THE OFFICIAL RASPBERRY PI MAGAZINE

The MagPi + HackSpace

Issue 146 | October 2024 | magpi.cc

Raspberry Pi PRESS

£7.99

10
9 772051 998001



PLAY RETRO HORROR CLASSICS ON RASPBERRY PI 5

POLICE LINE DO NOT CROSS

POLICE LINE DO NOT CROSS

POLICE LINE DO NOT CROSS

LEGO CARD SHUFFLER

TOP 10 SPOOKY PROJECTS

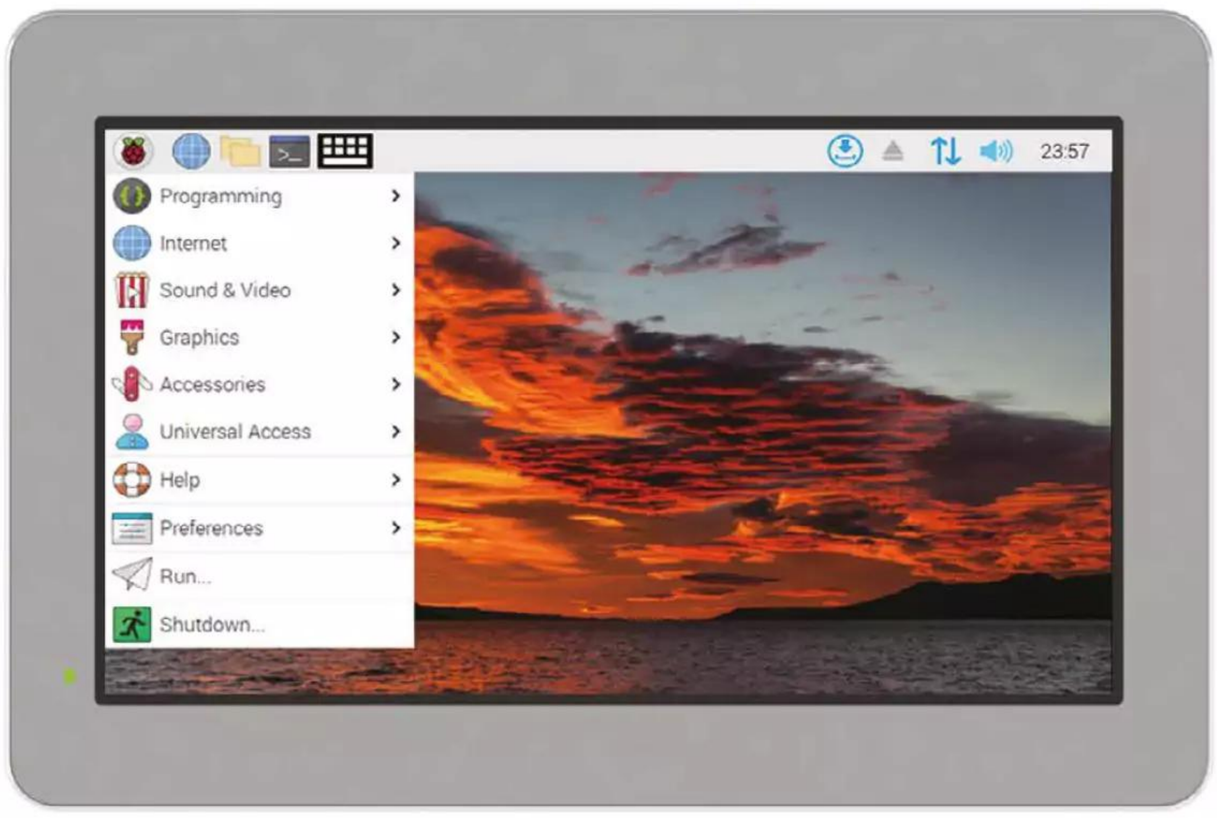
RECYCLE A FIGHTER JET JOYSTICK

TURN IT UP TO 11 WITH AUDIO UPGRADES!

Industrial Raspberry Pi

ComfilePi





UL LISTED

Powered by Raspberry Pi

The ComfilePi is a touch panel PC designed with high-tolerant components and no moving parts for industrial applications. It features a water-resistant front panel, touchscreen, color LCD (available in various sizes), RS-232, RS-485, Ethernet, USB, I2C, SPI, digital IO, battery-backed RTC (real-time clock), and piezo buzzer.

Use the rear-panel 40-pin GPIO header to expand its features and capabilities with additional I/O boards. The ComfilePi is UL Listed and employs Raspberry Pi Compute Module.

Visit www.comfiletech.com

© copyright COMFILE Technology, Inc. ALL RIGHTS RESERVED

COMFILE TECHNOLOGY

WELCOME
to The MagPi 146

Our resident retro-gaming expert KG Orphanides has spent much time perfecting the art of attaching floppy disks and CD-ROMs to Raspberry Pi 5.

When KG suggested a retro horror gaming feature for our October issue, we couldn't say no. The lure of revisiting multimedia classics such as *Alone in the Dark* and *Phantasmagoria* was too much to resist, and we hope you enjoy reading our spooky trip down memory lane.

Meanwhile, Rob has chosen ten creative Halloween projects. So if you fancy building something special to scare trick-or-treaters, be sure to scope it out.

This month's magazine has been a lot of fun to write. We hope you're enjoying our *HackSpace* integration. There's far too much inside the next 132 pages to outline everything, but I especially enjoyed the LEGO Card Shuffler and the repurposed MOD jet fighter joystick. Enjoy the issue!

Lucy Hattersley Editor



EDITOR

Lucy Hattersley

Speaking of retro, Lucy has been working from the British Library newsroom to the soothing clacks of microfilm readers.

magpi.cc

GET A RASPBERRY PI PICO W WITH A SUBSCRIPTION! PAGE 30



The parts we sell help us reach for the stars

Since first venturing beyond the safety of a cave, we have been natural explorers. Discovering what's beyond the next hill, horizon, and star, is a drive that is innately us.

The parts and services we provide help fuel the next era of exploration, and like you, we can't wait to see what's out there.

Discover millions of parts at digikey.co.uk

DigiKey

we get technical

DigiKey is a franchised distributor for all supplier partners. New products added daily. DigiKey and DigiKey Electronics are registered trademarks of DigiKey Electronics in the U.S. and other countries. © 2024 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

Contents

Issue 146 October 2024

Cover Feature

32 Play retro horror classics on Raspberry Pi 5

Regulars

10 World of Raspberry Pi
64 Into HackSpace
122 Your letters
124 Community events calendar
129 Next month
130 The Final Word

Project Showcases

16 LEGO Card Shuffler
20 Portable Pi 84
24 VespAI
28 LED umbrella



32



LEGO Card Shuffler

28



Light-up umbrella

The MagPi is published monthly by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Periodicals Postage paid at Williamsport, PA. POSTMASTER: Send address changes to The MagPi, c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

Tutorials

44 Program with Pico and VS Code
48 Build a private cloud server – part 3
52 Make a M.A.R.S. Rover – part 2
58 ML noise suppression on Pico 2

Top Projects

66 3D-printed loom
68 Arcade briefcase
70 HDSP wristwatch
72 Pilet

Forge

76 How I Made: a Pico joystick
82 Unusual tools: Gauging Blocks
84 Tin can audio player
88 Constructing hex tube rockets
92 Mosaics

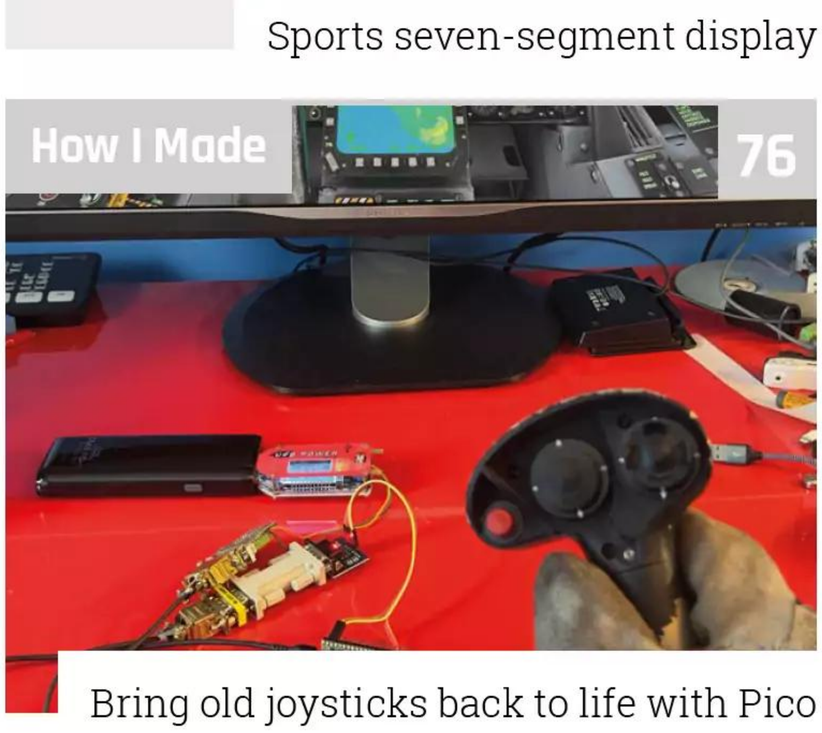


Make a M.A.R.S. Rover

Arcade briefcase



Sports seven-segment display

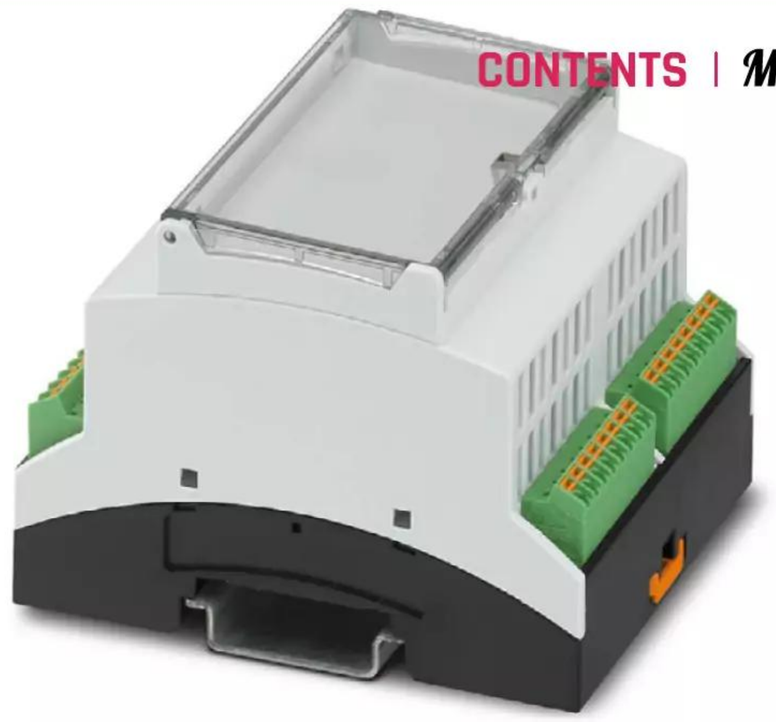


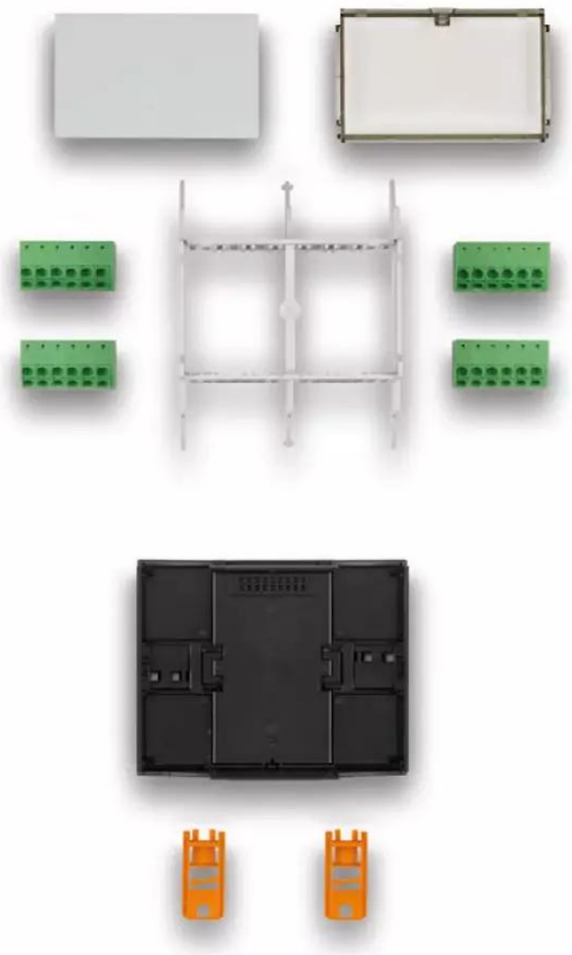
Bring old joysticks back to life with Pico

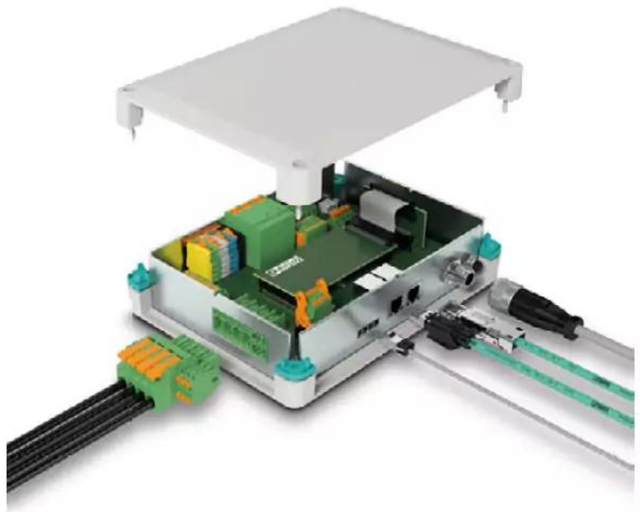


Constructing hex tube rockets









Electronics housings… as a complete kit.

Available exclusively from our global distribution and wholesale partners you will find the right housing for every application.

With wall, desk & DIN rail mounting options our kits make it easier for you to select the next housing solution for your electronics design.

For additional information call 01952 681700 or visit

https://phoe.co/housings-kits-uk

PHOENIX CONTACT
INSPIRING INNOVATIONS

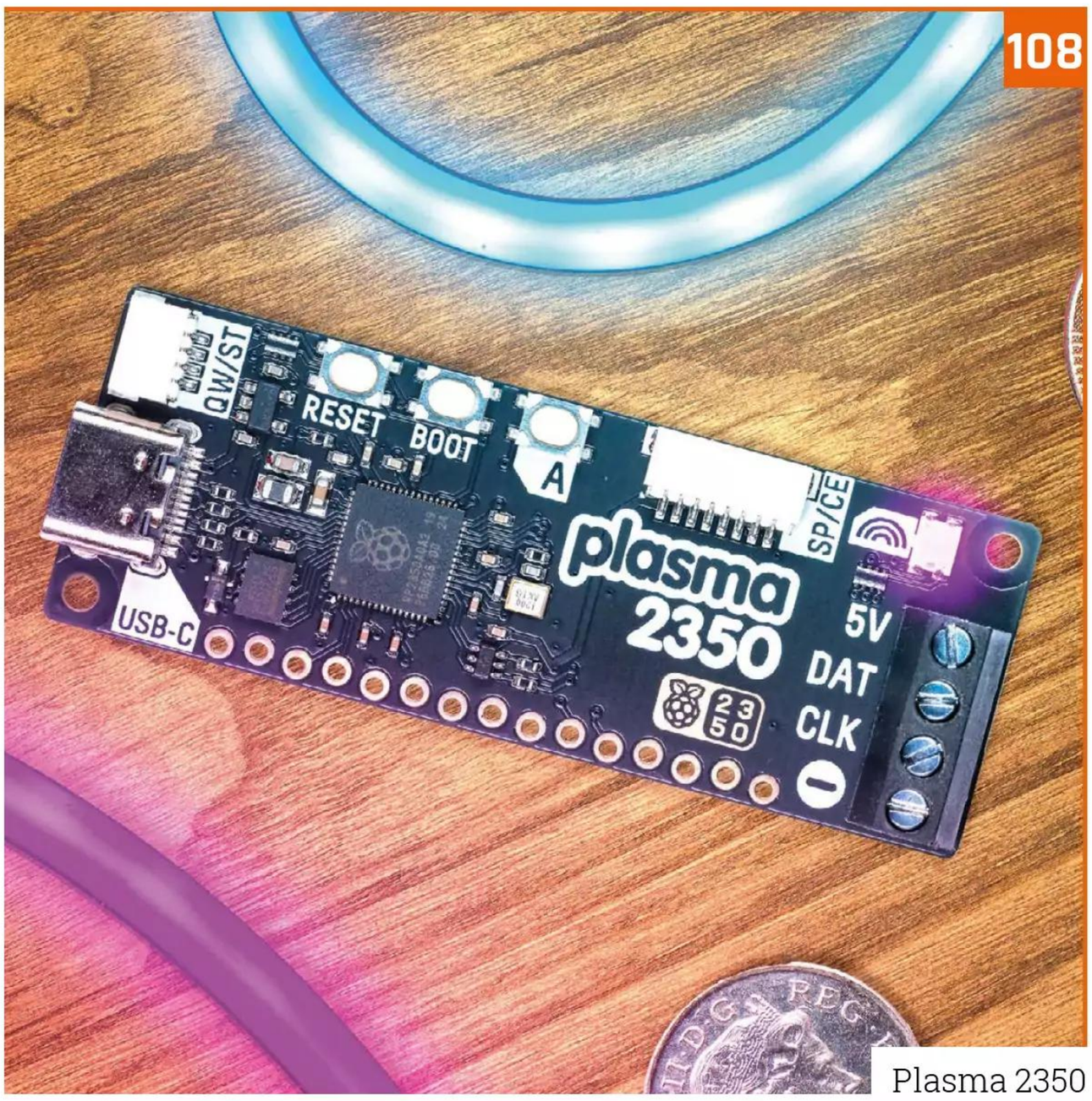
The Big Feature

98



Raspberry Pi audio upgrades

108



Plasma 2350

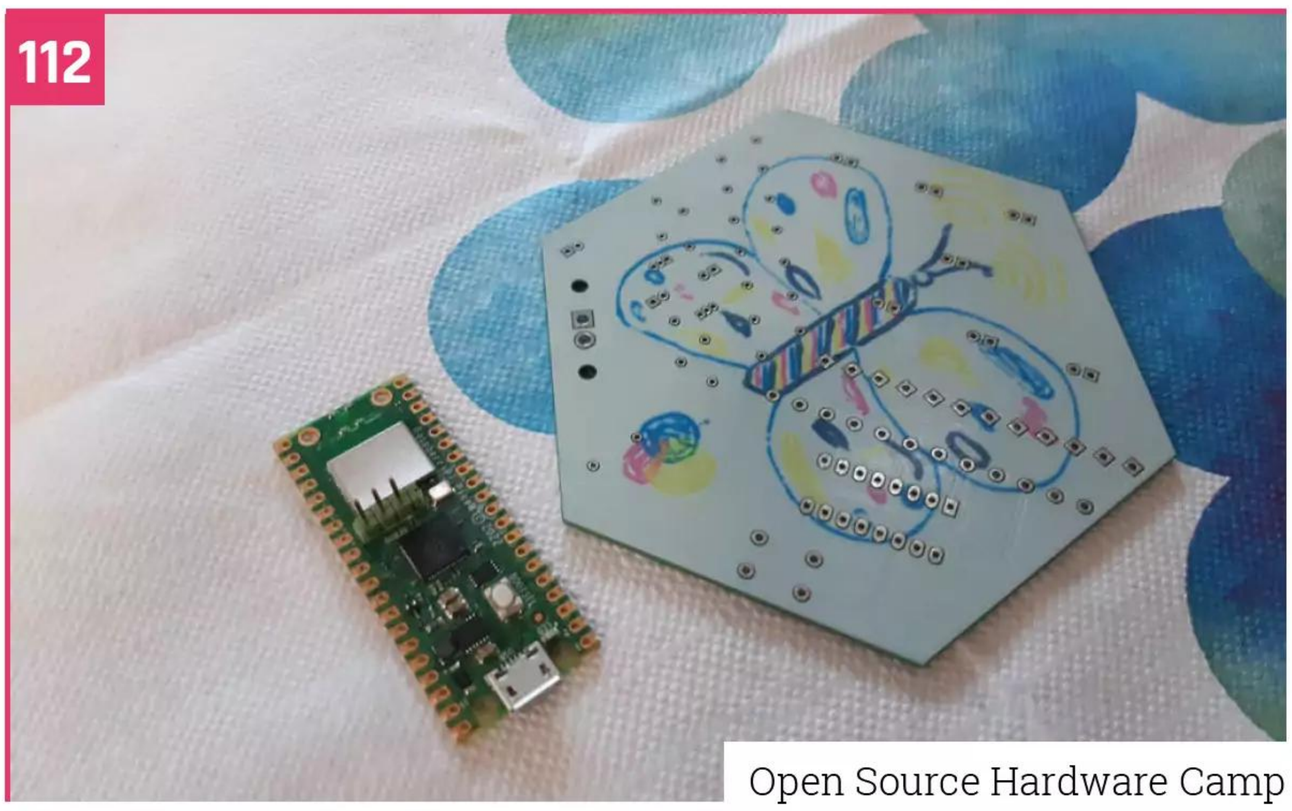
Reviews

106 CrowView Note
108 Plasma 2350
110 10 amazing Halloween projects

Community

112 Open Source Hardware Camp
116 Natalie Turner interview
118 This Month in Raspberry Pi

112



Open Source Hardware Camp

WIN 1 OF 5 MOTION **2350 PRO**

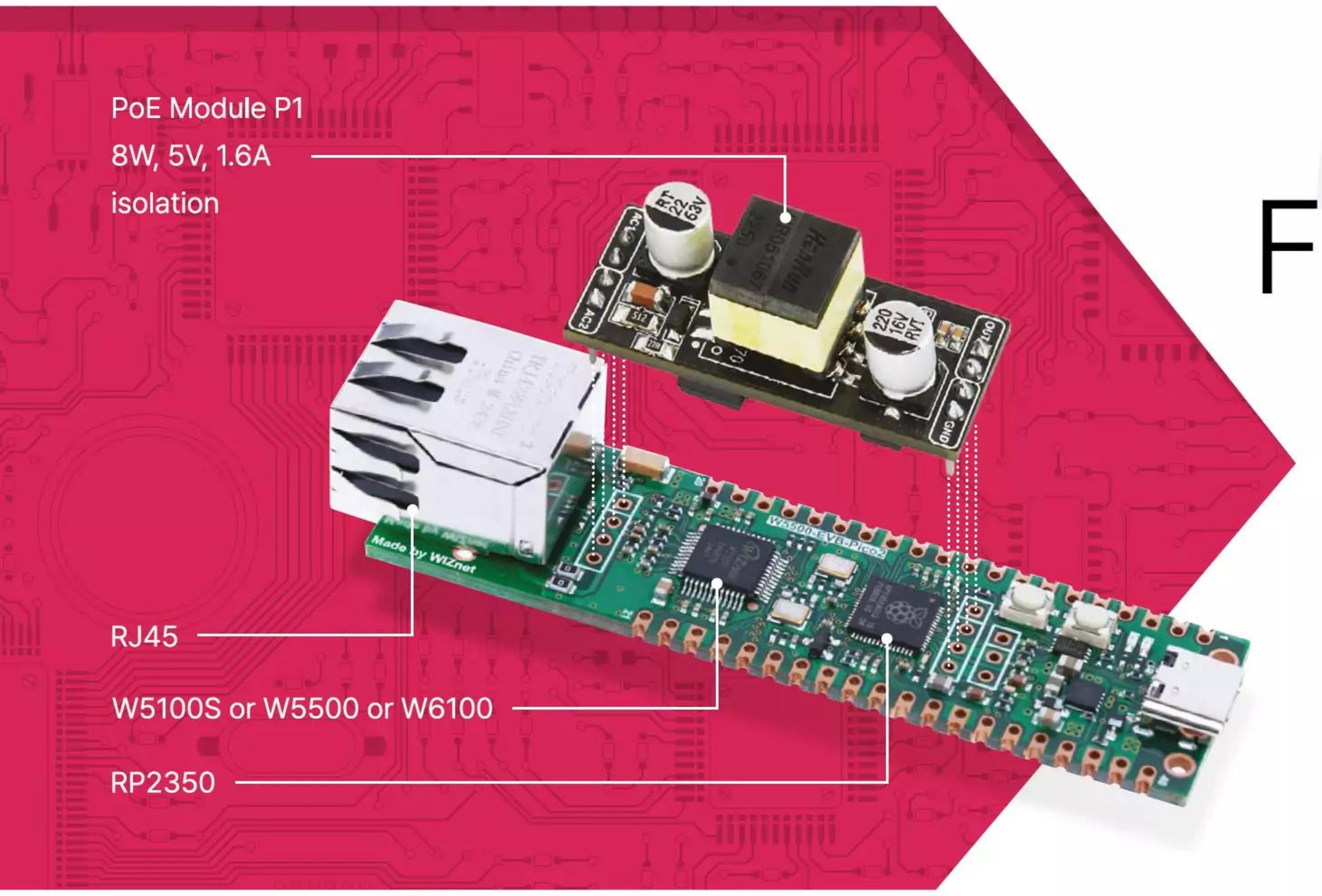
126

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

WIZnet

Ethernet connectivity solution
for **RP2350**

EVB-Pico2
Family boards



Features

Available in W5100S, W5500 or W6100 version

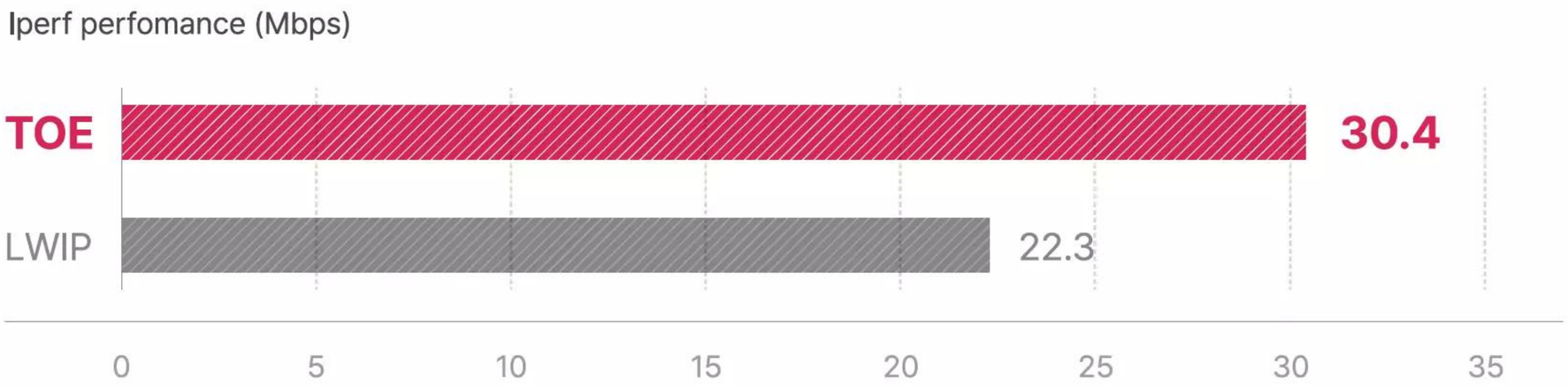
PoE enabled via add-on module

Identical pinout with Raspberry Pi Pico

USB Type C

Run button available

TOE vs LWIP comparison



Tests were done using
RP2350 (150Mhz) and
W6100 (SPI set to 37.5Mhz)

PoE Module specifications

WIZPoE - S1



IEEE802.3af compliant
Mode A(Endspan), Mode B(Midspan)
Wide input voltage range 40Vdc ~ 60Vdc
High DC/DC conversion efficiency

Non-Isolation
Internal build in 2 channel
bridge rectifiers
5V/8W Output

WIZPoE - P1



IEEE802.3af compliant
Mode A(Endspan), Mode B(Midspan)
Wide input voltage range 40Vdc ~ 60Vdc
High DC/DC conversion efficiency

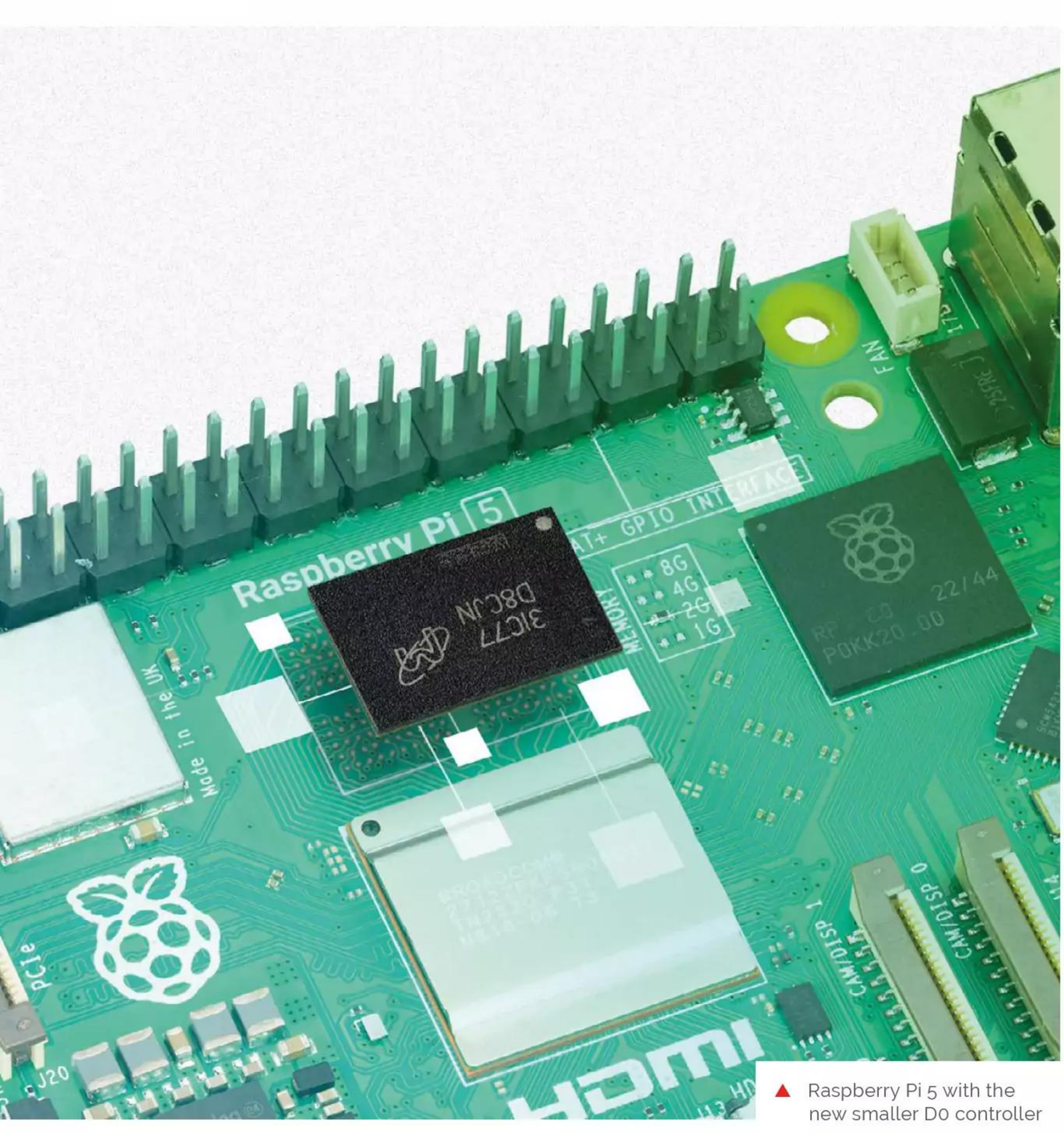
Isolation
Internal build in 2 channel
bridge rectifiers
5V/8W Output

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

More info

2GB Raspberry Pi 5 on sale now at $50

Welcome the 2GB version to the family. By **Eben Upton**



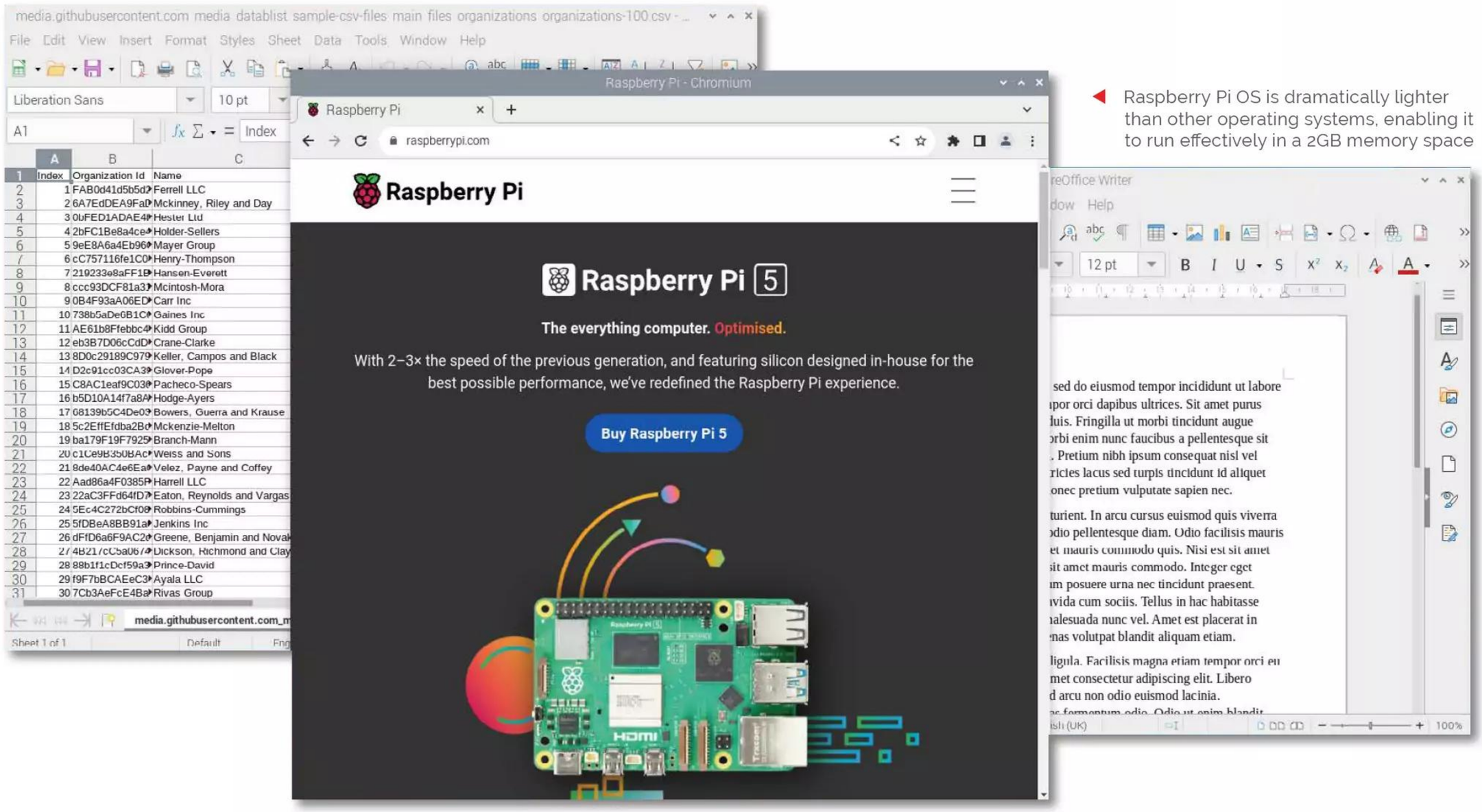
▲ Raspberry Pi 5 with the new smaller D0 controller

On 19 August, our flagship Raspberry Pi 5 family gained a new member. Priced at just $50/£47, the new 2GB variant continues our mission to bring high-performance general-purpose computing to the widest possible audience.

It's been a little over ten months since we launched Raspberry Pi 5. In many ways, this is the product that finally delivered on the original Raspberry Pi dream: an affordable general-purpose desktop computer, indistinguishable from a traditional PC for most users, and bundled with all the tools and collateral required for a beginner to go from "hello, world" to a career in engineering.

Raspberry Pi 5 is on the order of 150 times as powerful as the original Raspberry Pi that we launched back in 2012. Much of that performance increase comes from clever engineering, from the economies of scale that result from building millions of computers a year, and from the continued operation of Moore's Law. But as we've continued to reach for performance, some components of the design have inevitably become more expensive. Until now, the lowest-cost Raspberry Pi 5 was the 4GB variant, priced at $60/£56.

We're happy to announce the launch of the 2GB Raspberry Pi 5 (**magpi.cc/raspberrypi5**), built on a cost-optimised D0 stepping of the BCM2712 application processor, and priced at just $50/£47.



◀ Raspberry Pi OS is dramatically lighter than other operating systems, enabling it to run effectively in a 2GB memory space

New board, new chip

The 4GB and 8GB variants of Raspberry Pi 5 are built around two key chips: the RP1 I/O controller, developed here at Raspberry Pi and providing the interfacing capabilities of the platform; and BCM2712C1, a 16nm application processor built by our friends at Broadcom.

BCM2712C1 is a hugely complex and powerful device, with a quad-core Arm Cortex-A76 application processor running at 2.4GHz, and the latest iteration of the VideoCore multimedia platform. Alongside the features required to power a Raspberry Pi, it also contains functionality intended to serve other markets, which we don't need. This 'dark silicon' is permanently disabled in the chips we use, but takes up die space, and therefore adds cost.

> "Now show us what you can do with it!"

The new D0 stepping strips away all that unneeded functionality, leaving only the bits we need. From the perspective of a Raspberry Pi user, it is functionally identical to its predecessor: the same fast quad-core processor; the same multimedia capabilities; and the same PCI Express bus that has proven to be one of the most exciting features of the Raspberry Pi 5 platform. However, it is cheaper to make, and so is available to us at somewhat lower cost. And this, combined with the savings from halving the memory capacity, has allowed us to take $10 out of the cost of the finished product.

Saving memory, saving money

One of the many advantages of building our own operating system, Raspberry Pi OS, is that we get to focus on optimising resource usage. Historically, this allowed us to deliver a better user experience on devices with far less memory and processing power than today's flagship product. Retaining the ability to run the latest version of Raspberry Pi OS on those older products remains an important goal of our software work.

When running on modern hardware, the practical result has been a modern operating system with a dramatically lighter resource footprint than most general-purpose Linux distributions. So, while our most demanding users – who want to drive dual 4Kp60 displays, or open a hundred browser tabs, or compile complex software from source – will probably stick with the existing higher memory-capacity variants of Raspberry Pi 5, many of you will find that this new, lower-cost variant works perfectly well for your use cases.

You asked for a lower cost Raspberry Pi 5, so here it is: now show us what you can do with it!

30,000 badges and still no hack?

No one has managed to break the security on our new chip yet. If you're interested in collecting a bounty, read on.

By **Chris Boross**



▼ The RP2350-powered DEF CON badge

In early 2024, the stars aligned, and we got the honour of helping make a DEF CON badge. This is all thanks to our design partner friends at Entropic (**entropicengineering.com**) and the fantastic badge team at DEF CON (**defcon.org**) for trusting us to get silicon ready in time to build 30,000 badges. It was tight... but we made it happen!

The launch was amazing. We were able to give 30,000 DEF CON attendees zero-day access to our brand new chip, RP2350 (**magpi.cc/rp2350**), as well as a really cool gaming badge that runs a bare-metal GBA emulator made by our talented friend Dmitry Grinberg (**dmitry.gr**).

Double your fun!

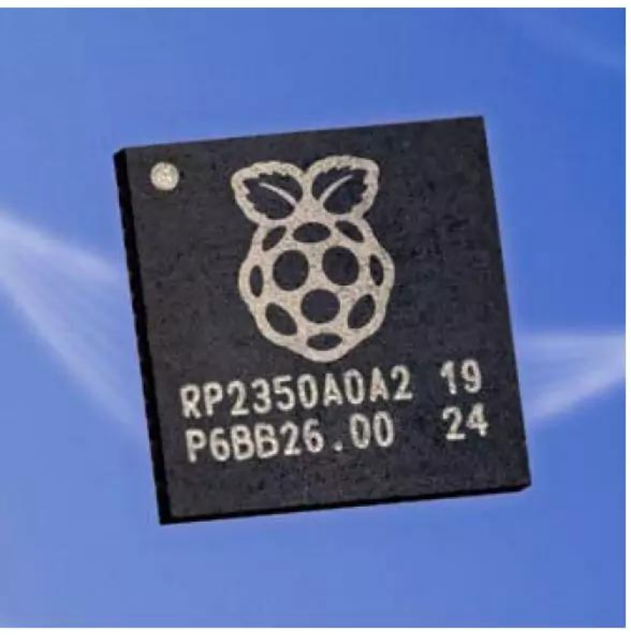
RP2350 is a dual-CPU, dual-architecture microcontroller. It has both Arm and RISC-V processors, allowing users to choose which CPUs they run code on. This gives CPU enthusiasts and people wanting to learn and experiment with RISC-V very affordable, easy-to-use access to this up-and-coming CPU architecture.

The RISC-V cores are called Hazard3, and they were designed by a very proficient member of the Raspberry Pi ASIC team called Luke Wren (**magpi.cc/lukewren**). Luke designed this in his spare time, documented the process, and then published (**magpi.cc/hazard3**) the designs with a permissive licence.



▶ Since DEF CON, Graham has kept on tinkering with *Doom* on the badges, enabling multiplayer by making them talk to each other via the SAO connector (SAO = sh!tty add-on – the de facto standard in badge add-on silliness)

> "I have never worked with a company that was so open and receptive about chip security"



◀ Hack me! (**magpi.cc/rp2350hack**)

The DEF CON badge team gave us another huge honour by inviting Luke to be part of their badge talk, where we got to address the audience about RP2350 and its Hazard3 cores. A big thank you to the badge team, and the DEF CON audience.

Does it Doom?

What kind of gaming device would it be if it didn't run *Doom* well? Our very own embedded software wizard, Graham Sanderson, got his cool RP2040/RP2350 *Doom* port up and running on the badge (with high frame rate) and we helped some attendees to program it onto their badges. Go here to see Graham's Doom port: **magpi.cc/rp2040doom**.

Challenge (still) accepted!

When we launched the chip at DEF CON, we issued a challenge to anyone with an RP2350 to see if they could hack around our security features and tell us the secret that has been programmed into the secure on-chip storage.

Currently, the security is still unbroken, and the $10,000 prize uncollected. The challenge was only due to run until September, but we've decided to goad the bounty hunters by doubling the prize money and extending the deadline to the end of the year. If you think you can hack it, be our guest (**magpi.cc/rp2350hack**).

This was the first official microcontroller bug bounty developed in partnership with the excellent humans over at **Hextree.io**, including their co-founder Thomas Roth (of Stacksmashing fame, **stacksmashing.net**). Thomas has said lots of lovely things about the "ready-to-glitch" chips we brought to the conference, but these were our favourites:

"Advanced hardware attacks such as fault-injection have moved from only being possible in a professional lab to being available to essentially anyone with some basic skill. RP2350 is the first microcontroller that reacts to this fact, integrates active countermeasures against hardware attacks, provides well designed secure-boot, and provides a bug-bounty for anyone that finds bugs in it."

"I have never worked with a company that was so open and receptive about chip security. At every point it was clear that Raspberry Pi wanted us to find issues so that they can fix them and be transparent about them. Allowing us to bring their chips ready to glitch to the world's biggest hacker conference attests to that!"

Roll credits

We are so thankful for the opportunity to collaborate with DEF CON on this badge, it was an absolute honour – we had an amazing time, and look forward to working with TEAM CATBALL again!

TEAM CATBALL:

- MAR WILLIAMS (**marwilliams.art**): Concept, design, and coordination
- BONNIE FINLEY (**magpi.cc/bonfin**): 3D modelling, game art and development
- CHRIS MALTBY (**chrismaltby.com**): Plug-ins and game development
- NUTMEG ANNE: Game development
- WILL TUTTLE: Concept and narrative collab
- ADA R-W: Cool dragon
- RASPBERRY PI: Hardware
- ENTROPIC (**entropicengineering.com**) W/ DMITRY GR (**dmitry.gr**): Hardware development, firmware, emulator
- JOE GRAND (**grandideastudio.com**): Hardware support and production testing
- LEGION303: Music and sounds
- ICSN (**icsngroup.com**): Manufacturing

We'd also like to thank Trevor Stevado and the rest of the Embedded Systems Village team (**embeddedvillage.org**). Thank you for making our first DEF CON so enjoyable and successful.

▼ Keep an eye on DEF CON's YouTube channel for a video of Luke's presentation

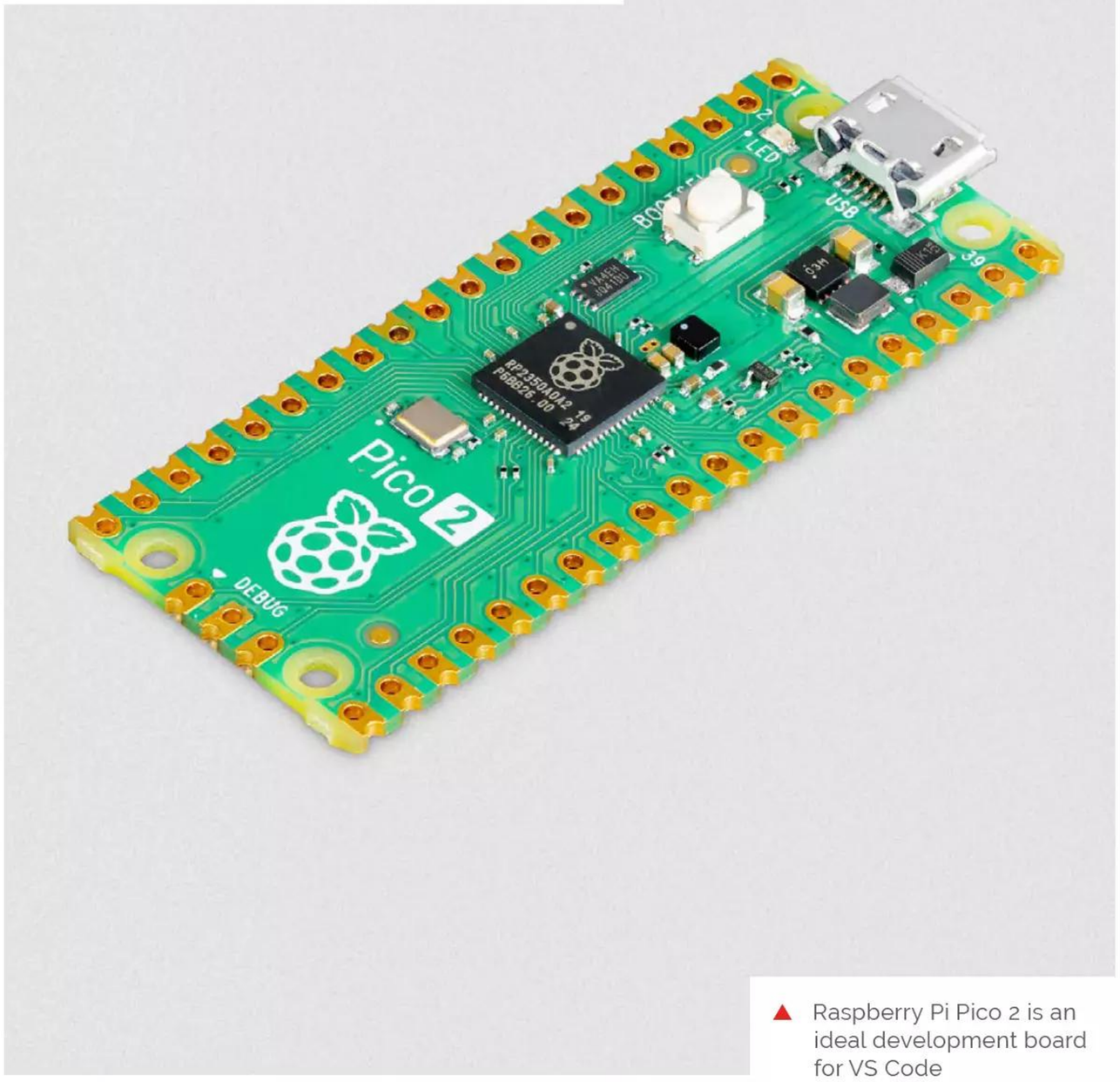


Boost your Pico projects with the new Pico VSCode Extension

Introducing the public beta of the Raspberry Pi Pico Visual Studio Code Extension, which makes it easier than ever to develop projects for Raspberry Pi Pico-series devices.

By **Paul Oberosler, Engineering Intern – Raspberry Pi**

> You can dive into development quickly and efficiently



▲ Raspberry Pi Pico 2 is an ideal development board for VS Code

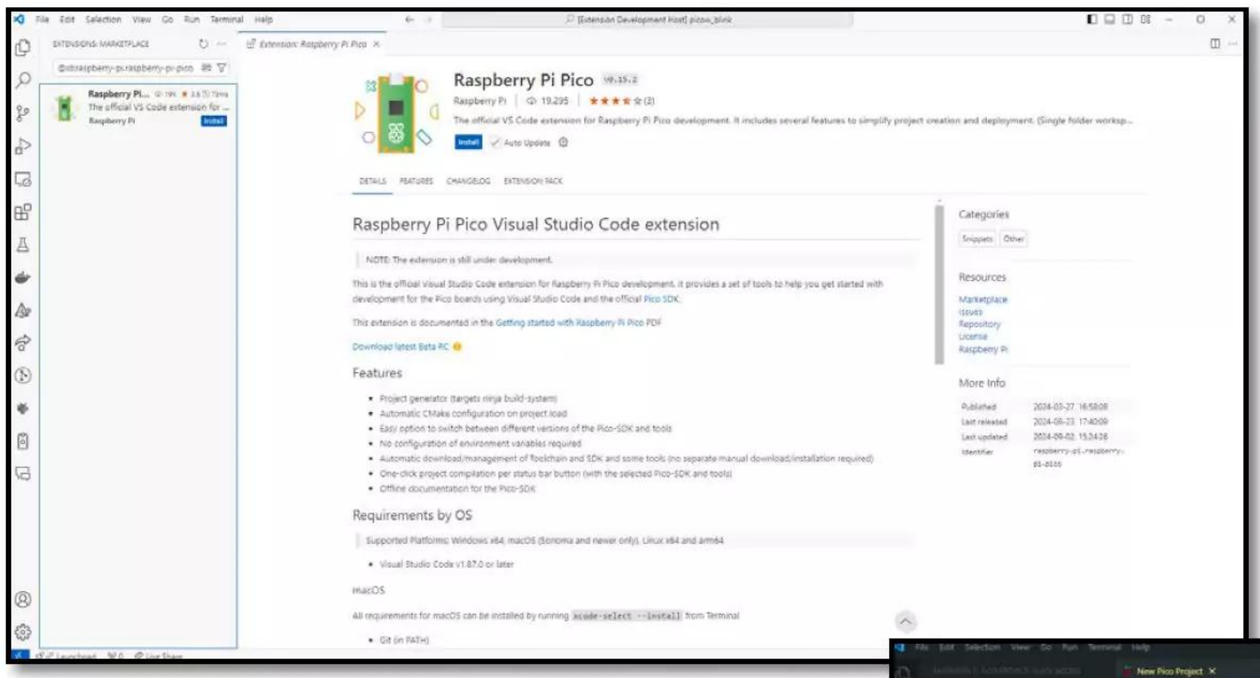
A few months back, we quietly dropped the Pico VSCode project on GitHub. It didn't take long before the feedback started pouring in. Since then, we've been listening and tweaking. Now, we're excited to officially unveil the public beta of the Raspberry Pi Pico Visual Studio Code Extension.

So, what exactly is Pico VSCode? It's a Microsoft Visual Studio Code extension designed to make your life easier when creating, developing, and debugging projects for Raspberry Pi Pico-series boards. Whether you're a total beginner or a seasoned pro, this tool is here to help you dive into Pico development with confidence and ease.

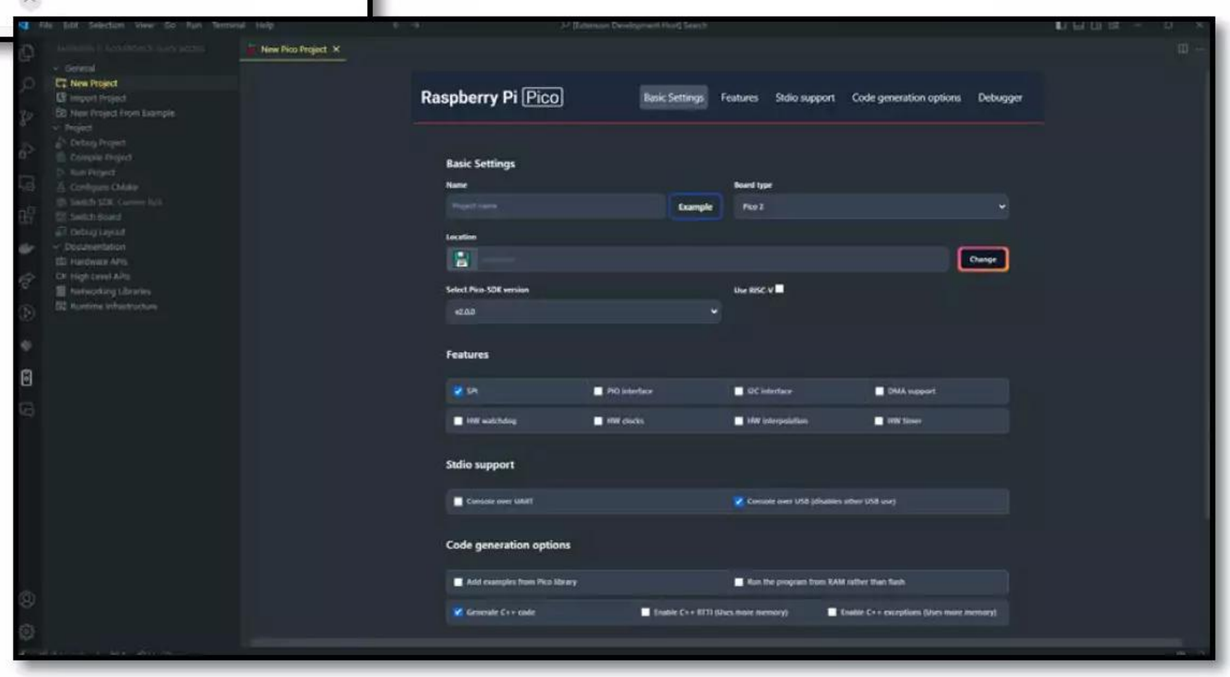
If you've ever tried to set up an embedded development environment, you know it's no small feat. Beginners often find themselves tangled up in the complexities of build systems, SDKs, and toolchains. And let's not even get started on cross-compilation; developing on one machine to run code on another introduces a whole new set of challenges.

Getting all the right configurations and installations in place can be intimidating for everyone, not just those new to the game. Even experienced developers can find themselves tangled in frustrating setup processes that eat into valuable development time.

That's why we created the Pico Visual Studio Code extension: a user-friendly tool that simplifies the entire development process. We wanted to offer



- Add Pico support to your VS Code environment using the extension
- The Basic Settings enables you to set up the basic features and STDIO of your project



something that takes the guesswork out of setting up your environment, so you can start coding in an interface you're already familiar with – Visual Studio Code – as quickly as possible.

With Pico VS Code, you won't have to worry about nitty-gritty details that trip up newcomers and sometimes stymie veterans. Instead, you'll be able to focus on what really matters: bringing your Raspberry Pi Pico projects to life. Whether you're working on your first blinking LED or a more complex project, Pico VS Code is there to help you get started and keep you moving forward.

How do I get Pico VSCode?

To get started with the Pico VSCode extension, you'll need to ensure that your development environment meets a few basic requirements. The extension is compatible with various platforms, including Raspberry Pi OS, Windows, macOS and Linux, each with its own set of prerequisites.

Take a look at our Getting started with Pico-series tutorial on page 40.

It also supports MicroPython

For beginners or developers who want to get their projects up and running on a Pico as quickly as possible, MicroPython is an excellent choice. It is a lean and efficient implementation of the Python 3 programming language, specifically designed to run on microcontrollers and in constrained environments. It includes a small subset of the Python standard library, making it a powerful yet lightweight option for embedded development.

To create a Pico project using MicroPython instead of C/C++, select New MicroPython Project. You can find this button either in our Quick Access panel, located in your sidebar or by running the New Pico Project command and selecting MicroPython as language. This will launch the familiar project creation wizard, now tailored for setting up a MicroPython project. Choose the location for your project folder and set a name for your project. When you click Create, the extension generates the new project and opens it for you, just like with a C/C++ project. But instead of using C/C++, your new project uses the MicroPico extension to run your code on the board and manage project configurations.

With MicroPython, you can quickly start prototyping and experimenting with your Raspberry Pi Pico-series device, making it an ideal option for both newcomers and seasoned developers alike.

Next steps

For more detailed information on using the Pico VSCode extension, including a comprehensive list of settings and additional guidance, visit our GitHub project page (**magpi.cc/vscodegit**). It's a great resource for getting the most out of the extension.

If you're new to developing Pico projects, don't forget to check out the Getting Started guide we mentioned earlier – it's packed with helpful tips to get you up and running.

If you're looking to create a project that makes use of the advanced features of Pico-series devices – such as I2C, PIO, or enabling stdio support – be sure to explore the New C/C++ Project interface. This tool allows you to customise your project setup to suit your needs, so you can dive into development quickly and efficiently.

LEGO

Card Shuffler

Who needs to laboriously shuffle their own deck when Raspberry Pi can do it for you?, asks **Rosie Hattersley**



MAKER

Louis Wood

Cambridge engineering student Louis spent an incredible summer at Raspberry Pi HQ building all sorts of projects in the dedicated maker space.

magpi.cc/pixieclock

Maker and engineering undergraduate Louis Wood first encountered Raspberry Pi while looking for a low-cost microcontroller that could be programmed with Python for an A-level project. Inspiring plenty of envy, he's just spent six whole weeks ensconced in Raspberry Pi HQ's very own maker space building a range of Raspberry Pi projects including the LEGO Card Shuffler. Basing it around the LEGO Build HAT (**magpi.cc/buildhat**) helped him evolve and improve upon a design he and his Queen's College Cambridge university friends Lucas Hoffman and Emily Wang devised. The card shuffler idea was their response to a design and build challenge based around a LEGO NXT system that was to demonstrate an aspect of engineering science. The dual-motor design was in need of some reworking, which Louis undertook while working as an intern at Raspberry Pi Towers alongside maker in residence Toby Roberts.

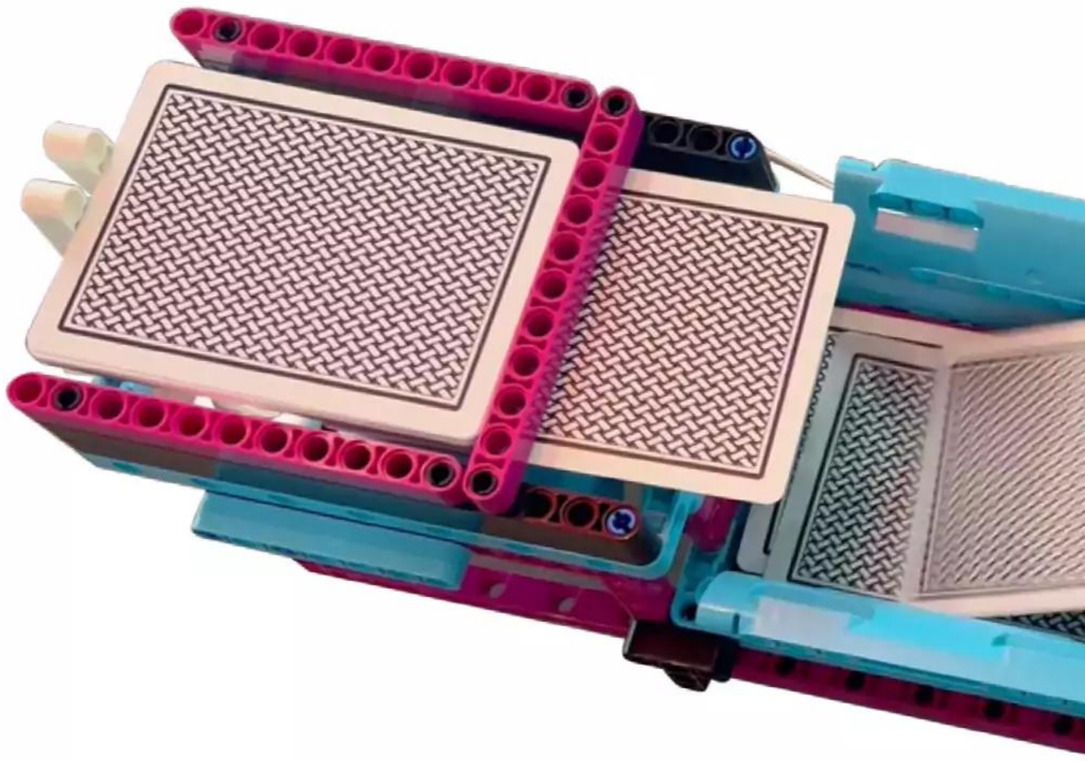
Quirky and cool

Louis used a LEGO Spike education kit with Raspberry Pi's LEGO Build HAT to create a simpler but more robust design. The kit includes cycle motors which he attached directly to the Build HAT's four connectors. "The Build HAT made it pretty easy to pick up all the motors and plug them in." He then programmed Raspberry Pi 4 over SSH "which made it easy to tweak code".

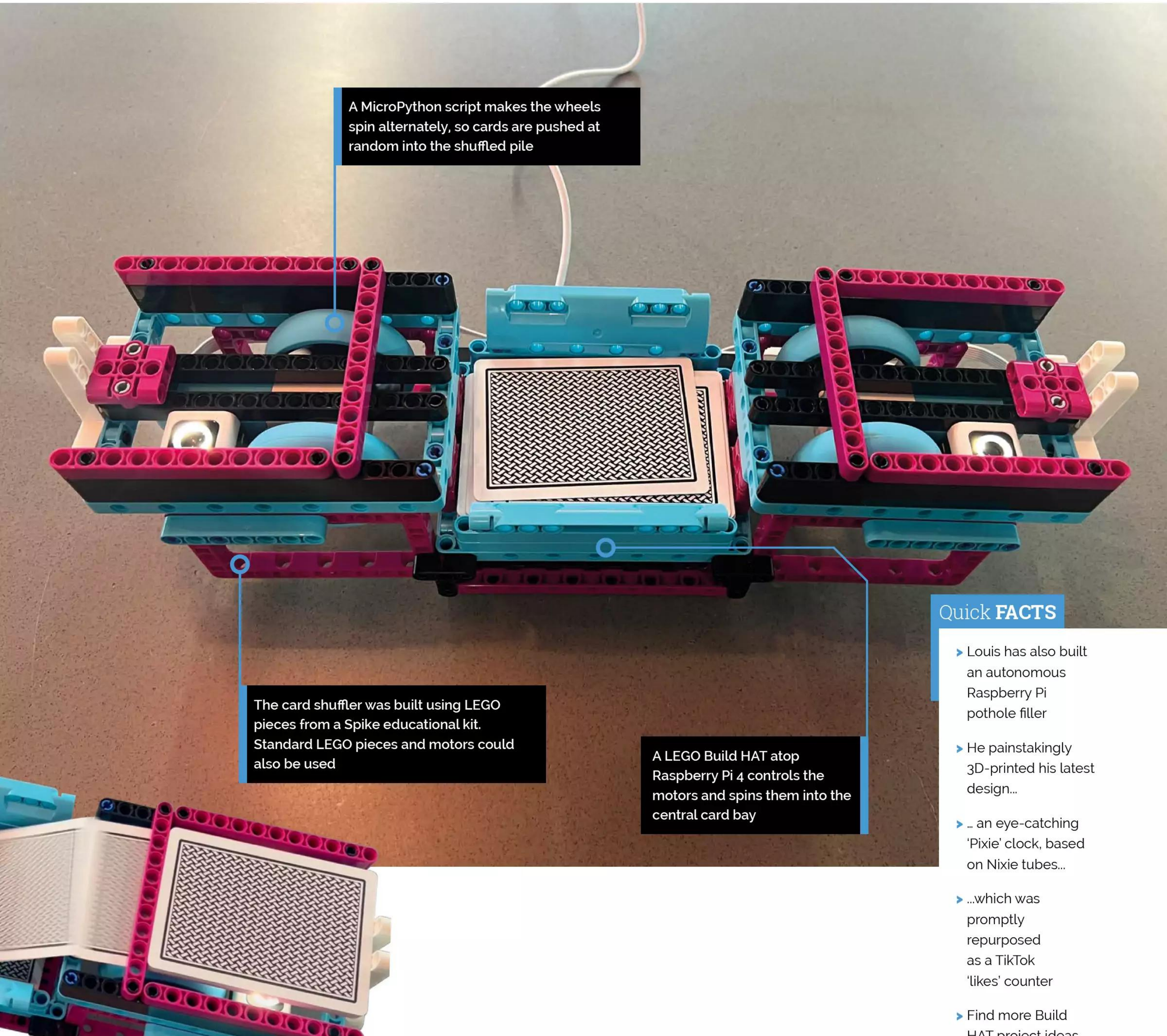
The MicroPython code produces either a one or a zero and spins either the left or right motor accordingly: "When the motor turns on, the wheel spins a few cards into the middle." The motors run on a loop, each powering on for a second or two, pushing cards from each side and randomly shuffling them into a central pile until the Build HAT colour sensors detect the black base of either card bay. The card shuffler then skips that side and only runs the opposite motor for a while to clear the rest of the cards. Once it notices it's done shuffling, it stops."

