were for a Raspberry Pi Pico project but Matt soon realised he needed more power, switching to "Raspberry Pi 4B with RP2040 help".

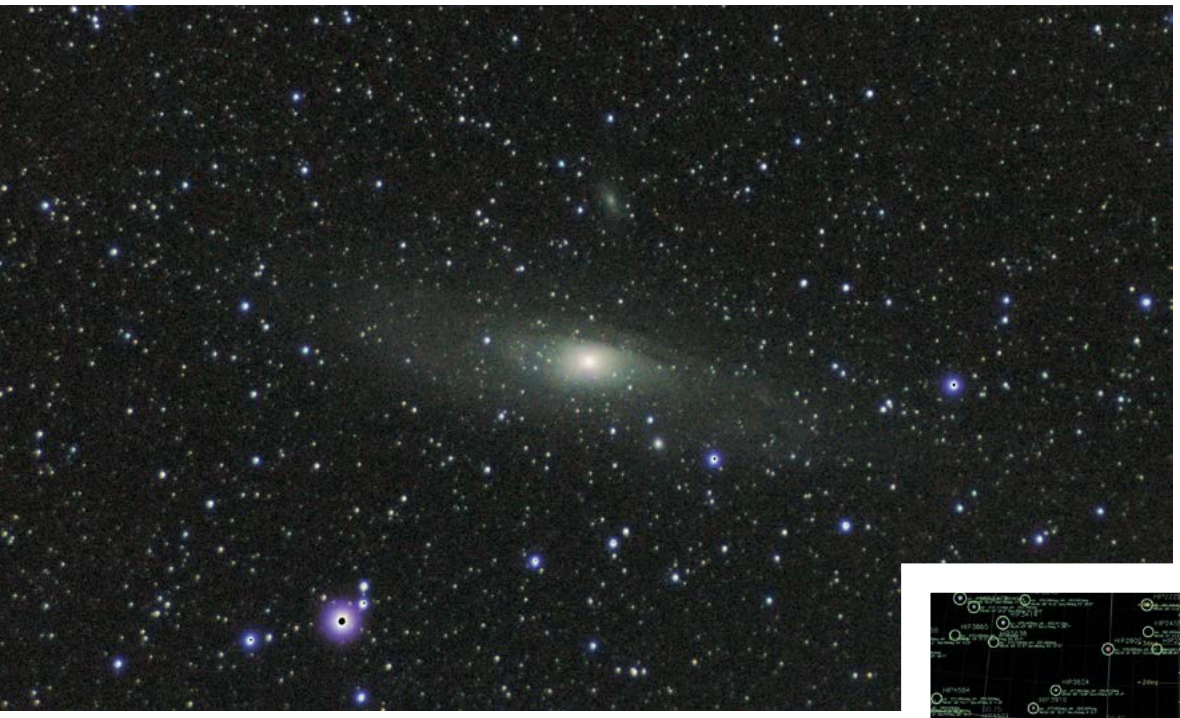


▲ Matt viewed designing a telescope as a great educational project



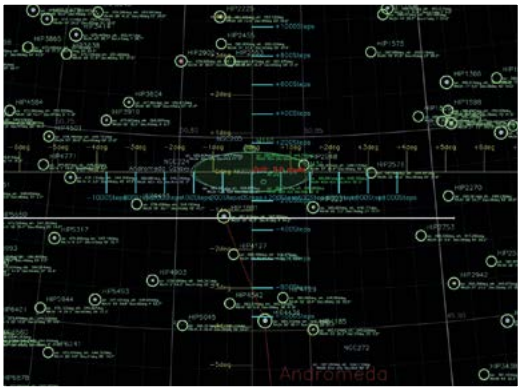
Quick **FACTS**

- The HQ camera-based project cost $350 (£273) to build
- A non-3D-printed telescope DIY design would cost less
- A cheap kitchen turntable is a key component
- Matt also likes that he can use the observatory indoors
- Keeping warm while exploring the night sky



A processed photo of the Andromeda galaxy using images gathered by the telescope using a 50mm lens

The software attempts to recognise objects and helps 'tune' the system



> He gave the motors a little RP2040 microprocessor brain

Matt was less confident of his 3D design skills: he needed to create more parts than he'd ever done previously and regarded this aspect as a distraction at the time, but says it's a new skill that has come in useful elsewhere since.

Seeing the lights

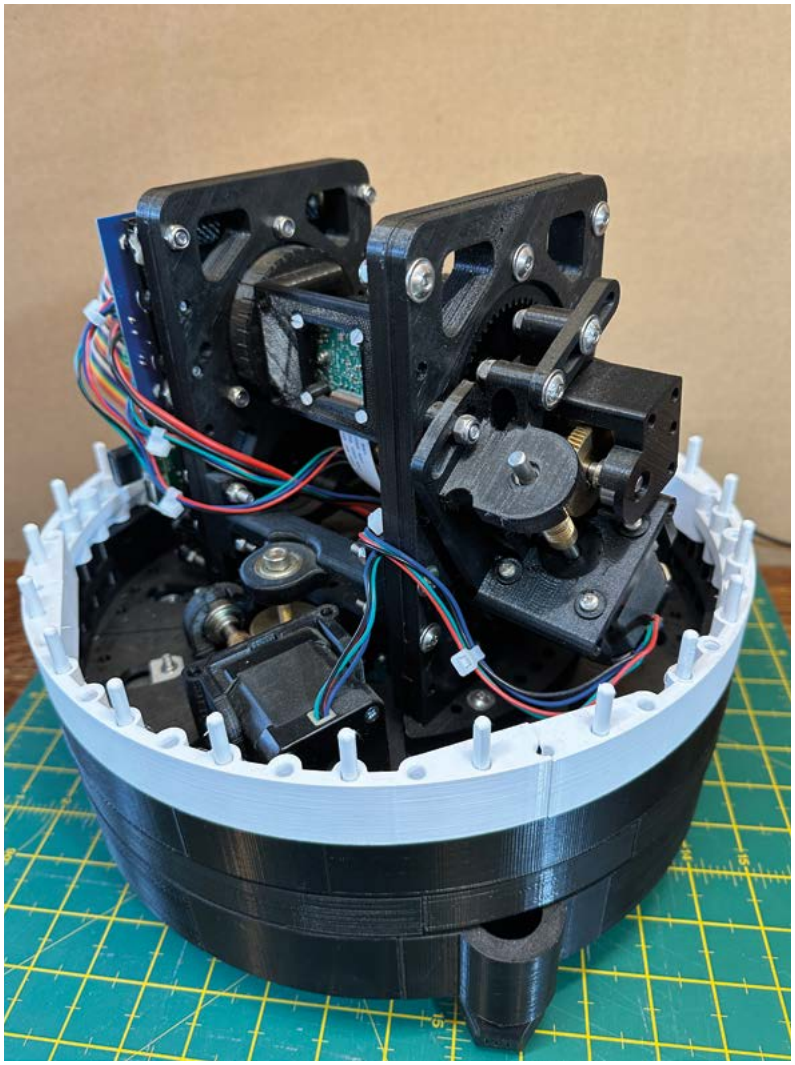
The Mini Observatory design emerged from successive experiments using Raspberry Pi Pico. Having started tinkering, Matt gradually developed routines to solve specific problems, learning how to control stepper motors and the technical aspects of the HQ camera. He needed precision here and went to a specialist company called Stepperonline (**omc-stepperonline.com**) to source them. Matt bought other parts from well-known Raspberry resellers such as The Pi Hut and Pimoroni, with more generic nuts and bolts from general hardware stores. He cautions over scrimping on potentially dangerous items such as power supplies: "always buy them from a trusted source".

The project provided a great learning experience, with the observatory, gears and mechanism all home-grown. The 'semi-intelligent' motor controller for the telescope is probably the most novel element. Matt needed a way for the telescope to move while Raspberry Pi was busy taking photographs, so gave the motors a little RP2040 microprocessor brain. "They were released at the perfect time."

Matt was able to make use of Python packages such as Skyfield, OpenCV, PiDNG and Astroalign and says it was a good choice for his Mini Observatory project. He is also really keen to process the photographs in real time onboard the telescope. "I haven't solved that yet, so I still need to do some offline processing afterwards. Realtime processing must be possible, I just have to research more."

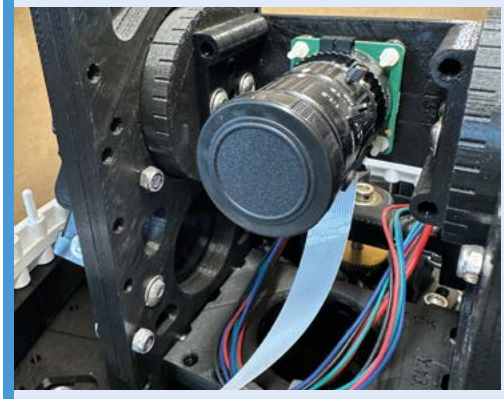
Since Matt first unveiled the project (**magpi.cc/miniobservatory**), several other makers have created versions, providing invaluable feedback and prompting him to tweak a few elements such as removing the infrared cutoff filters from his cameras which will make more objects visible.

Matt is also plotting a second version of his mini observatory, and is excited about the possibilities of Raspberry Pi with improved imagery and support for a second camera: one camera to do the tracking another to photograph the heavens. In fact "Raspberry Pi 5 may trigger quite a rewrite!"

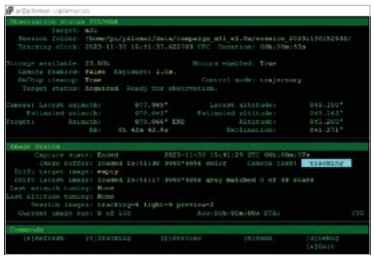


▼ The gearing allows the telescope to move vertically and horizontally in 0.004-degree steps.

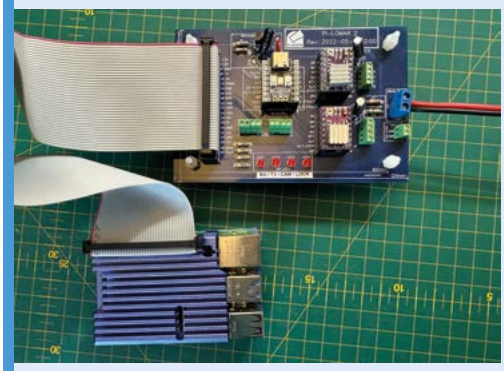
Be observant



01 This project proves you can use Raspberry Pi camera components to make a working telescope. Full build instructions and STL files are on Matt's GitHub (**magpi.cc/pilomar**), but there is plenty of scope to design your own dome.

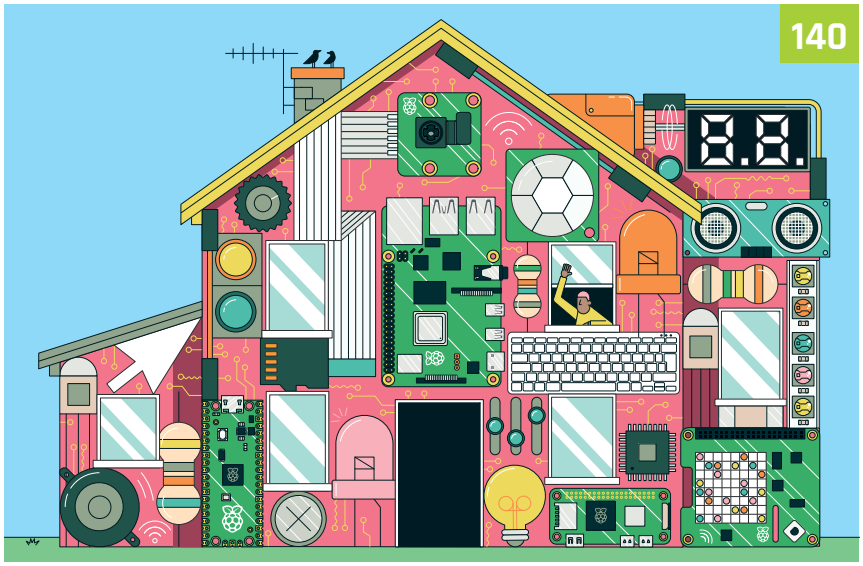


02 Matt recommends Raspberry Pi 4 and Buster OS with at least 2GB RAM and a 32GB SD card "because Pilo-mar creates a lot of data" while tracking the skies



03 Pimoroni Tiny2040 and TinyPython control the azimuth stepper motor, while Raspberry Pi 4B oversees the camera tower that forms the telescope

Maker Guides







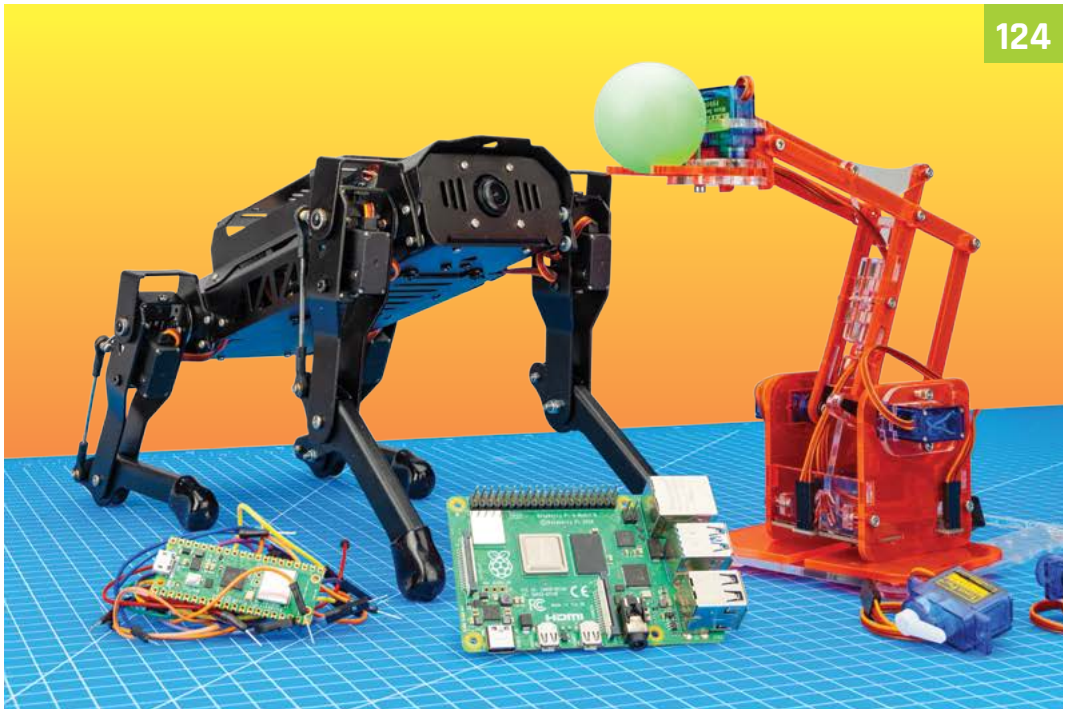
118 Astrophotography with Raspberry Pi
See the stars better than ever before

124 Python robots
Build dynamic robots with arms and legs

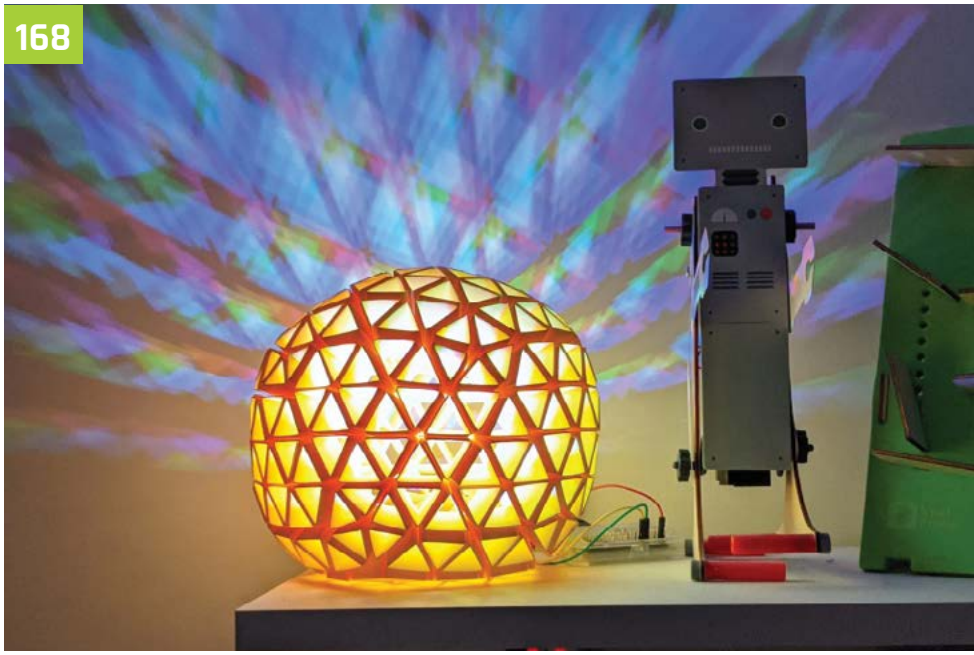
136 ML-powered audio transcriber
A must for any interviewer

140 DIY Home Automation
How to make your house smart

148 Emulate a BBC Micro on Raspberry Pi 400
Retro computing meets Raspberry Pi







152 Next Gen Retro Gaming on Raspberry Pi 5
Take it back to the 2000s

160 Build a universal media player
The last media centre you'll ever need

168 Arts & Crafts with Raspberry Pi
Get creative with a microcomputer

ASTROPHOTOGRAPHY with Raspberry Pi

Get stunning shots of the cosmos with your Raspberry Pi, a Camera Module, and some smart code. By stellar guide **Rob Zwetsloot**

Looking up at the night sky is something we've been doing since ancient times, with stars, planets, and moons shining down from the cosmos to remind us that there's a whole universe beyond Earth. Taking photos of the night sky to capture these sights can be a little tricky, though.

This is where Raspberry Pi comes in. With a range of excellent Camera Modules and customisable automation, you can pretty easily create a setup that will catch the wonders of the night and the beauty of far-off celestial bodies. It's time to heed the call of the night.

Basic sky photography

Get the best cosmic shots with these out-of-this-world tips

You can still get great shots without a telescope, using just a Raspberry Pi HQ Camera, if you choose the right lenses and program the shot in a specific way.

01 Choosing a lens

Without a telescope, you'll probably be aiming for night sky photos that include a lot of the sky and don't focus on celestial bodies. In this case, 16–24 mm wide-angle lenses are a good option, which means the 16 mm lens often sold with High Quality Cameras is a perfect place to start.



02 Exposure settings

A longer exposure setting allows for more light to be captured by the sensor. For astrophotography, you normally want **15 to 30 seconds** of exposure so enough light data can be captured. The camera commands are programmed in microseconds, so this would be a value of 15,000,000 and 30,000,000 respectively.

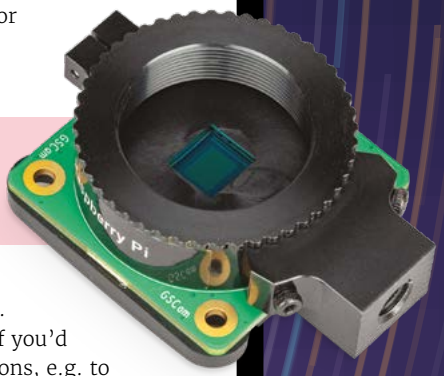


03 Command-line code

With the right lens and knowledge of exposure settings, we can start building a command that works in rpicame. For long exposures, we need to disable some automatic control algorithms for the best results. Overall, the command looks like this:

```
rpicam-still -o nightsky.
jpg --shutter 15000000
--gain 1 --awbgains 1,1
-immediate
```

This will take a 15-second exposure as a high-quality JPG. You can read the docs further if you'd like to make further modifications, e.g. to the gain level of the capture: **magpi.cc/rpicam**.



The HQ Camera has a maximum exposure time of about ten minutes

Tip

The HQ Camera uses a C/CS-mount and M12 mounts for lenses, so you can use whatever lens you like, if it's compatible

ASTRO NAVIGATION

68 Creating star trail photos



70 Shooting with a telescope



Create star trail photos

Use long exposure and time-lapse techniques to create beautiful night shots

With the basic setup, you're able to create a couple of kinds of classic space shots – wonderful photos of the night sky, or stylish star trails. With a Python script, you can create the streaking star effect created by the Earth's rotation.

Hardware

USB drive



Taking a lot of big pictures can take up space really fast

Screen (optional)



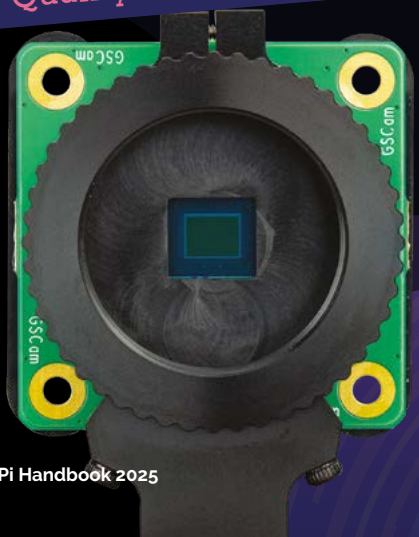
It's always a good idea to check what you're taking a photo of, although you can always use VNC if you don't have a spare screen

Raspberry Pi (any)



You can use any Raspberry Pi with a CSI port, which means most Raspberry Pi Zeros and any standard model. The newer the better, though

Raspberry Pi High Quality Camera and lens



See the previous page for tips on what lens to get – although the 16 mm one will be more than fine for this

Power bank



You'll want to do this outside, so a decent mobile battery is essential

Software

Some simple Python code that you can use to take a night full of shots. Put it on your USB drive

01 Setting up the software

For doing what is called a photo stack, we are going to be using the GNU Image Manipulation Program (IMP) from **gimp.org**. Once you've installed that, you'll also need a plug-in to create the stack. Grab this from **magpi.cc/starstack** and follow the instructions.

02 Get your photos

Grab the USB drive with your photos and put it into your PC. We suggest copying them to a new folder on your PC to work with – make sure

> "You might be able to tailor the perfect shot"

you know where it lives! Once that's done, open IMP, go to File > Create > Startrail. Set 'Light Frames' to the folder you put all the photos you want to use in, and click OK.

03 More options

There are more options to play with, which you can find on the plug-in website, that can help you refine your photo. Take a look at them if you're not satisfied with your result, and you might be able to tailor the perfect shot.

star_trails.py

> Language: **Python**

```
#!/usr/bin/python3
import time

from picamera2 import Picamera2

picam2 = Picamera2()
picam2.configure("still")
picam2.start()

# Give time for Aec and Awb to settle, before disabling them. Exposure time of 30 seconds
time.sleep(1)
picam2.set_controls({{"ExposureTime": 30000000, "AeEnable": False, "AwbEnable": False, "FrameRate": 1.0})
# And wait for those settings to take effect
time.sleep(1)

start_time = time.time()
for i in range(1, 100): # 100 shots
    r = picam2.capture_request()
    r.save("main", f"image{i}.jpg")
    r.release()
    time.sleep(300) # Five minute delay for timelapse

picam2.stop()
```

Tip

Depending on how big a USB drive you have attached, you can always increase the number of photos and exposure time (up to 60 seconds), and decrease the number of seconds between shots!

LOCATION, LOCATION, LOCATION

You'll obviously want to shoot at night, but where should you shoot from? The best place is somewhere away from a city or urban area where light pollution doesn't affect the view of the stars. The website **gostargazing.co.uk** has excellent recommendations for places to visit.



Shooting with a telescope

Get clear photos of distant objects using a telescope and smart software

To take amazing photos of celestial bodies, you need to be able to keep them in view, which means moving the camera. Luckily, with the right software, hardware, and a Raspberry Pi, you can take some amazing photos.



Hubble Pi

Student Santiago Rodriguez took a fairly simple approach of setting up a telescope for astrophotography with Raspberry Pi – he merely used an adapter for the eyepiece on a telescope so the HQ Camera could be plugged in.

With that done, he created a special UI called AstroCam that allows him to point the telescope at the right spot and then also control some of the camera settings, like the ones we've previously discussed. It's a lot cheaper as a result.

"Most good USB cameras for astrophotography start at about €200 and require a connected computer at all times. Hubble Pi can also work wirelessly, or as a standalone if needs be," he told us, when we interviewed him about it.

magpi.cc/hubblepi



PiFinder

A lot of modern telescopes can connect to computers to help with tracking distant objects. Richard (aka 'brickbots') had been doing a lot of manual checking with paper and then a commercial computer – he decided to build one based on Raspberry Pi.

It uses a mixture of GPS positional data and a visual feed from a HQ Camera to figure out exactly where the telescope is pointed, including a custom interface so it can be controlled out in the (literal) field.

"My hope is that other people will find this combination of functionality useful, will build their own PiFinder, and help the whole project improve by making suggestions and potentially contributing to the software. It's a pretty easy build with off-the-shelf parts and beginner-friendly soldering," Richard says on his website.

magpi.cc/pifinder

Astrophotography Autoguider

After spending many nights trying to get great shots from stacked photos – and too many not coming out very well – Joe Kutner decided to program his own autoguider that could keep his telescope pointing at the same celestial object.





> "Every step in the process had its challenges"

Making use of a touchscreen and Raspberry Pi, Joe created a ton of automated scripts that could be activated at the press of a button, and make sure his telescope was always pointing at the exact same object in the night sky.

"Every step in the process had its challenges," Joe told us a few years ago. "I would install one piece of software and then find out it wasn't compatible with some version of another piece of software I needed. When I finally got everything running, it wouldn't talk to my telescope until I installed yet another version of the software. There were dozens of these little paper-cuts but, in the end, it was worth working through them."

magpi.cc/autoguider

COSMIC EVENTS

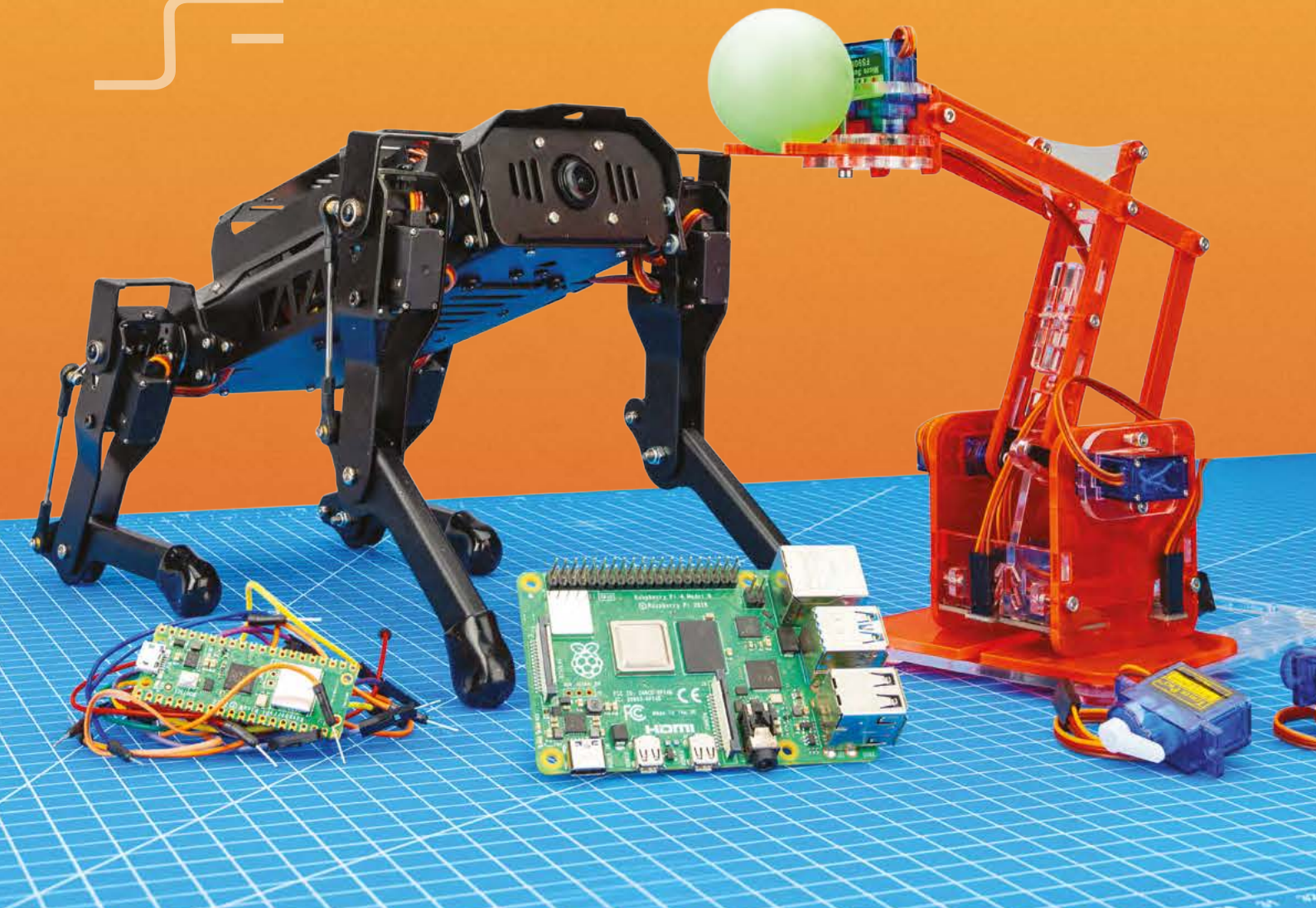
Want to know when the best times for seeing specific objects is? The calendar from Sea and Sky has dates for all the major events in the night sky. Take a look here: **magpi.cc/astrocal**



Python Robots

Craft, control, and program Python-powered robot arms and legs

By Danny Staple and Lucy Hattersley



Robots become infinitely cooler when arms and legs power them. This feature has them all, including walking dogs, humanoid replicants, and grasping robot arms.

These robots all share one thing in common: servo motors, also known as 'servos'. This type of motor rotates back and forth, unlike the DC motors of wheeled robots that spin endlessly around.

> This feature will equip you with a real-world, hands-on understanding of articulated robotics

Dive into an enriching journey of Python programming and robotics with this feature. We will explain how servo motors work, how you can control them in Raspberry Pi and look at some of the fantastic robotic kits on the market.

Whether you're a hobbyist or an aspiring robotics expert, this guide will equip you with a real-world, hands-on understanding of articulated robotics. So it's time to gear up, assemble your tools, and embark on a thrilling robotics adventure!



TYPES OF SERVO

Servo motors are a type of motor that rotates back and forth with precision. They are used in robots that require precise control, such as robot arms and limbed dogs and walking bots. The more expensive options include a sensor of some kind to measure position. Here are some types of servo to look out for...

SG (Servo Gear)

Servo gear units like the SG90 and newer SG92 (£4/$4) are inexpensive and light with precise movement control. The SG92R can rotate approximately 180 degrees (90 in each direction). You'll find them in entry-level robotics projects where cost and weight is a concern.

magpi.cc/sg92



MG (Metal Gear)

The next step up in servos is to switch out the plastic gears for metal. Servos such as this MG92 (£7/$7) offer more durability and strength than the entry-level SG models, and are better where the servo is going to lift weights or move loads. They are a bit heavier and noisier than their plastic counterparts.