The build took a couple of hours, then Louis spent a similar time coding and tweaking the build. "The hardest thing was making it so it doesn't just spit out the whole side [of cards] at once," he says.

His simple but effective barrier is positioned such that only a single card at a time can (usually) be shuffled along by the motor. The setup



▶ The LEGO Card Shuffler in all its glory



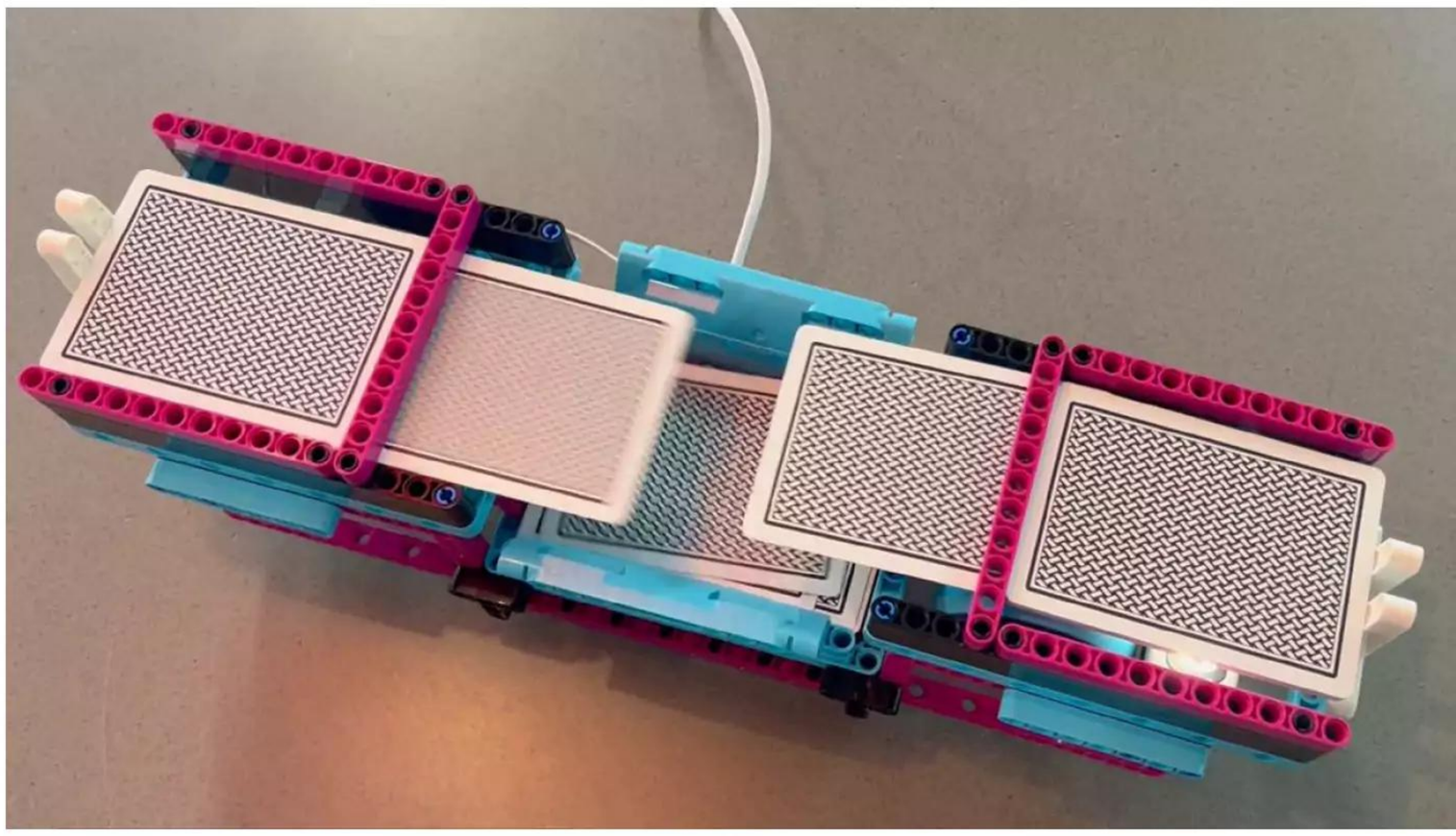
A MicroPython script makes the wheels spin alternately, so cards are pushed at random into the shuffled pile

The card shuffler was built using LEGO pieces from a Spike educational kit. Standard LEGO pieces and motors could also be used

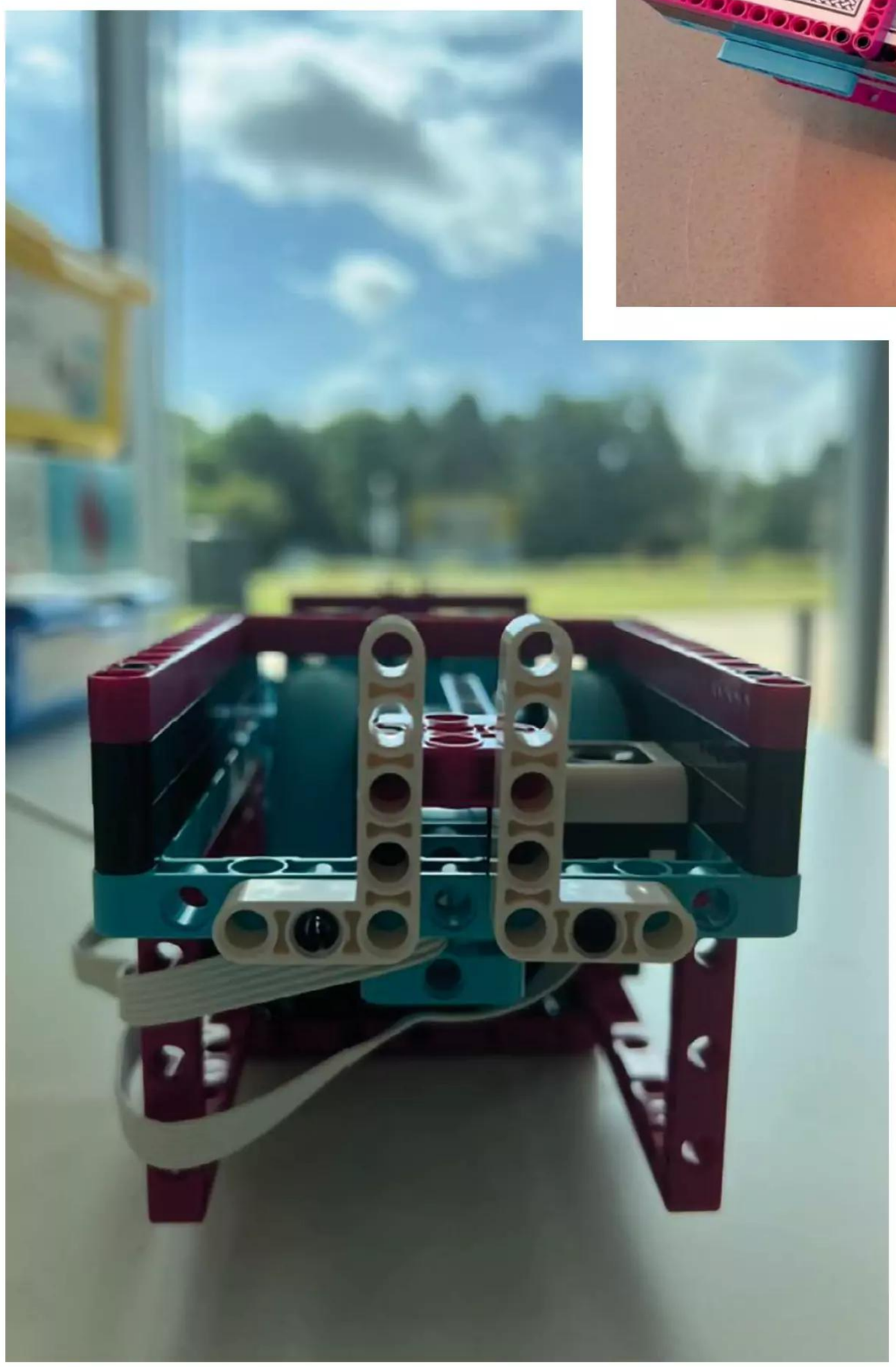
A LEGO Build HAT atop Raspberry Pi 4 controls the motors and spins them into the central card bay

Quick **FACTS**

- Louis has also built an autonomous Raspberry Pi pothole filler
- He painstakingly 3D-printed his latest design...
- ... an eye-catching 'Pixie' clock, based on Nixie tubes...
- ...which was promptly repurposed as a TikTok 'likes' counter
- Find more Build HAT project ideas at **magpi.cc/buildhproj**



- The card shuffler detects whether there are any cards in each bay
- LEGO Card Shuffler on show at Raspberry Pi's HQ



doesn't always work flawlessly, occasionally requiring the user to deftly flick a card back into place, but Louis aims to improve the design by moving apart the two card holding sides to prevent blockages".

The Build HAT came into its own thanks to the colour sensor which Raspberry Pi 4 used to detect whether there were still cards awaiting shuffling. The white background of the cards contrasted with the black base of the crates he'd created, which was visible only when the stack of cards was depleted. Other card decks, such as Uno ones, which usually have a black background, could be shuffled too, as long as the card holder base colour was changed.

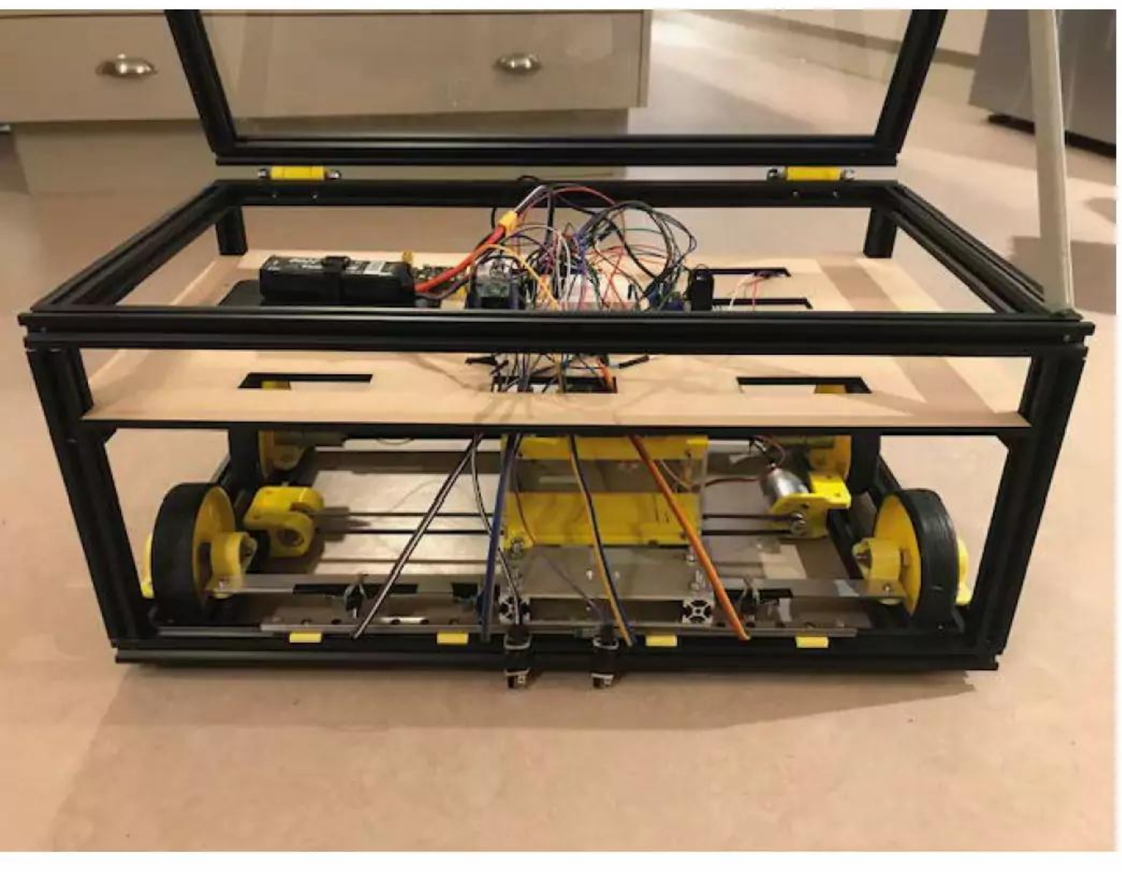
Makers gonna make

Two weeks in to his internship, Louis had already created and written about a 'Pixie' tube clock (**magpi.cc/pixieclock**) and had been building "sort of a Raspberry Pi mount and cooling system for one of the engineers upstairs, so you can sort of be running eight Raspberry Pis at the same time, fans and an enclosure," as well as a remote control based on the brand-new Pico 2 (**magpi.cc/pico2**)

Given this prodigious rate of design, we asked whether an engineering career or one as a maker is in his future, Louis confirms, "I'd like to be a maker, but I think it's quite hard to be a maker YouTuber takes a lot of work and time, I think, probably a bit risky".



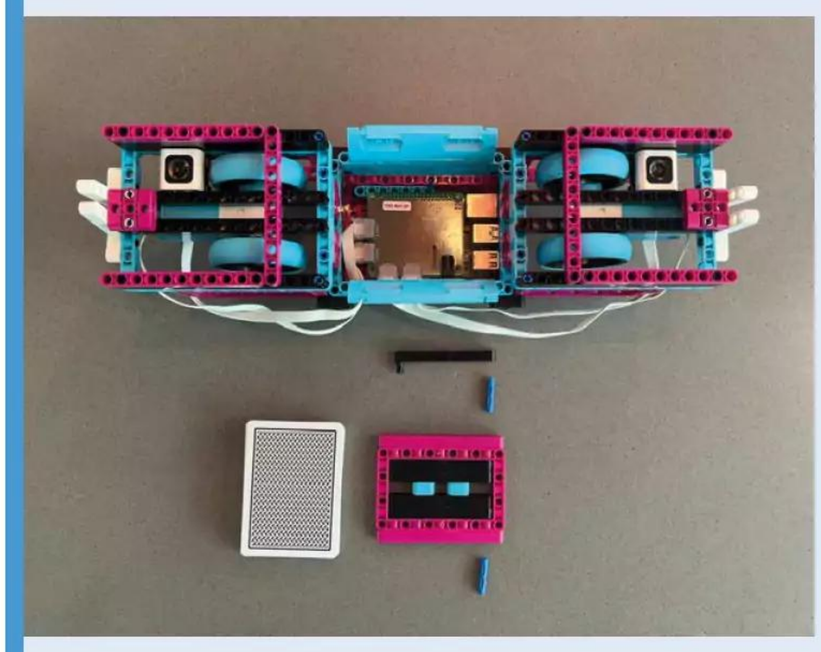
> Two motors take it in turns to run for a second or two, pushing cards from each side and randomly shuffling them into a central pile



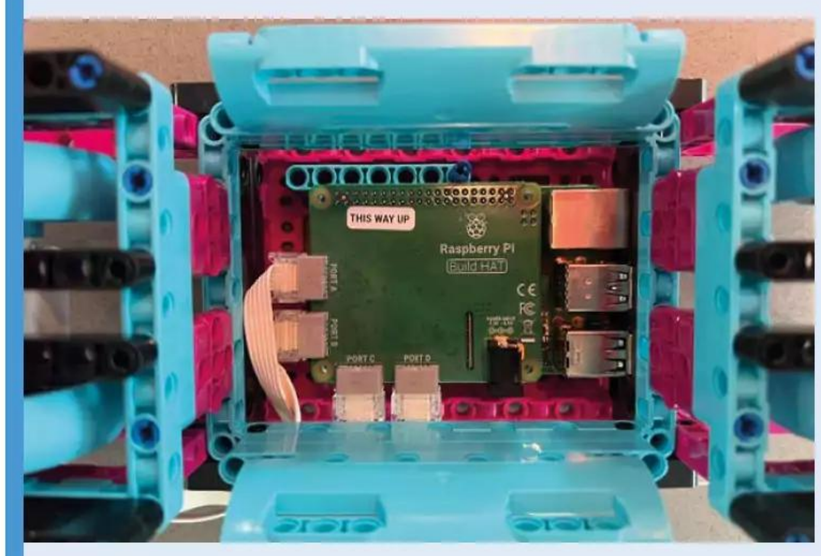
▲ Louis' stunning Pixie clock, designed and 3D-printed in our maker space

◀ Louis' first Raspberry Pi project was an autonomous pothole filler

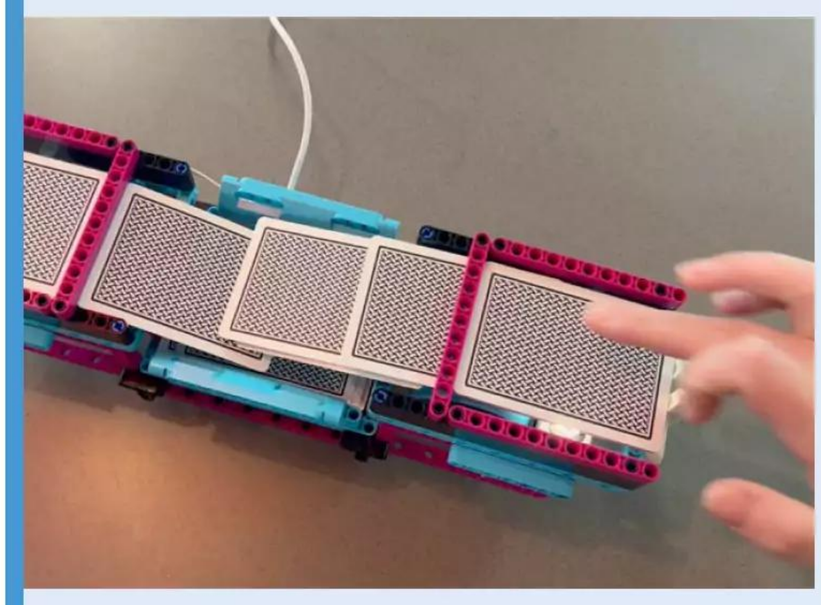
In a spin



01 The card shuffler was built using LEGO pieces from a Spike educational kit, but standard LEGO pieces and different motors could also be used.



02 Affix the Build HAT to Raspberry Pi, and attach the motors to the ports on the LEGO Build HAT. Attach your assembled card shuffler to a mains or portable power supply.



03 Use a second device to set up Raspberry Pi for headless control using SSH. Place your playing cards face down in the two card bays, then run the MicroPython script to start shuffling them.

Portable Pi 84

With a mechanical keyboard powered by Raspberry Pi Pico and with Raspberry Pi 4 at its heart, **David Crookes** has fallen in love



MAKER

Michael Mayer

Born in Austria, Michael was 12 years-old when he began programming in BASIC on a pocket computer. Today he works as a software engineer.

magpi.cc/pi84portable

As impressive looking as modern laptops are with their heights usually measured in millimetres, the chunkier devices of yesteryear still have the power to turn heads. Their solid builds gave the impression that you could drop them off a cliff and still manage to use them to write about the incident later. There was also something rather attractive about their no-frills, plasticky nature.

It is to those bygone laptops that Michael Mayer has turned when creating his own portable classic. Named the Portable Pi 84 – by virtue of being driven by a Raspberry Pi 4 computer – it takes inspiration from the machines that had very clear central hinges. It also sports a fun, red colouring and has a widescreen display that is more than large enough to be productive.

This little device includes a mechanical keyboard that, Michael says: was his primary motivation for making his retro marvel. Having first cut his teeth learning BASIC on a Sharp PC-1260, he quickly became a fan of portable computers and snapped up many more. But when, in recent years, he sought to revive his interest in such machines, he said he's been left disappointed.

"I have tried Raspberry Pi and Linux-compatible portables and laptops but I've never been really satisfied with their keyboards," he says, name-checking the Pocket C.H.I.P, Devterm and uConsole projects. "Even at work I used a mechanical keyboard with the company laptop so I started to search for smaller keyboards and portable Raspberry Pi projects. I then found ZeroWriter."

Key to success

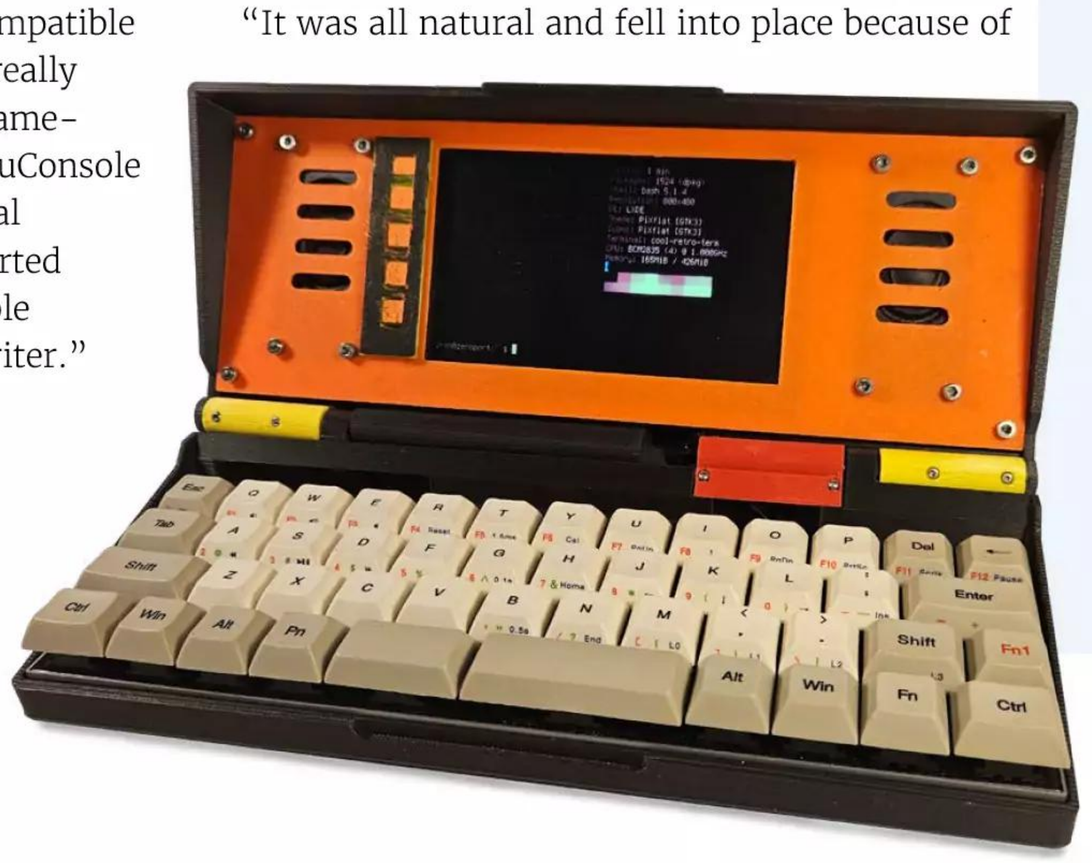
ZeroWriter is an open e-ink typewriter with a Raspberry Pi Zero 2 W board at its heart. Michael decided to create one for himself, and ordered the Vortex Core 40% keyboard while swapping out the WaveShare 4.2-inch e-paper display with a Waveshare five-inch monitor. "As there was still space, I added a speaker to the side," he adds. "But the project was limited in terms of the Python audio effects development I wanted to pursue."

Michael decided to take the project to another level, hence the use of Raspberry Pi 4. "With the experience of my first build, it was clear that I should go bigger and I also searched for another keyboard," he says. Looking on **pcbway.com**, he found the Happy Keyboard, a 47-key, 40% ortholinear mechanical keyboard that uses a Raspberry Pi Pico development board running KMK firmware. It formed the basis of the design for the entire project.

"I started with the keyboard because it was the single largest part of the build," he explains, adding that the next step was to find a display. "I searched for a screen with vertical resolution bigger than 400 or 480, and found a 9.3-inch display by Waveshare with a funky resolution of 1600×600 which fitted perfectly to the 40% keyboard." This setup dictated the design of the project's case, and it helped that every Waveshare screen comes with a CAD file and audio amplifier.

"It was all natural and fell into place because of





▶ Michael originally sought to create his own version of the ZeroWriter project incorporating a Raspberry Pi Zero 2 but felt it was too limited for what he wanted to do

The 3D-printed case was designed in FreeCAD to the fit the keyboard but, happily, the screen was almost the same size too

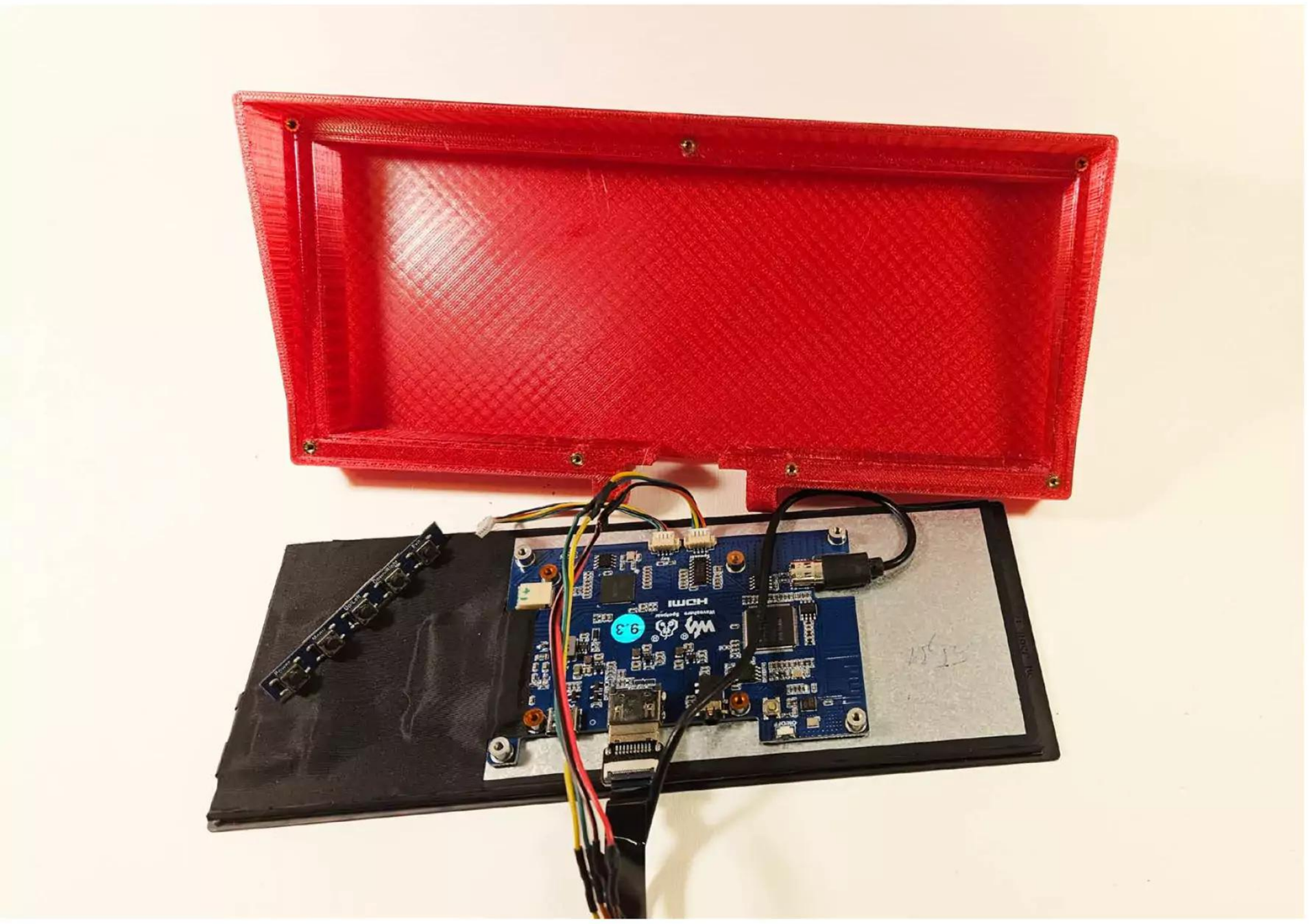


The project's 9.3-inch Waveshare touchscreen is actually inserted upside down so that the HDMI cable can run from the bottom of the screen

The mechanical keyboard is wired to the underside of the Raspberry Pi 4 computer

Quick **FACTS**

- The computer is based around Raspberry Pi 4
- It uses a hand-wired mechanical 40% keyboard
- There's a button for checking the battery life
- The device can be charged via USB when turned on
- The case files are available on Printables



The screen makes use of the touch input to deliver power. The real power USB port is not used. This allows the screen to fit into the case

the size of the components and the requirement that the keyboard should be as low as possible," Michael says. "Somehow I loved the wide display more than a normal 16:9 display, so this may be the reason for the centred screen."

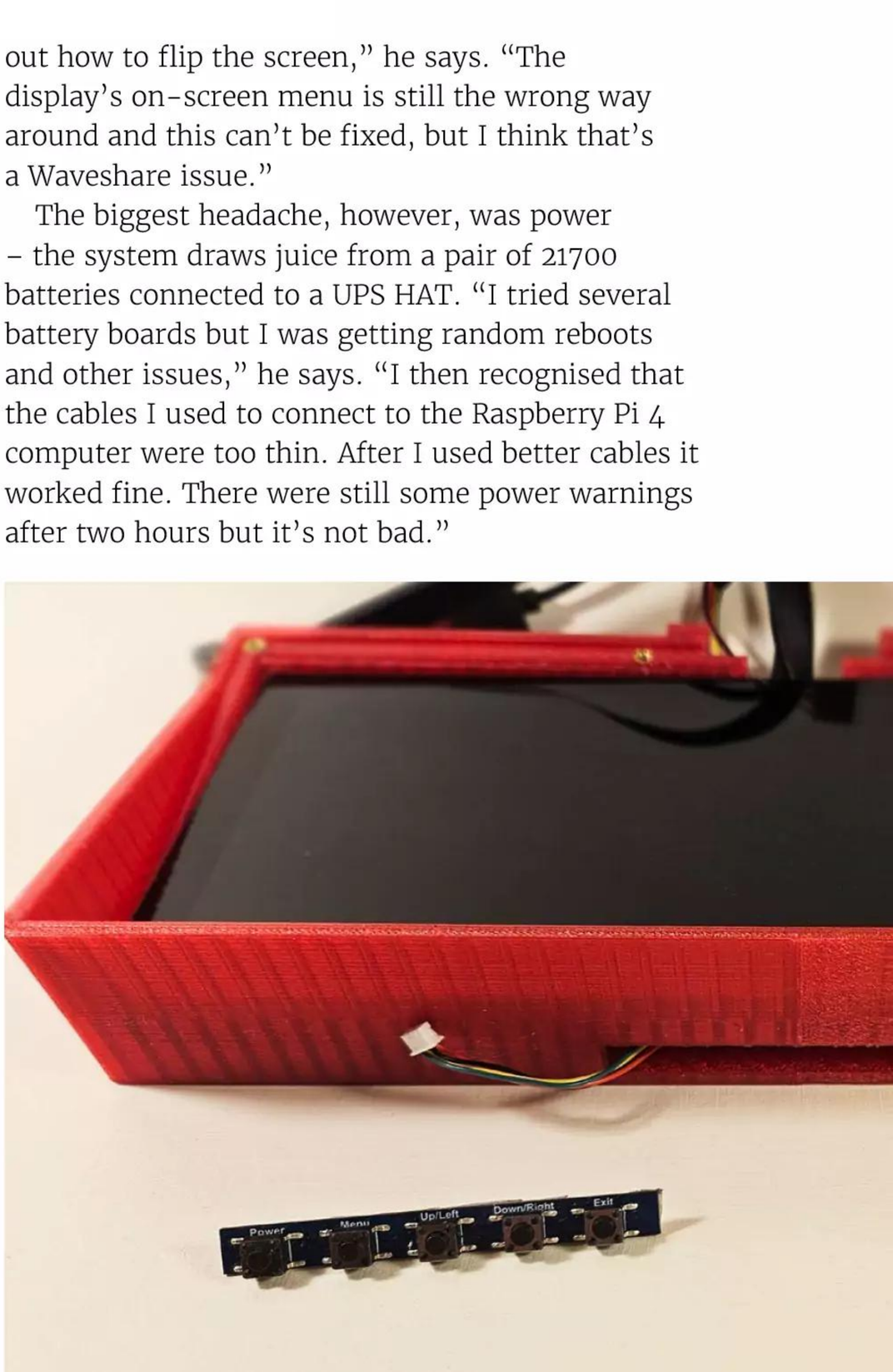
A great case

Other design considerations needed to be considered. "It was clear in the beginning that the Raspberry Pi 4 and the battery would have to sit behind the keyboard," Michael says, having decided to place them in a sizable rear compartment. The Raspberry Pi Pico development board also needed to be moved because there wasn't sufficient room on the main keyboard PCB.

In extending the back section by up to two centimetres and increasing its height to match that of the screen and keyboard, Michael found space to fit audio speakers. He worked on ways of being able to use the USB ports, too, by creating insert panels on the back of the case for flexibility. "Instead of having to print an entire case, taking tens of hours, every time I decided I'd want to reconfigure the ports, I could just print new panels, which takes about an hour," he explains.

Even so, there were other issues to contend with. Michael had trouble with the HDMI connection being on top of the screen. "I had to rotate it, and it took me some time to figure out how to flip the screen," he says. "The display's on-screen menu is still the wrong way around and this can't be fixed, but I think that's a Waveshare issue."

The biggest headache, however, was power – the system draws juice from a pair of 21700 batteries connected to a UPS HAT. "I tried several battery boards but I was getting random reboots and other issues," he says. "I then recognised that the cables I used to connect to the Raspberry Pi 4 computer were too thin. After I used better cables it worked fine. There were still some power warnings after two hours but it's not bad."



The screen buttons are fixed to the top of the case allowing for very easy access. They are recessed to prevent accidental pressing



▲ The hinges are fastened using M4 screws. Although Michael is seeking to improve the hinge design, it allows the display to rest nicely against the machine's trunk

Powering up

Battery issues are why he decided against using a Raspberry Pi 5 board, for now at least. "If I could find a battery able to power it, I'd use Raspberry Pi 5," he says. Michael also wants to be able to move the Raspberry Pi computer away from the side and make a side panel. "This would make it possible to use more single-board computers in the future, and help to connect the screen and keyboard to Raspberry Pi internally," he says.

> "If I could find a battery able to power it, I'd use Raspberry Pi 5"

But he's more than happy with the result so far and he says it's a build that has fulfilled his requirements. "It's also been a good way to learn FreeCAD which was fun most of the time – there was a lot of cursing before understanding, though." He uses the device for programming and watching streams and finds it to be very convenient. "It's easy to fetch and can be placed on the couch or kitchen table without taking up much room," he says. And that, in a nutshell, is what a portable computer should be like.



Creating a classic



01 Unlike the laptops that were available in the 1980s and 1990s, Michael's contemporary version packs a real punch thanks to the use of a Raspberry Pi 4 Model B computer. He is considering making use of Raspberry Pi 5 in the future for greater power.



02 The ortholinear mechanical keyboard has 47 keys and it uses the Raspberry Pi Pico W running KMK firmware powered by CircuitPython. It's possible to fit any keycap or switch to this keyboard, but Michael bought an inexpensive set of blanks from Amazon.



03 Removable panels at the back of the device allow the ports to be easily reconfigured without the need for a whole new case. The ports include a power button, an audio stereo jack, a battery charging port and a second HDMI-out, all within perfect reach of the user.

VespAI

AI models are adept at distinguishing one winged creature from another. **Rosie Hattersley** goes beyond the buzz



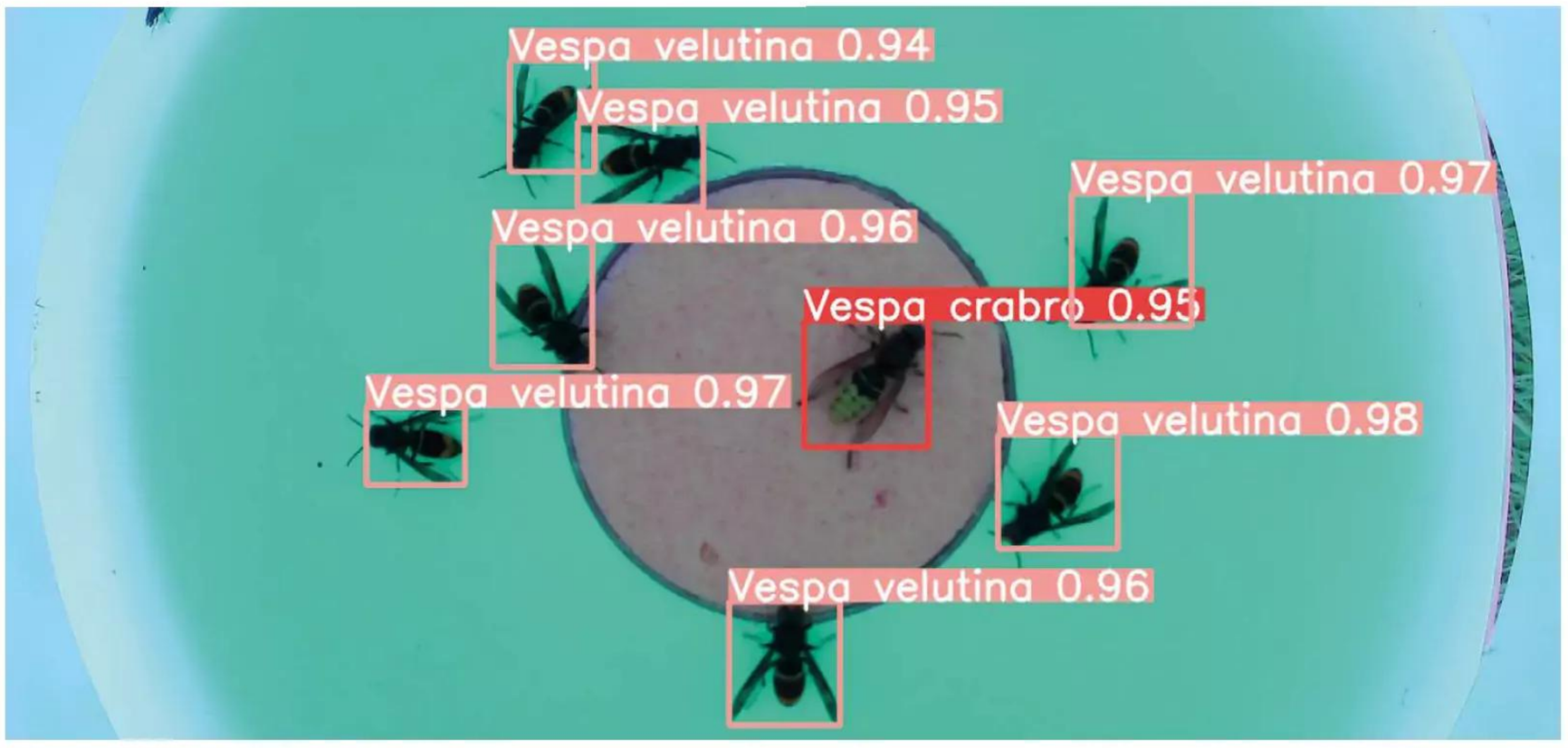
Fun fact that might get you a point in the local pub quiz: Vespa, Piaggio's iconic scooter, is Italian for wasp, which its buzzing engine sounds a bit like. Less fun fact: nature's counterpart to the speedy two-wheeler has an aggressive variant that has been seen in increasing numbers across western Europe and which is a direct threat to bees (**magpi.cc/beeshornet**) which are one of their key food sources. Bees are great for biodiversity; Asian hornets (the largest type of eusocial wasp) are not. But it's only particular hornet species that pose such a threat. Most citizen reports of Asian hornets are native species and a key issue is ensuring that existing hornet species are not being destroyed on this mistaken assumption. To combat misinformation and alarm at the so-called 'killer' hornet (itself a subset of wasp), academics at the University of Exeter have developed a VespAI detector that presents a positive identification system showing where new colonies of the invasive hornet *Vespa velutina nigrithorax* have begun to spread. The system works by drawing the insects to a pad that is impregnated with tasty (to wasps) smelling foodstuffs.

Considerate response

VespAI provides a nonharmful alternative to traditional trapping surveys and can also be used for monitoring hornet behaviour and mapping distributions of both the Asian hornet (*Vespa velutina*) and European hornet (*Vespa crabro*) which is protected in some countries. "Live hornets can be caught and tracked back to the nest, which is the only effective way to destroy them," explains the team's research paper: **magpi.cc/vespaihornet**.

Creepy feeling

VespAI features a camera positioned above a bait station that detects insects as they land to feed and gets to work establishing whether the curious mite is, in fact, an Asian hornet. The Exeter team developed the AI algorithm in Python, using YOLO image detection models: **magpi.cc/vespaigit**. These identify whether Asian hornets are present and, if so, send an alert to users. Raspberry Pi proved



▼ Non-Asian hornets are discounted, meaning non-invasive native species are not destroyed in a bid to eradicate the destructive newcomers

MAKERS

Exeter University VespAI team

The research team from University of Exeter's Environment and Sustainability Institute field-tested the Asian hornet detector in Jersey and Cornwall.

magpi.cc/vespaigit



VespAI can be fitted with solar panels so it can operate in remote locations and alert researchers of new Asian hornet populations

Raspberry Pi 4 provides connectivity and powers the AI algorithm that crosschecks a potential hornet against its 33,000-strong image database

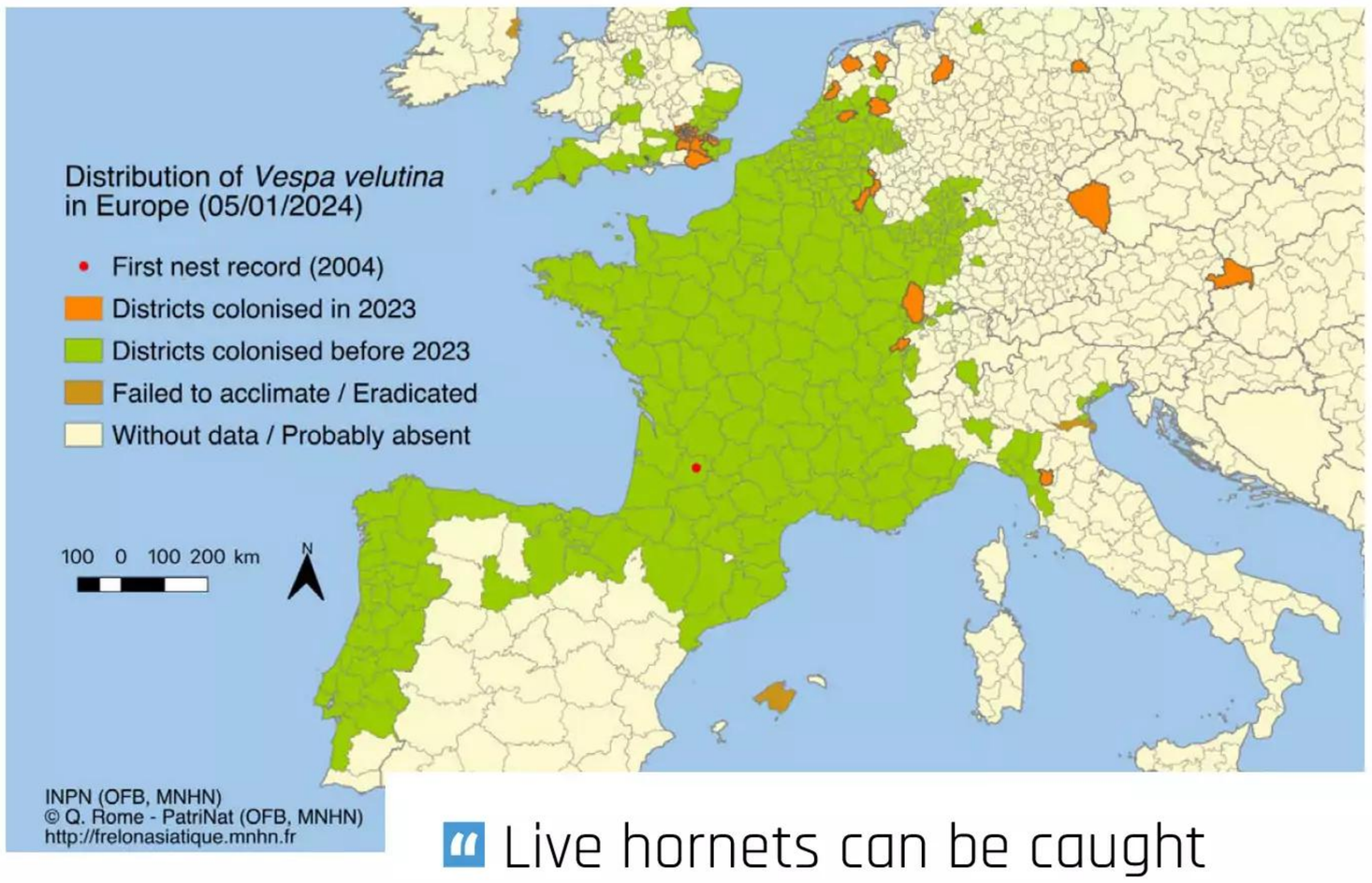
VespAI provides a non-harmful alternative to traditional trapping surveys, identifying Asian hornets using AI and a 16Mp camera. A plastic baffle deters squirrels attracted by its bait

Quick **FACTS**

- VespAI was trained on more than 3300 hornet images
- The detection unit is protected from raiders with a squirrel baffle
- The team has handled thousands of Asian hornets...
- ...and no one has ever been stung
- European hornets are protected in many countries

Warning!
Sharp sting

Wasps and hornets do not relish being prodded and poked, and may sting you if you seem to be a threat. If you create your own Asian hornet detector, inform a specialist to trace them back to their nest.
magpi.cc/insectbites



▶ Asian hornets have rapidly spread from southern Europe and are now increasing in numbers in the UK



▲ Dr Thomas O'Shea-Weller, Juliet Osborne, and Peter Kennedy

▶ Once attracted to liquid in a Petri dish, VespAI identifies any Asian hornets and automatically alerts researchers who trace them back to their nest

> "Live hornets can be caught and tracked back to the nest, which is the only effective way to destroy them"



a great choice because of its compact size, ability to run the hornet recognition algorithm, real-time clock and support for peripherals such as an external battery. The prototype bait station design was made with items that the team had at hand in their lab, including a squirrel baffle for the weather shield, Petri dishes and sponges to hold hornet attractant, and a beehive stand for the monitor to rest on.

Design challenges included optimising the hornet detection algorithm for use on Raspberry Pi. "An AI algorithm may work well during training or when validated in the lab. However, field deployment is essential to expose it to potentially unforeseen scenarios that may return errors", they note. The project also involved developing a monitor with an integrated camera, processor, and peripherals while minimising power consumption. To this end the VespAI team is currently optimising their software to run on Raspberry Pi Zero having watched footage of the AntVideoRecord device monitoring leafcutter ant (*Acromyrmex lundi*) foraging trails (**magpi.cc/antvideo**) and been impressed by its ability to run for extended periods remotely due to its low power consumption.

As this interactive map shows, Asian hornets have quickly made inroads across Western Europe: **magpi.cc/vespamap**.

The Raspberry Pi-enabled setup is "intended to support national surveillance efforts, thus limiting hornet incursions into new regions," explains Dr Thomas O'Shea-Wheller, a research fellow in the university's Environment and Sustainability Institute. He and his colleagues have been working on the AI project since 2022, conducting additional fieldwork this summer with the National Bee Unit and the Government of Jersey (Channel Islands) mapping new locations and fine-tuning its accessibility to potential users ahead of a planned commercial version.

Given Raspberry Pi's extensive and enthusiastic users, they hope sharing their code on GitHub will help expand the number of VespAI detection stations and improve surveillance and reporting of hornet species.

Wasp watch



01 The system is inactive unless an insect of the correct size is detected, in which case the 16Mp camera takes a photo and the YOLO algorithm begins its analysis, comparing it against its existing image database.



02 VespAI runs on Python and uses TensorFlow to facilitate hornet detection through the YOLO architecture. Should an Asian hornet be detected, researchers receive a smartphone or email alert.



03 A rugged external battery powers the wireless Raspberry Pi 4-based VespAI setup, making it suitable for use in remote locations. The team plans to launch a consumer version soon.

LED umbrella

With Thomas Killus' illuminating project, you won't be wishing the rain would go away any time soon, as **David Crookes** explains



MAKER

Thomas Killus

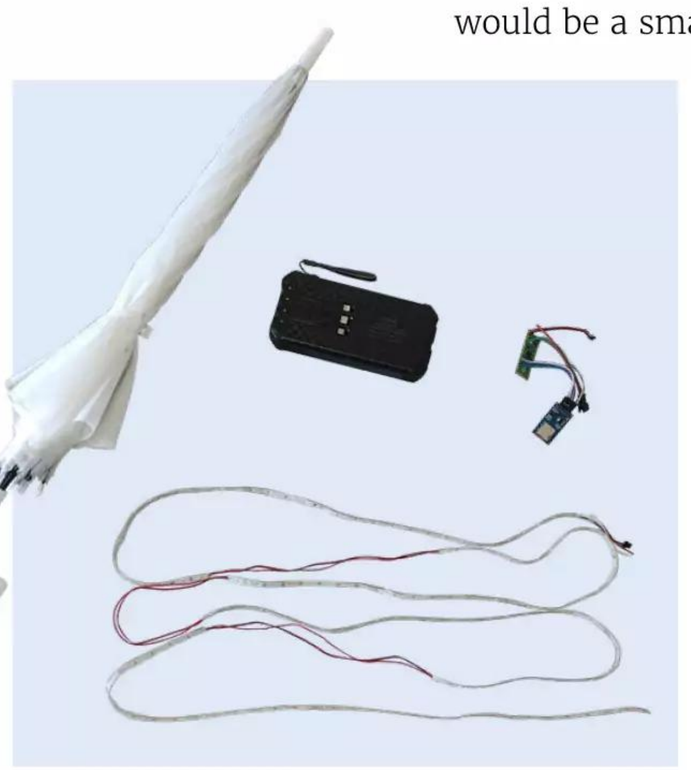
Thomas is a robotics enginner who grew up in Germany, did a master's degree in mechanical engineering and worked in Tokyo for two years.

magpi.cc/ledbrolly

Aside from being wet, rainy days are often dark and gloomy, seldom putting anyone in a particularly good mood. But imagine if your umbrella shone bright with an array of colours and had some snazzy light effects. Not only would that project a cool glow as you walk down the street, it would surely put a smile on people's faces too.

Thomas Killus has created such a brolly using a Raspberry Pi Pico microcontroller board. It sends instructions to strips of standard WS2812b addressable RGB LEDs that run along the ribs of an umbrella, and this in turn brightly disperses light across the adjoining panels. The umbrella would certainly shine bright in a crowd and maybe even prevent it being left behind on the bus. But it's also, curiously, the first step towards something bigger.

"I have always been fascinated with beautiful light installations," Thomas says. "My dream is to someday build a complete LED suit that can either be pre-programmed or made to interact with the environment through sensors. It would be nothing practical, but I hope to bring a smile to the people around me and myself. I thought an umbrella would be a small step into this direction."



▲ The project requires very few parts, but a portable charger will be required

Brolly good

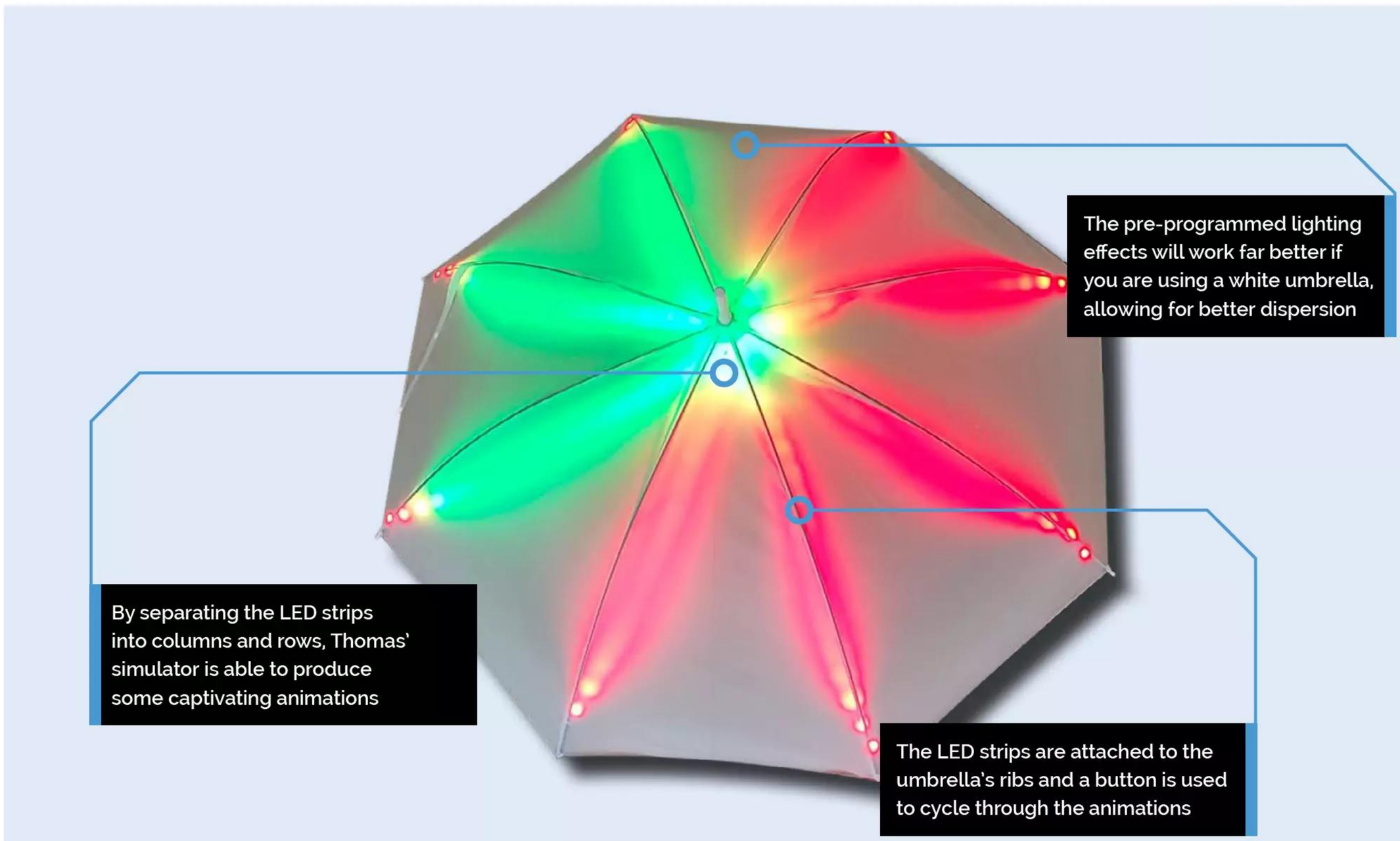
Planning the umbrella was straightforward enough. "Get an umbrella, get LEDs and put them together," Thomas explains. In the past he had tended to use an ESP32 microcontroller for his hardware projects and Python for his pure software projects. "But one day I saw a Raspberry Pi Pico microcontroller in a store and I recognised the great potential of finally using Python in my hardware projects, so I knew I had to give it a try," he explains.

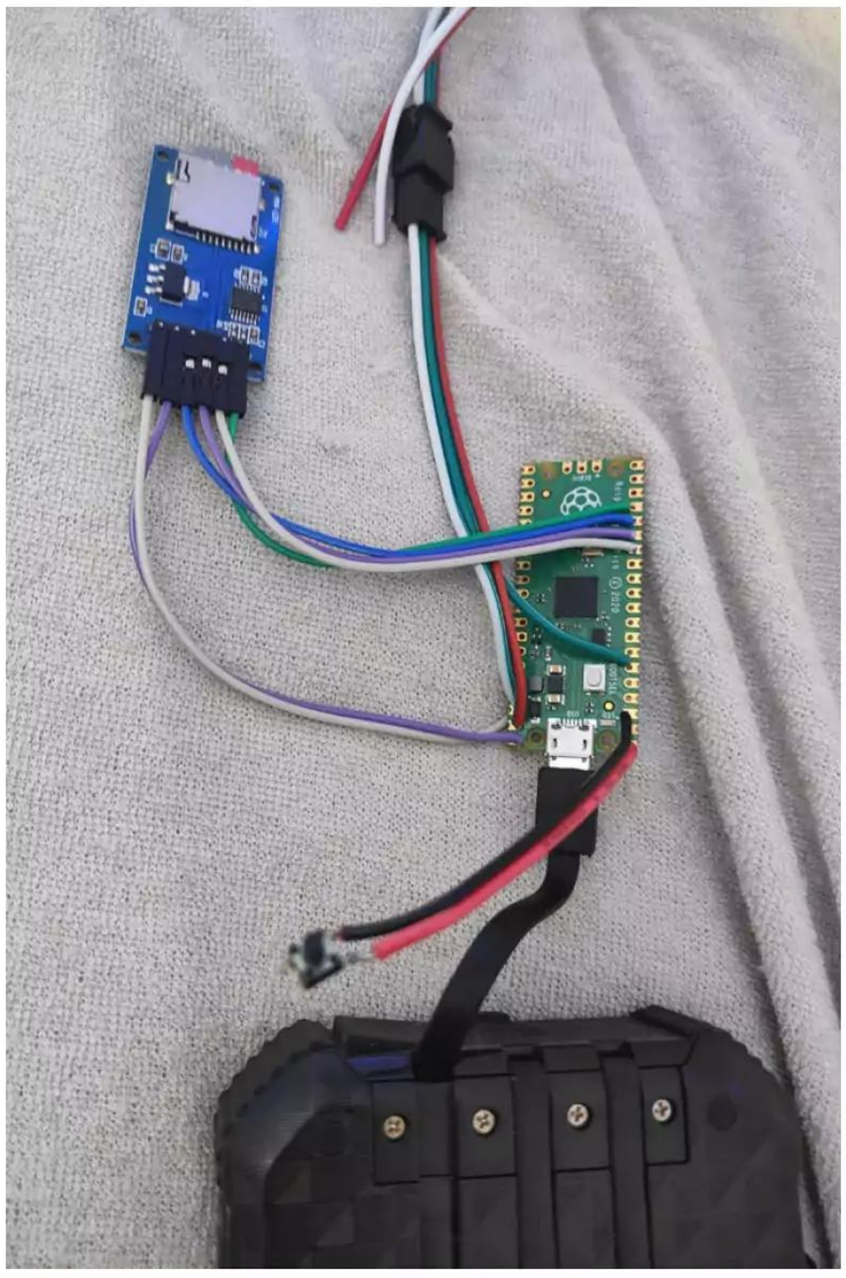
The idea was to create a set of pre-programmed animations that could be easily skipped through by simply pressing a button. These animations were created on a PC using a simulator Thomas had developed in Pygame. They were then saved to an SD card which plugged into the umbrella. "Raspberry Pi Pico can read the animation data and control the LEDs accordingly," he explains. "I thought this way, I wouldn't have to limit the animations by the computer power or memory limitations of Raspberry Pi Pico."

> "I have always been fascinated with beautiful light installations"



▲ Since all of the lights were connected in one long chain, Thomas only needed to use a single data pin on the Pico to control them all





Development went reasonably well. "It didn't cause me major headaches," Thomas says. "But in the beginning, I had some issues with reading the animation commands fast enough from the SD card while displaying them live but I eventually figured it out using byte files."

Shine a light

There are some features he would like to implement, though. Thomas originally planned to also add an MPU9250 nine-axis motion tracking sensor to measure acceleration, rotation and orientation. "I wanted to add features which will make the umbrella change its lights when you spin it, jump or turn into a different direction and I hoped it would give the user more interactions with the umbrella," he says.

Thomas also wants to reduce the number of visible cables and make the underneath "sparkle with the same joy," as he puts it. "LED light isn't diffused on this side and it takes away a lot of the effect," he laments. Even so, the umbrella has gone down well among the Raspberry Pi community even though many say it has no practical use. "I guess they are right," he says. "But I enjoyed building it, I learned some things and it made me smile when it lit up for the first time."

Quick FACTS

- It uses an ordinary umbrella and Raspberry Pi Pico
- Strips with a minimum of 118 LEDs are attached
- They display pre-programmed animations
- Thomas wants to add interesting interactions...
- ...but any singing in the rain is optional

◀ The SD card module is connected to Raspberry Pi Pico via the SPI port

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**

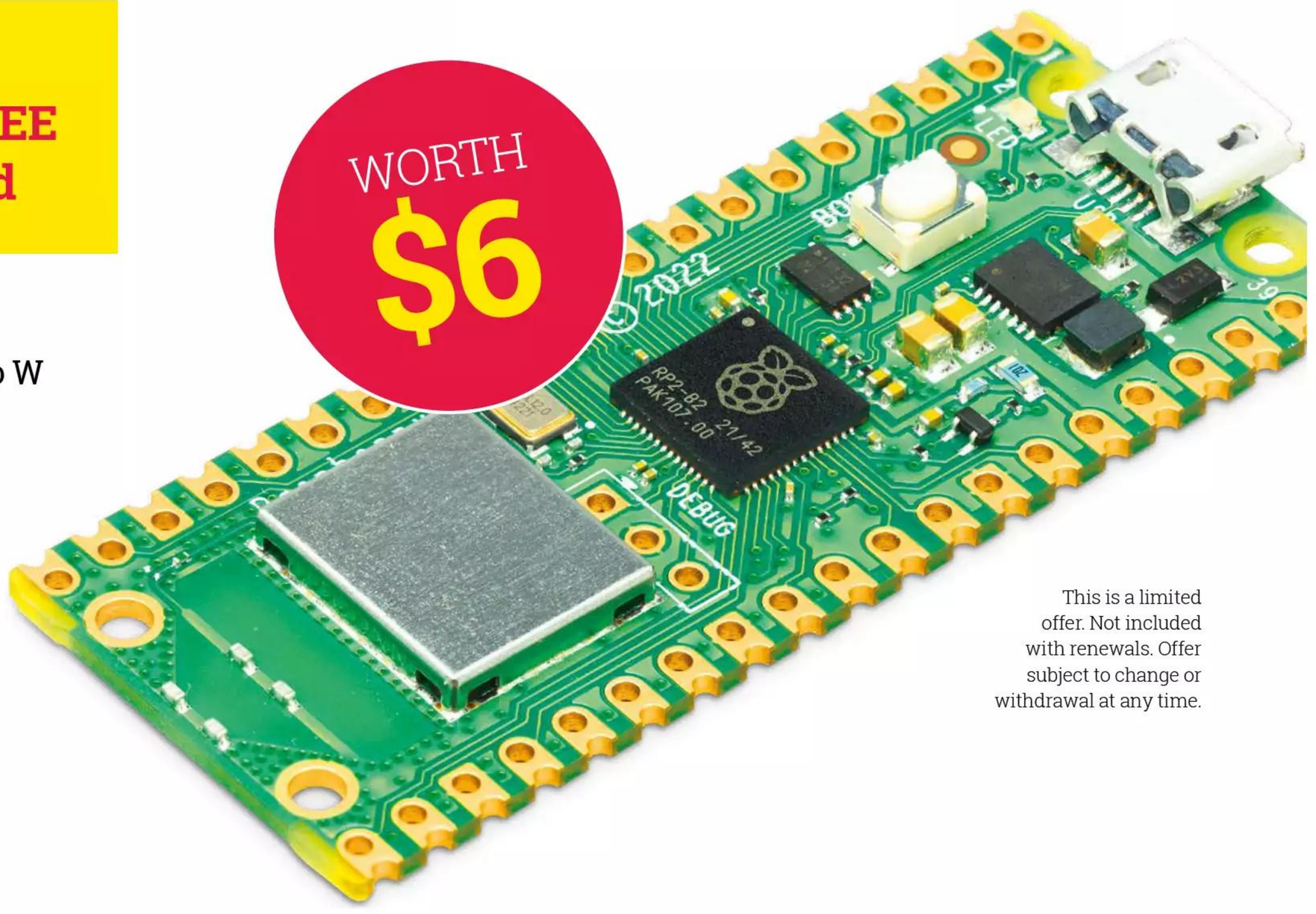
Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a six-month subscription in USA, Europe and Rest of World.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico W

Subscribe in print today and get a FREE development board

- 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



PLAY RETRO HORROR CLASSICS ON RASPBERRY PI 5

Spattered with blood, gore, and overcome by a desperate urge to prove that computer games weren't just for families, let's revisit the classics from gaming horror's great 1990s golden age.

By **KG Orphanides**

Retro styled horror games for modern platforms are numerous and many of them are genuinely outstanding. It's a genre that lends itself to small-scale indie development effectively as it does to massive triple-A studios, and where art house or B-movie sensibilities can often hit harder than glossy big-budget production values. But to get a real handle on where modern gaming's obsession with retro horror comes from, it's worth taking time to visit the genre's roots.

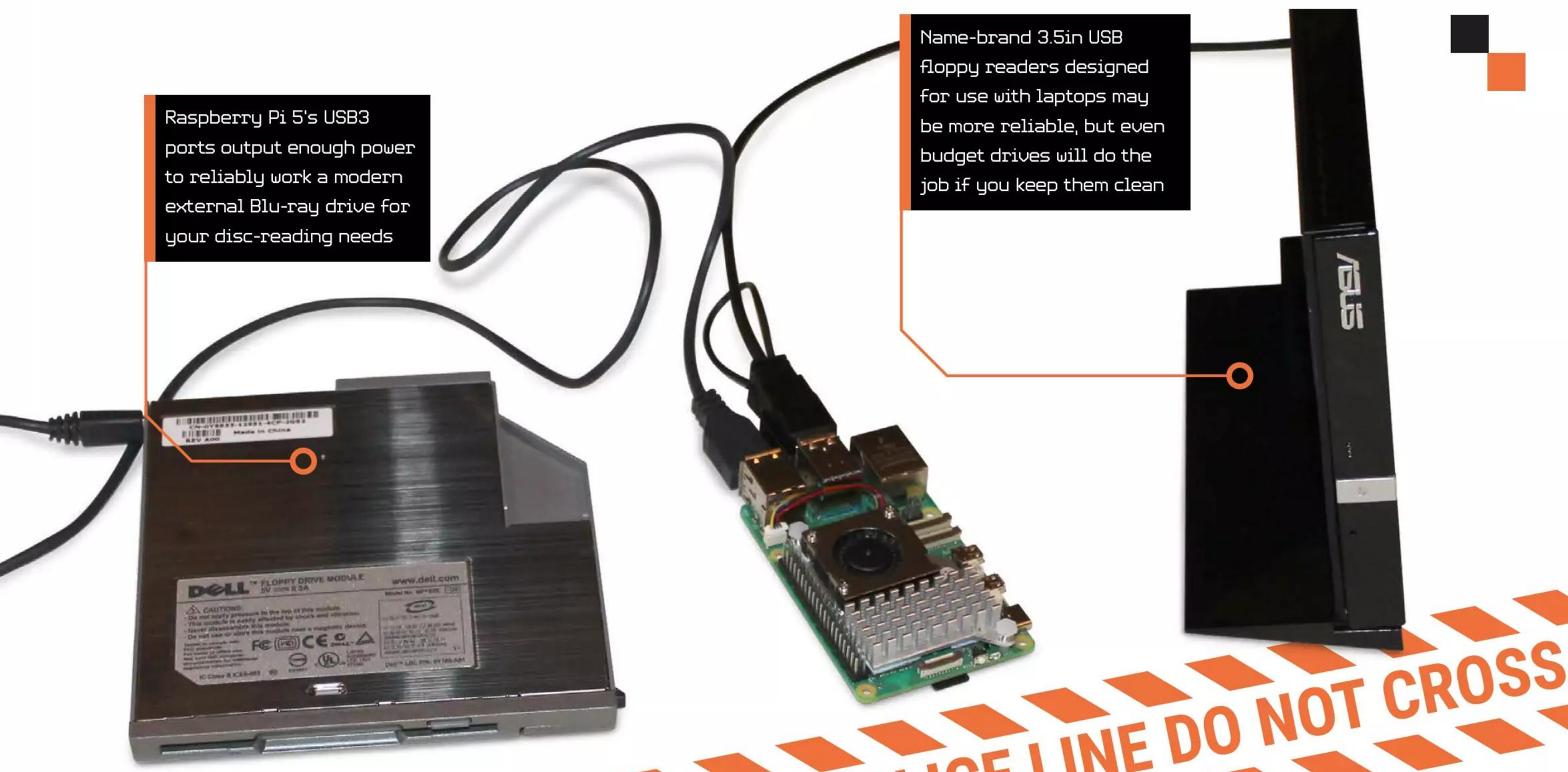
Although many older horror games can now feel stilted or clichéd, there's plenty to appreciate in terms of both their design, their clever use of the technologies available to their developers, and their writers' willingness to engage with some genuinely dark subject matter. For this latter reason, many of the games we've included received Mature or 18 ratings at the time of their release. While the gaming gore of the last century may no longer convince us, the subject matter of the games can still be disturbing, so we've added content warnings where appropriate. This project is not all-ages.

We'll start by assembling our retro toolkit - the hardware and software we'll need to play these old-school classics, from emulators to disk drives. Although we'll look at installing or running games from original hardware - the only viable approach for original PlayStation games, for example - where possible, we'll favour games that are currently legally available to buy, making this tour of retro gaming's dark side as accessible as possible for those who don't happen to have a stash of ancient floppy disks or CD-ROM installation material mouldering in a crypt somewhere.

Warning! Mature content

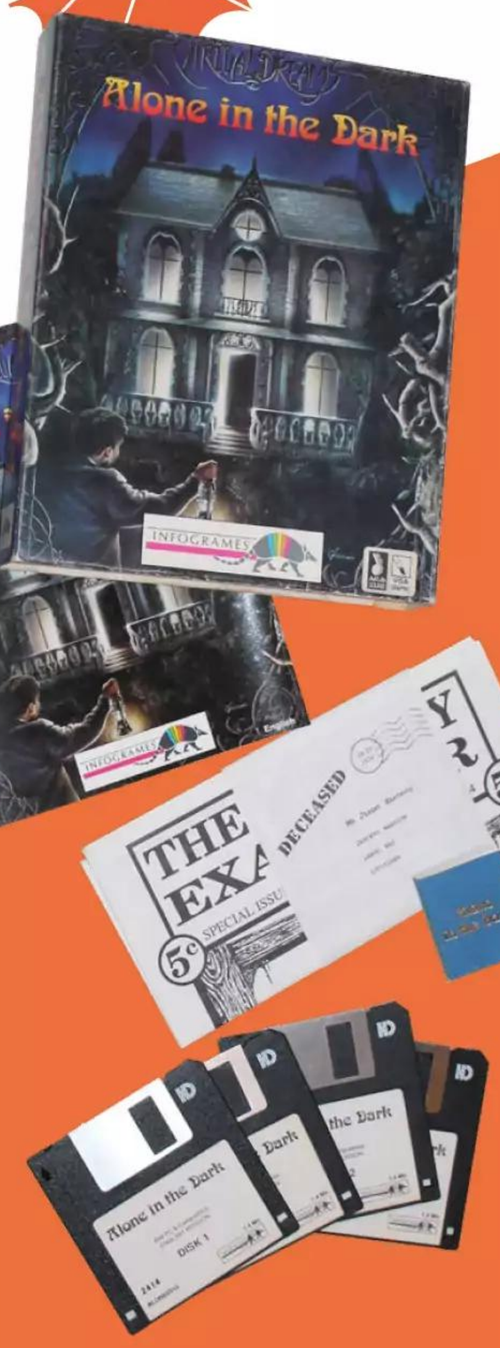
The games mentioned in this feature include themes and imagery that may not be appropriate for sensitive players. Several are rated by PEGI as only suitable for those aged 18 and over.

pegi.info



Your Retro Toolkit





Build DOSBox Staging

DOSBox Staging is easy to use and tries accurately handle retro quirks such as the reproduction of the rectangular pixels of the pre-640×480 graphics era (**magpi.cc/dbsgraphics**).

You'll find an aarch64 version on Flathub (**magpi.cc/dbsflathub**), but if you're on a 32-bit operating system or want to try pre-release versions, building from source is your best option. The source build also integrates more smoothly with your OS.

01 Dependency

First, open a terminal and install the build dependencies. These are for the latest version of Raspberry Pi OS Bookworm. Users on older releases or alternative distros may need to install meson via pip. See the included **BUILD.MD** file for details.

```
$ sudo apt install meson ccache build-
essential libasound2-dev libatomic1 libpng-
dev \
          libsdl2-dev libsdl2-net-dev
libopusfile-dev \
          libfluidsynth-dev libslirp-dev
libspeexdsp-dev libxi-dev
```

02 Bleeding edge

If you want to try the version of the code that's currently being worked on:

```
$ git clone https://github.com/dosbox-
staging/dosbox-staging.git
$ cd dosbox-staging
```

Otherwise, download the latest source code package from **magpi.cc/dbssource** – that was 0.80.1 at the time of writing, and extract it, thus:

```
$ wget https://github.com/dosbox-staging/
dosbox-staging/archive/refs/tags/v0.80.1.tar.gz
$ tar -xzf v0.80.1.tar.gz
$ cd dosbox-staging-0.80.1/
```

03 Meson scene

Regardless of how you obtain the source code, the following build process is the same:

```
$ meson setup build/release
$ meson compile -C build/release
$ cd build/release
$ ./dosbox
```

Once we've confirmed that it runs, and allowed it to create its config file, exit and type:

```
$ sudo meson install
```

04 dosbox.conf

A config file is automatically created at **~/.config/dosbox/dosbox-staging.conf**. This serves as your primary configuration. DOSBox Staging will also check the directory you run it in for an additional **dosbox.conf** file, so you can easily create optimised per-game configurations. Any instructions you put in a per-game **dosbox.conf** file will supersede those in the main **dosbox-staging.conf** file.

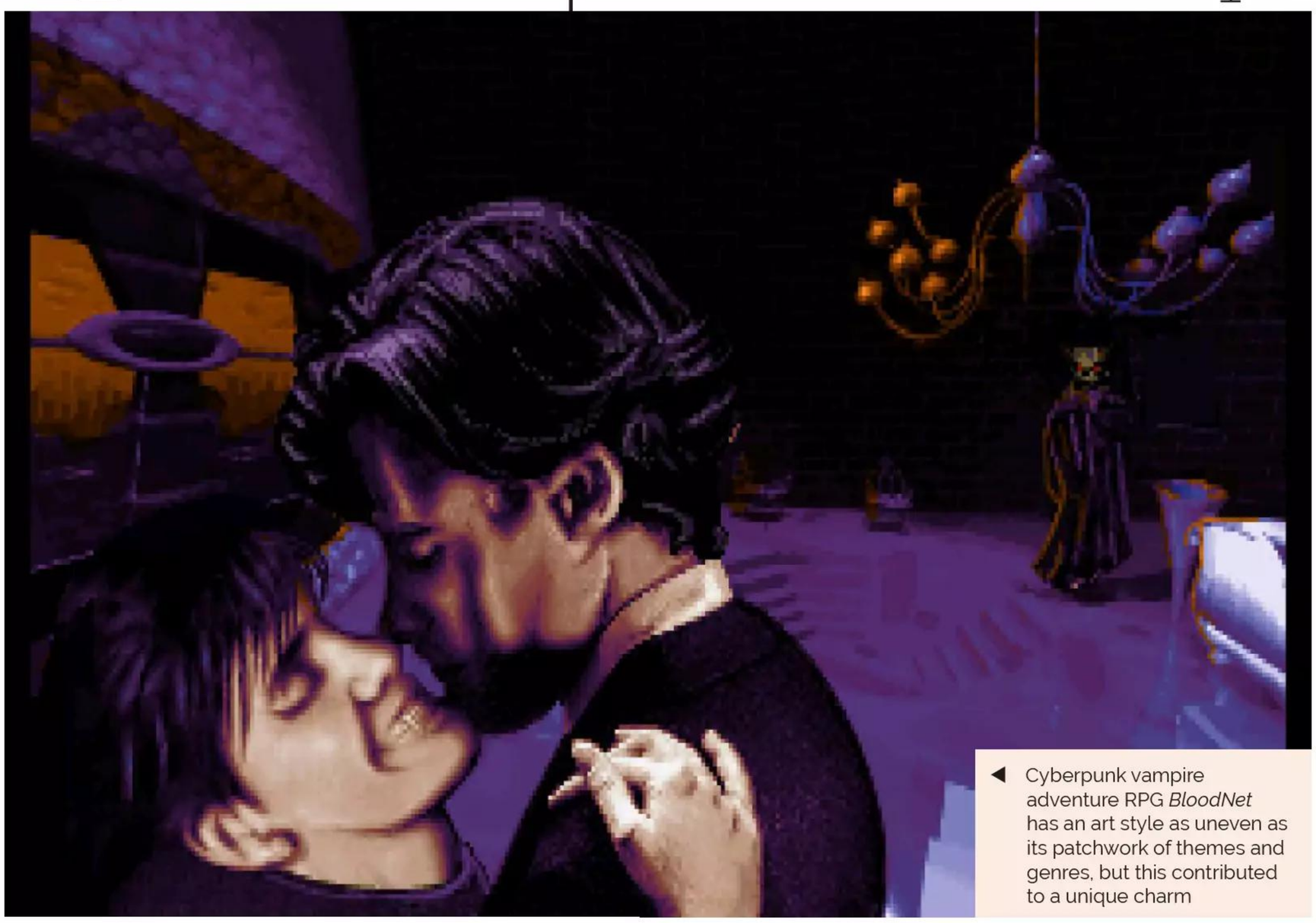
05 Ease of assembly

You can alternatively install DOSBox Staging as a Flatpak and then, because Flatpaks are deliberately air-gapped from the rest of your system, create an alias that will allow you to conveniently run it from the command line



▲ *Alone in the Dark*'s graphics are extremely low-poly, but its original release came with a rich stash of feelies and background lore

▶ *Veil of Darkness* might be an isometric game published by SSI, but it's firmly in the realm of adventure gaming



◀ Cyberpunk vampire adventure RPG *BloodNet* has an art style as uneven as its patchwork of themes and genres, but this contributed to a unique charm

```
$ sudo apt install flatpak
$ flatpak remote-add --if-not-exists flathub
https://dl.flathub.org/repo/flathub.flatpakrepo
$ flatpak install dosbox-staging
$ nano ~/.bashrc
```

Add the following line:

```
$ alias dosbox-staging="flatpak run io.github.
dosbox-staging"
```

Save, and then refresh your bash profile by typing this:

```
$ source ~/.bashrc
```

Build ScummVM

ScummVM was originally a tool for playing point and click adventure games created with LucasArts' SCUMM (Script Creation Utility for Maniac Mansion) engine, but now runs scores of games, from 1978's *Adventureland* to new releases like *Gobliiins 5*.

It supports aarch64 systems like Raspberry Pi 5 via its Flatpak release, or you want to build a release version of ScummVM yourself following the instructions below, official source code releases can be found at **magpi.cc/scummvmgit**.

To build these, you can follow the instructions below, but ignore the git clone stage and instead unzip the code you've downloaded.

If you need DEV version game support (**magpi.cc/scummvmcomp**), or wish to contribute to ScummVM, you should clone the git repository.

To build the current development version from the ScummVM Git repo:

```
$ sudo apt install g++ make git nasm libsdl2-
dev libsdl2-net-dev liba52-dev libjpeg62-
turbo-dev libmpeg2-4-dev libogg-dev libvorbis-
dev libflac-dev libmad0-dev libpng-dev
libtheora-dev libfaad-dev libfluidsynth-dev
libfreetype6-dev zlib1g-dev libfribidi-dev
libgif-dev libcurl4-openssl-dev libgtk-3-
dev libspeechd-dev libsndio-dev libvpx-dev
libmikmod-dev
$ git clone https://github.com/scummvm/
scummvm.git
$ ./configure
$ make clean
$ make -j4
$ ./scummvm
```

If it works and you wish to install it system-wide:

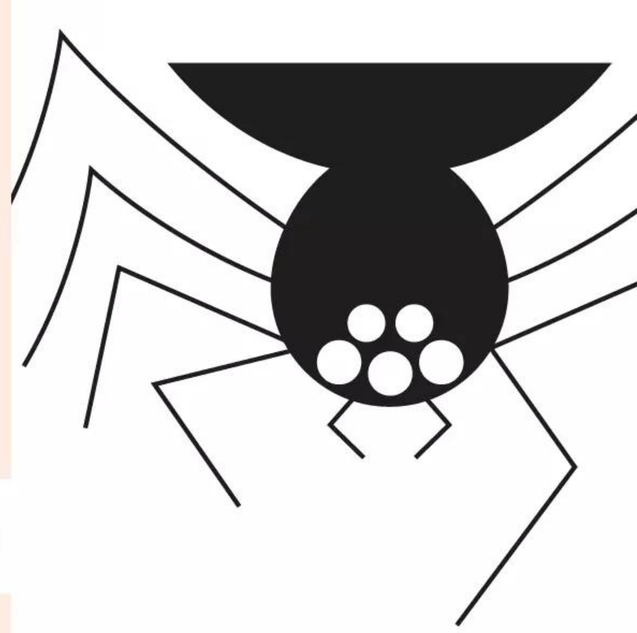
```
make install
```

Top Tip

The X Factor

DOSBox Staging is easy to use, but if you want a GUI and even more configurable hardware emulation, DOSBox-X (**dosbox-x.com**) is also outstanding.







▶ *Phantasmagoria* used state-of-the-art video alongside traditional 2D environments to bring a gory adventure to home computer screens

> "Plenty of online stores will sell you digital retro games"

Hardware: Floppy disk drive

To read floppy disks, you'll need an external reader. We use a DELL MPF82E. Multi-floppy games will have the same problem with disk swapping in DOSBox as multi-CD ones. To work around this, add the following to **~/etc/pmount.allow**:

```
/dev/sda
```

You can now mount and unmount floppy disks thus:

```
$ pmount /dev/sda /media/floppy
```

You can unmount them with:

```
$ pumount /dev/sda
```

However, physical floppy swapping does not require unmounting and remounting.

Hardware: External disc reader

We looked in detail at using external CD/DVD readers with Raspberry Pi in *MagPi* #140 (**magpi.cc/140**). Note that Raspberry Pi 4 and below will not reliably work with most modern external disc drives. If your software comes on a single disc, you can mount the CD or DVD and point your software at the folder your OS gives it under **~/media/YourUserName/**

If you need to swap discs, pmount, a user-space wrapper for the mount command, can arbitrarily create mount points.

```
$ sudo apt install pmount
$ sudo nano /etc/pmount.allow
```

Add the following to the bottom of the file:

```
/dev/sr0
```

Now open File Manager. Go to Edit > Preferences > Volume Management and untick all the Auto-mount options. Reboot your Raspberry Pi.

To mount a disc, insert it, let it spin up and, in a terminal, type:

```
$ pmount /dev/sr0 /media/cdrom
```

To unmount it:

```
$ pumount /dev/sr0
```

Buying digital games

If you don't have original releases, plenty of online stores will sell you digital retro games. We prefer GOG, as the Windows downloads of its retro games use a package format that we can extract directly on Raspberry Pi. Open a terminal and:

```
$ sudo apt install innoextract
```

Innoextract includes a special --gog (or -g) flag to handle GOG-specific packaging standards, but it's sometimes found it necessary to rename files in multi-file exe and bin sets before extraction.

If you've bought old games elsewhere that come packaged with DOSBox or ScummVM, you can usually transfer installed files from your x86-64 PC to play on Raspberry Pi.

You can also buy retro games on Steam (**store.steampowered.com**), Zoom Platform (**zoom-platform.com**) and - especially for new 'retro' games - Itch (**itch.io**).

Top Tip

Come get SCUMM

If you're looking for games to play on ScummVM, the team maintains a handy list of currently available free and commercial titles at **magpi.cc/scummgames**

Games

Alone in the Dark (1992)

Released in 1992, the original *Alone in the Dark* is one of the games that birthed the survival horror genre. Part of the '90s wave of French adventure games inspired by the work of author H.P. Lovecraft, its third-person view and tank controls would inspire later classics such as *Resident Evil*. Although the series would pass through the hands of many publishers and developers, with some less-than-inspiring instalments, this tensely ponderous classic is worth a visit in all its low-poly glory.

The original 1992 version of *Alone in the Dark* is available to buy from various digital storefronts. Although it's emblazoned with an 18 certificate on some of them, this is a lookup error confusing it with Atari's 2024 reboot. The original game has never been rated, and although foes like the Night Gaunt and bathtub monster can be scary, especially when you're trying to escape them using the awkward controls, there's no significant gore and the monsters' design is unlikely to inspire visceral terror.

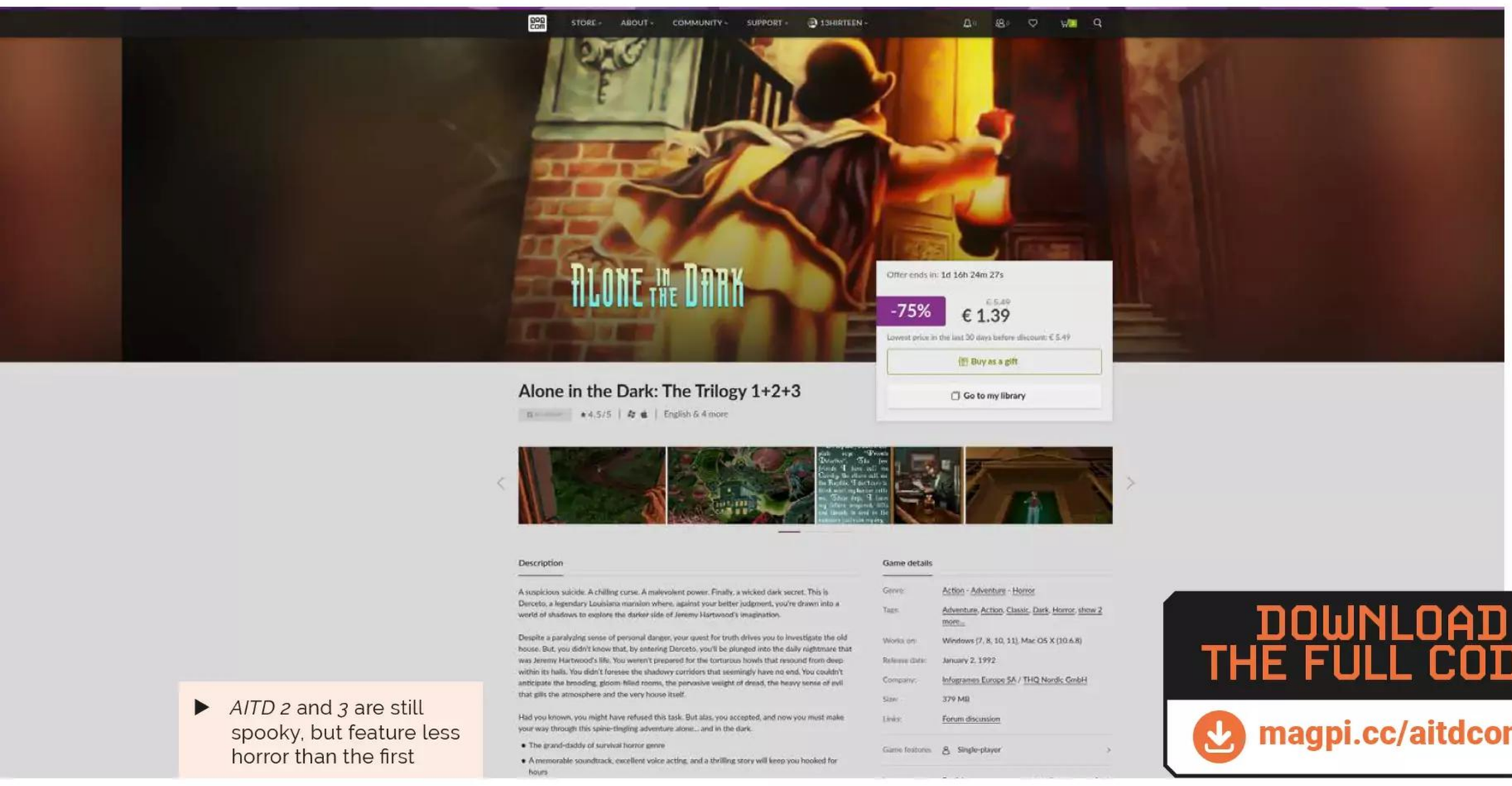
We'll use the GOG version. Buy and download the *Alone in the Dark* trilogy for Windows from GOG. This package gets you stand-alone downloads for each game, and the first includes both *Alone in the Dark 1* and the stand-alone mini adventure, *Jack in the Dark*, which was given away on magazine cover disks.

Copy the game's setup file into a directory called **~/dos/AITD** then open a terminal:

```
$ cd dos/AITD
$ innoextract -g setup_alone_in_the_dark_1.0_
cs_\(28043\).exe
```

GOG's own config files don't work well with DOSBox Staging, so we've created our own, incorporating the game selector from GOG's config. This **dosbox.conf** file should be placed in the **~/dos/AITD** directory. Then, to run the games, open a terminal and type:

```
$ cd  ~/dos/AITD
$ dosbox
```

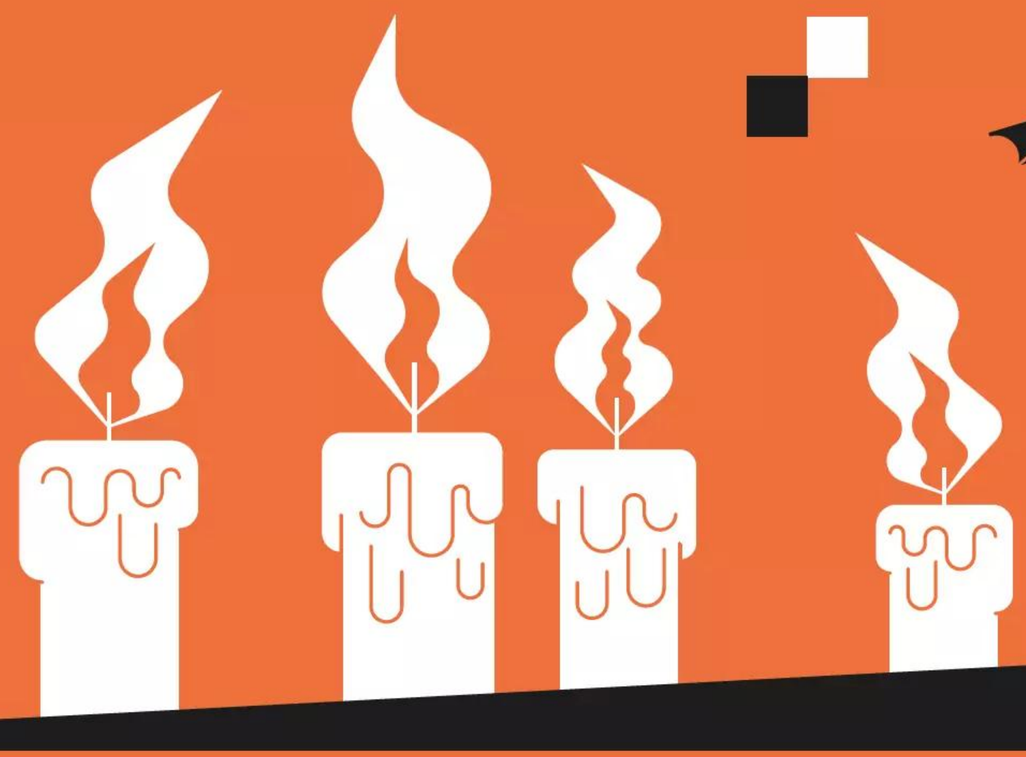


▶ *AITD 2* and *3* are still spooky, but feature less horror than the first

DOWNLOAD THE FULL CODE:

magpi.cc/aitdconfig







DOWNLOAD THE FULL CODE:

magpi.cc/vodconfig

Veil of Darkness (1993)

Strategic Simulations Interactive (SSI) is best known for its wargames and licensed Dungeons & Dragons RPGs, but also published some more unusual titles. *Veil of Darkness* is an isometric adventure game with RPG combat, developed by Event Horizon (later DreamForge Intertainment), which is also known for grimdark RPG *The Summoning*, horror-themed AD&D games in the Ravenloft campaign setting, and horror masterpiece *Sanitarium*.

We're going to install this one from floppy disks as an example of how to do so, but if you don't have a scruffy box of original '90s media lying around the place, you buy it on GOG (**magpi.cc/vodgog**) and use innoextract to unpack its files, as we did with *Alone in the Dark*, or install it from Steam and copy its files to Raspberry Pi.

> Veil of Darkness is an isometric adventure game with RPG combat

- We installed Veil of Darkness from floppy, but disks of this age are hard to find risk being deteriorated, so it might be best to buy a digital version

Open a terminal window and type:

```
$ pmount /dev/sda /media/floppy
```

Wait for this process to finish and for the disk to mount.

In a separate terminal window:

```
$ cd dos
$ dosbox
```

Once DOSBox Staging has loaded, type:

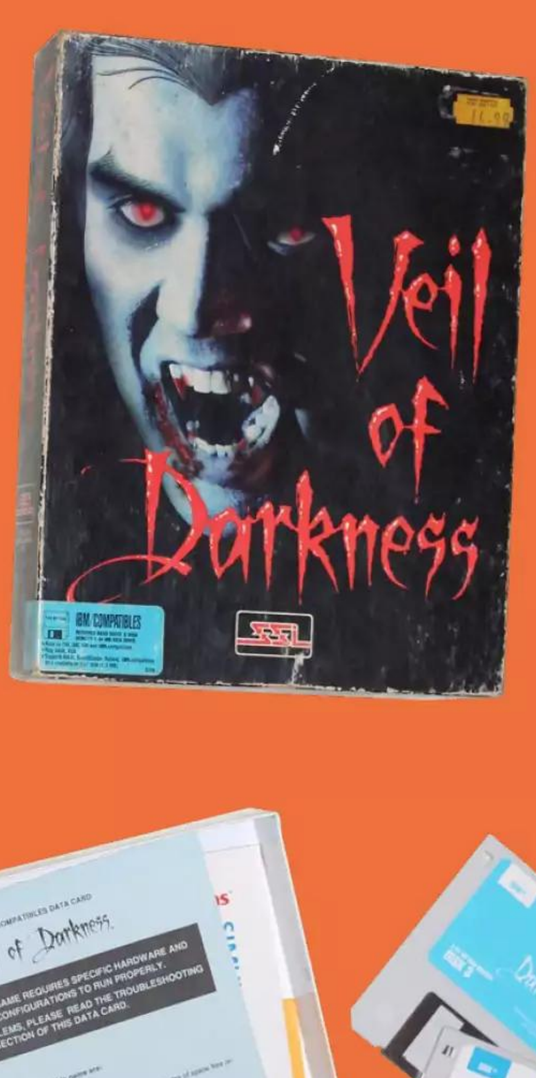
```
$ mount  c .
$ mount a /media/floppy
$ a:
$ install
```

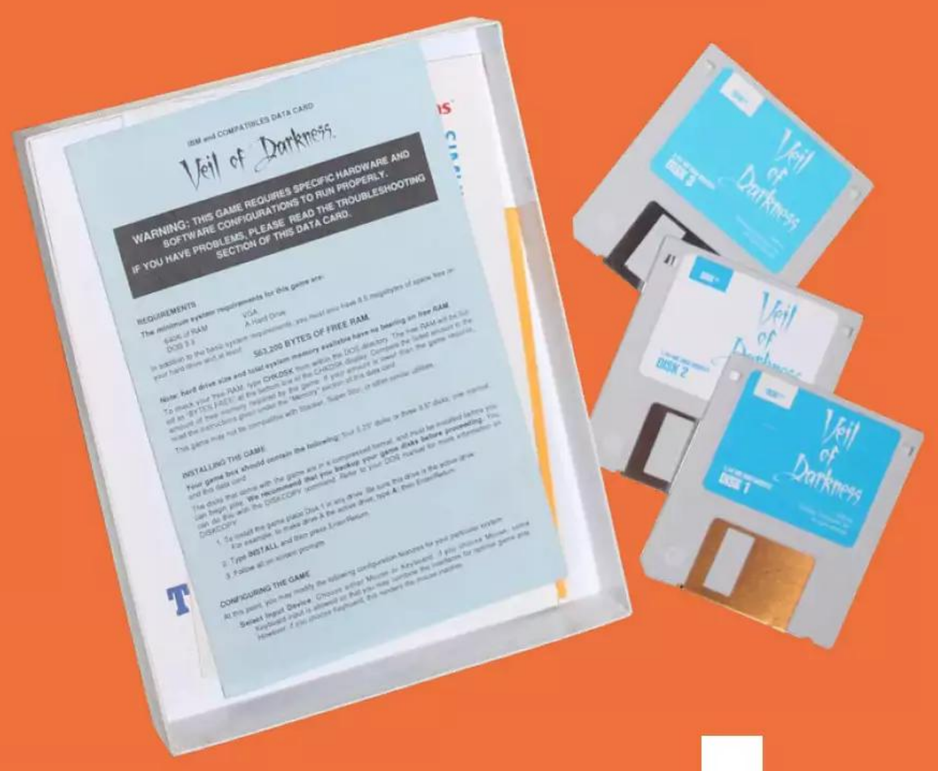
Wait a shockingly long time for the disk to unpack its contents. Read the manual. Make some tea. Installing from floppy was never quick, and installing from 30-year-old floppies is even slower. When you're finally told to insert disk 2 into A:, just eject the floppy, put in disk 2, press **Enter** to continue installation and wait.

When the installer produces its next distressing emulated PC speaker beep to tell you it's time to change the disk, repeat this process one last time. Once installation is complete, you'll be presented with a setup screen. It'll auto-select an Adlib card, but if you select No, go through all the setup options, and choose CMS Sound Blaster as your sound device, you should get full music and even a little intro voiceover.

Quit, and create the following **dosbox.conf** file in the **~/dos/VEIL** directory then, in a terminal:

```
$ cd ~/dos/VEIL
$ dosbox
```





BloodNet (1993)

MicroProse's *BloodNet* isn't an immortal work of art, but this gleeful, high '90s hybrid of vampire fiction, cyberpunk and noir detective tropes is a less-played entry in an unusual genre and a snapshot of a very particular time in game development. Even its art style is wildly, charmingly dissonant from one scene or UI element to the next. *BloodNet* never received an age rating and there's no real gore to speak of.

The CD version is, for better or worse, fully voiced. Wooden though protagonist Ransom Stark's actor is, the voice acting certainly gives us a feel for our cybernetically enhanced detective's inner sleazebag before he gets (partially) vampirised during the intro.

Manually create a character, and a series of ethical questions will determine your strengths and weaknesses. Being a mercenary, by picking violence-oriented responses, maximises your chances of surviving the game.

We're going to install it from a copy of the 1994 DOS CD-ROM release, but you can buy the game on GOG (**magpi.cc/bloodnet**).

First, let's pmount our CD-ROM somewhere sensible. Open a terminal and type:

```
$ pmount /dev/sr0 /media/cdrom
```

We already have a directory called **dos** in our home. Open another terminal and type:

```
$ cd dos
$ dosbox
```

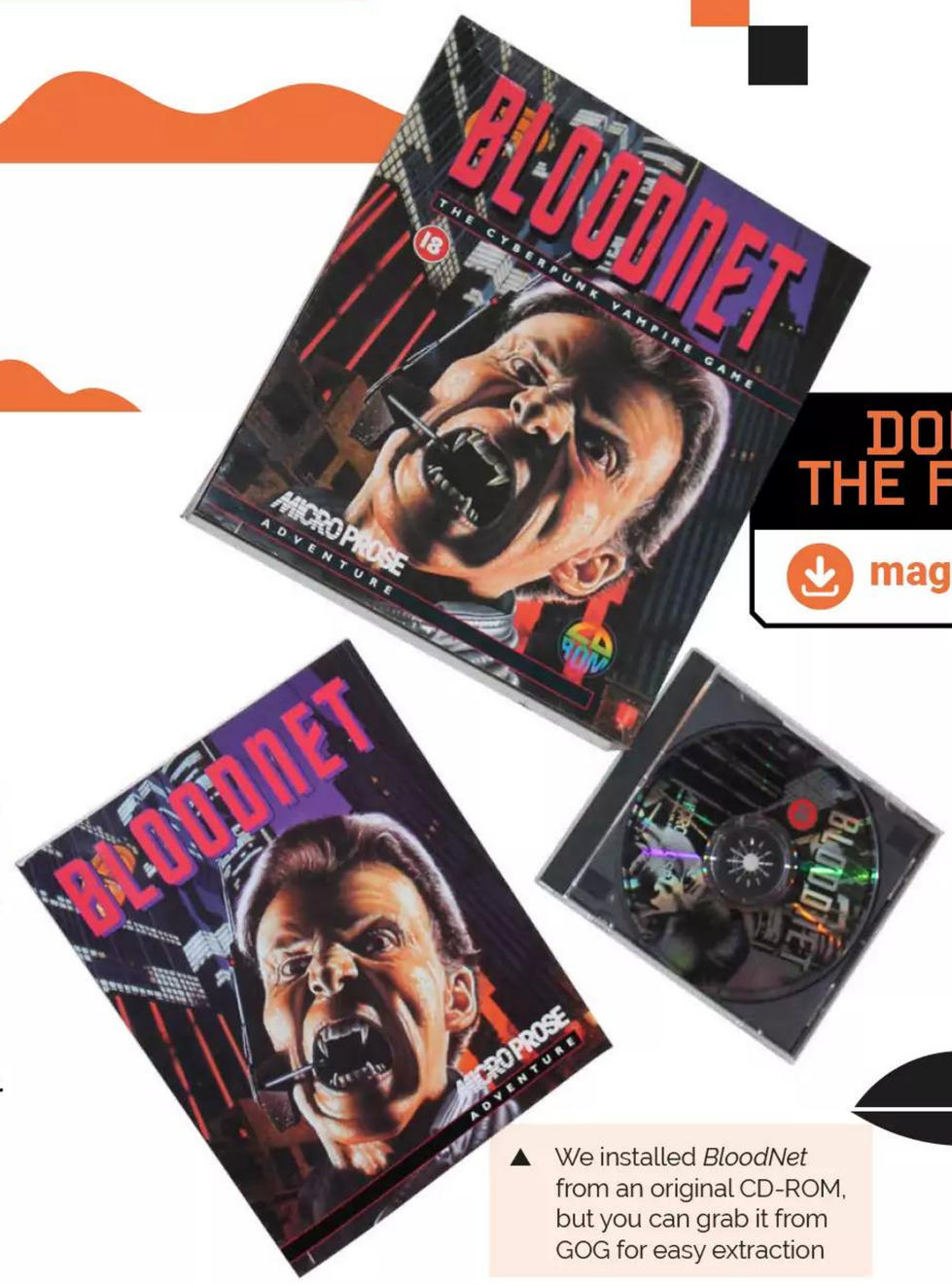
We'll create a dedicated **conf** file later, but for now, in DOSBox, type:

```
$ mount C .
$ mount D /media/YourUserName/UNTITLED/ -t
cdrom -usecd 0 -ioctl
$ D:
$ install
```

Confirm that you want it to install to drive C, use the arrow keys to go to Sound, press **Space** and select Sound Blaster Pro, and the Roland SCC-1 for music. Use the mouse to select Save and Exit the test the game by typing:

```
$ bloodnet
```

Quit the game and DOSBox. To add MIDI sound in the style of Roland's SCC-1, we'll use S. Christian Collins' GeneralUser GS 1.471 soundfont (**magpi.cc/gusoundfont**).



▲ We installed *BloodNet* from an original CD-ROM, but you can grab it from GOG for easy extraction

DOWNLOAD THE FULL CODE:

magpi.cc/bnetconfig

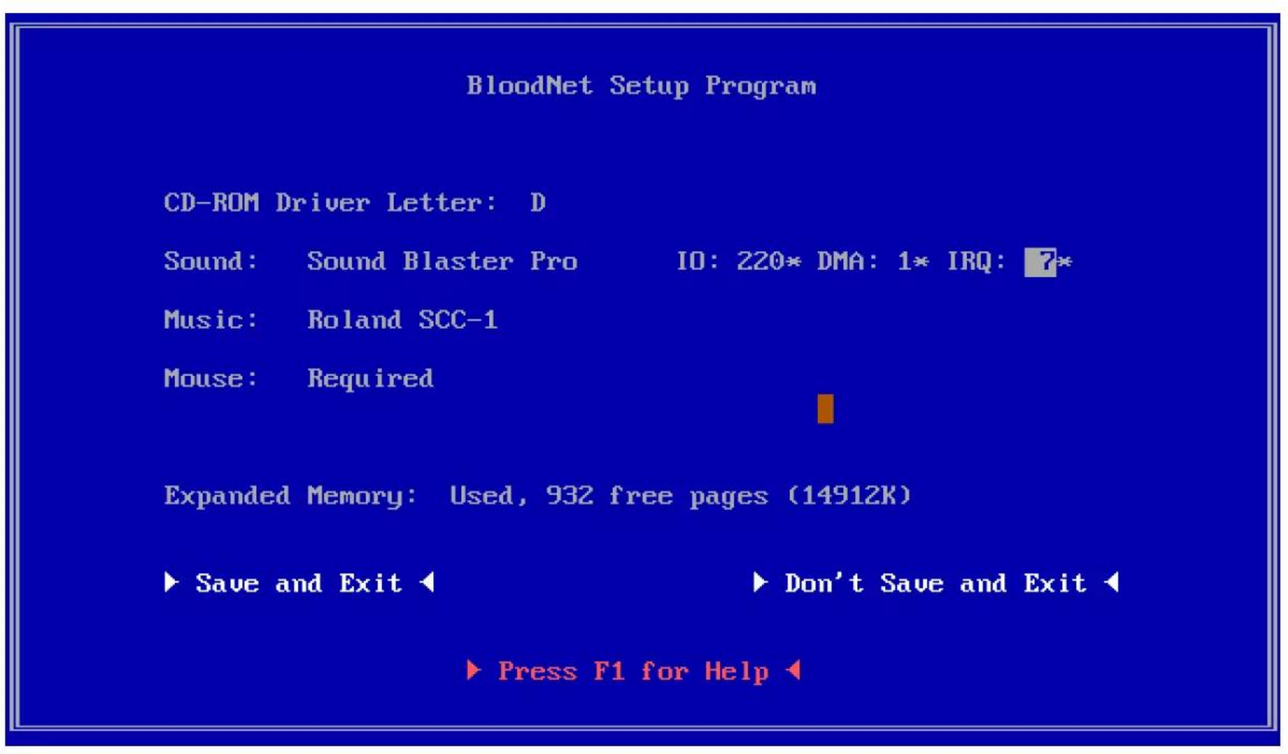
Open a fresh terminal window and type:

```
$ wget https://www.dropbox.com/
s/4x27l49kxcwamp5/GeneralUser_GS_1.471.zip
$ unzip GeneralUser_GS_1.471.zip
$ cp 'GeneralUser_GS_1.471/GeneralUser GS
1.471/GeneralUser GS v1.471.sf2'  ~/.config/
dosbox/soundfonts.
```

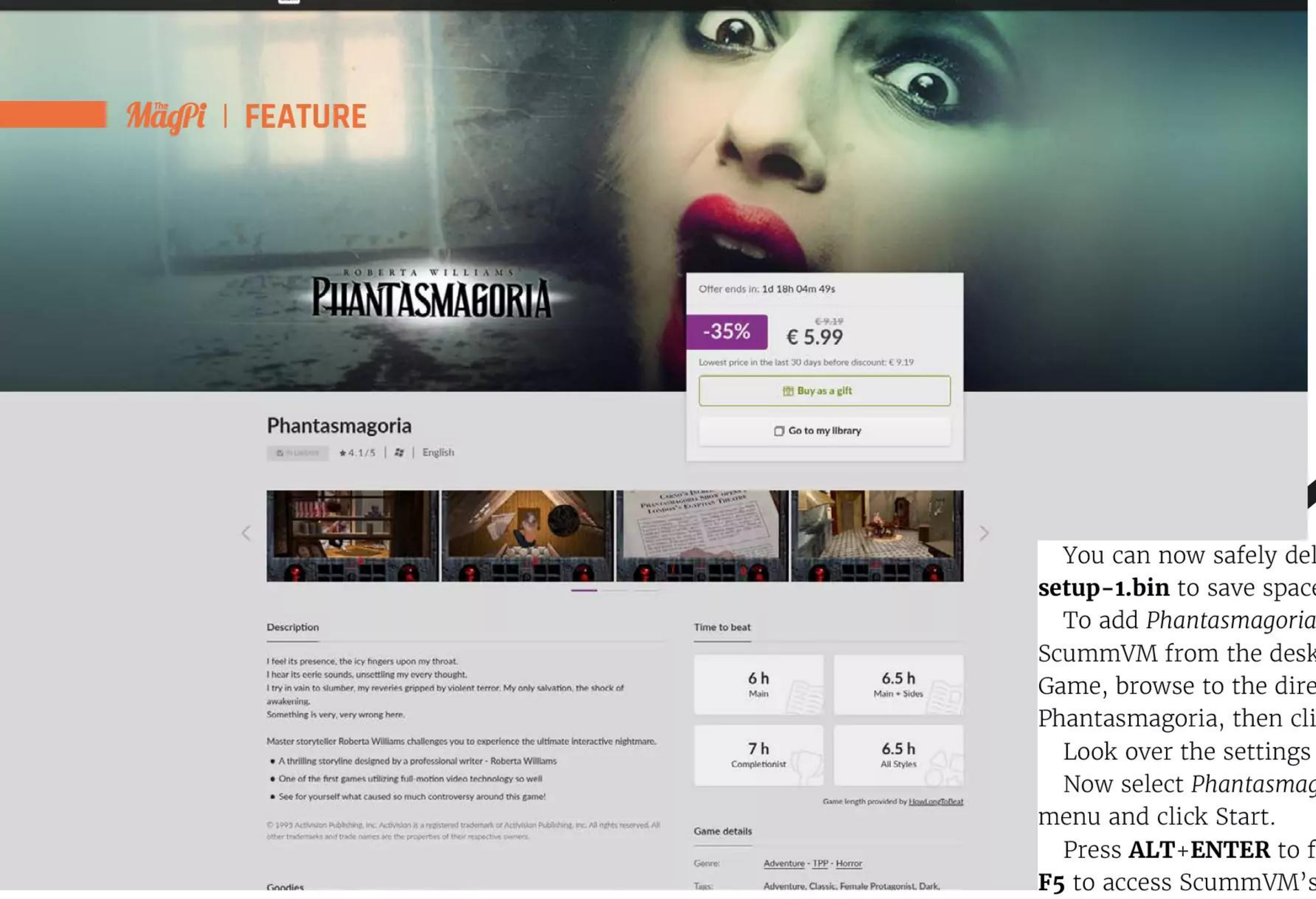
Now, create a custom configuration file:

```
$ cd /dos/MPS/BNCD/
$ nano dosbox.conf
```

Give it our supplied **conf** file and save, so that when you run DOSBox from *BloodNet*'s directory, it'll use our custom settings and auto-run it. Don't forget to leave the CD mounted while you're playing.



▼ Custom audio configuration will get you wavetable speech and effects and general MIDI music in *BloodNet*



▲ We bought and downloaded *Phantasmagoria* from GOG's extensive retro games collection.

Phantasmagoria (1996)

Phantasmagoria was written with the explicit intention of shocking. It includes scenes of violence, sexual violence and gore, and received a Mature rating at release. While the game's style is dated, writer/director Roberta Williams pulled the stops out on the unpleasantness for her venture into horror.

This includes scenes of torture, body horror, a pet cat whose name is a strong ableist slur in the UK and which dies unpleasantly, and a (clothed) FMV sexual assault scene against your protagonist that you're shown after completing Chapter 3, which you can't opt out of. You can enable and disable a censored mode in the settings to blur out violent and sexual cut-scenes, although the soundtrack plays on.

You can buy *Phantasmagoria* from GOG at **magpi.cc/phantasmagoria**. In your GOG library, search for the game, click it, then expand the Download Offline Backup Game Installers section. Download both parts of the Windows installer and put them in their own directory - we used **~/dos/phantasmagoria**. We'll use Innoextract to unpack them, and ScummVM to play them.

Innoextract isn't always happy with GOG's naming conventions, so we're going to rename the executable and its data files into something a bit easier to parse. Note that the version number in brackets may change if the version on GOG is updated at any point.

```
$ sudo apt install innoextract
$ mv setup_phantasmagoria_1.0_\(20239\).exe setup.exe
$ mv setup_phantasmagoria_1.0_\(20239\)-1.bin setup-1.bin
$ innoextract setup.exe --gog
```

You can now safely delete **setup.exe** and **setup-1.bin** to save space.

To add *Phantasmagoria* to ScummVM, open ScummVM from the desktop menu. Click Add Game, browse to the directory where you unpacked Phantasmagoria, then click Choose.

Look over the settings and click OK.

Now select *Phantasmagoria* from ScummVM's menu and click Start.

Press **ALT**+**ENTER** to fullscreen the game, and **F5** to access ScummVM's dedicated save menu.

Click Watch Intro to enjoy '90s FMV and *Hellraiser*-influenced 3D renders of protagonist Adrienne's nightmares, plus an extremely '90s softcore sex scene. Click Start New Game, name your save, select the first chapter, and start.

Phantasmagoria's blend of rendered environments and video-captured characters is unusual to the modern eye, and the low quality lends to its old-school B-movie charm. However, its single-mouse-button interface feels bare-bones compared to Sierra's earlier point and click adventures, so you'll want to read the manual.

Five more '90s horror classics to emulate

There are plenty more horror masterpieces and eccentricities to explore, available to buy now and play through DOSBox or ScummVM. Here are five of the best.

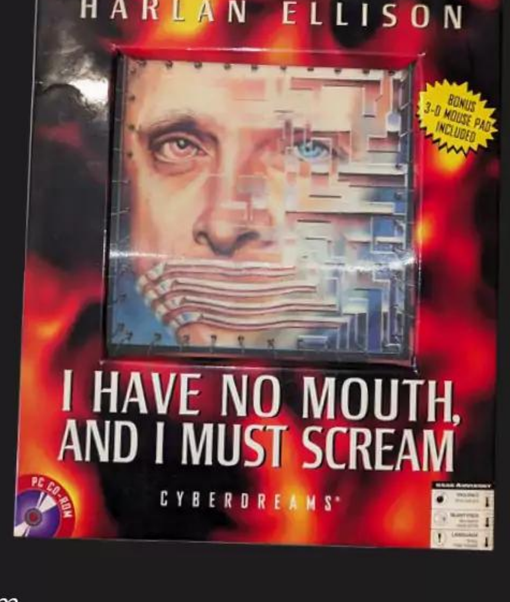
Gabriel Knight: The Sins of the Fathers
(ScummVM, DOSBox)
magpi.cc/gabknight

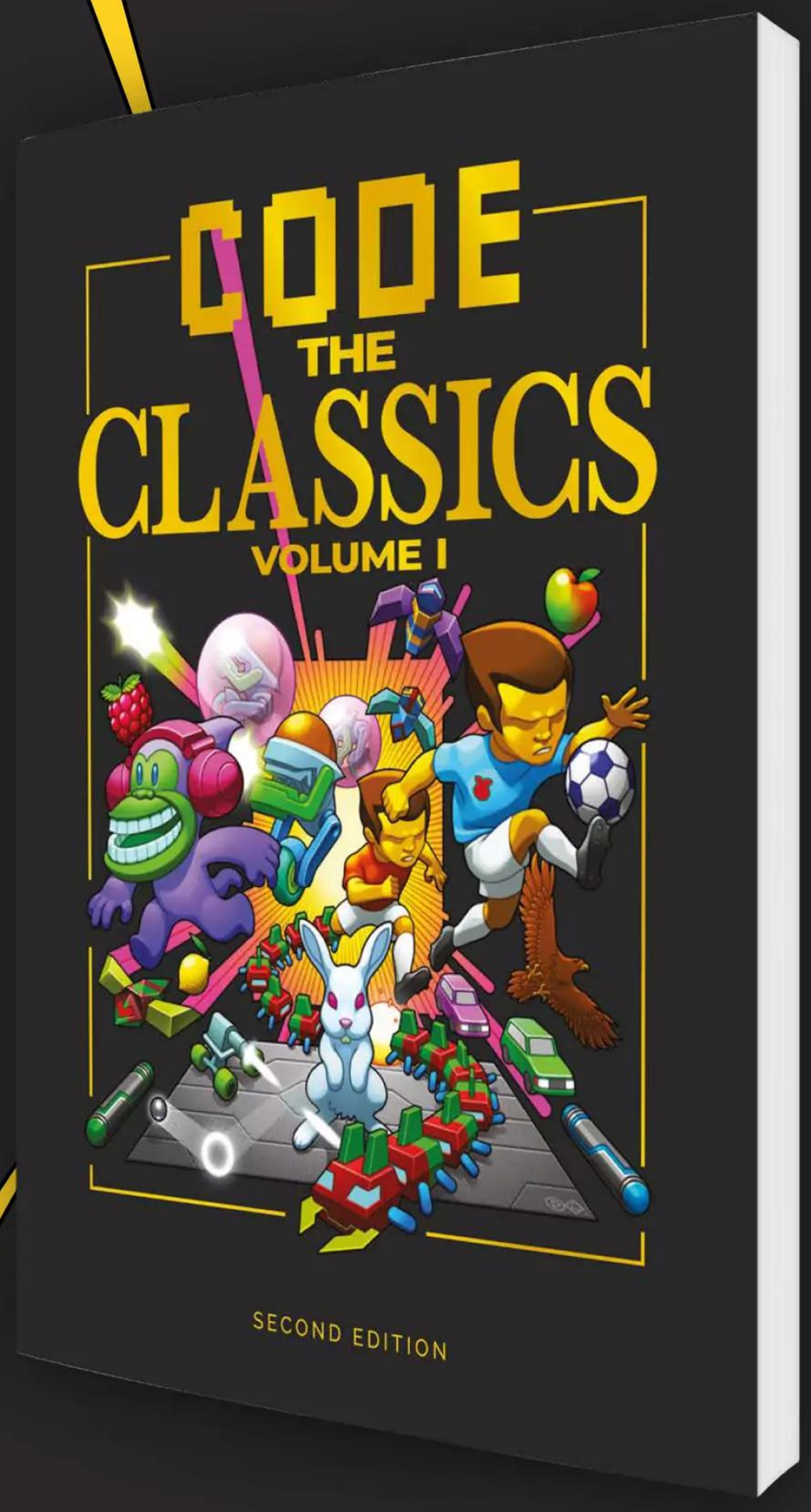
Sanitarium
(ScummVM)
magpi.cc/sanitarium

Realms of the Haunting
(DOSBox)
magpi.cc/rohgog

Killing Time
(DOSBox)
magpi.cc/killtime

I Have No Mouth And I Must Scream
(ScummVM, DOSBox)
magpi.cc/nomouthgog





New in the Second Edition

- *Expanded Python and Pygame Zero tutorials*
- *A GitHub tutorial for working with example code*
- ***Bug fixes and other improvements***

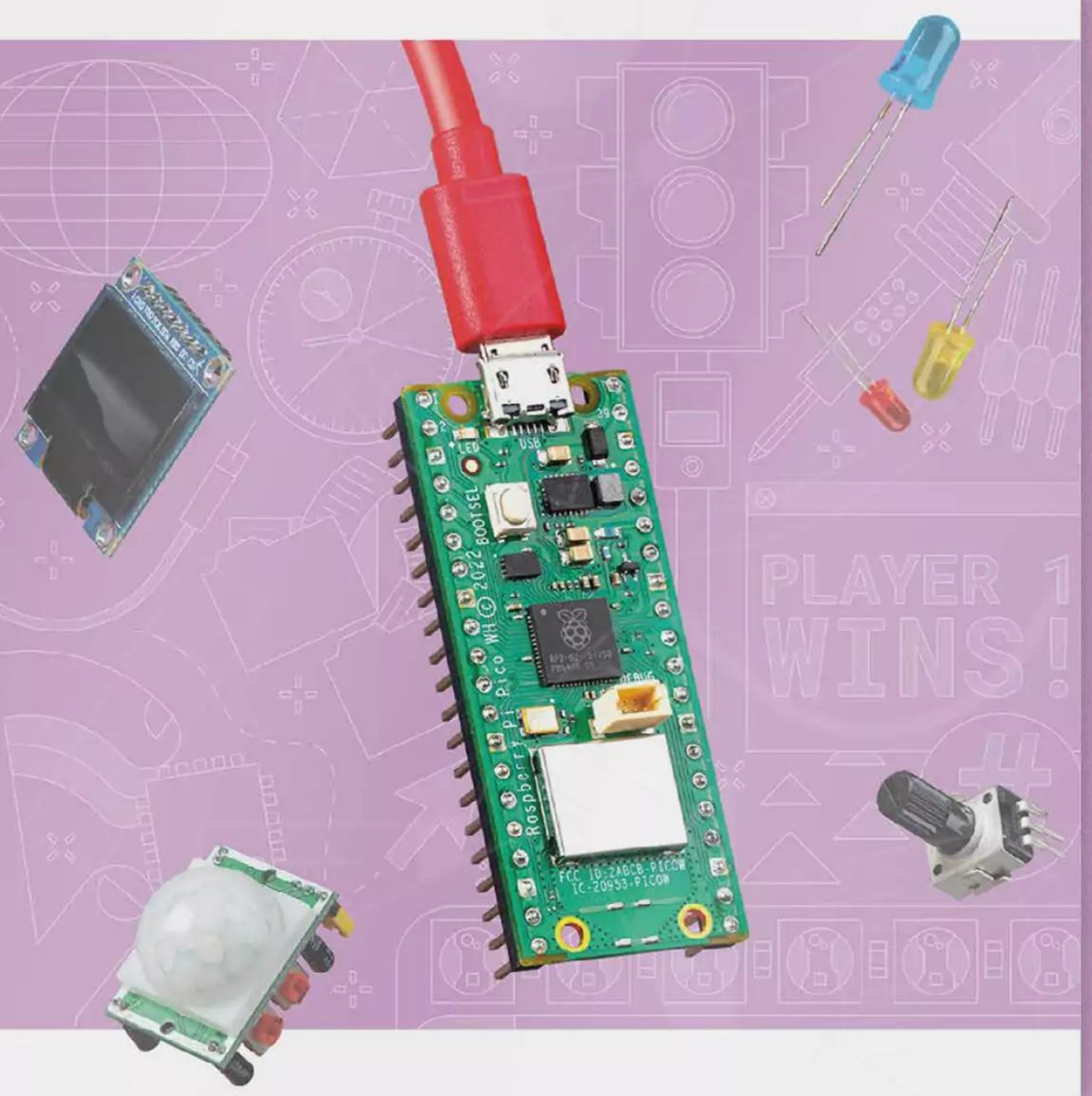
This stunning 241-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.

Available now: **magpi.cc/store**

2nd Edition
Updated for
Raspberry Pi Pico W

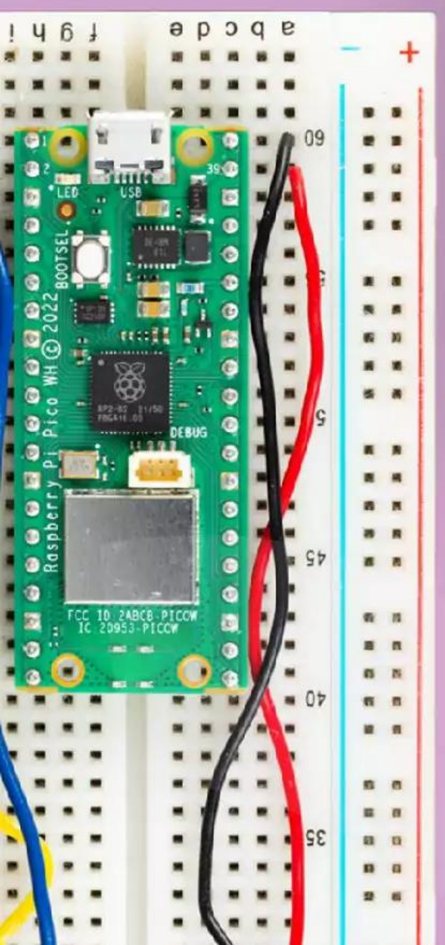
Get started with MicroPython on Raspberry Pi Pico

The Official Raspberry Pi Pico Guide



Raspberry Pi PRESS

Gareth Halfacree,
Ben Everard





Get Started with MicroPython on Raspberry Pi Pico

The Official Raspberry Pi Pico Guide



Learn how to use your new Raspberry Pi Pico microcontroller board and program it using MicroPython. Connect hardware to make your Pico interact with the world around it. Create your own electro-mechanical projects, whether for fun or to make your life easier

- *Set up your Raspberry Pi Pico and start using it*
- *Start writing programs using MicroPython*
- *Control and sense electronic components*
- *Discover how to use Pico's unique Programmable IO*

BUY ONLINE: **magpi.cc/picobook**

Get started with Raspberry Pi Pico-series and VS Code

Attach a Raspberry Pi Pico-series device and start development with the new VSCode extension



MAKER

Nate Contino

Nate is a retrofuturist and writes documentation for Raspberry Pi.

lambdalatitudinarians.org

You'll Need

- Pico-series device
  magpi.cc/pico
  magpi.cc/pico2
- Micro USB cable
  magpi.cc/microusb

The following tutorial assumes that you are using a Pico-series device; some details may differ if you use a different Raspberry Pi microcontroller-based board. Pico-series devices are built around microcontrollers designed by Raspberry Pi itself. Development on the boards is fully supported with both a C/C++ SDK, and an official MicroPython port. This article talks about how to get started with the SDK, and walks you through how to build, install, and work with the SDK toolchain.

To install Visual Studio Code (known as VS Code for short) on Raspberry Pi OS or Linux, run the following commands:

```
$ sudo apt update
$ sudo apt install code
```

On macOS and Windows, you can install VS Code from magpi.cc/vscode. On macOS, you can also install VS Code with brew using the following command:

```
$ brew install --cask visual-studio-code
```

The Raspberry Pi Pico VS Code extension helps you create, develop, run, and debug projects in Visual Studio Code. It includes a project generator with many templating options, automatic toolchain management, one-click project compilation, and offline documentation of the Pico SDK. The VS Code extension supports all Raspberry Pi Pico-series devices.

Install dependencies

On Raspberry Pi OS and Windows no dependencies are needed.

Most Linux distributions come preconfigured with all of the dependencies needed to run the extension. However, some distributions may require additional dependencies.

The extension requires the following:

- Python 3.9 or later
- Git
- Tar
- A native C and C++ compiler (the extension supports GCC)

You can install these with:

```
$ sudo apt install python3 git tar build-essential
```

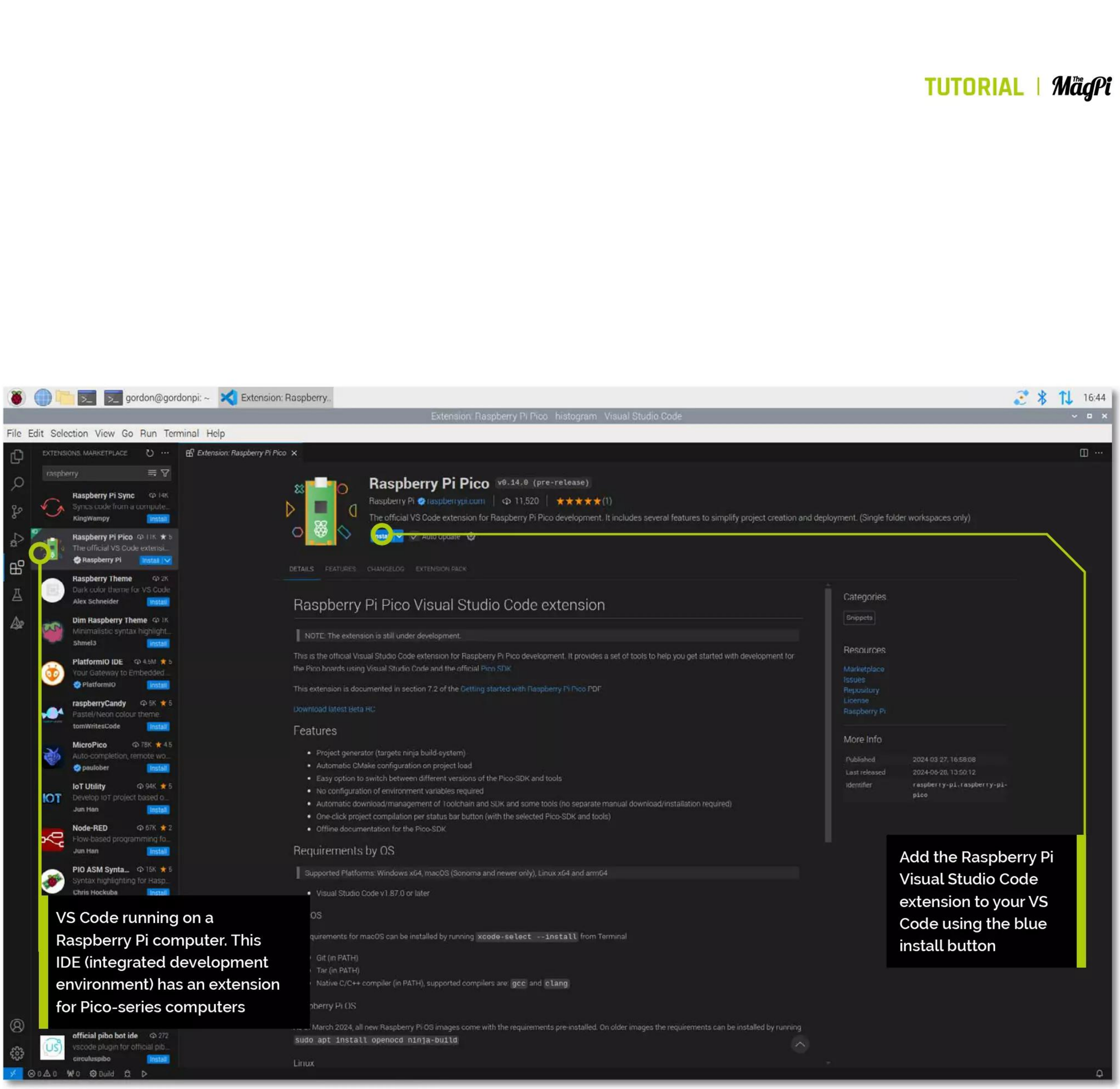
On macOS

To install all requirements for the extension on macOS, run the following command:

```
$ xcode-select --install
```

This installs the following dependencies:

- Git
- Tar
- A native C and C++ compiler (the extension supports GCC and Clang)

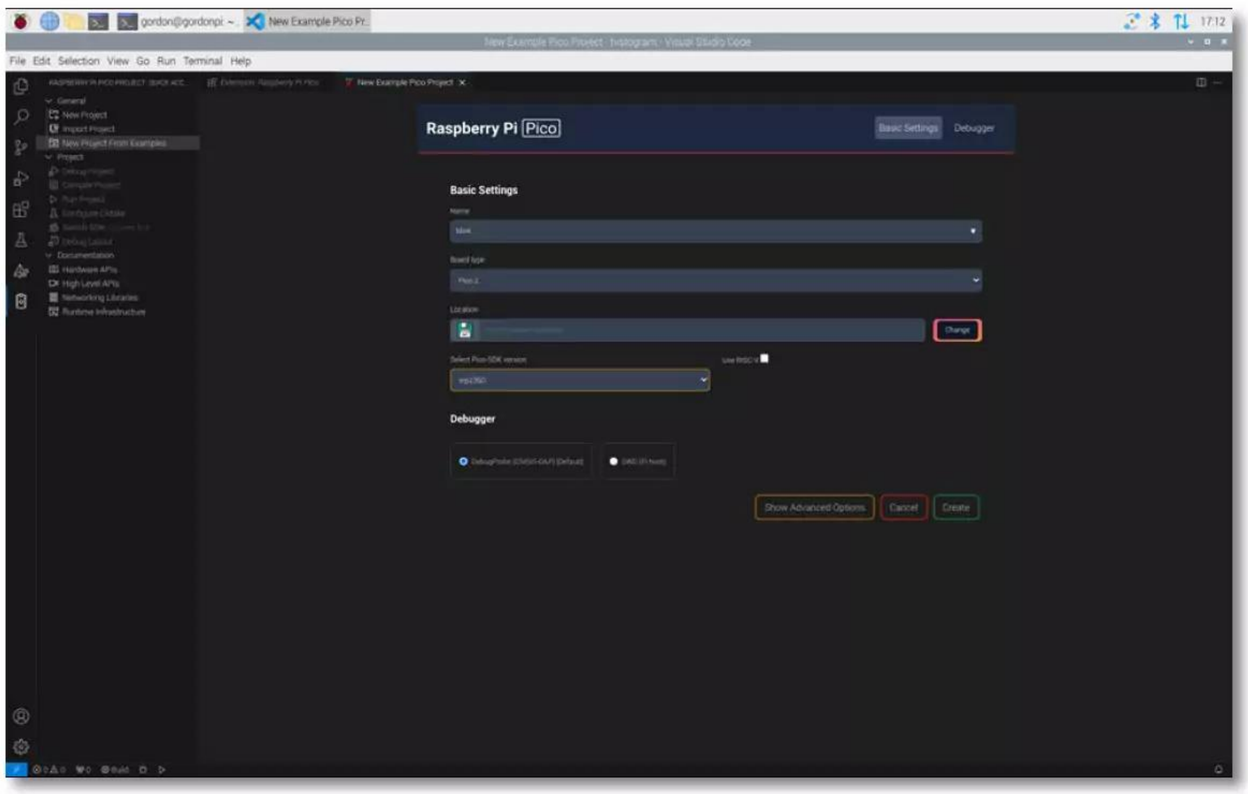


> Most Linux distributions come with all of the dependencies needed

Install the extension

You can find the extension in the VS Code Extensions Marketplace. Search for the Raspberry Pi Pico extension, published by Raspberry Pi. Click the Install button to add it to VS Code.