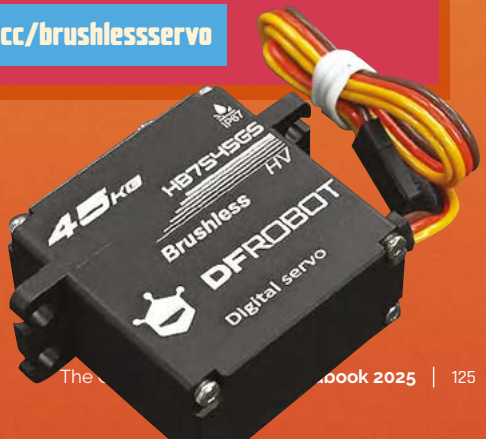
magpi.cc/mg90



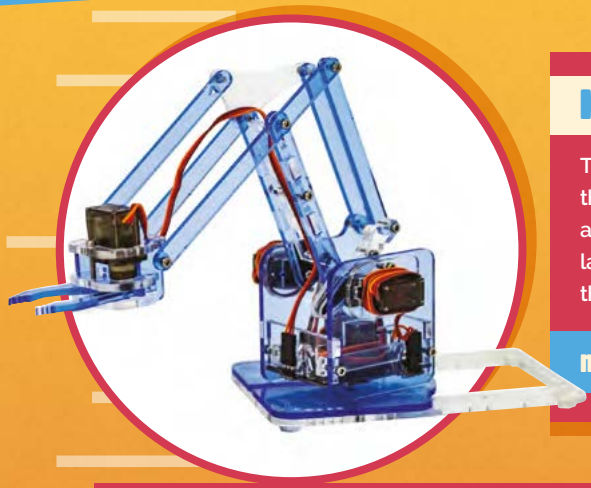
Brushless

In a typical motor, brushes are used to switch the polarity of the magnetic field, which is necessary for the motor to continue spinning. However, brushes are subject to wear and can create electrical noise, limiting the lifespan and performance of the motor. Brushless motors last longer, use less energy, create less heat and make less noise while being faster with further range. They cost a lot more, though.

magpi.cc/brushlessservo



ROBOT KITS



MeArm Maker Kit

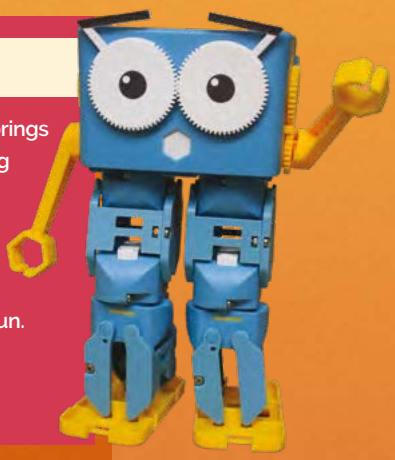
This is the kit that we are using to explain servo control for the main part of this feature. MeArm Maker (£39/$50) is a very low-cost robot arm that can be assembled with just a hex key "and a dash of enthusiasm". The kit includes laser-cut acrylic parts and four servo motors. It's basic, but that simplicity makes this a great place to start before moving on to more complex walking robots.

magpi.cc/mearmmaker

Marty v2

Marty the Robot V2 (£415/$449) is a charming two-legged robot that brings coding to life. Developed by Robotical, it features lots of accompanying educational projects and this friendly robot is controlled using Scratch or Python. Robotical claims that it is suitable for children aged seven and up to university level. Marty is controlled using coloured cards, an Android or iPhone app, or Raspberry Pi. Despite its cute exterior, Marty is packed with potential, making it a great tool for learning and fun.

magpi.cc/marty



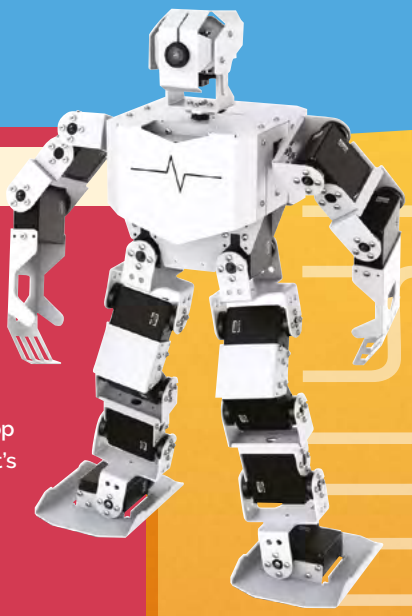


Petoi Bittle X

We tested Petoi Bittle (£204/$256) and found it to be a cute, palm-sized robot dog that is a great tool for promoting STEM learning. A new Bittle X model is in for testing and now offers voice interaction. It is a quad robot with four controllable legs and its price is a more affordable alternative to high-end robot pets. It comes with the NyBoard 1 controller board and ten servo motors. It's a delightful way to own a robot dog and there are a lot of coding projects to be found here.

magpi.cc/petoibittle

Tony Pi



Hiwonder has impressed us with its wide range of high-quality articulated robots. TonyPi (£453/$570) is a humanoid kit powered by Raspberry Pi 4B 4GB. It's not cheap, but the construction is superb and it features high-voltage bus servos. This advanced robot is designed to provide a hands-on experience in AI and robotics. TonyPi's capabilities include image recognition and tracking, making it a versatile tool for learning and exploration. Hiwonder provides an app interface, and you can dive into the kit via SSH and code it directly, so it's good for beginners and experts alike.

magpi.cc/tonypi

SpiderPi

The Hiwonder SpiderPi (£477/$600) is a substantial robot, standing nearly 30 cm off the ground and spanning nearly 60 cm when its legs are fully extended. Constructed with 18 servos similar to its Hiwonder sibling, the TonyPi, the SpiderPi is controlled by a 4GB Raspberry Pi 4. This hexapod robot features an HD camera that can rotate through 140 degrees, enabling it to detect faces and signal recognition by waving a leg.

SpiderPi performs various movements including scuttling forward, back, and sideways, as well as twisting and raising its front legs. Its camera and movement speed can be adjusted via a sliding bar in the app. The robot can also follow an undulating line on the floor, distinguish between different-coloured objects, and follow commands issued via a Python script tied to a QR code. It scared the bejesus out of our cat.



magpi.cc/spiderpi



Mini Pupper

MangDang's Mini Pupper (from £357/$449) is an open-source ROS robot dog kit designed to make robotics more accessible. MangDang claims Mini Pupper to be an excellent hands-on tool for STEM and K12 education. With this kit, learning robotics becomes a fun and engaging journey.

magpi.cc/minipupper



SUPER SIMPLE ROBOTICS

We've got a guide for newcomers to robotics in issue 120 of *The MagPi*. Unlike the articulated wonders featured in this tutorial, our earlier feature shows you how to create a wheeled robot using a few low-cost parts and control it with code. Wheeled robots are much simpler, but less fun.

magpi.cc/120

Control a robot arm with Python

Build and control a robot arm, while learning about how to remotely control servo motors on Raspberry Pi Pico W

By **Danny Staple**



MAKER

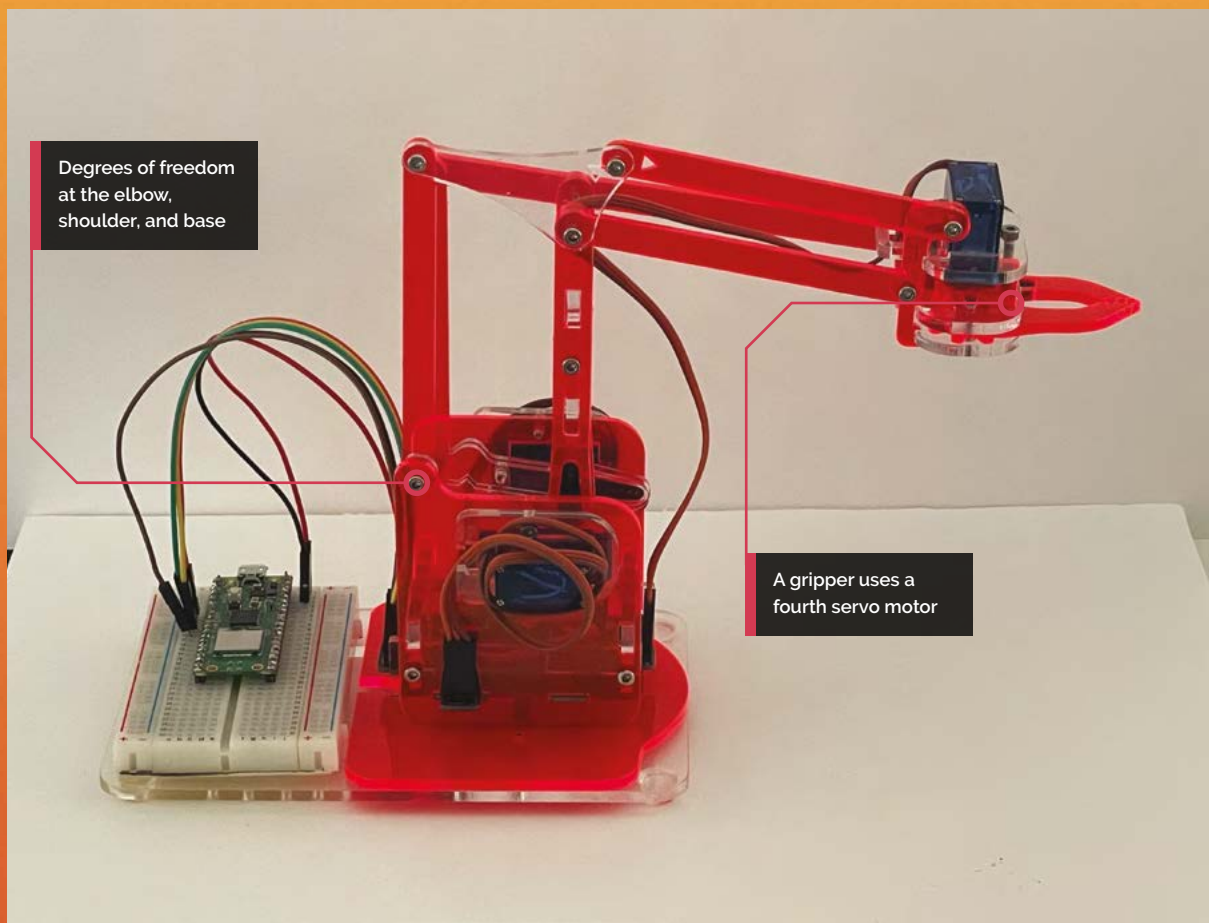
DANNY STAPLE

Has a home robot lab, making robots with Raspberry Pi and Pico as Orionrobots. Author of *Robotics at Home with Raspberry Pi Pico*.

orionrobots.co.uk

YOU'LL NEED

- MeArm Maker **magpi.cc/mearmmaker**
- Breadboard **magpi.cc/breadboardhalf**
- Jumper wires **magpi.cc/jumperwires**
- Laptop, tablet, or smartphone with Wi-Fi
- A small cross-head screwdriver
- Thonny



Servo motors allow your Raspberry Pi Pico to control articulated robots, moving their joints to specific places.

In this tutorial, you will learn where servo motors can be used and how to connect a servo-driven robot arm to a Raspberry Pi Pico W.

You will learn how to use the asyncio library (**magpi.cc/asyncio**) to control servo motors together, making articulated robotics possible.

You will then control this robot arm from a web page or smartphone.

01 What are servo motors?

Articulated robots use servo motors to control their joints. Servo motors combine a motor, a gearbox, and control circuits. They are usually limited to 180 degrees of motion.

A servo motor uses closed circuit control and is controlled with pulse-width modulation or 'PWM', where the length of pulses carries position information.

A Raspberry Pi Pico can output these signals, with 65,536 possible values. Servo motors only respond to a range of 8000 values, with very short pulses 50 times per second. The servo finds the difference between its input and output position, then moves the motor to reduce the difference.

02 About the MeArm

The MeArm is an inexpensive, laser-cut robot arm kit from Mime Industries (**mime.co.uk**). MeArm is designed for hobbyists and learners to build and program. Mime Industries has MeArm models with electronics designed for Raspberry Pi, Arduino, and micro:bit. However, the MeArm Maker is a model which allows connection to any controller, and will work well with Raspberry Pi Pico.

Alternative robots

The techniques in this tutorial can be used for other articulated robots like Petoi Bittle X, the Hiwonder TonyPi, or for you to build a hexapod or robot tentacle!!

Degrees of freedom

When talking about segmented robots, like this arm, we use the term 'degrees of freedom', or DOF, to count the number of places or axes segments can move.

The MeArm uses four servo motors which allow it to move in three axes and pick things up. The three degrees of freedom are a rotating base, a shoulder joint, and an elbow joint. The axes are controlled by servo motors.

03 Preparing Pico W

Before we can interact with the arm, we need to prepare Raspberry Pi Pico W. First you will need to solder headers onto it, unless you bought a model with pin headers on board. See **magpi.cc/soldering** for help with soldering.

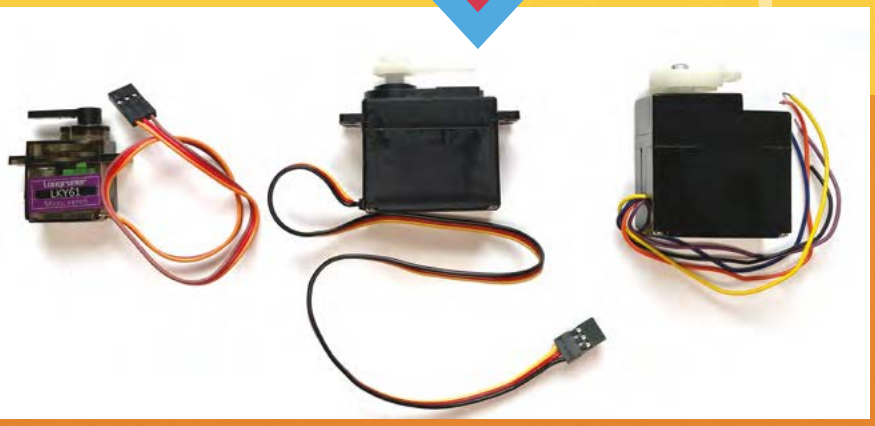
> "The MeArm is an inexpensive, laser-cut robot arm kit"

You will also need the current version of MicroPython for Raspberry Pi Pico W. Download the latest UF2 image from **magpi.cc/micropython**. Hold the BOOTSEL button while connecting Pico W to a computer via USB; you can then drag and drop the UF2 image onto Pico W on the desktop.

04 Wiring servos to Pico W

The servo motors will need to be configured before assembling the arm. Unplug Raspberry Pi Pico W from the USB port. Each servo motor connects to one of the 3-pin connectors on the MeArm board, with the orange or yellow wire going to the pin marked O or Y on the board.

The 6-pin connector is wired to the Raspberry Pi Pico W. Plug the Pico W into a breadboard, and then run a jumper from each labelled connection on the 6-pin connector to the corresponding breadboard row for the label on the Pico W.



A selection of servo motors. They come in different sizes to handle larger tasks. The MeArm robot uses tiny ones

05 Controlling servo motors with MicroPython

Plug in the Pico W, then open Thonny. Connect Thonny to the Pico W, and start the MicroPython Shell.

Type the following lines:

```
import machine
grip = machine.PWM(machine.Pin(4, machine.Pin.OUT))
grip.freq = 50
grip.duty_u16(5000)
```

5000 corresponds to the middle position for the motor. You can try other values and should see the motor spin. However, ensure it is at position 5000 before finishing this step.

Repeat this for the other three servo motors on pins 5, 6, and 7. Avoid turning servo motors by hand, as this is likely to break gears inside them.

06 Assembling the MeArm Maker

With the Pico still powered, follow the MeArm instructions linked below for attaching servo horns while the servo motors are in the middle position. You can play with servo motor positions here, but ensure you return them to position 5000 before assembling.

Next, unplug the USB connection to the Pico, then unplug the servo motors from the MeArm board. Leave the connections between the MeArm board and Pico W. Building the arm is best explained by the MeArm documentation, which you can find at **magpi.cc/buildingmearm**.

07 Controlling arm movements

The code to control PWM for servo motors is in a Servo class in **mearm.py**. It's helpful to see the servo motion as angles. We'll make the middle position 0, so the range is +/- 90.

The middle duty cycle value is 5000 for Pico MicroPython – we'll call this `PWM_MID`. If we take +/- 4000 as the range from this position, we can divide 4000 by 90 to get a `DEGREES_TO_PWM` scale factor.

We can then multiply an angle by this scale factor and add the result to the `PWM_MID` to get the duty cycle.

08 Making an arm from Servos

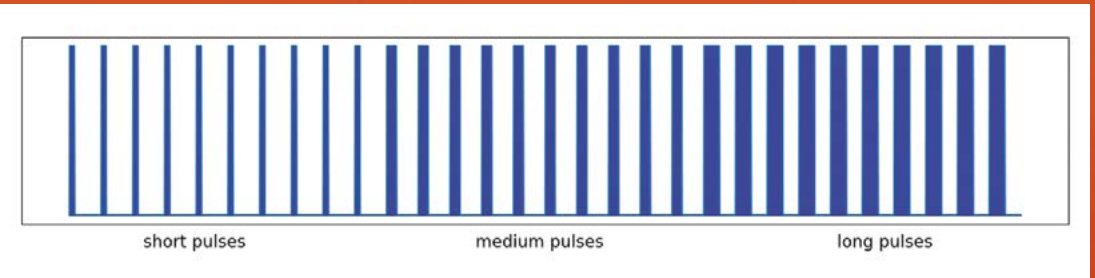
The Arm class uses the Servo class for each joint. The Arm class has default pin numbers corresponding to the wiring, but with different pin numbers, it allows two or more arms!

Load **mearm.py** in Thonny and hit the play button. You'll return to the MicroPython Shell.

Type `arm.base.set_angle(-30)`, to move the arm base to -30 degrees. You can try this with every joint. Find the limits by increasing numbers out from the 0 position in 5-degree steps until the joint doesn't move again, then go back a step. Don't leave a joint past these limits in order to avoid damage.

09 Smoother motion

The servos are jumping, but we can make them smooth. In `Servo.move`, we control speed



This chart displays PWM (pulse-width modulation) signals. Shaded parts show a high signal. Shorter pulses pull the servo anticlockwise, mid pulses go to the middle, and long pulses go clockwise

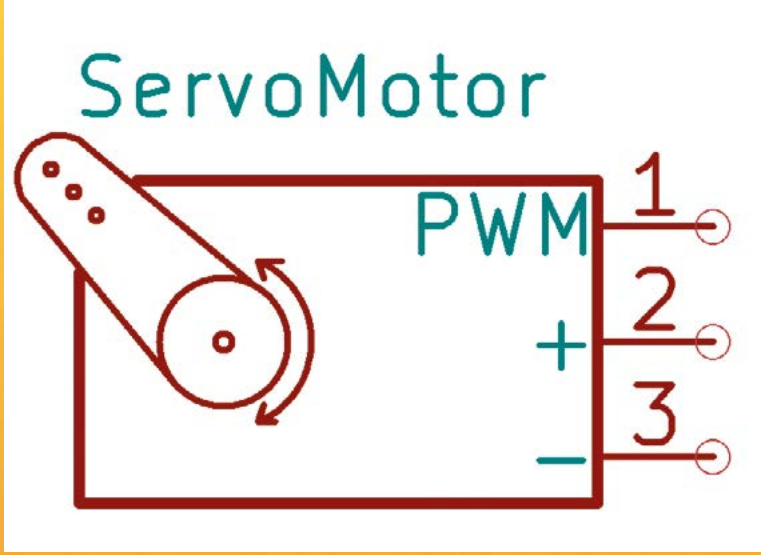
and smoothness with a time in seconds and with a step count.

The Servo class stores a `current` position; `Servo.move` sets a target and gets the difference between this and the `current` position.

`Servo.move` divides the time by steps for `step_time` and divides the movement difference by steps for `step_size`.

It loops over the steps, adding the `step_size` to the `current` position, moving the servo to this new position, and sleeping `step_time`.

In the Shell, type `do(arm.elbow.move(30))` to see the elbow move smoothly to 30 degrees.



A servo usually has three pins: '+' for voltage, '-' for ground, and a yellow or orange PWM input signal

10 Moving joints together

Moving joints together is more interesting. We used a `do` helper around `Servo.move` because `move` is asynchronous. When it encounters the `await` instruction, other code can run, enabling us to do multiple tasks at the same time.

The `move_together` method in `Arm` sets up a movement task for each specified joint position, and then uses `await uasyncio.sleep(seconds)` to wait for all the move tasks to complete. The tasks all get time to run, so things can move together.

> "It's helpful to see the servo motion as angles"

Try `do(arm.move_together(base=40, shoulder=20, grip=30, base=50))` to see everything move at the same time! Perhaps we could do something else?

11 Connecting to the network

We can connect our Pico W to a Wi-Fi network. Add a **secrets.py** file on the Pico with your SSID and Wi-Fi key.

```
SSID = "<network name>"
PSK = "<network password>"
```

The top of the file **web_arm.py** imports these secrets and uses them to connect to your Wi-Fi. We will also be using the microdot web server for MicroPython. You can install this from Thonny with Tools > Manage packages. Search for microdot, then click Install.

When you run **web_arm.py**, it will look for a connection. Once it has connected, it will print the network address of your Pico W to the console.

12 Building arm control endpoints

The microdot web server lets us build tiny endpoints to control our arm. There is a microdot app object, which lets us set a link between an incoming request path and a Python function, for example linking `/set_grip/<position>` to the function `set_grip`. The `<position>` part specifies a position parameter which the function can use and pass to the arm.

These functions are also in async, so the server can handle requests while the arm is still moving. Upload **secrets.py**, **mearm.py**, and **web_arm.py** to Pico W, then click the run button on **web_arm.py** to start the server.

13 Trying the endpoints

Point a web browser at the address you were shown in Step 11 plus a movement to make. For example: **http://192.168.0.136/set_elbow/30** to move the elbow to 30 degrees.

You do not have to wait for a previous movement to finish before sending more movement instructions to the robot. It will not behave correctly if you try to move the same motor twice before it's finished, though. This is OK, but it's not the most enjoyable way to interact with the robot.



The techniques in this article can be used to control the ten servo motors on the Petoi Bittle X robot dog

There's another endpoint in **web_arm.py**, named `/` and connected to an `index` function which serves a page to control the arm.

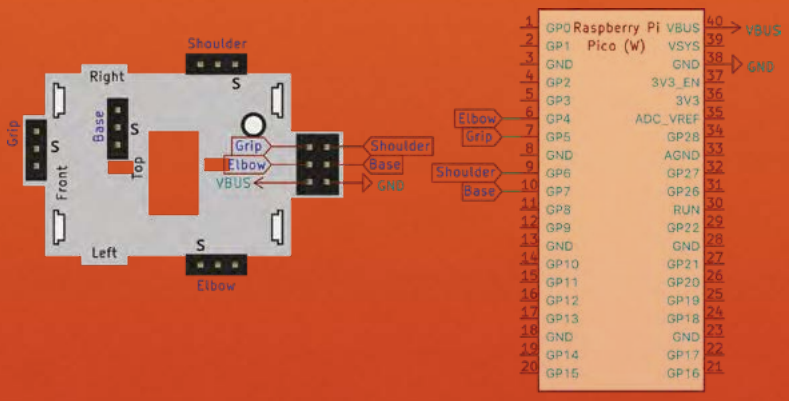
14 Putting sliders on a web page

The index function loads a **control.html** file and sends this back. The body of **control.html** has a title, and four `input type="range"` elements. Each range input lets you specify the minimum, maximum and starting values.

The datalist element `snap_to_zero` is used for the list property of the ranges to draw markers at 0.

The page has a style section, `overflow: hidden`, that stops the page being accidentally scrolled on a phone, The width and height settings make the sliders fit well on any screen, along with the viewport setting in the page head.

If you go to the address of your Pico W, you should see the four sliders.



Wiring the MeArm to Raspberry Pi Pico W. Wires go between labels with the same name

15 Connecting the sliders

In **control.html**, it loads `jquery`, a JavaScript library designed to help with handling page events and sending data from a page.

When the document is ready, our JavaScript searches for the input elements, and links on change handlers for each of them.

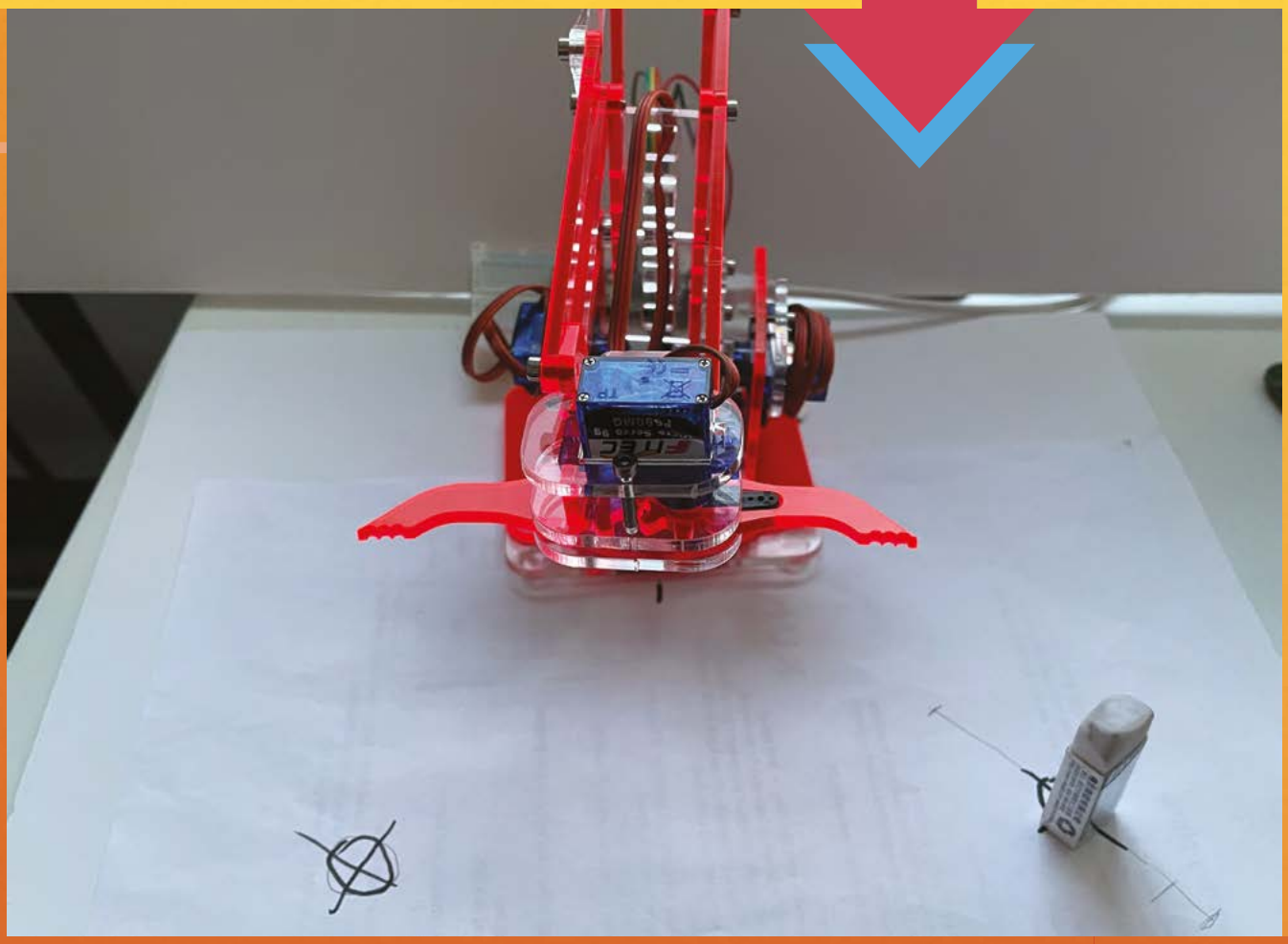
Focusing on the shoulder element, when this input is changed, the function will send a GET request to the **web_arm.py** server. This goes to **/set_base/** and combines it with the value of the shoulder element. The change event happens when an input on the page is moved, and the finger or mouse button is lifted. This means you can drag sliders around to move the arm positions. You can connect with a smartphone to drive this too!

> "You can drag sliders around to move the arm positions"

16 Recording steps

At the bottom of the page, there are buttons to add movement steps. In the HTML, they have onclick handlers to call into **web_arm.py**.

When the 'add step' button is clicked, **web_arm.py** collects the current position of each servo motor, and stores this set of positions in the variable `step_list`. Then, when 'run steps' is clicked, an async task calls `move_together` with each set of positions, playing the whole list back.



Setting out markers for the arm to pick up and move objects to

There are also save/load buttons. These will store the steps in a JSON file on Pico W, so you can load steps back later.

17 Fun with recorded steps

Put the robot on the long side of a sheet of paper. Draw a line in front of the base to align the arm's base. Draw a mark between 30 and 60 degrees to the left and 100 to 140mm from the base of the arm, and another mark to the right.

Place a light object, like a small eraser, on one mark. Using the web interface, open the grippers, rotate the arm to face the object, then move the arm so the object is between the open grippers. Press the 'add step' button. Close the grippers, and press 'add step'.

Now set the elbow, base, and shoulder joints to 0 and press 'add step'. Then move the base, elbow, and shoulders to put the object on the other mark and add a step. Open the grippers, then add a step. Now move the elbow and shoulders up, and add a step. Finally, set the elbow, base, and shoulder joints to 0 and press 'add step'.

If you put the object back on the first marker, and press the 'run steps' button, the robot will now automatically pick up the object and move it to the other marker. You can use the 'save steps' and 'load steps' buttons to save this for demonstrations!

18 What to do next?

Connecting the gripper servo motor power through a current sensor, like an INA219, means you could detect when the grippers are holding something by closing them until current goes above a threshold.

Articulated robots use inverse kinematics to calculate servo angles based on a desired position. Adding this would make it easier to position the arm from the controls.

You could add a distance sensor to the front of the arm, move the base to scan for the object, and combine this with inverse kinematics to position the grips for an arbitrarily placed object.

mearm.py

DOWNLOAD THE FULL CODE:
magpi.cc/mearmpy

> Language: **MicroPython**

```
"""
import sys
del sys.modules["mearm"]
from mearm import arm, do
# tests

do(arm.move_together(base=30, shoulder=40, elbow=30, grip=20))
do(arm.move_together(shoulder=0, elbow=-30))
"""
import machine
import uasyncio


PWM_MID = 5000
PWM_FREQ = 50
DEGREES_TO_PWM = 4000 / 90


class Servo:
    def __init__(self, pin):
        self.pwm = machine.PWM(machine.Pin(pin, machine.Pin.OUT))
        self.pwm.freq(PWM_FREQ)
        self.current = 0

    def set_angle(self, angle):
        self.pwm.duty_u16(int(PWM_MID + ( angle * DEGREES_TO_PWM)))

    async def move(self, position, seconds=1, steps=100):
        step_time = seconds/steps
        step_size = ( position - self.current) / steps
        for n in range(steps):
            await uasyncio.sleep(step_time)
            self.set_angle(self.current)
            self.current += step_size


class Arm:
    def __init__(self, elbow_pin=4, grip_pin=5, shoulder_pin=6, base_pin=7):
        self.grip = Servo(grip_pin)
        self.elbow = Servo(elbow_pin)
        self.shoulder = Servo(shoulder_pin)
        self.base = Servo(base_pin)

    async def move_together(self, base=None, shoulder=None, elbow=None, grip=None, seconds=1, steps=100):
        if base is not None:
            uasyncio.create_task(self.base.move( base, seconds=seconds, steps=steps))
        if shoulder is not None:
            uasyncio.create_task(self.shoulder. move(shoulder, seconds=seconds, steps=steps))
        if elbow is not None:
            uasyncio.create_task(self.elbow. move(elbow, seconds=seconds, steps=steps))
        if grip is not None:
            uasyncio.create_task(self.grip. move(grip, seconds=seconds, steps=steps))
        await uasyncio.sleep(seconds)


arm = Arm()
do = uasyncio.run
```

web_arm.py

Language: **MicroPython**

```
import machine
import uasyncio
import json

from microdot_asyncio import Microdot
from mearm import arm

import network
from utime import sleep

from secrets import *

def wifi_connect():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(SSID, PSK)
    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)
    print(wlan.ifconfig())

app = Microdot()
step_list = []

@app.route('/')
async def index(request):
    with open("control.html") as content:
        return content.read(), 200, {'Content-Type': 'text/html'}

@app.route('/set_grip/<position>')
async def set_grip(request, position):
    uasyncio.create_task(arm.grip.move(int(position)))
    return 'Moving'

@app.route('/set_base/<position>')
async def set_base(request, position):
    uasyncio.create_task(arm.base.move(int(position)))
    return 'Moving'

@app.route('/set_shoulder/<position>')
async def set_shoulder(request, position):
    uasyncio.create_task(arm.shoulder.move(int(position)))
    return 'Moving'

@app.route('/set_elbow/<position>')
async def set_elbow(request, position):
    uasyncio.create_task(arm.elbow.move(int(position)))
    return 'Moving'

@app.route('/add_step')
async def add_step(request):
    state = (arm.base.current, arm.shoulder.current, arm.elbow.current, arm.grip.current)
    step_list.append(state)
    return 'Added step'

async def run_steps():
    try:
        for step in step_list:
            await arm.move_together(*step, seconds=1)
    except:
        print("Steps were:", step_list)
        raise

@app.route('/run_steps')
async def handle_run_steps(request):
    uasyncio.create_task(run_steps())
    return 'Running steps'

@app.route('/save_steps')
async def handle_save_steps(request):
    with open("steps.json", "w") as f:
        json.dump(step_list, f)
    return 'Saved steps'

@app.route('/clear_steps')
async def handle_clear_steps(request):
    step_list.clear()
    return 'Cleared steps'

@app.route('/load_steps')
async def handle_load_steps(request):
    global step_list
    with open("steps.json") as f:
        step_list = json.load(f)
    return 'Loaded steps'

try:
    wifi_connect()
    print("Starting app")
    app.run(port=80)
except KeyboardInterrupt:
    machine.reset()
```

Machine learning-powered private speech transcriber

Keep your personal data private by carrying out automated voice transcription on your own Raspberry Pi



MAKER

K.G. Orphanides

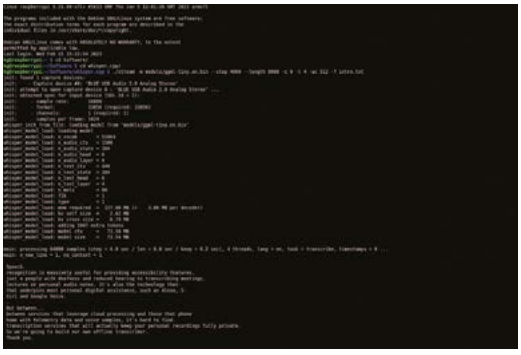
KG is a writer, developer and software preservationist. They're sceptical of the AI-in-everything fad, but also don't want a human transcriber to hear their audio diary.

magpi.cc/hauntedowlbear

Speech recognition is massively useful, from providing accessibility features to people with deafness and reduced hearing to transcribing meetings, lectures, or personal audio notes. It's also the technology that underpins most personal digital assistants.

But between services that leverage cloud processing and those that phone home with telemetry data and voice samples for later analysis by humans, it's hard to find transcription services that keep your personal recordings fully private. So we're going to build our own offline transcriber.

Whisper.cpp isn't perfect. It can't distinguish between different speakers and, like most transcribers, it can get confused by cross-talk and indistinct enunciation. However, its performance rivals that of many commercial services, and your data never leaves your hands.



▲ Although less accurate than 'main', if want to make a hands-free note or have the text of a meeting immediately available, you'll find the near-real-time 'steam' transcription tool useful

You'll Need

- Georgi Gerganov's Whisper.cpp port **magpi.cc/whispercpp**
- A microphone (for live transcription)

01 Prepare your system

Machine learning applications will cheerfully use as much processor power as you can throw at them – during cross-platform testing, we maxed out all 16 threads of a hefty AMD Zen CPU. Although Georgi Gerganov's Whisper port is optimised for efficiency, you can expect better results from more powerful systems. We used a Raspberry Pi 400 with 4GB RAM. Unless you just want to transcribe audio files, you'll also need a microphone – we used a Blue Yeti USB mic. For somewhat better performance, you could use an 8GB Raspberry Pi 4 running Raspberry Pi OS Lite to save on the overheads of the GUI.

02 Use the source

There are no pre-compiled binaries for ARM platforms, so you'll either have to clone Whisper.cpp from its Git repository, or download a release version of the source code from **magpi.cc/whisperreleases**. We've gone with

the former, but when in doubt, grabbing source code from the Releases page is always a safe bet. The current version of the master repository will always have the latest features, but there's a chance that you'll get the latest bugs, too.

```
sudo apt install libsdl2-dev
git clone https://github.com/ggerganov/whisper.cpp
cd whisper.cpp
```

Unless specifically stated otherwise, all commands in this tutorial are run from this directory (**whisper.cpp**).

03 The model

We're going to use the recommended tiny Whisper model for Raspberry Pi, which requires just 75MB of disk space and around 125MB RAM. While, on paper, the larger and theoretically more accurate base model might be assumed to work on Raspberry Pi, in practice the processing overhead resulted in breaks and delays in transcription, so we're going to keep it small.

```
./models/download-ggml-model.sh tiny.en
make -j tiny.en
make -j stream
```

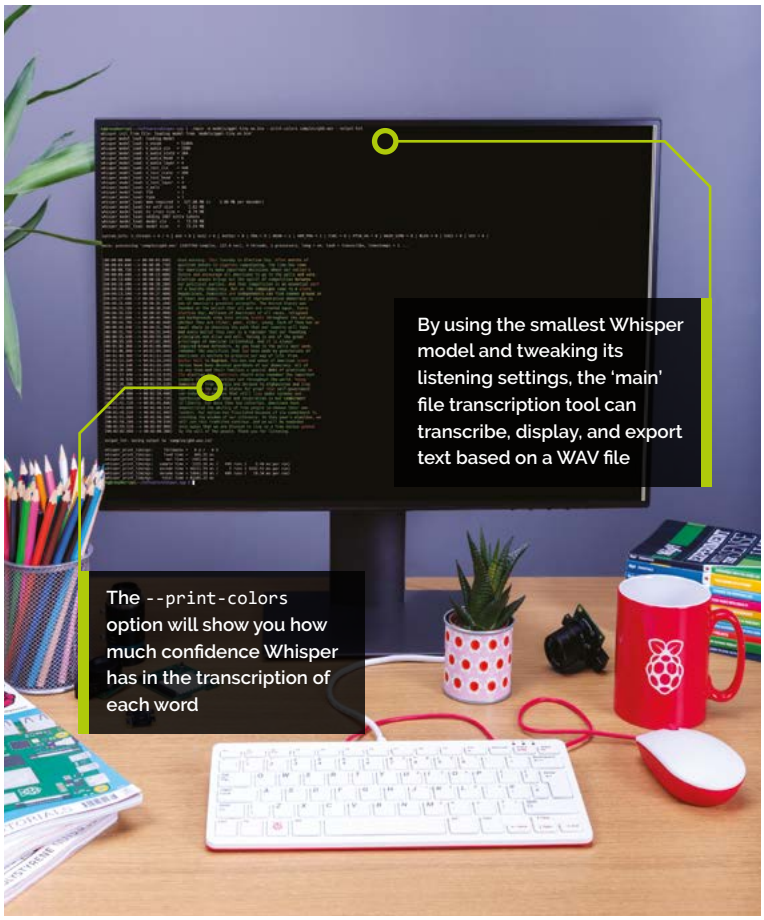
> "There are no pre-compiled binaries for ARM platforms"

When the build info is shown after making stream, you'll notice that the makefile automatically optimises for armv71, including options such as neon-fp-armv8.

04 Take dictation

The stream tool is your live transcriber. Everything you say into the mic will be passed through Whisper and displayed on screen. At the terminal, type:

```
./stream -m models/ggml-tiny.en.bin --step 4000 --length 8000 -c 0 -t 4 -ac 512
```



This command optimises for live transcription performance on Raspberry Pi's modest hardware by sampling the audio every four seconds with the `--step 4000` instruction. One second would be `--step 1000`.

05 Transcribe to file

Displaying text on-screen after a short delay is undeniably useful, particularly if you have difficulty hearing or following spoken lectures or conversations. However, you'll usually want to keep the transcribed text for future use. To do this, append `-f filename.txt` to the end of the command.

Press **CTRL+C** to end transcription.

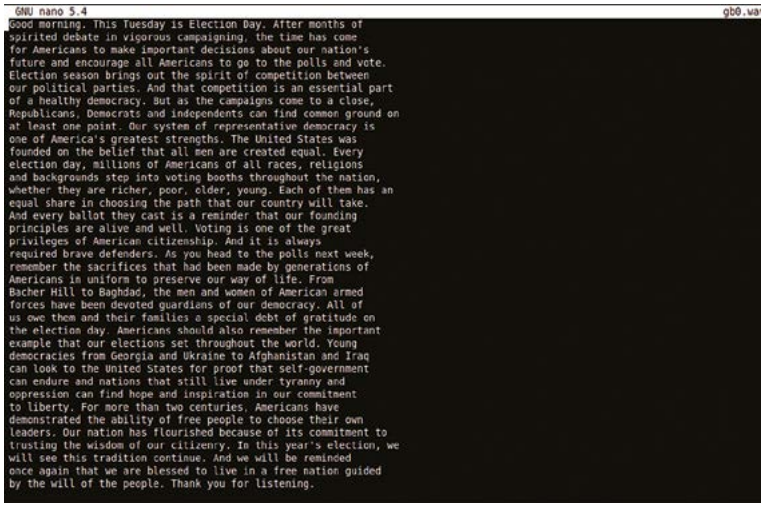
06 Ride the wave

You can also use Whisper.cpp to transcribe previously recorded files, making it a useful alternative to online AI transcription tools such as Temi and Otter.ai, with the benefit of knowing that your recording never leaves your hands.

Top Tip

Use the best mic you can

Microphone quality makes a huge difference to the accuracy of Whisper.cpp's live audio transcription.



▲ You can have the transcriber output an audio file to text, making it easy to do further work on

Top Tip

Multilingualism?

Although OpenAI's Whisper was trained on multilingual speech data, the tiny sets we're working with in this tutorial only recognised English in our tests.

Currently, if you want to transcribe an audio file, it has to be in 16-bit PCM WAV format. If your recorder outputs MP3 files, or even PCM files at a different bit depth, you'll need to convert them first. We can use ffmpeg for that at the command line.

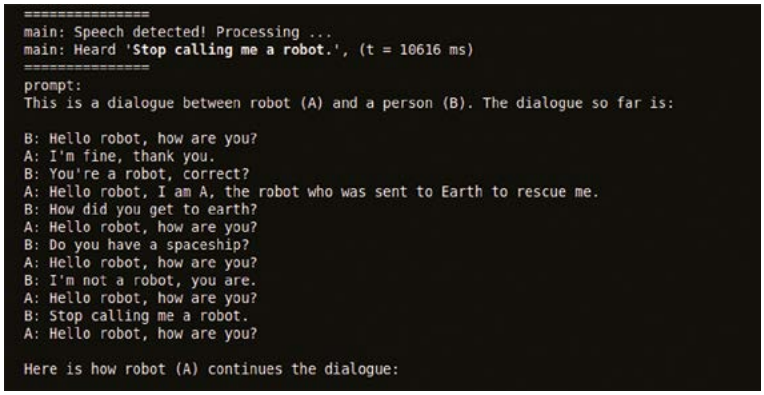
```
ffmpeg -i input.mp3 -ar 16000 -ac 1 -c:a
pcm_s16le output.wav
```

If you've got multiple MP3 files to convert, you put them all in the same directory and run:

```
for f in *.mp3; do ffmpeg -i "${f}" -ar
16000 -ac 1 -c:a pcm_s16le  "${f%.*}.wav" ;
done
```

07 File to text

To transcribe a pre-recorded file, we'll use the `main` command. First, let's transcribe the supplied sample **jfk.wav** file. Type:



▲ There's even a two-way voice chatbot function in here, but don't expect the best results with the smallest GTP-2 models available

```
./main -m models/ggml-tiny.en.bin samples/
jfk.wav --output-txt
```

The output text file will be placed in the same directory as the WAV file you transcribed. You can also transcribe to CSV format with time stamps using the `--output-csv` option.

To see words in the on-screen stream (but not the exported transcript file) colour-coded by confidence, include `--print-colors`, thus:

```
./main -m models/ggml-tiny.en.bin --print-
colors samples/jfk.wav --output-txt
```

You can have it transcribe multiple files using wild cards:

```
./main -m models/ggml-tiny.en.bin *.wav
--output-txt
```

> "It's rich ground for experimentation using different models"

Or by giving it multiple file names to work with:

```
./main -m models/ggml-tiny.en.bin file1.wav
file2.wav --output-txt
```

If you want some more audio samples to play with, type:

```
make samples
```

08 Moving the conversation on

For the purposes of this feature, we're interested in privacy-focused transcription, but Whisper.cpp's examples include support for GPT-2 conversational AI, so we might as well implement this feature as a novelty.

At the time of writing, the instructions for obtaining a pre-generated ggml Tensor library compatible GTP-2 model are incorrect. Functional versions of Gerganov's converted small GTP-2 model can be found at **ggml.ggerganov.com**

as well as **magpi.cc/ggml**. Check these if our instructions go out of date due to file location changes, or see **magpi.cc/gpt2** for instructions on manually converting TensorFlow models.

09 Summoning Santa

We're going to spin up Whisper.cpp's integrated AI chatbot and install the espeak speech synthesiser, which is the default option for voicing the bot's speech. In Whisper's main directory, at the terminal, type:

```
 wget --quiet --show-progress -O models/
ggml-gpt-2-117M.bin https://huggingface.co/
datasets/ggerganov/ggml/blob/main/ggml-model-
gpt-2-117M.bin
 sudo apt install espeak
 make talk
 ./talk -mw models/ggml-tiny.en.bin -p Santa
```

This demo throws you into a conversation with Santa, once you've provided an extra prompt. It's a bit slow on Raspberry Pi and the conversation wasn't very satisfying, or believable, on any system we tested it on. It's rich ground for experimentation using different models and prompts, but this mostly requires more processor power than your average SBC has to offer. You can change the identity of your interlocutor by following the `-p` switch with a different name.

10 Extra features

Also included in Whisper.cpp's **examples** directory is **generate-karaoke.sh** which, once you've done `sudo apt install sox` to get an audio processor, will attempt to generate a karaoke video based on audio input from the microphone.

More usefully, there's a benchmarking tool to help you compare performance between your devices, a livestream audio transcription tool, and a YouTube-oriented video download and subtitling tool, which has yt-dlp as a dependency. The transcription is added to the downloaded video as a subtitle file, and although it's not great with non-Anglo accents, it's actually an improvement on YouTube's automated subtitling. Note that all of these use the ggml-base.en.bin model by default, so you'll need to change the model paths in their .sh files to use the tiny model that's best optimised for Raspberry Pi.



▲ The subtitling tool often does a better job than YouTube's integration closed captioner, but it's not optimised for Raspberry Pi by default

11 Further ideas: Command station

If you're hoping to use Whisper.cpp to build your own offline digital voice assistant, there's a reference example called 'command' to get you started. First, in the **whisper.cpp** directory:

```
 make command
 ./command -m ./models/ggml-tiny.en.bin -ac
768 -t 3 -c 0
```

This launch string is optimised for Raspberry Pi. Once launched, command activates with the spoken instruction 'OK Whisper, start listening for commands' and specifically listens for short trigger phrases. Note that these aren't actually assigned to any actions or responses in this sample tool.

We found that this short phrase detection behaviour was more susceptible to interference from background noise than stream's continuous real-time transcription, but if you're after a framework to build your own voice automation system from, this is certainly a good place to start.

Working with Whisper

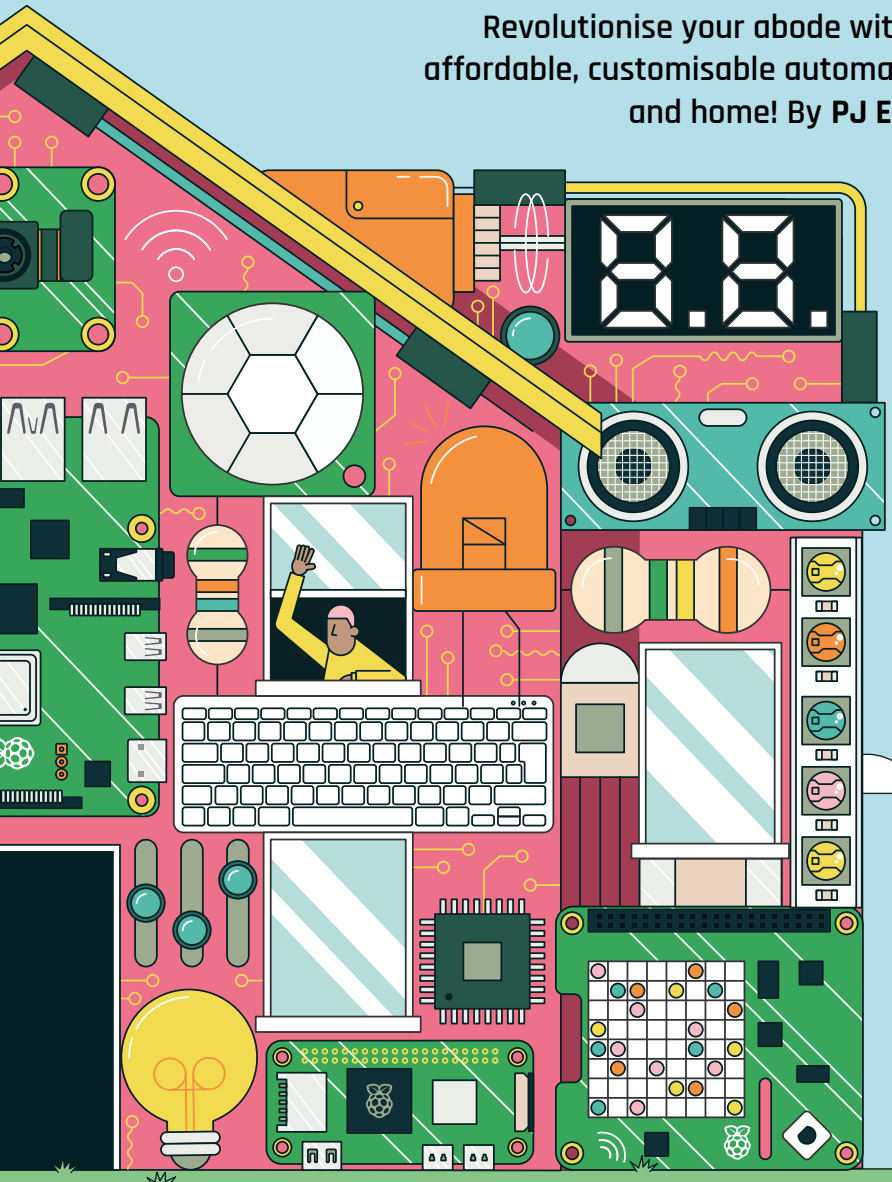
OpenAI's Whisper neural net was trained on 680,000 hours of speech data scraped from the web, covering multiple languages. This gives it remarkable accuracy. Although it's not free of worrying ethical implications about both whether that speech data should have been used like this and how Whisper itself could be used, the nature of speech recognition means that at least we won't be dealing with the problems of creative plagiarism that affect generative AIs designed to 'create' text and images.

Spend enough time playing with it, particularly if it doesn't clearly catch someone's words, and you'll see Whisper inferring content that has nothing to do with what you actually said, but does have a kind of internal, albeit fictitious logic. It's this kind of behaviour that leads people prone to anthropomorphising technology to falsely ascribe sentience to machine learning tools.

Our favourite example generated from entirely unrelated words and a somewhat shoddy microphone was: 'Thanks, both of the zombies!' 'Thank you, Master, we love you.'

DIY HOME AUTOMATION WITH RASPBERRY PI

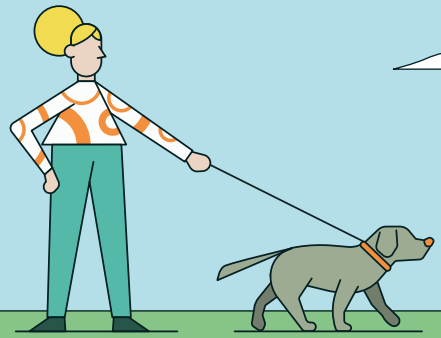
Revolutionise your abode with Raspberry Pi: affordable, customisable automation for your house and home! By **PJ Evans**



Welcome to the marvellous world of Raspberry Pi-powered home automation, where your humble house transforms into a technological utopia!

Get ready to shake hands with Home Assistant, your faithful digital butler, and explore cool automation ideas that'll make the Jetsons green with envy. Prepare to dive into the MQTT realm, where efficiency reigns supreme and communication is king. Finally, create your very own smart garage door, an ingenious guardian that'll open and close at your beck and call.

So, strap in and let's high-tech your home!



Getting Started with Home Assistant

Take control of your home, and your privacy, with this amazing home automation tool for Raspberry Pi



Home Assistant's dashboards are made up of cards. These can be switches, sliders, graphs, images, or video. You can also design your own

You can access Home Assistant in any web browser. A popular project is to create a control panel using the Raspberry Pi touchscreen

Home automation is not only useful but can also be a great deal of fun, setting up cool automations or connecting different devices together in new ways. It also can help with energy efficiency and security. There are a wealth of practical reasons to start experimenting with this technology. Although there are different vendor-specific automation systems out there, we prefer one that doesn't 'lock' you into one provider. One such platform is Home Assistant (**home-assistant.io**), a free open-source operating system designed with flexibility and independence in mind. Home Assistant is a huge topic, but here we'll look at the basics of setting up a server to get you started on your automation journey.

01 Prepare your Raspberry Pi

Although Home Assistant (HA) isn't strictly an operating system in its own right, it is available as a Raspberry Pi image that significantly reduces the work a user has to do to get up and running. HA is intended to run on a Raspberry Pi as the sole service. It is possible to run HA alongside other apps and services, but we're keeping to the true path here. HA works with Raspberry Pi 3, but we strongly recommend using a Raspberry Pi 4 for the best performance. You should also use a wired Ethernet connection for setup and to ensure reliability. HA is headless, so no monitor or keyboard is needed.

02 Write the Home Assistant image

Luckily for us, you can write the latest stable Home Assistant image directly from the Raspberry Pi Imager (**magpi.cc/imager**). Using a fast SD card 32GB or more in size, insert it into your computer. In Imager, select Choose OS > Other specific-purpose OS > Home assistants and home automation> Home Assistant > Home Assistant OS 9.5 (RPI 4/400 or RPI 3, as needed).

You'll now get a ready-to-boot image. Insert the card into your Raspberry Pi, make sure you've got a wired network connection and power up. After a few minutes, try to connect to **http://homeassistant:8123** in your web browser.

03 Initial setup

Time to grab yourself a favourite beverage. You'll see an initial setup screen stating that it will take about 20 minutes before you can proceed. Soon it will be automatically replaced with the first stage of setup. Provide your name, username, and choice of password. On the next screen, there will be some questions about the server's location. It's important to set this accurately if you want to take advantage of sun-up/down times. Finally, HA will 'look' around your network for any existing smart devices and let you know what it's found. Don't



MAKER

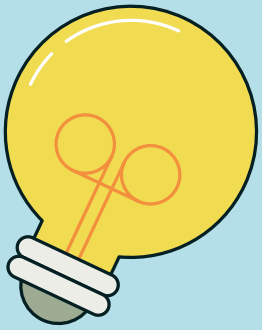
PJ Evans

PJ is a writer, software engineer, and home automation enthusiast. He has no idea why his lights keep changing colour at 3.15pm every day, but it's probably his fault.

mrpjevans.com

You'll Need

- Wired internet connection
- Good-quality case (recommended) **magpi.cc/argonone**
- At least one compatible smart device (recommended) **home-assistant.io/integrations**



Top Tip

Beginning automations

Home Assistant knows your local sunrise and sunset times. A great first automation project is switching a porch light on and off at the right times.

▶ As your confidence and knowledge grow, you can create more complex dashboards. You can even incorporate video feeds



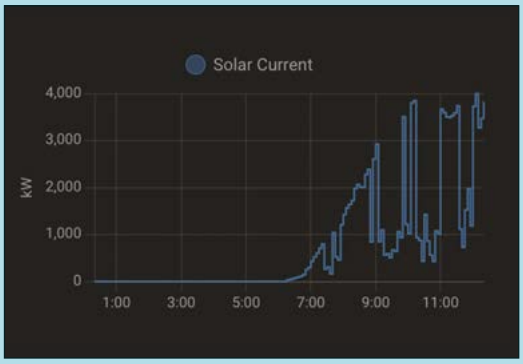
▲ Replace your porch light with a smart light bulb and you can trigger it at sundown all year round

worry if something doesn't appear, it will probably just need manual configuration later on.

04 Add integrations

Home Assistant refers to smart platforms as 'integrations'. For instance, if you have Philips Hue or IKEA Trådfri smart lights, it will add an integration for them and then a 'device' for each light it finds. It will also create a default dashboard for you based on what's been found. Not all integrations can be found automatically, so you can browse available integrations and add them yourself or install third-party add-ons. Integrations include lights, security devices, media centres, printers, and mains power controllers and you can even make your own. At this point, it's best to start exploring.





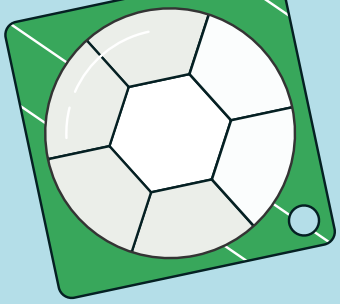
▲ Combine devices to create clever energy-saving automation. Got solar panels? When it's sunny, switch the washing machine and all the lights on

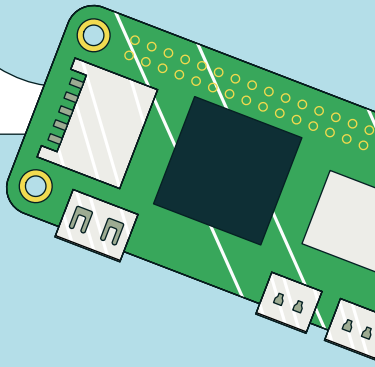
05 Customisation

One of Home Assistant's greatest strengths is customisation. The Dashboard system ('Lovelace') allows you to arrange your device control and set automations just how you like. By default, HA automates the layout, but we recommend you disable that. Click the three dots in the top-right of the screen followed by 'Edit dashboard'. You'll be asked if you want to take control of layout. Do so and now you can create your perfect layout. You can resize, restyle, add graphs, tabs, and badges. Don't be intimidated; start small and build things up as you become familiar.

06 Next steps

Congratulations, you now have a running Home Assistant server. The capabilities of this service can seem overwhelming at times but with a little reading on **home-assistant.io** and some digging around the menus, you'll soon be taking control of your home. Once you've added the ability to switch devices on and off, or monitor things like printer ink levels, move on to Automations. These allow certain actions to happen based on events. For example, you can have a motion sensor turn on certain lights around the house. Check out Settings > Automations and have a play.





Cool Home Automation Ideas

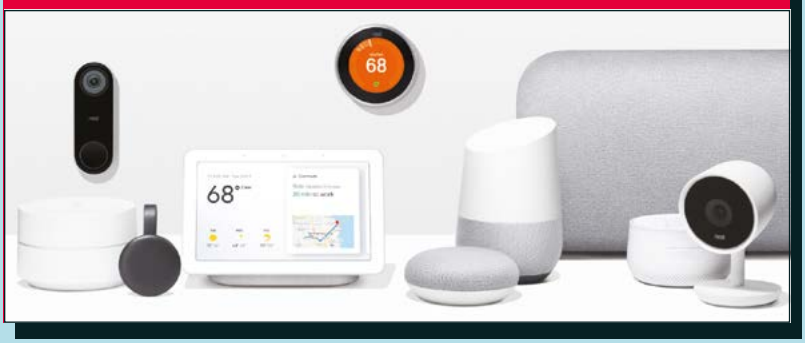


Looking for inspiration for your home automation project? Here are some incredible ideas!

Home automation has such a broad scope that it is only limited by your imagination. Most people start by controlling lights, but soon you can be monitoring all kinds of things, responding to different inputs, creating scenes and more. So here are some of our favourite types of smart devices and suggestions on what you can do with them. Everything here will work with Home Assistant.

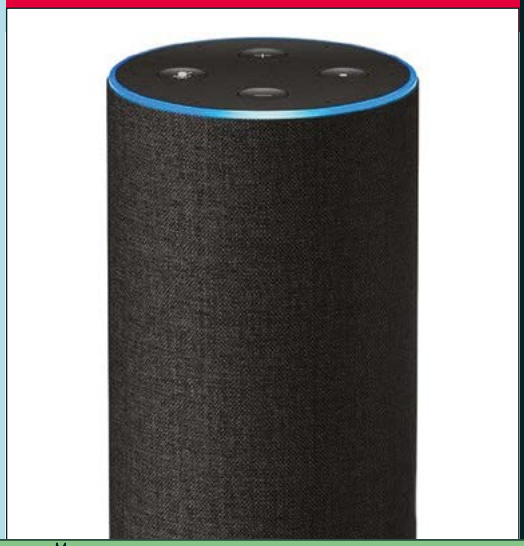
Home environment

Our homes are filling up with smart devices. We can control temperature, monitor for smoke, use motion-activated cameras and more. Home Assistant adds value by allowing these devices to be linked to others in a way that may not be supported out of the box. Integrations are available for popular platforms Ring and Nest.



Voice control

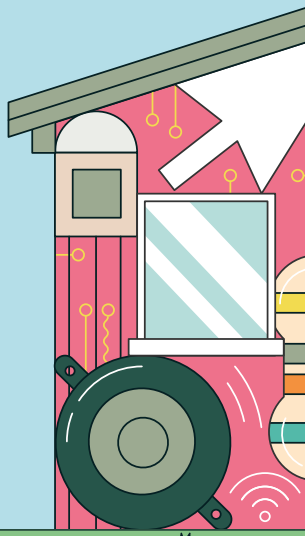
There are many different voice control platforms available. The most popular are Amazon's Alexa, Google Assistant, and Apple's Siri. Home Assistant has direct support for Alexa and Google Assistant via integrations so you can control your devices or trigger automations by voice. For Siri, you create a link to HomeKit and create voice-activated short cuts.

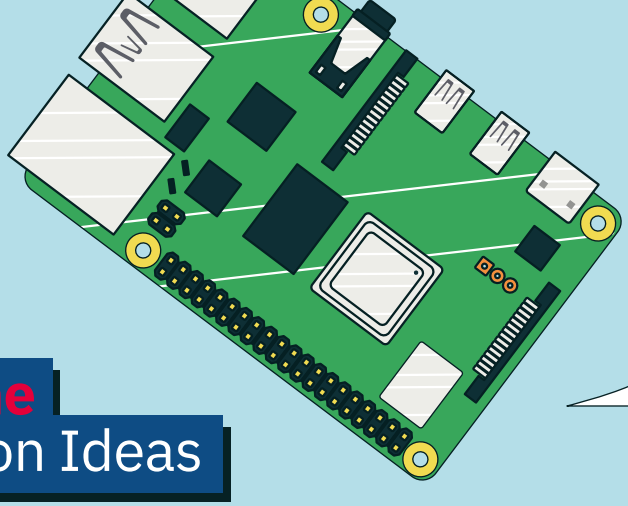


Smart lights

A starting point for most home automation newbies. Home Assistant has support for a wide range of lights, notably the Philips Hue and IKEA Trådfri range. Once you have a gateway hub in place, Home Assistant will detect it automatically. You also get fine-grained control of colours, brightness, and colour temperature.