You can find the store entry at **magpi.cc/vscodeext**. You can find the extension source code and release downloads at **magpi.cc/picovscodegit**. When installation completes, check the Activity sidebar (by default, on the left side of VS Code). If installation was successful, a new sidebar section appears with a Raspberry Pi Pico icon, labelled "Raspberry Pi Pico Project".



▲ **Figure 2:** Creating a project in VS Code



Top Tip

Read the docs

For more information on Raspberry Pi Pico series and RP2350 see the official documentation (**magpi.cc/docs**). Also, be sure to check out our book: Get Started with MicroPython on Raspberry Pico (**magpi.cc/store**).

Load and debug a project

The VS Code extension can create projects based on the examples provided by Pico Examples. For an example, we'll walk you through how to create a project that blinks the LED on your Pico-series device:

1. In the VS Code left sidebar, select the Raspberry Pi Pico icon, labelled Raspberry Pi Pico Project.
2. Select New Project from Examples.
3. In the Name field, select the blink example.
4. Choose the board type that matches your device.
5. Specify a folder where the extension can generate files. VS Code will create the new project in a sub-folder of the selected folder.

◀ Create code to blink the LED on a Pico 2 board

6. Click Create to create the project. The extension will now download the SDK and the toolchain, install them locally, and generate the new project. The first project may take five to ten minutes to install the toolchain. VS Code will ask you whether you trust the authors because we've automatically generated the .vscode directory for you. Select yes.

The CMake Tools extension may display some notifications at this point. Ignore and close them.

On the left Explorer sidebar in VS Code, you should now see a list of files. Open **blink.c** to view the blink example source code in the main window. The Raspberry Pi Pico extension adds some capabilities to the status bar at the bottom right of the screen:

- Compile. Compiles the sources and builds the target UF2 file. You can copy this binary onto your device to program it.
- Run. Finds a connected device, flashes the code into it, and runs that code.

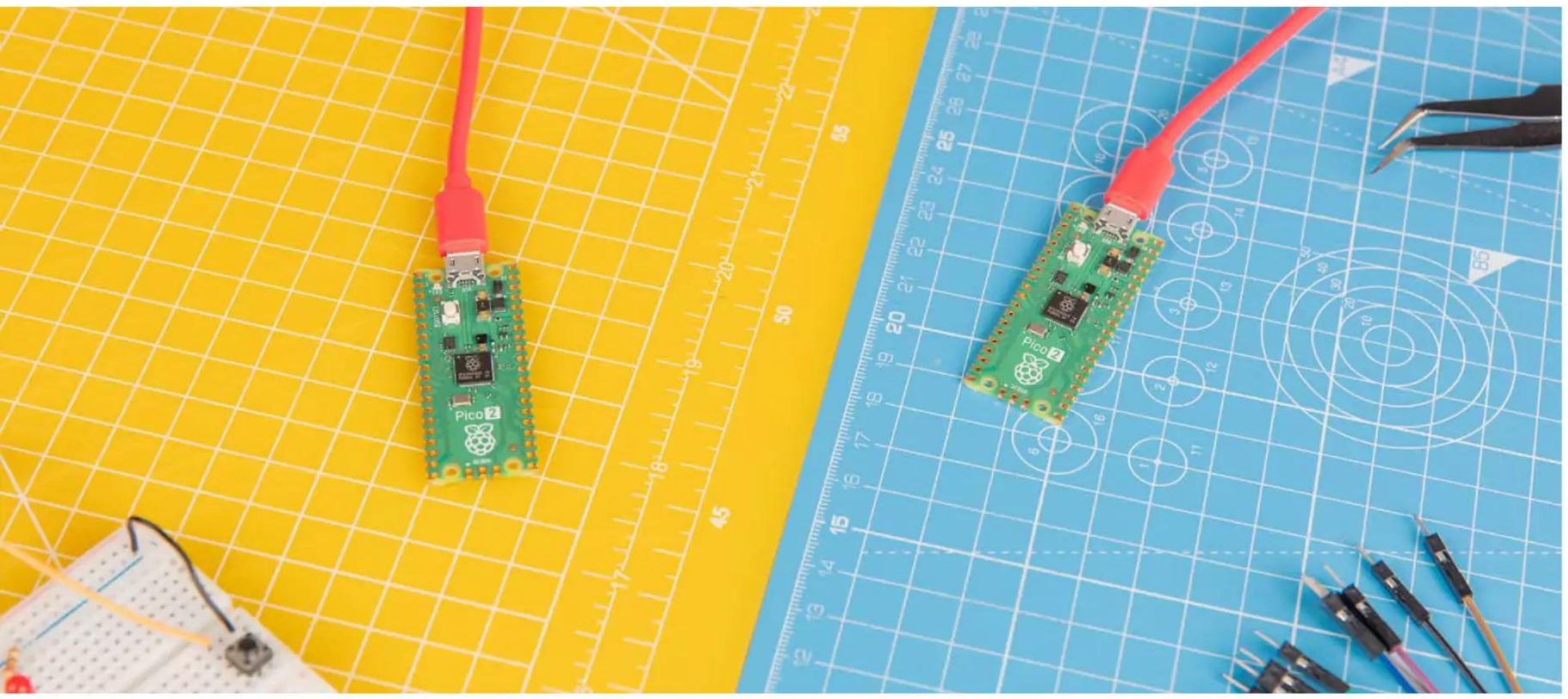
The extension sidebar also contains some quick access functions. Click on the Pico icon in the side menu and you'll see Compile Project. Hit Compile Project and a terminal tab will open at the bottom of the screen displaying the compilation progress.

> "The Pico extension adds some capabilities to the status bar"

Compile and Run blink

To run the blink example:

1. Hold down the BOOTSEL button on your Pico-series device while plugging it into your development device using a Micro USB cable to force it into USB Mass Storage Mode.
2. Press the Run button in the status bar or the Run project button in the sidebar. You should see the terminal tab at the bottom of the window open. It will display information



Pico's Micro USB connector makes sending code easy

concerning the upload of the code. Once the code uploads, the device will reboot, and you should see the following output: The device was rebooted to start the application. Your blink code is now running. If you look at your device, the LED should blink twice every second.

Make a Code Change and Re-run

To check that everything is working correctly, click on the **blink.c** file in VS Code. Navigate to the definition of LED_DELAY_MS at the top of the code:

```
#ifndef LED_DELAY_MS
#define LED_DELAY_MS 250
#endif LED_DELAY_MS
```

Change the 250 (in ms, a quarter of a second) to 100 (a tenth of a second): `#ifndef LED_DELAY_MS #define LED_DELAY_MS 100 #endif LED_DELAY_MS`

3. Disconnect your device, then reconnect while holding the BOOTSEL button just as you did before.
4. Press the Run button in the status bar or the Run project button in the sidebar. You should see the terminal tab at the bottom of the window open. It will display information concerning the upload of the code. Once the code uploads, the device will reboot, and you should see the following output:

```
The device was rebooted to start the
application.
```

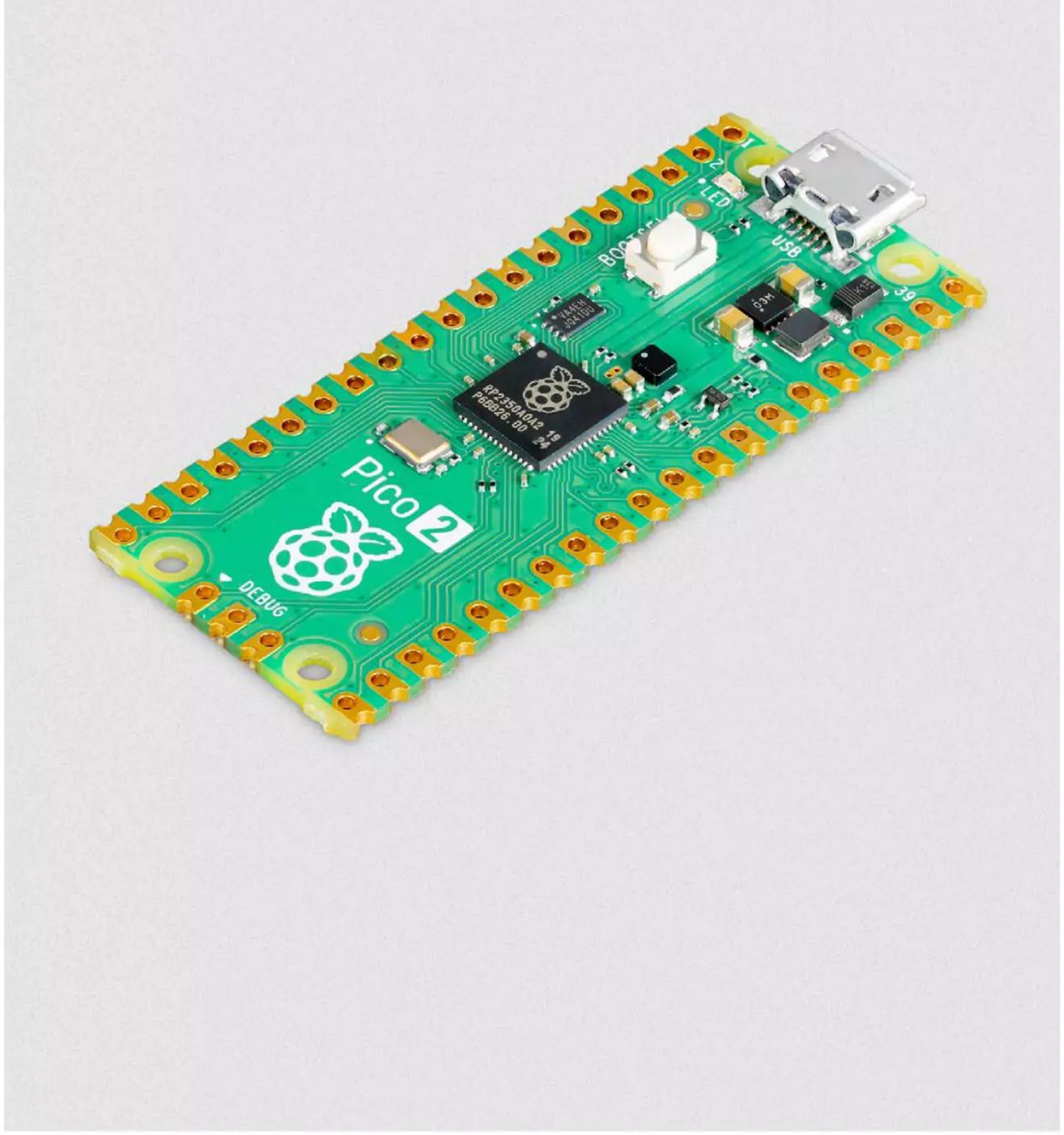
Your blink code is now running. If you look at your device, the LED should flash faster, five times every second.

Top Tip

Read the online guide

This tutorial also features in the Raspberry Pi Datasheet: Getting started with Pico. It also features information on using Raspberry Pi's Debug Probe. **magpi.cc/getstartedpico**

The new Raspberry Pi Pico 2 has upgraded capabilities over the original model



Part 03

Build a private cloud server: share with friends

Share files with friends, and let them share with you, with no service provider required



MAKER

PJ Evans

PJ is a writer, software engineer and home server enthusiast. (At least, his server's AI tells him he is)

mastodon.social/@mrpjevans

Previously, we have built our network-attached storage server and set up VPN access to your files from anywhere in the world. Now we're going to take it to the next logical step with a way to easily share your files with family and friends and let them upload files to you as well. We'll do this by installing a private server platform to your existing NAS and then adding new functionality. This will form the basis of the extra services we can add. So let's get started with the world of Docker and CasaOS.

01 Introducing CasaOS

Rather than all those fiddly command line instructions, we're going to install CasaOS (**casaos.zimaspace.com**) a dedicated solution for running home cloud servers. It uses Docker containers, a way of wrapping apps up in their own dedicated OS, to make installing server applications a breeze. Everything is done using their web-based interface, so it's a lot friendlier than a blinking cursor. We'll be using CasaOS from now on, so even if you don't want to implement file sharing, make sure you follow this section.

02 Reconfigure OpenMediaVault

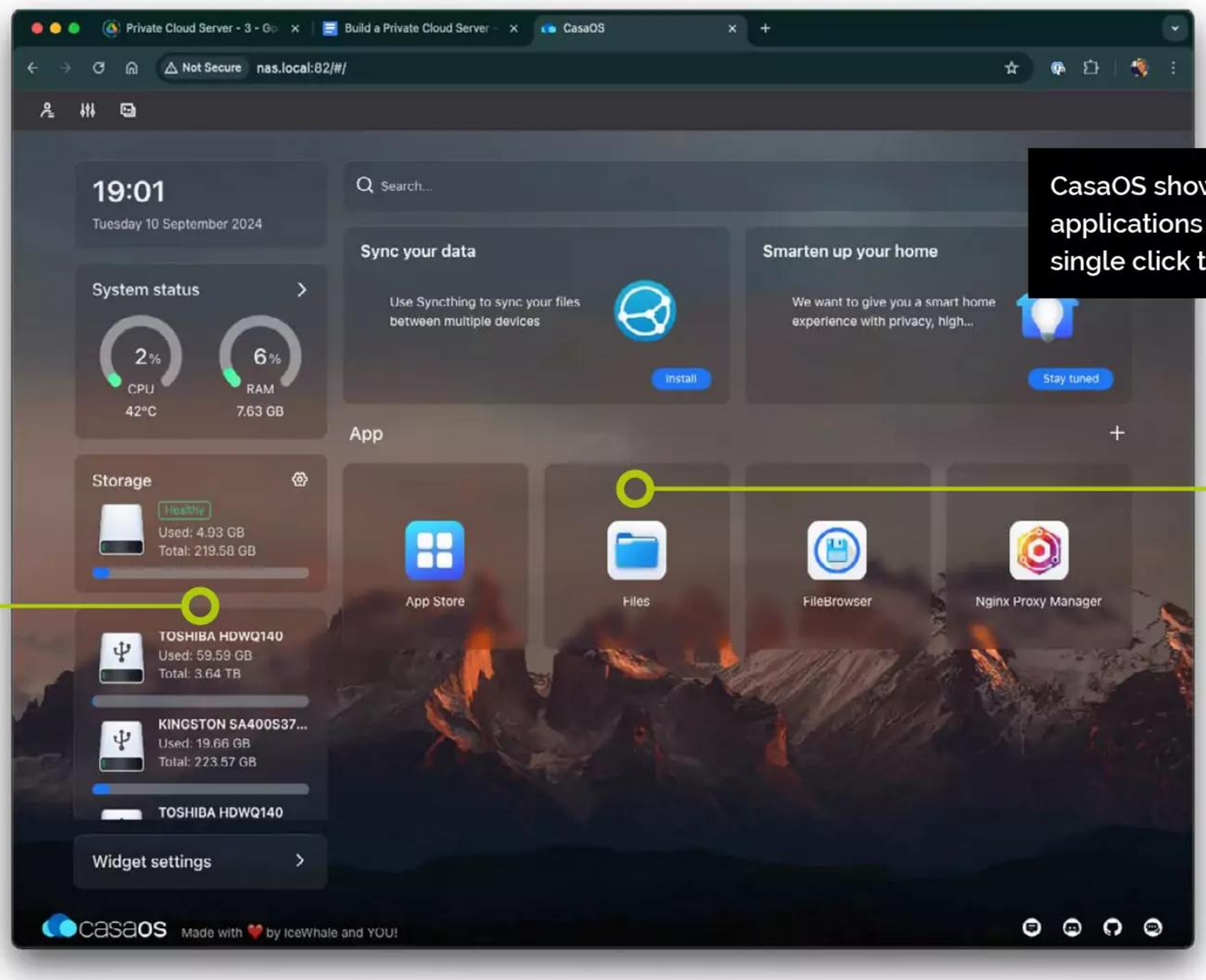
CasaOS installed its web server on port 80, which means it will clash with OpenMediaVault. We have to move OMV first. Open a web browser and go to your server (ours was called 'nas' so we went to **http://nas.local**). Login to OMV and select 'System' then 'Workbench'. Change the value of 'Port' from 80 to 83. Click 'Save' and apply the settings when prompted. You'll eventually get a red failure message. Don't worry this is just because you've moved the port. Browse to **http://nas.local:83/** and you'll be back.

03 Install CasaOS

CasaOS acts as a dashboard for all your self-hosted apps, making installation and maintenance as easy as possible. To install, start by getting a terminal window open (or using SSH) and updating using:

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

Once that has completed, run the following terminal command:



CasaOS shows all your installed applications as 'tiles' with a single click to get access

CasaOS also provides useful information on your Raspberry Pi's performance. Everything is highly customisable

```
$ curl -fsSL https://get.casaos.io | sudo bash
```

There's a lot to install, so it may take a few minutes based on your internet connection. If all is well, you'll be informed that CasaOS' web server is running. You can now visit the original URL of **http://nas.local** and you should see the CasaOS welcome screen.

04 Install FileBrowser

From the welcome screen you'll be invited to set up your account. Then you'll find yourself on the dashboard. You can immediately see that your CPU, disks and network are being monitored. In the main panel are Files and App Store. Files enables you to access your entire storage including lower-level configuration files. What we're interested in is the App Store. Click the App Store tile, then search for 'FileBrowser' and click Install. Within a few seconds, you'll be returned to the dashboard, and a shiny new tile will have appeared.

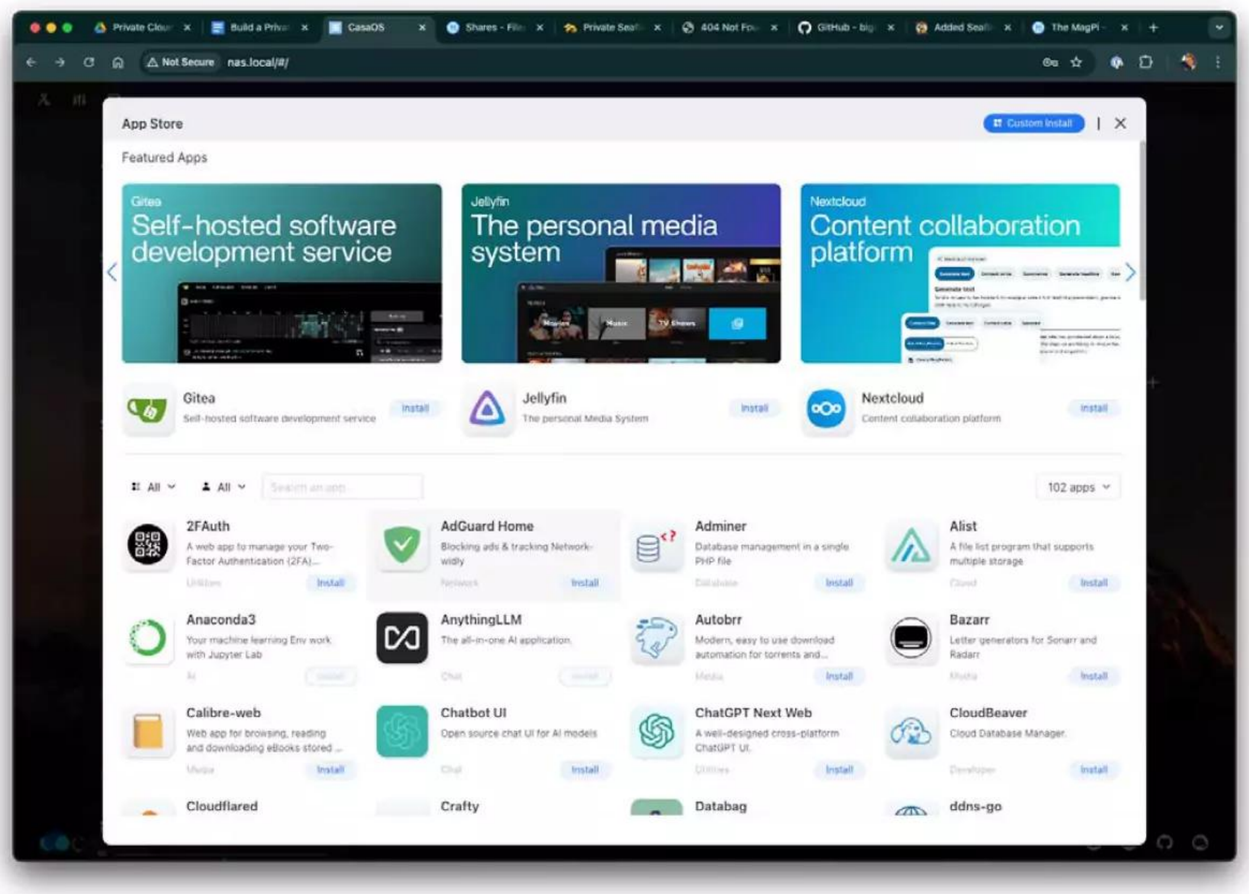
- CasaOS' App Store provides instant installation of many self-hosted services. You can add other repositories too

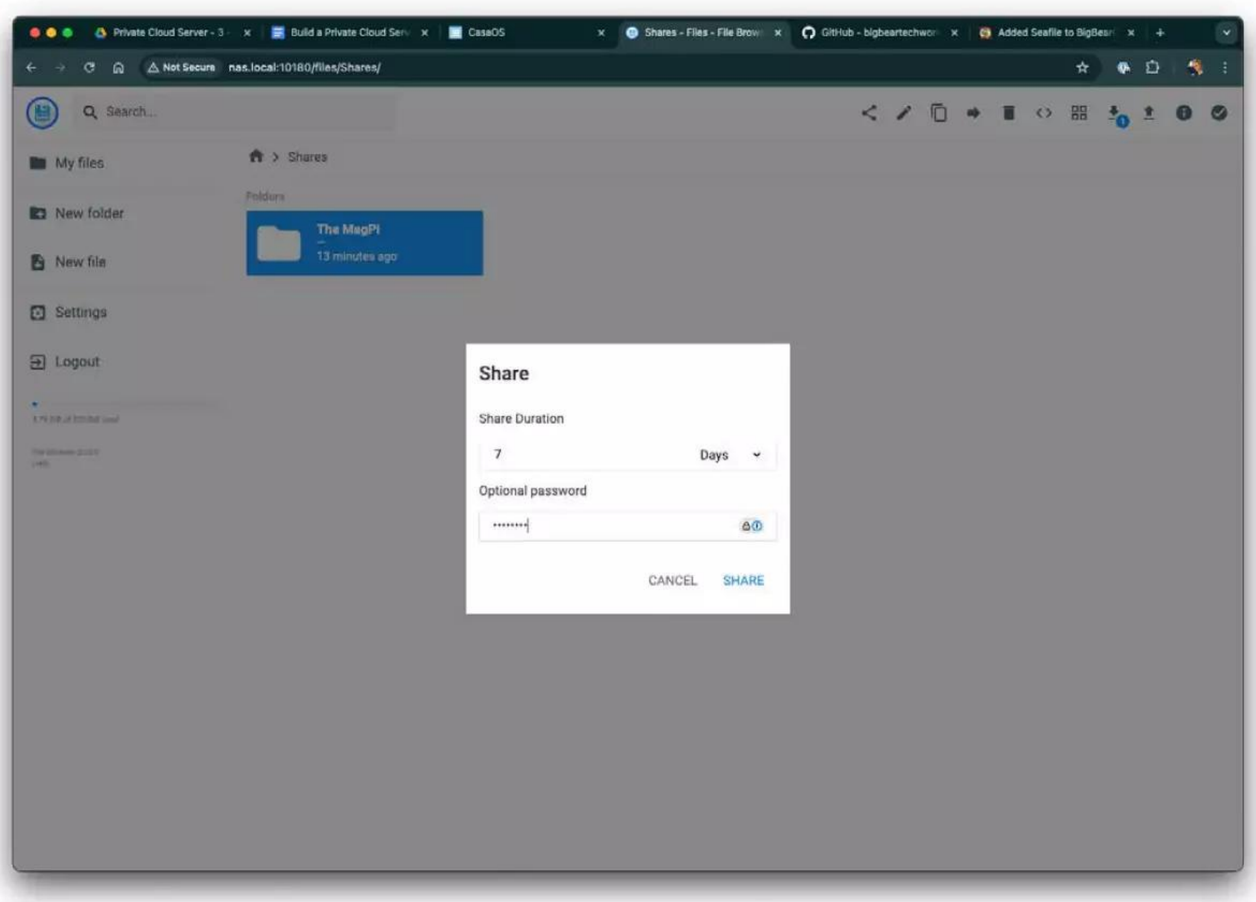
05 Set up a share space

To access FileBrowser, just click the tile. Both the username and password are 'admin' (you can change this in settings and we strongly recommend you do). You'll see a number of folders that correspond to CasaOS' **DATA** directory. Click 'New Folder' and call it 'Shares'. This can be the home for all your shared data. We created another folder in this called **The MagPi**. Go into the directory and you can upload files and folders using the top toolbar. You can create as many directories as you like and sub-directories too.

You'll Need

- Private cloud server (Issue #144) **magpi.cc/144**
- Admin access to your internet router
- A domain name (Issue #145) **magpi.cc/145**





Top Tip

Adding your own tiles

You can use CasaOS' tiles to add your own bookmarks. For example, you can create a shortcut to OMV on port 23.

06 Share and share alike

You've now got something you want to share, but how to do so? There are two ways to share files. The quick way is to navigate to a folder and click the Share icon on the top toolbar. Here you can set an expiry time for the share and optionally set an access password. Click Share, and a unique link will be generated that you can share with whoever you like. For longer-term shares go to Settings then User Management. Here you can create multiple users and restrict them to their own spaces. They can then log in and be taken straight to their shared directory. However, there's a problem...

07 Getting access

Your server is behind your router, which acts like a firewall, so your friends and family have no access to your server. To give them access, and to do it securely, we're going to have to make some changes. This is only advisable if you need to share a lot of files. If it's infrequent and with people you trust, you might want to consider giving them VPN access to your network using WireGuard (**magpi.cc/145**). If you're happy exposing a service to the internet, we're going to have to install a SSL (aka TLS) certificate to ensure traffic is encrypted so passwords and data cannot be sniffed.

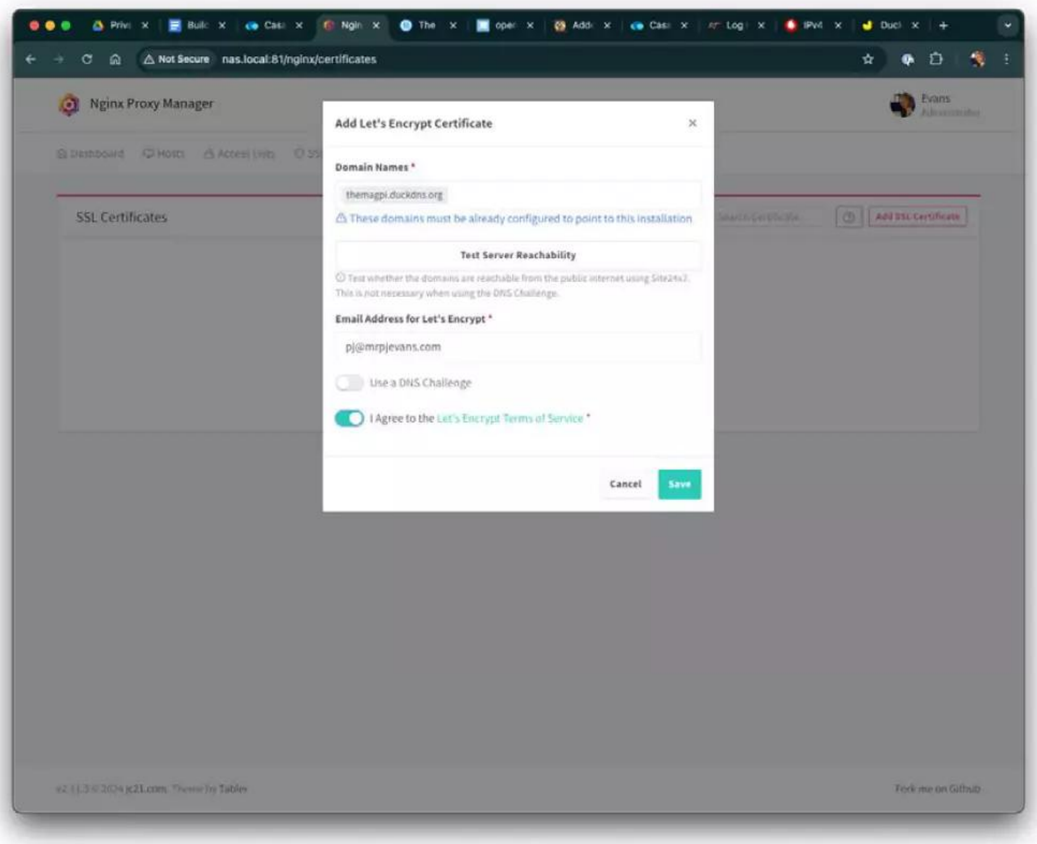
> "We're going to have to install a SSL (aka TLS) certificate to ensure traffic is encrypted"

File Browser is a simple server app that makes sharing files with others easy and secure

08 Moving time again

To be able to handle a TLS certificate, we need to use Nginx (engine-x), a web server that can proxy connections to our existing services. It must run on port 80 so again we need to move things around. We're going to move CasaOS to port 82 so 80 is freed up. On the CasaOS dashboard, click the settings icon in the top-left corner. On the panel that appears, change the WebUI port from 80 to 82. As before, your connection will now break. Go to your equivalent of **http://nas.local:82/** to get back in. Now go to the App Store, search for 'nginx' and install Nginx Proxy Manager.

The Nginx Proxy Server app takes the pain out of creating a proper SSL-encrypted web service



09 Router configuration

Nginx is now our gatekeeper and running on ports 80 (plaintext) and 443 (encrypted). We now need to route traffic from your Internet router to Nginx. This varies by router: it normally entails setting up port forwarding TCP traffic from ports 80 and 443 on the router to the same ports on your private cloud server. Once implemented, the outside world can reach your Nginx server using the domain name you configured in *The MagPi* #145, but there's nothing to see yet bar a 'Congratulations' message. In CasaOS, click on the Nginx tile to get to its control panel. The username is 'admin@example.com' and the password 'changeme'. You'll be guided through changing these.

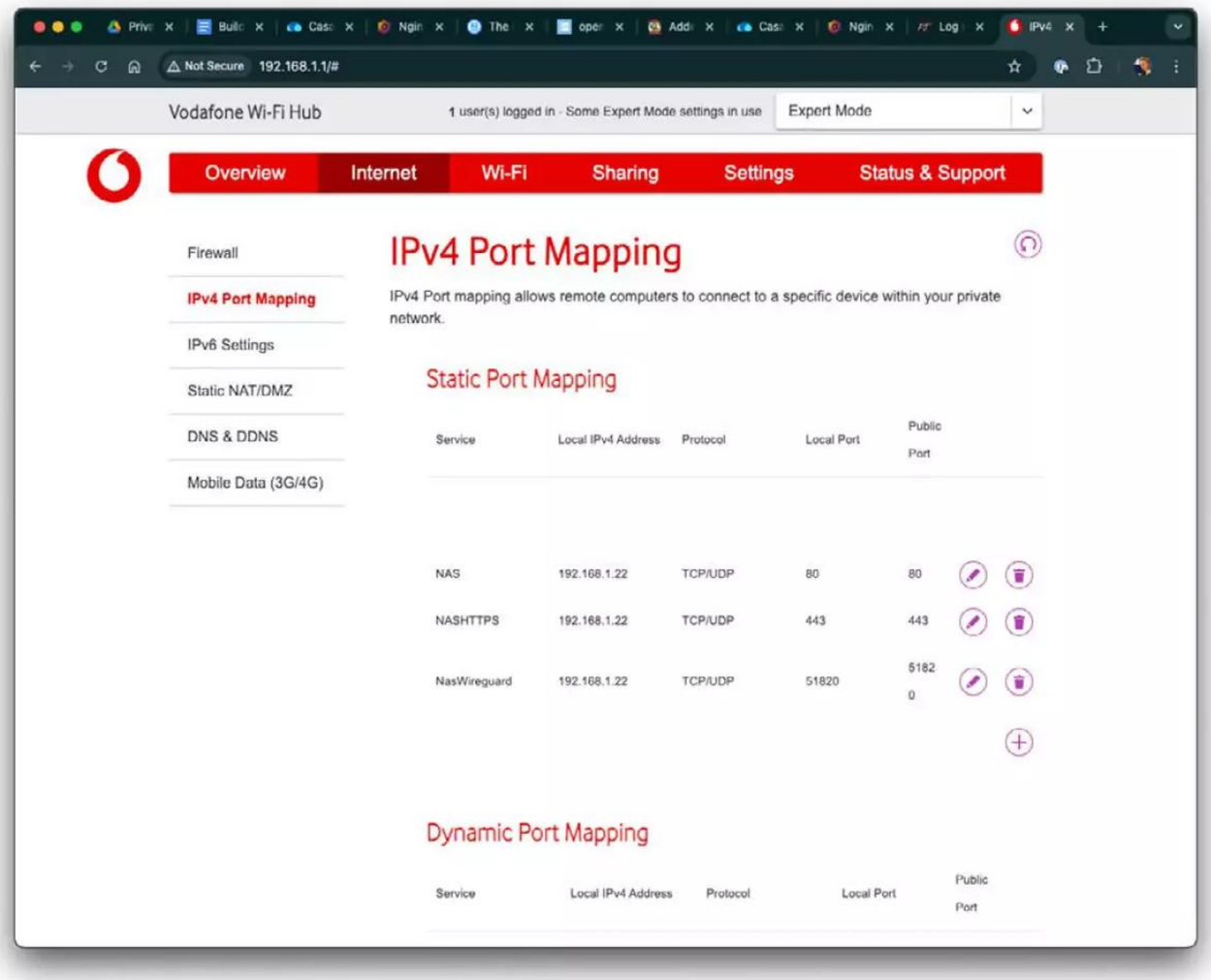
10 Get certified

Acquiring an SSL certificate used to be a complicated process costing hundreds of pounds. Thanks to the amazing work of Let's Encrypt, it's now free and takes seconds. In the control panel click SSL Certificates then Add SSL Certificate. In the panel, add your domain name and check your server is reachable. Add your email address and accept the terms and conditions. Click 'Save' and the process will begin. This may take a few minutes to complete. Once done, you have a certificate and can configure secure access.

11 Proxy work

The final step is to proxy your new secure connection to FileBrowser. Staying on the Nginx Proxy Manager configuration page, click on Hosts then Proxy Hosts. Click Add Proxy Host then complete the form. Add your public domain name at the top, leave the scheme as 'http', add the IP address of your server next followed by port 10180 (the default port for File Browser). Check Block Common Exploits then go to the SSL tab and select the certificate you just created. Now save. If you now go to your domain name prefixed with 'https://', you should be taken straight to File Browser. You now have an encrypted service to share with your friends, family and colleagues.

You will need to forward ports from your router to your server. Here's an example for a Vodafone router



"We've added CasaOS, installed File Browser, and provided secure access"

12 Summary and next steps

Let's look at what's been done. We've added CasaOS which will be endlessly helpful for adding new services. We installed File Browser so we could share files with others, we then used router port forwarding and Nginx to provide secured access to File Browser using an SSL certificate. Please note that File Browser can now be 'seen' by anyone who comes across it, and if you don't plan on using it for a while, consider disabling the port forwarding. Next time, we start building on our new infrastructure with some business apps and advanced collaboration tools.

Top Tip

Other sharing apps

File Browser is admittedly simple and great for uncomplicated uses. Check out Seafile and Project Send if you want something with more features.

Part 02

MARS Rover: autonomous driving

Get your all-terrain Raspberry Pi rover to drive by itself and detect obstacles

DOWNLOAD THE FULL CODE:
magpi.cc/marsrov2

MAKER

Phil King

A long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

Loosely based on NASA's Curiosity rover on Mars, the 4tronix MARS Rover is a six-wheeled robot with steerable wheels and a rocker-bogie suspension system that enables it to crawl over small rocks.

In the first instalment (MagPi #145), we covered how to assemble the rover and control it manually from a remote computer using key presses. This time, we'll get it to move autonomously – first in preset patterns, and then intelligently, using the ultrasonic sensor on its mast to detect obstacles and take evasive action. We can even get it to navigate a simple maze.

If you have a different wheeled robot, the same concepts can be applied (using similar functions in GPIO Zero) to program its movements and reactions – see the CamJam EduKit #3 worksheets for some examples: **magpi.cc/edukit3guide**.



▲ Test the ultrasonic distance sensor by moving your hand in front of it to see the measured distance change

01 SSH in

As before, you'll need to connect remotely to the rover from another computer. After powering it up, wait a little while for Raspberry Pi Zero W to connect to the Wi-Fi before SSHing in from another computer – in a Terminal window, enter:

```
ssh [username]@[IP address]
```

Then change to the `marsrover` directory and list its contents:

```
cd marsrover
ls
```

You'll see the existing code files that were downloaded when you installed the software library, which is named **rover.py**.

02 Create a file

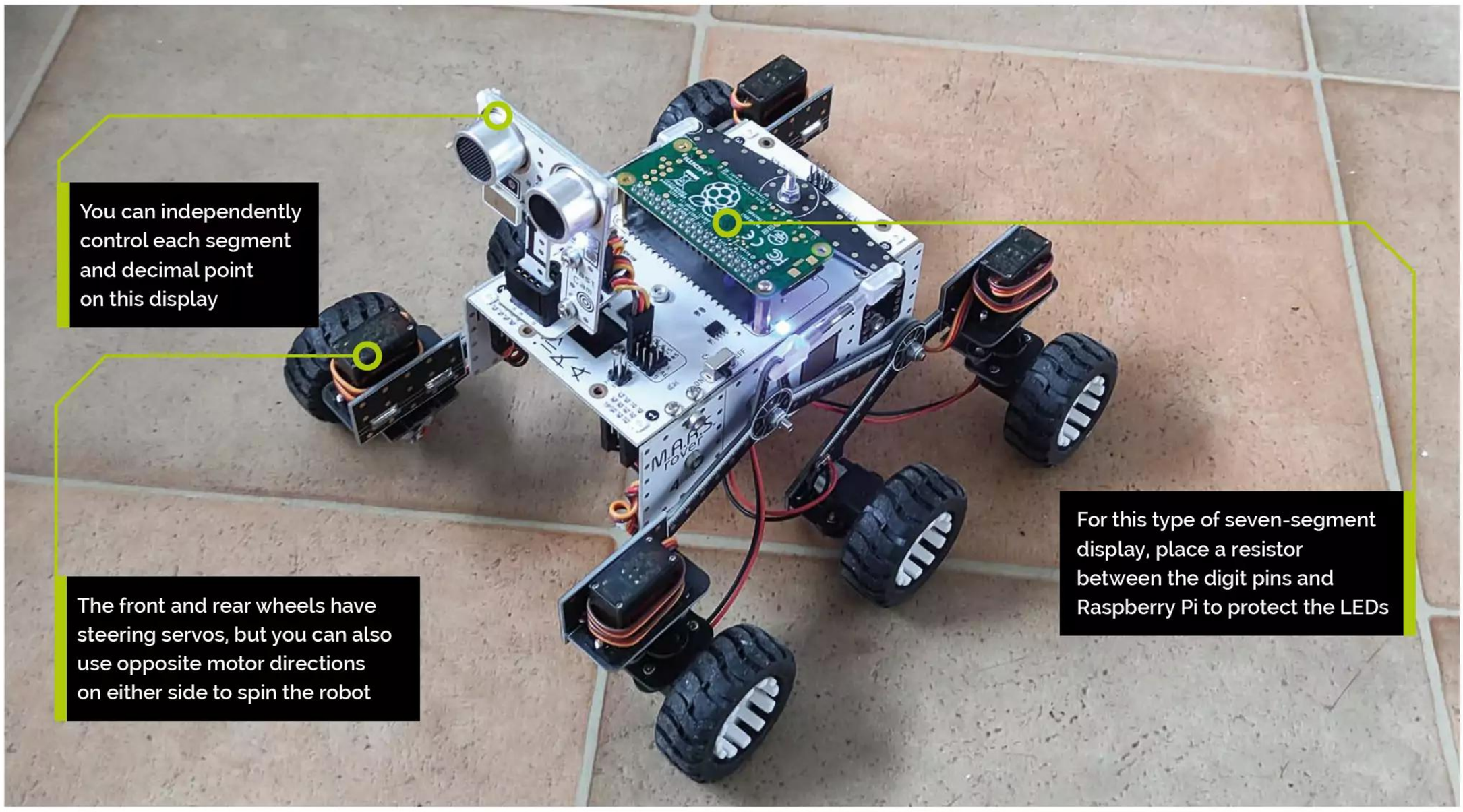
To start coding, create a new Python file and open it in the nano text editor:

```
nano mars_patterns.py
```

Then, whenever you want to exit and save the file, press **CTRL**+**X** then **Y**.

Alternatively, you can write your code on the remote computer and transfer it via SFTP. To do that, open a connection to the rover with:

```
sftp [username]@[IP address]
```



The password is the same as for SSH. Change directory with `cd marsrover`, then use the put command to send a file from the remote computer. For example:

```
put mars_patterns.py
```

To send a file the other way (from rover to computer), use the `get` command.

03 Start coding

In our code file, we start by importing the rover and time libraries:

```
import rover, time
```

We then assign variables to the servos, based on the numbers of their connections, and define functions for forward, reverse, and turn left and right. You can copy these code blocks from **driveRover.py** if you like, then add functions for spin left and right:

```
def spinL():
    rover.setServo(servo_FL, 0)
    rover.setServo(servo_FR, 0)
    rover.setServo(servo_RL, 0)
    rover.setServo(servo_RR, 0)
    rover.spinLeft(speed)

def spinR():
    rover.setServo(servo_FL, 0)
    rover.setServo(servo_FR, 0)
    rover.setServo(servo_RL, 0)
    rover.setServo(servo_RR, 0)
    rover.spinRight(speed)
```

Spinning, by reversing one side's wheels while moving the other side's forward, enables the rover to turn far more quickly than with the steering servos.

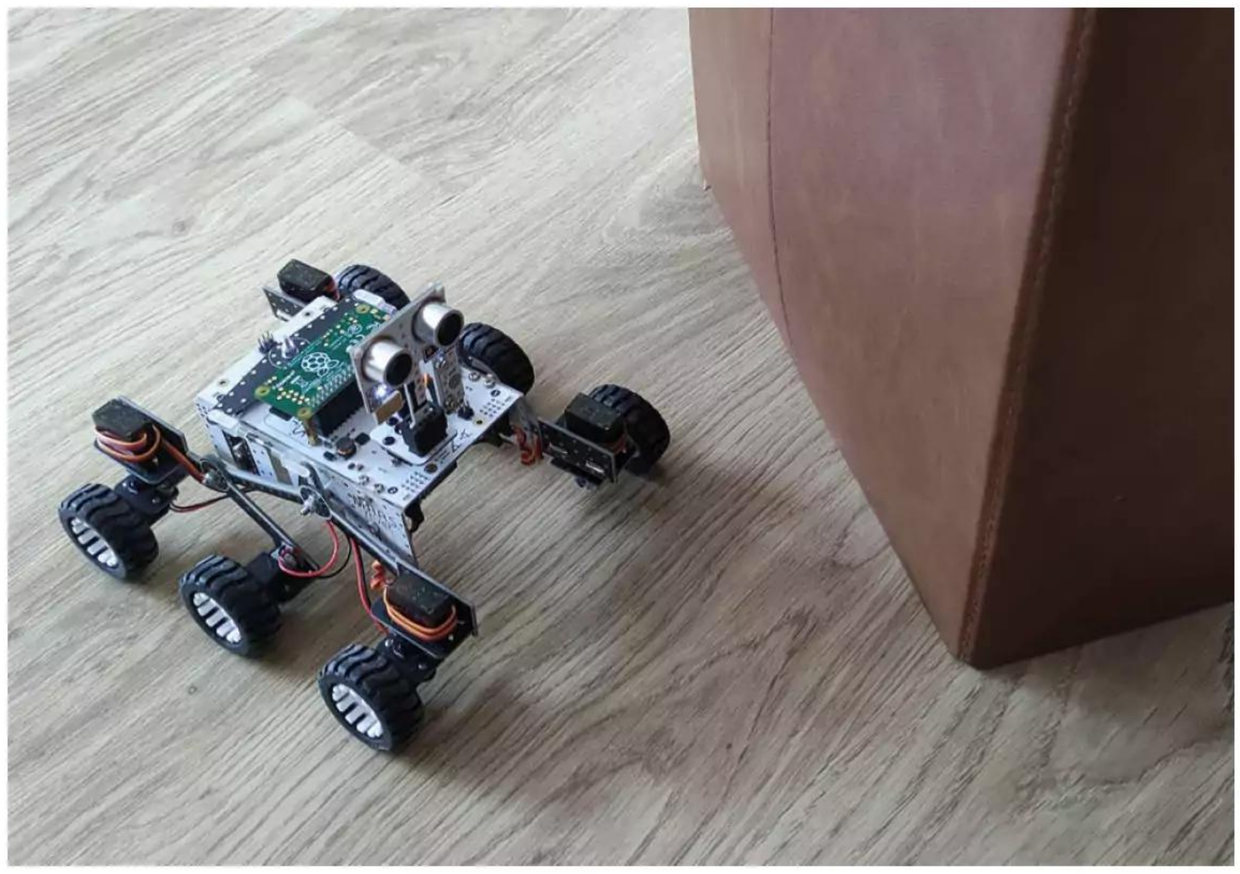
We also need to set the default speed and initialise the rover's GPIO pins, LEDs, and motors:

You'll Need

- MARS Rover for Raspberry Pi Zero kit **magpi.cc/marsrover**
- Raspberry Pi Zero W or Zero 2 W
- microSD card with Raspberry Pi OS (Legacy 32-bit)
- MARS Rover software (see part 1)
- 4 × AA batteries

The MARS Rover can clamber over smaller obstacles, but it'll need to detect and avoid larger ones





▲ Once its ultrasonic sensor detects a nearby obstacle ahead, the rover reverses a little and then turns to avoid it

```
speed = 60
rover.init(0)
```

04 Set a pattern

We can now create some code to move the rover in a pattern. Let's start simple, moving it forward, back, then turning left and right (while going forward). We'll place the pattern in a function:

```
def pattern():
    goForward()
    time.sleep(2)
    goReverse()
    time.sleep(2)
    goForward()
    goLeft()
    time.sleep(2)
    goRight()
    time.sleep(2)
    rover.stop()
```

▶ Getting a robot to navigate a simple maze is a good test of your problem-solving skills

The `time.sleep` value sets the delay and therefore how long the rover will move in the direction above it. To run the function, call it with:

```
pattern()
```

After saving the code, run it from the command line with:

```
python mars_patterns.py
```

05 Full circle

Let's create another function to move the rover in a circle. This is simple to do, by steering it left or right continually as it moves forward.

```
def circle():
    goForward()
    goRight()
    time.sleep(40)
    rover.stop()
```

Again, we call the function with its name and two brackets:

```
circle()
```

The `time.sleep` value needed to complete a full circle will depend on the speed setting. Try running the code and adjust the number as required.

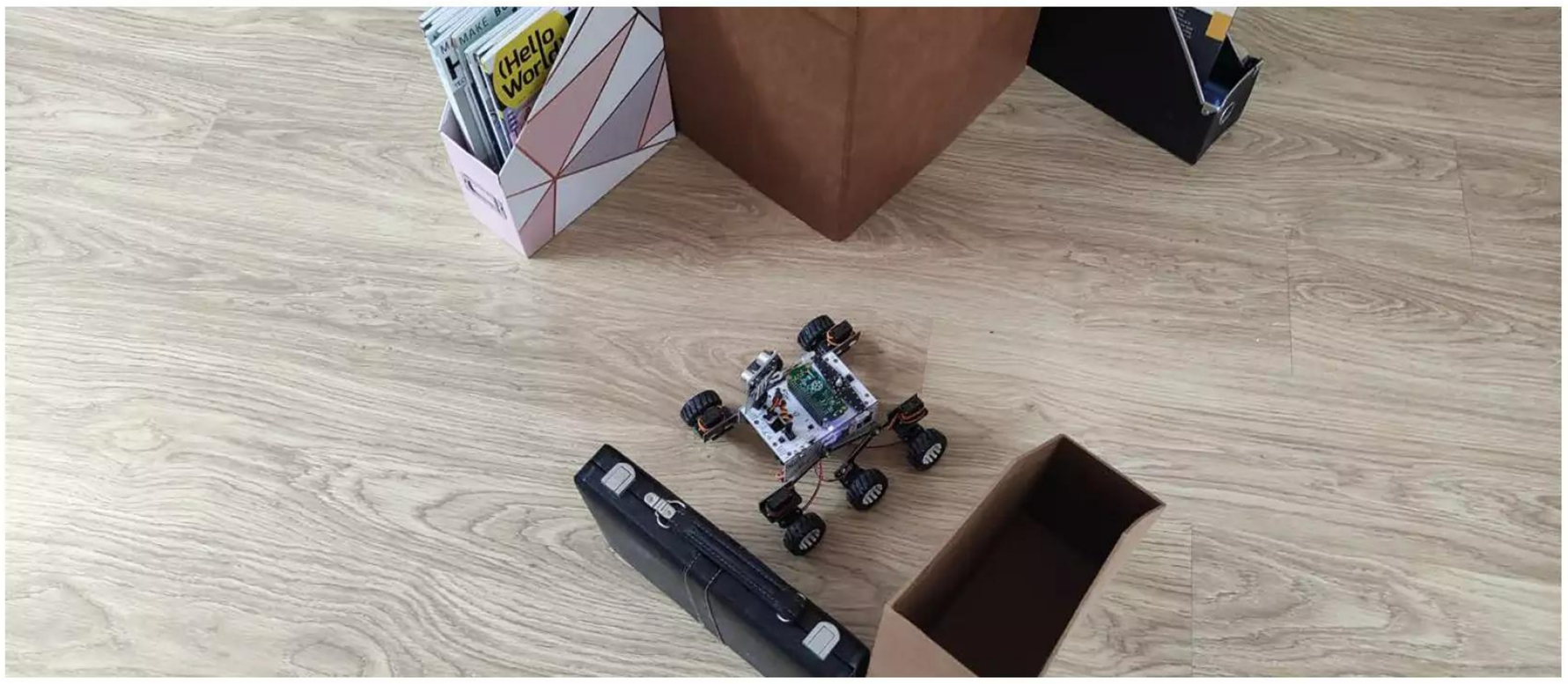
06 Square dance

Let's try another simple pattern, getting the rover to move in a square. Again we create a function:

```
def square():
    for i in range(4):
        goForward()
        time.sleep(5)
        spinR()
        time.sleep(2.5)
    rover.stop()
```

This contains a definite loop (`for`) that repeats four times (set by the range number), moving the rover forward for a certain time, then spinning it for a set time. Note that the latter number will vary according to the speed setting, so a bit of trial and error is needed to get it close to 90° for each corner of the square.





By swivelling the rover's mast, we can detect walls on either side before deciding how to proceed

The final code for all three patterns can be found in the downloadable **mars_patterns.py** file. If you only want to run one pattern, comment out the others' function call lines by adding a `#` to the front.

07 Obstacle detection

The rover is equipped with an ultrasonic sensor on its mast. This measures the distance to any object in front of it. Run the **sonarTest.py** code and put your hand in front of the sensor, then move it nearer and further away to see the measured distance change.

While the rover can clamber over smaller obstacles, we can use the sensor to detect larger ones. Create a new file, **mars_obstacle.py**, and start with the same blocks of initial code as for **mars_patterns.py**, including movement functions and `speed = 60` and `rover.init()` lines. At the bottom, add some new variables:

```
hownear = 15.0
reversetime = 2
turntime = 2.5
```

The first, `hownear`, sets how near (in cm) the rover needs to be to an obstacle to take evasive action, for which the other two variables determine the time spent reversing and then turning.

Let's create a function to take a sensor reading and print it to the command line:

```
def obstacle(localhownear):
    dist = rover.getDistance()
    print ("Distance: ", (int(dist * 10)) /
10.0)
    if dist < localhownear:
        return True
    else:
        return False
```

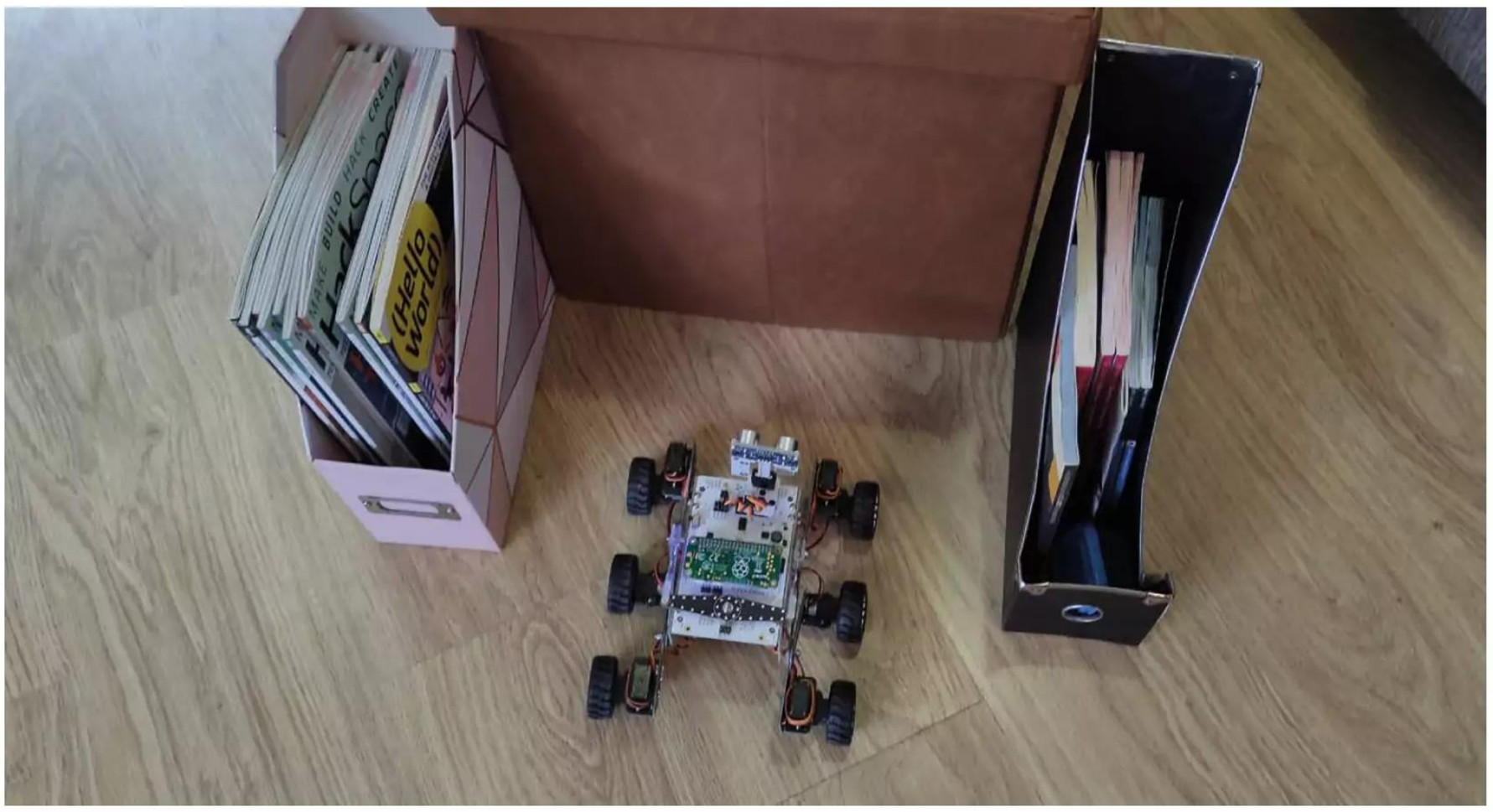
Note that this function is taking an input parameter (we're calling it `localhownear`), which we'll assign later when calling it. It then returns True or False depending whether the sensed distance is less than that value.

08 Evasive action

We need to create a function to determine how the rover will move to avoid the obstacle.

```
def avoid():
    goReverse()
    time.sleep(reversetime)
    rover.stop()
    spinR()
    time.sleep(turntime)
    rover.stop()
```

In this case, we reverse a little, then get it to spin right. We can alter the `reversetime` and `turntime` variable values at the top of the program to adjust how far it reverses and turns.



If the rover reaches a dead end, we'll get it to reverse up before spinning left or right

09 Moving forward

We get the rover to move forward continually, taking evasive action when needed.

```
try:
    # repeat the next indented block forever
    while True:
        goForward()
        time.sleep(0.1)
        if obstacle(hownear):
            rover.stop()
            avoid()
```

In this endless loop (`while True:`), we check the sensor at 0.1 second intervals and, if the obstacle function returns True (i.e. the distance is less than `hownear`), we call the `avoid` function.

Note that we've placed the loop within in a `try` block so we can stop the robot moving when we quit the program with **CTRL**+**C**:

```
except KeyboardInterrupt:
    pass
```

And finally:

```
    rover.cleanup()
```

The **obstacle.py** program is fine for detecting obstacles and avoiding directly ahead, but tends to fail if they're approached at an angle. By swivelling the sensor's mast left and right continually, we found the detection performance improved. Check out the downloadable **obstacle_mast.py** code to see the `mastservo` mast-swivelling function (with some extra variables) used.

> "By swivelling the sensor's mast, the detection performance improved"

10 Maze navigation

Getting an autonomous robot to navigate a maze is a classic test. Based on our **obstacle.py** code, we get the rover to stop once it detects a wall straight ahead, then swivel its sensor mast left and then right to detect any side walls. See the downloadable **mars_maze.py** code for details.

To determine the action taken, we modified the `obstacle` function to add more conditionals. If there is a side wall only on the left (or right), we get the rover to spin right (or left) by 90°.

We also cover the possibilities of there being side walls to both the left and right, or none. In the first case, we reverse a little. Otherwise, the action taken for both cases is the same, spinning either left or right depending on a random 'coin toss' – using `random.randit(0,1)`, which returns 0 or 1.

Our code worked pretty well in tests using a very simple maze, but see if you can improve on it and let us know.

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*
- *Learn to code retro-style games*
- *Emulate classic computers and consoles*
- *Build a console, handheld, and full-size arcade machine*

BUY ONLINE: **magpi.cc/store**

Real-time ML audio noise suppression on Raspberry Pi Pico 2

Use machine learning to create a noise-cancelling Pico 2 audio device



MAKER

Sandeep Mistry

Sandeep is a principal software engineer at Arm.

github.com/sandeepmistry

Machine Learning (ML) technologies have revolutionised the way many software applications are developed. Application developers now curate datasets with numerous example inputs and outputs for a desired system, then use these datasets to train ML models. During training, the ML model learns patterns from the inputs and outputs. Trained models are then deployed to devices which perform inferencing on inputs from the real-world and use the ML model's predicted output to perform one or more actions.

Smaller ML models that require kilobytes of memory can be deployed to microcontroller-based devices, like the Arm Cortex-M33-based Raspberry Pi RP2350 microcontroller used in the new Pico 2 board (**magpi.cc/cortexm33**). Deploying ML models to microcontrollers enables systems to have lower latency, as the data is processed on devices close to the input data source.

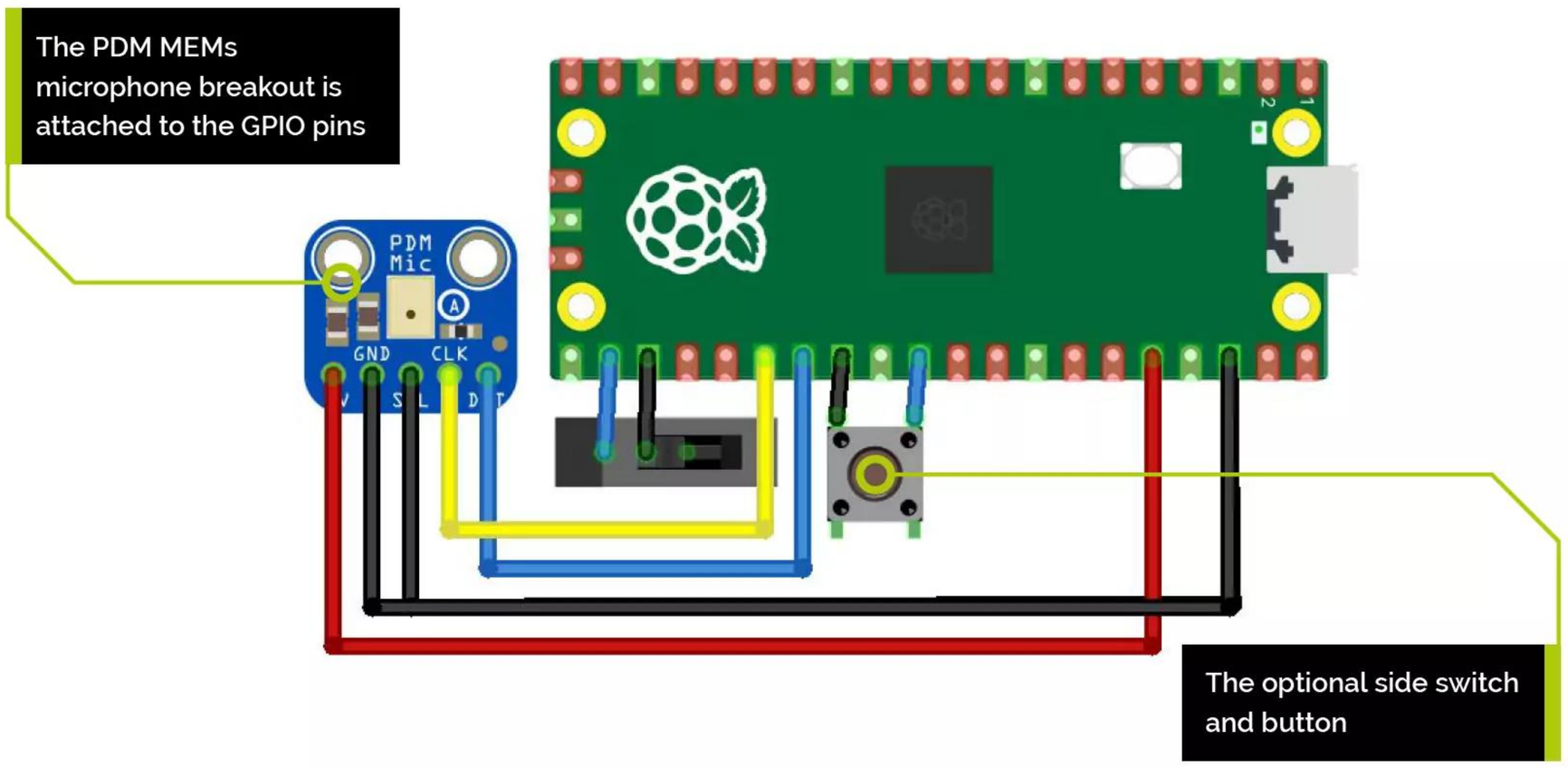
This tutorial will delve into how an existing ML-based audio noise suppression algorithm can be deployed to a Raspberry Pi RP2350 microcontroller used in the new Pico 2 board. The RP2350's dual-core Arm Cortex-M33 CPU enable application developers to deploy more compute intensive applications that exceed the performance capabilities of the RP2040 microcontroller used in the original Raspberry Pi Pico board.

The algorithm will then be integrated into the USB microphone application I developed for the original Pico board (**magpi.cc/usbmicpico**). The original application captured data from a digital pulse-density modulation (PDM) microphone and processed it into a format compatible with the USB Audio standard for transmission over USB.

In 2018, Jean-Marc Valin published a paper on "A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement" (**magpi.cc/rnnoise2018**). The paper covers how a recurrent neural network-based ML model can be used to suppress noise in an audio source. If you're interested in learning more about the algorithm, read Jean-Marc's "RNNoise: Learning Noise Suppression" page (**magpi.cc/rnnoise**). The page covers details of the algorithm and includes interactive examples. Source code for the project is available in the RNNoise Git repository (**magpi.cc/rnnoise-xiph**).

At a high level, the algorithm extracts 42 features from 10 milliseconds of a 48 kHz audio source by splitting the signal into 22 frequency bands.

The 42 features are then used as the input to a neural network which calculates gains for the 22 frequency bands. The calculated gains can then be applied to the original audio signal to produce a denoised version. The neural network also outputs a "voice activity detection" output, which indicates the predicted confidence level of a voice being present in the input signal, with values between 0 to 1.



The PDM MEMs microphone breakout is attached to the GPIO pins

The optional side switch and button

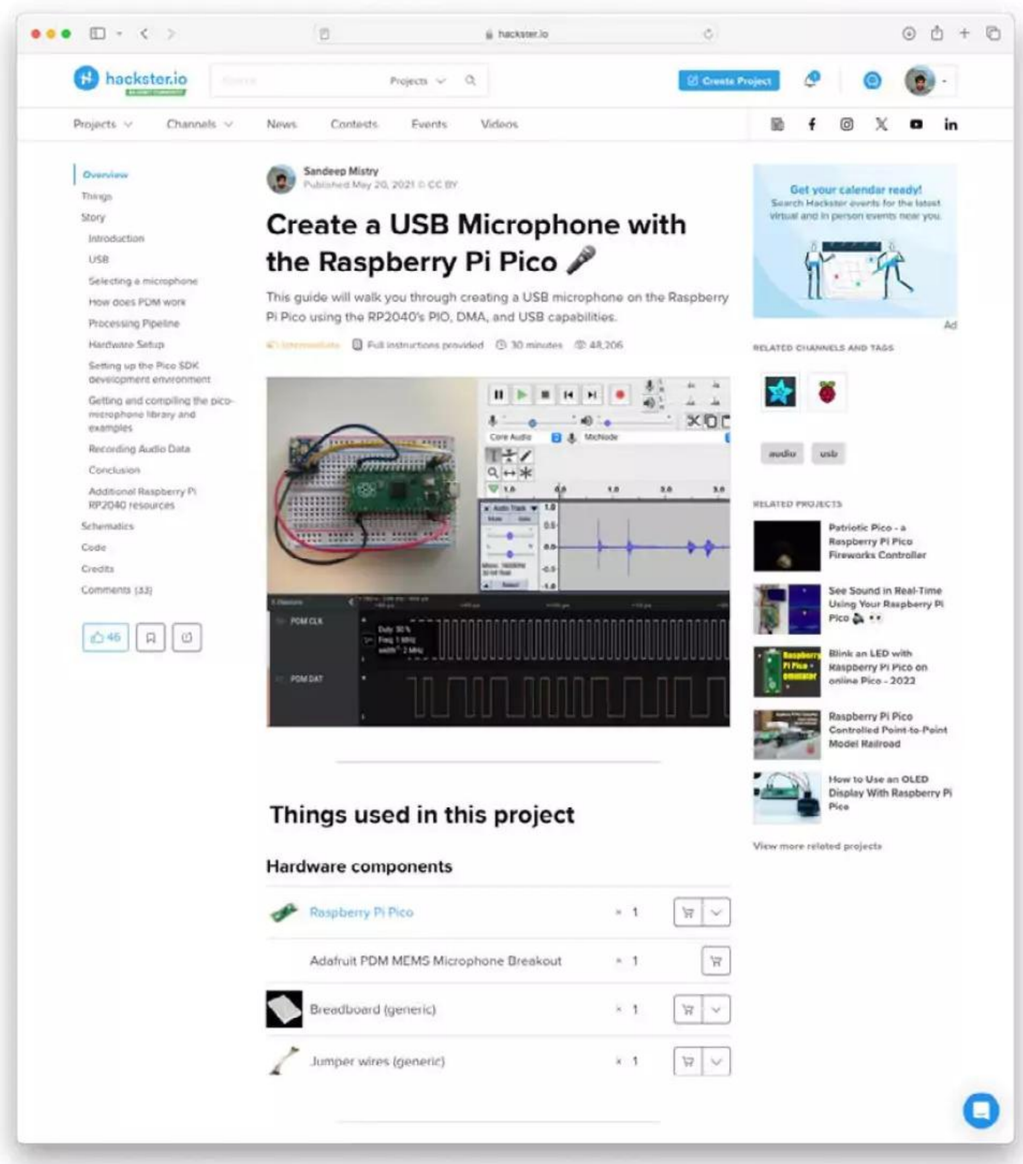
GPIO pin guide			
Raspberry Pi Pico 2	**PDM MEMS Microphone Breakout**	**Slide switch (optional)**	**Tactile button (optional)**
3V3 (OUT)	3V		
GND	GND SEL	Middle pin	Bottom pin
RUN			Top pin
GPIO21	DAT		
GPIO22	CLK		
GPIO17		Bottom pin	

You'll Need

- Raspberry Pi Pico 2 **magpi.cc/pico2**
- Adafruit PDM MEMS Microphone Breakout **magpi.cc/pdmmems**
- Breadboard and jumper wires **magpi.cc/breadboard** **magpi.cc/jumperwires**
- Slide switch (optional) **magpi.cc/slideswitch**
- Tactile button (optional) **magpi.cc/adaswitch**

> Trained models are then deployed to devices which perform inferencing

- The *Create a USB Microphone with the Raspberry Pi Pico* guide on **hackster.io**

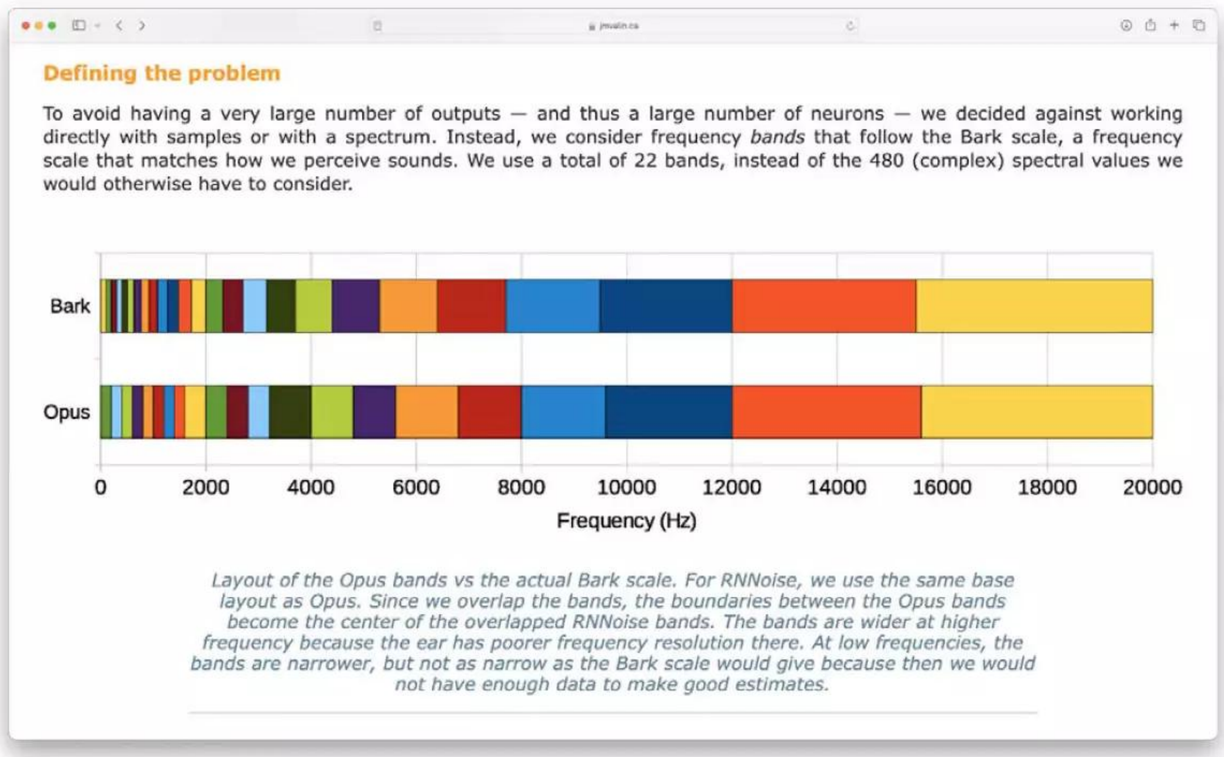


Porting and benchmarking

The original C code of the RNNoise project can be integrated into a CMake (**cmake.org**) project that uses the Raspberry Pi pico-sdk (**magpi.cc/picosdk**). All source code for the porting can be found on the rnnoise-examples-for-pico-2 (**magpi.cc/rnnoise-pico2**) repository on GitHub. A new CMake library target was created with the **celt_lpc.c**, **denoise.c**, **kiss_fft.c**, **pitch.c**, **rnn.c**, and **rnn_data.c** files from v0.1.1 of the RNNoise project.

Some minor modifications were made to **denoise.c** to use single precision floating point calculations in the `biquad` function, as well as use `log10f(…)` and `sqrtf(…)` instead of `log10(...)` and `sqrt(...)` functions.

The library can then be integrated into a



▲ *Defining The Problem* section of the *RNNoise: Learning Noise Suppression* page

benchmarking application (**magpi.cc/rnnoise-benchmark**) that calls the `rrnoise_create(...)` function to initialise the model before measuring how long the `rnnoise_process_frame(...)` function takes to process 480 samples.

This benchmarking application can be deployed to a Raspberry Pi Pico or Pico 2 board by first following sections two and nine of the "Getting started with Raspberry Pi Pico" C/C++ SDK guide (**magpi.cc/getstartedpico**). Then running the following commands to build the .uf2 application to deploy to the board:

```
git clone --recurse-submodules https://
github.com/ArmDeveloperEcosystem/rnnoise-
examples-for-pico-2.git
cd rnnoise-examples-for-pico-2
mkdir build
cmake .. -DPICO_BOARD=pico2
make rnnoise-benchmark
```

▶ **Figure 3:** The breadboard, all wired up

After compiling, the **examples/benchmark/rnnoise-benchmark.uf2** file can be deployed to the board by holding down the board's white BOOTSEL button while plugging in the USB cable to your computer and copying the .uf2 file to the Pico's USB disk.

Here are results of the benchmark on a Pico and Pico 2 board:

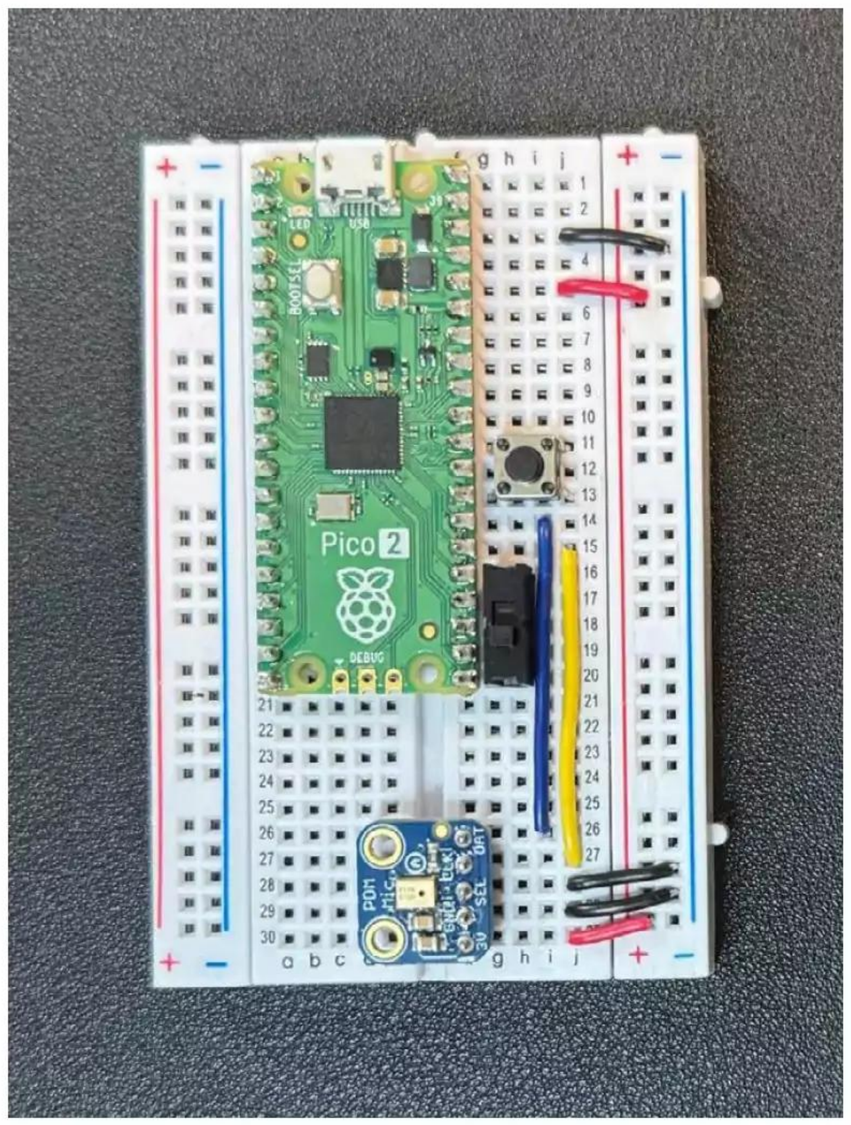
	Pico (RP2340) Cortex-M0+ @ 125 MHz	Pico 2 (RP2350) Cortex-M33 @ 150 MHz
rnnnoise_process_frame(...)	372,644 microseconds	22,093 microseconds

The original Pico takes approximately 372.6 milliseconds versus the new Pico 2's 22.1 milliseconds, this is a 16.87× speed up between boards.

Modifying the algorithm

For the board to process 480 samples of audio at a sample rate of 48 kHz it must take less than 0.01 seconds (480 / 48,000) or 10 milliseconds to complete the `rnnoise_process_frame(...)` function. Our benchmark result for the Pico 2 takes 22.1 milliseconds and is not fast enough for 48 kHz audio, but it is fast enough to process audio with a sample rate of 16 kHz, which requires that the audio be processed in under 30 milliseconds.

The `eband5ms` variable in **denoise.c** can be easily modified to adjust the algorithm for processing 16 kHz data. This variable controls the start range used in 22 frequency bands. It can be adjusted by multiplying the original values by three, as 16 kHz audio takes three times longer to collect samples than 48 kHz audio, and setting the maximum starting position to 120.



Here's the original value:

```
static const opus_int16 eband5ms[] = {
/*0  200 400 600 800  1k 1.2 1.4 1.6  2k 2.4
2.8 3.2  4k 4.8 5.6 6.8  8k 9.6 12k 15.6
20k*/
  0,  1,  2,  3,  4,  5,  6,  7,  8, 10, 12,
14, 16, 20, 24, 28, 34, 40, 48, 60, 78, 100
};
```

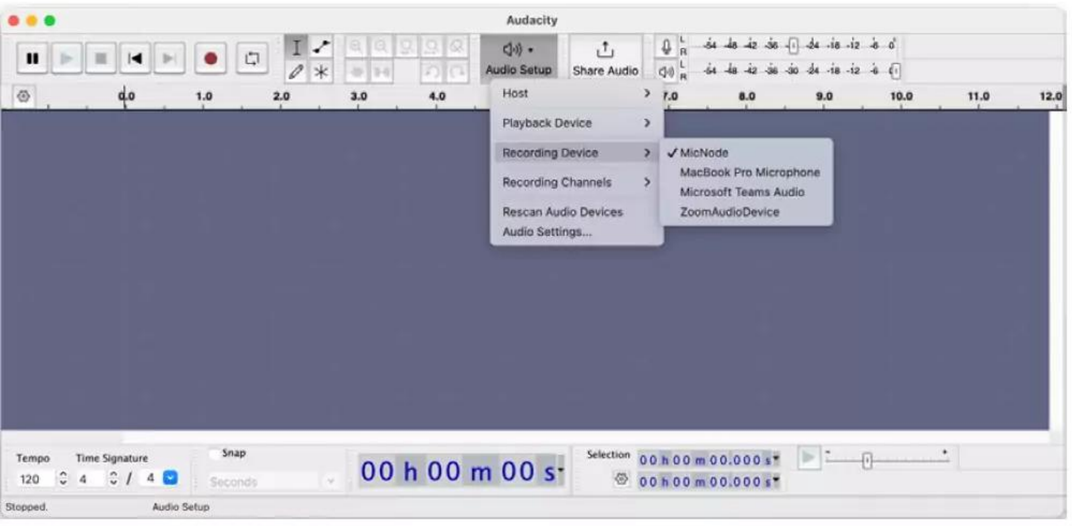
And here's the modified value to use with 16 kHz audio:

```
static const opus_int16 eband5ms[] = {
/*0  200 400 600 800  1k 1.2 1.4 1.6  2k 2.4
2.8 3.2  4k 4.8 5.6 6.8  8k 9.6 12k 15.6
20k*/
  0,  3,  6,  9,  12, 15, 18, 21, 24, 60, 36,
42, 48, 60, 72, 84, 102, 120, 120, 120, 120,
120
};
```

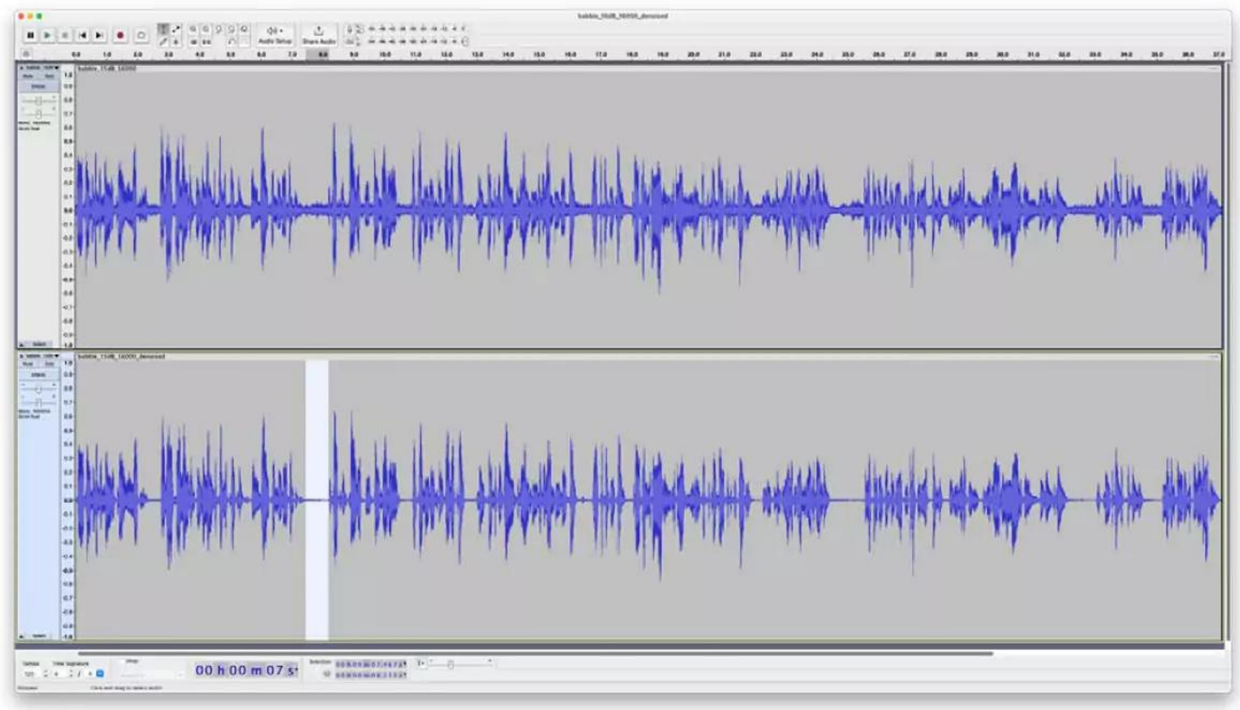
The serial (**magpi.cc/rnnoise-serial**) example can be compiled and deployed to the board to test the modified algorithm. This example loops continuously to receive 480 16-bit audio samples over USB, processes them with the denoising algorithm, and then transmits the denoised samples over USB. On a PC, the **serial_denoise.py** (**magpi.cc/serialdenoisepy**) Python script can be used to send raw 16-bit, 16 kHz audio from a file and save the denoised audio to a file.

These raw values can be imported into an application like Audacity (**audacityteam.org**) for visualisation and playback. **Figure 1** is an example, the first track is the original audio (noisy) and the bottom is the version denoised on the Pico 2.

I've selected a region where the noise is visually reduced. So far so good; the algorithm has been validated to run on the board with a 16 kHz audio source streamed from a PC over USB.



▲ Selecting the MicNode as the Recording Device in Audacity



▲ **Figure 1:** Audacity application with two audio tracks open: Top: original audio, Bottom: de-noised version of the audio

USB microphone denoising

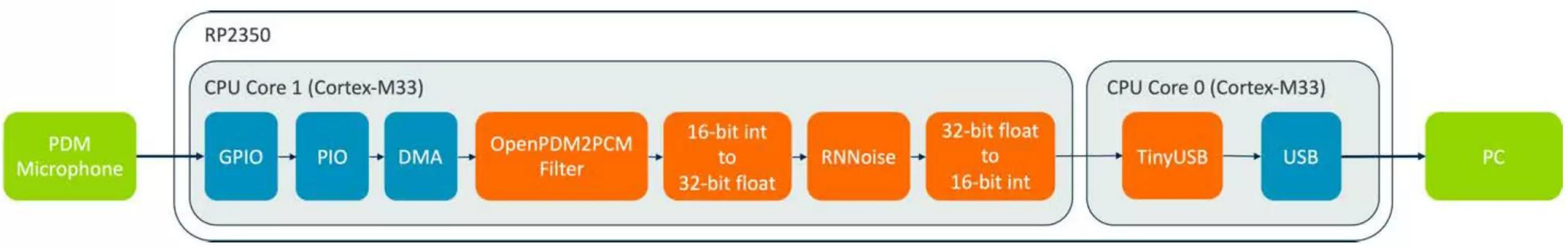
The USB microphone application originally built for the original Pico can now be enhanced with on-board denoising.

The optional slide switch will be used as a toggle to disable or enable the noise suppression processing at run-time. While the optional tactile switch will provide a convenient way to reset the board.

Wire up the hardware as in **Figure 2** and use the GPIO table for guidance. Once wired your project should look like **Figure 3**.

Software

The application will collect 480 16-bit samples from the PDM microphone at a sample rate of 16 kHz using the microphone-library-for-pico (**magpi.cc/mic4pico**). This library combines the Programmable I/O (PIO) and Direct Memory Access (DMA) features of the RP2350 with the OpenPDM2PCM (**magpi.cc/xccapdm**) library to convert the raw PDM data into Pulse-Code Modulation (PCM) format. The 16-bit PCM data is converted to a 32-bit floating point and denoised using the RNNoise algorithm. After this the denoised frames are converted to 16-bit integers and sent over USB with the TinyUSB library (**magpi.cc/tinyusb**). The USB transfer will send 16 denoised samples every millisecond.

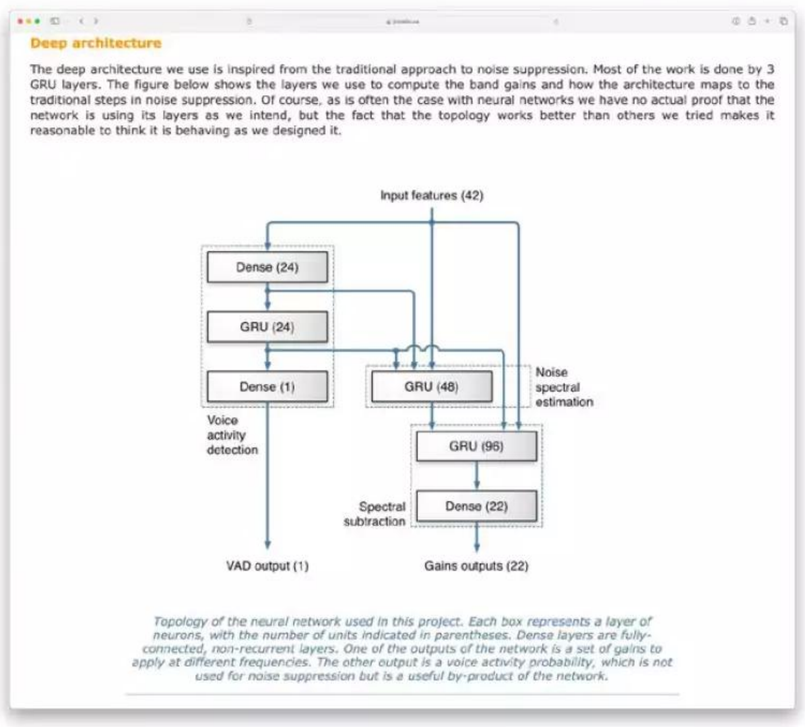


▲ Block diagram of the USB microphone with denoising

Both of the RP2350's Cortex-M33 cores are used in this application. Core 1 captures raw data from the PDM microphone, filtering, and denoising it. Core 0 handles transferring the de-noised data over USB using the TinyUSB library and the RP2350's USB interface.

The voice activity detection output of the RNNoise model will be displayed on the Pico 2's built-in LED using Pulse Width Modulation (PWM). When VAD output is close to 1.0 the LED will be brighter and when close to 0.0 the LED will be off.

▼ Screenshot of the RNNoise: Learning Noise Suppression page's Deep architecture section



Source code for the application can be found in the examples/usb_pdm_microphone folder of the rnnoise-examples-for-pico-2 GitHub repo (**magpi.cc/usbpdmmic**). The application can be compiled in a similar manner to the benchmarking application, using the following make command:

```
make rnnoise_usb_pdm_microphone
```

▶ Screenshot of the RNNoise: Learning Noise Suppression page

Once compiled, the **examples/usb_pdm_microphone/rnnoise_usb_pdm_microphone.uf2** file can be copied to the Pico 2's USB disk after holding down the BOOTSEL button and resetting this board.

Testing

Once the application has been loaded to the board, you can test recording audio using Audacity by first clicking Audio Setup button > Rescan Audio Devices, then Audio Setup button > Recording Device > MicNode, and clicking the record button.

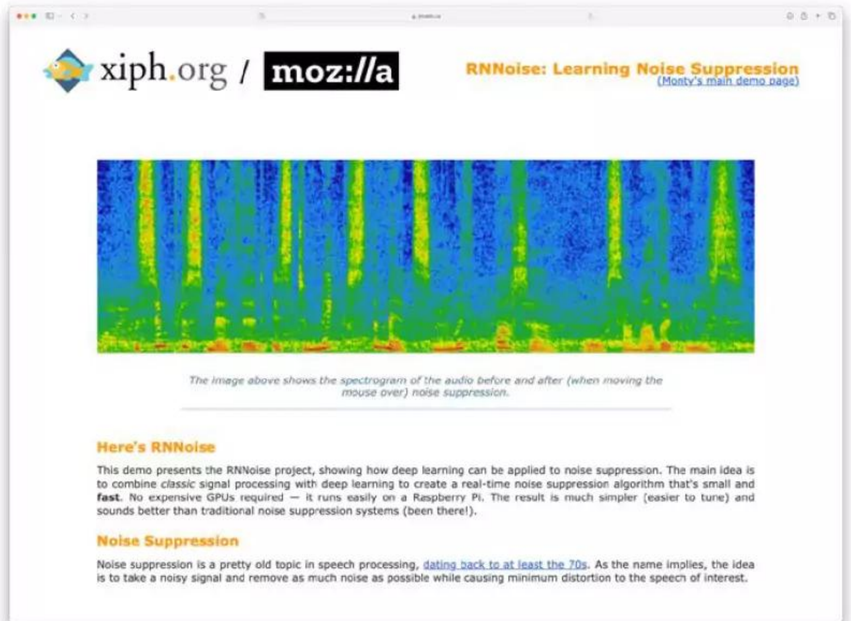
If you connected the optional slide switch, you could disable the noise suppression by sliding the switch towards the Pico 2's USB connector and re-enable noise suppression by sliding the switch away from the USB connector.

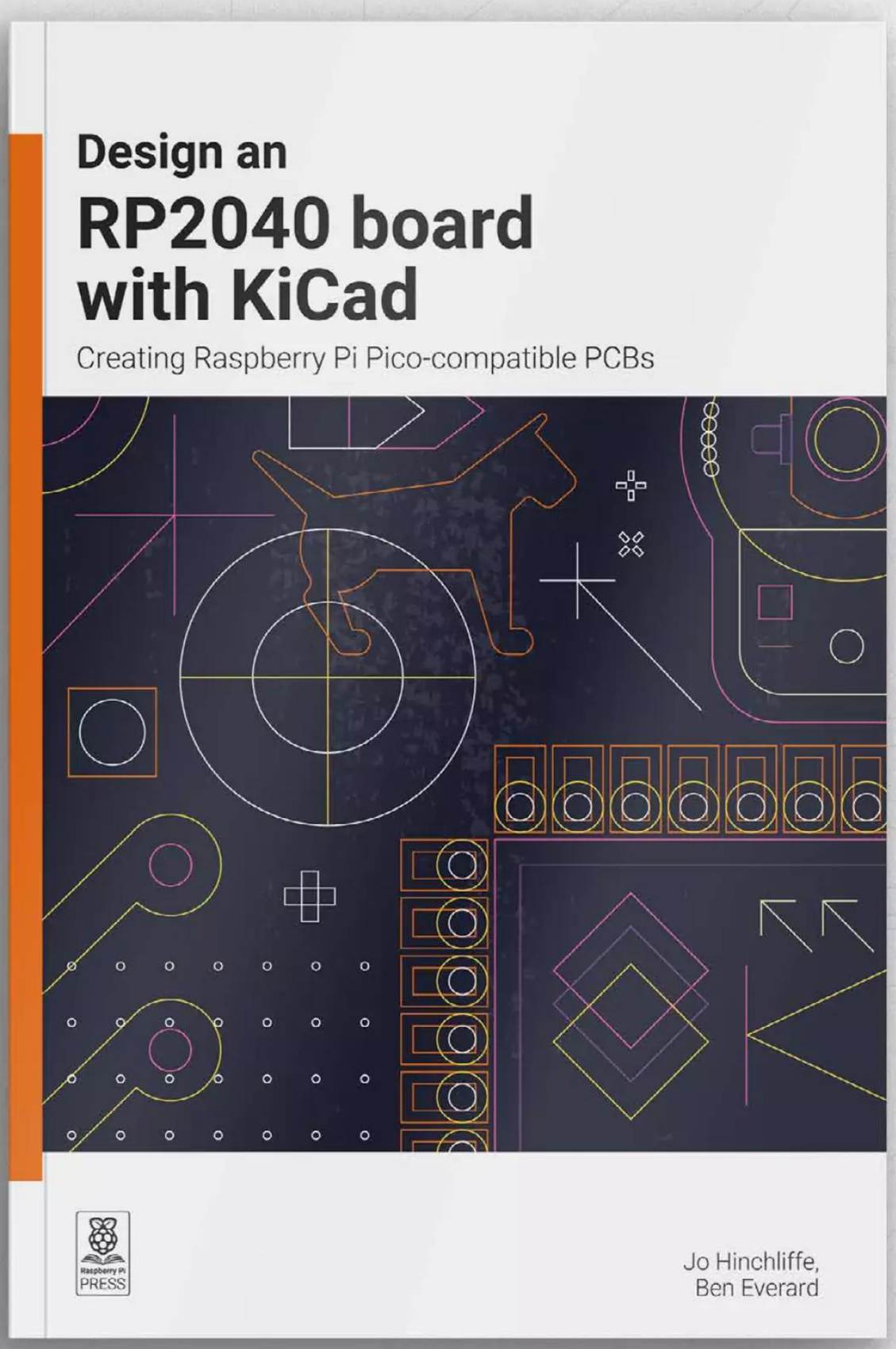
The recorded demo videos use the Pico 2 as a USB microphone, first with noise suppression off (**magpi.cc/mlnoisedisabled**) and then the same input with noise suppression enabled (**magpi.cc/mlnoiseenabled**). See and hear the results of the noise suppression algorithm!

That's a wrap

This article has demonstrated how the additional capabilities of the Pico 2's Arm Cortex-M33 CPU can be used to denoise real-time 16-bit audio data captured from a PDM microphone at 16 kHz with an ML model. The denoising algorithm makes use of the Cortex-M33's floating point unit (FPU) and runs 16.87× faster than on the Cortex-M0+ of the original Pico board. The application uses one CPU core to capture, filter, and denoise data, and the other to transfer the audio data to a PC over USB.

As a next step, you could modify the application to add Automatic Gain Control (AGC) before the denoised data is sent over USB to the PC. Alternatively, the denoised data can be used directly on the board, as input to another digital signalling processing (DSP) algorithm or a ML model that is run on Core 0 instead of the USB stack.





KiCad is an amazing piece of free and open source software that allows anyone, with some time and effort, to make high-quality PCB designs.

- ***Create a schematic for a microcontroller board using Raspberry Pi's RP2040***
- *Select the right components*
- ***Customise the hardware for your needs***
- *Lay out and route the PCB design*
- ***Prepare your board for manufacture and assembly***
- *Write software to get your design working*

Buy online: magpi.cc/kicad2040

HackSpace

TECHNOLOGY IN YOUR HANDS

HACK | MAKE | BUILD | CREATE

TOP PROJECTS

PG 66

3D PRINTED LOOM

A hand loom that can be used to create patterned cloth

PG 68

ARCADE BRIEFCASE



Portable arcade gaming delivered in style by an MDF case and a Raspberry Pi 5

PG 70

HDSP WRISTWATCH

In the olden days, this is what they thought the future would look like (and it looks pretty good)

PG 72

PILET



Modular, portable computing with awesome battery life and a full-fat Linux desktop

FORGE

PG 82



UNUSUAL TOOLS

Incredible accuracy from an age before computers

PG 84

TIN CANS

Don't throw those tin cans away: they're affordable metal cylinders perfect for one-off projects

PG 88



HEX TUBE ROCKET

Cylinders are boring – we're building a rocket shaped like an HB pencil

PG 92

MOSAICS

This ancient art form was good enough for the Romans, and it's good enough for us

OBJET 3D'ART

PG 74

SMART CLOCK

A glowing seven-segment display with many more features up its sleeve



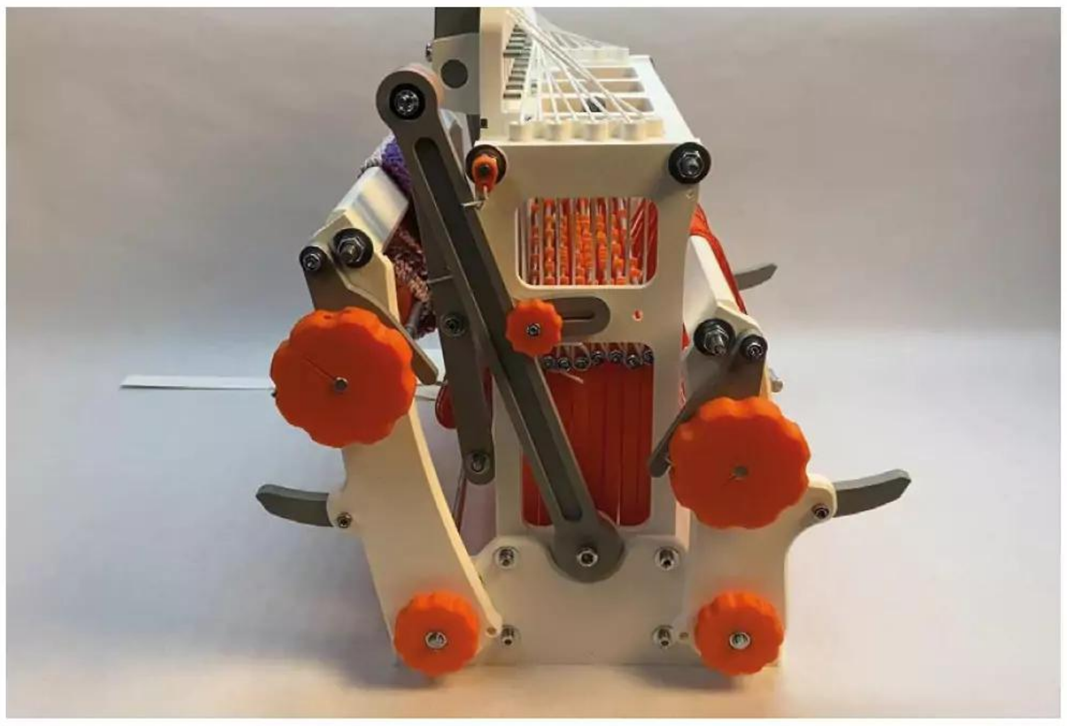
HOW I MADE

PG 76

MILITARY JOYSTICKS

Decommissioned simulator sticks take to the skies once again





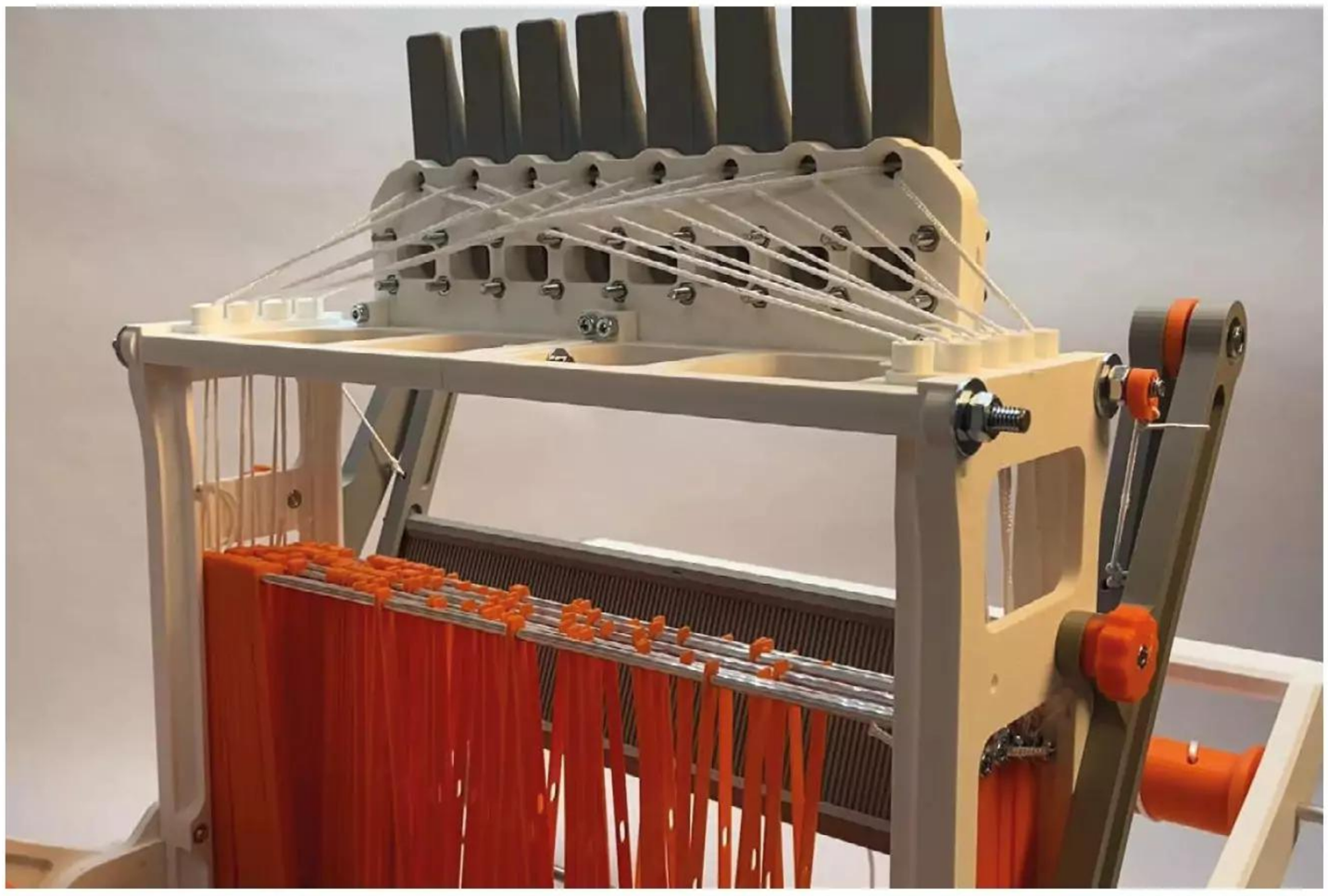
3D-printed loom

By **Fraens** fraensengineering.com

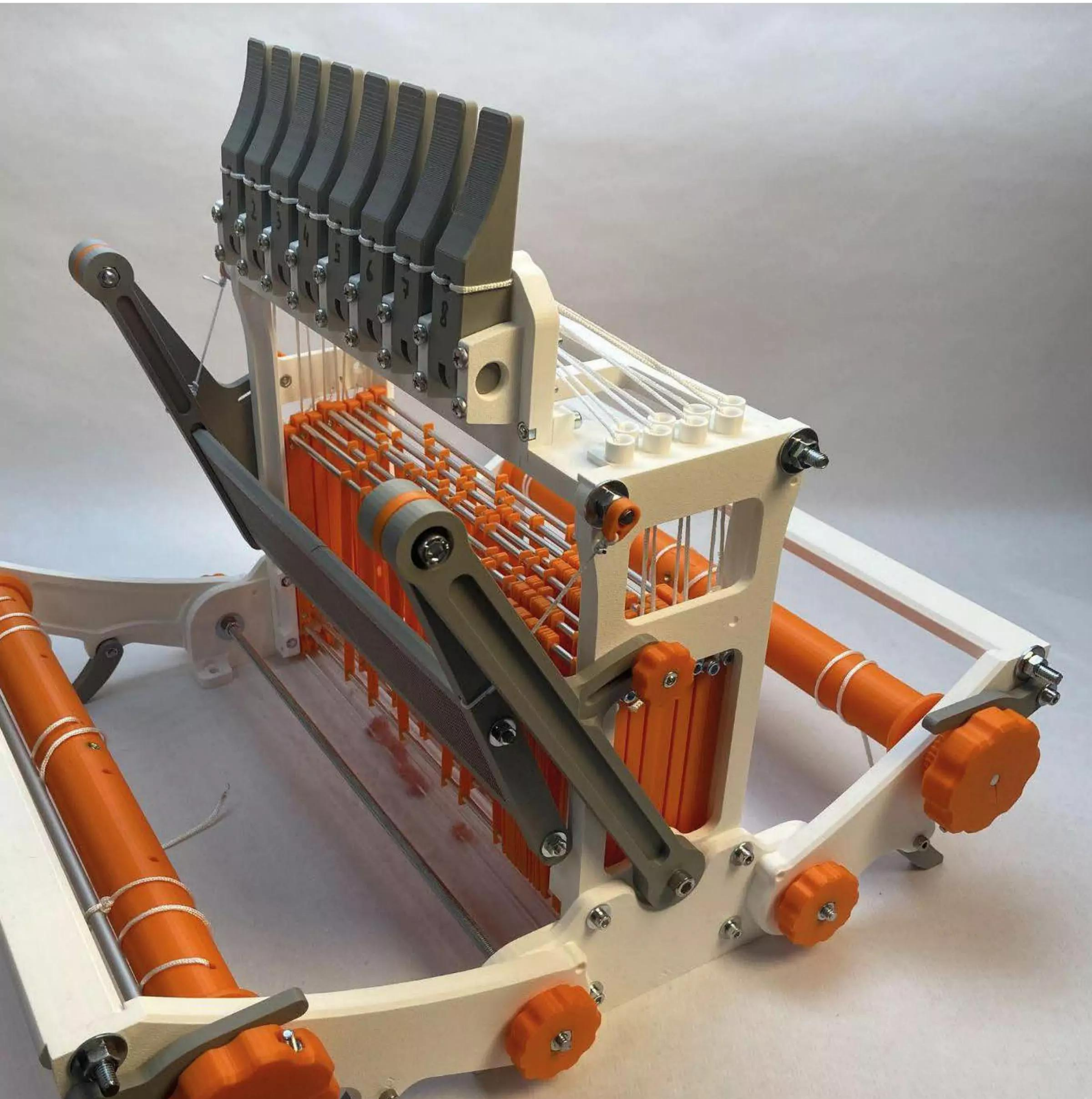
If you've ever tried to specialise in any field of making, you'll find that at some point you'll have stopped – or at least delayed – creating things, in order to make things that help you make things. If you're at the start of your journey into woodworking, for example, you'll very sensibly want to start with a bench hook, to hold workpieces steady while you're sawing them. Then, of course, you'll need a bench, otherwise the bench hook is useless. A few weeks pass, and at every turn, you're spending more time making jigs than you are making the thing you wanted to make – what even was that again?

Into this grand tradition of making in order to make, we have Fraens, who has also made something that makes other things. Unlike us and our never-ending woodworking rabbit hole, Fraens has used 3D printing to make a useful machine in one attempt: this hand loom can be used to create patterned cloth, exactly like humans have been doing for thousands of years.

Also unlike us, Fraens is something of an engineering genius. It's easy to give the credit to the 3D printer, but in reality we all have the same access to the tools; it takes a special ability to imagine new ways of doing old things.



Right
Humans have woven cloth ever since there were humans. Not many have done it with a 3D-printed loom, however



Arcade briefcase

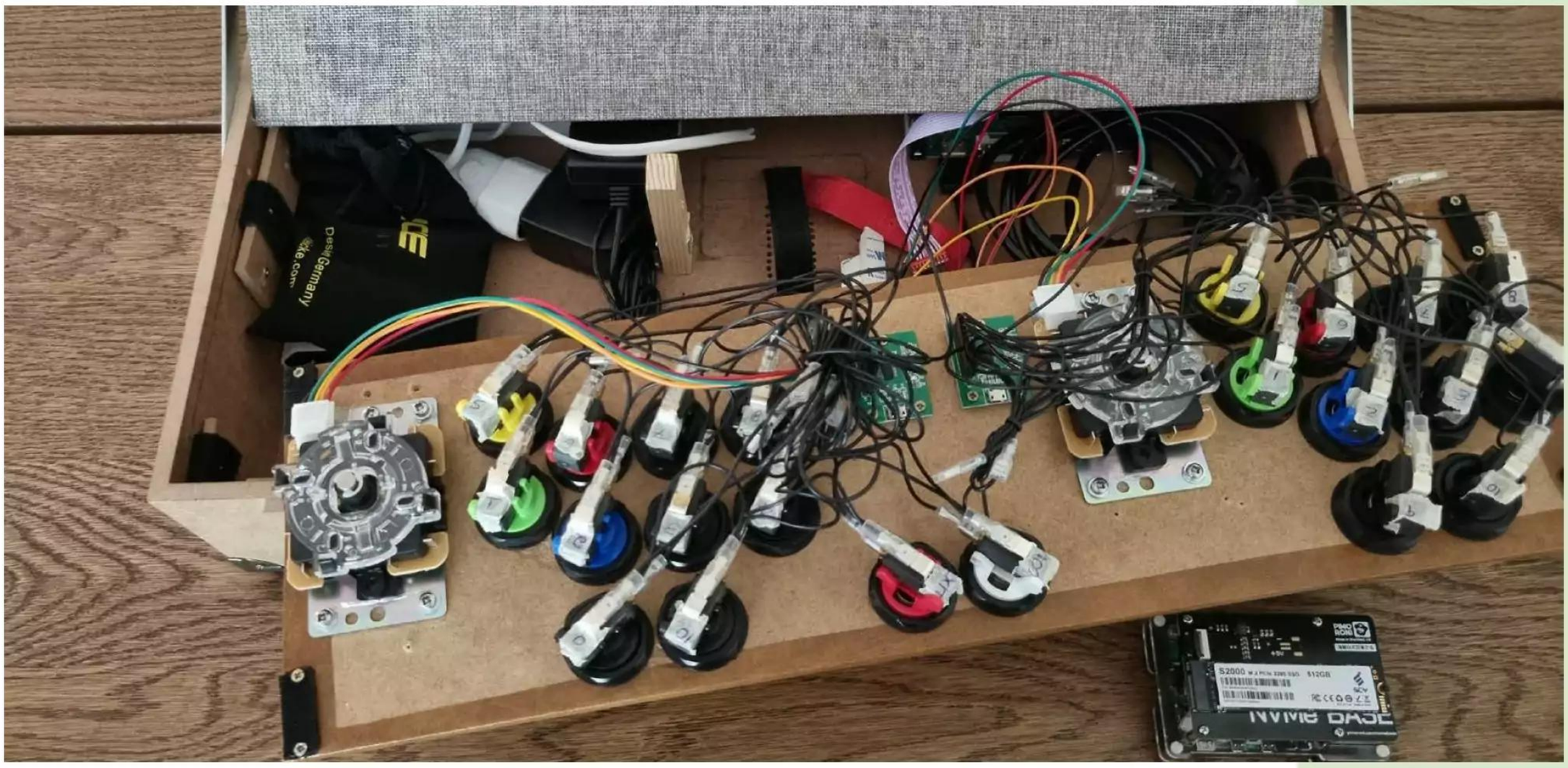
By **SrGamer** hsmag.cc/BriefcaseArcade

Readers who are fans of retro gaming will love the Picade, the tiny desktop arcade machine, powered by Raspberry Pi, which was released back when Pimoroni was in short trousers. With its powder-coated MDF frame, clicky buttons, and proper joystick, the Picade is the executive toy of choice for anyone who still has the Dragon Punch joystick movement from *Street Fighter 2* burned into their muscle memory.

Alternatively, if you have access to a soldering iron and a drill, you can build your own home arcade setup. This build by SrGamer is based on a Raspberry Pi 5, and features two joysticks, loads of buttons and a gloriously chunky red power switch built into the case. The case was bought from Etsy rather than handmade, as the maker didn't have much space to work, no woodworking experience and no access to workbenches, table saws, and other tools. We're all standing on the shoulders of giants. All SrGamer had to do was see the potential, then bring it to life. We're slightly worried by the choice of an MDF case, as it's not the strongest material, and if you spill beer on it then it'll swell up and deform, so if you use it in any of your project, be sure to finish with a coat of paint.



Right
Into retro games? Turn to page 32 for a deep dive into '90s horror





Warning!
Electrical Safety

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

magpi.cc/electricalsafety

HDSP wristwatch

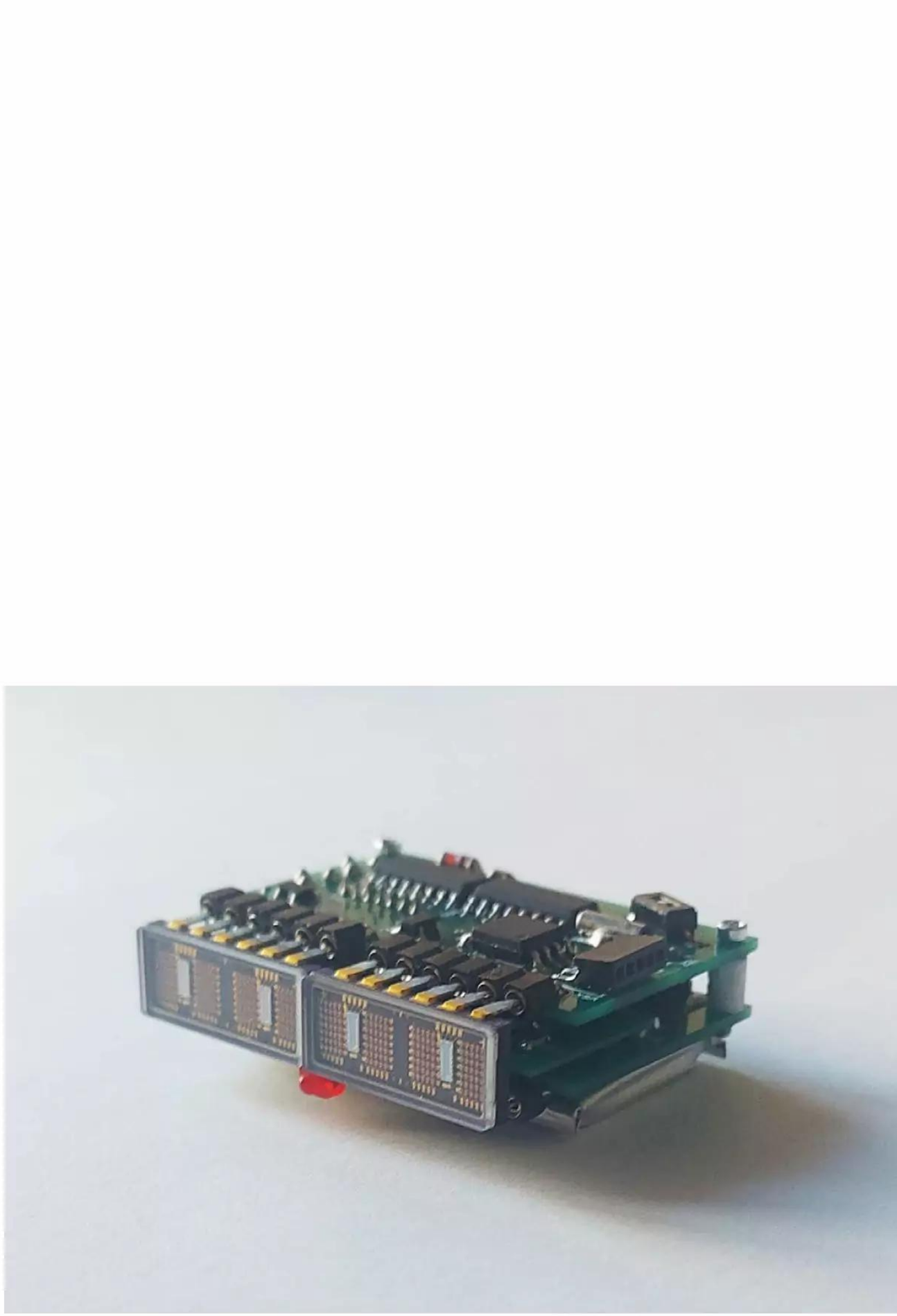
By **Vitalii**

hsmag.cc/HDSP-2000Watch

According to Ukrainian electronics designer Vitalii, the most basic microcontroller for a clock/watch would be an ATmega328 (as seen in the Arduino Uno). However, he wanted something more challenging for his latest build, and instead plumped for an ATtiny85. This unusual choice of microcontroller presented several challenges, not least of which is the fact that it only has five usable I/O pins.

With a six-digit, seven-segment display such as the HDSP-2000 (itself an unusual choice – he hasn't made this easy), Vitalii needed to find a way to multiplex the signals coming out of the chip, multiplying the I/O signals with transistors until he had enough to control each of the segments in the display. The result is this wonderful wristwatch, the custom PCB that enables the ATtiny85 to control the display, and a great deep dive into multiplexing written up on **Hackaday.io**.

We're seriously impressed by this feat of electronic engineering. If you are too and you want to try it yourself, we'd suggest that you start with a single seven-segment display, a breadboard, and go from there – this tiny form factor presents loads of difficulties, all of which have been overcome here with aplomb



Right
Putting the technical mastery here to one side, this is a seriously good-looking project





Pilet

By **Soulcircuit** soulscircuit.com/pilet

Do you want a portable mini modular computer based on Raspberry Pi 5? If so, you're in luck. A small outfit (boasting one and a half people) called Soulcircuit is working on one right now, called the Pilet (it was called Consolo, but is now called Pilet, which according to the maker "reflects the project's aim to appeal to a wider global audience").

Two 8000mAh batteries give the device a claimed seven-hour lifespan, which if true will put a lot of computing power in your pocket for a productive day's work. The basic unit houses a Raspberry Pi 5 and a touchscreen, running a full-fat version of the Linux operating system (it looks like Debian with a KDE desktop, which wouldn't really have been practical with any model of Raspberry Pi until now).

Soulcircuit claims that the Pilet is "built by open source software for the open source community," and credits Kicad, FreeCAD, Blender, Linux, Raspberry Pi and KDE. As we've seen so many times though, it's not enough just to have the right software; a device this good takes expertise and imagination, and if it can come in at the expected price of under $200, we'll sure it'll be popular with open-source geeks who want to get work done but also quite like leaving the house every now and then.







Right
The keyboard module includes a touchpad and a scroll wheel for navigation



Objet 3d'art

3D-printed artwork to bring more beauty into your life

3D printing has really expanded the possibilities for making unusual shapes. If you can design it in software, then you can print it. We think about shape all the time when we're designing for 3D print; but we rarely think about the benefits of combining different filaments into one design.

This clock can tell the time, the temperature and the weather, and it has a speaker and an amplifier module so it can read out this information to you in robotic English.

This feature-packed example of physical computing by Mert Kaya hides LEDs behind white PLA to create a gorgeous diffused effect (unusually, these are plain, bog-standard LEDS rather than the addressable RGB lights that we've become used to). There's also a light-dependent resistor to adjust the brightness of the LEDs, so that when the room is dark, the display is dimmed (you don't need the lights on at full blast in the middle of the night). ◻

hsmag.cc/Opaque7Segment





Right
The whole thing is built into a 3D printed enclosure, complete with chunky buttons

HOW I MADE

A PICO KIT TO BRING JUNKED JOYSTICKS BACK TO LIFE

By **DAVID MILES**

One of the many great things about the EMF Camp events (www.emfcamp.org) is the swap shop tent where all kinds of things are brought to be sold and exchanged.

On one of my many visits there I found a slightly worn (but very heavy) pair of joysticks which looked as if they had been part of a professional simulator at some point. In this article, I'll talk about how I reverse-engineered them to create a fully fledged flight simulator controller. Along the way I happened to create a Pico program that makes it easy to use any input device as a USB joystick.

GETTING STARTED

Figure 1 shows the joysticks as I found them. Ultra Electronics is a manufacturer of devices for the Ministry of Defence in the UK, so this looked like something interesting. My hope was to try and get them working so I could use them with a flight simulator with a plane that used joysticks like these. This meant I had two challenges:

1. Getting data out of the joystick
2. Making something which connects to a PC as a game controller



Figure 1
The joysticks as I found them in the EMF Camp swap shop

"CREATE A FULLY FLEDGED FLIGHT SIM CONTROLLER"

FOLLOW THE SIGNALS

To discover how to get data out of the joystick, I had a look at the wires that came out of it. The main unit has two plugs on the end of a (surprisingly long) wire – what looks like a nine-pin RS232 serial connector, and a 15-pin game port connector. The secondary joystick (the one on the right in **Figure 1**) has a 25-pin connector which plugs into the primary one, which suggests that it just contains switches, and that the first is the brains of the operation. The connector types fit the late-'90s feel of the hardware and give clues as to how we can talk to it, but we can take a look inside to confirm some assumptions.

Figure 2 shows the view inside the device. There are four screws holding the top part of the joystick inside the case. After removing these, the whole stick and gimbal assembly lifts out and we can see this circuit board. Some of the components on here give us some pointers on how this works.

There's a chip with a label covering it. This is normally a sign of some sort of microcontroller, indicating that there's more going on than just a simple controller. This is backed up by seeing some chips with 'ADC' on them – this stands for 'Analogue to Digital Converter', and could be used to turn an analogue value (e.g. the position of a potentiometer in a joystick) into a digital value sent to a computer. →

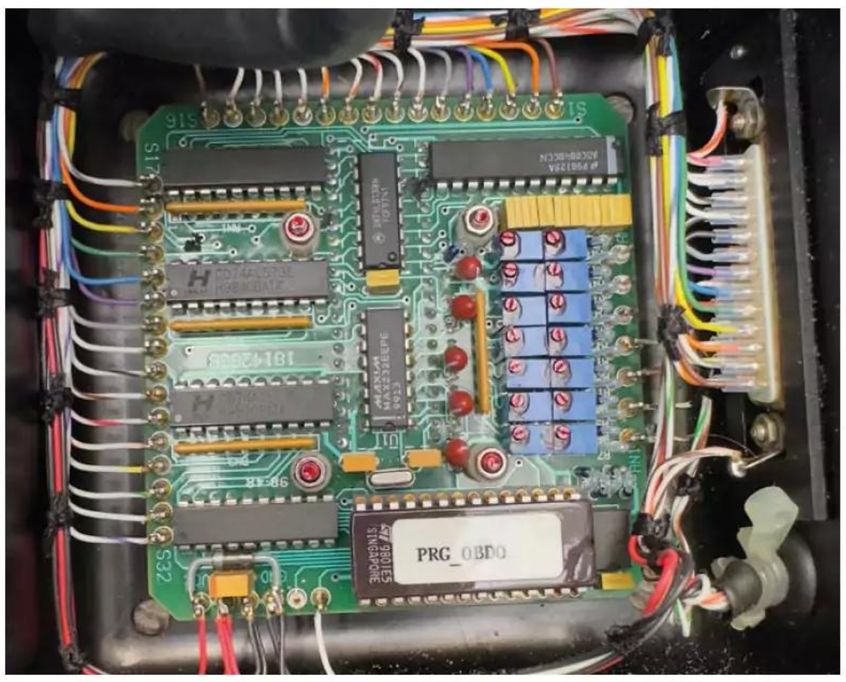


Figure 2
The PCB at the heart of the joystick. Look at all those carefully adjusted and sealed calibration resistors!

GOING FOR GAME

I wanted this to work as a game controller, as that allows the joystick to send analogue values from axes, digital buttons, and hats which can be read by programs running on PC.

Most flight simulators let you bind actions in the program to whatever your controller supports, so you can customise how it controls the simulation. In this case, we'll want the two main analogue axes to control the ailerons (on the wings) and elevators (on the tail), and use the buttons and hats to control actions such as trim or flaps.

Figure 3
All the wires have unique colour combinations



PAYING A COMPLEMENT

Two's complement is the most common way of representing negative numbers using binary values. An 8-bit binary number can represent 256 possible values, which we normally treat as 0-255.

Two's complement instead lets you treat an 8-bit value as -128 to 127, by using the left-most (or most significant) bit to indicate if the value is negative, and then for negative numbers, subtracting the value from -1, which is equivalent to flipping all the bits.

You can then map these back to regular Python integers by code which looks like:

```
def fix_analog(val):
    if val > 128:
        return - (256 - val)
    else:
        return val
```

The value -128 corresponds to the joystick value being all the way in one direction, and +127 all the way in the other.

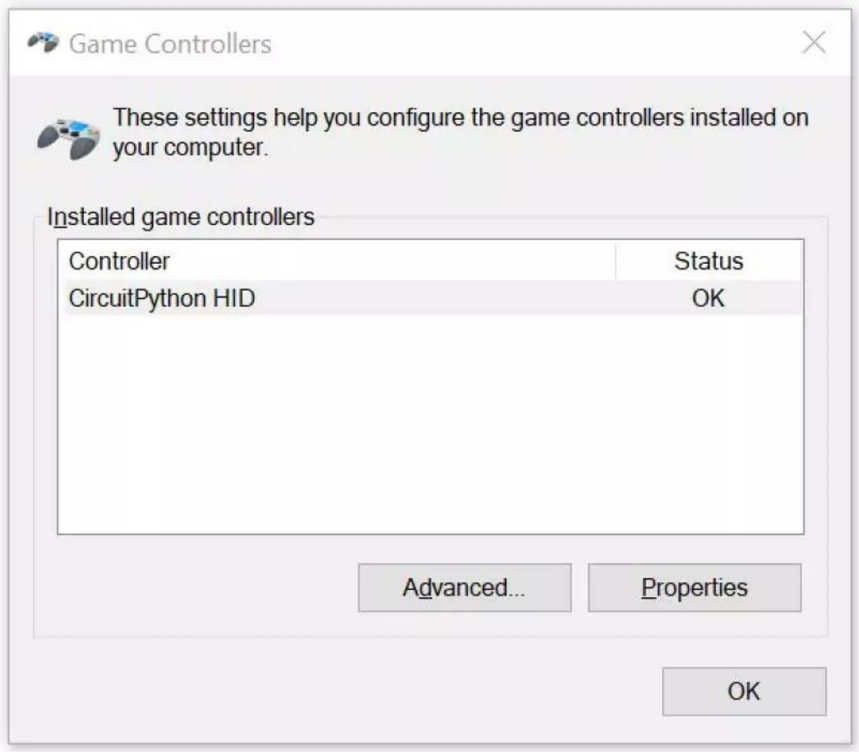
Finally, in the middle, there is a chip with 'MAX232' written on it. This family of chips are used to convert 5v logic levels you often see used in microelectronics to the 15v levels used in RS232 communications. This is another sign that the nine-pin connector is probably a regular RS232 connector I can use to get data from the joystick.