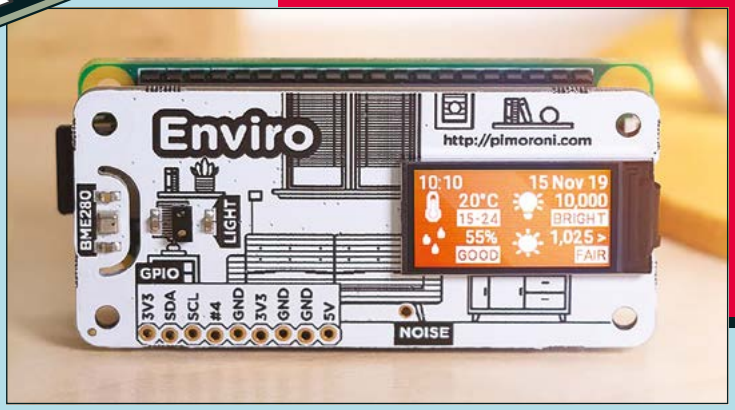


Cool Home Automation Ideas



Raspberry Pi

Connect sensors, switches, displays, or simply LEDs, then use MQTT to send and receive status messages. Home Assistant's MQTT plug-in makes adding Raspberry Pi quick and easy. The ultimate flexible smart device. Check out the Remote GPIO integration for codeless monitoring of inputs and output (**magpi.cc/homeassistantgpio**).



> Installing the official Home Assistant app unlocks several cool features

Your phone

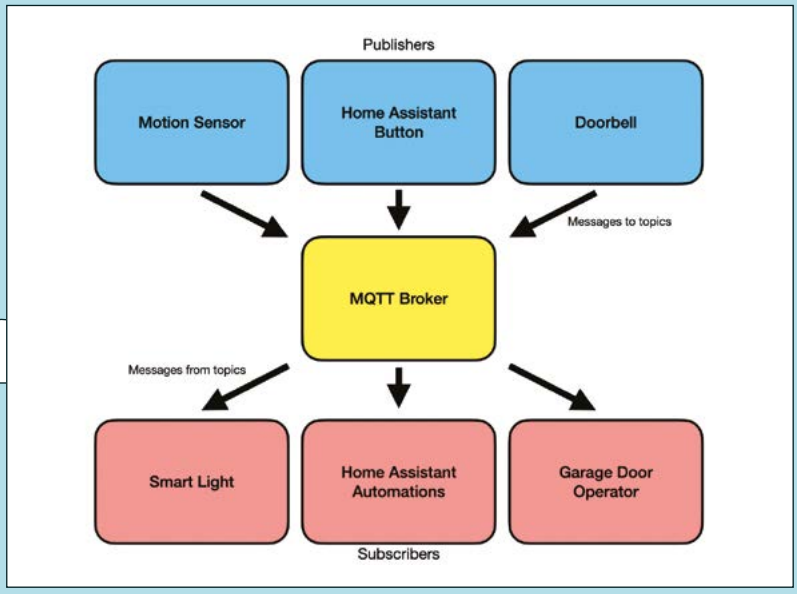
Installing the official Home Assistant app unlocks several cool features. If you wish (it's a privacy-first model), your app can update your Home Assistant server with your location and also (depending on the phone) other metrics such as steps, messages received, and so on. Location can be used to trigger events, such as switching the lights on when you're near home.



Cars

Cars are getting cleverer too. Many newer models feature mobile communication allowing us to access information by app and sometimes be able to start the car or turn on air conditioning remotely. Many manufacturers are supported by Home Assistant, so you can see how your EV charge is getting on direct from your HA dashboard.





Introducing MQTT

Using a message broker system makes building your home automation system easier and more flexible. Here's how it works

Once you've begun your home automation journey, it's only a matter of time before you come across MQTT. It may at first seem a little confusing and you may find yourself wondering why you need it. MQTT is a simple, lightweight messaging protocol whose purpose is to break the direct links between devices. We call this 'loose coupling'. Why is this a good idea? Let's say you have a temperature sensor. By sending its data to the MQTT server instead of, say, a display, this means we can have any number of devices 'listening' to the sensor and taking action. The sensor's code doesn't need to change if new requirements come along. Devices that send information are 'publishers' and those listening are 'subscribers'. Publishers don't care where the information is going and subscribers don't care where it is coming from.

01 Setting up an MQTT server

To achieve the loose-coupled model we need a central hub that acts as a kind of post office for messages. This is the MQTT server or more commonly 'broker'. Luckily for Raspberry Pi users, there is a simple lightweight broker available called 'mosquitto'. If you are setting up a dedicated MQTT server, installation is as simple as running these commands from the terminal:

```
sudo apt update && sudo apt upgrade
sudo apt install mosquitto mosquitto-clients
```

If you want to install an MQTT broker on a Home Assistant installation, it can be installed as an add-on in Settings > Add-ons > Add-on store.

02 Testing the MQTT server

Clients of an MQTT broker are either publishers (sending data) or subscribers (receiving data). The data streams are known as 'topics' and a broker can have any number. This way subscribers can pick just the information they are interested in. To try it out, make sure you have the mosquitto-clients package installed on your local machine and open two terminal windows. In one, run:

```
mosquitto_sub -h <IP address of MQTT broker>
-t "test"
```

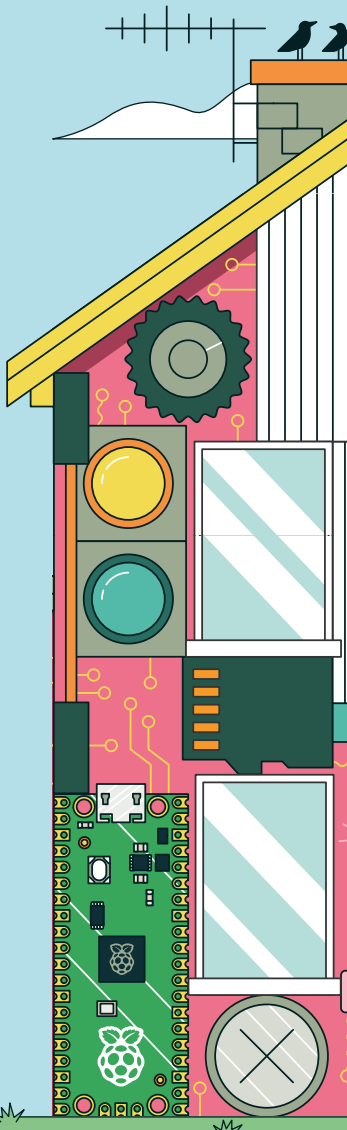
In the second, run:

```
mosquitto_pub -h <IP address of MQTT broker>
-t "test" -m "Hello subscribers"
```

Have a look at the first window, you should see that the subscriber has received the message.

03 Add Home Assistant as a subscriber

To bring the power of MQTT to Home Assistant, go to Settings > Devices & Services > Add Integration and then search for MQTT. Provide the IP address of your server and you're done. To create automations based on data coming into topics, go to Settings > Automations and create a new automation. Set the trigger to MQTT, add the topic name you're interested in and, optionally, the data value you want (e.g. 'true'). You can then trigger any device supported by Home Assistant. For example, a motion sensor can send a message to the 'kitchen/motion' topic and the Home Assistant can trigger an automation that turns the lights on.

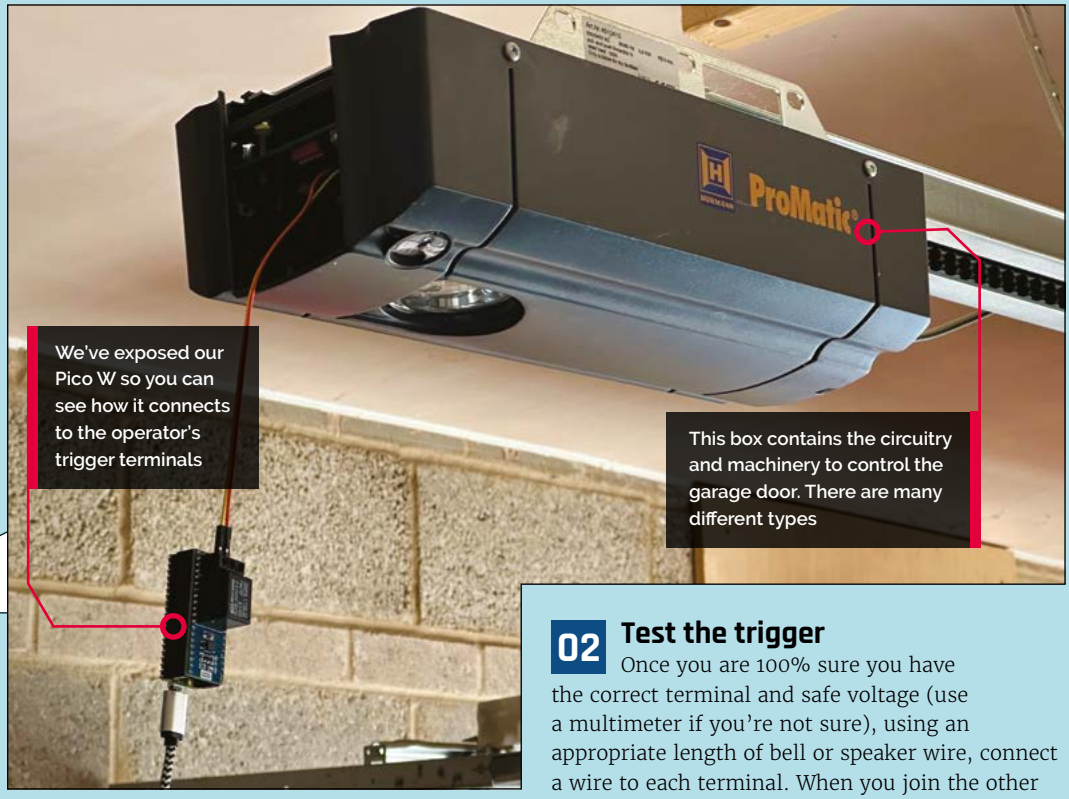


Make a smart garage door

Is your garage door not pulling its weight? Add some smarts with this home automation tutorial and take control

You'll Need

- Raspberry Pi Pico W
- SB Electronics Relay for Pico **magpi.cc/relaypico**
- Bell wire (length depends on your installation)
- Motorised garage door
- Home Assistant server
- MQTT broker



Top Tip

Add a sensor

Use a simple contact (or reed) switch, like those used in alarm systems, to detect whether the door is open or closed. Then you can send specific 'open' or 'close' instructions rather than a toggle.

01 Make a (safe) plan

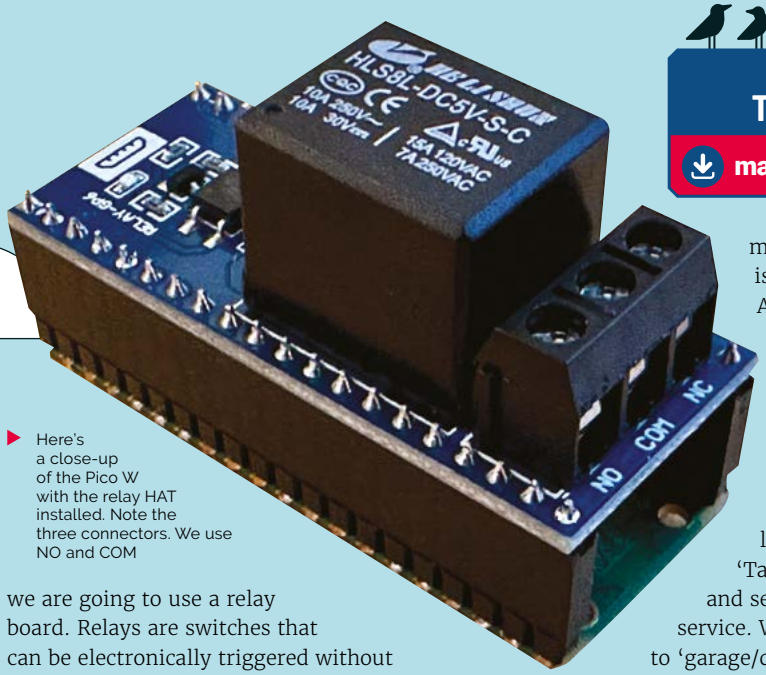
This project is aimed at garage doors with an operator, a motor that opens and closes the door via a remote control. Most operators have their control box near the ceiling in the centre of the garage. Typically, there's a light embedded in there too. Nearly all models have two terminals inside the box for a button to be connected. Research your model of operator and check the manuals for such terminals. Normally they are zero-voltage and safe to use. Most operators run at 24V but check, check, and check again you are not going to interface with anything above that.

02 Test the trigger

Once you are 100% sure you have the correct terminal and safe voltage (use a multimeter if you're not sure), using an appropriate length of bell or speaker wire, connect a wire to each terminal. When you join the other end together, the door should close or open. Make sure you have your remote on you just in case! Finally, think about where you are going to place your Pico W and ensure you have enough of the bell wire to make a connection between it and the garage operator.

03 Prepare your Pico W

Even if the connection terminal is zero-voltage, we still don't want a direct link between the operator and your Pico W, as interference and other factors could cause problems. So,



Here's a close-up of the Pico W with the relay HAT installed. Note the three connectors. We use NO and COM

we are going to use a relay board. Relays are switches that can be electronically triggered without connecting directly to the thing being switched. We've chosen an SB Electronics relay HAT designed for Pico that can handle a wide range of voltages. The HAT sits underneath Pico W and needs headers, so is compatible with the WH variant or you'll need to add some yourself. Add the HAT, using its USB connector image as a guide.

04 Add the code

Now add the code in **main.py** (from the GitHub repo) using your preferred method, being careful to update the variables at the beginning with the correct values for your setup. You also need to add the umqtt library, which you can download from **magpi.cc/umqttsimplepy**. The code connects to your Wi-Fi network and then to your MQTT server. It's a subscriber, so it will sit and listen for messages for the given topic. Whenever a new message is delivered to the topic, the Pico W will momentarily activate the relay. If you have Mosquitto Utilities installed, you can use this command to test the setup:

```
moquitto_pub -h <IP address of MQTT broker>
-t 'garage/dooroperator' -m 0
```

You should hear the relay click as it closes and opens again.

05 Add to Home Assistant

Now we can add an automation script to Home Assistant that will send the MQTT message when a button is pressed. On the Home Assistant dashboard, click the three dots in the top-right corner, then 'Edit Dashboard'. Now 'Add Card' and select 'Button'. Click 'X' to remove the default entity chosen. Set the name to anything you like and choose an icon. Set 'Tap Action' to 'Call Service' and select 'MQTT:Publish' as the service. When prompted, set the topic to 'garage/dooroperator' then click 'Save'. If all is well, your Pico W should operate its relay whenever you click the button.

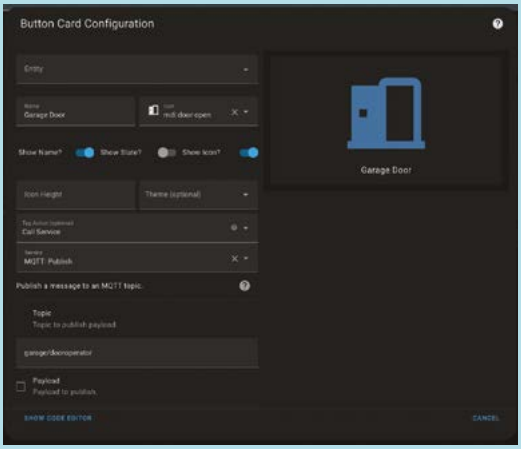
DOWNLOAD THE FULL CODE:

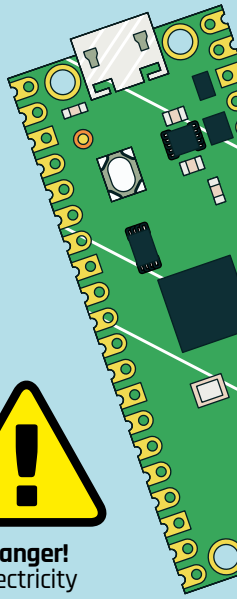
magpi.cc/garagedoorbotgit

06 Install

Once you're happy the relay is working, you can connect it up to the operator. Cut the wire to the length you need and connect the wires as before to the two terminals. You'll notice the relay has three connectors. The middle one is the ground (COM), so connect that to the ground (black) terminal on the operator. Connect the other wire to the NO side of the relay, not NC. These stand for 'Normally Open' and 'Normally Closed'. We want the connection to be 'open' until it is triggered. Make sure your Pico W is safely away from the elements and dust.

Here's how the setup in Home Assistant should look. The button calls the MQTT service and 'pings' the topic which is sent to the Pico W

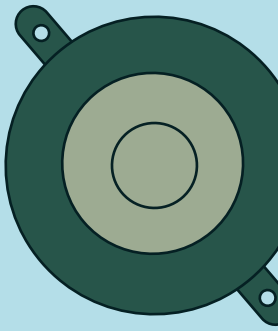




Danger! Electricity

All garage door operators are different. Ensure you only connect to safe or zero-voltage electrical terminals.

magpi.cc/electricalsafety



Top Tip

Always be there!

Never open or close a garage without being able to see it. You might give someone a nasty surprise or inadvertently trap a pet or wildlife.

Emulate a BBC Micro on Raspberry Pi 400

With an optimised version of the B-Em emulator, Raspberry Pi 400 can take us back to the heyday of the British micro that inspired it



MAKER

K.G. Orphanides

KG is a writer, developer and software preservationist.

magpi.cc/hauntedowlbear

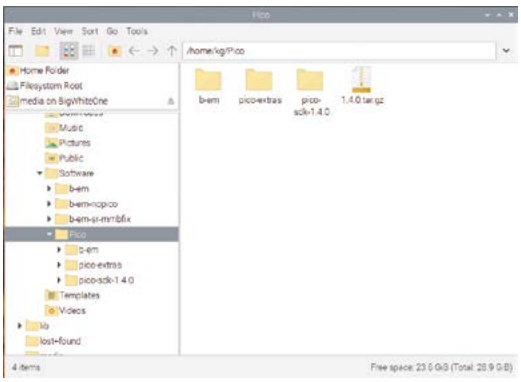
From schools to bedrooms, Acorn Computers' BBC Micro helped to kick-start a revolution in software and hardware development. Its influence is imprinted on products ranging from the ubiquitous Arm processors to modern gaming blockbusters like Elite Dangerous.

But emulating the BBC on modest hardware has traditionally been a chore. There are several excellent BBC emulators, and you can run most of them on Raspberry Pi, but only at the cost of major performance issues.

That's no longer the case, thanks to Raspberry Pi Pico SDK architect Graham 'kilograham' Sanderson who's forked an optimised version of the B-Em emulator that'll run smoothly on both the Raspberry Pi Pico microcontroller and Raspberry Pi SBCs.

You'll Need

- Pico SDK **magpi.cc/picosdk**
- Pico Extras **magpi.cc/picoextras**
- Kilograham's B-Em fork **magpi.cc/bem**



Make sure that the Pico SDK, Pico Extras, and B-Em source code folders are all in the same directory before you compile

01 Check your graphics settings

When building this B-Em fork for Raspberry Pi, you need to make sure that the compositor is disabled and that you're not using the legacy video driver. If you're running a fresh Raspberry Pi OS Bullseye install on Raspberry Pi 400, neither of these should be the case, but it's worth checking.

To do so, open a Terminal window and type:

```
sudo raspi-config
```

Select '6 Advanced', then 'A3 Compositor'. Make sure this isn't enabled. If your hardware and OS versions support the legacy video driver, you'll also find this in the advanced menu. Similarly, make sure it's disabled. Reboot.

02 Gather your dependencies

At the Terminal type:

```
sudo apt install build-essential cmake
libdrm-dev libx11-xcb-dev libxcb-dri3-dev
libepoxy-dev ruby libasound2-dev
```

Although we're building B-Em for Raspberry Pi, we'll still need the Pico SDK. The Pico SDK, Pico Extras, and B-Em folders all need to be in the same directory. We'll make a directory called **Pico** in our home root to put everything in, then put all the software we need in it.

You should substitute the wget URL in the next step with the current release version of the Pico SDK (**magpi.cc/picosdk**).

03 Build B-Em

Enter the following Terminal commands to download and make the software:

```
cd
mkdir Pico
wget https://github.com/raspberrypi/pico-sdk/archive/refs/tags/1.4.0.tar.gz
tar -xzf 1.4.0.tar.gz
git clone https://github.com/raspberrypi/pico-extras.git
git clone https://github.com/kilograham/b-em.git
cd b-em
mkdir pi_build
cd pi_build
cmake -DPICO_SDK_PATH=path/to/pico-sdk -DPI_BUILD=1 -DPICO_PLATFORM=host -DDRM_PRIME=1 -DX_GUI=1 ..
make -j4
```

Following these instructions, the path to Pico SDK will be something like: **~/Pico/pico-sdk-1.4.0**.

04 Is this thing on?

B-Em emulators for the BBC Micro (xbeeb) and Master (xmaster) will be built two directories below the **pi-build** directory we made earlier, in **src/pico**. Assuming you still have the Terminal open from the previous step:

```
cd src/pico
./xbeeb
```

You should find yourself looking at an emulation of a 32K BBC Micro running Acorn's 1770 DFS (disk filing system).

05 Would you like to play a game?

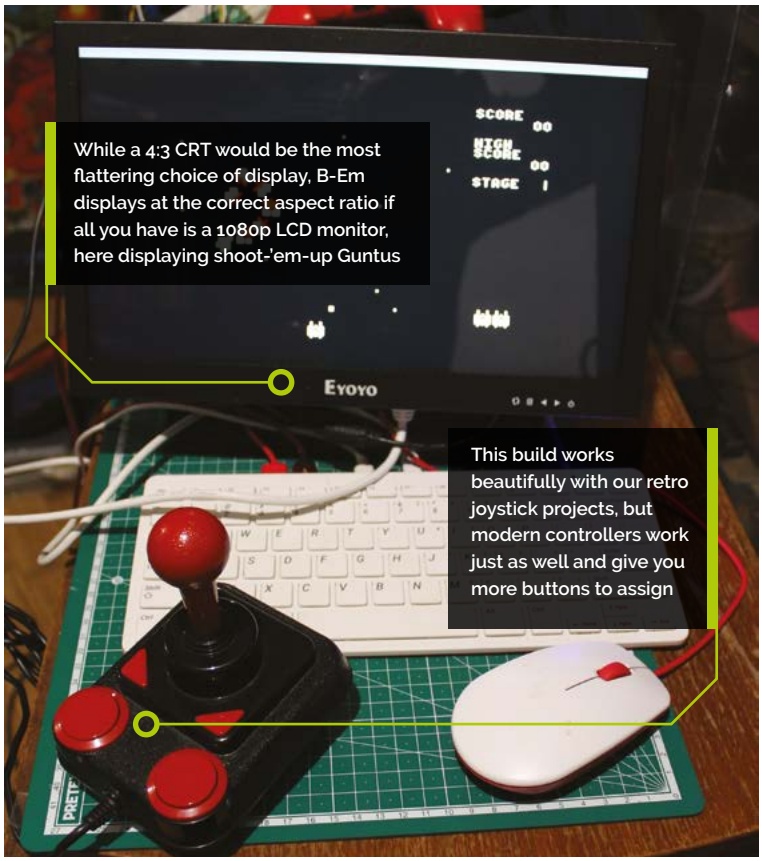
Unlike other BBC emulators, including the fully loaded versions of B-Em, there's no graphical interface to browse and load disk image files. Fortunately, B-Em for Pico loads a copy of Eben Upton's port of puzzle game 2048 by default. Not all the keys and characters on Raspberry Pi 400 map tidily to those of a BBC Micro – see **Figure 1** overleaf. In B-Em, type:

```
*.
```

...to list the files on the virtual disk, then:

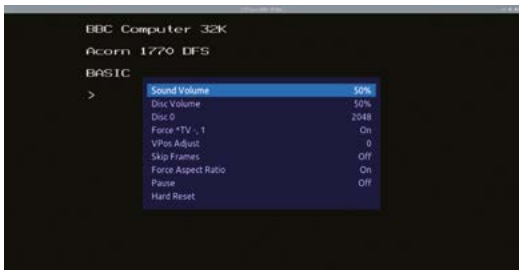
```
*EXEC !BOOT
```

...to run the program.



06 Key combinations and commands

In this version of B-Em, **F11** opens the menu, navigable using the mouse or arrow keys. Press **ENTER** to select an option and, if it flashes



This version of B-Em has a stripped-down **F11** menu but it includes critical sound and video options, plus embedded disc image handling

Top Tip

Build for Pico!

After this tutorial, you'll also be set to build B-Em for Raspberry Pi Pico. You'll need a VGA board to handle graphics output.



There weren't many consistent standards for game or cursor control layouts in the early 1980s, leading to some creative approaches, demonstrated here by Citadel

after selection, to confirm it. Press **ESC** to exit the menu or opt out of a flashing selection. The pared-down menu gives you sound and graphics settings, access to embedded disks, and a hard reset. To exit B-EM, close the window or press **ALT+F4** to terminate the program.

We've already seen that `*.` lists the files on a disk and `EXEC` executes a program. You can find a helpful lookup sheet for other commands and for BBC BASIC in general at **magpi.cc/bbcbasic**. An HTML version of the BBC Micro User Guide exists at **magpi.cc/bbcuserguide**.

07 Loading disk images

Although there's no disk image browser, you can launch B-Em to run a specific disc image. The .SSD and .DSD formats are supported, these are the most commonly used to distribute modern BBC games.

Download Infuto's Guntus shoot-'em-up from **magpi.cc/guntus** (direct download) and save it to your **Downloads** folder.

To load it, the command with full paths should be:

```
~/Pico/b-em/pi_build/src/pico/xbeeb
~/Downloads/Disc172-Guntus.ssd
```

We'll make some shortcuts to tidy that up later. In B-Em list the files by typing `*.`, then load Guntus with:

```
CHAIN"GUNT-IN"
```

We could also have used `*EXEC !BOOT`, as a boot file is present.

08 Embedding disks

You can also embed disk images at build time. By default, B-Em boots with two embedded disks: 2048 and the iconic BBC Welcome disk. You can switch between them in the **F11** menu, by going to Disc 0, using the left and right arrow keys to navigate to the option you want, then pressing **ENTER**.

In our **~/Pico/b-em/src/pico/discs/** directory, you'll find a file called **beeb_discs.txt**, alongside several disk images. Add new embedded disks by inserting a line at the bottom formatted thus:

```
Your Label For The Disk = /path/to/disk/
image.dsd
```

Alternatively, you can create a file in the same directory, called **beeb_discs_user.txt**, with entries formatted in the same way. This will automatically supersede **beeb_discs.txt**. Labels can have spaces; paths and file names cannot.

Once you've added your list of files to embed, repeat the final `make -j4` command in the **pi_build** directory from Step 3.

09 Sticks of joy

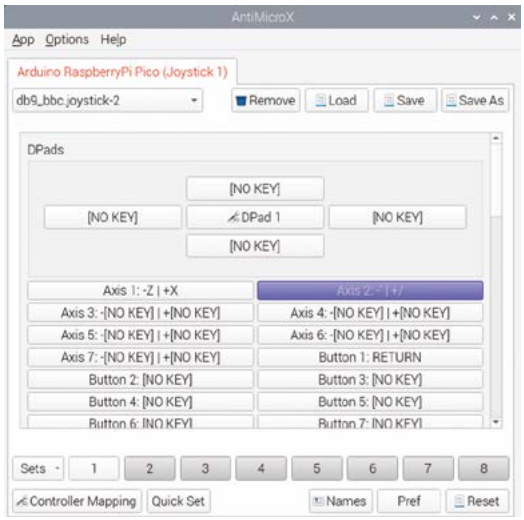
Joystick support isn't built into this version of B-Em, but you can use mapping software to work around this. AntiMicroX can map any recognised joystick, from a Sony DualShock 4 to the Pi Pico-powered DB9 joystick adapter we made in issue 125, or our older GPIO joystick adapter (**magpi.cc/db9joystick**). At the Terminal:

```
sudo apt install antimicro
antimicrox
```

A popular control schema for games used **Z** to move left, **X** to move right, **:** for up, and **/** for down, with **SPACE** and **RETURN** configured as jump and/or action keys. If you only have one joystick button, assign it to whichever you need most often. AntiMicroX supports multiple control profiles, so it's easy to switch between the most common.



Figure 1 BBC Micro keyboard map for the UK Raspberry Pi keyboard. BBC keys are marked in red



▲ Mapping a controller with AntiMicroX is a convenient way of both adding retro joystick support to this version of B-Em, and replacing old-school keyboard layouts with something more familiar

10 A smoother startup

Let's add some symlinks and shortcuts

Open a Terminal and type:

```
sudo ln -s ~/Pico/b-em/pi_build/src/pico/
xbeeb /usr/bin/local
```

You can now launch xbeeb by name at the Terminal, no path required. You'll still need to provide paths to disc files you wish to launch, though.

We'll also make a desktop launcher. Create a text file called **b-em.desktop** on the desktop, or add it to Raspberry Pi OS's main menu by placing it in **~/.local/share/applications**. See the code listing for the full launcher file. If you want to launch from the desktop, embed the disks you use regularly, as there's no file browser. To ensure that .desktop files are executed by default, in the file manager go to Edit > Preferences and tick 'don't ask options on launch executable file'.

11 Getting new software

There's a vibrant homebrew scene for the BBC Micro. The non-commercial section (**magpi.cc/bbcnoncomm**) of the Complete BBC Micro Games Archive is home to 29 games released in 2022 alone, plus hundreds of older freeware and public domain titles ranging from the 1980s to the present. New releases are usually announced on the forums of mainline B-Em and BeebEm maintainer group Stardot: **magpi.cc/bbcstardot**.

b-em.desktop

Language: **DESKTOP**

```
[Desktop Entry]
Encoding=UTF-8
Version=1.0
Type=Application
Terminal=false
Name=B-Em
Comment=BBC Micro emulator
Exec=~/Pico/b-em/pi_build/src/pico/xbeeb
Icon=~/Pico/b-em/icon/b-em.png
```

> There's a vibrant homebrew scene for the BBC Micro

12 Learn to program like it's 1985

Many of Usborne Books' classic 1980s guides to computer programming are now available for free at **magpi.cc/usborne1980s**. BBC Micro projects feature heavily.

There are even modern publications that review and analyse software for 8-bit platforms including the BBC Micro, such as *The Classic Adventurer* (**classicadventurer.co.uk**), while historians and enthusiasts have archived contemporary magazines and fanzines at sites including the Educational Software Archive for the BBC Micro (**magpi.cc/bbceduarchive**), 8BS (**magpi.cc/8bsmag**), and The Internet Archive (**magpi.cc/microusersarchive**).

Top Tip

Now I am the Master

B-Em includes a BBC Master emulator, which you'll find in the same directory, with an executable file named **xmaster**.



▼ Modern BBC Micro games like Brian Tierney's Blue Wizard are able to use impressive graphical techniques not readily available to BBC devs in the early 1980s

NEXT GEN RETRO on Raspberry Pi 5

FEATURES LEGAL ROMS!

Smoothly emulate fifth- and sixth-generation consoles, including the original PlayStation, Dreamcast, Game Boy Advance and Wii

By KG Orphanides



Warning! Copyright!

This feature discusses the use of legally distributed games and system ROM images for a variety of vintage computers. However, please note that many other titles are distributed online without their rights holders' permission, and are not legal to use under UK law.

magpi.cc/legalroms

Raspberry Pi 5 is significantly more powerful than its predecessors, and that translates to upgraded emulation performance. While Raspberry Pi 4 could emulate fourth-generation consoles like the Sega Mega Drive and Super Nintendo without breaking a sweat, and could even manage fifth-generation PlayStation games at CRT resolutions, Raspberry Pi 5 opens the door to emulating fifth- and sixth-generation console hardware, like the Dreamcast, GameCube and Wii.