This leaves the 15-pin connector. Is it a game port? Game port connectors were often used in the '90s to connect a joystick to your PC (confusingly, often using a socket on your sound card). They allowed you to connect two joysticks to your computer, each with two buttons. We can see that this joystick has more than four buttons – doing a quick count we seem to have about 30, so maybe it's being used for something else.

If we pull the back off the connector, we can see that there are only two pins connected using red and black wires. If we check those pins on the game port specification, we see that these are connected to ground and 5v.

MADE

“I WANTED ALL THE ABILITIES OF THE JOYSTICK”



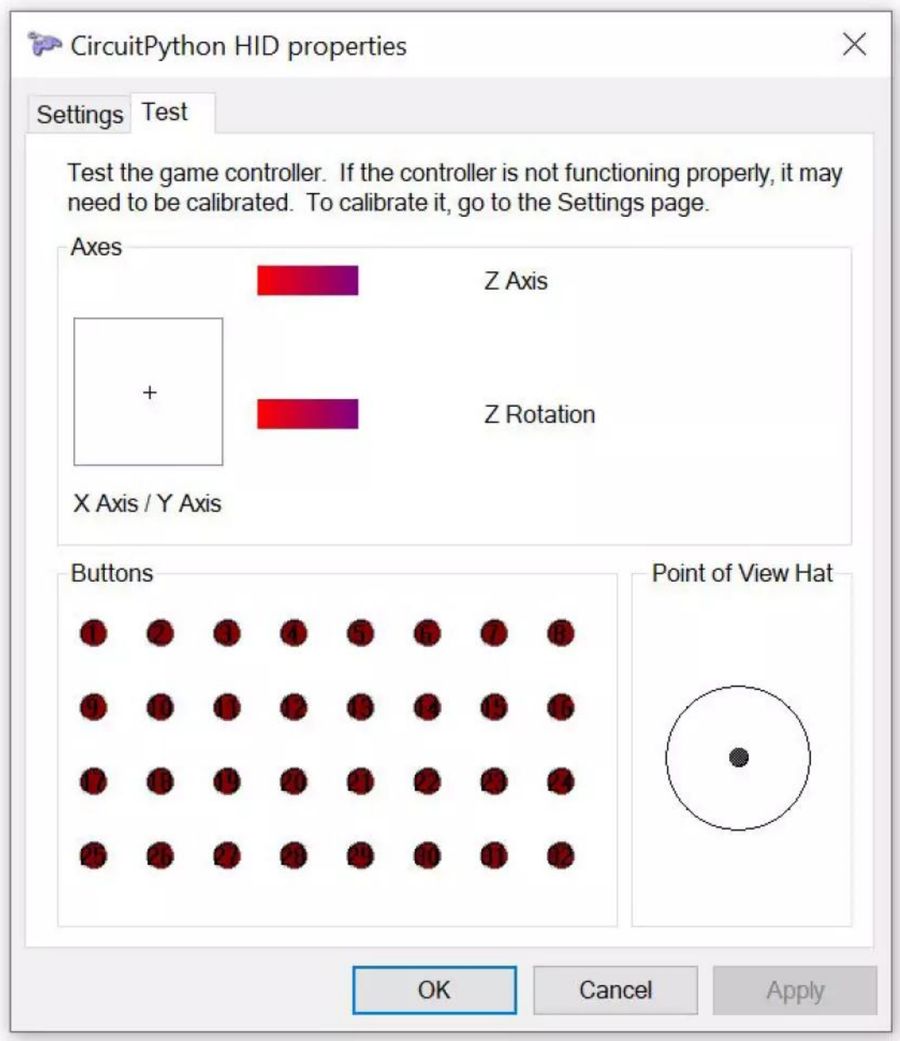


Figure 4
This is how Windows sees the controller. The Raspberry Pi has a similar interface for joysticks

This fits in with our earlier findings. There’s no way to power a device over a regular serial connection, but using a game port alongside a serial connector would allow you to connect to a PC and transfer power and data.

PLUGGING IT IN

Now that we think we know how this works, we can have a go at connecting it to a computer. To do this, I used a regular USB-to-serial converter and a power adapter connected to a USB power bank.

After connecting power, I could see a steady 100ma power draw, which felt reassuring, so I hooked up the serial connection and tried a few different configurations to see if I could receive anything sensible. Eventually, I found one which gave me a regular stream of data which changed as I moved the controls and pressed buttons. It turns out that these joysticks work at 9,600 baud (around 100 characters per second) and send 8-bit data.

Now I had data, the next thing was to decode it. I started by writing a Python program to read from the joystick and write it out. By starting to build something on the PC, I would have something I could then use on a Pico with CircuitPython.

Working through the controls on the joystick, I found that each button corresponds to a single bit in the data, and each of the axes (X and Y movements of the sticks) with a byte in the output, ranging from -128 to 127, and encoded using two’s complement.

MAKING A GAME CONTROLLER

Now that I could read from the joystick, I needed to send these to the computer as a proper game controller that could be used by programs such as *Microsoft Flight Simulator*. **Figure 3** shows what I was aiming for: I wanted all the abilities of the joystick to be exposed for use on the PC. →

QUICK TIP

The joystick can be used on any computer that knows about USB devices, including the Raspberry Pi.

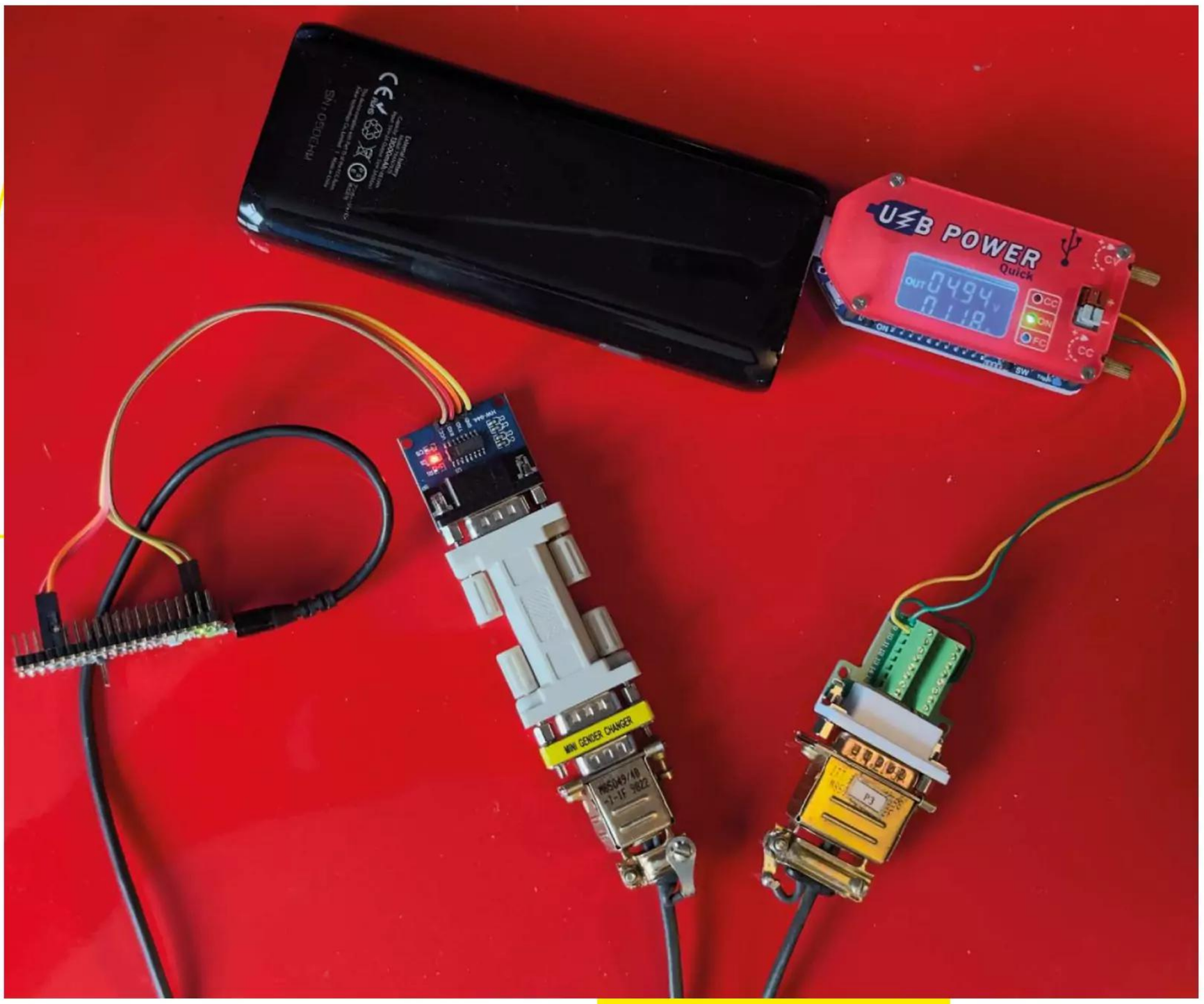


Figure 5
The system laid out on the desk. At the top is the power supply provided by a USB battery, at the left is a Raspberry Pi Pico connected to the PC using USB, in the middle is the RS232 interface, and on the right is the game port adapter

Figure 6
The EMF joystick with its virtual counterpart. This looks like a close match to the real thing

A Raspberry Pi Pico was the perfect thing to use here as it has great IO options and support for working in Python. I needed to configure it to announce itself as a game controller over USB, and then adapt my code to run on the Pico and send the appropriate messages to the PC.

Fortunately, CircuitPython has great support for making custom devices like this. First, we need to define how we want our device to appear over USB, and write some Python to make sure this is registered whenever the Pico connects over USB.

This involves digging into the world of USB HID descriptors. HID stands for Human Interface Device, and covers a wide range of gadgets which you can connect over USB, including mice, keyboards, and game controllers, but going as far as volume controls and exercise equipment.

These descriptors contain information about the device, such as what class it is, and what types of data it will send to the PC. By default, CircuitPython supports only a mouse and keyboard. We need to send a descriptor which contains the different axes, buttons and hats the controller supports.

The full specification for this is on the official **www.usb.org** site, but we can extend the Adafruit example to increase the number of buttons to 32, and to add an additional entry for the joystick's hats.

PUTTING THE BOOT IN

When building CircuitPython projects you will normally write Python in the **code.py** file. Code in this file runs after the Pico has initialised USB, so it's too late to change anything about the device. To do this, we need to look at the lesser-known **boot.py** file. This file runs when CircuitPython first boots, before USB is configured, which means that you can make changes to the USB devices that are registered.

The program in the **code.py** file reads data packets from the joystick, decodes values from the data, and sends USB messages corresponding to the state of the joystick. The present version is in two parts. The first reads the data into a buffer, and the second pulls values out of this buffer and sends USB messages.



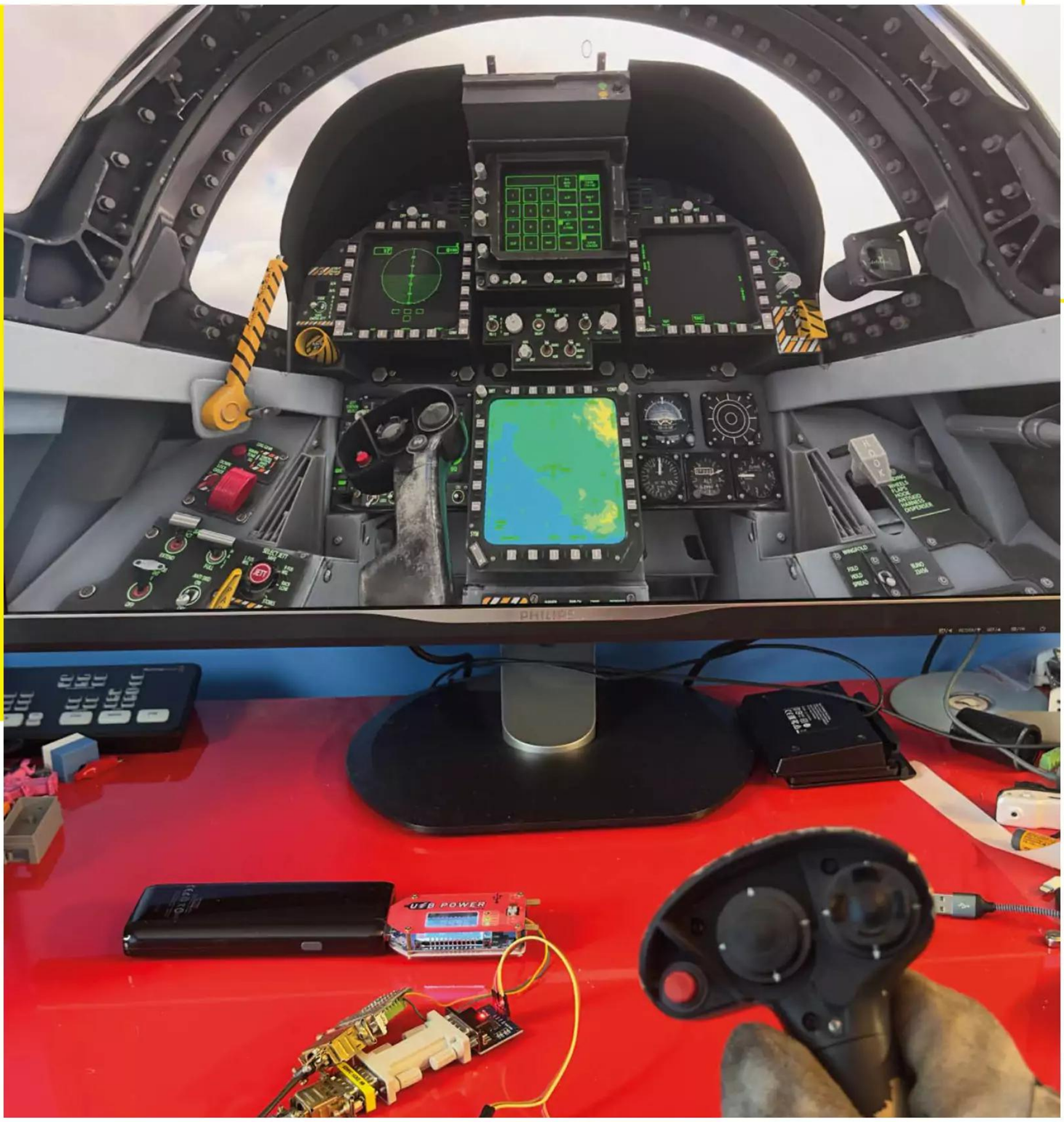


Figure 7 The joystick in action

BRINGING IT ALL TOGETHER

Finally, we need to integrate our project which reads from the PC serial port with our game controller code. As the Pico only supports 5v logic levels, we can use our own MAX232 to map the RS232 voltages to something the Pico can handle. To do this I used an integrated module with a DB-9 connector on one side, and a pin header on the other connected to the Pico. I could also power the joystick from the Pico's 5v supply.

Now that I had the two halves of the project working together, it was a case of going through the buttons on the device and mapping them across to the controls defined for the controller. This was one of the more tedious parts of the project, but it was a good chance to make sure everything was working correctly.

Finally, I've got everything I need to use the joystick to control a plane running on the PC. Here I'm using it to control the McDonnell Douglas F/A-18 included with *Flight Simulator*, which has a similar stick to this, but with two hats swapped. I still don't know precisely which plane (if any) this joystick is based on. The AV-8 Harrier has those hats in the correct order, but has an extra button just below them on the face of the joystick.

I've not been able to find any images of joysticks similar to the second stick anywhere, so if you have any idea where that could have come from, please let me know!

QUICK TIP

The source for this project is at **github.com/Davermouse/emfstick** if you want to see how it works or find one of these yourself.

Unusual tools: Gauging Blocks

Compare sizes like a professional





Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

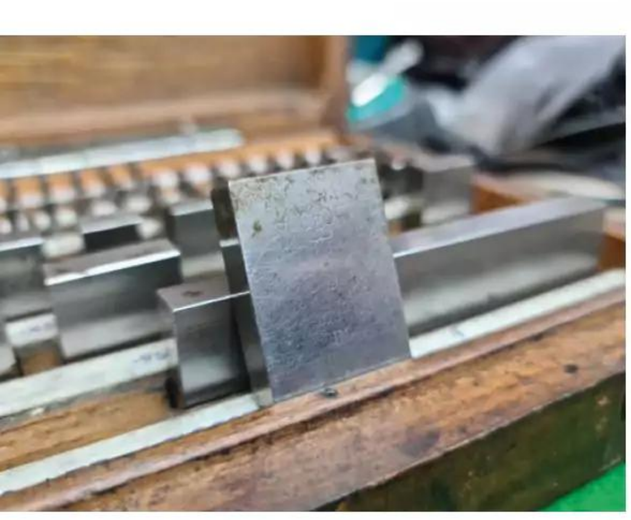
Gauging blocks are hidden wonders of the machinist's workshop. While they're a common sight in metal workshops, they're almost unknown in the larger community.

Gauging blocks are the dimensional equivalent of the set of weights from a balance scale. They are a chunk of metal with an exact size. They are not an adjustable measuring tool like a micrometer or calliper gauge, they are just a very accurate representation of a particular size. They are the objects that measuring tools are calibrated against, and they are incredibly useful to have in your toolkit. Gauging blocks are normally made from hardened steel or ceramic, although other hard materials can be used to reduce wear and maintain their accuracy for as long as possible. The sides of a gauging block are extremely flat, precision ground to maintain a near-perfect surface. The level of precision for a particular set of blocks will be rated differently depending where you are in the world but, in simple terms, very precise gauging blocks are extremely expensive. For a general home user, a basic set that costs less than £100 will be more than adequate.

Left
Sometimes called slip gauges, gauging blocks are available in both metric and imperial sizes. They can be purchased individually or as part of a set, like the one shown here

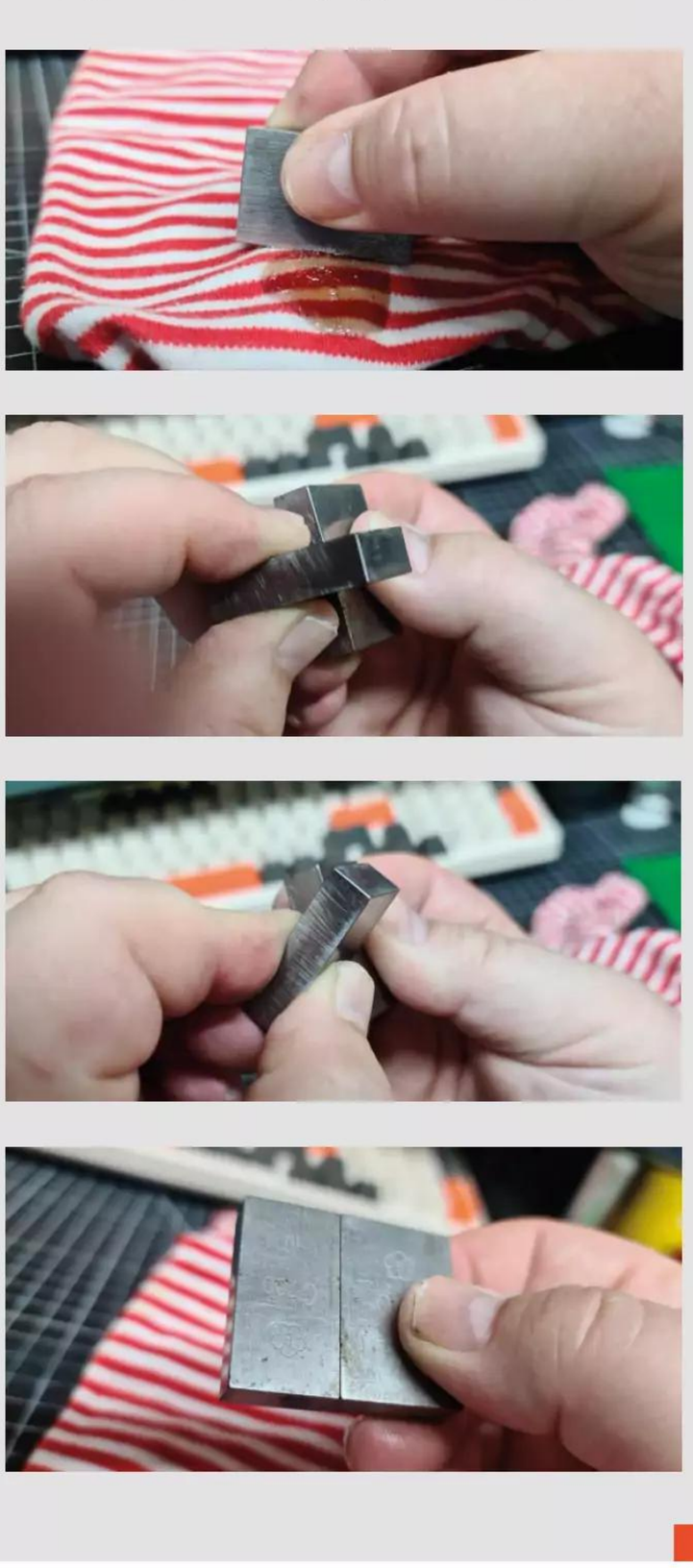
Below
Gauge blocks are precision tools and can be completely ruined if they are allowed to rust. For professional engineers, this is a problem. For the tolerances most people need on a home workbench, it probably doesn't make much difference unless the blocks are very rusty. A quick clean with some spirits and a cleaning stone should bring them back into usable condition

When you think about calibration with gauging blocks, you probably imagine milling machines and lathes, or extremely precise callipers and micrometers. You might not think of using them with your laser cutter. If you have a K40 laser cutter, then you probably have a lens with a 50.8mm focal length. Ideally (for cutting most materials), that 50.8mm distance should be focused at the centre point of the material you're cutting. Different K40 laser heads hold the lens at different distances from the end of the lens holder, but if you measure the distance from the end of the lens holder to the lens itself, you can put a gauging block on your material while you adjust the height of the laser bed. That way, you know when the lens is in exactly the right place for the optimum cut. As an example, if your laser lens is 33mm from the end of the (for want of a better word) nozzle and has a 50.8mm focal distance, then you can do 50.8mm-33mm and figure on a 17.8mm gauge block to get you to the surface of your material. However, since you probably want your focal point to be in the middle of the material, you need to take the extra step and subtract half of the thickness of your material from 17.8mm, so that the material is raised slightly closer to the end. For a 3mm thick material, that would take you to 16.3mm, or 41/64 of an inch (or 0.6417") in imperial measurements. You could either join some blocks together to get the exact measure for setting the laser bed, or accept that it doesn't have to be that accurate and use the nearest block you have to that number. A 0.650-inch imperial block is roughly 16.5mm, which is close enough for laser calibration most of the time.



WRING THE CHANGES

The surfaces on a gauging block are so smooth that it's possible to join blocks together using nothing more than a thin coating of oil. Joining blocks this way allows you to create stacks of blocks that have a non-standard size. Although it's counter-intuitive, a thin layer of oil on the surfaces will join the blocks together very firmly provided you use the proper technique. The technique is called "wringing", and is easy to master.



Step 1
Clean your gauging blocks with degreaser and then add a small amount of oil to the joining surfaces using a pad

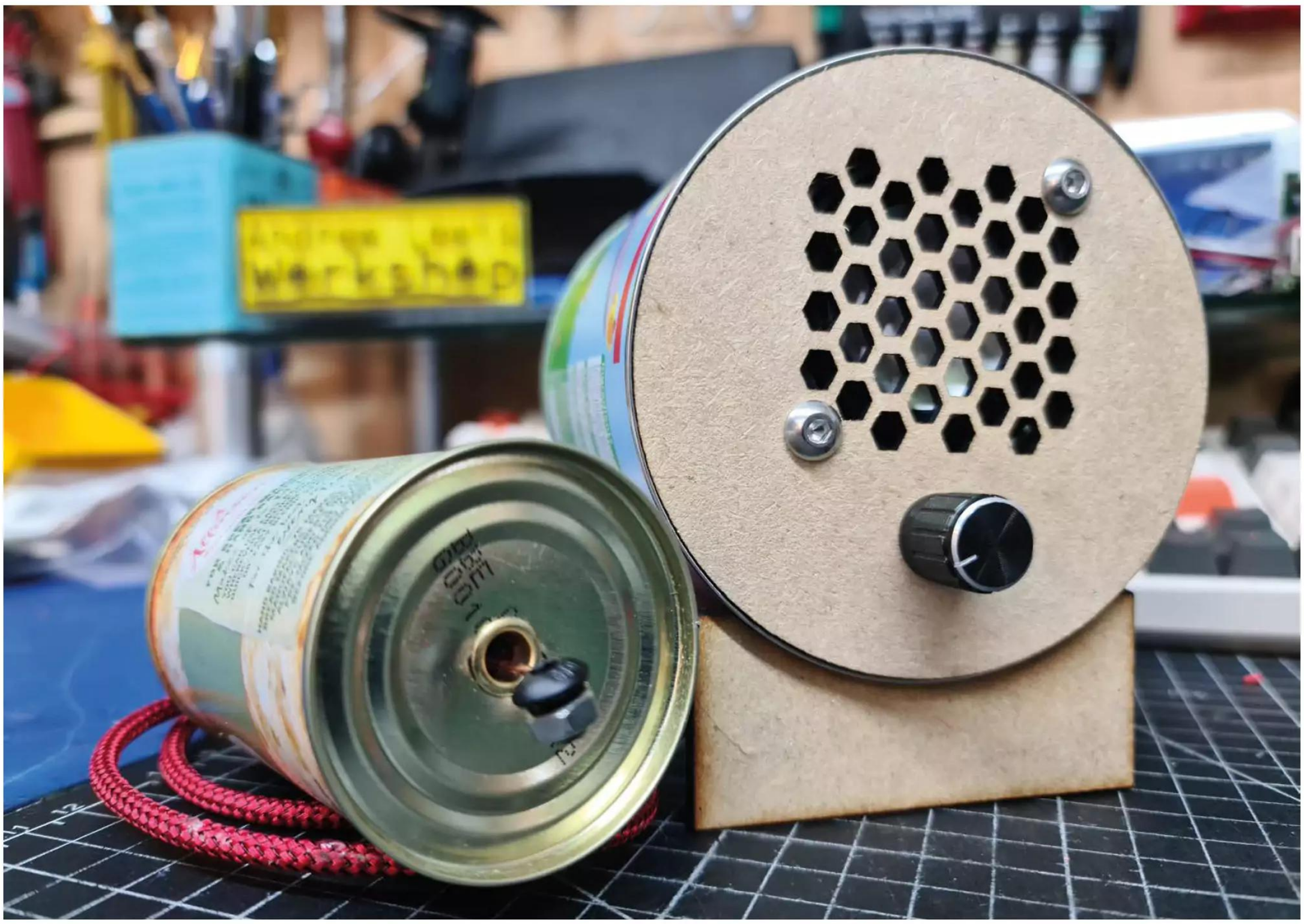
Step 2
With the pieces positioned at right angles to each other, apply gentle pressure and slide them together so that they form the shape of a cross

Step 3
Keep applying the pressure and rotate (wring) the pieces so that the two blocks are in line with each other

Step 4
If you do it properly, the combination of suction, attraction, and surface tension will make the blocks very difficult to pull apart

QUICK TIP

Gauging blocks are actually very slightly smaller than their marked size, so that they allow for a thin coat of oil on the surface.



Playing the can-can on a tin can

When you've finished your beans or peas, recycle the container into your kit



Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Years ago, electronics projects would turn up in the most unexpected places. Crystal radios would be housed in leather vanity cases, computers could be hiding in a discarded pizza box, and power supplies sometimes lived in ornate wooden cigar boxes. Somewhere along the way, the aesthetic of the 'professional' project box started to overtake the sentiment of reuse and recycle. With the coming of 3D printers, it's possible to make production-quality enclosures for any project. While it's nice to be professional, there's still a lot to be said for projects that reuse existing objects like tin cans. Tin cans are cheap, ubiquitous, more recyclable than plastic, and come in regular sizes. In this article you'll see a couple of projects that show the humble tin can shouldn't be discarded from our toolbox.

There's a common belief that you can't mount a wireless device inside a metal case because the case itself will attenuate the signal. This is true, but may not be as much of an issue as you imagine. For one

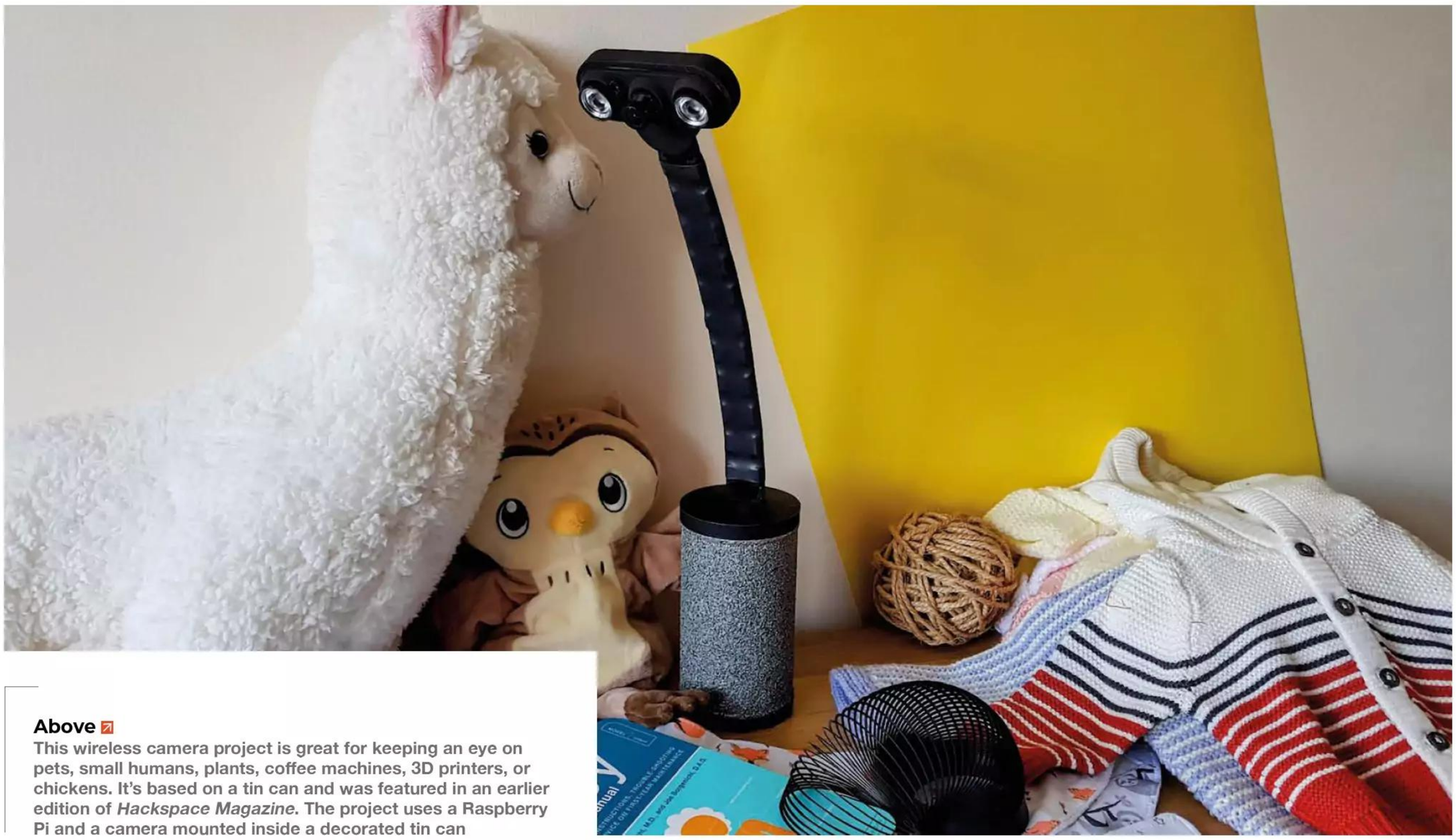
thing 'attenuate' isn't the same as 'completely block' and you might find that in a signal-rich environment you don't have any real problems. The level of attenuation is dependent on the type and thickness of the material that's blocking it, and conductive metals like steel and aluminium are especially good at blocking signals. But that's not the full story. Radio waves are (for want of a better word) sneaky. Completely blocking a signal is hard, unless you're deliberately trying to do it. We won't go into the hair-pulling intricacies of RF shielding and Faraday cages here, but suffice to say that even professionals can come to grief when trying to block a signal in the real world. Something like a tin can isn't usually thick enough or tightly sealed enough to completely block a signal. Let's make a couple of projects that prove it's okay to house your Bluetooth or Wi-Fi projects in reused tins.

One of the simplest home automation projects that you can make is an internet-enabled button. Whether it's a doorbell, a light switch or an emergency disco button, the design and implementation are practically the same. A microprocessor board with a Wi-Fi connection sends a signal to a service over the internet whenever a switch is pressed. It's a common project, and a Google search will probably show dozens of different write-ups on how to implement it. To add a little bit of interest, let's do something a little bit more unusual. Instead of a button press, let's consider a different sort of trigger – vibration. You can make a rudimentary vibration sensor using a tin can and some piano wire, and then house a Raspberry Pi Pico inside the sensor itself. →

MIND YOUR FINGERS

The big disadvantage of tin cans is that when you start cutting into them they have sharp edges. If you're working with a tin can remember to wear protective gloves and ideally make metal edges safe by double folding or adding a trim. Research tin-smithing and jewellery techniques to find more information on how to safely work with sheet metal.

Left
People sometimes avoid using metal cases for their projects because they don't want to block the wireless signal from their board. But a metal case doesn't automatically block every signal, and in reality it's actually quite difficult to completely block a radio signal even if you want to. These devices show that metal isn't always the barrier that people think it is



Above
This wireless camera project is great for keeping an eye on pets, small humans, plants, coffee machines, 3D printers, or chickens. It's based on a tin can and was featured in an earlier edition of *Hackspace Magazine*. The project uses a Raspberry Pi and a camera mounted inside a decorated tin can

REPEAT YOURSELF

If you need to generate a better signal inside a metal enclosure but can't use a directly connected antenna, you can try using a passive repeater. A passive repeater is about as low tech as you can get – it's an antenna connected to another antenna by a wire. It's an easy way to get a signal inside a dead-spot, but it isn't an ideal solution. For one thing, the signal output of the second antenna is going to be much weaker than the signal received by the first antenna. Additionally, if you're broadcasting the signal inside a small metal box, you're going to have problems with interference as the signal bounces around inside the box, reflecting off all of the metal surfaces. Not an ideal solution, but a possible fix in some cases.

Below
You can mount your microcontroller (in this case, a Raspberry Pi Pico W) inside the case so that the USB socket is facing the lid of the tin can. Some non-conductive stand-offs or even just some lollipop sticks and hot glue will be useful for this. With the Pico in place, cut out a hole in the lid of the tin can to allow the USB cable to pass through



Below
Position the wire inside the can so that it passes through the full length of the can and out of the hole at the end. Add weight to the end of the wire so that it wobbles more freely. You can do this by bending a loop in the end and adding a nut or split lead fishing weight

Begin with a small tin can. A yeast tin or round anchovy tin is a good size for this. Make a roughly 10mm hole in the end of the can, at the centre. This hole will be one side of your vibration switch. If you have a brass eyelet, use this to finish the edge of the hole. The brass eyelet will look neater, and will offer a better surface for electrical contact. Next, take a piece of non-conductive material (plastic or wood is fine) and cut it to the same width as the inner diameter of the tin can. To the centre of this piece, add a length of piano wire. If you don't have any piano wire, you can use thick solid copper wire, although it will be less resilient and will bend more easily. You can either glue this in place or loop the end of the wire and put a bolt through the non-conductive piece. The wire should be about 3cm longer than the outer length of the tin can.



This very simple vibration switch works because the inertia of the weighted wire will cause it to bend and strike the side of the hole in the can when it is moved. Connect the brass eyelet to the 3.3v of your microcontroller board by soldering a wire in place, then connect the piano wire to an input pin (GPIO14 is used in the example code in the tutorial link).

Now, you can follow the online instructions at **hsmag.cc/picowdb** to connect your Pico to IFTTT, or roll your own code to connect to an alternative service if you prefer.

Despite the fact that the Pico is largely enclosed by a metal can, it will be able send a request to IFTTT when the wire comes into contact with the brass eyelet. This works because the metal can is not thick enough to completely block the Wi-Fi signal, and the can is not completely enclosed at the ends.

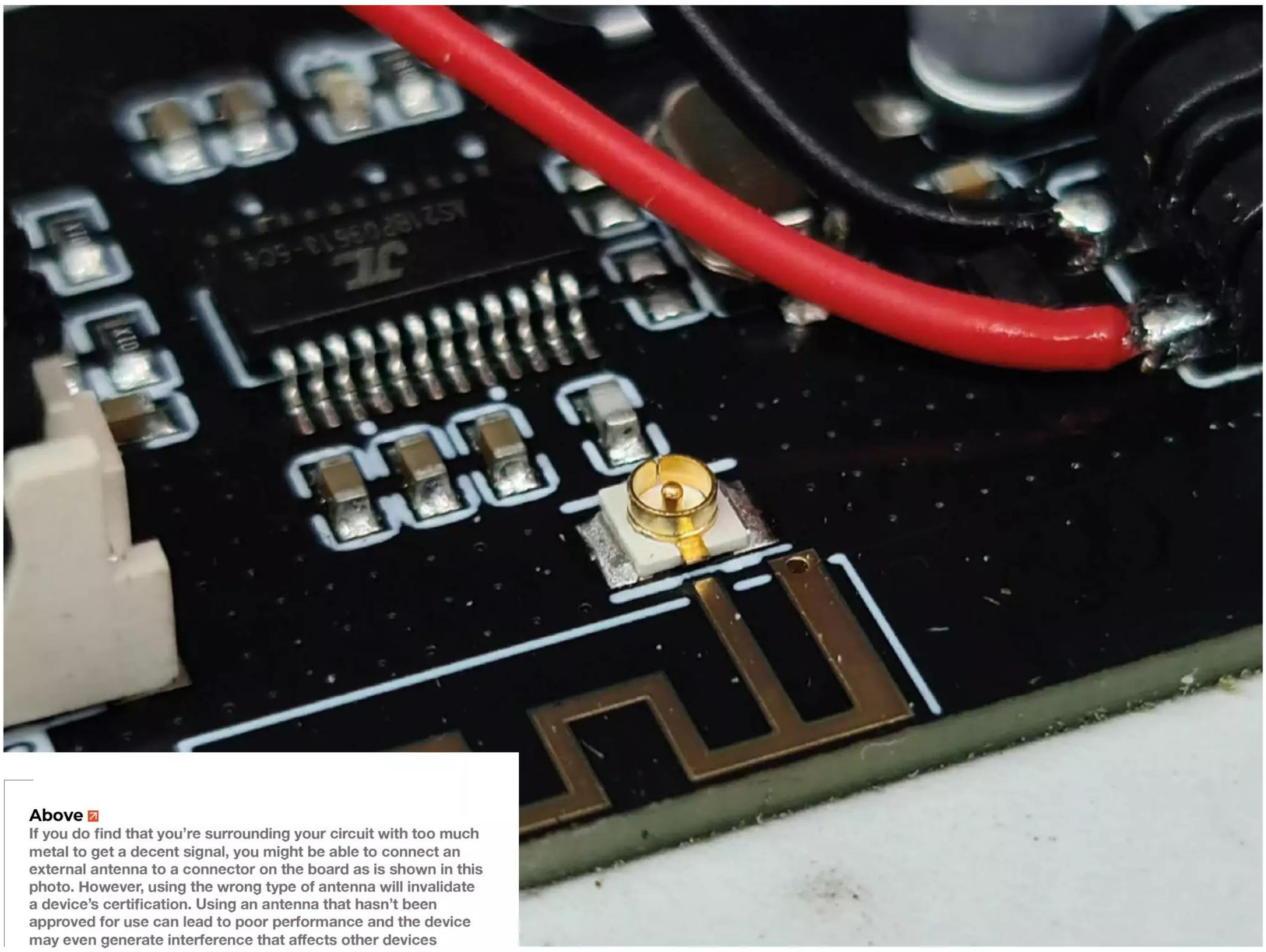
Another nice project that demonstrates the radio permeability of a tin can is a simple Bluetooth amplified speaker. Bluetooth audio modules and amplifiers are cheap and easy to get hold of, so stuffing them inside a tin can is a no-brainer if you have some old speakers laying around. This is a great project to try if you're not very comfortable writing code, because there's no need for a microcontroller. You can just plug the components into each other, solder the wires where necessary, and get a functional Bluetooth speaker.

QUICK TIP

De-bounce the contact by adding a time delay, to prevent multiple notifications being sent. In the linked example above, this is accomplished with time.sleep(2), to introduce a two-second delay after vibration is detected.

> **This works because the metal can is not thick enough to completely block the Wi-Fi signal**

At the minimum, you will need a large tin can, a Bluetooth audio module, a pair of three-watt speakers, and a three-watt amplifier module. To add some battery life to the project, you will also need an 18650 battery and a suitable USB charge circuit with a step up power circuit to increase the output from the 18650 battery to 5v. The idea behind the project is to connect the Bluetooth audio module to the amplifier, and the amplifier to the speakers. If you're making a portable unit with a battery, you'll also need to make sure that you include a power switch that isolates the battery. That's all you need to do – the project really is as simple as connecting power to the modules and linking the output of the Bluetooth audio module to the input of the amplifier.



Above
If you do find that you're surrounding your circuit with too much metal to get a decent signal, you might be able to connect an external antenna to a connector on the board as is shown in this photo. However, using the wrong type of antenna will invalidate a device's certification. Using an antenna that hasn't been approved for use can lead to poor performance and the device may even generate interference that affects other devices

HEXA: an open-source hexagonal rocket

Designing, building and launching a hexagonal model rocket



Jo Hinchliffe

Jo Hinchliffe (AKA Concretedog) has a house and shed full of lathes, milling machines, 3D printers and more. Jo is a constant tinkerer and is passionate about making. He's obsessed with rockets and robots and much more besides, and often releases designs and projects as open source.

In competition rocketry there are model rocket classes where people sometimes build super lightweight rockets by rolling thin card or paper into body tubes. Indeed I've been experimenting with this approach and even used my vinyl cutters to cut more complex transition cone shapes from card (**Figure 2**). This led me to the idea for the HEXA body tube. Why not create a rocket with a cut and scored foldable tube leading to a lightweight,interesting design?

The HEXA consists of two card hexagonal tubes, the upper payload section and the lower main body tube. The great thing about a hexagonal shape is it lends itself well to a three-fin design as you can use every other edge/corner of the hexagon for equal fin spacing. I decided early on to add the fins to the corners rather than the flat faces as the overall profile height of the fins adds to the stability margin and the centre of pressure (more on that later) which meant I could use smaller fins for less weight and drag, but get the most performance out of them.

Internally, HEXA uses all 3D-printed parts. Using FreeCAD I designed some internal near-hexagonal parts. Why 'near-hexagonal'? Well, I realised early in prototyping that the side of the hexagonal body tube where the tube overlaps with the glued flap is thicker, and this needs to be accommodated. Therefore, every internal hexagonal component has an edge where around half a millimetre has been shaved away from the surface. This can be hard to identify visually, so I got into the habit of adding a mark in the parts to help quickly orient the component.



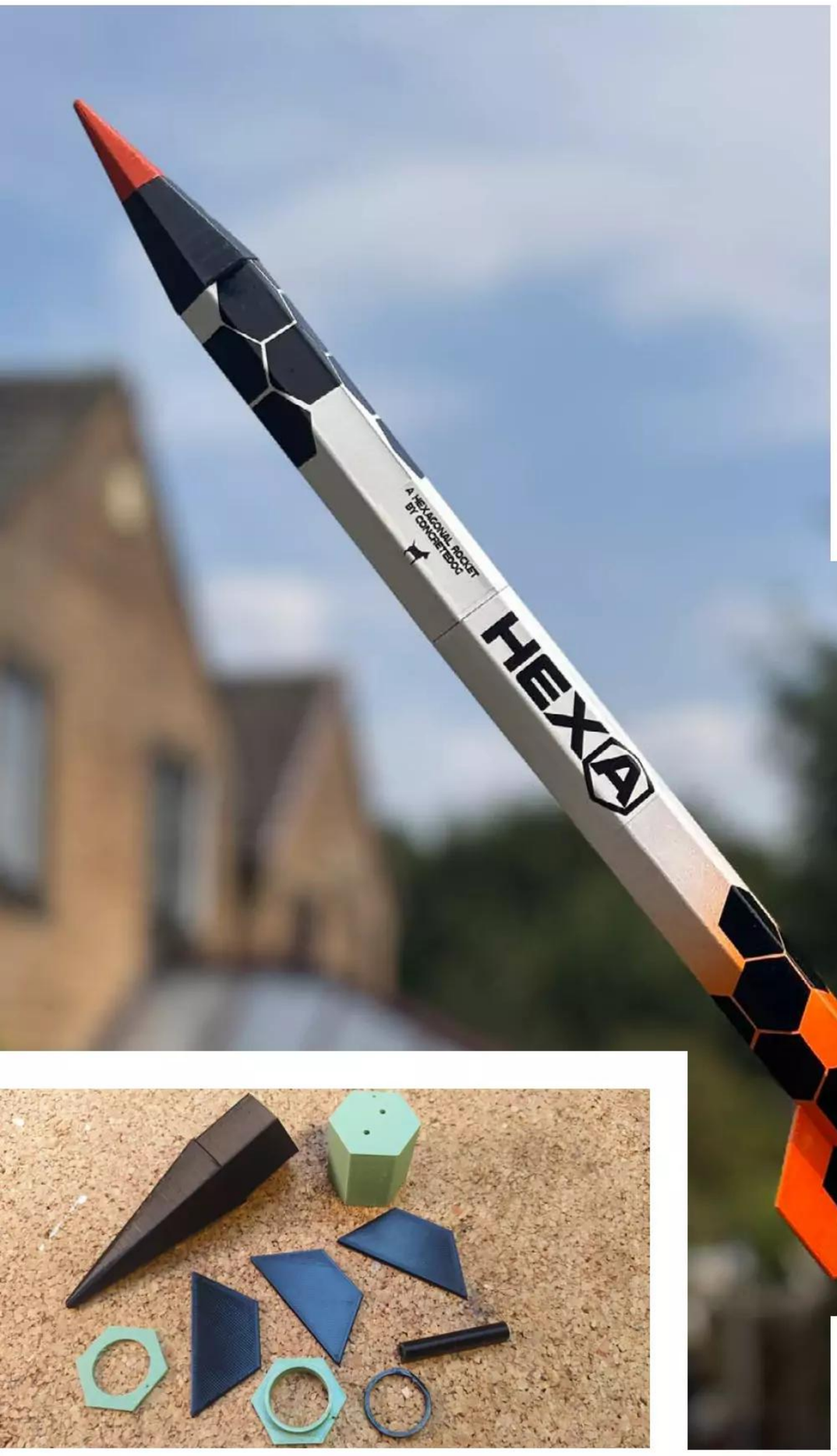
Figure 2 an endurance parachute competition style cardstock rocket

Creating a hexagonal nose cone takes a little bit of thought. Like many items, FreeCAD probably offers numerous approaches that could be used to create a given geometry. I settled on an approach where I drew the base hexagon of the widest part of the nose cone in the XY plane and then created a separate sketch containing a small circle. I moved the small circle sketch so that it was 66mm above the base hexagon sketch and then used the additive loft tool to loft between the two sketches. This gives a pleasing sweeping form from the hexagon towards a circular tip. Obviously we don't want the flat upper surface, so I created a sketch in the XZ plane and drew the cross section of an arc from the edge of the circle to the centre line of the design. I closed this arc with some straight lines and then could use the revolve tool to extrude the sketch around 360 degrees to create the nose cone tip (**Figure 4**).

With more conventional tubular rocket designs, adding the fins is a little complex in terms of getting them on straight and aligned with the centreline of the tube. The hexagonal tube has some advantages here, in that you can use the clearly defined edges of the hexagon as a mounting guide. In the CAD model I even designed in a 120-degree internal groove on the base of the fin so that the fin will centre on the edge perfectly when being glued on. Adding this groove

Figure 1
A HEXA kit built and beautifully finished by Pete Waddington

Photo credit: Aeryn Waddington

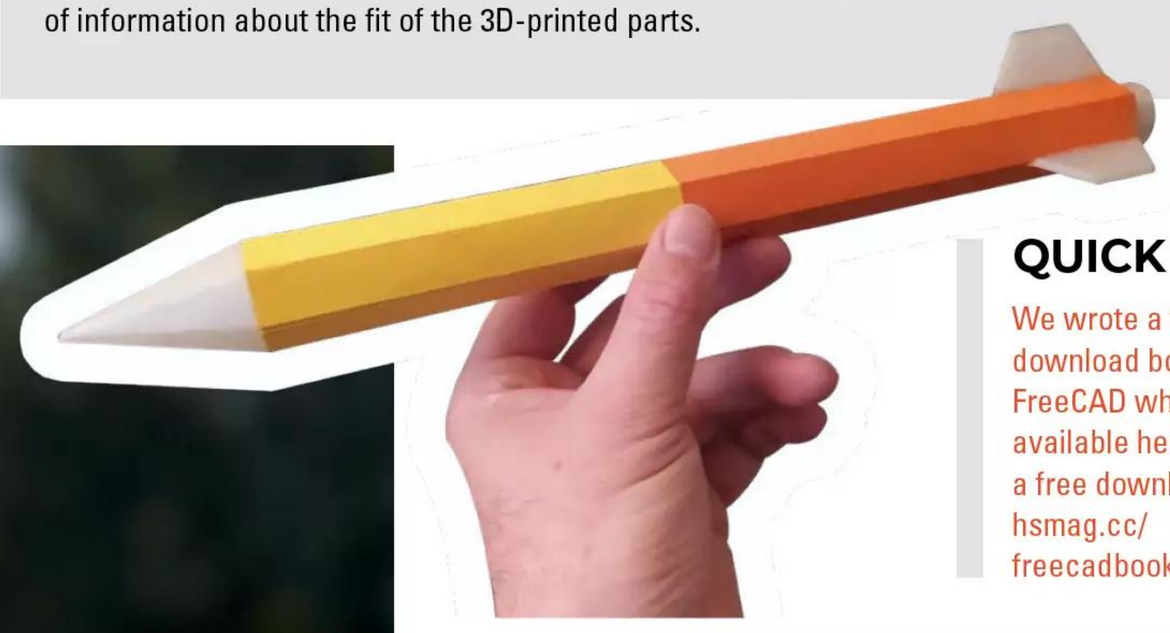


ROCKING ROCKETRY

For more traditional round tubed rocket designs I'd often jump straight into Openrocket, a free and open-source piece of software that allows you to design and simulate the flight of rockets. It's an excellent tool as it automatically calculates the position of the centre of gravity and a point called the centre of pressure. You adjust the design to get these points in a stable arrangement where the centre of pressure is a small distance behind the centre of gravity. However, Openrocket isn't able to simulate weird-shaped rockets such as those predominantly made from hexagons!

For this design I relied slightly on my previous rocketry design experience, knowing roughly what size fins and length of tubes looked similar in dimension and proportion to round-tube rocket designs I'd built. I also later used a classic old test, the swing test, to ascertain if the rocket was stable. A swing test is where you attach a string to the rocket airframe at the centre of gravity point and swing it around you in a large circle. If the rocket is stable it should turn into the direction of travel and keep its nose cone pointing forward. Anything else, and it's not a stable design!

The first full prototype was made in orange and yellow card with PLA printed parts. I realised that I would definitely need to lengthen the design which would give me more space inside the lower body tube for a parachute but also move the nose cone mass further forward, in turn moving the centre of gravity forward. While this first prototype would never fly, it totally validated the cut-and-score tube approach and provided a lot of information about the fit of the 3D-printed parts.

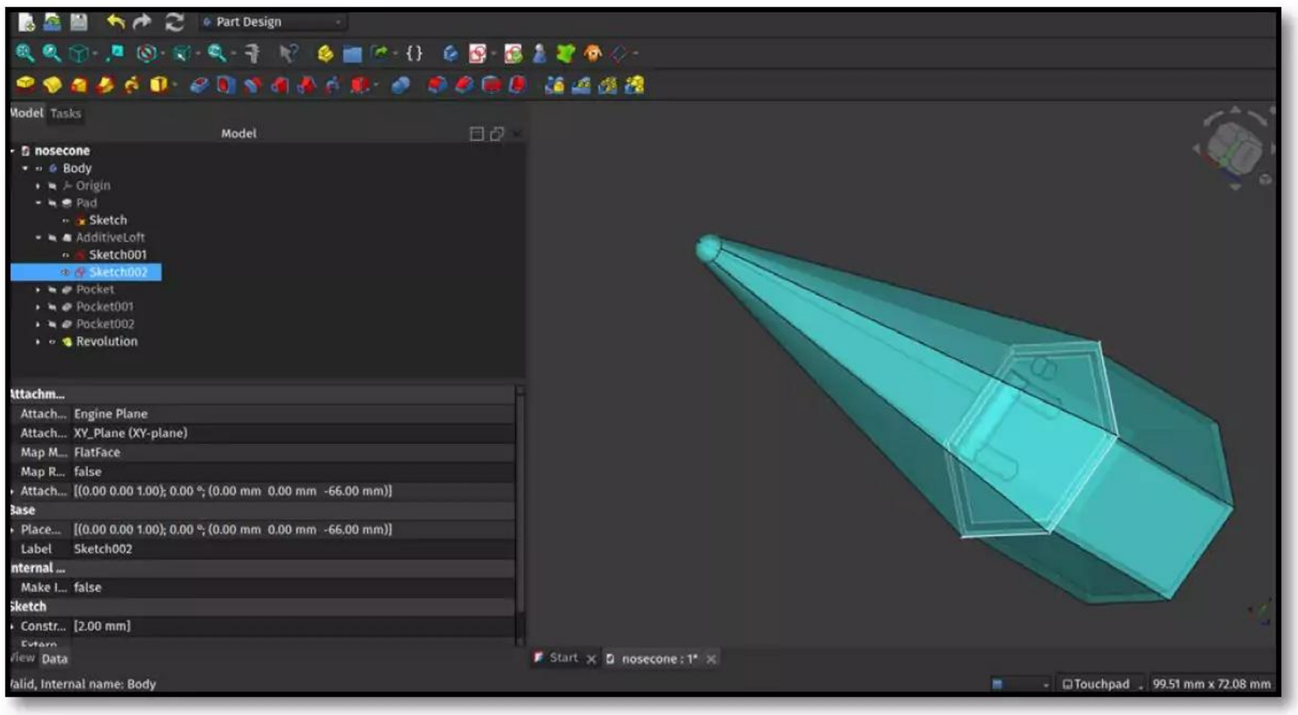


QUICK TIP

We wrote a free to download book on FreeCAD which is available here as a free download: hsmag.cc/freecadbook

Figure 3
The 3D-printed parts needed to assemble a HEXA rocket

Figure 4
The HEXA nose cone designed in FreeCAD



means you also maximise the glue contact area for each fin, and it leads to a pretty robust mount. Internally a printed bulkhead is placed level with the upper end of the fins and with the motor mount bulkhead at the bottom of the rocket is all becomes pretty strong.

The tube sections are simply made from card stock. The design is made primarily for 220gsm card stock which is pretty commonly available, but in reality any thin card could be made to work with a few tweaks of the design files. While it would be →



Figure 6
Using the cut and score body tube design means any decorated card is fair game for rocketry!

Figure 5
A small Graphtech vinyl cutter being used to cut and score the card body tube components

totally possible to cut and score these tube designs by hand, as essentially they are rectangles scored with vertical lines, it's much easier to cut them using a craft/vinyl cutter, as in **Figure 5**.

We looked at using vinyl cutters with Inkscape back in issues 58 and 59 of *Hackspace Magazine*. One of the many wonderful things about Inkscape is the broad range of community contributed extensions. I use one in particular to drive my vinyl cutters directly from Inkscape, this allows me to dial in approaches with different feed rates, cutting depths and other options. For the HEXA main tubes I can attach a piece of card to a cutting board which is slightly adhesive and holds the work in place. After loading the card and the board into the machine I have one Inkscape project with the score lines which I send to the machine with reduced pressure and depth settings so that the blade scores the card. I then load another Inkscape project which has the external rectangular path and I send the job over with settings to cut through the card fully. To avoid little snags and errors I set the machine to go over the cut lines twice to ensure an accurate cut. While most of the time I've opted for plain card stock, this approach does mean that if you find some snazzy sparkly holographic card in a craft store you can indeed create a disco rocket, seen in **Figure 6**!



Returning to the 3D-printed parts of the HEXA design, all the prototypes, including the first flight prototype, were entirely printed out of PLA. For this level of rocketry and this size of motor PLA works perfectly well as the amount of heat and exposure to the 3D-printed parts is minimal. The PLA versions of HEXA have flown a lot of times, and there are no visible issues or deformations to the PLA components. That said, when it came to creating some parts for HEXA saleable kits, I decided to print all the parts that come into contact with the rocket ejection gases (the charge that fires inside the rocket body tube that blows off the nosecone and ejects the parachute) from PETG. This thermoplastic polyester has a much higher melting point than PLA while retaining a similar density, so it's an easy upgrade that adds more thermal security to the design.

Continuing the journey to a final design, I continued to tweak and reprint and built numerous prototypes and performed numerous swing tests to dial in the stability of the rocket. One great advantage of 3D printing is that you can adjust the mass of objects by increasing or decreasing the infill amount, the internal generated support structure inside 3D prints. For the HEXA rocket, the rocket is just stable if you print the nose cone part with a 10% infill amount. If though you wanted to fly a small object, an altimeter or other experiment, in the payload section you can print a lighter, less filled nose cone to optimise the weight of the rocket further. Eventually I got to the point where I was happy that I had a

flyable prototype. I grabbed and attached a small parachute from my collection and went to the flying field. The HEXA uses 18mm Estes-style motors and in its standard form is designed to fly with a B6-4 motor. It's a great feeling after a lot of development work to see a rocket fly, particularly when it goes well! I got three perfect flights on the first day of testing in excellent conditions. It flies very well, being quite a lightweight design, and achieves a very respectable altitude. At some point I need to do some flights with an onboard altimeter to get some data, but as a simple model rocket it is great fun (**Figure 7**).



I spent some time tidying up the project files and created a large PDF of both build documentation which also contains some tips and techniques around how to print and create your own parts. The HEXA design is completely open source, and also uses a totally open set of software tools, primarily FreeCAD and Inkscape, so it's totally open for people to create their own version with all the files on this repository: **hsmag.cc/hexagit**. I also opted to make a few complete kits and put them on my Tindie store if people want to buy a kit or even just support the project by making a purchase. It was suggested to me that I should apply for Open Source Hardware Association (OSHWA) certification which, if awarded, is an indicator that product/design meets some good practice standards for open source. I'm pleased to say this application was successful and funnily enough HEXA is the first OSHWA certified rocket: **hsmag.cc/hexacert**.

> **One of the best parts of creating open source projects is that you get to see** what other people do with your design

One of the best parts of creating open source projects, and selling kits, is that you get to see what other people do with your design. It's fair to say that my main area of focus is to create functional devices that work well and I put less emphasis on necessarily making a design look great.

I was therefore really pleased that a fantastic UK rocket model builder Pete Waddington picked up on my design. Pete had great fun creating a beautiful custom livery for the airframe which you can see in **Figure 1**. Another notable set of builds comes from Dr Laurence Stant. HEXA definitely has a pencil-like look, and Laurence leant into this using a filament swap technique to make the nose cones look like pencil crayons and using his vinyl cutter to cut add on graphics to emulate his favoured stationery (see **Figure 8**)! I can't wait to hopefully see more images or footage of these HEXA rockets taking flight!

Figure 7
It's always an excellent feeling when a rocket you have designed from the ground up flies



Figure 8
A beautiful pair of scratch built HEXA finished as pencils!

Make a modern mosaic

Maybe you've admired the ancient mosaics in Pompeii or Morocco… now, make your own!



Nicola King

Nicola King is a freelance writer and sub-editor. Her growing vintage stack of crafty tomes is making her very happy.

The art of making mosaics has a long and rich history and, having covered the topic of tessellations very recently, this author thought it appropriate that we now take a look at the tessellation of small tiles, or 'tesserae,' from the Latin word meaning a small square or cube. Going back in history, Roman and Etruscan mosaics were created by highly trained craftsmen, their skills held in high esteem by their contemporaries. Many of us are familiar with examples of their intricate work that have survived the centuries, and the ornate mosaic designs – using hand-cut tiles, covering floors, ceilings, walls, and ornamental pieces – that we can visit in museums. As mosaic makers became more skilled, mosaics became more complex, with tiles set into backgrounds at angles which created texture and reflected back the light. The Vatican Mosaic Studio (**hsmag.cc/MosaicStudio**) is a good example of how the art-form is still preserved, in the conservation of the Basilica's mosaics, restoration of various mosaics, and the production of mosaic works for sale to the public.

Of course, it's worth alluding briefly to the fact that we have another art/craft and mathematics crossover here – for example, in the symmetry and geometry of some mosaics. Mathematical mosaics are a group of shapes with sides of equal length and measurement formed in a repeating pattern and sharing sides at each corner point.

> **Roman and Etruscan mosaics were created by** highly trained craftsmen

Indeed, the creation and analysis of simple mosaic art patterns in school mathematics lessons can assist in the development of matching, sorting, problem-solving, logic, and creativity skills, and the understanding of concepts such as perimeter and area and fractions (e.g. by identifying the fractional amount of each colour in a design), to name but a few benefits. And, as with tessellations, we can find numerous mosaics in everyday life if we just look hard enough.

In this tutorial, we will take a look at how to get started in the accessible handicraft of mosaic making, with simple starter projects based purely on creating something with an aesthetic appeal. It's an ancient art, but anyone can attempt it and give their make a modern take.

QUICK TIP

We used an old white tile to work on to protect our table surface. Great for leaving things to dry on too, once glued, painted, or grouted.



TRENCADIS TECHNIQUE

Catalan architect and designer Antoni Gaudi (1852-1926) is known for his Catalan modernistic work, and he was part of the Modernista movement. Mosaics played a key role in his designs a number of times, and one of the most famous of these designs is Park Güell, a Barcelona park built between 1900 and 1914, which includes gardens and an area full of eccentric sculptures and magnificent mosaic work. It's been said that the enchanting Park Güell illustrates Gaudi at his most playful as an architect, and forms part of his naturalist phase, with bright bursts of mosaic colour interspersed with organic and powerful architectural designs. El Drac, a large mosaic dragon which many visitors photograph on their visit, and one of several animal structures that he created, is a good example of Gaudi's use of Trencadis – a mosaic technique that involves breaking chinaware, marble, other ceramics, or glass materials into irregular shapes, and then reassembling and cementing together those pieces to form diverse patterns covering a surface. Gaudi was famous for using this technique, and the major difference between Trencadis and traditional mosaics is the non-uniformity of the ceramic pieces employed.

Trencadis is also known as 'pique assiette' or 'broken tile' mosaics, but the word 'trencadis' is actually of Catalan origin and means 'broken up'. When you think about it, it makes complete sense that Gaudi would use mosaic techniques: his sinuous architectural designs often involved curved surfaces, and mosaic work was a perfect way to easily cover those surfaces in a bright and exciting way. He often favoured the use of brightly coloured glazed ceramic shards. In a way, you can argue that Gaudi was just recycling broken pottery for a better and more enduring purpose.

This is a technique that anyone can try at home using old discarded and broken pottery – give it a new lease of life!

Above
Slivers, shards, and fragments – a colourful concoction for a unique collage

HAPHAZARD HANDIWORK

If you want to try an almost foolproof mosaic project, there are numerous kits available to buy, where the imagining of the design, the cutting of the tiles, and even the process of adhesion to a mosaic base are all made very easy for you. If, however, you want to take a more unique, bespoke approach to mosaic design, letting your creativity shine through, read on.

We talk in the box to the left about the 'Trencadis' technique, which is essentially the making of mosaics with broken tiles and irregular shapes. We feel we have attempted that technique in our two projects, which follow a very random approach, and employ some irregularly shaped tiles that we purchased online. It's a great way to start dabbling as you learn about adhesives, priming your base etc. without needing any true artistic talent… as you will see. We bought an inexpensive MDF coat hook, and primed the area we were going to stick our tiles to with a sealant. Once the primer was completely dry, we then started to stick down our tesserae (**hsmag.cc/MosaicTileBox**) with a mosaic glue in a fairly random fashion, although we did try to mix up our colours and select pieces that would 'fit' next to other pieces. →

YOU'LL NEED

- **Tesserae/tiles/pottery shards**
  see box on the types available
- **Glue**
  suitable for the project
- **An item to stick the tiles to**
  such as a wooden or ceramic blank coaster
- **Primer**
  depending on mosaic base
- **Sandpaper**
  if required
- **Mosaic tile grout**
  optional
- **Gloves**
  for hand protection while mixing your grout
- **Goggles with good eye coverage**
  only required if cutting tiles
- **Pair of nippers**
  only required if cutting tiles
- **Pair of tweezers**
  optional, but handy
- **Acrylic paint and paintbrush**
  optional
- **Old ceramic tile to work on/protect table**

QUICK TIP

Be patient – creating a mosaic will not be a quick process. Don't try and rush your piece of work, as the results will likely not be as good as if you just take your time.



Right
You can buy wooden coaster bases which have an indent in them, which would make the whole process even easier – the choice is yours

It can be useful if you plan your design to a certain degree before you begin. There should be a good amount of spontaneity when creating mosaic designs of course, but if you sort your tesserae beforehand into groups of colours/tones, it will aid your design process and make it less likely that you will need to try and prise off a piece once you've stuck it down, after realising it's not where you want it. If you try and remove pieces, that can have an impact on other pieces already in place, make a bit of a mess, and frustration may ensue.

> **You really don't need a huge amount of glue,** so use it sparingly

On a related point, do be aware that you really don't need a huge amount of glue, so use it sparingly. Another tip is to use tweezers to pick up and place your tiles, especially if they are very small/fiddly. If you don't, you are liable to get very glue-covered digits, and it is worth talking a little here about the type of glue you need. This author bought a 'mosaic glue' from a well-known craft store, but a cheap PVA glue will suffice if your project is flat and gravity won't be an issue. For example, if you are gluing tiles to a plant pot, you need to be aware that glued pieces will not be flat, as on a coaster, and will be need to withstand a subtle gravitational pull. PVA does not dry immediately, so you have a little time to play with your design before the tiles fully bond. If you are working on curved surfaces, such as pots, then a fabric glue might be worth a try as it bonds more quickly than a standard PVA glue, so tiles are less likely to slip out of your desired position before the glue dries. Do use any adhesives, though, in a well-ventilated area. Once happy with your design, leave to dry for 24 hours before doing anything else to your work.

Below
The various accoutrements that we assembled before we began work. A pair of rubber gloves are also always useful when glue or grout is involved!



COASTER CREATIVITY

It didn't matter with our first piece if the tesserae were of different depths, as the mosaic area was purely decorative. If you were adding tiles to something such as a placemat, where a plate would actually need to sit in a stable fashion on your mosaic work, you would need the surface to be flat, so all tiles would need to be the same depth. That point applied to our second make – a coaster. Again, we primed the a wooden base before applying any tiles and, this time, we used some pre-cut squares to make a border. Once you are more proficient, you can start cutting your tiles to the shapes you require, but this will require specialist cutters and goggles, so it's something to consider later on.

Having sorted our tiles into hues of blue for that coastal theme (as we live by the sea), we then again used our 'random' approach to adhering the tiles. If you are artistic, you can of course draw a design on your base and mosaic around that but, as beginners, we wanted to just get a feel for the art of making mosaics. Having let that dry, we then grouted it, painted around the edge with white acrylic paint, and then sealed the piece to protect it. Sandpapering when all is dry will smooth down any raw edges.

GROUNDS FOR GROUT

You don't have to add grout to a finished piece, but there are reasons to do so. Firstly, it helps bring the piece together and unite the various dispersed tesserae. Grout fills the gaps that you've been unable to fill using pieces that don't fit together perfectly, and arguably gives that 'finished' look. If your work is going outside, grouting also helps protect both it and the adhesive you've used. Finally, you can buy different coloured grouts which will all, ultimately, give a very different look to your work, and are worth experimenting with. Of course, there are alternatives, such as epoxy resin which gives a glossy coating, but it's completely up to you what you use. Whatever you use, try and purchase the best-quality grout and adhesive that you can.



Left Below and Above
Here you can see the difference that some grout makes to a design. We painted around the edge with some inexpensive white paint before grouting, but it really changed the look. Mosaic 'experts' appear to agree that grey grout is a great colour to use on most tesserae as it has the most 'unifying' effect, but play around with colours per your own taste

QUICK TIP

If you break a plate, bowl, or mug at home, keep the shattered pieces, especially if it has sentimental value. You can use them in your mosaics and preserve it for posterity!

When adding grout, first make sure that the mixture is easy to spread and not too runny or too thick – a toothpaste consistency works well. Wear rubber gloves and use a tool to apply it that helps you get into the nooks and crannies – we used a plastic spatula-type implement. Once we were happy with our grouting of our two pieces, we left them to dry lightly for around an hour, and then came back with a damp cloth to smooth over the surface to remove any grouting residue on the coloured tiles. If you wipe it with a cloth as soon as you've grouted, we found that this process just unhelpfully drags wet grout over the surface.



TILING TAKEAWAY

There are lots of things you can do to make your mosaic unique – think about using translucent tiles and maybe a more colourful base to reflect through the tiles. The beauty of making mosaics is that, by their very nature, the designs are 'fractured'. Therefore, it can be quite a forgiving hobby as it's easy for any 'mistakes', as you see them, to become subsumed into the piece fairly easily and not stand out. Remember that, at the end of the day, you are creating a piece of art or something to be used, and you really shouldn't get too hung up about it being 'perfect' anyway. Few people looking at your mosaic will even notice anything that you may feel is out of place. Your mosaic will be completely individual, and that is to be celebrated. →

BASE MATERIALS

So, you have decided to create a mosaic, but what can you adhere your work of art to? Tiles can be glued to many types of backing or, to use a more technical term, 'substrates'. There are a number of options, and you should really first determine the use/purpose of the mosaic, what kind of adhesive you will need to use for that specific base type, and whether it will be outside (and thus require some degree of weather-resistance in the form of a sealant), as each substrate has its own unique properties that need to be considered. Let's explore some of the assorted bases that you can use:

- Wood – this is readily available and inexpensive. If you use a wood base, it is sensible to use some sort of waterproofing sealant or primer on the back and front of the piece before you actually start adding tiles. MDF is a great option, as are hardwoods, but wood is not your best option for outdoor use as it can swell or contract if it gets wet, and then your grout might crack over time.
- Pottery – if you have any old bowls or plates or vases hanging around that you want to transform, pottery is a great stable base to try out. If there is any chance that the finished piece will have contact with water, such as a vase, it will need to be protected with a sealant.
- Glass – consider what you will use the item for. For example, a glass bowl that you intend to use to put odds and ends in will need to have some sort of protective sealant so that the tiles don't get scratched. Also consider using a transparent adhesive to let the full beauty of the piece shine through.
- Old pieces of furniture – if you are considering giving an old table, for example, a new lease of life, you will need to consider what it is made of first. The surface will need to be cleaned thoroughly, and any grease or wax removed. Loose paint can be removed with sandpaper. When the surface is clean, you can apply a primer to prepare it for the mosaic.
- Metal bases – this will need sanding before you can begin, so that the tiles have something to grip to.
- Plastic – these backings are usually smooth, so you would need to clean them first, and then rough the surface up with a coarse sandpaper so that the tiles adhere properly.
- Concrete – perhaps you want to add a mosaic to a durable concrete garden pot that you have cleaned thoroughly first. Again, this will need weather-proofing, so you may need to use a waterproof sealant.
- Terracotta – this will require sealing too, before you add tesserae. If you have a terracotta saucer, for example, it's a great way of getting started in mosaicking 3D surfaces, and would make a fantastic bird bath, once water-proofed of course.

You can also use other backings that lend themselves well to mosaics, including fibre glass mesh which gives you a very lightweight mosaic, or 'Jackoboard 'which is a construction insulation board made from extruded polystyrene foam, and is easy to hang, so would make a great substrate for some wall art.



> **Remember that you are creating a piece of art,** and you really shouldn't get too hung up about it being perfect

Below
Once again, this author was lucky enough to find a very inexpensive second handbook or two on the subject, but visit your local library or charity shop to see what you can find

TILE **STYLE**

There are a plethora of different types of tesserae or tiles that you can use for making mosaics. Listed below are just a few of them, and these are particularly suitable if you are just starting to learn about mosaic making. Many of these tiles are pre-cut, so they can be used straight out of the box, and therefore really require no cutting to shape, although you can try that if you wish. They come in different sizes and shapes, e.g. squares, circles, and triangles. Pre-cuts save the beginner a huge amount of time:

- Vitreous glass tiles – these can be found in a massive array of colours, so you will be spoilt for choice. They also come with glitters and in metallic hues. The main advantage is that they are very inexpensive, always a plus when you are beginning a new hobby. Also, these tiles are very durable, so are great for projects that might eventually live outside. A disadvantage is that the back is usually ridged for better grip (as they are often used in bathrooms) but if you want to cut them, the ridge can be problematic and the small pieces can wobble around a little when you are trying to stick them down. Vitreous tiles are great for using in the background, and then you can use more expensive tiles more sparingly in the main design. They can be cut, and won't produce as many shards as other glass tiles if you do cut them.
- Smalti – this is a type of glass tile that is handmade in Venice. Glass is melted, poured onto a sheet, pressed, and left to cool. Then it's chopped up into small pieces.
- Stained glass tiles – not overly expensive, but they have sharp edges, so be cautious when handling. They give a lovely unique effect and are thinner generally than other types of tile.
- Mirror tiles – obviously, a type of glass, so take care when handling or cutting.
- Glass nuggets/gems and millifiori – nuggets are great for adding features to a mosaic (such as eyes), but these can't be cut very easily. Millifiori (translation: a thousand flowers) are small glass cylinders that come in a vast range of colours with designs, such as flowers or hearts, incorporated inside them. These tiles do tend to be more expensive, but have an impact on any design.
- Terracotta tiles (unglazed) – a great choice if you want to cut the tiles yourself, as they are fairly easy to cut into shape. These tend to be less bright colour-wise, and have soft, earthy tones, so if you are going for a neutral and understated kind of a look, these will fit the bill.

In addition to the above, you can buy translucent tesserae, which have the obvious advantage of letting light shine through. You can also use beads, crystals, shells, seeds, or broken pieces of porcelain, or you can purchase glass pebbles from craft stores. The best piece of advice that we've picked up while researching this topic is that it is easiest to start with one type of tesserae that you are interested in working with, so they are all uniform, and also make sure they are flat for easy adhesion. Later, once you are more practised, you can delve into making a more 'mixed media' type of mosaic. Focus your intention for your first project, and you will avoid being overwhelmed by the enormous choice that exists in terms of tesserae.





The parts we sell help us reach for the stars

Since first venturing beyond the safety of a cave, we have been natural explorers. Discovering what's beyond the next hill, horizon, and star, is a drive that is innately us.

The parts and services we provide help fuel the next era of exploration, and like you, we can't wait to see what's out there.



Discover millions of parts at digikey.co.uk

DigiKey

we get technical

DigiKey is a franchised distributor for all supplier partners. New products added daily. DigiKey and DigiKey Electronics are registered trademarks of DigiKey Electronics in the U.S. and other countries. © 2024 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

Listen to Raspberry Pi

Hear *sweet music* and *soothing sound* from your Raspberry Pi

By MC Rob Zwetsloot

Raspberry Pi can be a lot of things – a home assistant, a home theatre PC, an alarm, a retro games console, or even just a regular PC. All of those things have one thing in common – they all use sound.

Raspberry Pi is actually great with audio, and we've used this flexibility in many applications that require sound to be played. Like with a lot of Raspberry Pi uses, we can make it better though, through a selection of add-ons, HATs, and extensions to get the most out of Raspberry Pi's musical capabilities.

Audio upgrades

Quick fixes for better sound

Audio HATs

All-in-one add-ons



Raspberry Pi DAC Pro

magpi.cc/dacpro | £29 / $25

The highest fidelity official HAT for Raspberry Pi works on any Raspberry Pi with a 40-pin header and provides 24-bit 192kHz sound through the 3.5mm headphone jack or the stereo RCA sockets. There's also a hi-fi version called the DigiAMP+.



Audio DAC SHIM (Line-Out)

magpi.cc/dacshim | £13/$15

This minimalist DAC is great for Raspberry Pi Zero and Zero 2 but also works on other full Raspberry Pi boards with 40-pin GPIO. It provides a 3.5mm stereo jack at 24-bit 192KHz over I2S and barely changes the footprint of a Raspberry Pi Zero.

Other accessories



HDMI audio splitter

Unlike previous versions, the Raspberry Pi 5 does not include a headphone jack, so the main way to play audio is via HDMI. With an HDMI audio splitter, you can separate the sound from the video signal and plug it into your own sound system.



USB sound card

A very cheap way to add audio output (and input) to a Raspberry Pi, these plug straight into one of your Raspberry Pi's USB ports. If you want to spend more money you can get very fancy USB amps too, so you can hide your Raspberry Pi away.



Bluetooth speakers

No cables required – every full-size Raspberry Pi since Raspberry Pi 3, and the W variants of Raspberry Pi Zero, can connect to Bluetooth speakers very easily. If you have another kind of Raspberry Pi, you can also use a Bluetooth dongle.

Hearing Pico

Raspberry Pi Pico can actually play audio as well with the right additions – like the Pico Audio Pack from Pimoroni (**magpi.cc/picoaudio**). This specific version sits on top of Pico, however you can also make your own DIY 3.5mm output for projects that need a much smaller footprint.

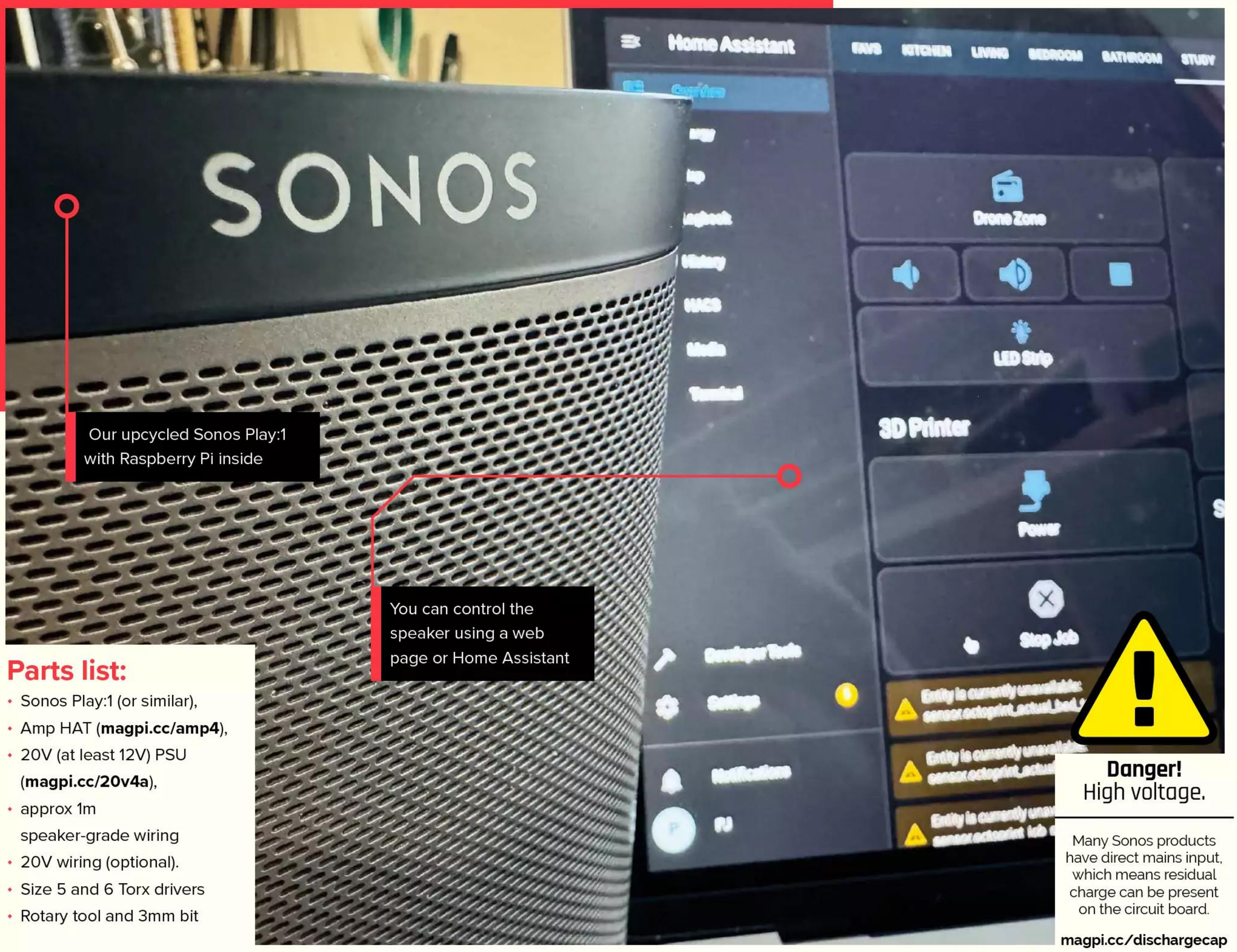


Practical audio projects

Raspberry Pi projects for any avid music fan

Upcycle a Sonos Play:1

Sonos hardware is generally very high quality, with an eye-watering price to match. However, if you find a broken Sonos speaker like our man PJ Evans did, you can install a Raspberry Pi into it and turn it into a highly customisable, fantastic sounding smart speaker. The broken ones tend to be a bit cheaper at the car boot too. This tutorial was spread out in two recent issues of *The MagPi*.



Our upcycled Sonos Play:1 with Raspberry Pi inside

You can control the speaker using a web page or Home Assistant

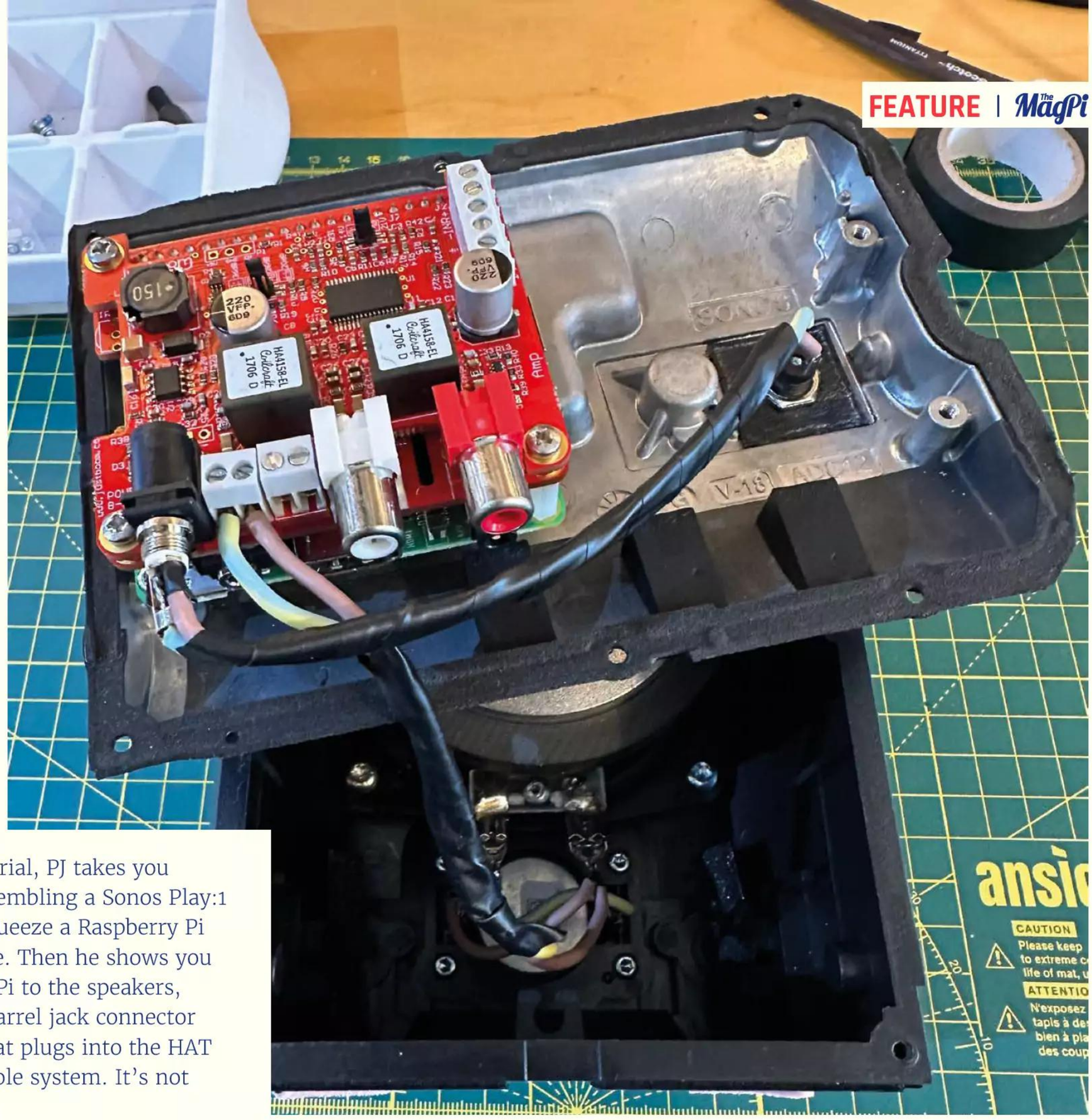
Parts list:

- Sonos Play:1 (or similar),
- Amp HAT (**magpi.cc/amp4**),
- 20V (at least 12V) PSU (**magpi.cc/20v4a**),
- approx 1m speaker-grade wiring
- 20V wiring (optional).
- Size 5 and 6 Torx drivers
- Rotary tool and 3mm bit

Danger!
High voltage.

Many Sonos products have direct mains input, which means residual charge can be present on the circuit board.

magpi.cc/dischargecap

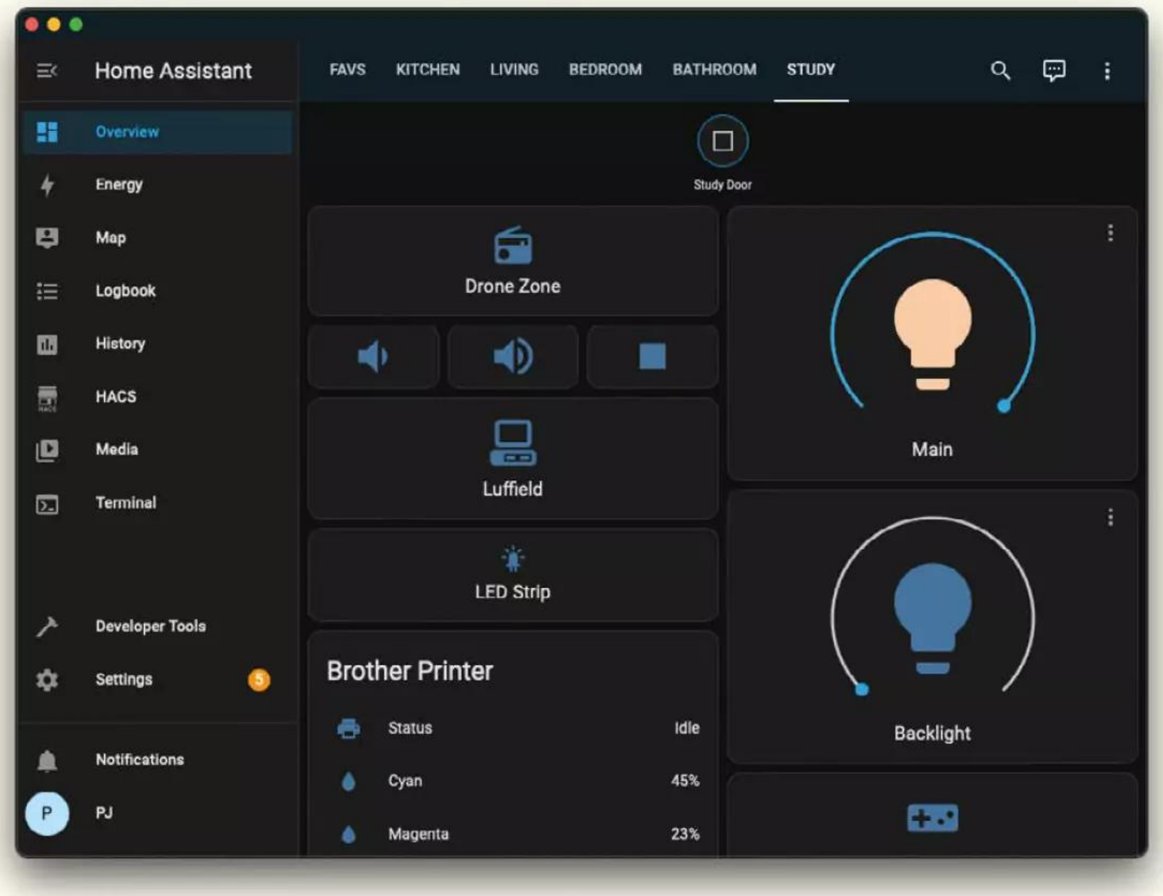


Part 1

magpi.cc/139

In the first part of the tutorial, PJ takes you through (carefully!) disassembling a Sonos Play:1 speaker so that you can squeeze a Raspberry Pi with an audio HAT in there. Then he shows you how to wire up Raspberry Pi to the speakers, and then finally attach a barrel jack connector to the rear of the Sonos that plugs into the HAT to supply power to the whole system. It's not too difficult a build, really.

▲ A Raspberry Pi with an Amp HAT was carefully connected to the Sonos speaker, and a custom hole for the power supply was cut out



▲ By installing Home Assistant, you can make your new smart speaker power your smart home

> If you find a broken Sonos, speaker, you can install a Raspberry Pi into it

Part 2

magpi.cc/140

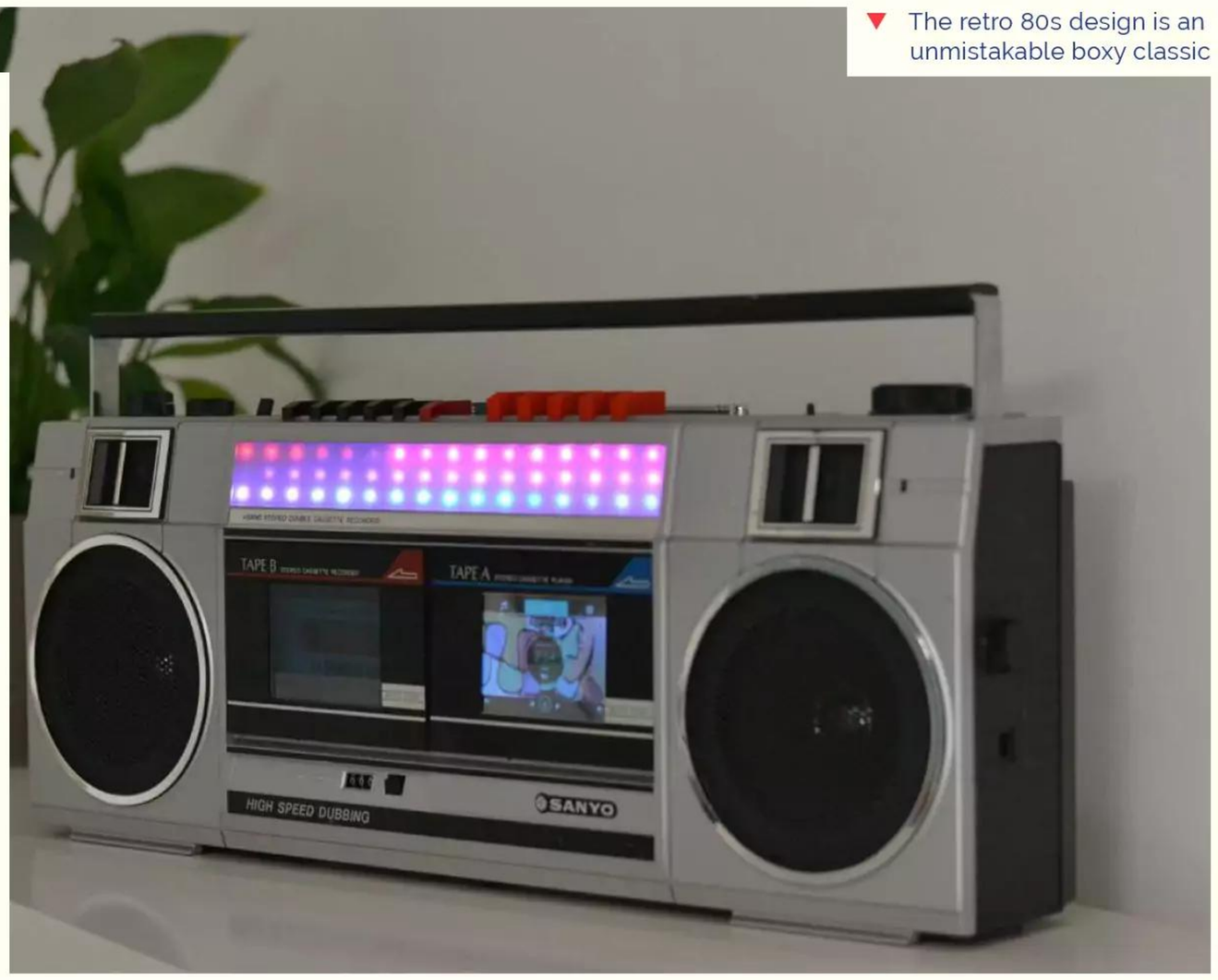
Now the hardware is done, it's time to configure the software – starting by correcting the audio output to the speaker, and then creating audio profiles to play music stored on Raspberry Pi. From there, PJ shows you how to AirPlay music to the speaker, add radio stations, and even make it a hub for home automation with Home Assistant.

Remodelled ’80s boombox

magpi.cc/boombox

By Mosivers

Parts list: Sanyo M W200L boombox, 3.5” TFT touchscreen, 20000 mAh powerbank, 1 m WS2812b LED strip, Arduino Nano, Panel Mount Extension USB Cable, Ground Loop Isolator, DC to DC Boost Converter, 2x 1.8 kOhm and 1x 4.7 kOhm resistors, push-button switch, 1000 µF, ~16 V capacitor



▼ The retro 80s design is an unmistakable boxy classic

While we may be a bit too old to breakdance, we still love this souped-up portable stereo with a touchscreen, LED visualisers, a rechargeable battery, and even a working tape deck. Even the chunky old buttons can be used for playing some rad hair metal.

Maker Mosivers was able to use the original amp and speakers. However, adapting the Sonos:1 tutorial from PJ on the previous page, you can always just make use of the speakers and plug them into a HAT for stereo amps.

PI DIY Music Player

magpi.cc/diymusic

By BalderDragonSlayer

Parts list: Adafruit 1.3” OLED Bonnet, USB sound card, LiPo battery, LiPo battery charger, button caps, 3D-printed case

While a lot of people would be happy with just playing music on their phone, maker BalderDragonSlayer reckons “getting my music away from my phone has been so nice, not feeling like checking every other app when just switching songs has done wonders for my walks too,” and we can see their point.

This is a very DIY build using a Raspberry Pi Zero and a bunch of off-the-shelf components that will simply play MP3s just like the old days. There are a few similar projects to this, some with a twist like the iPod Classic Spotify player (**magpi.cc/ipodspot**) which is a gutted iPod Classic (the one with the wheel) that streams music from Spotify.



▲ A big enough microSD card in this can store more than most old MP3 players

Big sound projects

Get inspired to do something cool with Raspberry Pi and audio

RadioGlobe

magpi.cc/radioglobe

By Jude Pullen

Even with pop culture exports and the advent of the internet, each country still has very unique radio stations that speak of history, tradition, politics, and local culture. Maker Jude Pullen was researching the history of telecommunications for a TV show and developed an interest in radios before coming across Radio Garden (**radio.garden**), a website that lets you point to a spot on Google Maps and listen to the radio there.

Taking an old toy globe, some rotary encoders, and a Raspberry Pi, he made a device that lets you point at a part of the world on the globe and play radio from that spot in the same way as Radio Garden. We love the simple use of two encoders to figure out the co-ordinates.



▲ Learn world geography and find out what people listen to at the same time





◀ This colourful record works as a music controller for the DJ

Spin

magpi.cc/spin

By Arvind Sanjeev

Spin uses an (ethically sourced) AI library of sounds to produce music and mashups. The side buttons select genres, moods, instruments, and samples, and the AI library mixes it up based on your selections. Like a real record player, you need to put the needle on the disc for the music to play – which you can then use to scratch the record.

The wooden enclosure gives the project a cool retro feel, while still making use of cutting-edge technology in a way more interesting than typing 'lo-fi jazz 95bpm' into an AI prompt field.

DingsALot

magpi.cc/dingsalot

By Cyber City Circuits

Like the funny invention of a madcap inventor, this choir of bells is played with a piano (well, keyboard) thanks to Raspberry Pi. Its creator, David Ray of Cyber City Circuits attached 20 handbells to cabinet knobs via strings, which are able to pull on hammers of the various bells.

The strings are controlled by a Raspberry Pi, and can be played with a keyboard hooked up the project. Mute hammers are attached to each bell rig to help control the sound a bit better as well. We don't recommend wheeling it around your neighbourhood at Christmas, though.



Credit: cybercitycircuits.com – The build is very DIY, using various furniture parts

> Not the only project that lets you control a videogame with a piano



You must defeat my arpeggio to stand a chance

Soundfighter

magpi.cc/soundfighter

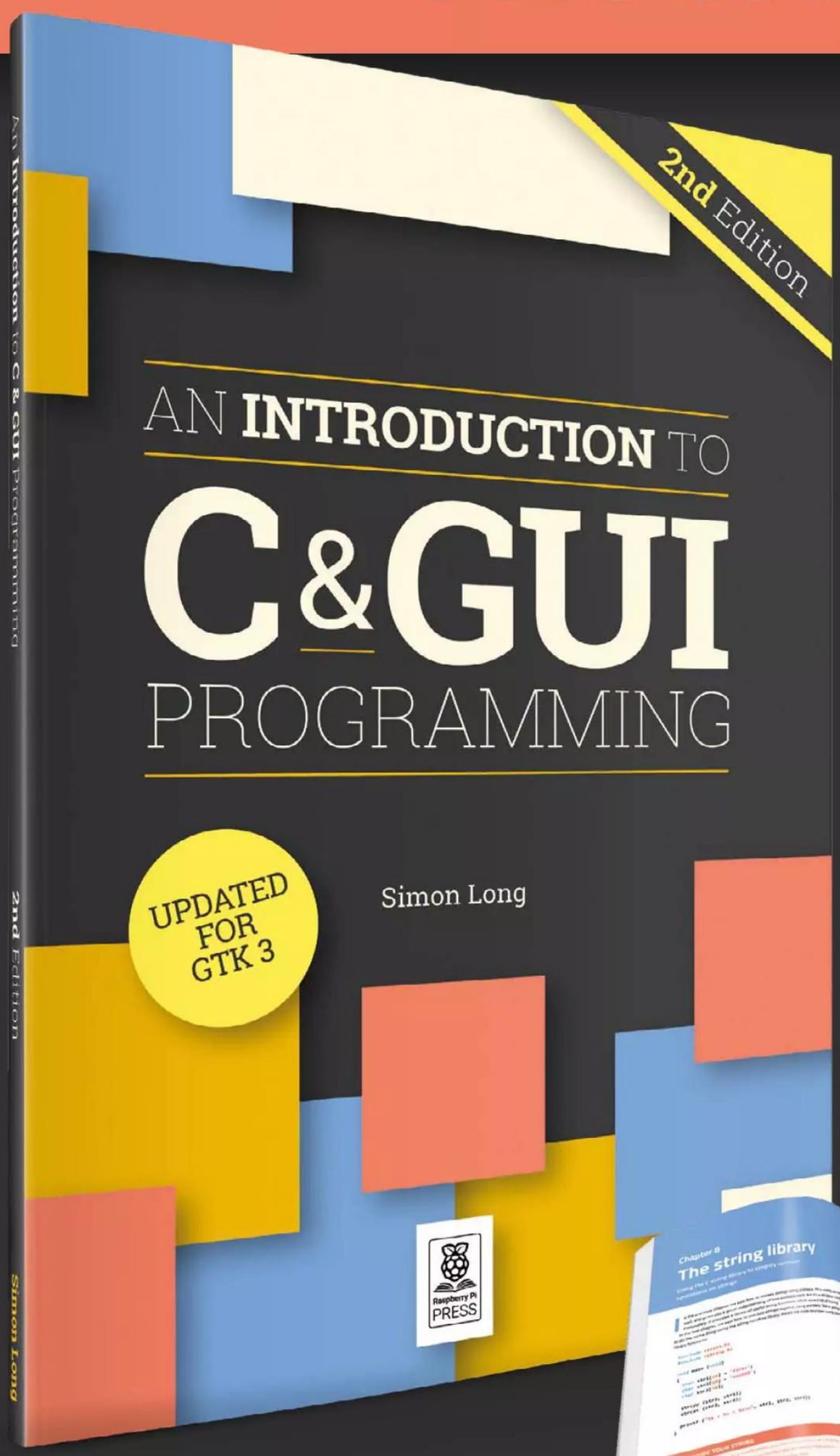
By DDB Paris

Amazingly, this is not the only Raspberry Pi project that lets you control a videogame with a piano. It's the only one we know of where you can challenge your friend though, to a match of *Street Fighter Alpha 3*. Various hammers on the piano are connected to pressure sensors and ADCs so that the signals could be converted to the console controller inputs – which chord is it for a Dragon Punch we wonder?

This was built for the reopening of the Maison de la Radio et de la Musique, the HQ of Radio France. The Maison also has a philharmonic orchestra, which must feel honoured to share its space with such a legendary fighting game.

AN **INTRODUCTION** TO

C&GUI PROGRAMMING

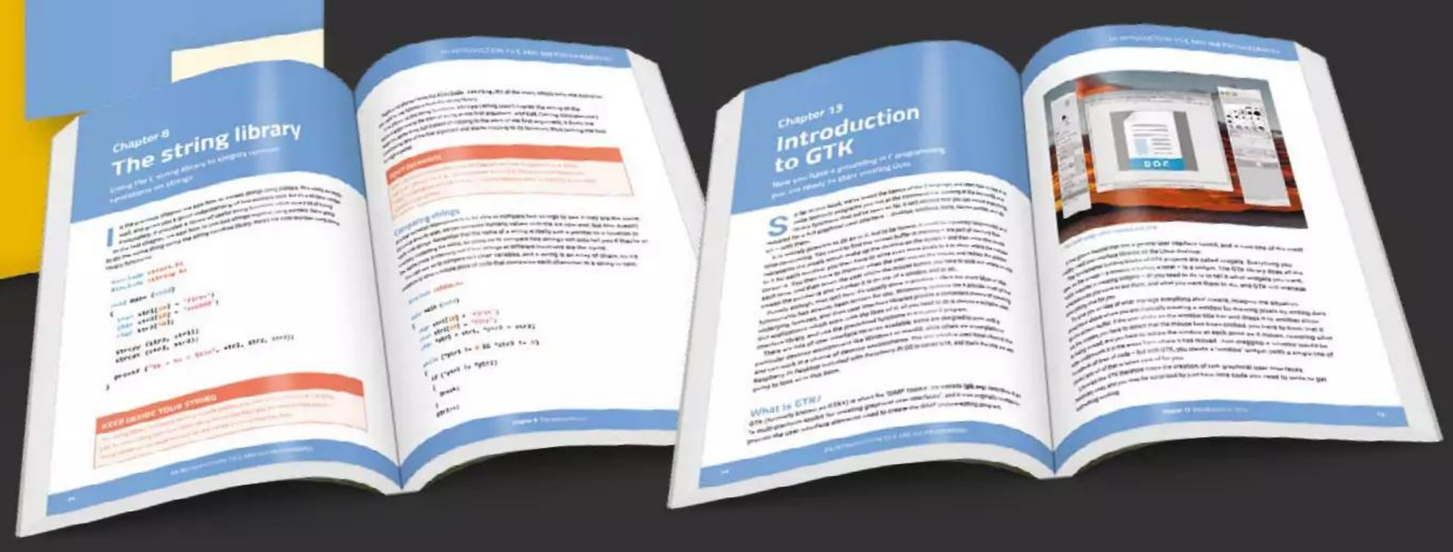


All you need to know to write simple programs in C and start creating GUIs

Inside:

- Create simple command-line C programs
- Control flow with conditions and loops
- Handle variables, strings, and files
- Design graphical user interface applications in C
- Handle user input with buttons and menus
- Use advanced UI features such as data stores and dialogs

£10 with worldwide delivery



Buy online: magpi.cc/cgui

CrowView Note

▶ Elecrow ▶ **magpi.cc/crowviewnote** ▶ £128 / $169

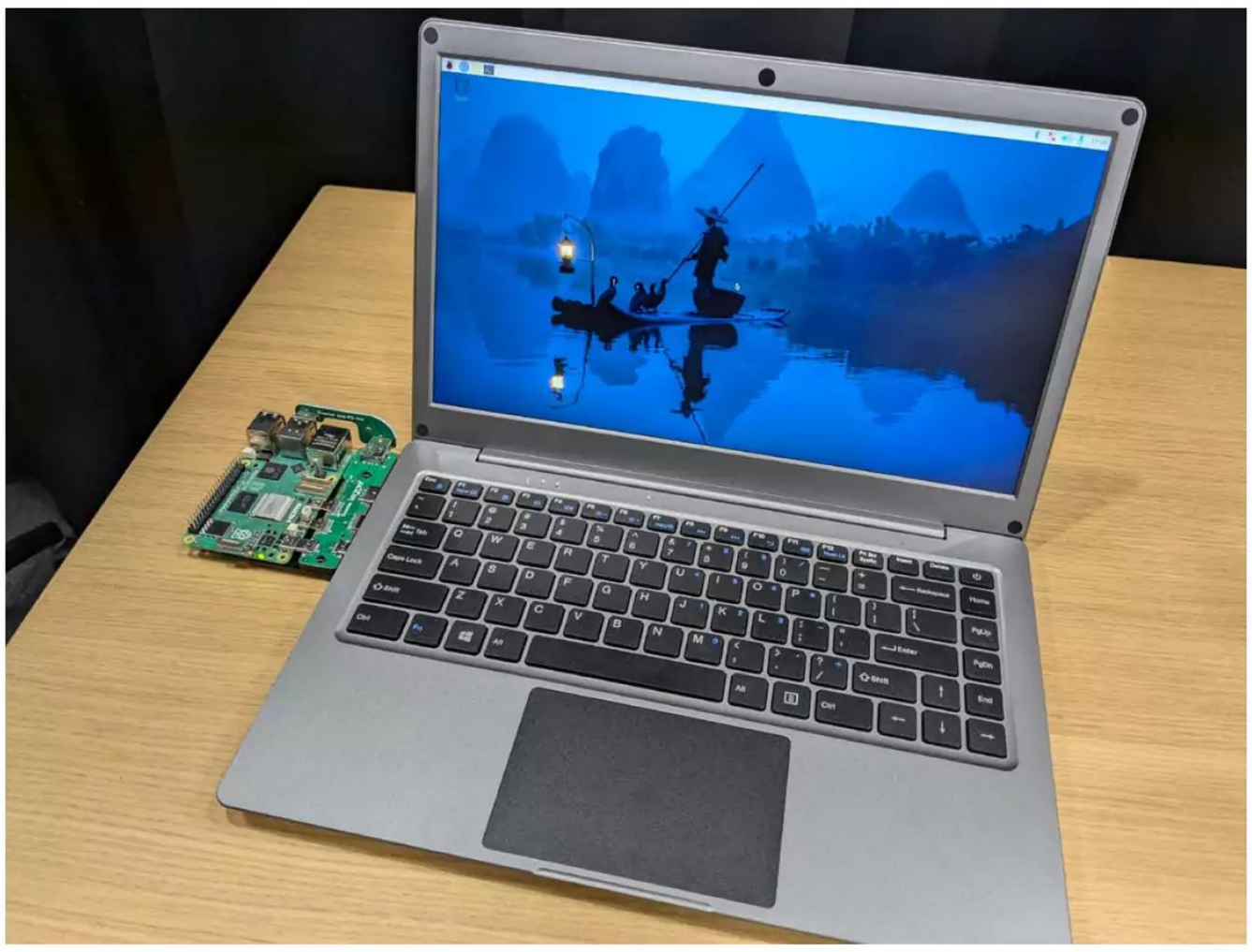
This laptop dock-style monitor has been optimised for Raspberry Pi 5, but can be used with much more. **Rob Zwetsloot** takes a look

SPECS

DISPLAY:
14-inch IPS, 1920×1080

BATTERY:
5000mAh

I/O:
Keyboard, touchpad, mouse, webcam, USB A, USB-C, speakers, 3.5mm headphone jack, mini HDMI in



▶ The CrowView with a Raspberry Pi 5 attached looks a little odd

Laptop docks have been around for a little while now – in this case, laptop-shaped devices you can dock a smart phone or other computer into – and there have been plenty of Raspberry Pi versions over the years too. The CrowView Note is a bit different, as it can be used while attached to any Raspberry Pi, but also has a special way to dock a Raspberry Pi 5 to make it somewhat of a Raspberry Pi 5 laptop.

Connecting Raspberry Pi 5 is easy – a couple of little add-on PCBs slot on, one after the other, and it can then be plugged into the laptop. Power is delivered from the laptop's battery, and a USB 3.0 port is used for all the non-AV I/O. It looks a little awkward attached to the side of the laptop in this manner, however the main PCB does have a base to it, keeping the bottom of your Raspberry Pi a bit more safe from surface contact.

You don't have to use this little dock mechanism – we can definitely imagine 3D-printing an enclosure that sticks to the back of the lid with a few cables between the laptop and Raspberry Pi. However you choose to connect your Raspberry Pi though, it doesn't require any extra software setup and is immediately good to go, which is very nice.

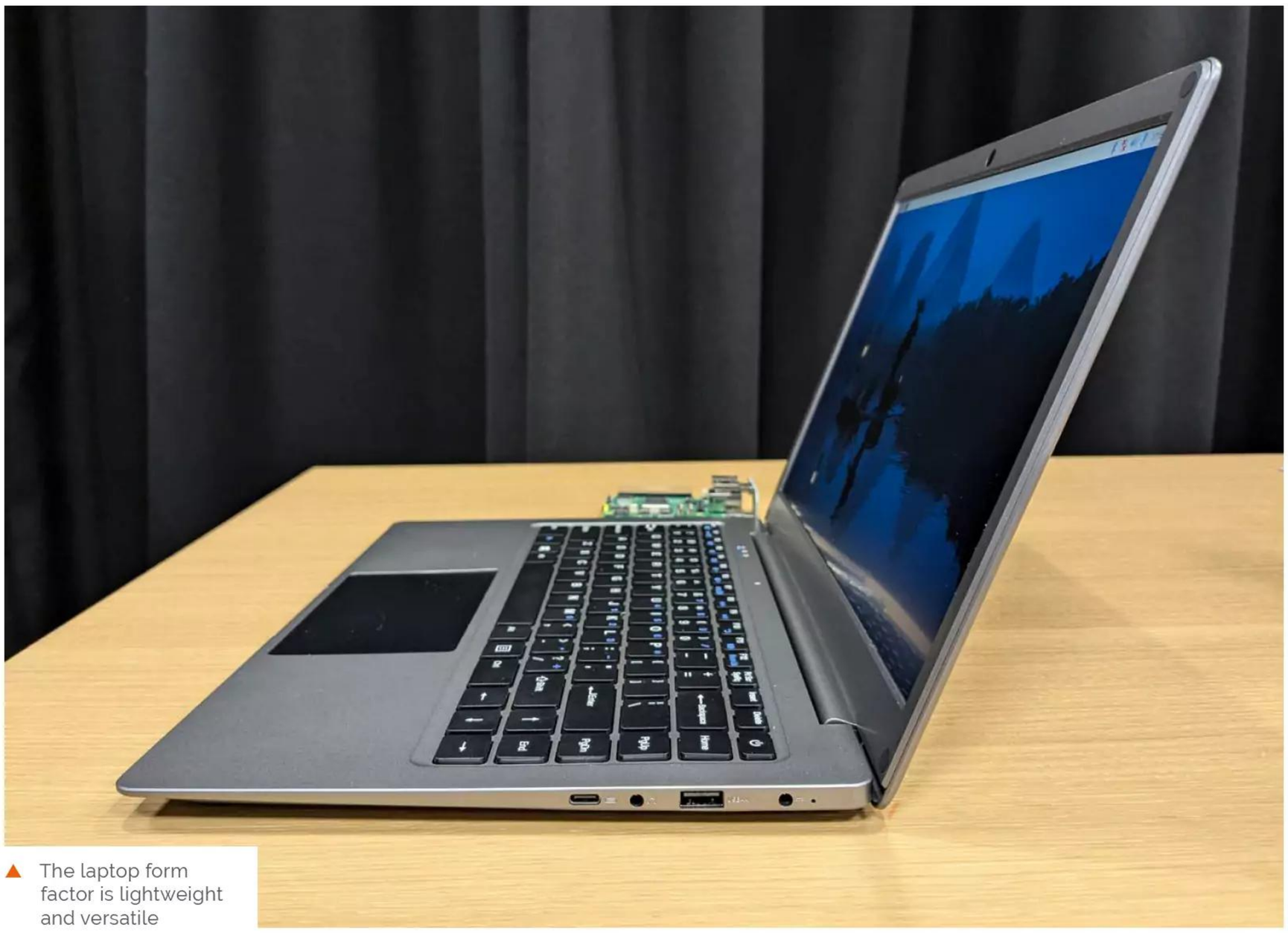
Take it with you

As there are no computer parts inside the shell, the whole thing is very light, even with a Raspberry Pi attached to it (although it does throw off the balance a little). The heaviest part of the assembly is probably the 5,000mAh battery, which charges

Verdict

There are some minor build quality issues but the product itself is fantastic and does exactly what you'd want it do.

9/10



▲ The laptop form factor is lightweight and versatile

nice and quickly, although only with the included DC barrel jack. With a Raspberry Pi plugged in the battery lasts for a couple of hours just fine with fairly normal use, and you can also check the battery with a touch of a button.

Perhaps due to either how light it is, or the materials used in construction, but the Note does feel a little bit flimsy while using it. The mousepad is fine, however the physical clicks for right and left click feel clunky and only work near the bottom of the touchpad. Compared to other normal laptops the keyboard is actually pretty good though, and is nice to type on – it has an extra bit of resistance that makes the keys satisfying to use.

Multi-purpose

We didn't find much use for it with our other PCs – although we don't have any mini PCs like a Mac Mini or such around, which seems like a good fit for the CrowView. It's quite nice for watching videos from a phone though, especially if you're travelling and don't want to lug a regular, heavier laptop around with you.

However, it's as a Raspberry Pi dock that it really shines for us, and despite our concerns with the build quality and awkwardness of the provided dock system, it works more than well enough that we're definitely going to be using it in the future instead of juggling cables on monitors and PSUs. We might use a Bluetooth mouse though.

> "It doesn't require any extra software setup and is immediately good to go, which is very nice"



▲ The entire laptop works immediately with no need for extra drivers

Plasma 2350

▶ Pimoroni ▶ **magpi.cc/plasma2350** ▶ From £12 / $13

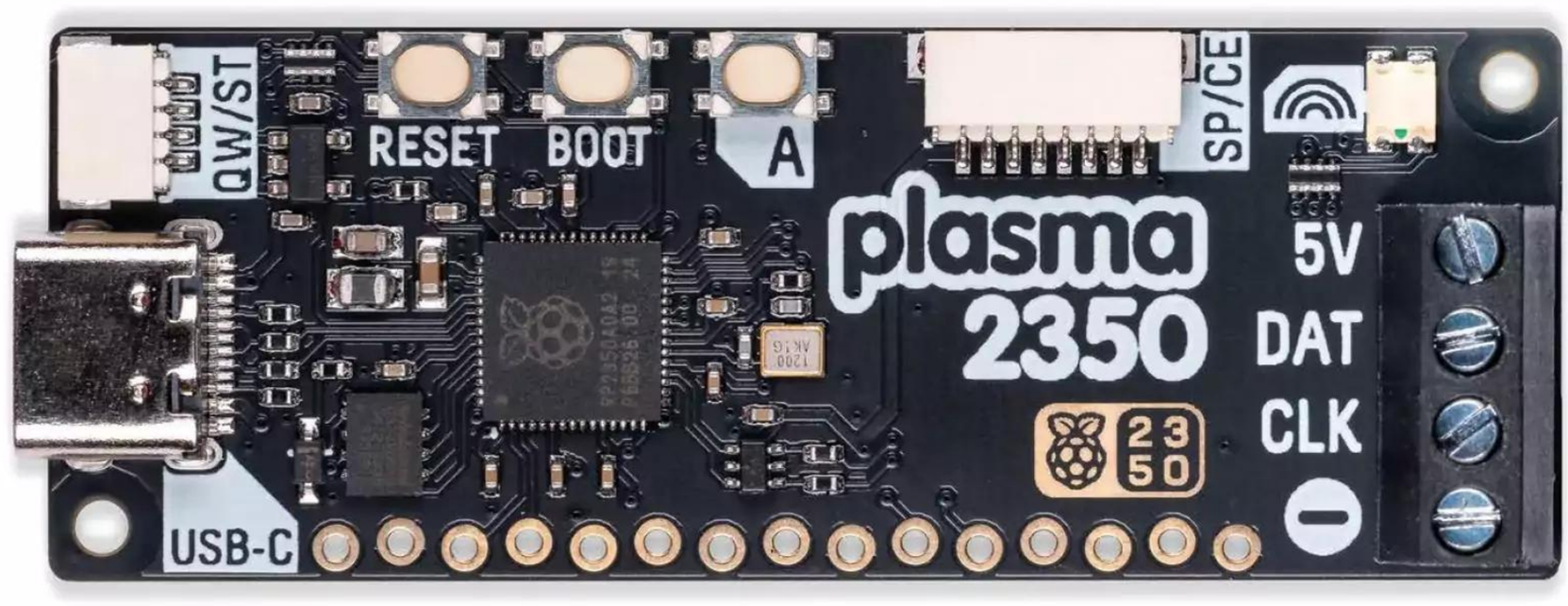
SPECS

FEATURES:
RP2350A processor; Boot, Reset, and user buttons; on-board RGB LED

LED COMPATIBILITY:
5V WS2812/ NeoPixel, APA102/DotStar

CONNECTIONS:
4 × screw terminals, USB-C port for power/ programming, Qwiic/STEMMA QT, SP/CE, unpopulated 15-pin GPIO header

An LED strip controller with even more processing power. By **Phil King**



▲ Only slightly longer than a Raspberry Pi Pico 2, the Plasma 2350 makes use of the same microcontroller chip for fast processing and flicker-free lighting

▶ The Starter Kit includes a 10m string of frosted RGB LED stars that showcase the Plasma 2350's capabilities with some eye-catching effects

Featuring the same RP2350 microcontroller chip as Raspberry Pi Pico 2, the Plasma 2350 can illuminate strings of 5V WS2812/ NeoPixel or APA102/DotStar individually addressable RGB LEDs with some eye-catching lighting effects.

Based on a Dual Arm Cortex M33 running at up to 150MHz with 520kB SRAM, the processor is a major upgrade on the RP2040 used in previous Pimoroni products such as the Plasma 2040 and Plasma Stick 2040 W. While lighting LEDs isn't the most demanding of tasks, the extra processing power may come in useful for more advanced projects linked to breakout boards. There's double the amount of QSPI flash storage (4MB) to play with, too.

You can buy the Plasma 2350 board on its own or in a Starter Kit (£34.50 / $38) with a USB-A to USB-C cable and 10m string of 66 individually addressable frosted LED stars. These look superb when lit up, and are a great way of showcasing the capabilities of the Plasma 2350. We also tried out a long 300-LED strip and there was easily enough current (up to 3A) from the USB-C power connection to light them all.

Wired for light

Connecting your LED string or strip to the board is simple. As on the Plasma 2040, there are four screw terminals on one end: for 5V power, data, clock, and ground. While WS2812/NeoPixel LED strips have only three wires, omitting the clock

Verdict

An easy way to control NeoPixel/ DotStar LED strings with programmable effects, with extra processing power if you need it.

9/10

connection, the latter is needed for DotStar LEDs. A little care needs to be taken to make sure each wire is in the correct terminal and that they're screwed securely.

Despite coming in a slimmer 'gum stick' form factor than the Plasma RP2040, the board manages to cram in many useful features. There's an unpopulated header down one long edge to break out selected GPIO pins, offering access to UART and I2C interfaces, along with analogue inputs, PWM outputs, and PIO state machines. There's also a Qwiic/STEMMA QT connector for breakout boards, as well as Pimoroni's new proprietary SP/CE (Serial Peripheral / Connector Evolution) port – an eight-pin JST-PH connector including four pins for SPI.

> "Impressive lighting effects include blinkies, sparkles, snowfall, fire, pulsing and a lovely rainbow"

Buttons for everything

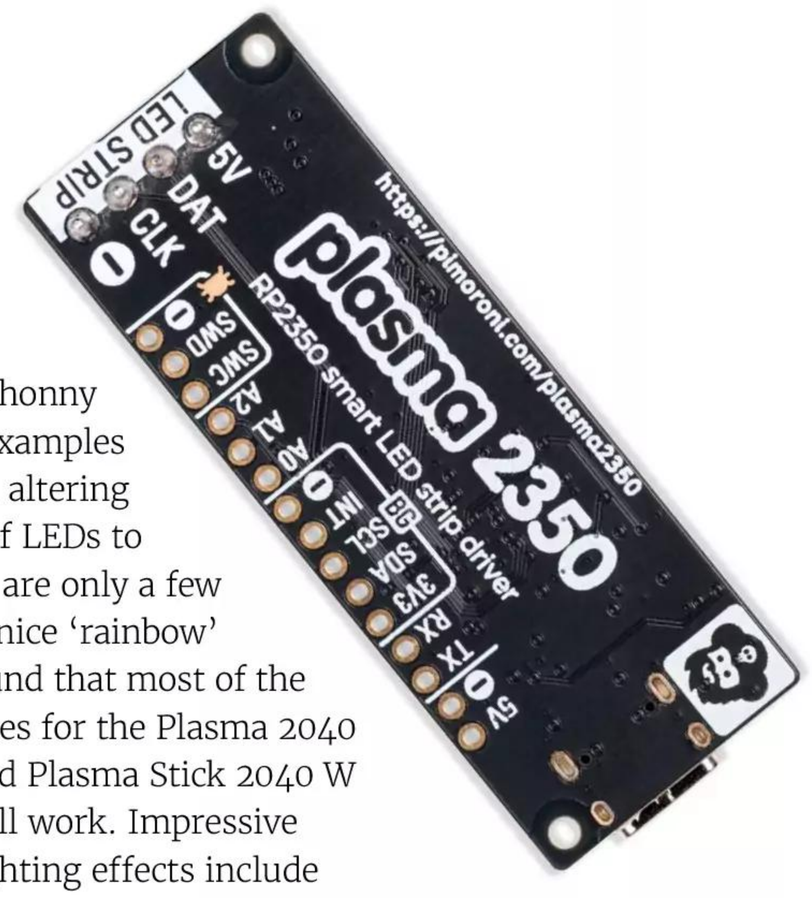
One slight downside of the SP/CE's inclusion is that there's no room for a B user button next to the A one (as on the Plasma 2040), although the Boot button doubles as a user input. There's also a handy Reset button, to save repeatedly disconnecting and connecting the USB-C power. An on-board RGB LED is a nice touch, too.



Before you can start programming some light patterns, you'll need to install MicroPython. In its RP2350 GitHub repo (**magpi.cc/pimoroni2350gh**), Pimoroni provides a custom UF2 file for the Plasma 2350. To install it, connect the board to your computer via USB while holding the Boot button, to mount it as a drive, then drag the file over to it. Alternatively, if you prefer CircuitPython, with which you can utilise Adafruit's excellent LED Animation library, there's also a UF2 image for that: **magpi.cc/plasma2350cp**.

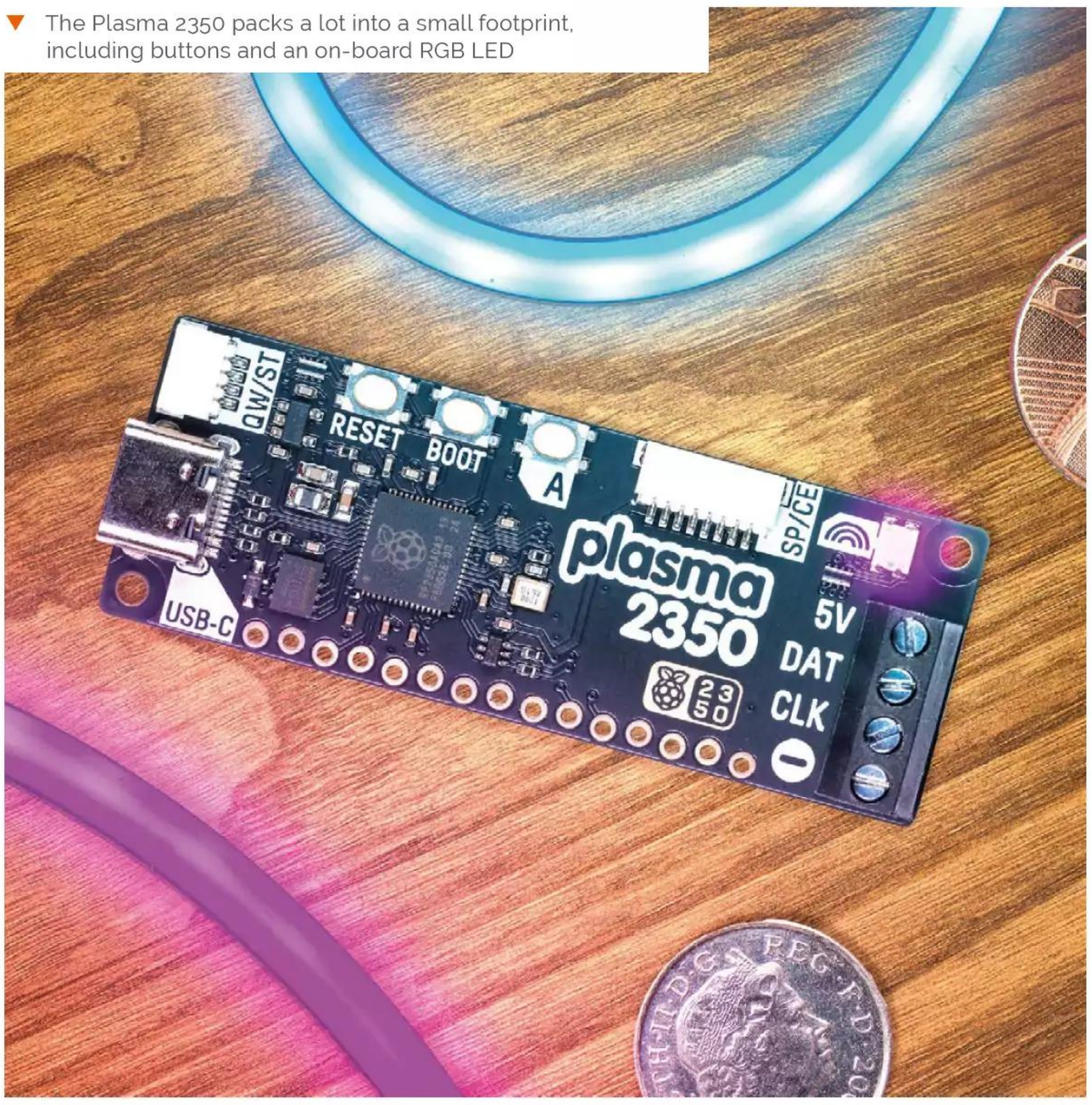
Using MicroPython in the Thonny IDE, we tried out some code examples from Pimoroni's GitHub repo, altering the constant for the number of LEDs to match our string. While there are only a few examples there – including a nice 'rainbow' colour-cycling effect – we found that most of the ones for the Plasma 2040 and Plasma Stick 2040 W still work. Impressive lighting effects include alternating/random blinkies, sparkles, snowfall, fire, pulsing and a lovely rainbow sweeping across the string of LEDs.

Since the Plasma 2350 lacks on-board Wi-Fi, you can't get your LEDs to react to data from the network, but you could always connect a breakout input such as a temperature sensor



▲ The rear of the Plasma 2350 board, showing helpful labels for the LED strip screw terminals and unpopulated GPIO header

▼ The Plasma 2350 packs a lot into a small footprint, including buttons and an on-board RGB LED



10 amazing: Halloween projects

Let's get spooky this October with these malign makes

We're positively petrified by the ghastly games featured in KG Orphanides' frightening feature on page 32. However, if playing vicious videogames isn't on your bucket list for Halloween, we have a scary selection of deliciously devilish alternatives that are sure to raise the dead.