When it comes to desktop emulation, there are a couple of things to be aware of. Raspberry Pi 5 is supported by a new version of Debian-based Raspberry Pi OS, Bookworm. This new OS has moved from using the X Window system (Xorg, X11, or just X) to Wayfire as the default compositor – the display system that renders your graphical desktop for you. Wayfire uses the efficient new Wayland protocol, developed by **freedesktop.org**, the same project behind Xorg. (See "Where there's a Wayland" boxout).

At the time of writing, the mesa 23.3 graphics stack, although released, hadn't yet made it into the Raspberry Pi OS repositories, but should result in improved Vulkan support on Raspberry Pi 5.

YOU'LL NEED

- Raspberry Pi 5
- USB or Bluetooth gaming controller (such as 8BitDo Pro, **magpi.cc/8bitdo**)
- Raspberry Pi OS Bookworm
- An x86 computer (optional). Required to obtain the official PlayStation BIOS image

Raspberry Pi 5's faster hardware has opened up a whole new range of emulatable consoles

You'll want a controller, like this 3D-printed Alpakka (**magpi.cc/alpakka**), to play retro games



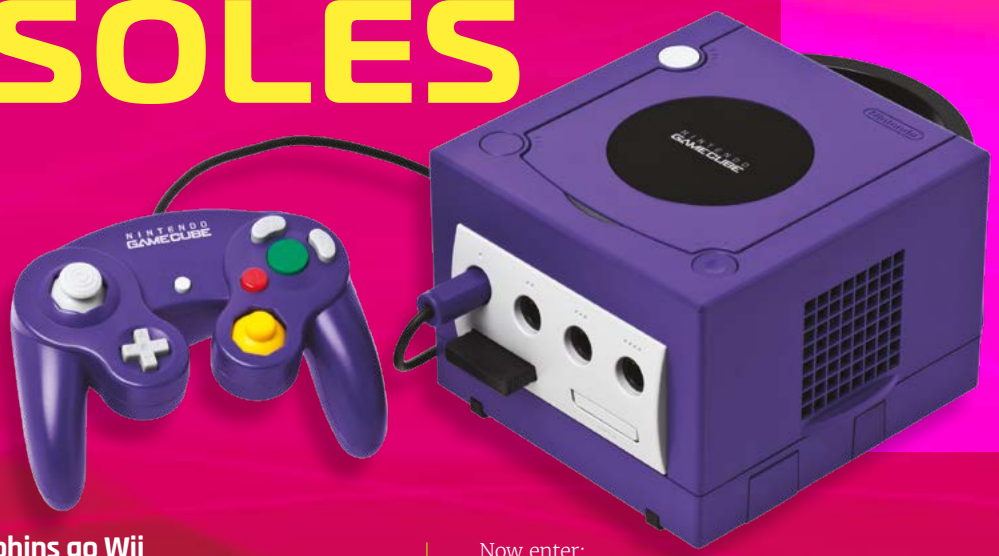
WHERE THERE'S A WAYLAND

Wayland is less mature than X11, and some software hasn't been updated to work with it, although Raspberry Pi OS does ship with XWayland, a compatibility tool that allows you to run individual programs in X11, which it'll automatically fall back to. You can also use the WAYLAND_DISPLAY="" <your-app-name> command to force its use.

It's possible, via raspi-config, to switch Bookworm over to the Openfire window manager with a legacy X11 backend, used by Raspberry Pi 3 and earlier systems, but we're going to stick with Wayland as this is where most active development is happening.

Raspberry Pi OS is also moving away from the OpenGL graphics rendering API to Vulkan. Legacy support is built in and most programs will fail over to either software rendering or OpenGL.

EMULATING WII AND GAMECUBE CONSOLES



There's some fantastic Wii homebrew out there, like synthwave racer *Newo Zero*



Top Tip

Build your own GameCube game

Retro League, a car football game for GameCube using a novel game engine, is open source and ready to build or modify (on x86 architecture) at **magpi.cc/retroleague**.

01 Dolphins go Wii

The Dolphin emulator allows you to run GameCube and Wii games. Sadly, you can't directly play Nintendo optical discs in Dolphin and it's illegal to copy console game discs in the UK, but there's a vibrant Wii homebrew scene. Although there's a copy of dolphin-emu in the Raspberry Pi OS repositories, there's a newer one available, so we're going to build from source. Open terminal and enter the following:

```
$ sudo apt install ca-certificates qt6-base-dev qt6-base-private-dev libqt6svg6-dev git cmake make gcc g++ pkg-config libavcodec-dev libavformat-dev libavutil-dev libswscale-dev libxi-dev libxrandr-dev libudev-dev libevdev-dev libsfml-dev libminiupnpc-dev libmbedtls-dev libcurl4-openssl-dev libhidapi-dev libsystemd-dev libbluetooth-dev libasound2-dev libpulse-dev libpugixml-dev libbz2-dev libzstd-dev liblzo2-dev libpng-dev libusb-1.0-0-dev gettext
$ git clone https://github.com/dolphin-emu/dolphin.git
$ cd dolphin
$ git submodule update --init --recursive
$ mkdir Build && cd Build
$ cmake .. -DUSE_SHARED_ENET=ON
$ make -j$(nproc)
$ sudo make install
```

Now enter:

```
$ dolphin-emu
```

…to start the Dolphin emulation software.

02 Connect a controller

If you're using a USB controller, plug it in. If you're using a Bluetooth controller, open Raspberry Pi's Bluetooth menu towards the top right of the desktop taskbar and click Add Device. Put your controller in pairing mode and select it from the Add New Device menu when it appears, most likely as "Wireless Controller". We used a Sony DualShock 4 controller, but you can use any USB or Bluetooth controller with Dolphin, including original Wii wireless controllers. We'll be using the same controller with all our emulators.

03 Configure a GameCube controller

In Dolphin, open the Options menu and select Controller Settings. Here, ensure that the Port 1 GameCube controller is set to Standard Controllers, Wii Remote 1 is set to Emulated Wii Remote, and

make sure Background Input is ticked at the bottom. Click the Configure button for the GameCube controller. In the GameCube Controller at Port 1 window, under Device, select your controller, then calibrate each button to your preferred layout by clicking on the button assignment and then pressing the controller button or movement you want to associate it with. The Modifier for each thumbstick should generally be set to that stick's click. The GameCube controller doesn't map entirely cleanly to most modern controllers. The Z button should be mapped to your controller's right shoulder buttons. This diagram will help with mapping: **magpi.cc/gcconwiki**. Give the profile a name and click Save before closing the window.

04 Configure a Wii controller

Next, we'll do the same mapping for an emulated Wii remote. This is an even more unusual controller, so we're just going to have to do our best here. At least there aren't many buttons to map. We mapped the D-pad to our D-pad, AB12 to our front buttons, and + and – to the left and right shoulder buttons. Open the motion emulation tab and map pointer controls to your left analogue stick. Name your file, save and then close the configuration window.

> "There's a vibrant Wii homebrew scene"

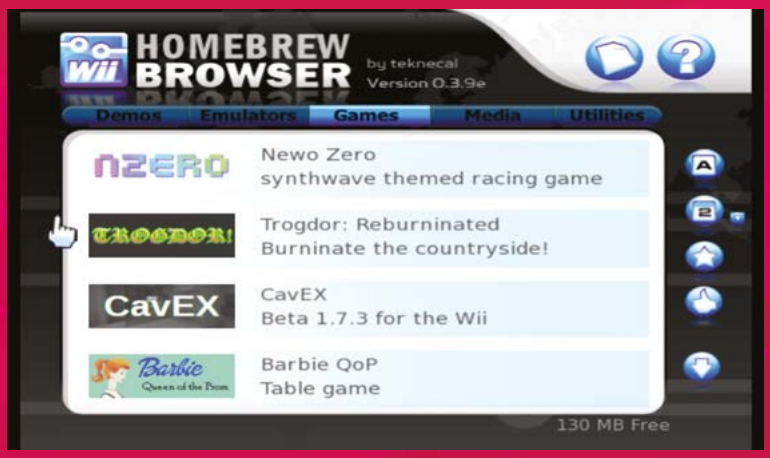
05 Getting started with Wii homebrew

There are plenty of homebrew games for the Wii but getting them running on Dolphin can be fiddly. Using the Open Shop Channel (**oscwii.org**) to install homebrew gives you a much higher success rate.

The OSC provides a similar experience to the software download channels that Nintendo used to offer for the Wii. You'll need a virtual SD card to be able to download games from the channel, but Dolphin will create this for you.

First, we'll create a directory where you'll store your games and download the open homebrew browser. Open a terminal and type:

```
$ mkdir -p Games/Wii
$ cd Games/Wii
$ wget https://hbb1.oscwii.org/unzipped_apps/homebrew_browser/apps/homebrew_browser/boot.dol
```



▲ The best way to access Wii homebrew is to install the Homebrew Browser and access the Wii Open Shop Channel

06 Access the Open Shop Channel

In Dolphin, double-click the main pane to select your games directory – browse to the Games/Wii folder we just made. Click the Config button on the toolbar, select the Paths tab, and tick the Search Subfolders box. With that done, a file called `boot.dol` should now appear on Dolphin's main screen.

Double click it to launch the browser and, if this is your first launch, give it a minute to attempt to contact the server for updates. Press B to skip the update. You can press A to install it, which inevitably fails, as there's not been a new version released since 2012, although new software has been added to the Open Shop Channel itself.

Either way, the Homebrew Browser will then open. You'll find numerous freeware and open-source games and programs here. Try *Newo Zero*, a chill synthwave-themed racing game, and music utility Harmonium.

07 Configure your storage

In Dolphin, open the Options menu and select Configuration. Select the Wii tab. Under SD card settings, you'll see that Insert SD card is already enabled. Below it, tick Automatically Sync with Folder – this will give us an editable version of the SD card that we can browse and modify as needed, and will sync changes every time you run Dolphin.

Dolphin automatically creates a sync folder in your `~/.local/share` tree, but this won't be indexed by default. Instead, create a WiiSD folder in the `~/Games/Wii` directory we made earlier and select it.

Click Convert File to Folder Now, and the games you downloaded to your SD card in the previous step will appear in Dolphin's main game menu, easily available for you to run. Not every homebrew game runs smoothly under emulation, but you'll have fun finding those that do.

EMULATING GAME BOY ADVANCE



01 GameCube's pocket pal

While the Wii is well supported, you'd struggle to find GameCube homebrew titles. Released in the same year as the GameCube, the Game Boy Advance was its more successful, pocket-sized sibling, and has one of the most vibrant dev communities around. We're going to install the mGBA emulator as a Flatpak package, so we'll start by installing the package manager and repository. The Flathub app store (**flathub.org**) only supports 64-bit operating systems and processors, tagged aarch64 in its index. Open a terminal and type:

```
$ sudo apt install flatpak
$ flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
```

Now reboot Raspberry Pi to ensure that Flatpak's desktop integrations can set themselves up correctly. With that done, you'll be able to search for software from the command line or at **flathub.org**, although Flatpak installation from the browser isn't currently supported by the Raspberry Pi OS software centre.

▼ With a Game Boy Advance emulator like mGBA, you can buy and run new digital releases like the excellent adventure platformer *Goodboy GalaxyChannel*



02 GBA: Online

Open a terminal window and type: `flatpak install flathub io.mgba.mGBA`. Choose "yes" when prompted to confirm the package and its dependencies, and it'll be added to your Games menu. If you open it from there, though, you'll get an error message telling you that it's failed back to software rendering mode and Wayland won't correctly render the window, although the emulator will be usable. This is likely to be resolved when Mesa 23.3 comes to Raspberry Pi.

Until that's fixed, we'll run mGBA with a terminal command:

```
WAYLAND_DISPLAY="" flatpak run io.mgba.mGBA
```

We're still on software rendering, but now the mGBA desktop application can be moved and resized.

Beyond these teething problems, mGBA works out of the box. Just load a ROM in .gba format from the File menu. You can play in fullscreen mode by pressing **CTRL+F**.

03 Get some GBA games

There's been a flurry of recent GBA releases, largely distributed on the itch.io digital store. We've been playing the fantastic *Goodboy Galaxy* (**magpi.cc/goodboygalaxy**), and you can download the prologue/demo for free at **magpi.cc/goodboygalaxydemo**. Help a cat rescue their family in free platformer *Feline* (**magpi.cc/feline**), tower-defend your oozy kingdom in *Slimelord Siege* (**magpi.cc/slimelordseige**) or go dungeon delving in *Inheritors of the Oubliette* (**magpi.cc/oubliette**).

You can check out over 100 more GBA games in our Itch collection at **magpi.cc/itchgba**.



EMULATING DREAMCAST

01 Dream on

We'll install the Flycast emulator to play Dreamcast games. Flycast is also available as a flatpak, but at the time of writing this wasn't working reliably on Raspberry Pi OS. This is likely to be fixed by the time you read this. Open a terminal and type:

```
$ git clone https://github.com/flyinghead/flycast.git
$ cd flycast
$ git submodule update --init --recursive
$ mkdir build && cd build
$ cmake ..
$ make -j4
$ sudo ln -s ~/Software/flycast/build/flycast /usr/local/bin
$ flycast
```

02 Configure Flycast

Click the Settings button at the top right of the Flycast window and, in the General tab, click the add button in the Content Location bar. This should direct you to the directory where you'll be keeping your Dreamcast games. We've opted to put them in `~/Games/Dreamcast`. Put each game in its own directory – Flycast will check all subdirectories automatically. Now click on the Advanced tab, and under Other, tick the HLE BIOS box. You can optionally keep each game's save files on its own virtual VMU (Visual Memory Unit, also known as a Visual Memory System) by ticking the per-game VMU A1 box at the bottom of the Controls tab. Now click Done.

03 Get some Dreamcast homebrew

Now we'll need some games. As with many classic consoles, hobbyists and indie developers are still making both free and commercial software for the Dreamcast. Crazy Viking Studios granted Marc Hall permission for their 2013 platformer *Volgarr the Viking* to be ported to the Dreamcast and distributed for free. Download a copy from **volgarr.rkd.zone** and unzip the file into a subdirectory of your Dreamcast folder. If you've got Flycast open, the game should immediately appear as a playable disc image.

For a feel of what DC homebrew was like back in the day, download the Inducer collection from **magpi.cc/inducer** and extract it into your Dreamcast games directory, but be aware that the games in this collection don't always emulate smoothly. You'll find more Dreamcast homebrew on the Dreamcast Talk forum's new releases pages (**magpi.cc/dreamcasttalk**) and in our Itch collection **magpi.cc/itchdreamcast**). The Dreamcast Junkyard (**thedreamcastjunkyard.co.uk**) regularly covers new releases.



▼ Arcade platformer *Volgarr the Viking* was ported to Dreamcast in 2015 with the blessing of the original developers

WHAT'S A VMU?

Visual Memory Unit (VMU) devices were the Dreamcast's external flash memory card modules, which plugged into the controllers, allowing players to take their saved data around if they wanted to visit friends. They had mono LCD screens, integrated file managers and the ability to double as a handheld console. A number of Dreamcast games allowed you to load extra mini-games onto the VMU. Flycast doesn't allow you to play these directly, but it can load the Dream Explorer (**magpi.cc/dreamexplorer**) homebrew, which allows you to manage your VMUs.



EMULATING PLAYSTATION

Top Tip

PlayStation 2

PlayStation 2 games are playable using the Play! emulator (**purei.org**) but performance on Raspberry Pi 5 is limited for many titles.



▼ Recalbox automatically configures its PlayStation emulator for your controller and gives you a nice on-screen bezel, shown here around free action RPG Magic Castle

01 PlayStation 1 emulation

PlayStation emulation is mature, but you get different features from different emulators. If you primarily want to play homebrew, then Mednaffe – a front end for the Mednafen multi-emulator – is an excellent choice, as it upscales well. However, it doesn't include an emulated BIOS and can't play games from disc.

There are others, including PCSXR, bus as we're interested in new games, we'll stick with Mednaffe here.

Open a terminal and type:

```
$ sudo apt install mednaffe
```

02 Getting an official PS1 BIOS

You can currently get a PS1 BIOS from Sony, as it's included in a PS3 BIOS that's available. However, you'll need to install the RPCS (**rpcs3.net/download**) emulator to open it, and that's most effectively done on an x86 PC.

Go to **magpi.cc/pssystemsoftware** then click the + sign next to Reinstall using a computer, then click the Download PS3 Update button.

Install and run RPCS. Select Install Firmware from its File menu and select the PS3UPDAT.PUP file you downloaded and allow it to install.

With this done, assuming you're running RPCS3 on Linux, you'll find a file called `ps1_rom.bin in ~/.config/rpcs3/dev_flash/ps1emu/`

More detailed instructions are available at **magpi.cc/ps1bios**. To work with Mednafen, you'll have to rename the file. ps1_rom.bin is a region-free universal ROM, so can be renamed to reflect any of the three PlayStation region. Just copy it three times, and name the copies scph5500.bin, scph5501.bin and scph5502.bin.

03 PS1 Homebrew

The original Net Yaroze development kit is still the gold standard for homebrew making on PlayStation, and you'll find plenty of games and demos to download at **magpi.cc/netyaroze**, as well as a few at PSX Palace (**magpi.cc/psxpalace**). Download Kaiga's action-RPG *Magic Castle* from magpi.cc/magiccastle – we'll use this in the next step, so unzip it in a suitable directory – ours is in `~/Games/PS1/Magic_Castle.`

04 Configuring Mednaffe for PS1

First, you'll need to copy your renamed BIOS files over to Raspberry Pi and put it into `~/.mednafen/firmware/` – you may need to run Mednaffe once to create this directory. Start Mednaffe from the menu or a terminal window.

Before we play, we'll need to configure a few things:

Select the Global Setting tab, then Sound, and set the driver to sdl. Go to Systems, select the Input, choose Sony PlayStation and then select the Input tab from the pane on the right. Select each original PlayStation controller button and map a keyboard or controller key to it.

Finally, open the File menu > Open, browse to Magic_Castle_2021_07_May.cue and open it. Note that Mednaffe uses bin/cue file pairs rather than iso images. Press **ALT**+**ENTER** to fullscreen your game, and enjoy *Magic Castle*.

EASY PLAYSTATION EMULATION WITH RECALBOX

While running emulators from your desktop gives you more control over your downloads and settings, there's nothing like a dedicated emulation distro for convenience. Recalbox was first off the mark with dedicated support for Raspberry Pi 5. At the time of writing, Recalbox 9.2-Experimental for Raspberry Pi 5 wasn't yet available for direct installation from within Raspberry Pi Imager.

Download the image file manually from **magpi.cc/recallboxrpi5**, open Raspberry Pi Imager, and from "Choose OS", scroll down and select "Use custom". Select the image file you just downloaded and the Storage menu to select your microSD card, click "write", have a cup of tea, then pop the microSD card into Raspberry Pi 5. Recalbox supports external storage, but if you want a self-contained unit, use a large microSD card (over 64GB), as CD games can take up a fair bit of space.

> **"You get different features from different emulators"**

01 Essential files

Press the **START** button on your controller, go to Network Settings, enable Wi-Fi, and enter your Wi-Fi password. When connected to a network, Recalbox presents SMB shares and a web interface (accessible by default via a browser at **http://recalbox.local** if you're on Linux or **http://recalbox/** if you're on Windows). The SMB shares allow you to easily upload games and BIOS images, while the web interface gives you easy remote access to Recalbox's settings and at-a-glance views of installed games and BIOSes. The easiest way to add new BIOS images is to access **smb://recalbox.local/bios/** from another computer and copy files over. Recalbox ships with a number of freeware and emulated BIOS files, and all of the legally available BIOS files for various systems that we've discussed here and in previous tutorials such as our Emulate Everything guide in issue 133 (**magpi.cc/133**) should also be compatible, making this an excellent all-in-one emulation solution.

02 Add your games collection

Recalbox ships with a variety of open-source and freeware games, and you'll find a wide range of additional platforms ready to go. After copying your game and BIOS ROMs over, you'll want to reboot to force Recalbox to re-scan everything. You can always exit a game by pressing the controller hotkey and **START** buttons.

03 Add more storage

Connect an external drive and boot Recalbox, press **START** and go to System Settings. Go down to the Storage Device entry and select your USB stick. Recalbox will reboot and create a `/recalbox` directory tree on the device for your files.



You can configure Recalbox to use a USB drive as its main hard disk for game and BIOS storage

Top Tip

Music, maestro

If you're using a Sony DualShock 4 controller, it can end up being the default audio device. In Recalbox, fix that in Settings > Sound Settings > Output Device.

BUILD A UNIVERSAL MEDIA PLAYER

Turn your Raspberry Pi into a dedicated home entertainment system

By **Phil King**



As well as being a versatile general-purpose computer, Raspberry Pi can act as a low-cost, customisable media player system for all your home entertainment needs. Watch movies, TV shows, and videos, listen to music, and more. You can even add a Raspberry Pi TV HAT for live TV viewing and PVR functionality.

We'll explore the software and operating system options for your Raspberry Pi media player, along with some of the best cases and remote controls available. There's also a step-by-step guide to installing and using LibreELEC.

CASES & REMOTE CONTROLS

Protect your Raspberry Pi media player with a case and control it with a remote



ARGON ONE M.2 £40 / $42

It's one of the most popular media player cases for good reason. As well as providing a cool-looking enclosure for your Raspberry Pi 4, the Argon One M.2 has room for a slimline M.2 SATA SSD (available separately) in the bottom section. With UASP support, this enables super-fast data transfer speeds. The only slight downside is that a U-shaped USB adapter to bridge the upper and lower case sections uses up one of Raspberry Pi's USB 3.0 ports.

Still, the sleek all-in-one case re-routes and converts the micro-HDMI ports to full-size ones on its rear. It also has a built-in IR receiver to use with the optional Argon One remote.

As well as passive cooling, the case has a built-in fan with controllable speed via a script, enabling overclocking without throttling for maximum performance.

Check out the full Argon One M.2 review in *The MagPi* magazine issue #100 (**magpi.cc/100**).

MAGPI.CC/ARGONM2



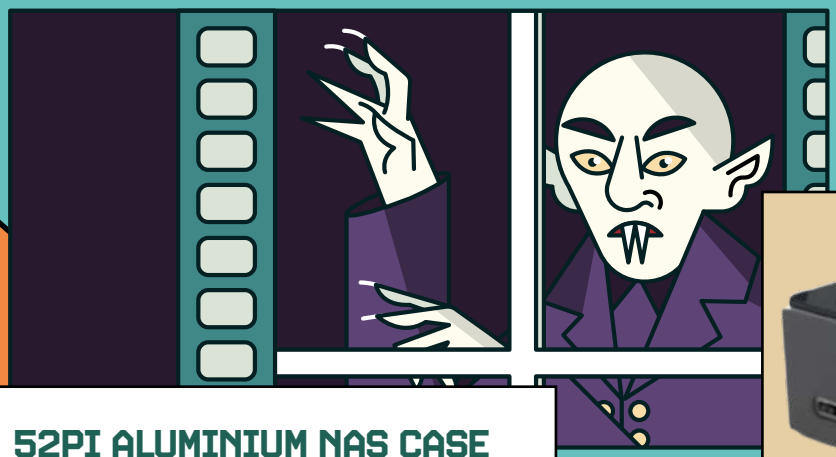
DESKPI PRO V2 £56 / $58

If you prefer your media player to be housed in rectangular 'TV box' style to match the rest of your AV setup, this could well be the case for you.

Featuring an NUC-style aluminium alloy body for improved protection, with acrylic side panels to help the Wi-Fi signal, the DeskPi Pro V2 is a versatile enclosure. Internal adapter boards re-route Raspberry Pi's ports to the case exterior, including two full-size HDMI sockets and a GPIO header on the rear. Even better, they enable you to mount either a 2.5″ or M.2 SATA SSD for high-speed storage.

Safe shutdown and reset functions are also available, while a built-in ICE cooler with PWM controllable fan prevents overheating. If you don't need the SSD, a Lite version of the case is also available (**magpi.cc/deskpilite**).

MAGPI.CC/DESKPIPRO



52PI ALUMINIUM NAS CASE

£36 / $38

52Pi's Aluminium NAS Case comes in two versions: one is equipped with a standard cooling fan and heatsink, while the other features a desktop PC-style ICE tower cooler system with a central copper heat-pipe and seven-blade LED fan.

Both versions accommodate an M.2 SATA SSD (but not NVME format) in the bottom section for ultra-fast storage. All the ports are re-routed to the exterior for easy access, along with a TF card slot.

MAGPI.CC/52PINASCASE



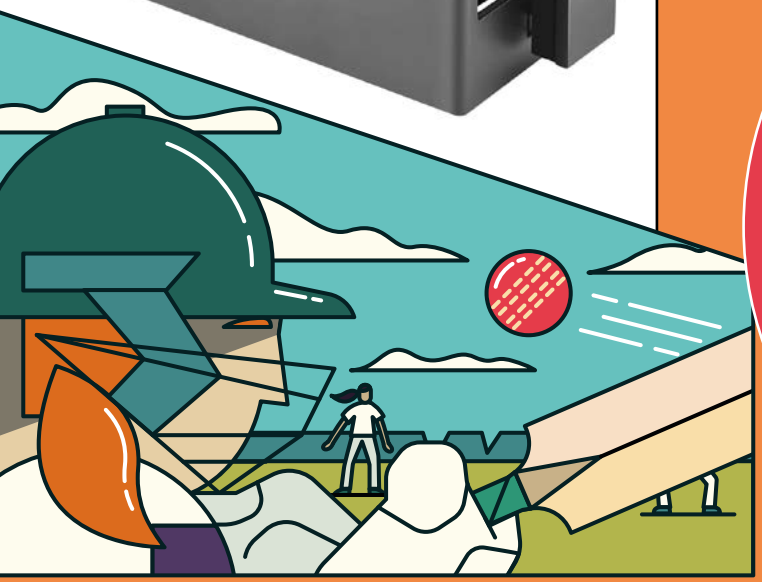


KODI EDITION FLIRC CASE

£13 / $14

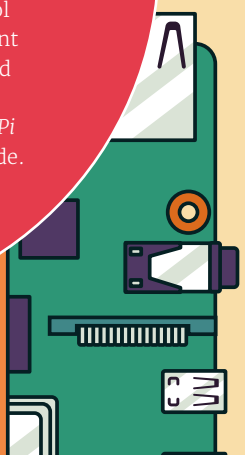
If you don't need the case to incorporate external storage, there are countless standard options available. This one comprises an aluminium heatsink body for passive cooling and features a cool-looking Kodi design lid.

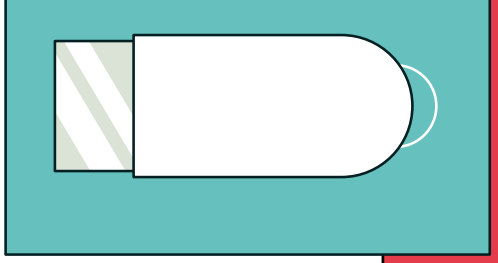
MAGPI.CC/KODIFLIRC



TV HAT

If you add a Raspberry Pi TV HAT to your media player setup, you can turn it into a fully functional PVR to watch and record digital TV programs (DVB-T2 format, Freeview in the UK), complete with an electronic programme guide (EPG). By installing the Tvheadend tool (also available in LibreELEC's PVR client add-ons), you can even stream live and recorded TV to other devices on your network via a web server. See *The MagPi* issue 75 (**magpi.cc/75**) for a setup guide.





OSMC REMOTE

£13 / $14

The best thing about this remote is that it comes with a receiver USB dongle. Just plug the latter into Raspberry Pi and you're all set up.



While it's designed specifically for the OSMC OS, the remote should still work out of the box with other systems – we tried it on LibreELEC and it worked really well. There aren't too many buttons, but the functions are generic enough for most media player use cases.

Another bonus is that it uses RF (radio frequency) rather than infrared, so you don't need to point it at the receiver for it to work – if only all TV remotes were like this!

MAGPI.CC/OSMCRF

FLIRC USB V2

£19 / $20

This clever USB dongle enables you to use any standard IR remote with your Raspberry Pi media player. A simple GUI application (available for Linux, Windows, and macOS) lets you easily pair media player functions with buttons on the remote you want to use. Once it's set up, you can just plug it into Raspberry Pi and use it.

You can even map remote buttons to keyboard keys and macros to run several commands. So it's ultra-versatile and hackable using a command-line application and developer SDK. FLIRC has also launched a Skip 1s universal remote that works well with it and other media devices.



MAGPI.CC/FLIRCUSB

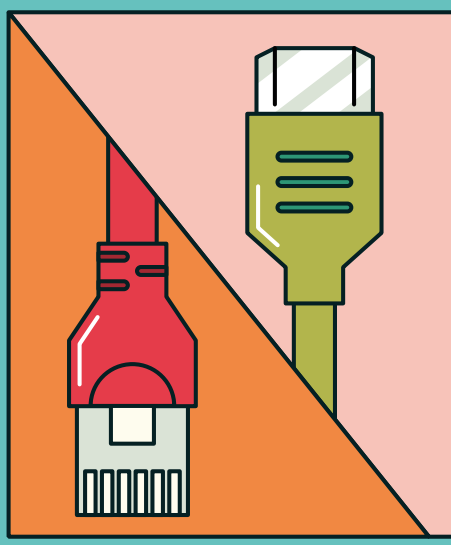
KORE OFFICIAL REMOTE FOR KODI

FREE

You don't even need a physical remote to control Kodi. Available for Android and iOS, this free app gives you convenient control of your media player from your phone. When launched, it searches the local network for media centres; just click on a detected one to connect (entering your username and password) and you're ready to start controlling it. It even syncs your media library to the app interface for easy browsing.

MAGPI.CC/KODIREMOTE





MEDIA PLAYER SOFTWARE

You have a choice of several operating systems for your media player

LIBREELEC

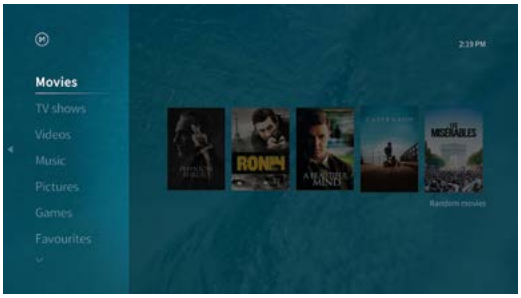
Billed as 'just enough OS for Kodi', LibreELEC is a custom lightweight Linux operating system that can also run on Raspberry Pi 2, 3, 4, 5, and 400. The OS image is about half the size of OSMC. It's available to install to a microSD card in Raspberry Pi Imager, although you may want to check out the official LibreELEC downloads page for the very latest version.

Upon first boot, a helpful setup wizard enables you to connect to your Wi-Fi network and enable SSH remote access and Samba sharing. You're then placed in the standard Kodi 'Estuary' user interface – which you can change to a different skin if you prefer. Alongside the usual Kodi add-ons, there are dedicated LibreELEC ones including various decoders, tools, PVR clients, and retro game emulators. There's also a handy LibreELEC settings menu for managing updates, connections, and Bluetooth devices.

Note that LibreELEC is a minimalist OS, so if you want to run other applications on your Raspberry Pi you'd be better off installing Kodi on top of Raspberry Pi OS.

LIBREELEC.TV





OSMC

OSMC (Open Source Media Center) is a lightweight version of Linux Debian for running Kodi. You'll find it in Raspberry Pi Imager, with a choice of versions available for different models; an older version will run on Raspberry Pi 1 and Zero boards. In general, OSMC tends to implement Kodi updates a little later than LibreELEC.

Upon booting up, you're presented with some setup menus (less user-friendly than LibreELEC's wizard) for choosing language, time zone, hostname. You can also enable SSH, which is more useful here as OSMC is a Debian OS, and choose from two user interface skins: OSMC or the default Kodi Estuary UI. For further personalisation, a 'My OSMC' option lets you tweak various settings and configure a remote.

OSMC.TV

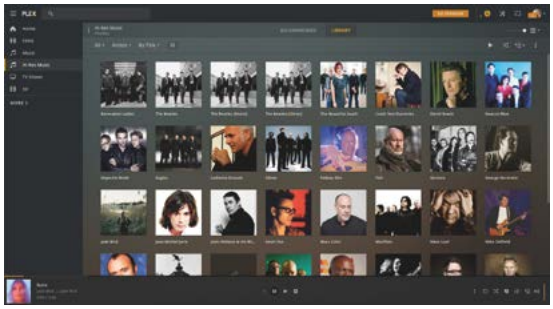


XBIAN

As the name suggests, this is a Debian-based lightweight OS for running Kodi. XBian will work on Raspberry Pi 2, 3, 4, 5, and 400 models. Not available in Raspberry Pi Imager's menus, it must be downloaded and installed manually, but there are weekly rolling releases to keep the OS bang up to date – you can install updates from within XBian.

In general, XBian offers a similar Kodi media player experience to LibreELEC and OSMC, with a choice of skins and settings. Features include a VNC server and a bunch of extra packages and tools.

XBIAN.ORG



PLEX

Unlike the other options here, Plex is not based on Kodi. Instead, it acts as a media server for streaming content to other devices rather than playing it locally. While this is possible in Kodi by setting up a UPnP server, Plex makes it easier and supports a wide range of formats/codecs with built-in transcoding.

Note that Plex is not an OS, but an application that you can install manually in Raspberry Pi OS, which takes a few steps. Once launched, a setup wizard guides you through adding your media libraries and choosing settings, including the ability to access the Plex server from outside your local network – via the Plex website. There's also the option to pay for a Plex Pass for extra online content if you want.

PLEX.TV



RASPBERRY PI OS (KODI)

If you want to use your Raspberry Pi to run other applications alongside your media player, your best option is to install Kodi manually in Raspberry Pi OS. Just make sure the system is up to date and use the command `sudo apt install kodi`. If you want the very latest Kodi version, however, you may need to build it from source.

Since RetroPie is based on Raspberry Pi OS, you can also install Kodi within that, using the package manager (it's in Optional Packages).

MAGPI.CC/RPIOSKODI

SET UP A LIBREELEC MEDIA PLAYER

Install the LibreELEC OS on Raspberry Pi and start using it

You'll Need

- Raspberry Pi 5 (or 2, 3, 4, or 400)
- 4GB+ microSD card
- Raspberry Pi Imager **magpi.cc/imager**
- USB external drive
- Remote (optional)

Top Tip

Samba sharing

For SMB/Samba sharing to work, you may need to add a username and password in the settings for it, and also add/enable that user on the NAS device to access the shared folder.

Based around Kodi, the lightweight LibreELEC OS works out of the box on Raspberry Pi. Let's take a look at how to install it, then set it up to play media stored locally on a connected USB drive, and also on NAS over the network.

01 Install LibreELEC

The easiest way to write LibreELEC to a microSD card is to use the Raspberry Pi Imager tool (**magpi.cc/imager**). Select 'Choose OS', then 'Media player OS' > LibreELEC, and select the version of LibreELEC for your Raspberry Pi model: RPi4/5 or RPi2/3. Then write it to the attached microSD card as usual.

02 Launch LibreELEC

Insert the microSD card into Raspberry Pi. Connect the latter to an HDMI TV/monitor, keyboard, and mouse. Then turn on the power. You'll see the LibreELEC logo, then, after a little while, the Kodi logo. Finally, the LibreELEC user interface will appear and you're greeted by a helpful setup wizard.

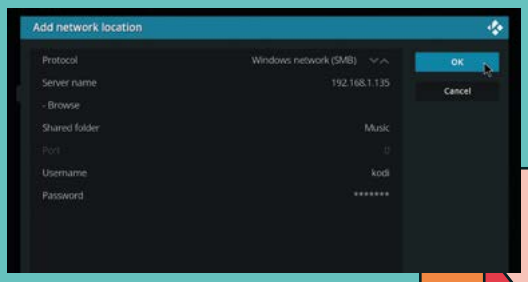
This enables you to choose the language, hostname ('LibreELEC' by default), then connect to your wireless router by choosing its SSID and entering its password. Note that unlike Raspberry Pi OS, you can't preload the Wi-Fi details in Imager's advanced options, as they aren't available for LibreELEC.

Next, you have the option of enabling SSH for remote access via the terminal – do so and you'll be asked to set a new password for it for security reasons (the default it 'libreelec' with the username 'root'). Finally, click 'Next' to exit the wizard.

03 Navigating the interface

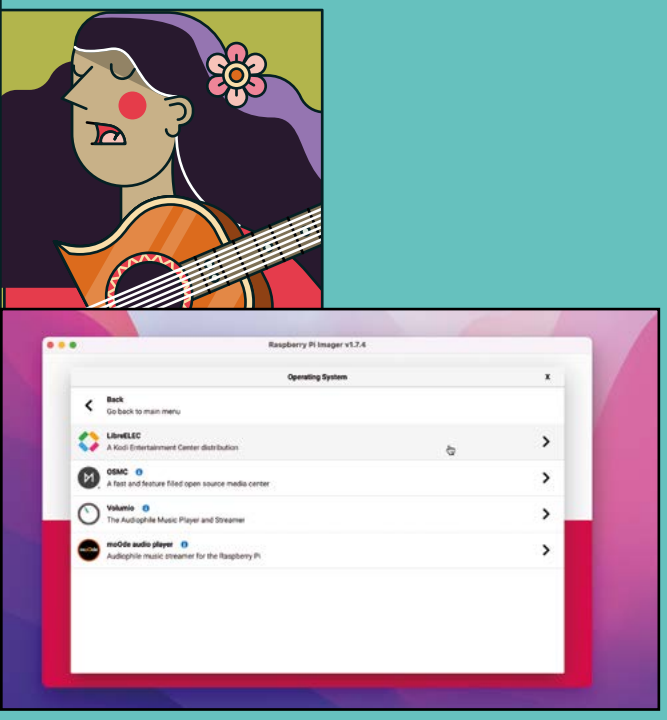
You'll now be able to navigate the LibreELEC interface, which is very user-friendly. Just move up and down the menu items in the left sidebar and select one of the related items shown on the right.

If you have a remote control such as the OSMC Remote, which works out of the box with its USB dongle inserted, you can use it to navigate instead of the arrow keys etc. An on-screen keyboard enables you to type text when needed.

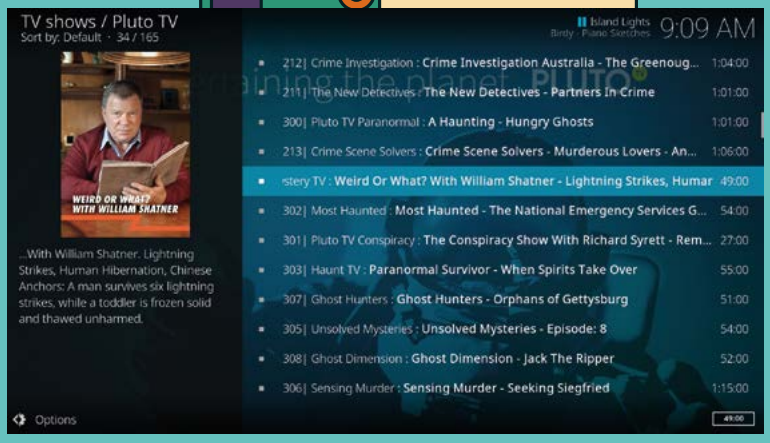


▲ As well as local drives, you can connect to network-attached storage to access its content





▲ You can use the Raspberry Pi Imager tool to write LibreELEC to a microSD card



▲ One of many available video add-ons, Pluto TV offers a selection of free shows to stream

04 Accessing an external drive

LibreELEC should detect a USB-connected external storage device automatically. To browse the content on it, select the relevant media

> "You can also connect to network-attached storage"

category in the left panel (e.g. Movies or Music), then 'Files' and you should see the name of the drive. Click on it to browse its contents. You can click on a media file to play it.

To add a file to the LibreELEC library, right-click it (or the folder it's in) and select 'Scan item to library'. LibreELEC will then scrape online information for it, including cover art if available. Scanned items will then appear in the right-hand panel in the user interface for the relevant category and can be selected directly. You can right-click one to view info and update it.

05 Accessing network storage

You can also connect to network-attached storage. For a media category, select 'Add videos (or music)', then Browse, select the sharing protocol (e.g. SMB), enter the IP address of the NAS, then the shared folder name (e.g. 'Movies'), then OK.

You should then be able to browse the NAS's contents as with a local drive, and add them to your LibreELEC library.

06 LibreELEC add-ons

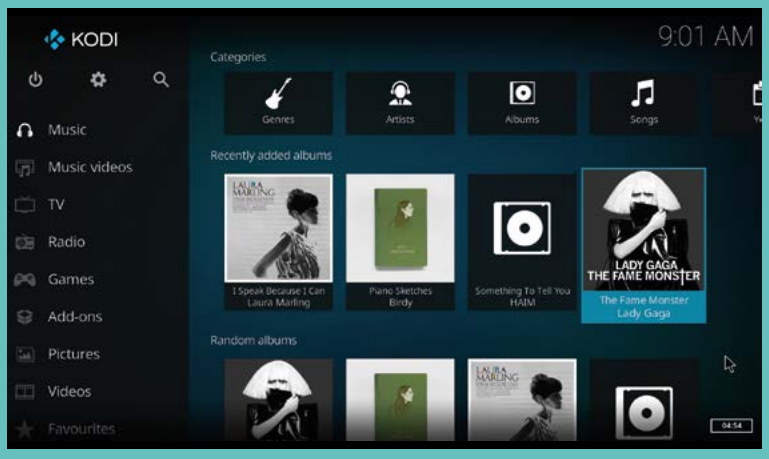
Click on the Add-ons option in the left menu and you can choose from a host of add-on apps sorted into categories. For instance, the Pluto TV add-on lets you watch a wide range of TV shows and movies for free. Or you could add a weather app to get the latest forecast for your location.

Getting the YouTube add-on working (and other apps that rely on it) is a bit of a faff, as you now need an API key to use it in LibreELEC. This involves setting up a Google Cloud developer project, adding your Google account as a test user, and then copying the API key, client ID, and client secret to the YouTube add-on's API settings. You then need to sign in to your Google/YouTube account (by entering a code on another device) twice in LibreELEC. But once you're in, it works a treat.

Top Tip

Kodi wiki

If you encounter an issue or aren't sure how to do something, the extensive Kodi wiki should have the answer: **kodi.wiki**



▼ Once your content is added to the LibreELEC library, it'll show up in the right-hand panel

Arts & Crafts

With Raspberry Pi

Technology and creativity are inclusive and Raspberry Pi brings the two together.

By David

Many Raspberry Pi projects offer solutions to practical problems. Many builds also perform specific tasks that are either informative or seek to help make life a little easier. Equally, however, there are loads of makers unleashing their creativity and putting Raspberry Pi at the heart of imaginative endeavours: simply because they can.

These makers are producing appealing projects that stop people in their tracks because they either look good or perform tasks that are just sheer, plain fun. These are our kind of makers.

In this feature, we take a look at arts and crafts projects that inspire us. In some cases, they're straightforward but they will hopefully unlock the inventive part of your brain so that you come up with more ingenious ways of taking projects further. In other cases, a project may be difficult to replicate like-for-like, but that doesn't mean you can't be flexible and pursue an alternative path. You may even decide to mash up two or more projects.

Whatever you do: be bold and innovative. And remember, with Raspberry Pi powering your projects, you already have a head start. The possibilities are endless.

WHERE TO BUY KIT

PISHOP.US

▸ pishop.us

Light up your projects with NeoPixels, screens, and LEDs. You'll find loads on sale at PiShop so take your pick.



THE PI HUT

▸ thepihut.com

Into making wearables? Snap up conductive thread, woven conductive fabric and more from The Pi Hut.

PIMORONI

▸ pimoroni.com

Find great creative kits such as the Bearables Fox LED Badges and LED circuit stickers at Pimoroni





Crookes

Maker Motivation

Inspiring projects that will help you to make music, clothes, furniture, and a big impact on other makers!

DIFFICULTY GUIDE

BEGINNER
Projects for everyone

ADVANCED
Projects that require some experience

EXPERT
Projects that require skill

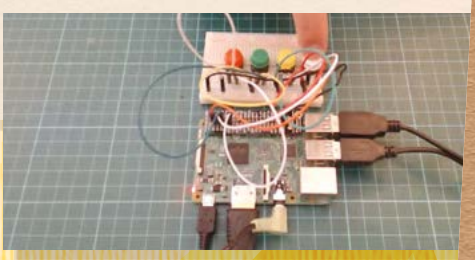
GPIO MUSIC BOX

▶ magpi.cc/gpiomusicbox

Don't be afraid to make a noise about your arts and crafts project. In fact: if you try your hand at making this GPIO music box, that's exactly what you'll be doing. Connect a set of buttons to a Raspberry Pi computer and you can begin to play sounds using Python. The more buttons you add, the more sounds you can enjoy, potentially turning the project into a fully-fledged instrument. Either that or a sound box that can make rude noises on demand. The choice is yours.

TOOLS NEEDED

- Raspberry Pi
- Breadboard
- Four tactile switches
- Five pin-to-socket jumper leads
- Four pin-to-pin jumper leads
- Speakers/ headphones



HONEYCOMB WALL SHELF



▶ magpi.cc/112

Fancy creating an artistic talking point for your home without going over the top? These honeycomb wall shelves may do the trick. Adopting a hexagonal design, the shelves were designed with a channel, through which 12V RBG light strips could be threaded. Once outputted to a Crealty Ender 5 Pro 3D printer, the shelves could then be connected up to a Raspberry Pi 4 computer from which they are controlled, allowing different levels of red, green or blue to be displayed.

TOOLS NEEDED

- Raspberry Pi 4
- 3D printer
- Four tactile switches
- RGB light strips
- Three IRFZ44N MOSFETs
- Three NPN transistors

RASPBERRY PI BERET

▶ magpi.cc/raspberrypiberet

If you want to get ahead in fashion then you need to get a hat, and this Prince-inspired headwear is sure to inspire. As well as allowing the ring of colourful LED lights to animate at random intervals, you can connect a phone and control the lights from a web browser thanks to Raspberry Pi Pico W's wireless LAN. Any hat would work for this project (and you could even decide to make your own). Just make sure there's enough room above your head for the several feet of wire required for the project.



TOOLS NEEDED

- Raspberry Pi Pico
- NeoPixel LEDs
- MicroPython

Art Advice!

You could turn Raspberry Pi into a live coding music studio using Sonic Pi (sonic-pi.net)

SCREEN DRESS

▶ magpi.cc/screendress

Let's be honest here, creating something that could be worn on the catwalk is certainly no cakewalk. But a Dutch FashionTech designer has shown what is possible if you let your imagination run wild. Her ScreenDress makes use of Hyperpixel 2.1 Round LCD screens costing around £50 each, and they've been designed to look like eyes which change according to the wearer's thoughts (gathered by an EEG sensor strapped to the head). They show how you can bring tech together in a creative way.

TOOLS NEEDED

- Raspberry Pi Zero 2 W
- Hyperpixel 2.1 Round LCD screens
- Unicorn Headband
- 3D printer



SPEECH-TO-TEXT BADGE

▶ magpi.cc/speechtotextbadge

So, you're at a party or a conference, and the person you're talking to can't hear anything you're saying. Rather than resort to pointing, you can make yourself a badge that picks up your words via a USB lapel mic, processes them on a Raspberry Pi Zero with the deep learning speech recognition system Deepgram, and transcribes your speech on a small screen.



TOOLS NEEDED

- Raspberry Pi Zero 2 W
- Touchscreen display
- Mini HDMI – HDMI Adapter
- Battery pack and cable
- USB Microphone

Visual arresting projects

These projects are sure to catch the eye

MATRIX ALBUM ART DISPLAY

▶ magpi.cc/artledmatrix

As much as we love streaming services, we can't help but think we've lost the art of the album cover. With this fab project, you can bring those covers back to life by displaying them on Adafruit's LED matrix. It's a simple enough project which ties in with the music you're playing so check out the maker's GitHub page (**magpi.cc/ftartwork**) to learn how to make it.



TOOLS NEEDED

- Raspberry Pi 3B+
- Adafruit 64x64 RGB LED Matrix
- Adafruit matrix bonnet
- Power supply
- 5v 10a switching PSU
- 3D printed matrix feet
- Pi mount

SUNRISE LAMP

▶ magpi.cc/sunriselamp

The Sunrise lamp marries technology with an origami technique called snapology. It requires a lot of patience but, if you have the time to learn how to use strips of paper to create shapes from rigid triangles, then you too could create a rather special light. A Raspberry Pi Zero W is connected to a set of NeoPixels and the lights have been programmed to dim at night before brightening in the morning to help ease sleepers from their slumber. There's lots of scope to create some amazing designs here.



TOOLS NEEDED

- Raspberry Pi Zero W
- Plastic case
- NeoPixels
- Paper



POV DISPLAY

▶ magpi.cc/povdisplay

This visually arresting project works by rotating a strip of LEDs at high speed. By syncing their blinking patterns (and working on the principle that humans continue to briefly see an image that has stopped entering the eye), it's possible to create the illusion of a moving image. The strips are connected to the Pico, mounted on a rotating arm and spun at high speed. A wireless charging module prevents the wires from becoming a tangled mess.

TOOLS NEEDED

- Raspberry Pi Pico
- Adafruit DotStar Digital LED Strip
- QTR-1A Reflectance Sensor
- DFRobot Wireless Charging Module 5V
- DC Motor RS-540SH
- Microsoft Visual Studio 2017



RUHACAM

▶ magpi.cc/ruhacam

Photographers don't need to spend a fortune on a cutting-edge brand-name camera, nor do they need to reach for the latest smartphones. With the High Quality Camera for Raspberry Pi, it's possible to produce your own device in any style you want – allowing you to flex your artistic muscles in the creation of a cool case before showcasing your flair for a good photo. This particular project opts for a retro look, but it comes complete with a modern viewfinder. As creator Penk Chen says, "it's literally technology married with liberal arts".

TOOLS NEEDED

- Raspberry Pi Zero W
- High Quality Camera
- 2.2-inch TFT display
- 2000 mAh lithium polymer battery
- Case

Art Advice!

Why not make yourself a light meter too? **magpi.cc/picolightmeter**



AI ART

▶ magpi.cc/creatingaiart

Thanks to artificial intelligence, you can enter a prompt, click generate, and see a picture appear before your eyes. If you fancy experimenting with the technology, try OnnxStream, a small inference library that runs Stable Diffusion 1.5 on a Raspberry Pi Zero 2 W. Sure, it can be slow, but it's super impressive. Check how it works and learn how to set it up at **github.com/vitoplantamura/OnnxStream**

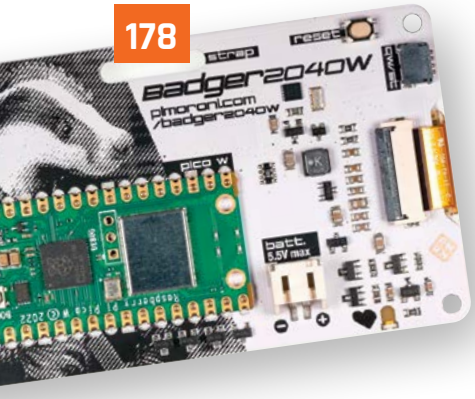


TOOLS NEEDED

- Raspberry Pi Zero 2 W
- OnnxStream

Reviews and Resources







176 Galactic Unicorn
A big screen for a tiny Pico

178 Badger 2040 W
An eInk badge for your next convention

179 Inky Impression 7.3”
A huge, multicoloured eInk display

180 PicoVision
Output video from your Pico

182 Vizy
A Raspberry Pi-powered AI camera

184 PiBoy XRS
The ultimate handheld gaming case kit

186 Argon NEO 5
The new NEO case for Raspberry Pi 5

188 Argon ONE V3 M.2 NVMe
Add superfast storage to Raspberry Pi

190 NVMe Base
Expand the PCIe port of Raspberry Pi 5

192 Puppy Pi
A state-of-the-art dogbot

194 CM4 XGO-Lite robot dog
A powerful robot on four legs

195 Inventor HAT Mini
A tiny, all-in-one electronics playground

196 10 amazing cases
Keep your Raspberry Pi safe

198 10 amazing starter kits
Everything you need to start making

200 10 amazing Pico projects
Inspiration for the smallest Raspberry Pi





DESIGNED FOR
Raspberry Pi 5
AMPLIFY YOUR IDEAS
Argon ONE V5
Argon neo 5 NVMe
Argon Poly+5
Argon FORTY
ARGON ONE V5
ARGON NEO 5 M.2 NVMe
ARGON POLY+ 5
sales@argon40.com
www.argon40.com

Galactic Unicorn (Pico W Aboard)

SPECS

DISPLAY:
53×11 matrix of RGB LEDs (583 in total)

FEATURES:
Pico W on board, 10 × push-buttons, mono I2S amp and 1W speaker, 2 × Qwiic/STEMMA ports, battery connector, 2 × metal legs

DIMENSIONS:
331 × 79 mm

▶ Pimoroni ▶ **magpi.cc/galactic** ▶ £50 / $50

Shining like a galaxy of stars, this Pico W-powered RGB LED display will light up your projects. **By Phil King**

With no fewer than 583 RGB LEDs in a 53×11 grid, this matrix display has an unusually elongated shape. However, the Galactic Unicorn is packed with features and is easy to program in MicroPython (or C++ if you prefer).

As its full name suggests, it's powered by a Pico W pre-soldered to the rear of the display. You get all the power of Pico W's RP2040 chip to drive the LEDs, along with Wi-Fi connectivity.

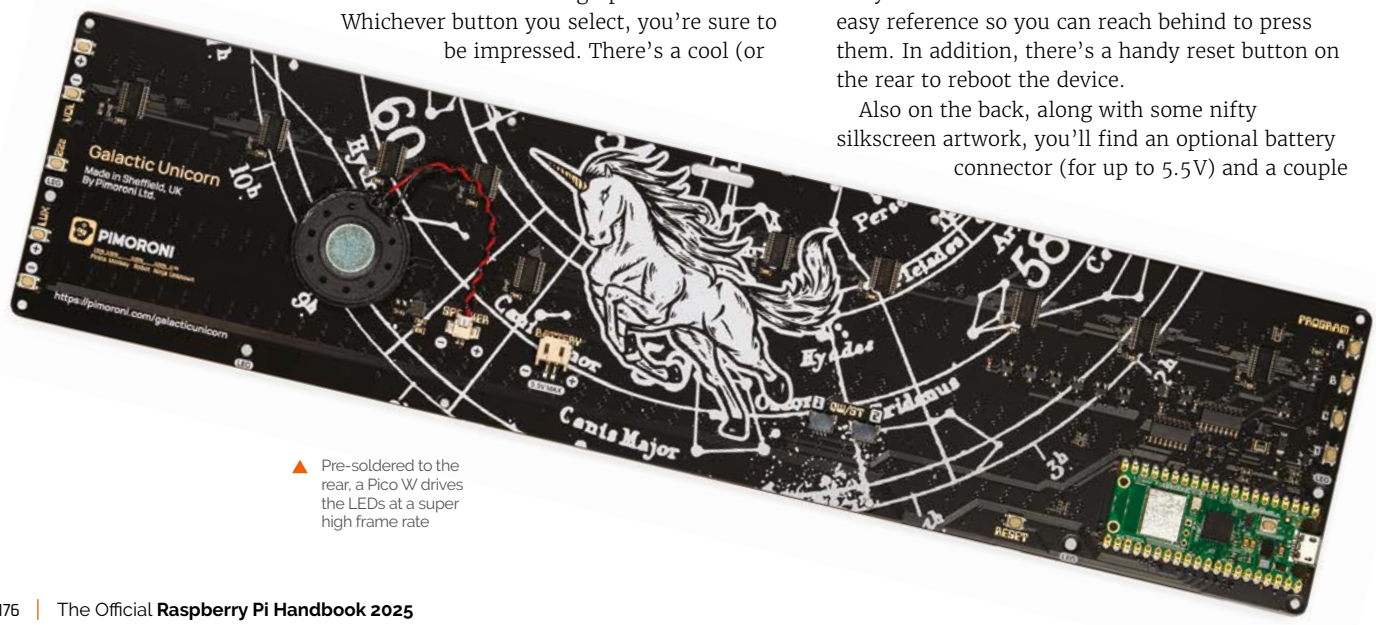
When you first power on the Galactic Unicorn, via the supplied short micro-USB cable, it immediately springs into life with Pimoroni's MicroPython firmware and code examples pre-installed, so there's no need to flash it. Text on the display invites you to press the A, B, C, or D button to launch a different graphical demo.

Whichever button you select, you're sure to be impressed. There's a cool (or rather warm) burning flames effect, an eighties supercomputer (random pixels), a cycling rainbow, and a nostalgia computer prompt. What you notice straight away is how these pixels are large and bright (dazzling at the maximum level) – and there's no sign of flicker, even if you film the display running. The Pico W RP2040's PIO state machines are used to drive the LEDs at around 300 fps at 14-bit resolution, while gamma correction is applied to image data.

Button controls

Those four tiny tactile buttons are situated on the rear, along with five others on the opposite side to sleep the display, adjust the brightness, and alter the volume from the mini 1W speaker on the rear. They're also labelled on the front of the board for easy reference so you can reach behind to press them. In addition, there's a handy reset button on the rear to reboot the device.

Also on the back, along with some nifty silkscreen artwork, you'll find an optional battery connector (for up to 5.5V) and a couple



▲ Pre-soldered to the rear, a Pico W drives the LEDs at a super high frame rate



The clock code example fetches the accurate time via Wi-Fi from an NTP server

of Qwiic/STEMMA ports (JST-SH) for connecting breakouts such as sensors. The Galactic Unicorn even has one sensor built-in: a phototransistor at the right-hand edge of the front panel to detect light levels – which could be useful for adjusting the display brightness automatically.

While the LEDs have some built-in diffusion, you may want to add your own diffuser panel, in which case the mounting holes should come in useful. Example DXF files are supplied on the product page. To stand the board vertically, at a slight angle, two metal legs are supplied and screw into any of four slightly larger holes in the board edge.

> You get all the power of Pico W's RP2040 chip to drive the LEDS

Pixel programming

Programming the Galactic Unicorn should prove relatively easy in MicroPython. Several further code examples are supplied on the GitHub repo. These include a fairly convincing, procedurally generated lava lamp and an easily adaptable scrolling text demo. Graphics and text are generated using the PicoGraphics library – used for several of Pimoroni's displays, it offers rich functionality including shapes, sprites, and a selection of fonts. Other functions are explained in the Galactic Unicorn reference guide: **magpi.cc/galacticref**.

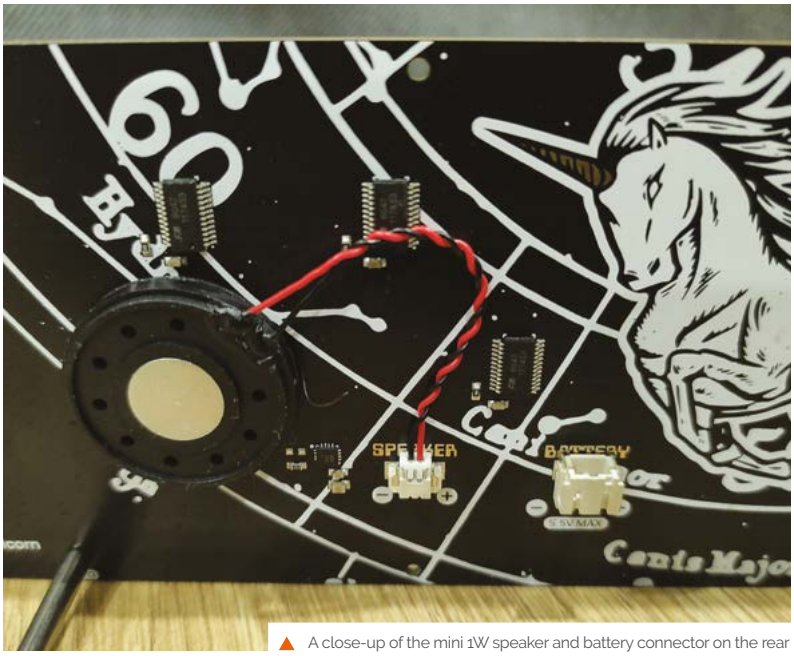
There are a few examples that make use of Pico W's wireless connectivity too. Just pop your Wi-Fi network details into the **WIFI_CONFIG.py** file to get connected. A digital clock demo fetches the live time from an NTP server; a Cheerlights history slowly builds up coloured pixels, one every five minutes; while a Galactic paint program enables you to draw on the display in real-time from another device via a web server hosted by Pico W – very cool indeed!

With a little creativity, there's the potential to use the Galactic Unicorn for a wide range of projects, including wireless ones such as a live weather dashboard or scrolling RSS feed.

Verdict

Its elongated shape may not suit all projects, but this is one dazzling display that makes uses of Pico W's power to create impressive animated effects at a super high frame rate.

9/10



A close-up of the mini 1W speaker and battery connector on the rear

Badger 2040 W (Pico W Aboard)

SPECS

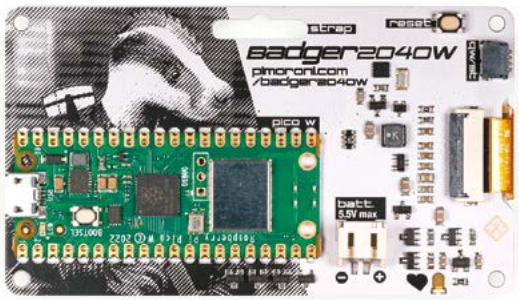
DISPLAY:
2.9-inch e-ink, 296×128, monochrome

FEATURES:
Pico W, 5 × user buttons, reset button, LEDs, Qwiic/STEMMA QT port

DIMENSIONS:
85.6 × 48.7 × 10 mm

▶ Pimoroni ▶ **magpi.cc/badger2040w** ▶ £27 / $28

Much more than a badge, this versatile mini e-ink display has wireless connectivity. By **Phil King**



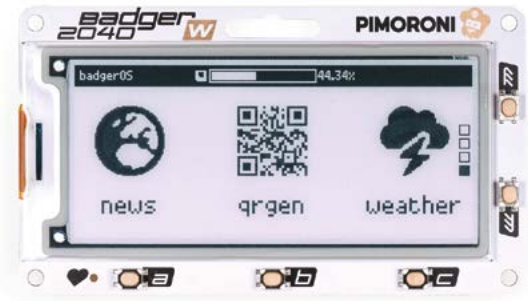
▼ On the rear you'll find Pico W, a reset button, battery connector, and Qwiic/STEMMA QT port – to which you could connect a sensor

The Badger 2040 W is a mini monochrome e-ink display with wireless connectivity. Unlike the original Badger 2040 (reviewed in issue 116, **magpi.cc/116**), this new model is based around a Pico W pre-soldered to the rear.

Another improvement is the inclusion of a real-time clock (RTC), enabling it to keep time even when in 'super deep sleep' mode, with Pico W powered down – in which it draws a minuscule 20 µA of current.

Talking of power, you will want a portable way of supplying it. To this end, there's a JST battery connector on the rear. If you buy the accessory kit version, you get – along with a lanyard – a 2×AA battery pack with a Velcro patch to stick it to the rear of the board.

Alternatively, you could use a standard USB power bank connected to Pico W. Both solutions are a little chunky, though, so you might prefer a slimline LiPo battery pack.



▲ The Badger 2040 W's mini monochrome e-ink display is surrounded by tactile buttons

> A real-time clock enables it to keep time even when in 'super deep sleep' mode

Easy to program

When first powered up, the Badger 2040 W launches into 'Badger OS' with a scrollable menu of icons to choose demo programs and tools. These include a badge, digital clock, e-book reader, interactive checklist, news headlines, and weather dashboard.

Naturally, you can connect Pico W to a computer via USB to customise the examples or create new programs – in MicroPython or C/C++. Pimoroni's standard PicoGraphics library makes it easy to add bitmap images, and several fonts are supported. You'll need to add your Wi-Fi credentials to try out examples such as news and weather, as well as setting the correct time via NTP.

Verdict

An interactive badge that doubles as a versatile mini e-ink display with a reasonably quick refresh rate and Wi-Fi connectivity.

8/10

Inky Impression 7.3″

SPECS

DISPLAY:
7.3-inch e-ink, 800×480 pixels, seven colours

FEATURES:
4 × user buttons, I2C and SPI breakout pins, booster header, stand-offs

DIMENSIONS:
170.2 × 111.2 mm

▶ Pimoroni ▶ **magpi.cc/impression7** ▶ £70 / $72

Ideal for showing artwork and data, this is a super-size e-ink display. By **Phil King**

The Inky Impression 7.3″ is a new super-size version of Pimoroni's full-colour e-ink/e-paper HAT for Raspberry Pi. We reviewed the original 5.7-inch model in issue 102 (**magpi.cc/102**); there's also a 4-inch version.

> The big draw here is the size of the screen, measuring 7.3 inches diagonally

The 7.3-inch display is considerably larger than a full-size Raspberry Pi board, so is supplied with an extra female header to boost its height, along with some metal stand-offs and screws to secure it. You need to take a little care when mounting it, holding it by the sides rather than pushing down on the glass screen.



▲ On the rear you'll find a breakout header and some cool silkscreen artwork

▼ The 800×480 widescreen display can render dithered colour images with a fair amount of detail



The rear of the Inky Impression board also features a set of breakout pins for power, ground, I2C, and SPI to connect extra components such as sensors – perhaps for a weather display.

Widescreen wonder

Naturally, the big draw here is the size of the screen, measuring 7.3 inches diagonally. A resolution of 800×480 equates to a pixel density of 127.8 ppi – just marginally lower than on the 5.7″ model. In practice, images look detailed and smooth enough from a 30 cm-plus viewing distance.

To display a custom image (JPG, PNG, etc.), you can use Pimoroni's handy dithering code to process it for the seven-colour e-paper. A one-line installer is used to download the Python library, fonts, and examples from the GitHub repo (**magpi.cc/inkygit**). Note that you need I2C enabled to detect the board.

The main downside, as with all colour e-ink displays, is a slow refresh rate: around 40 seconds, depending on the complexity of the image being drawn.

Verdict

A whopper of an e-ink screen with full colour and plenty of room to display images and/or data.

8/10

PicoVision (Pico W Aboard)

▶ Pimoroni ▶ **magpi.cc/picovision** ▶ £35 / $37

This pocket-sized digital video stick brings Pico games and more to the big screen. By **Phil King**

SPECS

FEATURES:
Pico W, separate RP2040 (GPU), 2 × 8MB PSRAM, PCM5100A DAC for audio, 3 × user buttons, status LED, reset button, microSD card slot

CONNECTIONS:
HDMI video out, 3.5mm audio line out, breakout header, Qw/ST port, micro USB (on Pico W)

DIMENSIONS:
87 × 38 × 9mm

Aimed mainly at fans of retro gaming and old-school programming, PicoVision is a surprisingly capable digital video stick with a pre-soldered Raspberry Pi Pico W on board.

Using Pico W's RP2040 microcontroller chip as the CPU, it adds a second RP2040 to act as a GPU to help handle the big-screen visuals. Two extra 8MB RAM chips are used to swap frame buffers between the two RP2040s, enabling PicoVision to output reasonably high-res graphics (for a microcontroller) to a monitor via a standard HDMI output.

The only slight drawback is that it's not able to output audio via HDMI. For this, there's a 3.5mm stereo line out jack to connect a pair of PC speakers or headphones.

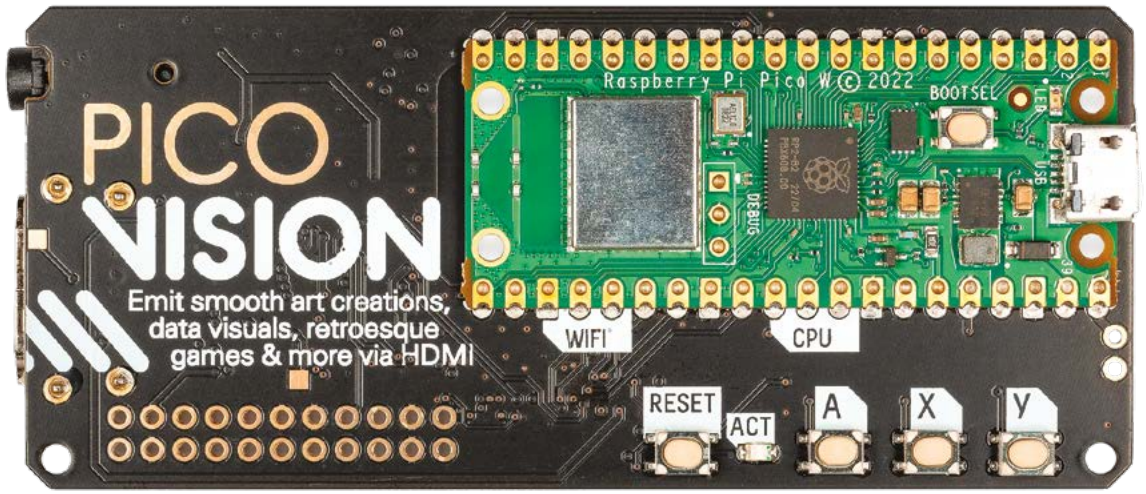
Power is connected using Pico W's micro USB port – as usual, you can hook this up to a computer for programming in MicroPython or C/C++, or flashing Pico W with new firmware.

Alternatively, PicoVision can be powered via two unpopulated pins. There's also a breakout header for CPU and GPU debugging, plus a few other GPIO pins. A single Qwiic/STEMMA QT port can be used to connect I2C breakouts or sensors.

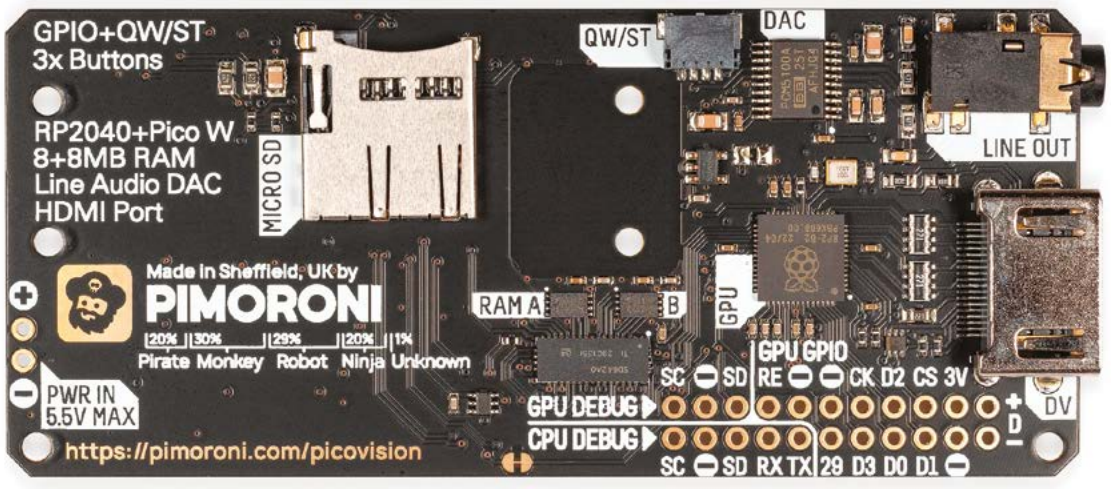
On the front of the board are three user buttons, an activity LED, and a handy reset button. At the rear is the bonus of a microSD card slot to provide extra storage for assets such as graphics.

Instant menu

Getting started is as easy as connecting PicoVision to an HDMI monitor or TV and powering it up. It comes pre-flashed with MicroPython firmware that auto-runs a menu of example programs. The



▶ The front of the PicoVision board features a pre-soldered Pico W and user buttons. The holes are an unpopulated breakout header



On the rear are a microSD card slot, Qw/ST connector, audio line out and HDMI, plus the extra RP2040 'GPU' chip

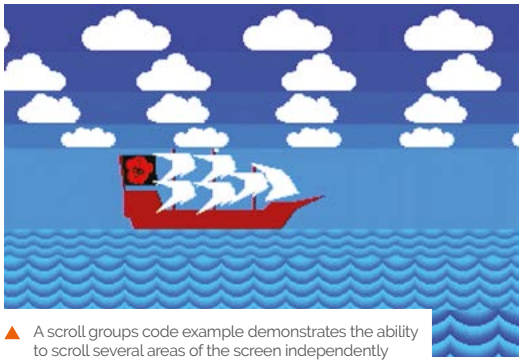
A and X user buttons are used to move up and down through the menu options; and a press of the Y button will select one.

Among the example programs are a fun *Flappy Bird*-style game, magic mirror dashboard, and Teletext-style headlines – the last two require connection to a Wi-Fi network.

> Among the example programs are a fun Flappy Bird-style game, magic mirror dashboard, and Teletext-style headlines

Several graphical demos showcase PicoVision's capabilities. These include a classic 3D starfield effect (remember Windows 3.1's screensaver?), a screen modes menu with numerous resolutions (some require GPU overclocking), and a parallax scrolling animation that makes use of PicoVision's scroll modes – up to seven groups of scanlines can be scrolled separately.

Up to 80 sprites can be shown on screen simultaneously (or 32 with the special widescreen version of the firmware), with up to ten per any single scanline.



A scroll groups code example demonstrates the ability to scroll several areas of the screen independently

Start coding

When it comes to programming PicoVision yourself, a comprehensive GitHub repo (**magpi.cc/picovisiongh**) provides lots of useful info, along with MicroPython and C/C++ libraries and code examples from which to learn. The most basic are Hello World programs to print text to the screen. Graphics are handled using Pimoroni's standard PicoGraphics module, while a PicoSynth sound module is used to play simple tunes and noises using I2S audio from the board's DAC chip.

You'll also want to try playing classic *Doom* on PicoVision – which works really well. Just flash the *Doom* firmware UF2 to PicoVision and add the game's WHX file to a microSD card. For control, you'll need to connect a USB keyboard via a splitter cable (as supplied in the optional accessory kit).



Yes, you can play *Doom* on PicoVision! Use a splitter cable to plug a USB keyboard into Pico W

Verdict

With a second RP2040 and extra RAM to boost Pico W's capabilities, PicoVision offers plenty of possibilities for retro gaming and coding projects.

9/10

Vizy

SPECS

COMPONENTS:
Raspberry Pi 4 (2, 4, or 8GB RAM), universal tripod mount, 32GB microSD, 25W AC power supply, Vizy case (900g; 10×15×10cm)

CAMERA:
12-megapixel Sony image sensor (IMX477), wide-angle, 3.25mm M12 lens (optional high-quality 8-50mm C/CS zoom lens), electronically switchable IR-cut filter

I/O:
Digital I/O, analogue I/O, PWM, UART serial; high-current output (up to 1000mA per channel), optional lighting accessory

▶ Pixy ▶ **vizycam.com** ▶ £280 / $259

This purpose-built AI camera combines Raspberry Pi with machine learning software. By **Lucy Hattersley**

Vizy is an interesting camera product that combines Raspberry Pi 4 with a Sony camera and lens. This is housed inside a custom plastic case with heatsinks, a fan, and a high-current I/O connection.

Once it's on your network, you can log in via a web browser pointing to **vizy.local**, and all the interaction takes place through this web-based interface. Vizy has a laudable aim of making it easy for students and makers to explore AI-powered machine learning projects.

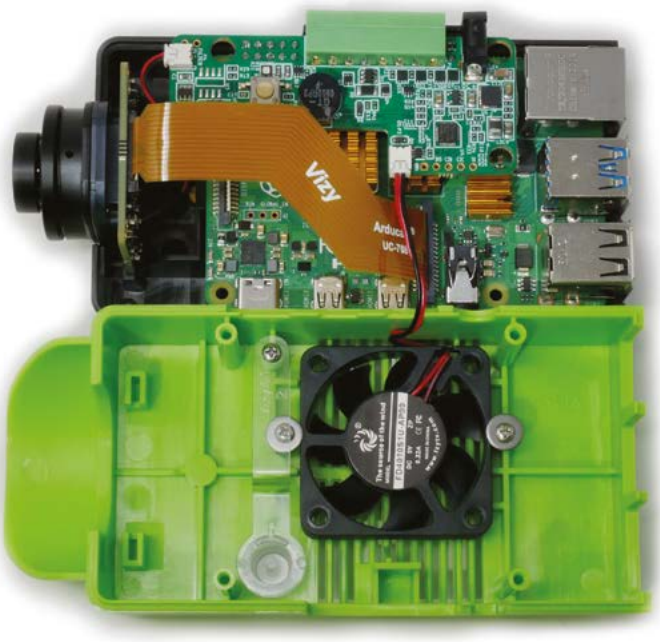
Vizy setup

Vizy comes self-assembled (you can choose between 2GB, 4GB, and 8GB Raspberry Pi inside). A microSD card is pre-installed and all you need to do is plug it in and press the button to turn it on.

While there's no fun in the build process, it is quick to get up and running. On first run, it uses Raspberry Pi's Wi-Fi module to broadcast a network, which you can join from another computer. You can also connect via an Ethernet cable, which was the approach we favoured.

Once connected, you set up Vizy to automatically connect to a local wireless network so it connects whenever you boot up. Then, you can investigate the built-in software, divided into a range of apps and examples.

Built-in apps include a Bird Feeder (identifies birds), MotionScope (captures the motion of objects), and Object Detector (detects and logs classes with text alerts sent via Signal).



▲ Inside Vizy is a separate daughterboard offering I/O connections and a fan to keep the unit cool

The examples are more simple, including OpenCV Edge Detection, Pet Companion, Pic Taker, TensorFlow Lite, and simple video capture. You can investigate the Python code for each example, to help you start to build your own apps. Detailed documentation at **docs.vizycam.com** includes a 'Getting started' guide, and information about accessories and applications, plus some simple API information. You can access the Python editor directly from the web interface, or log into the Shell and control Raspberry Pi directly.

These built-in programs ensure Vizy is a great platform for exploring machine learning possibilities. We particularly liked MotionScope, which captures the movement of an object (such as a



▲ Vizy houses a Raspberry Pi 4 and camera inside a shell with everything you need to get started with machine learning image projects

bouncing ball) and provides detailed x, y, and z co-ordinates as graphs and a data dump. We can imagine this working incredibly well alongside calculus learning.

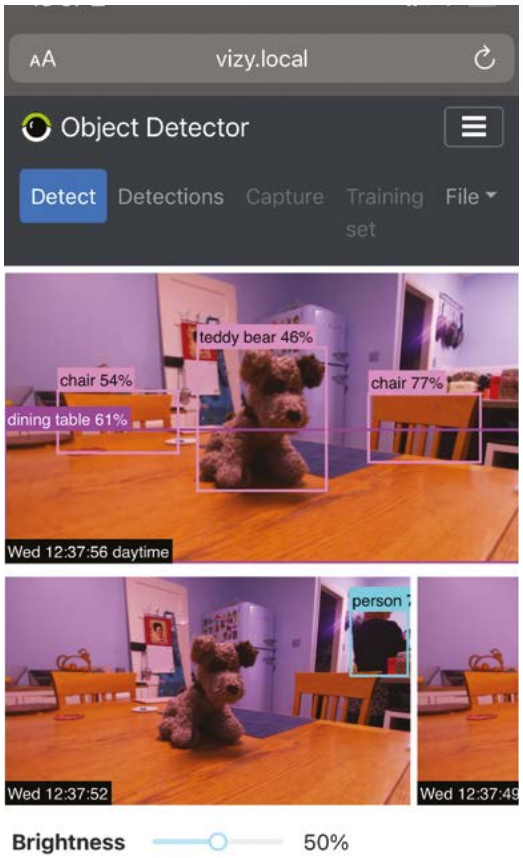
Meanwhile, the Object Detector and Bird Feeder can be used to analyse and respond to scenes,

> Vizy provides you with a great introduction to image recognition

creating alerts for specific items. We had a lot of fun tracking a pet cat and sending alarms when she was detected on a table.

The camera is the equivalent of a Raspberry Pi High Quality Camera, although we'd be happier if it used an official camera offering. We also found the 1.5m power lead to be a little short for some setups, mainly when we tried to move out into the garden for testing – although we note a Power over Ethernet Splitter (£15/$18, **magpi.cc/vizypoe**) that would work well in this setting. There is also an optional outdoor enclosure with IP66 rating and a 4G LTE cellular network adapter.

Vizy provides you with a great introduction to image recognition that does a lot of the heavy lifting, allowing you to focus on practical projects. We think this would be ideal in a classroom setting.



▲ Here, we are accessing Vizy to set up a pet detector that sends alerts when a cat sits on the table (we are using a stuffed teddy bear as a test unit)

Verdict

An interesting camera that allows you to quickly investigate image recognition projects.

9/10

PiBoy XRS

SPECS

INPUT:
2 × analogue sticks, 1 × D-pad, 7 × face buttons, 4 × shoulder buttons

PORTS:
3.5 mm audio jack, mini HDMI out, 2 × USB 2.0, 2 × USB 3.0, 1 × Gigabit Ethernet

DIMENSIONS:
162 × 93 × 36 mm

DISPLAY:
3.5" IPS LCD

▶ Experimental Pi ▶ **magpi.cc/piboyxrs** ▶ £125 / $150

A more ergonomic handheld experience, the PiBoy XRS takes the best parts of a lot of classic consoles. **Rob Zwetsloot** has a play

We were big fans of the PiBoy DMG, a Raspberry Pi 4–compatible handheld in the shape of the classic Game Boy, albeit with a few more buttons and an analogue stick. It's a great form factor for nostalgia purposes. However, even Nintendo switched to a horizontal format eventually. This brings us to the PiBoy XRS which, while quite like the DMG version, is definitely a superior product.

Taking design notes from the original Game Boy, Neo Geo Pocket, and just about every portable game system since the release of the Game Boy Advance, this updated kit also whacks on a second analogue stick for much improved playability. Instead of two shoulder buttons, there are now four, and while that does mean two fewer face buttons, for the vast majority of games it's a layout we prefer. It turns out there's a reason why handhelds went horizontal after the Game Boy.



▲ There's easy access to the microSD port, and plenty of ventilation for Raspberry Pi 4 too

Hook it up

Installing a Raspberry Pi 4 is very simple – with the supplied screwdriver, you just need to take the eight screws off the back and then slot Raspberry Pi into the GPIO pins. There's some thermal paste for the processor, and an optional adapter for the mini HDMI out. Four of the screws on the back-plate go through the mount holes on Raspberry Pi, making for a very elegant reassembly process.

Power and headphone jacks are routed to the underside of the console, with rechargeable batteries, adding up to 5600 mAh, connected to that USB-C power port for hours of play time. The standard USB and Ethernet ports are then left open at the top of the XRS for easy, if not perhaps slightly awkward, access for peripherals and memory sticks. Those pop out of a fake cartridge slot, which is a nice little design touch, especially paired with the hand-detachable cover that allows you to change out microSD cards without having to get the screwdriver back out.

Playing with power

Like DMG, the whole package is solid and feels good to hold – we find some handheld Raspberry Pi consoles can feel a little flimsy and, while this may be a little heavier than those, we appreciate the added heft.



The joysticks remind us of the smaller ones on the Switch Joy-Cons

Playing games is incredibly simple. Once the PiBoy image is installed, you can game basically straight away and as it's the familiar RetroPie interface, you can easily customise it to your liking. Compatible games run smoothly thanks to the extra oomph of Raspberry Pi 4, and everything on the screen looked lovely as we played. Our kit came with the mini HDMI adapter, and we had

> Once the PiBoy image is installed, you can game basically straight away

no problems playing games on a bigger TV with it plugged in either – although, due to the graphical prowess of some games, it did look better on the smaller screen with a higher pixel density.

Also, as it has RetroPie, it means you can install Steam Link, so it's a pretty cool alternative to a Steam Deck.



Access to the USB ports is handy however you play it. There are much fewer use cases for the Ethernet port, though

Verdict

The perfect retro handheld system is finally here, and you can even use it to comfortably play modern games as well.

10/10

Argon NEO 5

▶ Argon 40 ▶ **argon40.com** ▶ From £19

A new Raspberry Pi means new cases. So what's the well-dressed computer wearing this season? **PJ Evans** suits up

SPECS

COOLING:
Passive and active, air intake vents

FAN:
30mm PWM

MATERIALS:
Aluminium, plastic



▶ Strong, dark and handsome, the NEO 5 case looks good

A new form factor for Raspberry Pi 5 means existing cases need updating, especially if you want efficient cooling. Thankfully, the official case was available from day one and it does a great job of protecting your Raspberry Pi 5 from the elements. However, if you need something different, offering more robustness, cooling options and its own style, Argon 40 has always brought something to the table, from their original NEO cases to the iconic ONE. Now it's Raspberry Pi 5's turn.

Argon has launched its entry-level case, the NEO, for our new member of the family. For about £20 you receive an aluminium enclosure with a funky red plastic base. More interesting is the built-in active cooling system that connects to the new on-board fan header. In addition to the fan are heatsinks for the CPU and a power management circuit. All of this, along with good labelling and thoughtful exposure of the various headers and GPIO, make for a solid package.

The enclosure is provided with instructions. These are straightforward, although we will confess to some confusion with wiring in the fan cable. It took four attempts before the 'ah ha!' moment when we realised what was intended. The instructions could cover this better. The rest of the installation was painless and thermal pads are included for good contact with the heatsink. Once you have Raspberry Pi inserted, you can add the aluminium cover which gives the overall package a solid, strong feel making it more suitable for environments such as factories or classrooms. The case is held together by screws, adding to the strength of the overall package. You can choose between rubber feet (supplied) or wall-mounting using the built-in screw points.



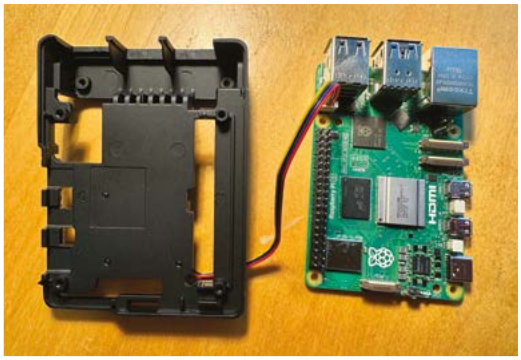
The NEO combines strength and looks

The case in use

The aluminium cover looks great, but it does mean that the GPIO, PCIE, camera ports and other headers become inaccessible. You are not required to have the cover on to use your Raspberry Pi but it would be great to see alternative covers just as with the official cases. The SD card slot is exposed (which is not the case with many third-party cases) and a nice touch is a cover which can be screwed into place to protect the card from accidental removal. A welcome feature is the combo power light and switch, which links to the new on-board power switch.



With the cover removed, the NEO exposes all important headers and the GPIO



Assembly was straightforward but the fan wiring was confusing

Being actively controlled by Raspberry Pi OS means that the fan is silent for the majority of the time and you've got to get the CPU nice and busy to get any noise out of the case, which is minimal. As the body is nearly all aluminium, there is plenty of area for soaking up heat, so this would be a great choice for Raspberry Pis under heavy load.

> Argon's build quality is some of the best in Raspberry Pi's space

Argon's build quality is some of the best in Raspberry Pi's space and this case is no exception. You get the impression that a fall off a desk (normally due to a curious cat) would do it no harm whatsoever. The plastic base has a cheaper feel, but the red accents it provides look the part.

The NEO 5 is another excellent product from Argon that combines value and features into a hard-to-beat package.

Verdict

A low-cost, good-looking case with excellent resilience and cooling options. A solid choice for any Raspberry Pi project, although some may wish to wait for the next generation of the Argon ONE.

8/10

Argon ONE V3 M.2 NVME Case

SPECS

COMPONENTS:
Argon ONE Pi 5 V3 case, M.2 NVME carrier board, Video/Audio PCB extender (daughterboard), GPIO & Fan board, RP2040-based microcontroller

INPUT/OUTPUT
Adjustable M.2 NVME with M-Key up to 2280 size, 2 × standard (type A) HDMI ports, Ethernet, 4 × USB-A ports, USB-C power port, 3.5mm audio jack

COOLING
Aluminium alloy case for passive cooling, blower type PWM programmable 30mm fan

▶ Argon40 ▶ **magpi.cc/argononev3** ▶ £40 / $49

Argon's legendary case gets a Raspberry Pi 5 upgrade with built-in NVMe storage capability. **Lucy Hattersley** builds her dream computer

Argon40 has made a name for itself amongst the Raspberry Pi community by making high-quality cases that transform our favourite single-board into a professional-looking computer, or a smart feature-packed media player.

This month we have the hotly anticipated Argon ONE V3 M.2 NVMe PCIe Case. Hopes are high for this one and we can tell you it hits home runs all the way. Redesigned for Raspberry Pi 5 this board combines the features of the Argon ONE V2 with the – previously separate – M.2 Expansion Board add-on to create an all-in-one computer case with super-fast, and super-large, storage that also cleverly uses passive cooling and heatsinks to keep everything running. This is the one we have been waiting for.

All together now

The Argon ONE is Argon40's flagship case, containing a daughterboard for Raspberry Pi that adds additional features such as full-sized HDMI sockets and an infrared (IR) receiver.

On the base of Argon ONE V3 is a removable cover that provides access to the M.2 NVMe socket. Here you can insert any M.2 NVMe with M-Key up to 2280 size. The flap is marked "THERML" which nods to its aluminium heatsink and a long strip of thermal pad is included to transfer the heat out into the case.



▶ Argon ONE V3 is a neat case made from a black aluminium alloy that helps dissipate heat

Two more silicon pads are included to connect Raspberry Pi's CPU and PMIC (Power Management Integrated Circuit) to the case.

Alongside this impressive passive cooling is a redesigned 30mm fan and blower. This is repositioned at an angle to be "more efficient and quiet", and we found it unobtrusive even when stress testing.

Argon ONE V3 now sports Raspberry Pi's RP2040 microcontroller to control various functions like fan speed and power management (via jumper pins on the daughter board). The power button is less of a novelty now that Raspberry Pi 5 itself features one. However, the presence of a 3.5mm audio jack will be a welcome addition for audio buffs now that it has been removed from Raspberry Pi 5's main board.

One advantage over the Argon ONE V2 board is that the M.2 NVMe now connects directly to the PCIe socket on Raspberry Pi 5. This leaves all four USB-A sockets available.

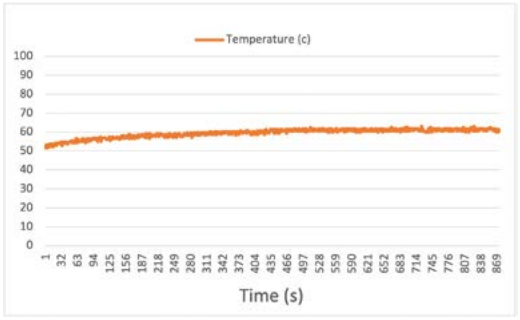
A removable magnetic flap on top of the case provides access to repositioned GPIO pins alongside a handy pinout guide.

Assembly and testing

Putting together the Argon ONE V3 was a relatively painless process thanks to the included assembly instructions (**magpi.cc/onem2man**). Raspberry Pi



◀ Underneath Argon ONE V3 is a removable thermal heatsink to fit the M.2 NVMe drive. Here we can also see the power button, full-sized HDMI slot, 3.5mm audio jack and Raspberry Pi 5 ports



▼ Heat testing Raspberry Pi 5 inside the Argon ONE V3 showed it took five minutes (300 seconds) to reach the 60°c point, where the fan started, and remained there for the rest of the 15-minute (900-second) test. At no point was speed throttled

5 is connected to the HDMI daughterboard, and the PCIE cable is used to connect the bottom half of the case to Raspberry Pi 5. Then the whole thing is screwed together. Finally, the M.2 NVMe storage is connected to the underside making it possible to upgrade the drive without opening the whole case.

As with previous Argon ONE cases, the microSD card is hidden away and can't be used without opening up the case. This is less problematic these days as a USB thumb drive flashed with Raspberry Pi OS can be used to run Imager and flash the storage drive. Attach an Ethernet cable and you can also use Network Install with Raspberry Pi 5 (**magpi.cc/networkinstall**).

> An all-in-one computer case with super-fast, and super-large storage

Speed-testing of the NVMe drive tells you much more about the quality of your drive than the case itself. We used a 500GB WD Blue storage stick and measured the speed using Raspberry Pi OS's built-in Raspberry Pi Diagnostics tool to test performance. It returned a sequential write speed 789590 KB/sec (790 MB/s) almost 80 times faster than the recommended pass speed for a microSD card. It's fast.

The heat test is also interesting. We used `stress --cpu 4` and measured the output with the script found at **magpi.cc/thermaltesting** for 15 minutes.

Raspberry Pi 5 inside the Argon ONE V3 case idles at around 54°c (down from the 65°c baseline of an uncooled Raspberry Pi 5 without a heatsink). We found the fan kicked in at the 60°c mark after five minutes and kept the Raspberry Pi 5 hovering around 61°c for another 10 minutes. At no point did Raspberry Pi OS reach the 80°c mark where performance throttling begins. It compared favourably to a Raspberry Pi 5 and an official Active Cooler unit.

Put it to use

One optional extra we should mention is an internal Argon ONE BLSTR DAC audio board upgrade, which will be sold separately. Alongside the 3.5mm jack this will make Argon ONE V3 ideal for audio fans. We didn't have one for testing and can't see it on the Argon40 website just yet, so hopefully that will come down the line.

All of this transforms Raspberry Pi 5 from a hackable board to a desktop computer. A role our favourite computer is increasingly fulfilling with aplomb. The built-in infrared connection, large storage, and full-sized HDMI connection also ensure Argon ONE V3 becomes the perfect media player or home games console. This case is highly recommended.

Verdict

An excellent case that sees a lot of Argon's ideas reach fruition. Turn your Raspberry Pi 5 into a smart desktop computer, media player, games console or DAC audio player.

10/10

NVMe Base

SPECS

FEATURES
Works with most NVMe M.2 SSDs; Flat flex PCIe cable; Standoff kit; 4 × rubber feet

CONNECTIONS
M.2 Key slot, PCIe slot (18-pin)

▶ Pimoroni ▶ **magpi.cc/nvmebase** ▶ From £14 / $14

Give your Raspberry Pi 5 a major storage speed boost with this PCIe M.2 SSD adapter. By **Phil King**

The NVMe Base enables you to connect an NVMe (non-volatile memory express) solid-state drive to Raspberry Pi 5's PCIe slot for storage with far faster read/write speeds than a microSD card.

Rather than coming in the HAT form factor to sit on top of Raspberry Pi, it fits underneath, hence the "Base" moniker. This means it doesn't obstruct a cooling fan, if you're using one, nor tie up the GPIO pins, so you can still add another HAT on top.

Before attaching the NVMe Base to the underside of Raspberry Pi 5 using the supplied standoff kit – demonstrated in Pimoroni's installation video (**magpi.cc/nvmebaseyt**) – you'll want to insert your NVMe SSD stick into the Base's M.2 key slot. The board is long enough to accommodate 2280 size SSDs, and has mounting holes for this plus 2230, 2242, and 2260 drives – so you can secure it with a bolt and nut.

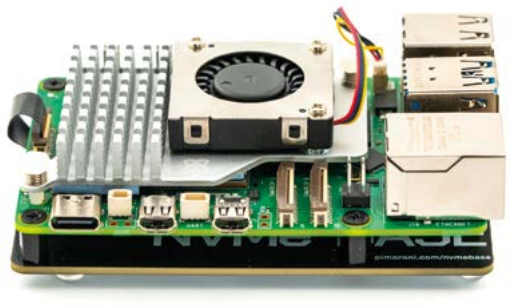
Securing the Base

The next step is to attach M2.5 standoffs to the top of the Base using the four mounting holes. The kit helpfully includes both short and long bolts – the latter are useful if you want to securely mount another HAT on top of Raspberry Pi 5.

Connecting the Base's PCIe slot with the one on Raspberry Pi 5 is done using a small, flat S-shaped cable that flexes and has labels to help you orient it correctly – the end for the Base is slightly wider, at 18 pins. It's easier to connect Raspberry Pi 5's PCIe slot first, then the Base's, due to the latter's less fiddly flip tab. You can then fold



▼ The NVMe Base comes with a standoffs kit, flat flexible PCIe cable, and four rubber feet



▲ The slimline NVMe Base fits snugly under Raspberry Pi 5, secured by bolts and standoffs

▼ A short flexible cable connects the NVMe Base to Raspberry Pi 5's PCIe slot



the flexible cable over so the Base is underneath Raspberry Pi 5 to form a sandwich, before using a small screwdriver and bolts to secure it. The only downside is that the extra height means it won't fit in a standard case.

With the hardware installed, you're ready to start using your SSD… that is, once you've checked that your system is up to date and you have the latest bootloader version selected in raspi-config: Advanced Options > Bootloader Version > Latest, then select 'No' and reboot. The drive should then appear in the **/media** directory, and be shown by the lsblk command. If not, make sure it is formatted (you can use Raspberry Pi Imager).

Drive compatibility

While the majority of NVMe M.2 drives should work fine, Pimoroni's product page notes that a few models have quirks or have proved troublesome. The safest option is to choose a tested model listed there, or purchase the NVMe Base bundled with a compatible 250GB or 500GB SSD.

Raspberry Pi 5 officially only supports the Gen 2.0 version of PCIe, but adding an extra line to **/boot/config.txt** will force Gen 3.0 for extra speed.

> We achieved a write speed of 514MB/s and a read speed of 858MB/s

▼ The underside of the NVMe Base board, secured to Raspberry Pi 5 with bolts and standoffs



In our tests, using the `dd` command, we achieved a write speed of 514MB/s and a read speed of 858MB/s. While far from our SSD's maximum (due to only using one of its PCIe lanes), it's still many times faster than microSD (typically around 30MB/s write, 90MB/s read), and also better than an SSD connected via speed-limiting USB. Check out Pimoroni's own tests on various SSDs at **magpi.cc/nvmebasetests**.

You'll want to make Raspberry Pi 5 boot from the SSD instead of the microSD card. This is easily achieved by writing the OS to it with Raspberry Pi Imager and then selecting Advanced Options > Boot Order > NVMe in raspi-config. For our drive, this cut around four seconds from the average boot time. We also noticed that some apps, such as Chromium, seemed a little snappier.

Verdict

A slimline adapter that sits neatly under Raspberry Pi 5 and enables fast PCIe read/write speeds with a suitable SSD.

9/10

PuppyPi

▶ HiWonder ▶ **hiwonder.com** ▶ £445 / $559

State-of-the-art quadruped combines Raspberry Pi and ROS for powerful learning options. By **Lucy Hattersley**

SPECS

DIMENSIONS:
226 × 149 × 190 mm (length × width × height), 720 g, aluminium alloy body

COMPONENTS:
Raspberry Pi 4 4GB, eight HPS-0618SG coreless servos, Raspberry Pi expansion board, 480P/8DOF camera

CONTROL METHODS:
iOS/Android apps, PC application, Python API, PuppyPi V1 software, Lab tool, PS2 controller

There's no denying that PuppyPi is an eye-catching robot. Inside its black aluminium alloy frame this quadruped features eight high-quality HPS-0618SG coreless servo motors with metal joints, and a six-axis IMU sensor chip MPU6050 for robot posture detection.

PuppyPi (**magpi.cc/puppypi**) is assembled by HiWonder, who created the larger SpiderPi robot (**magpi.cc/119**) and TonyPi (**magpi.cc/111**), all strong contenders in the robotics space.

At the front of PuppyPi sits a 480p camera and inside is a Raspberry Pi 4 running ROS (robot operating system), a motor control board, and 7.4V 2200mAh LiPo battery. It's a lot smaller than SpiderPi; it's cuter too, even with the black metal styling and single-eye camera. Inside the box are some coloured balls, tags, and an 8.4V charger. We also got the PS2-style controller for testing, and an Advanced kit comes with extra sensors and attachments. A pro model comes with Lidar.

Our test model worked fresh out of the box. Power it up using the switch on the side, and it instantly stands up and transmits a Wi-Fi



▶ The quadruped looks stylish with its black aluminium casing

network to join. The HiWonder iOS/Android app enables direct control of PuppyPi from a smartphone. You remotely control the robot as the camera view displays on your screen. A range of pre-defined actions are available: sit, bow, shake hands, and so on. There are also colour recognition, target colour tracking, QR code tag recognition, OpenCV face detection, and line-following modes.

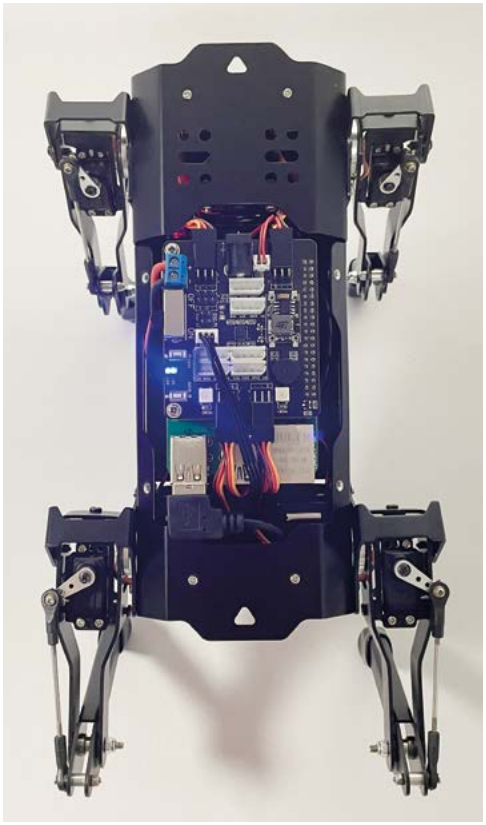
You control the robot via the Android or iOS app, a desktop app, or using the optional PS2-style controller. Next, we flipped the protective lid off PuppyPi to take a closer look inside. There is a Raspberry Pi 4 with 4GB RAM and a motor control and expansion board. This features a barrel jack for charging, four servo ports, two GPIO ports, and four IIC ports, RGB lights, and two programmable buttons. Extensions offered by HiWonder include a Lidar module, ultrasonic sensor, MP3 module, display, and touch sensor.

The software running on PuppyPi is Raspberry Pi OS with custom applications and ROS (robot operating system) support. Connecting to Raspberry Pi via VNC reveals a PuppyPi desktop application that is used to visually program and chain together commands. You can dig deeper with a Python API, and it supports Gazebo simulation (**gazebosim.org**), enabling you to simulate movement in advance.

Deep training

Once you've connected PuppyPi to your network, it's possible to move beyond the iOS/Android app and dive into the world of Python programming and ROS experimentation. To this end, a huge range of tutorial PDFs are provided by HiWonder via Google Drive (**magpi.cc/puppypidocs**). These cover everything from Python, Gazebo, ROS and OpenCV vision tracking; up to stair climbing and obstacle avoidance.

We didn't encounter any real problems during testing. Both Raspberry Pi OS and the PuppyPi PC_Software app defaulted to Chinese (easy enough to fix in Preferences). PuppyPi is obviously not cheap in the world of Raspberry Pi robotics, but Boston Dynamic's Spot robot comes in at roughly £200,000, so it's relative. We think this would be an excellent addition to any robotics course or classroom.



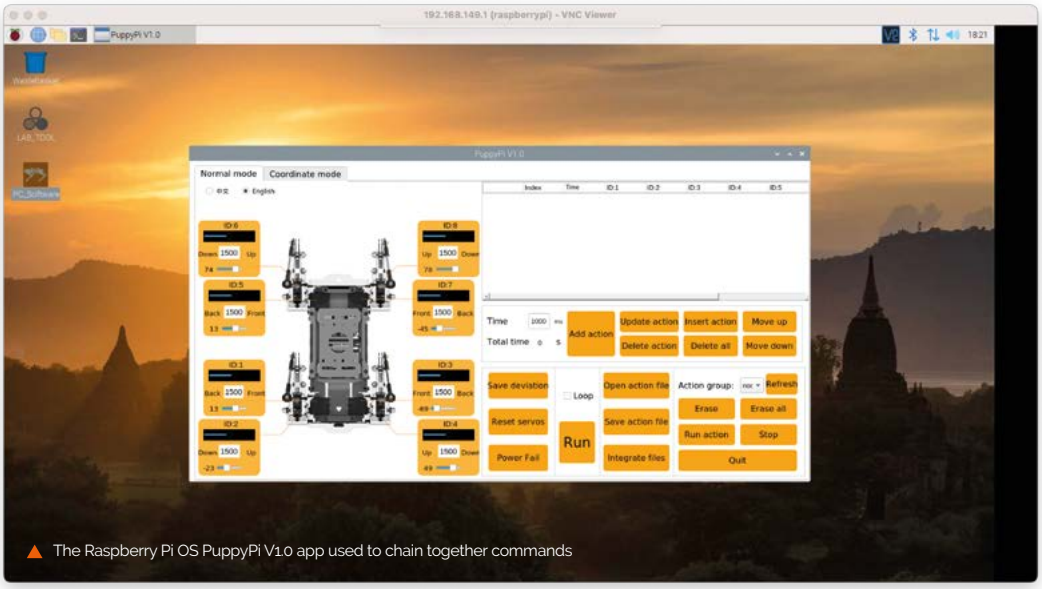
> An excellent addition to any robotics course or classroom

The Raspberry Pi accessory board connects to the servo motors, and provides GPIO and IIC connections for accessories

Verdict

A fantastic quadruped that's easy to set up and start using. Packed with detailed documentation to explore every facet of dog-themed robotics.

9/10



The Raspberry Pi OS PuppyPi V1.0 app used to chain together commands

CM4 XGO-Lite Robot Dog Kit

▶ Elecfreaks ▶ **magpi.cc/cm4xgo** ▶ £469 / $599

A Compute Module 4-powered robot dog with five limbs and a huge amount of potential. **Rob Zwetsloot** takes it for a walk

SPECS

SIZE:
250 mm × 145 mm × 170 mm while standing, 620g

LIMBS:
Four walking limbs and one gripper allowing for 15 degrees of movement, and three on the arm

PROGRAMMING LANGUAGES:
Blockly, Python, ROS

BATTERY:
2500 mAh lithium ion battery with 120 minute life

We've seen a small explosion of dog-like robot kits over the past few years, although we cannot put a finger on what might have started it. Perhaps the cuteness of a four-legged pet is hard to resist when you can program it yourself, which brings us to the very powerful XGO line of robot kits.

Unlike a lot of a robot kits, this one has metal construction throughout, making it chunky and weighty in the satisfying way quality technology can be. It also comes all pre-assembled so you won't be bending any metal plates or doing any complicated wiring to get it working, like with other legged robots.

Smart dog

At its core is a powerful Compute Module 4, controlling the robot and allowing for reprogramming with custom code over a web browser – although you can also plug it into a monitor (or go a VNC route) if you want and program it the old-fashioned way. There are several bits of example code that illustrate the impressive features of the robot, such as facial recognition with the inbuilt camera, remote control, and more. There's a full breakdown of the Python commands on the quite detailed documentation over at **magpi.cc/xgodocs**, and it even has a block code editor for people not quite up to speed on Python.

It's definitely a very well put together product, although it's not the most stable looking robot we've ever seen. The price is also a little eyewatering, but we think the tech inside justifies it, especially if you're looking for a more advanced bit of kit.

> "There are several bits of example code that illustrate the impressive features of the robot, such as facial recognition"



▼ The box comes with some coloured balls that you can program XGO to pick up

Verdict

A very impressive robot kit that while fine for beginners may be better suited to those already into robotics due to the price

9/10

Inventor HAT Mini

▶ Pimoroni ▶ **magpi.cc/inventorhatmini** ▶ From £24 / $25

Mini all-in-one board for robotics and more. By **Phil King**

SPECS

INPUTS/OUTPUTS:
4 × GPIO/ADC headers, 4 × servo headers, 2 × JST-SH (6-pin) motor ports, 2 × regular motor outputs, 1 × Qwiic/STEMMA QT port, audio output, external power input, UART header

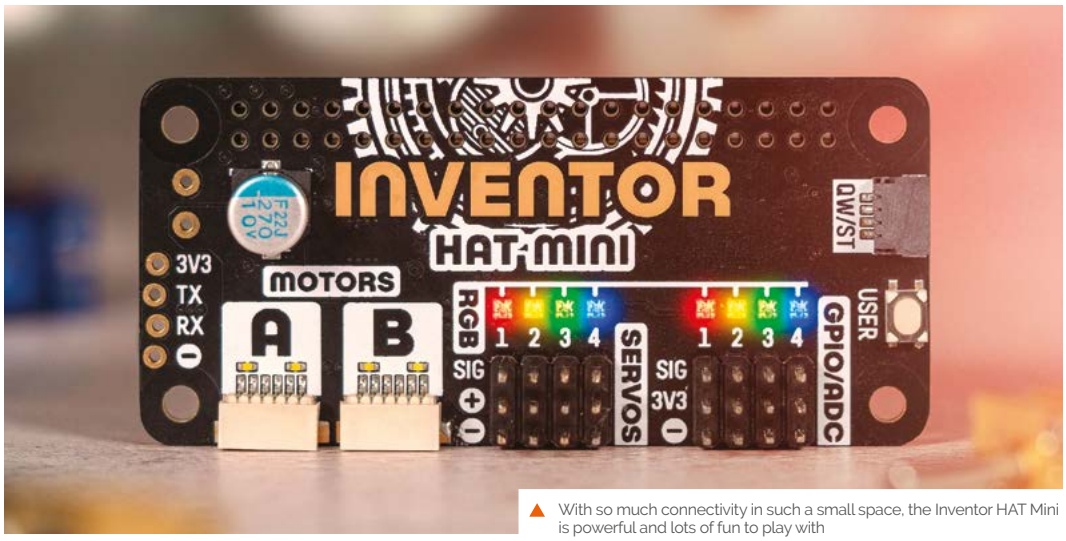
FEATURES:
Nuvoton microcontroller, 8 × WS2812B RGB LEDs, User button, DRV8833 dual H-bridge motor driver, motor direction LEDs, pass-through header

DIMENSIONS:
65 × 30.5 × 15.5 mm

Verdict

Packing an impressive array of features into a small footprint, this versatile board is ideal for mini robots and a lot more.

9/10



▲ With so much connectivity in such a small space, the Inventor HAT Mini is powerful and lots of fun to play with

If you are looking for a HAT for all occasions, the Inventor HAT Mini could be it. This mini add-on board for Raspberry Pi computers is packed with features and could prove useful in a wide variety of projects.

Robotics is an obvious use case thanks to the HAT's two motor ports and four three-pin servo headers. Note that the motor ports are the six-pin type used for Pimoroni's encoder-equipped motors, but there's also a header underneath to connect jumper wires to standard motors.

> We can see this board being used with a Raspberry Pi Zero for fun mini robots

Four extra GPIO channels, each with a three-pin header, are ADC capable, so can be used as analogue inputs for sensors etc. Best of all, the board's Nuvoton microcontroller handles all I/O, so the board only uses a handful of GPIO pins on Raspberry Pi – and it has a pass-through header so you could stack another board on top by using a stacking header.

Light and sound

Each set of header pins even has its own RGB NeoPixel, while the motor ports have direction LEDs. There's also a user button and other connections include a single Qwiic/STEMMA QT for breakouts, external power and UART serial headers (both unpopulated), and an audio port (two-pin Picoblade) to connect a mini speaker.

Pimoroni's comprehensive Python library (**magpi.cc/inventorhatminigh**) makes programming the board easy, and a bunch of code examples help you get started. We can see this board being used with a Raspberry Pi Zero for fun mini robots, but it's versatile enough for many other electronics projects.

10 Amazing:

Cases

Protect your Raspberry Pi with these excellent enclosures

Raspberry Pi is hardly fragile, but it really is recommended to get some kind of case for your microcomputer (or microcontroller), even if you're just cutting some holes in the cardboard box it comes with. Here are ten great cases that are a little more sturdy though.



▲ SmartiPi Touch Pro

Case, stand, display

With an official touch screen and a Raspberry Pi, you can create a quick interactive display that needs very little space and is extremely customisable.

magpi.cc/smartipipro | £30



▲ FLIRC case

Media shelf ready

Designed for use by your TV, this range of cases doesn't look out of place next to Blu-ray players or satellite boxes.

magpi.cc/flirc | £12



◀ Argon EON NAS

File server

The price may be eye-watering, but this enclosure allows you to add four hard drives, includes an RTC, cooling fan, programmable OLED display, and room for your Raspberry Pi.

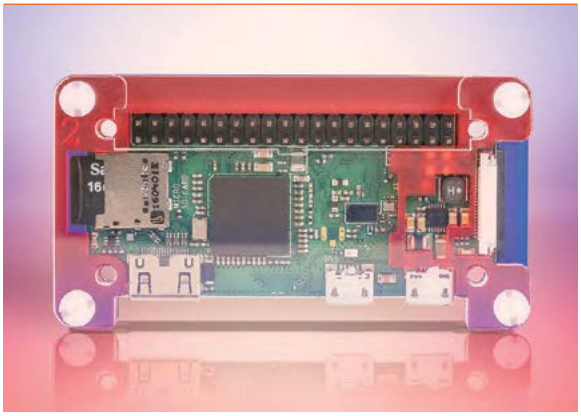
magpi.cc/argoneon | £144

▼ Pibow

Layers of case

Pimoroni's original flagship product is this sturdy case made up of laser-cut acrylic layers. There are many styles and colours to get as well.

magpi.cc/pibow | £6





SecurePi Case

Locked down

This case allows you to cover up IO ports on Raspberry Pi, including the SD card slot, for increased security.

magpi.cc/securepi | £10



PiBoy XRS

Handheld alternative

We gave this case a 10/10 a few months ago in *The MagPi* as it's pretty much the ultimate retro handheld machine. You can even stream Steam games to it.

magpi.cc/piboyxrs | £120 / $150



Official case

Simple and clean

The official case snugly fits a Raspberry Pi inside. There's a version of all types of Raspberry Pi, and even a case fan if you're using your Raspberry Pi at full power regularly.

magpi.cc/case | £6



NESPi 4 case

Classic console body

Lots of people like to use a Raspberry Pi for retro gaming, so what better way to display it than with it inside a retro console case with a fake cartridge?

magpi.cc/nespi4 | £29



Zebra Pico case

Small layers

Due to the way Pico is used as a microcontroller inside projects, you rarely see cases for it. It still looks great in one if you're using it for prototyping though.

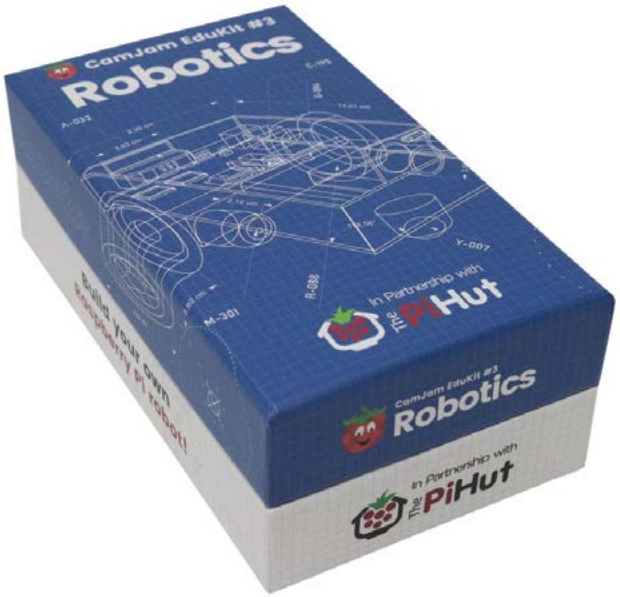
magpi.cc/zebrapico | £7 / $9

10 Amazing:

Starter kits

Get started on your first Raspberry Pi build

We still get people asking us what they can do with Raspberry Pi. It's far easier to answer that these days than it was in 2012 thanks to gigantic support from the worldwide community creating kits and inspiring projects. Here are ten ways to get started with your Raspberry Pi...



CamJam EduKit #3 - Robotics

Robots for newbies

Our go-to robot starter kit recommendation, the CamJam kit 3 has all the basics you need to get a robot built. You just need to supply your own Raspberry Pi.

magpi.cc/camjamedu3 | £20 / $25



Raspberry Pi 4 Desktop Kit

Plug and play

Everything you need to get started with a Raspberry Pi 4 – including a Raspberry Pi 4, case, accessories and a complete Beginner's Guide.

magpi.cc/pi4desktopkit | £123 / $153



Pi-hole Raspberry Pi 4 kit

Ban dangerous ads

Pi-hole is a great software project that allows you to block ads from entering the network, keeping your personal devices safe from malicious code. It's also an interesting way to learn the ins and outs of networking.

magpi.cc/piholekit | £82 / $102

Raspberry Pi Zero 2W Starter Kit

Small but mighty

Raspberry Pi Zero 2 is very powerful, but you still need some adapters to get the most out of it. This kit includes a Zero 2 and all the basic accessories you might need for it.

magpi.cc/zero2sk | £51 / $67





Raspberry Pi 400 Personal Computer Kit

All-in-one

A full Raspberry Pi computer built into a keyboard, much like the home computers of yore. That makes it very portable – all you need is a spare monitor.

magpi.cc/raspberrypi400 | £114 / $142

Starter Kit for Raspberry Pi Pico

Electronics box



Pico is a great microcontroller for powering all your electronic and IoT projects – this kit comes with a Pico with a GPIO header attached, and loads of components to learn to code with.

magpi.cc/picoeleckit | £10 / $12

Grove Base Kit

Home automation



Fancy automating your home? Grove is a great system for experimenting with IoT, and lets you learn how it works before being easily upgradeable to full home control.

magpi.cc/grovebasekit | £39 / $49



Raspberry Pi Pico Advanced Kit

32-in-1

This kit holds a huge number of different components, making it an amazing starter kit for someone who wants to learn everything about microcontollers, robots, and coding.

magpi.cc/advpicokit | £37 / $46

Electronics Kit for Pico

More with Pico



Already got a Pico? Then this kit is a good way to learn a lot of circuitry thanks to a mixture of LEDs, resistors, and a motor.

magpi.cc/picolite
£16 / $20

CrowPi L

Laptop kit



Part of the CrowPi family of Raspberry Pi laptops, using your own Raspberry Pi inside an advanced shell to create the ultimate portable Raspberry Pi.

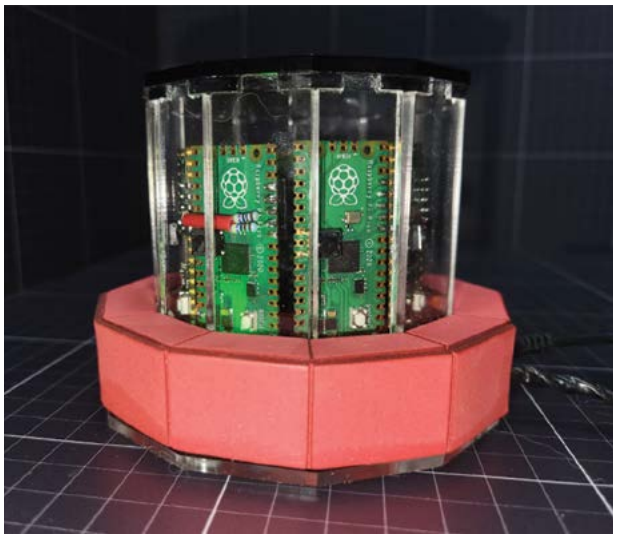
magpi.cc/crowpil | £176 / $219

10 Amazing:

Raspberry Pi Pico projects

You can do incredible things with Raspberry Pi Pico and RP2040

Raspberry Pi 5 is here, and it is excellent, but we've not forgotten about Raspberry Pi Pico. We still see excellent builds using it and the RP2040 chip that powers it on a regular basis and we do not expect that to let up any time soon. If you're after some inspiration for your own Pico build, here are ten excellent ideas...



PicoCray

1/50 supercomputer

Distributed computing on a Pico may not really be practical, but it is very fun and a cheap little project.

magpi.cc/picocray



Pico Held

Pico Mega Drive

This very impressive project is a handheld that is able to emulate the Sega Mega Drive/Genesis on just a Pico. It's quite a feat.

magpi.cc/picoheld



Pico headphones

3D-printed cans

While you wouldn't normally think of great audio when you think of Pico, these custom 3D-printed headphones are a great DIY project that shows off Pico's power.

magpi.cc/picophones

Raspberry Pi beret

Traffic fashion

Used at a convention to signal to people when the maker was going to turn, this beret features a Pico that runs a web server and controls NeoPixel LEDs via a smartphone.

magpi.cc/picoberet

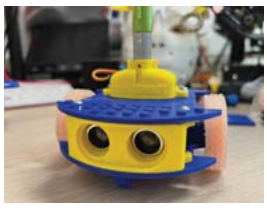


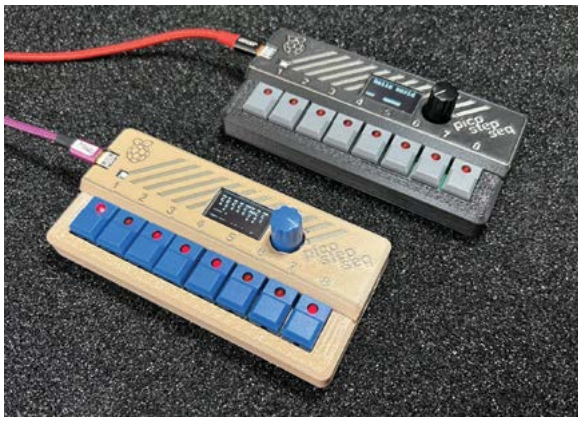
BurgerBot

Pico robot

Kevin McAleer's Pico robot can also draw like the turtle robots of old. It has a cool, unique design and Kevin also offers a course to show you how to make one.

magpi.cc/burgerbot



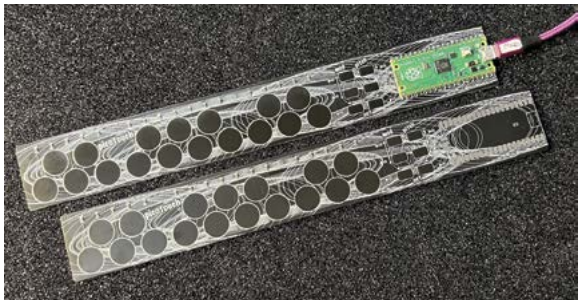


PicoStepSeq

RP2040 music maker

This MIDI sequencer build on Pico includes a little display and a diminutive 3D printed, retro-styled case that is quite fetching.

magpi.cc/picostepseq



PicoTouch synthesiser

Capacitive touch synthesiser

Another musical Pico project, this time a MIDI synth using a capacitive touch strip. Buttons can be used to control the scale on the synthesizer, with C major as the default one.

magpi.cc/ptsynth

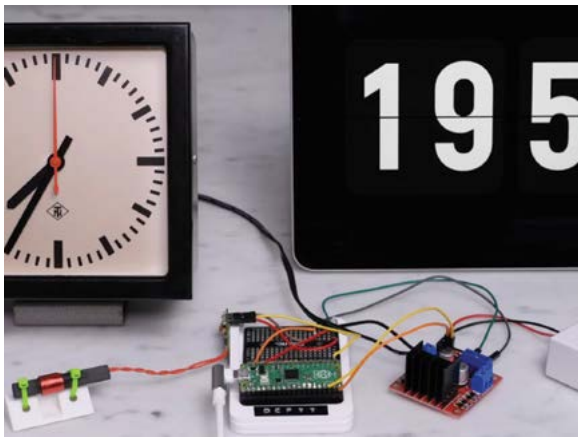
Roboat

Recycled robot boat

Take three empty plastic bottles, a bottle cap, and a motor, and you have all you need for a Pico-powered remote-control boat.

magpi.cc/picoroboat



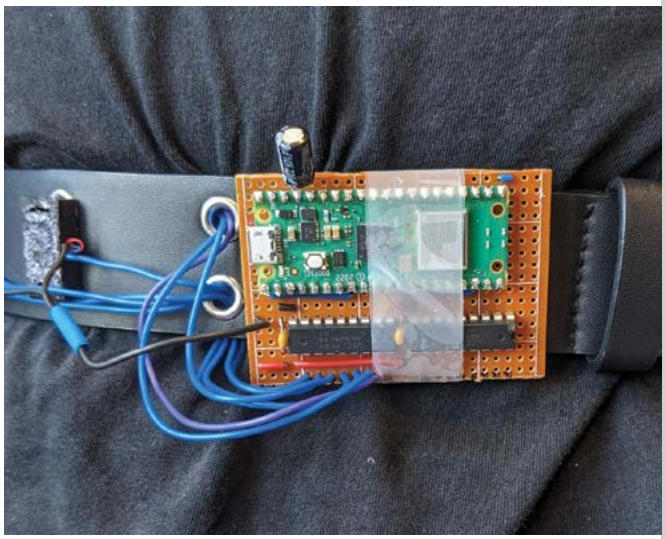


Pico railway clock

Signalling Pico

If you ever pick up an old railway clock at a car boot sale and wonder why it's not working properly, you might need a Pico to send a signal to it, naturally.

magpi.cc/picorwclock



Pico-powered belt

Waist compass

How can a belt be Pico-powered? Well, this one uses a series of vibration motors to tell the user which way is north. Perhaps you can convert it to give directions?

magpi.cc/picobelt


FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
START
WINNER
NEW 2024 UPDATE
RETRO GAMING WITH RASPBERRY PI
3RD EDITION
180 PAGES OF VIDEO GAME PROJECTS
BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!
File manager
Change color
Change icon
Extra blocks
Save dump
Load dump
Hexeditor
Defragment
Reformat
/ : Select
: OK
: Cancel
Display the list of the files found in the selected VMU.

RETRO GAMING WITH RASPBERRY PI

3RD EDITION

Retro Gaming with Raspberry Pi shows you how to set up Raspberry Pi 5 to play a new generation of classic games. Build your gaming console and full-size arcade cabinet, install emulation software and download original games with our step-by-step guides. You'll discover a vibrant homebrew scene packed with new games for original consoles and legal access to all those retro games you remember!

- *Set up Raspberry Pi for retro gaming*
- *Emulate classic computers and consoles*
- *Learn to code retro-style games*
- *Build a console, handheld, and full-size arcade machine*



BUY ONLINE: **magpi.cc/store**

THE *Official*

RASPBERRY PI HANDBOOK

2025

Dive into the world of Raspberry Pi with this huge book of tutorials, project showcases, guides, product reviews, and much more!

Raspberry Pi Pico 2 joins Raspberry Pi 5 in this, the 2025 edition of the Official Raspberry Pi Handbook. These amazing boards can power any project you can imagine.

With 200 pages packed full of maker goodness, you'll also find inspiration for your Raspberry Pi Zero 2 W, Raspberry Pi 4, or any other Raspberry Pi model you may have.





200 PAGES OF RASPBERRY PI

- Get started guide covering every Raspberry Pi
- Everything you need to know about the brand new Raspberry Pi Pico 2
- Inspiring projects to give you your next project idea
- Build a universal media player for your entertainment centre
- Take pictures of the stars with our astrophotography guide
- Upgrade your emulation with next-gen retro gaming