▲ Remote Controlled Billy

See him rolling

This creepy *Saw* doll rides its trike thanks to Raspberry Pi-controlled motors that turn the pedals his feet are attached to, giving the illusion the doll is moving the trike on its own.

magpi.cc/billysaw

▶ Possessed portrait

Grim grinning ghost

This portrait is actually a motion-activated display in a custom-made faux-antique frame. Normally the portrait remains still, but when someone walks by, it becomes ghoulish and screams.

magpi.cc/possessedpic



▼ DIY interactive animated pumpkins

Jack-o-lantern tenors

These 3D-printed pumpkins will sing tunes for trick-or-treaters at the press of a button, with mini projectors powered by Raspberry Pi animating their faces to the tune.

magpi.cc/pumpkinsing



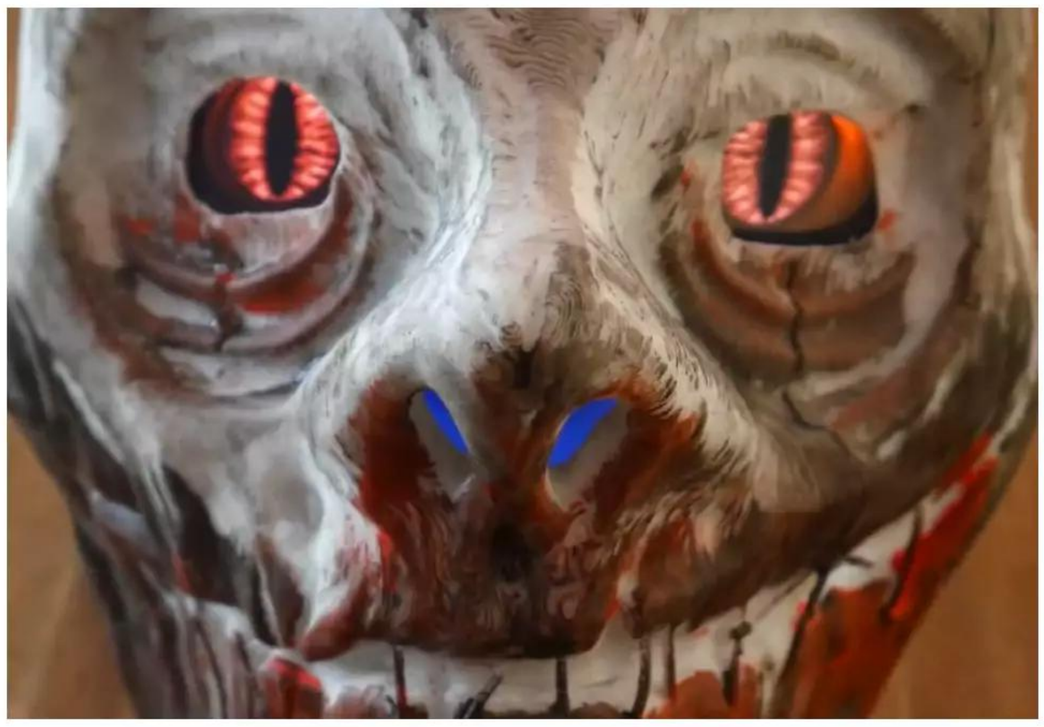


▲ Red light, Green light

Doll of doom

The spooky doll from *Squid Game* has been recreated here as a 3D print, with glowing red eyes and the ability to turn her head through 180 degrees. I'm sure it will be fine if she catches you.

magpi.cc/squidgame



▲ Horror mask

Frightening facade

This scary mask has creepy animated eyes using the classic Adafruit Snake Eyes Bonnet behind a 3D-printed disguise.

magpi.cc/horrormask

▼ Cheap and Easy Halloween Window Projection

Ethereal decorations

Use a Raspberry Pi to play spooky YouTube videos on a white sheet in front of a window to frighten any trick or treaters. You can easily use your own video or make it more interactive.

magpi.cc/halloproj





◀ Eye following Halloween prop

Internet of Terror

This scary pumpkin will follow you around using its eerie Adafruit Snake Eyes. Its name is Gourdan. Make sure to say hi to it.

magpi.cc/pumpkineyes

▼ Interactive Halloween decorations

Remote controlled creeps

These decorations are controlled by a Raspberry Pi via a web app, created during COVID so that kids could have some Halloween fun while keeping their distance.

magpi.cc/hallointeract



▼ GitHub pumpkin

Pulse-pounding pull requests



3D-printed pumpkins with a Raspberry Pi Zero inside that can show you the status of your builds. Hopefully it will warn you off pushing to production on a Friday.

magpi.cc/pumpkingit

▼ Halloween automation

Internet of terror

Automate your house to become truly horrifying when you have network issues, post certain social media hashtags, or get YouTube notifications.

magpi.cc/halloauto



Open Source Hardware Camp

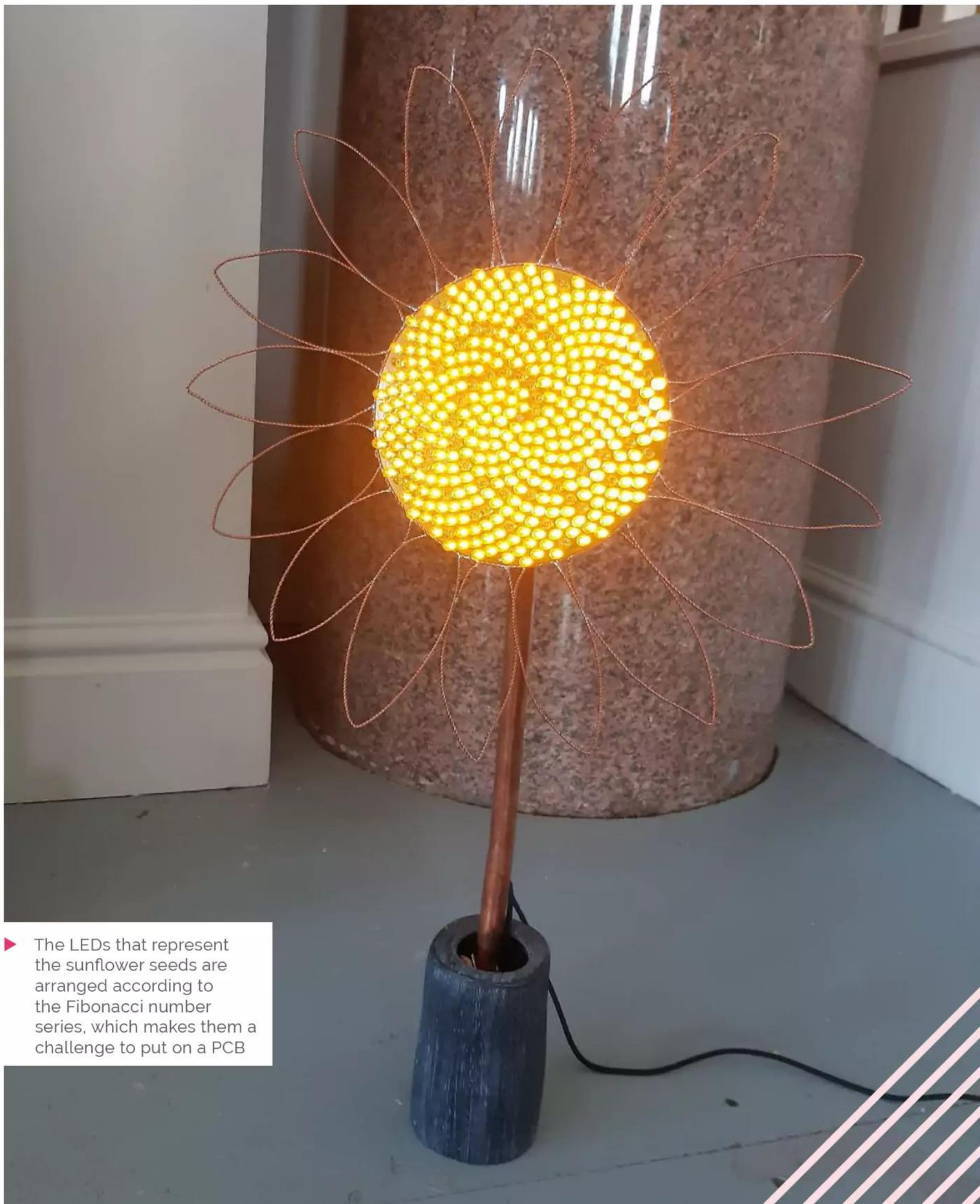
Hardware hacks out on the wily, windy moors. By **Andrew Gregory**

Roughly half way between Leeds and Manchester sits the pretty town of Hebden Bridge. In the 1700s, enterprising young men in this part of the world tried to open-source the money supply of England by coining their own counterfeit currency, which worked well for a very short time, then went badly pretty quickly.

A more sustainable approach to open source enterprises can be found nowadays at Wuthering Bytes, a festival of all things open that has been held here since 2013, with this year's festival also playing host to Open Source Hardware Camp 2024, organised by OSHUG – the open Source Hardware User Group.

Wuthering Bytes bills itself as a festival of technology, which is a pretty broad target, but one that means it's packed with interesting stuff. Most of the action took place from 23–25 August at the Birchcliffe Centre, a former Baptist chapel now in use as a community venue.

Friday kicked off with a talk on Dina St Johnston, founder of the UK's first independent software company, which she started in 1959. After that came computing with human-worn sensors; mainframes; human creativity in the age of AI; and a look at Raftabar the robot, which uses facial recognition (and two Raspberry Pi boards) to attempt to engage humans in conversation. The day also featured an exploration of modular synthesis by musician Loula Yorke; how to poke holes in things with prototypes; and a look at the work being done by Open Innovations, an organisation that's applying open data to policy recommendations in the north of England.



The LEDs that represent the sunflower seeds are arranged according to the Fibonacci number series, which makes them a challenge to put on a PCB

> We often wonder if we could quit our jobs and go on to invent the Next Big Thing

Saturday was the start of Open Source Hardware Camp, and featured a brilliant range of projects. *Hackspace* contributor Jo Hinchliffe gave a talk on open-source rocketry and the tools he uses to build flying machines, with particular reference to open source design software KiCAD. Omer Kilic and Stuart Childs taught us how to go from 10 units to 10,000 with their Adventures in Manufacturing talk. As DIY electronics enthusiasts we often wonder if we could invent the Next Big Thing, and this talk explored "the strange space between engineers, product owners and factories – setting up production lines and working with a variety of suppliers, from prototypes to mass production".

There was plenty for fans of vintage computing: Tony Abbey is part of the team that rebuilt the EDSAC computer at the National Museum of Computing in Bletchley, and he was there to tell us all about that project. EDSAC was one of the first general-purpose computers, built in 1949, and even though the clunking electromechanical technology of those days has been far superseded by microcontrollers that you can buy for pennies, the lessons learned by rebuilding an early computer are well worth a look.

Andy Bennett shared his steampunk sunflower (left), which taught us that getting organic shapes to fit on PCBs isn't quite as easy as it looks. He's influenced by the work of Mohit Bhoite and Jiri Praus, both wonderful makers who have documented their build process to produce stunning open circuit sculptures. In the next talk, Roger Light explained how he built a digital camera sensor, spending £50,000 to make a device capable of capturing images at a resolution of 256×256 pixels.



◀ Open source rockets designed on open source software

- This year's Open Source Hardware Camp was held at the Birchcliffe Centre, Hebden Bridge





▼ Sunday was filled with hands-on workshops





▲ You too can print classical artworks on to your PCBs

Our favourite talk, and one which really encapsulates the brilliance of the one hardware movement, was by Spencer Owen. In 2013, Spencer built a clone of a Z80 computer on a breadboard, which went on to become the RC2014 kit computer. His talk this year was on dye sublimation printing onto PCBs. He's worked out that with the same hardware you might use to print on to mugs and T-shirts, you can print on to the silkscreen layer of a PCB, opening up all sorts of colours and designs. Our favourite bit of Spencer's talk is that he used the process to make a computer with rainbow PCBs, which he sold to raise money for LGBT charities; our second favourite bit of the talk is that, as JLCPCB now offers full-colour silkscreens, he wouldn't have bothered with sublimation printing if he were starting today, but he did it anyway.

That's something we love about open source hardware – very often, the point isn't that you can do it better, or cheaper, but that you've done it for yourself. And we love it that events like this keep happening, where we share the knowledge and enthusiasm that keeps communities thriving.

wutheringbytes.com



Natalie Turner

We've gained another designer – meet Natalie!

> Name **Natalie Turner** | > Occupation **Graphic designer**
> Community role **Magazine designer** | > URL **magpi.cc**

As *The MagPi* grows larger with the addition of *HackSpace*, that means our team has grown larger so we can produce our fantastic new magazine for our readers. This includes having Natalie Turner join us as a full-time designer alongside Sara.

"I have grown up around art and design from very young," Natalie tells us. "My dad was a graffiti artist when he was younger, then became a graphic designer/web developer. He gave me access to his Photoshop where I got to explore digital art. I also grew up drawing a lot from manga books! After college I did an art foundation degree, which is where I decided I wanted to be a graphic designer, so went to Nottingham Trent University to study graphic design and develop my skillset more."

How did you join Raspberry Pi?
I was working part time alongside job hunting after uni, and came across the application for a graphic design role at Raspberry Pi. I then got invited for an interview where I got to meet my lovely team, and speak through a few of my projects. After a couple of weeks I received an email saying I didn't get the graphic design position advertised...however I was offered an internship for six months as a graphic designer! After two months of my internship I was offered a full time position.



▶ Much missed *Wireframe* magazine also benefited from Natalie's skills



◀ Packaging design for something like the Raspberry Pi PoE HAT requires a sharp eye

▼ The official pop-up store tote bag has a fun look



What did you know about Raspberry Pi before joining?
I've known about Raspberry Pi for many years through my dad. On one of his birthdays we got him a Raspberry Pi 4B from the store in Cambridge!

What Raspberry Pi design stuff have you worked on?
I have worked on a variety of projects ranging from packaging to web assets to magazines to events. My first packaging project was reskinning the PoE HAT.

I also designed the tote bags you may have seen at any of the pop-up stores we've had, as well as on past magazines, like a feature page I had the pleasure of working on in *Wireframe*.

Have you made anything with a Raspberry Pi, or have any plans to?
I currently have a Raspberry Pi 5 at home that is in desperate need of a project for it! My dad and I are constantly looking for inspiration, so plans are pending...

What other hobbies do you have?
Over lockdown I taught myself to crochet: cardigans, plushies etc. I even made some cutesy leaf coasters for some of my colleagues. I have also completed two half marathons so far. Unfortunately – or thankfully – I didn't get a ballot place in the London marathon next year.

> "I currently have a Raspberry Pi 5 at home that is in desperate need of a project"

Events in pictures:

Raspberry Pi Community Gathering Taiwan #43

Community and official events in the wild

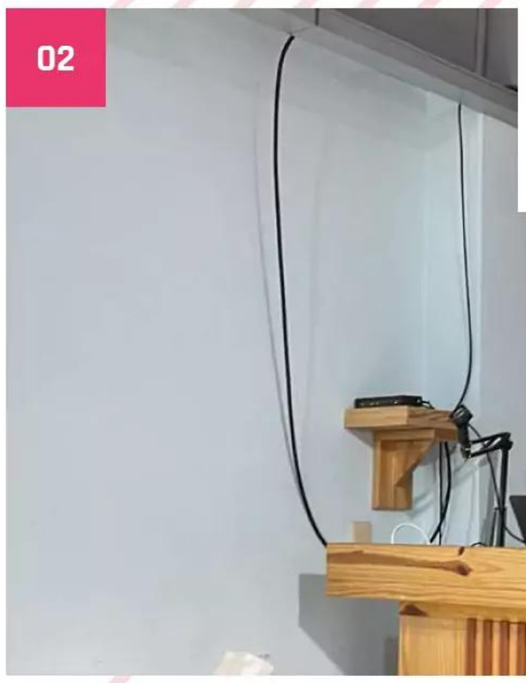
Hosted by friend of *The MagPi* Nai-Wen Hsu, these regular meetups in Taiwan also often feature Masafumi Ohta of the Japanese Raspberry Pi Users Group, another friend of this magazine. A recent event in August had Masafumi show up in person as the attendees talked about streaming music around their home, and the new Raspberry Pi AI Kit.

01. The small venue is packed with Taiwanese Raspberry Pi community members
02. Masafumi talks about the differences between Apple Airplay and UPnP streaming
03. The new Raspberry Pi AI Kit was a hot topic at the event
04. Folks giving talks can get up close with the audience

01



02



FIND OUT ABOUT NEXT MONTH'S EVENTS AND POP-UPS ON PAGE 124

03





04

Manual light automation

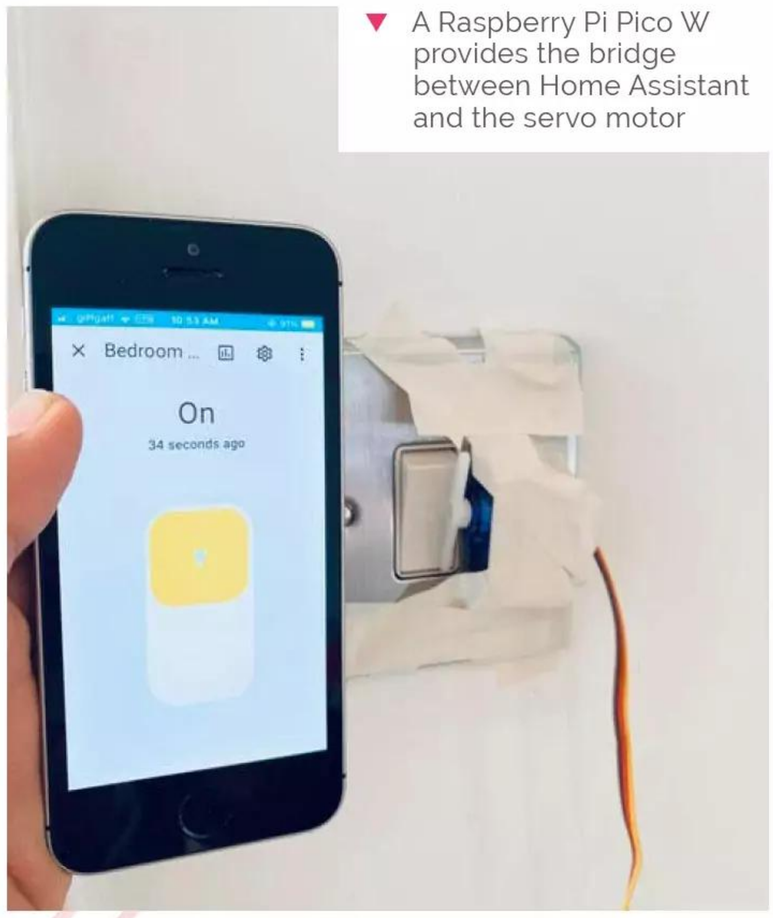
Taking a more direct approach to switching lights on and off with Raspberry Pi Pico

When we do home automation with lights, usually we're connecting to them via software to turn them on or off (or dim, or change colours, etc) as part of a big IoT network. We got an email from Nirvaan, 13, about his low-tech high-tech solution to home automation: physically turn the switch on and off with a Pico W on the Wi-Fi network.

"I started out by making a smart light switch that uses two servos and a Pico W to connect to my Home Assistant (also Raspberry Pi)." Nirvaan writes. "When I pressed ON in Home Assistant (HA), the servo would press the light switch and manually turn it on! I coded this in MicroPython, using a web server to communicate with HA."

He went on to control a dimmer switch in a similar manner, being able to control the level of light with a slider, and another switch which was activated by a wireless button next to his brothers bed. The systems are all battery powered, and simply taped to the light switch.

"I have uploaded all the code and basic instructions for the three types of switches at: **magpi.cc/picowlight**," Nirvaan also mentions. Go take a look!



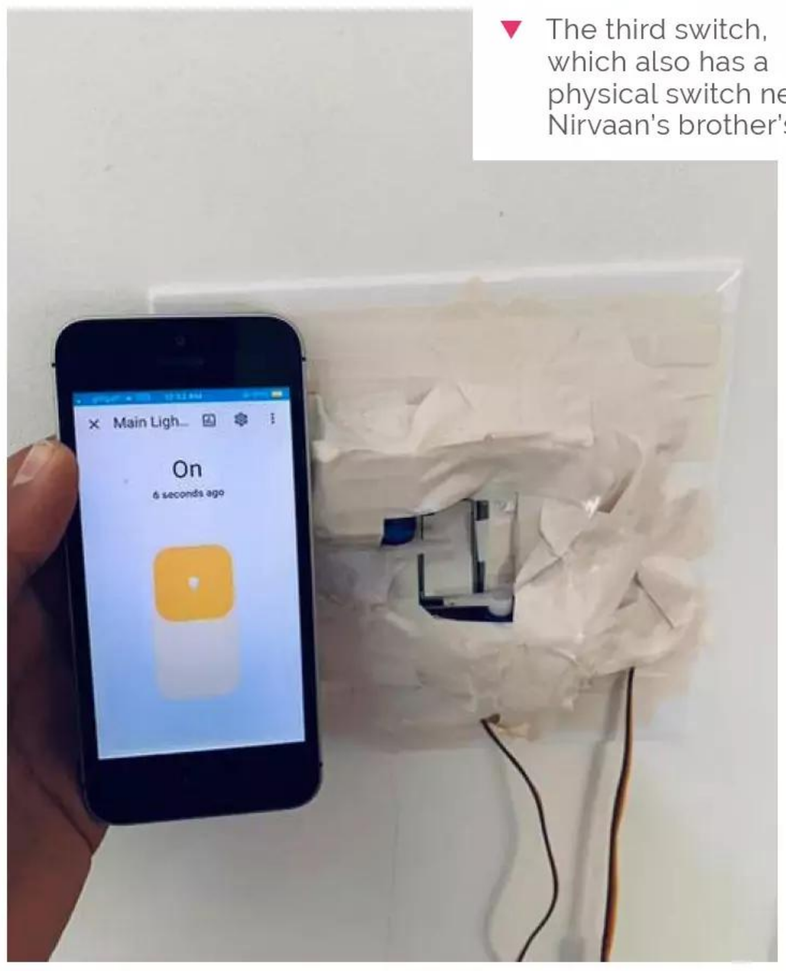
▼ A Raspberry Pi Pico W provides the bridge between Home Assistant and the servo motor



▲ An ingenious way of turning the dimmer switch using a servo and a sliding meter



▲ The tape is necessary to stick everything to the switch



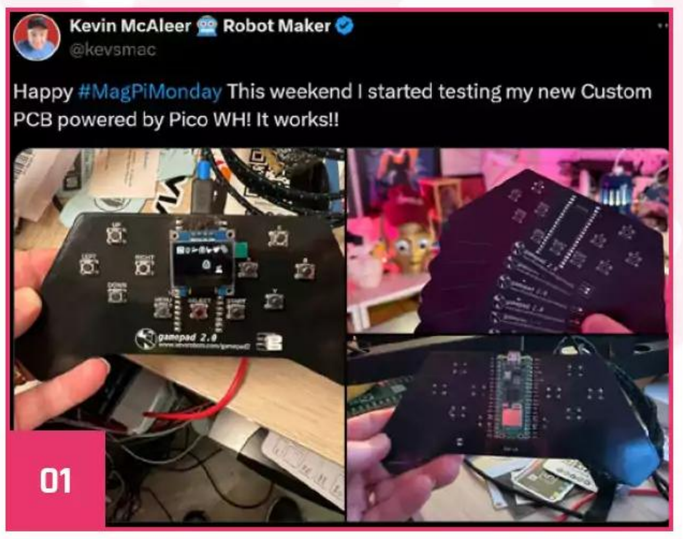
▼ The third switch, which also has a physical switch next to Nirvaan's brother's bed

MagPi Monday

Amazing projects direct from social media!

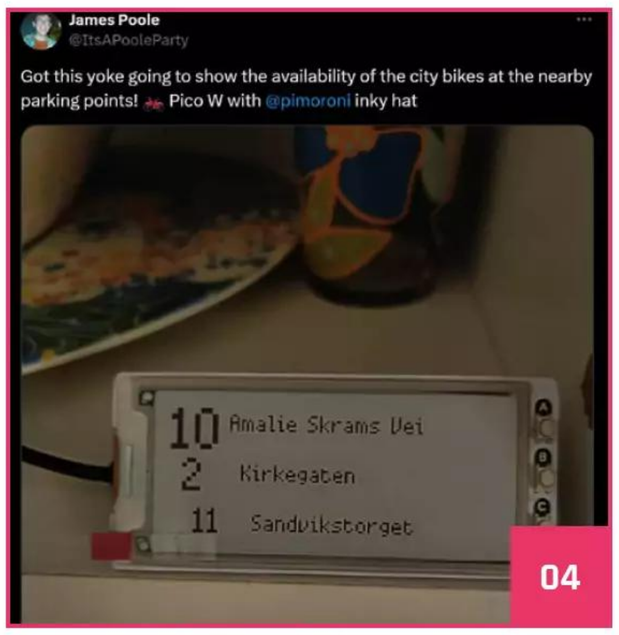
Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they have made.

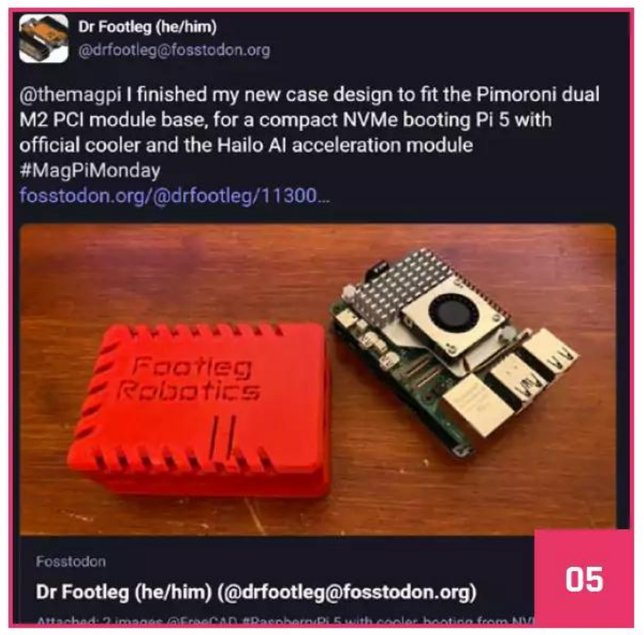
Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!



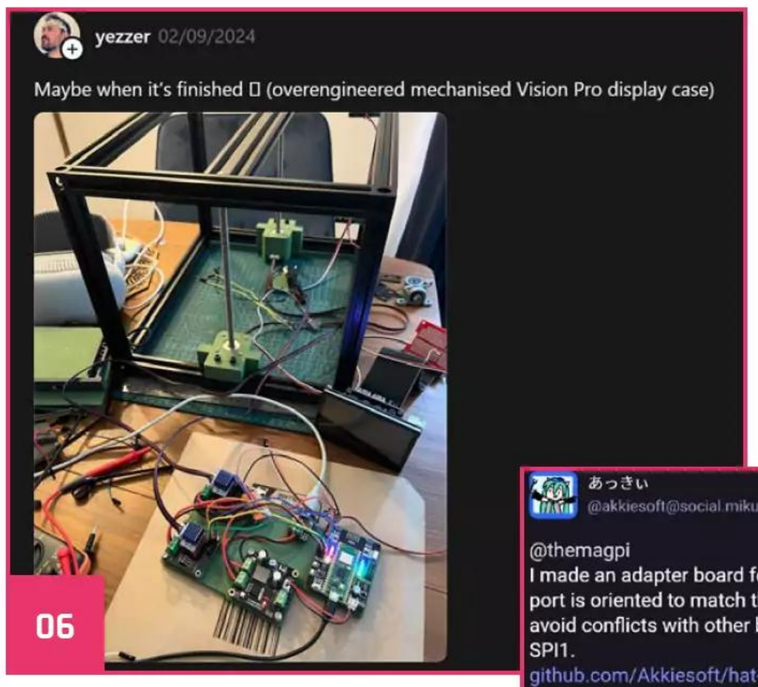






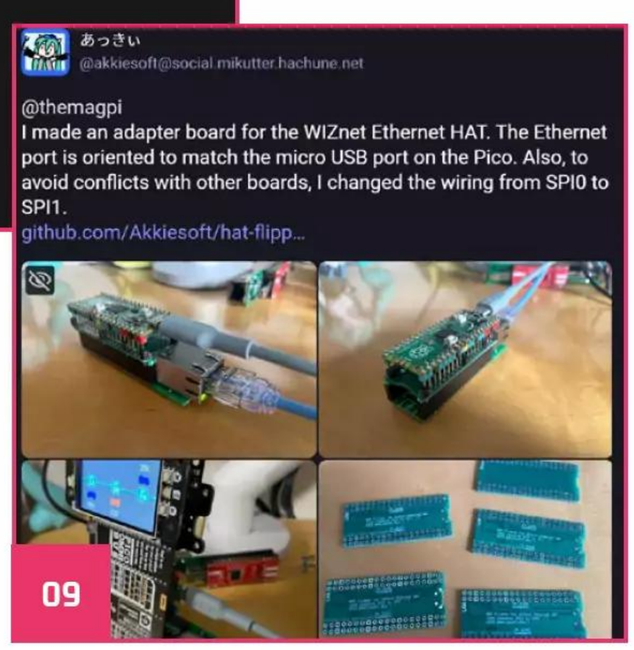


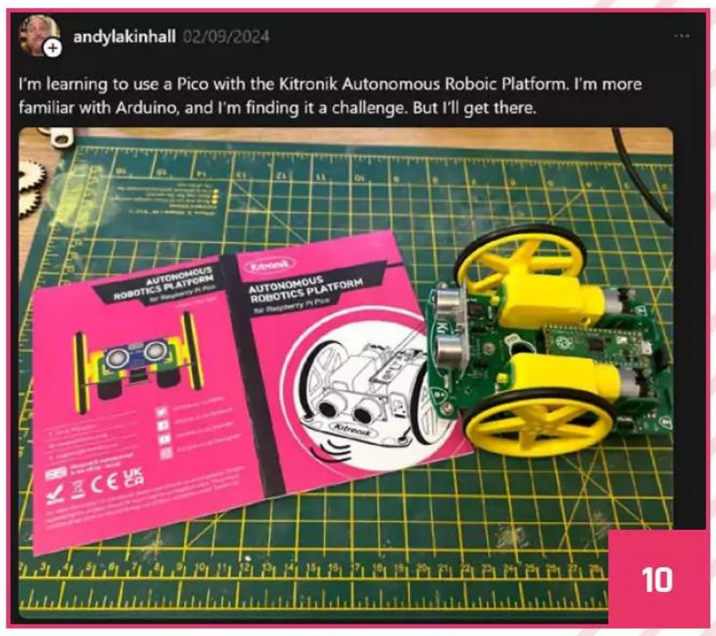
01. We love some custom controllers! We're interested to see where Kev's goes...
02. Not just any mysterious ghost, this madame resides in a haunted house in the happiest place on Earth...
03. We wonder why Mando Rick needs a Geiger counter? Should we all have one?
04. We would love one of these for our monthly office visits as we often find the rentable scooters and bikes all taken
05. Dr Footleg is a master of 3D printing, and we love this new all-in-one case for his setup
06. Yezzer is slowly revealing to us an interesting build...
07. This is quite a big board and we think that's great. Perhaps you could use it as an electronic DJ platter?
08. Ah, the classic phone dock that popularised the concept. The CrowView Note we review on page 104 seems to be a spiritual successor
09. This Pico stack looks like it's ready for some serious business
10. We love this little robot platform – a great way to learn electronics











Crowdfund this

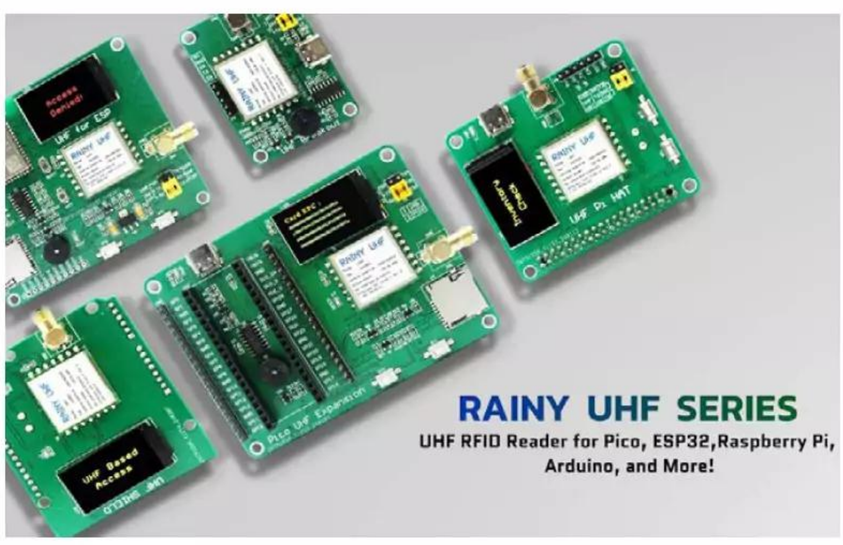
Raspberry Pi projects you can support this month

MD Robot Kit

A learning kit featuring many MD Robot kits, including the Compute Module 4-powered MiniPupper robots. "Go from zero to robot engineer," it claims. "The affordable Turtle Bot and the Mini Pupper 2G are open-source. It's easy to master robotics from robot motion control, ROS SLAM&Navigation, and generative AI applications. Our open-source robots provide everything you need to get started and succeed."

▶ **kck.st/3XuzYcd**

Rainy UHF Series



This range of RFID boards works with a wide range of microcomputers and microcontrollers, including HATs and other Raspberry Pi accessories. Apparently they can read up to 60 tags a second at a range of 3.2 metres that can be expanded with optional antennas. Very impressive.

▶ **kck.st/4dMtEm2**

Your Letters

Roadside telemetry

I am interested in using a Raspberry Pi to send odometer readings from my car to another Raspberry Pi unit by the roadside. Do you have any projects that are similar that I could consult?

Francis via email

We did a big feature about car computers way back in issue 81 of *The MagPi* (**magpi.cc/81**) – car technology of that kind is an old standard so it should be a good place to start in terms of getting all the car telemetry you could want. You could probably use some GPS add-ons for other location-based information as well.

As for transmitting to a roadside unit, projects with Bluetooth and Wi-Fi hotspots are a good start, however we'd be concerned with the amount of time that would be required to transmit data, so perhaps some RF transmission would be better there.

We've never had a project that does all of this in the magazine before though, so if you do complete it, let us know.

▼ This project from *MagPi* #81 is a cool way to get more information about your car and driving habits





▼ Students can still use the remote robotics lab set up by Wojciech Domski

Mother of invention

[In regards to RemoteLab from *MagPi* #145 (magpi.cc/145)] – I'm always amazed at what people do with Raspberry Pi hardware. But it required human initiative and dedication. He could have just blamed COVID and sat back. But because he is dedicated to his students and teaching, he took the initiative and found a solution to the problem. Great work.

Johannes via the blog

We've had a lot of projects in the magazine over the last few years that exist partly in thanks to people adapting to COVID. RemoteLab is a very impressive solution that, like a lot of technology inspired by that time, still has a long and useful life ahead of itself.

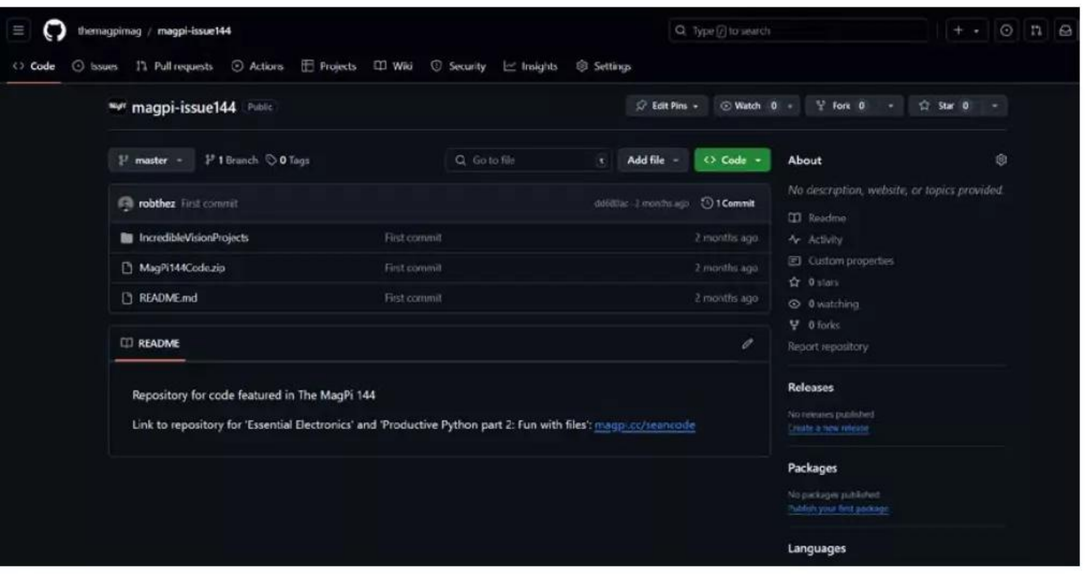
Software repositories

Can you please tell me where I can access the code for the projects in your publications. I have the *Official Raspberry Pi Handbook 2024* and subscribe to access the monthly PDFs, but need the code to build the featured projects.

John via email

Any code we feature in the magazines, and in the Official Handbooks, will have a link to go with the code printed on the page. We also provide a zip file with all the code for each issue on our GitHub which you can find at *magpi.cc/github*.

If you're having trouble finding a specific piece of code, don't hesitate to email us with the project and issue number you've seen it in – we can usually then hunt it down for you!



▲ All the code published in *The MagPi* can be found at links provided in the article – otherwise we put them all in a zip file on the repo for each issue

Contact us!

- Mastodon **magpi.cc/mastodon**
- Threads **@themagpimag**
- Facebook **magpi.cc/facebook**
- Email **magpi@raspberrypi.com**
- Online **forums.raspberrypi.com**



Community Events Calendar

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

01. Raspberry Pi Jam Kajiado

- **Saturday 28 September**
- **Merishaw School, Isinya, Kenya**
- **magpi.cc/rpjk146**

This Jam is not just for students and learners; it's a thrilling invitation for everyone passionate about technology. Whether you're a teacher, educator, parent, school, college or university student, academic, industry expert, technologist, volunteer, or even a curious banker, there's something for everyone at the Raspberry Pi Jam Kajiado.



02. Experience Raspberry Pi @ Purple Space

- **Sunday 29 September**
- **Purple Space, Manipal, India**
- **magpi.cc/ps146**

Raspberry Pi enthusiasts come together to share their passion for creating and designing projects with Raspberry Pi. This is an opportunity to empower young minds to learn computing using Raspberry Pi and become confident innovators anddigital makers. Get to see the newest addition to the Raspberry Pi product family, the Raspberry Pi AI Kit with the Hailo-8L co-processor, a 13 trillion operations per second (TOPS) neural processor that makes edge machine learning and computer vision much faster on a Raspberry Pi.

03. Cornwall Tech Jam

- **Saturday 12 October**
- **The Pavillion, Egloshayle, UK**
- **magpi.cc/ctj146**

The Cornwall Tech Jams are an opportunity for children to learn to write code outside of the formal school setting. Experimenting with ideas and seeing just where it takes them. All the equipment is provided and it's all free to attend. As usual there will be a whole host of Raspberry Pi equipment to use and see and loads of other technology to interact and build with. We dare say a rover or two will be there as well.



04. Southend Raspberry Jam Maker Meetup

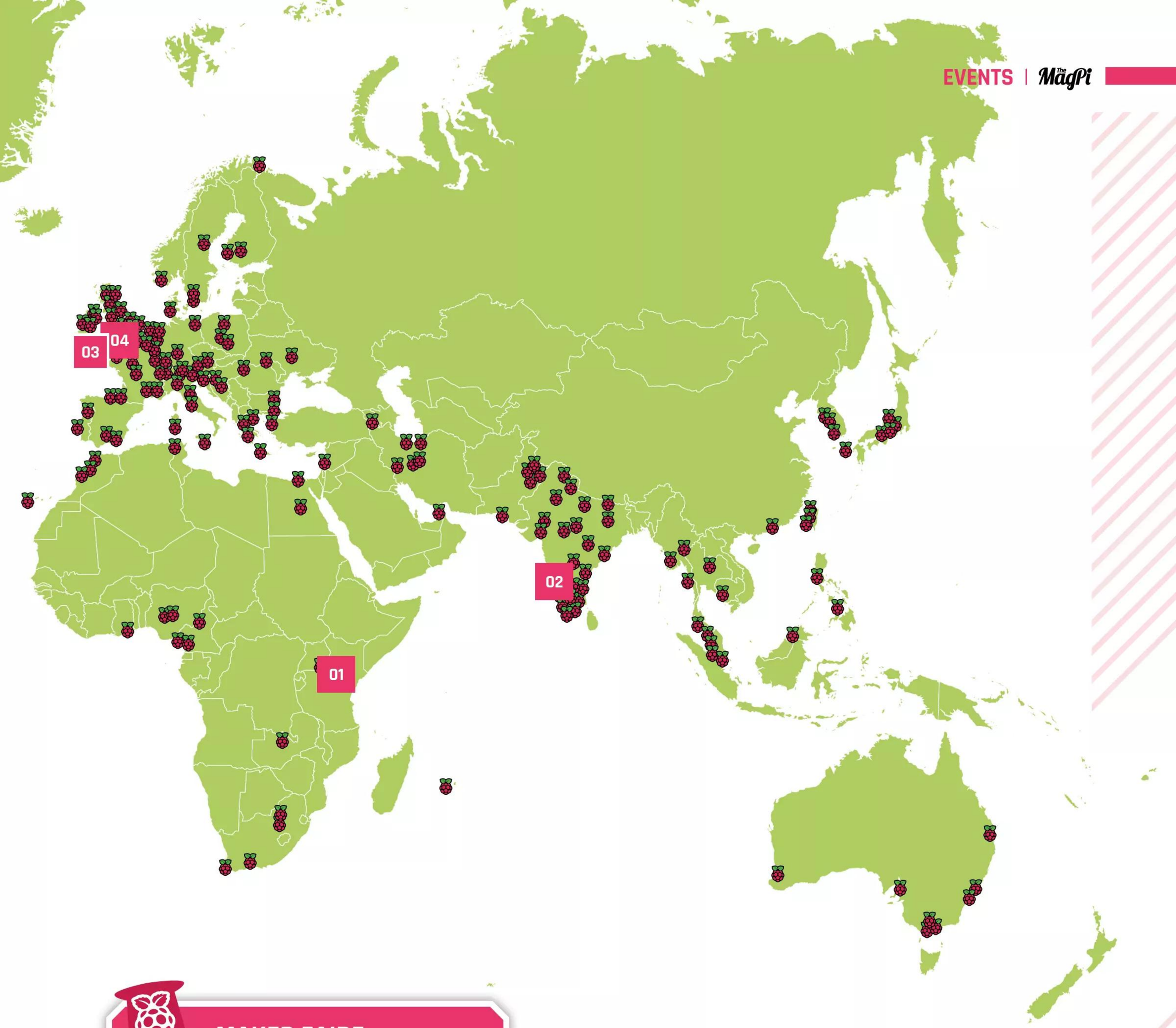
- **Thursday 17 October**
- **The Board Game Hut, Southend on Sea, UK**
- **magpi.cc/srjmm146**

Southend Raspberry Jam Maker Meetup is a monthly meetup for those who are interested in building projects using Raspberry Pi hardware and want to join a friendly group of enthusiasts and makers. We welcome beginners to professionals. If you have any ideas for projects, talks or demos, we're especially keen to hear from you.

FULL CALENDAR

Get a full list of upcoming community events here:

magpi.cc/events





Maker Faire Bay Area

- Where Mare Island Naval Shipyard, Mare Island, CA, USA
- When Friday 18 October to Sunday 20 October

"The Raspberry Pi team is eager to be back at Maker Faire Bay Area once again. Come experience the Raspberry Pi AI Kit and tell us about what you're making with Raspberry Pi technology."

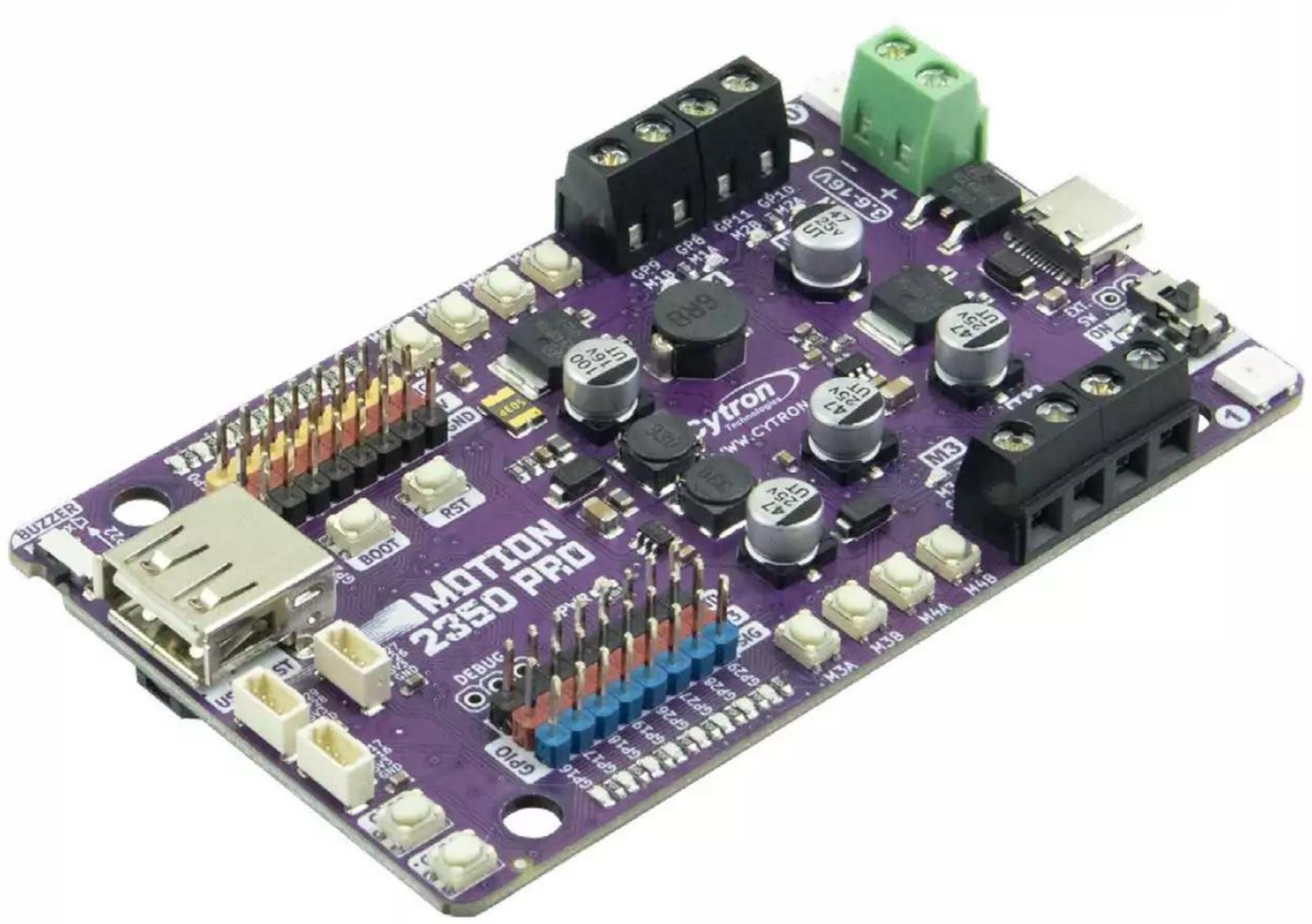
magpi.cc/mfba24

The MagPi | COMPETITION

WIN 1 OF 5 MOTION 2350 PRO

SPONSORED BY Cytron Technologies

One of the new products to use RP2350, this advanced robotics controller has a four-channel DC motor driver, LEDs, buttons, and sensor inputs – perfect for your next robot competition!

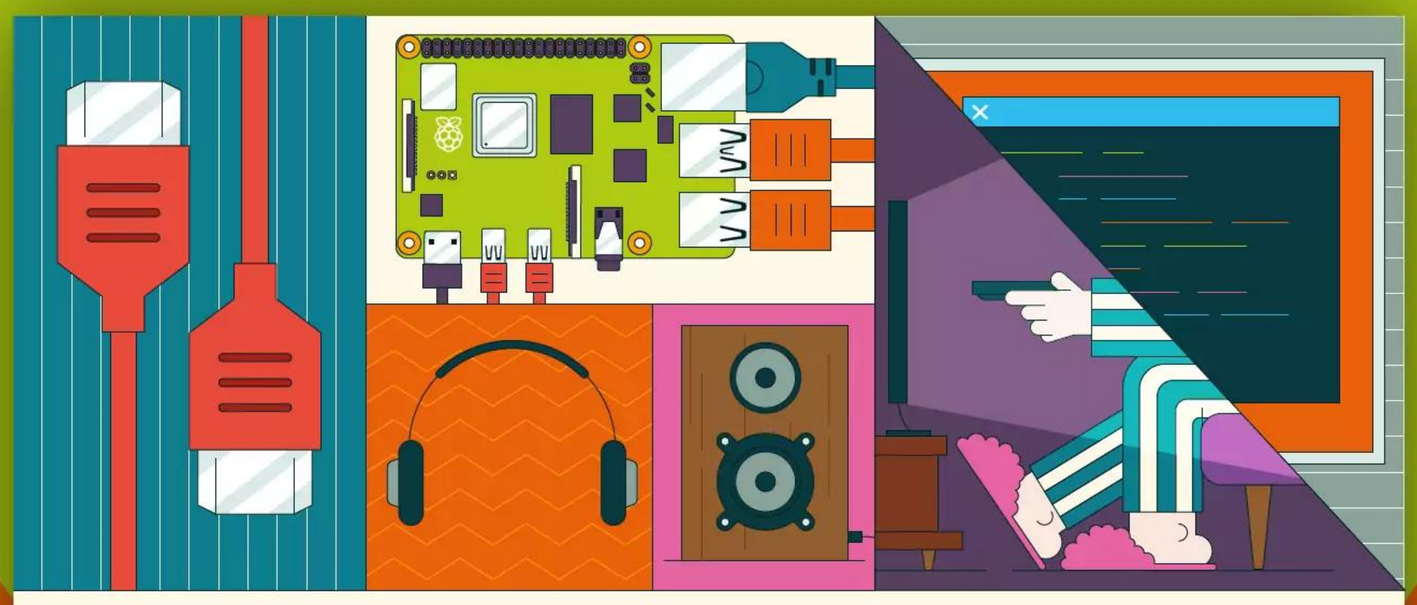


Head here to enter: magpi.cc/win | Learn more: magpi.cc/motion2350

Terms & Conditions

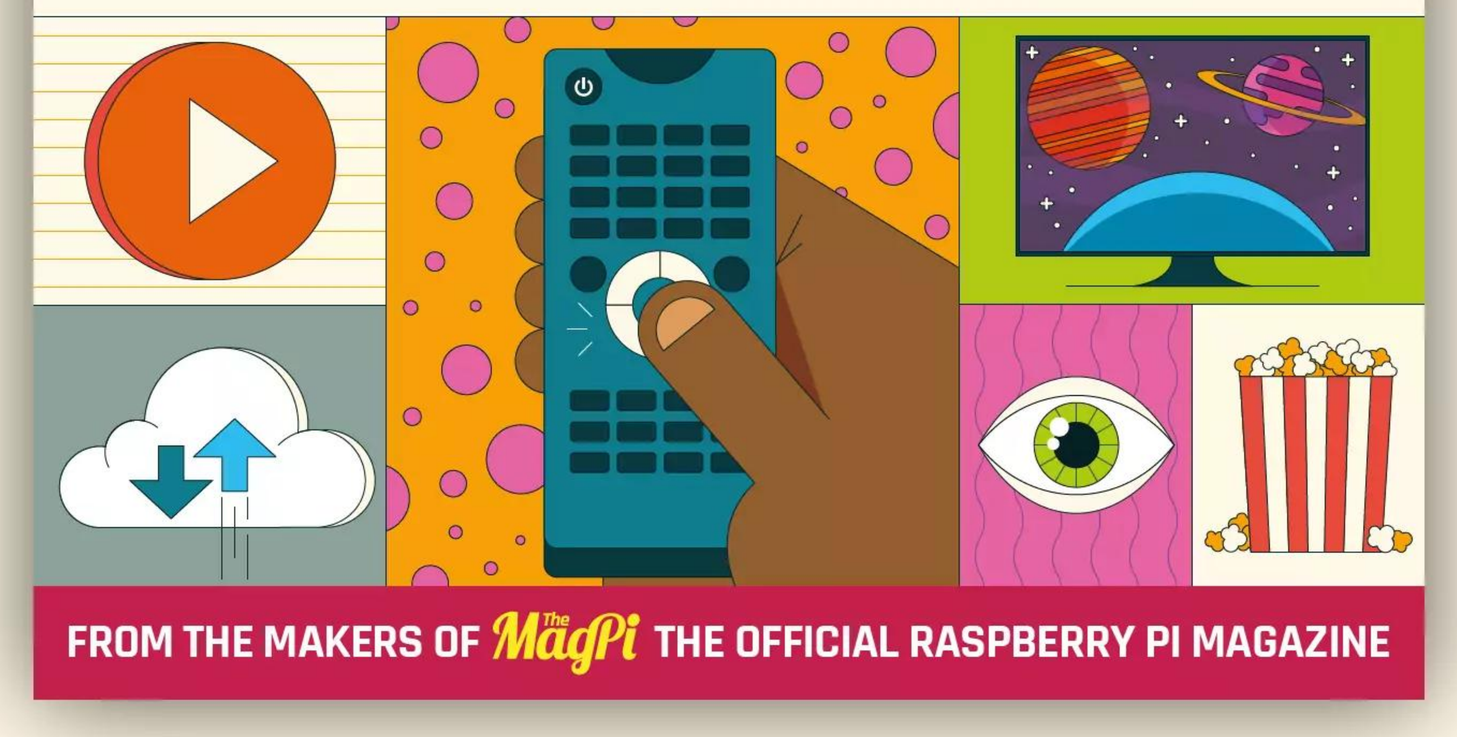
Competition opens on **25 September 2024** and closes on **24 October 2024**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from *The MagPi* magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter (X) or any other companies used to promote the service.

Your FREE guide to making a smart TV



BUILD A RASPBERRY PI MEDIA PLAYER

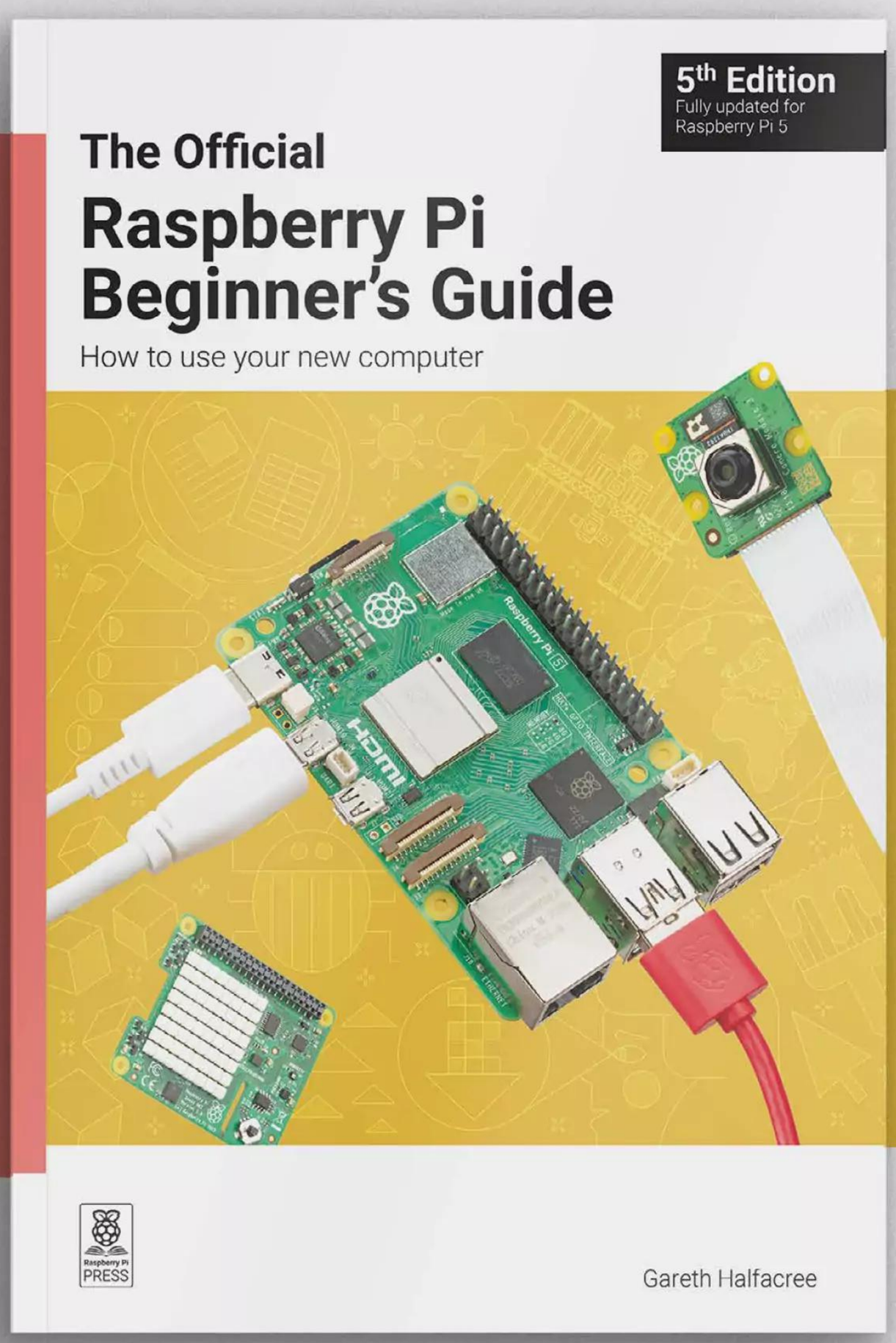
Power up your TV and music system



FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE

magpi.cc/mediaplayer

- Learn coding
- Discover how computers work
- Build amazing things!

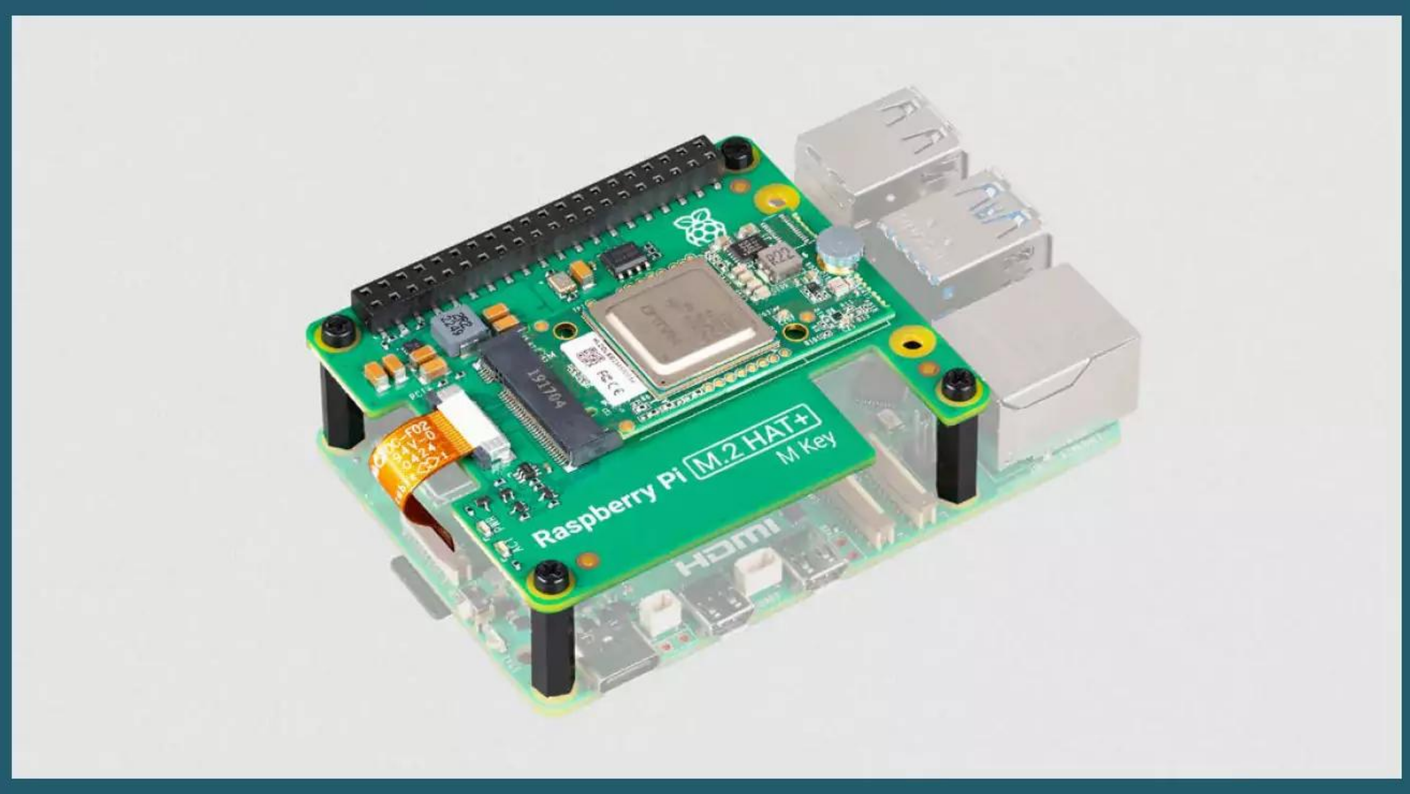


magpi.cc/beginnersguide

Raspberry Pi 5

AI PROJECTS

Super-smart builds making the most of **Raspberry Pi 5** & **AI Kit**



The Magpi **#147**
On sale **24 October**

Plus!

Build a Jellyfin home music streaming service

Make your own handheld console

Discover the DEC FlipChip Tester

DON'T MISS OUT! **magpi.cc/subscribe**

MASTODON	magpi.cc/mastodon
THREADS	@themagpimag
FACEBOOK	magpi.cc/facebook
EMAIL	magpi@raspberrypi.com
ONLINE	forums.raspberrypi.com

EDITORIAL

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Ian Evenden

ADVERTISING

Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN

Head of Design
Jack Willis

Designers
Sara Parodi, Natalie Turner

Illustrator
Sam Alder

Photographer
Brian O Halloran

CONTRIBUTORS

Chris Boross, Nate Contino, David Crookes, PJ Evans, Jo Hinchcliffe, Rosemary Hattersley, Phil King, Nicola King, Andrew Lewis, David Miles, Sandeep Mistry, Paul Oberosler, K.G. Orphanides, Eben Upton

PUBLISHING

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications
Helen Lynn

CEO
Eben Upton

DISTRIBUTION

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

SUBSCRIPTIONS

Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

recycle

CC BY NC SA

This magazine 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.

The MagPi magazine is published by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB. 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).

Raspberry Pi PRESS

ISSN: 2051-9982.



The more things **change...**

There's comfort in things staying the same, says **Andrew Gregory**

This month Raspberry Pi released its latest model. And what was the new feature that sets this model apart from all the rest? More I/O possibilities? Greater security? A faster processor? The ability to handle AI, correct your spelling as you type or remind you when it's time to take the bins out? No: the big new feature of the new Raspberry Pi 5 is that it has less RAM. The latest version of the best-selling single board computer is now available with just 2GB of RAM, as well as variants with 4GB and 8GB. It's also priced at around £30 less than the 8GB version.

It's refreshing that newer doesn't always have to mean bigger and better. I've been using a Raspberry Pi 4 for the last few years as a desktop Linux machine, and although it could be faster, then thing that holds me back from upgrading it is nothing to do with the Pi itself. It's the peripherals: if I moved to a Raspberry Pi 5 I'd have to get new headphones, as the Raspberry Pi 5 moved away from a 3.5mm headphone jack. I'd have to spend money on headphones that use Bluetooth, and forever be losing them or running out of charge. I realise this makes me unusual in the world of the tech enthusiast, but once I get something that works, I just don't want to go to the effort of changing it.

> "Newer doesn't always have to mean bigger and better"

That may make me an outlier in terms of individuals, but there's another consumer sector that really wants to be able to get the bare minimum and stick with it: businesses. Upgrading is a cost. Buying new cables because the new version of a device uses USB-C instead of Micro USB is a cost. Any change at all imposes a cost, and if you can avoid that, you're winning.

We all know that things like connectivity and processing speed are features. But price is also a feature, and a really big one. So too is backward compatibility. If I don't want to buy a new thing because it messes with my minimal setup of screen, keyboard, mouse and headphones, then it's vanishingly unlikely that I'd invest in a new device if it meant I would have to build a whole new factory to accommodate it.

It's great there are new devices to play with. But it's even greater knowing that you can buy one, and build it into your project, or your product, or even your manufacturing setup, in confidence that it won't be rendered obsolete by the next new model. It sounds counter-intuitive, but that's how you keep customers coming back: not by locking them in, but by making it clear that they don't have to upgrade if they don't want to.

AUTHOR

Andrew Gregory

Andrew will one day do something useful with a Raspberry Pi Pico and the batch of Ukrainian-made VFD tubes that have been rolling around his desk for the last five years.

magpi.cc

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team



- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiShop.us PiShop.ca PiShop.mx ThePiHut Pi-Shop.ch Welectron.

Contact your favorite Pi store if it's not listed here

PiKVM

Remote control **redefined**

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

- Power consumption in idle mode: just 2.67 Watts!
- Transfer your mouse and keyboard actions
- Access to all configuration settings like UEFI/BIOS
- Capture video signal up to 1920x1200@60 Hz
- Take full control of a remote PC's power

PiKVM V4 Plus

The most feature-rich edition

- More connectivity
- Extra storage via internal USB3.0
- Upgraded powering options
- More physical security features
- Extra HDMI output
- Advanced cooling solution



A cost-effective solution for data-centers, IT departments or remote machines!

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBii · Cytron Marketplace

ROBU.IN Your ideas, Our Parts · NETTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

No reseller in your country?
Check shop.hipi.io (import fees might apply).

List of official resellers by country: