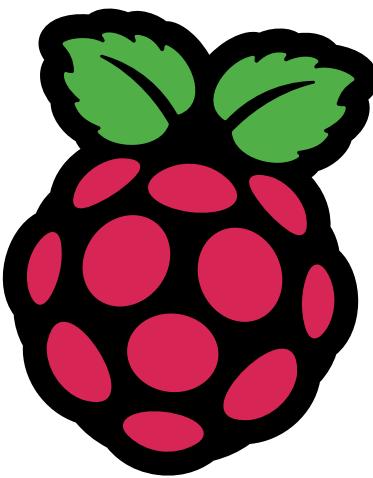




YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi



Issue 138 | February 2024 | magpi.cc

The official Raspberry Pi magazine

NEXT GEN RETRO

ON Raspberry Pi 5

BACK TO
2000!

FEATURES
LEGAL
ROMS!



ARTS & CRAFTS
PROJECTS

NEW! HAT +
HARDWARE

RASPBERRY PI
AT MAKER FAIRE

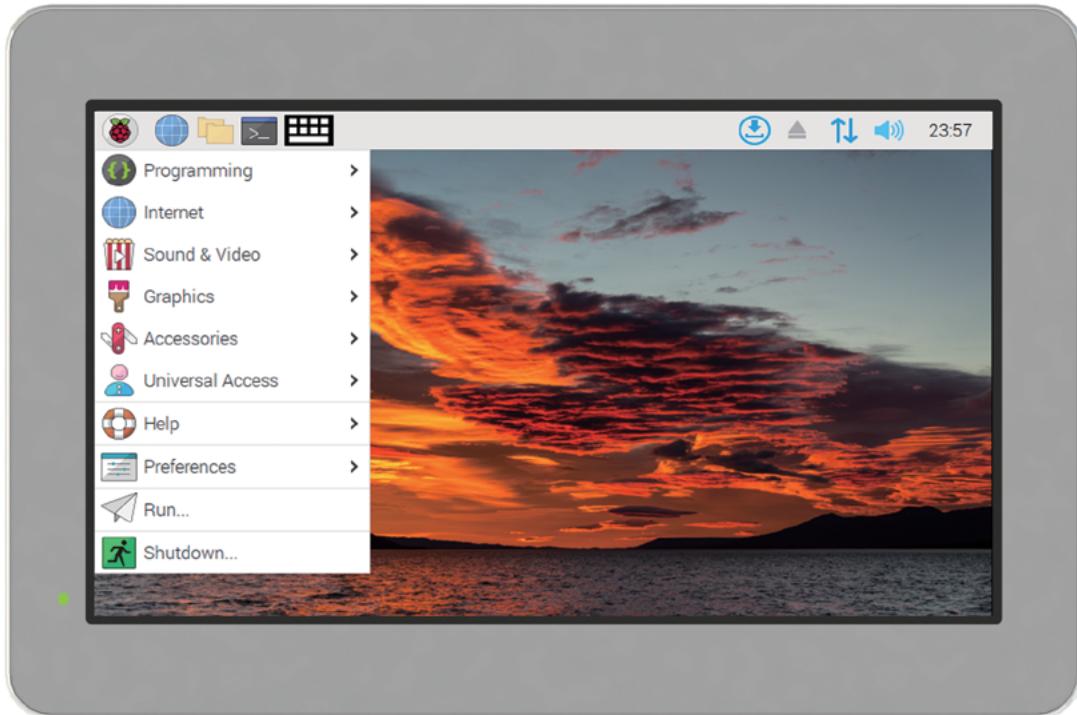
£5.99



LEARN HOW SOUND SYNTHESIZERS WORK



Industrial Raspberry Pi



ComfilePi

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.



WELCOME to The MagPi 138

One glance around Raspberry Pi Towers and you'll find no shortage of retro computing enthusiasts.

While Raspberry Pi is an ultra-modern and incredibly powerful computer, it is adored by people originally enthused by the home computing boom of the 1980s and '90s.

So it's no surprise that Raspberry Pi is used to emulate classic computers and retro gaming consoles. For many Raspberry Pi owners this is their very first project: to recreate one of the esteemed computers of the past, built by the likes of Commodore, Atari, and Acorn. Personally, I'm a Commodore fan: first the C64 and then a beloved Amiga.

The massive speed boost offered by Raspberry Pi 5 is a game-changer for retro gaming: opening up a whole new slew of 5th and even 6th generation consoles (such as the Dreamcast, GameCube and Wii). There's a vibrant homebrew development scene for all of them, packed with brand-new games for you to discover.

Our Next Gen Retro on Raspberry Pi 5 feature (page 30) is written by KG Orphanides who is an absolute expert on all matters retro. It's a fantastic feature!

Lucy Hattersley Editor



**Lucy
Hattersley**

EDITOR
Lucy is editor of The MagPi magazine and spent so much time in *Baldur's Gate 3* over Christmas that she has started mentally rolling a D20 before making everyday decisions.

magpi.cc

GET A
**RASPBERRY PI
PICO W**
WITH A SUBSCRIPTION!
PAGE 24





Innovation demands selection

Let us help you engineer the future with millions of products from over 2,300 name-brand, industry-leading suppliers.

Find what you need at digikey.co.uk or call 0800 587 0991.

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. © 2023 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

 **ECIA MEMBER**
Supporting The Authorized Channel

Contents

► Issue 138 ► February 2024

Cover Feature

- 30** Next Gen Retro Gaming

Regulars

- 08** World of Raspberry Pi
- 26** Case Study: Green Custard
- 92** Your Letters
- 94** Community events calendar
- 97** Next Month
- 98** The Final Word

Project Showcases

- 12** MacQuadra PiSCSI
- 16** POV candle
- 18** OneInchEye
- 22** CarbonaraGPT



POV candle



30



OneInchEye

18

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

Tutorials

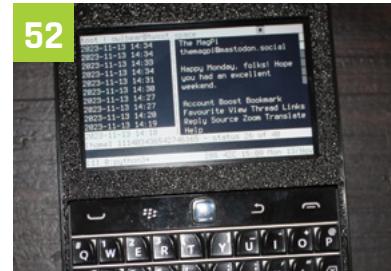
- 40** Using an RTC on Raspberry Pi 5
- 44** Create your own PCB
- 52** Play games on a Beepy palmtop
- 58** Make an old phone voice assistant

The Big Feature



Arts & Crafts with Raspberry Pi

66



Play games on a Beepy palmtop

58



Make an old phone voice assistant

Reviews

- 72** Bullfrog
- 74** Argon NEO 5
- 76** 8BitDo Retro Mechanical Keyboard
- 78** Revolution Pi Connect 4
- 80** 10 amazing automation projects
- 82** Learn algorithms

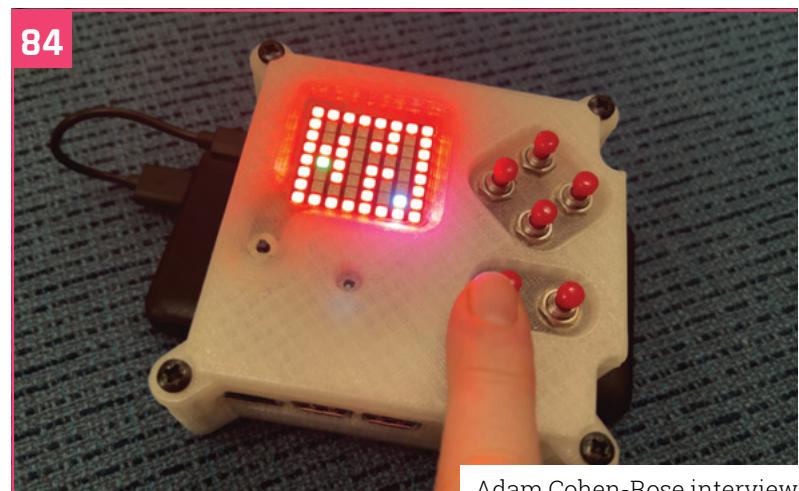


72

Bullfrog

Community

- 84** Adam Cohen-Rose interview
- 86** Maker Faire Shenzhen report
- 88** This Month in Raspberry Pi



Adam Cohen-Rose interview

84

WIN 1 OF 3

ELECROW CROWVISION

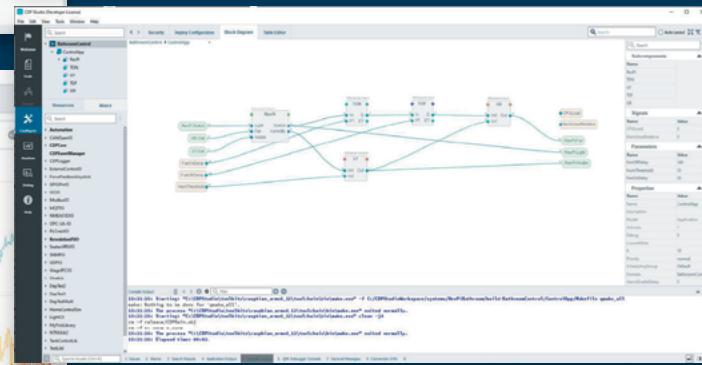
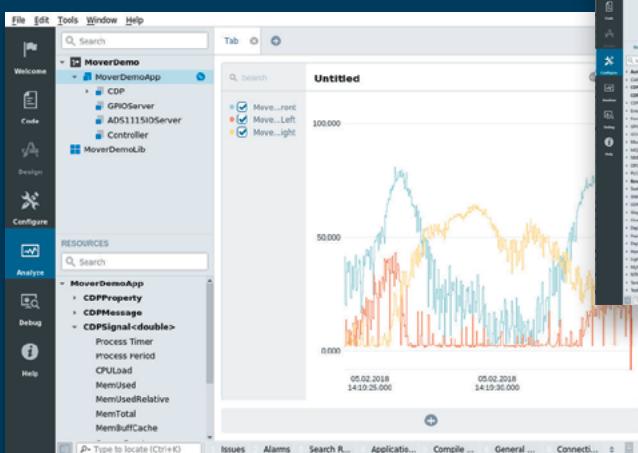
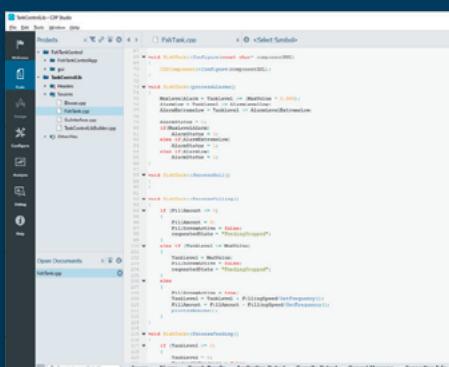


96

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.

Design

Code



Configure

Analyze

PROFESSIONAL CONTROL SYSTEM DEVELOPMENT TOOL

Home projects made easy.

CDP Studio, a great software development tool for your home projects. Build systems for Raspberry Pi, use C++ or NoCode programming, open source libraries, out of the box support for GPIO, I2C, MQTT, OPC UA and more. Create beautiful user interfaces. Built for industrial control system development, **FREE for home projects.**

cdpstudio.com

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

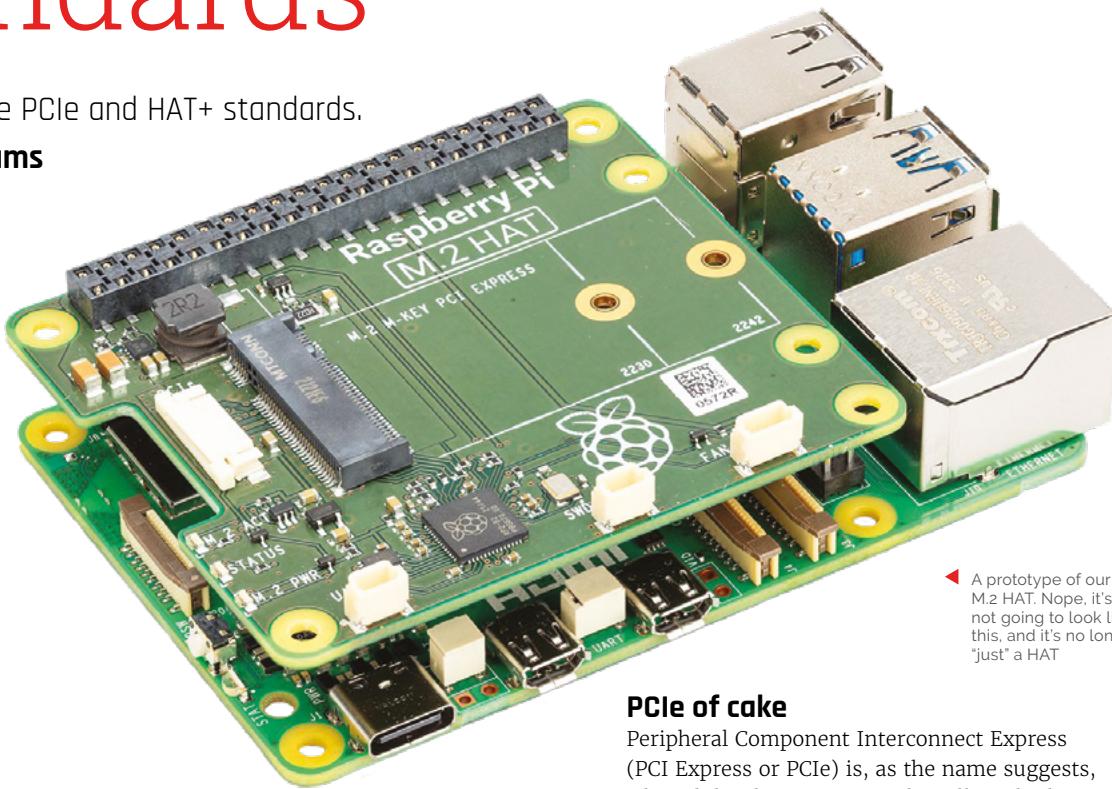
CDP Technologies AS // Hundsværgata 8, 6008 Ålesund, Norway



Double standards

Introducing the PCIe and HAT+ standards.

By **James Adams**



A prototype of our M.2 HAT. Nope, it's not going to look like this, and it's no longer "just" a HAT

It's now been a little over three months since we announced Raspberry Pi 5 (Issue #134, magpi.cc/134), and the time has flown by.

We knew we'd built something quite special, but we've been overwhelmed by the positive response from the community.

The production ramp has been steeper than for any previous flagship product: we've been producing 70,000 units a week for the last few weeks, and this rate is set to increase to 90,000 units a week by the end of January.

Once people had recovered from the shock of seeing both a power button and a real-time clock (magpi.cc/rtdocs) on a Raspberry Pi, one of the most commented-on features of the new platform was the small, vertical, 16-way FFC (Flat Flexible Cable) connector on the left-hand side of the board, which exposes a single-lane PCI Express interface.

PCIe of cake

Peripheral Component Interconnect Express (PCI Express or PCIe) is, as the name suggests, a board-level interconnect that allows high-speed data transfer between a processor chip (in our case BCM2712, magpi.cc/bcm2712) and external peripherals such as NVMe SSDs, Ethernet cards, or more exotic things such as AI/ML accelerators.

PCIe works by serialising data transfers and sending one bit at a time down a single channel. Higher-capacity PCIe interfaces have more lanes ($\times 2$, $\times 4$, $\times 8$, $\times 16$); on Raspberry Pi 5, BCM2712 is connected to our RP1 I/O controller via a $\times 4$ interface. Each lane runs at 5Gbits/s for PCIe 2.0 (the fastest mode that we officially support on Raspberry Pi 5); after coding overhead, this translates into a capacity of 4Gbytes/s. Even taking into account other protocol overheads, you're likely to see ~450MBytes/sec to and from a good NVMe SSD. Pretty fast!

Alongside the data and clock channels, the PCIe specification requires some sideband

signals like reset, clock request (which does double duty as a power state signal), and wakeup. Our 16-way connector provides all these signals. We also have two pins that allow us to control board power, and to ensure that an appropriately designed PCIe peripheral is automatically detected by the Raspberry Pi firmware.

Not an M.2

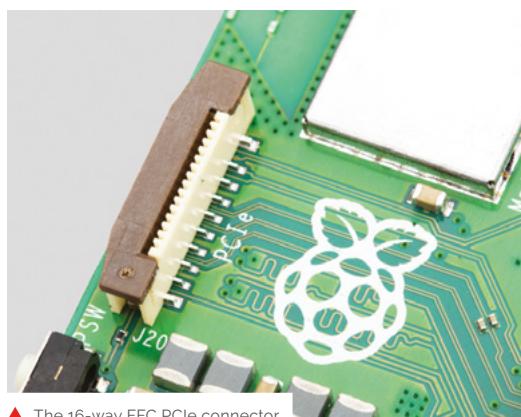
Why didn't we add an M.2 connector to the Raspberry Pi 5? The M.2 connector is large, relatively expensive, and would require us to provide a 3.3V, 3A power supply. Together, these preclude us offering it in the standard Raspberry Pi form factor.

Using a small, low-cost FFC connector allowed us to provide a PCIe interface without growing the board, or imposing the cost of an M.2 connector and its supporting power-supply circuitry on every Raspberry Pi user.

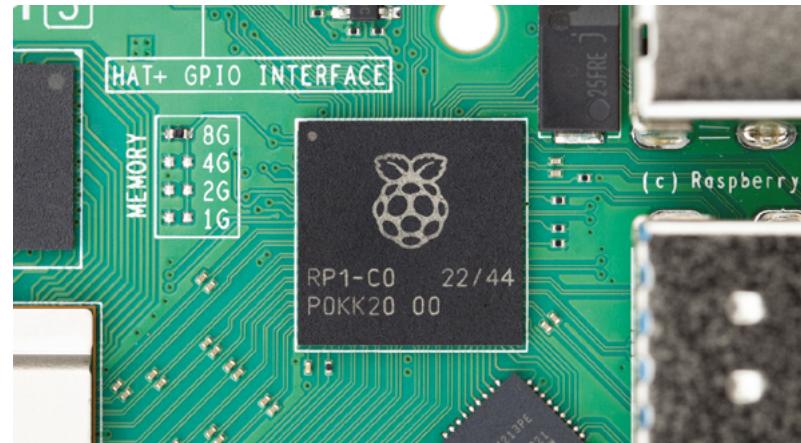
Specification the first

One thing we did not have ready at the time of the Raspberry Pi 5 launch was a specification for how to build peripherals that attach to the 16-way PCIe connector. The interaction of PCIe peripherals with Raspberry Pi power states and firmware required detailed consideration, and we wanted to make sure we had done extensive testing of our own prototype product to make sure everything was working exactly as expected.

Today, we're releasing the first revision of that specification: Raspberry Pi Connector for PCIe A 16-way PCIe FFC Connector Specification (magpi.cc/pciestandard). Our own M.2 M Key HAT+ is in the final stage of prototyping, and will be launched early next year.



▲ The 16-way FFC PCIe connector



▲ HAT+ on the Raspberry Pi 5 silkscreen, sort of gave the game away?

Specification the second

For those of you reading closely, you'll have noticed that we're calling our M.2 HAT a "HAT+". If one new specification wasn't enough for you, today we're also releasing a preliminary version of the new Raspberry Pi HAT+ Specification (magpi.cc/hatplusspec).

The original HAT specification was written back in 2014, so it is now very overdue for an update. A lot has changed since then. The new specification simplifies certain things, including the required EEPROM contents, and pulls everything into one document in the new Raspberry Pi documentation style, along with adding a few new features.

There's still work to be done on this standard, and our EEPROM utilities

M Watch this space for the new M.2 HAT+, and a final version of the HAT+ standard **M**

(magpi.cc/eepromutilsgit) haven't yet been updated to support the generation of the new style of EEPROMs. So this release is very much for people that want to get a feel for how the HAT standard is changing.

We really wanted to get the HAT+ standard right, as it's likely to be around for as long as the old HAT standard. One of the reasons for the delay in getting the PCIe connector standard published was our sense that PCIe boards that go on top, rather than boards that go beneath, should probably be HAT+ boards. Ours is going to be!

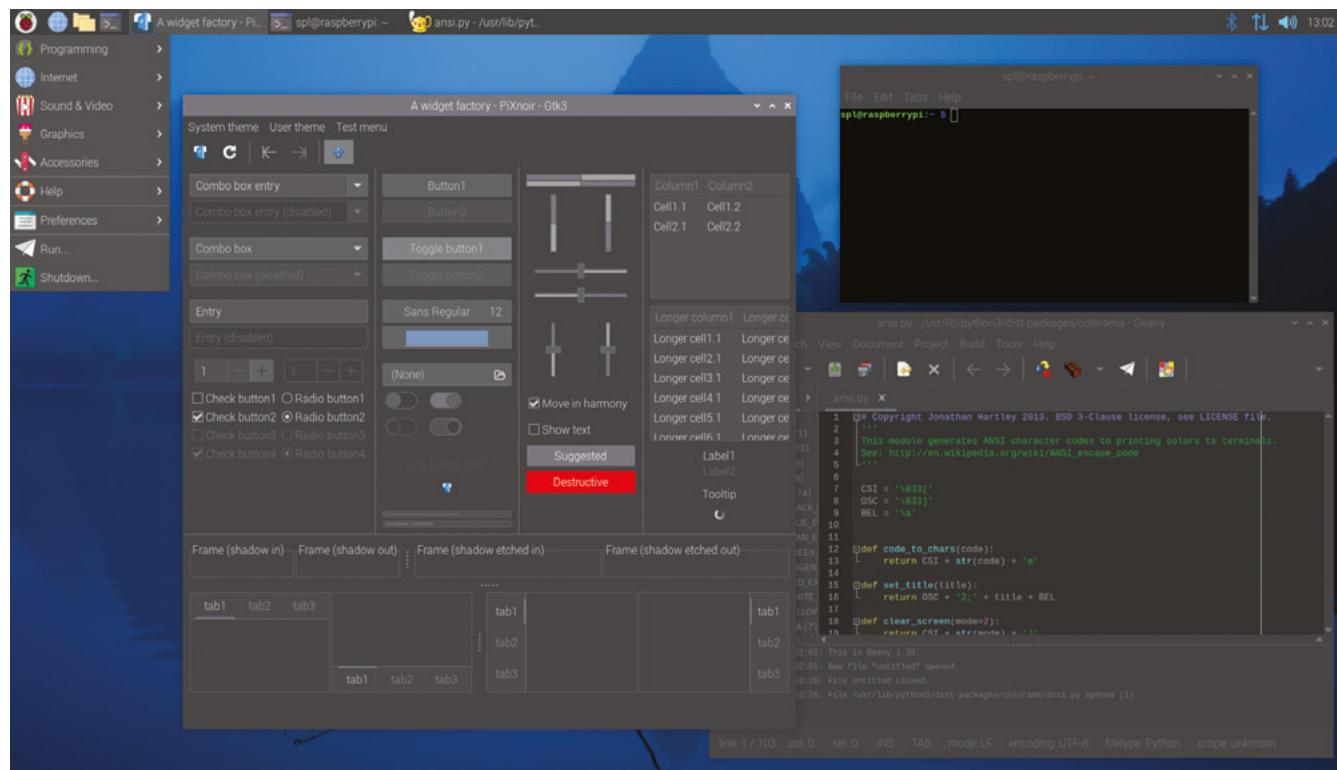
Standards for all!

If you want to discuss them with the community, head over to the Raspberry Pi forums (raspberrypi.com/forums), where you'll find a dedicated area to talk about HATs, HAT+ and other peripherals.

Watch this space for the new M.2 HAT+, and a final version of the HAT+ standard, which we'll release alongside it in 2024 **M**

Raspberry Pi OS goes dark

You can now use Raspberry Pi OS in dark mode. By **Simon Long**



▲ Raspberry Pi OS in dark mode

A little under two months have passed since the release of the Wayland-based Raspberry Pi OS Bookworm (magpi.cc software). Whenever we do a major version release like this, we invariably spend the next few weeks fixing all the bugs that real users have found but our pre-release testing didn't, and then make a bug-fix release with them (hopefully) fixed.

This has taken a few weeks longer this time, simply due to the sheer amount that changed under the hood in Bookworm, but the bug-fix release is now ready and can be installed from today via Raspberry Pi Imager, or downloaded from the usual place on our

website (magpi.cc/osimages). Or, to upgrade an existing image, simply use the updater icon on the taskbar, or (if you're more old-school) open a terminal and type

```
$ sudo apt update
```

followed by

```
$ sudo apt full-upgrade
```

This update includes improved support for encrypted connections in WayVNC; the latest version of Thonny; Mathematica and Scratch 3

working on Raspberry Pi 5; and a bunch of other small bug fixes and tweaks. But we thought we'd give you a little bonus in this release too...

A few weeks ago, Eben wandered past my desk, and remarked, "wouldn't it be nice if we had a dark theme?" (He's not the first person to suggest this, but he is the boss, so I tend to pay more attention when he suggests things!) And as it happened, I wasn't particularly busy that day.

Our PiXflat theme has been around for a few years now – we launched it along with Raspberry Pi OS Buster back in 2019. It started out as a mildly tweaked version of the default GTK theme called Adwaita, but it changed quite a bit over the years, so now doesn't have all that much in common with Adwaita any more.

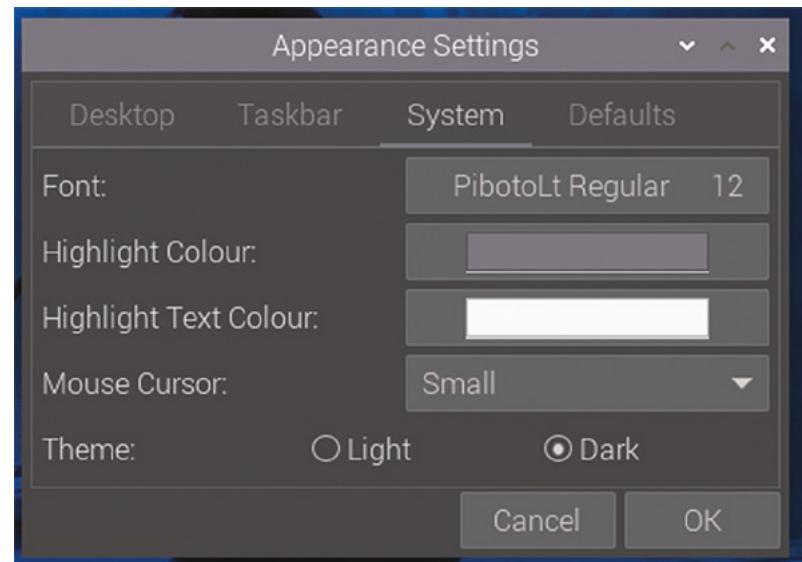
A theme is basically a big CSS (cascading style sheet) file as used for styling web pages, e-books and the like, which defines the appearance of every widget used to draw applications. (A widget is a user interface element such as a button or a text label.) You can set the colour, the font, the background and numerous other aspects for each widget, and you can set different values depending on whether the widget is active (e.g. a button which is being pressed), whether it is disabled, whether the mouse pointer is over it, and so on. And when I say it is a "big" file – it's really big: PiXflat is around 4500 lines.

“Creating a dark theme can be relatively easy, or really hard”

Dark mode

Creating a dark theme can be relatively easy, or really hard, depending on how the colours have been defined in your original theme. If all your colours are defined as variables, it is relatively easy – but if all your colours are hard-coded values then it is rather more time-consuming. In PiXflat, as in Adwaita before it, the colours were all hard-coded! So the first job was to go through 4500 lines of CSS and find all the hard-coded colours, replacing them with variables, and then setting those variables to the original hard-coded values so I didn't break the original light theme.

Then, having done that, it was time to consider how each one of those colours should change to produce a suitable dark appearance. The obvious



first step was simply to invert each colour, so that black became white and so on. The trouble with this is that while what you end up with is indeed dark, it doesn't usually look that good.

So then the hard work began – choosing exactly which contrasting colour was going to be used in the dark theme to replace the light version. This involved setting the values of around 70 colour variables to create a set that worked together; you have to take into consideration having sufficient contrast between elements so that you can still see everything clearly, so that text is still readable against darker backgrounds, so that the colour change when you move the mouse over an element is still obvious, and so on. And it all interacts, so you find that if you tweak one colour, you then need to tweak four or five others to keep all the contrast correct.

Endless themes

From past experience with PiXflat, creating a theme is very much like the proverbial painting of the Forth Bridge; it's never really finished, because you keep making little tweaks. But the theme which I have called PiXnoir – the dark version of PiXflat – is now ready to use; like PiXflat it will doubtless continue to evolve over time, but the first version is included in this release.

We've made it easy to switch between the themes. Just open 'Appearance Settings' from the 'Preferences' section of the main menu, go to the 'System' tab, and switch the 'Theme' option from 'Light' to 'Dark'. Most applications will load the new theme on the fly, but some applications – particularly Geany and Calculator – use their own internal themes as well, so you'll need to close those first if they are running in order to get the theme to change. ■

▲ The Appearance Settings menu in Raspberry Pi OS

MacQuadra PiSCSI

Reviving a beloved Apple Mac with a view to retro gaming led one computing fan to embrace Raspberry Pi-based emulation, learns **Rosie Hattersley**



Brian Jepson

MAKER

Brian is The MagPi's publishing director and a volunteer and restorer at Rhode Island Computer Museum.

jepStone.net

Brian Jepson developed an interest in home computing and programming not long after receiving a Radio Shack 150-In-1 electronics kit from his parents. He went on to write about computing and making, culminating in his role as Publishing Director of *The MagPi* and *Hackspace* at Raspberry Pi. "When Raspberry Pi originally came out, it really tied both interests together in ways that no other development board had.", he says. In his spare time, Brian volunteers at Rhode Island's Computer Museum (ricomputermuseum.org), restoring old computers. He had a couple of SCSI-based (Small Computer System Interface) computers – "a PowerPC ThinkPad and an Apple Macintosh

Quadra, all of which used SCSI drives" – and was keen to bring them back to life, complete with their peripherals.

Mac revival

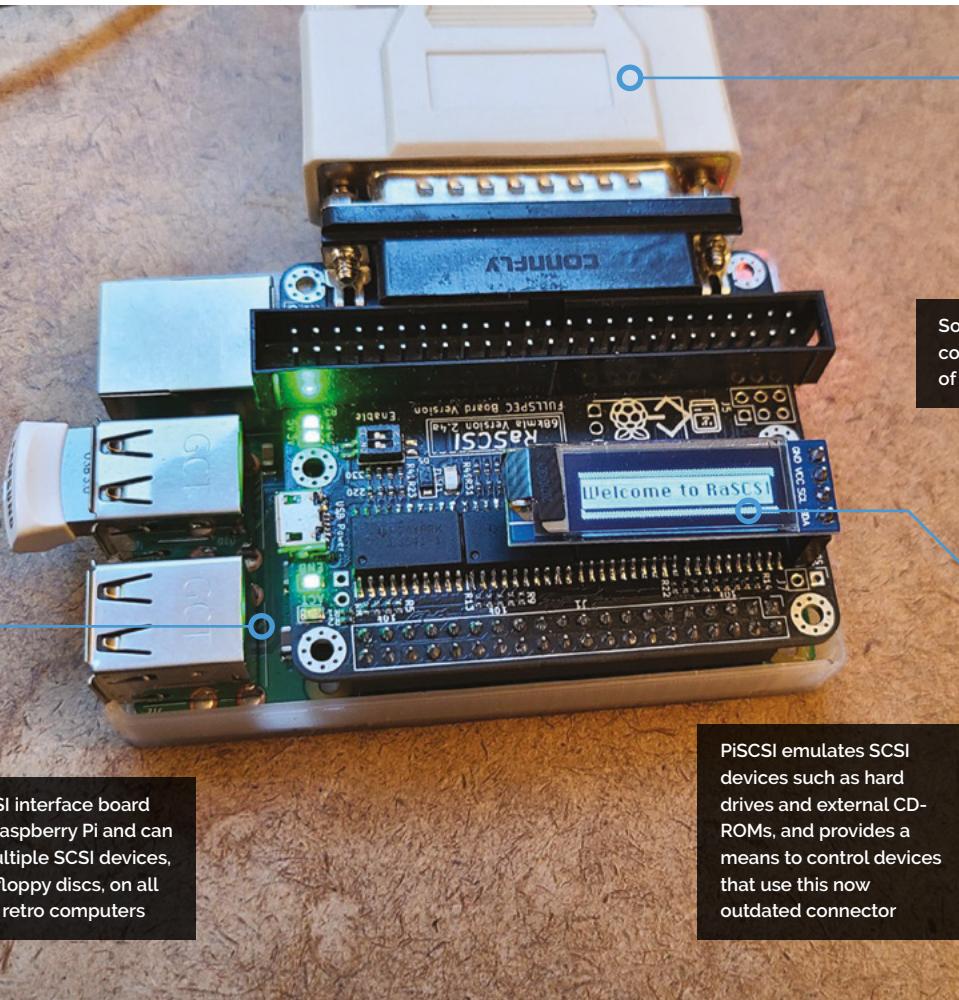
Getting an old Mac running can be a bit of a project, says Brian. You will almost certainly need to replace the Parameter RAM (PRAM) battery and you may find a leaky battery or capacitors, which can damage the motherboard requiring cleanup and repair. But it's worth the effort. Macs are great for gaming, he says.

To revive his old Mac Quadra, Brian needed something that could present itself as a variety of SCSI devices. He was drawn to RaSCSI, "a fork of an earlier project" that has since been renamed to PiSCSI: magpi.cc/piscsi. The emulator was developed by GitHub user GIMONS (originally for his Sharp X68000 controller) and allows Raspberry Pi to function as emulated SCSI devices (hard disk, CD-ROM, and others) for vintage SCSI-based computers and devices.

PiSCSI's success on dozens of retro platforms won Brian over. "Raspberry Pi is going to run faster, more quietly, and be much more reliable than some old SCSI drive. You also have the full power of the hardware and operating system to do other cool things, such as run a vintage web proxy like macproxy (magpi.cc/macproxy). I've used



► The Raspberry PiSCSI emulator brings old floppy disk and hard disk drives back into use



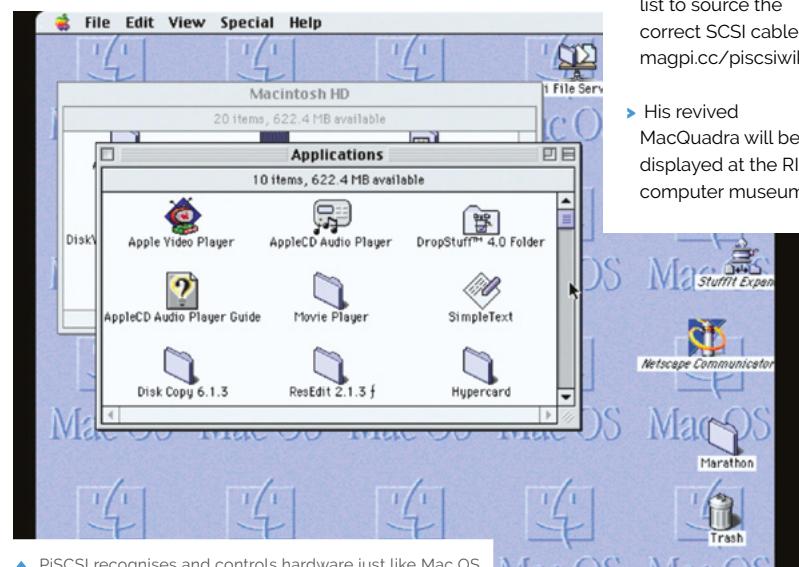
The RaSCSI interface board sits atop Raspberry Pi and can control multiple SCSI devices, including floppy discs, on all manner of retro computers

PiSCSI emulates SCSI devices such as hard drives and external CD-ROMs, and provides a means to control devices that use this now outdated connector

Sourcing a suitable SCSI connector and cable is part of the revivalist challenge

Quick FACTS

- ▶ Brian ignored the bespoke install options...
- ▶ ... And used a "vanilla" Raspberry Pi OS installation: magpi.cc/piscsisetup
- ▶ He recommends using PiSCSI with the most powerful Raspberry Pi you have...
- ▶ ... And using this list to source the correct SCSI cable: magpi.cc/piscsiwiki
- ▶ His revived MacQuadra will be displayed at the RI computer museum



▲ PiSCSI recognises and controls hardware just like Mac OS

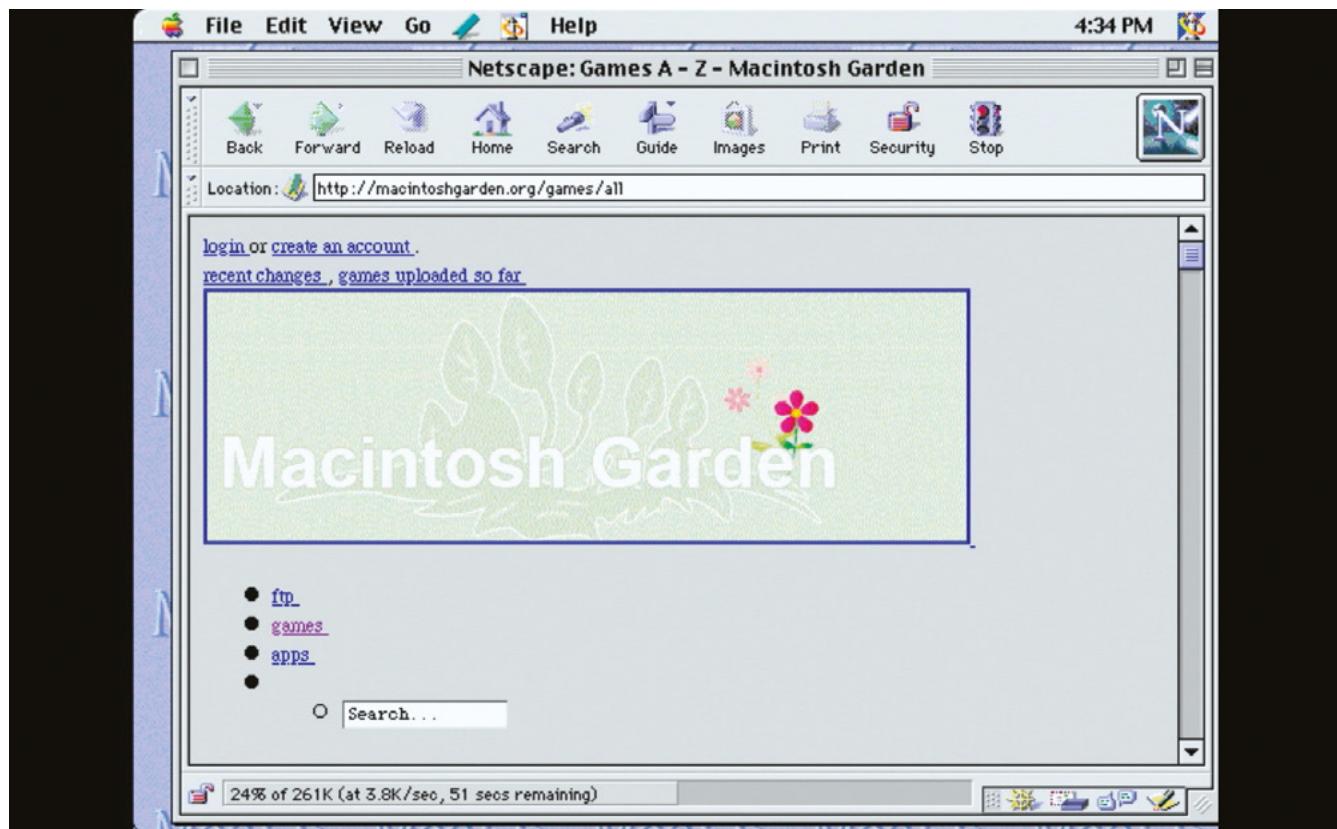
Raspberry Pi 4 and also Raspberry Pi Zero 2 in a RaSCSI configuration."

Clever old computer kit

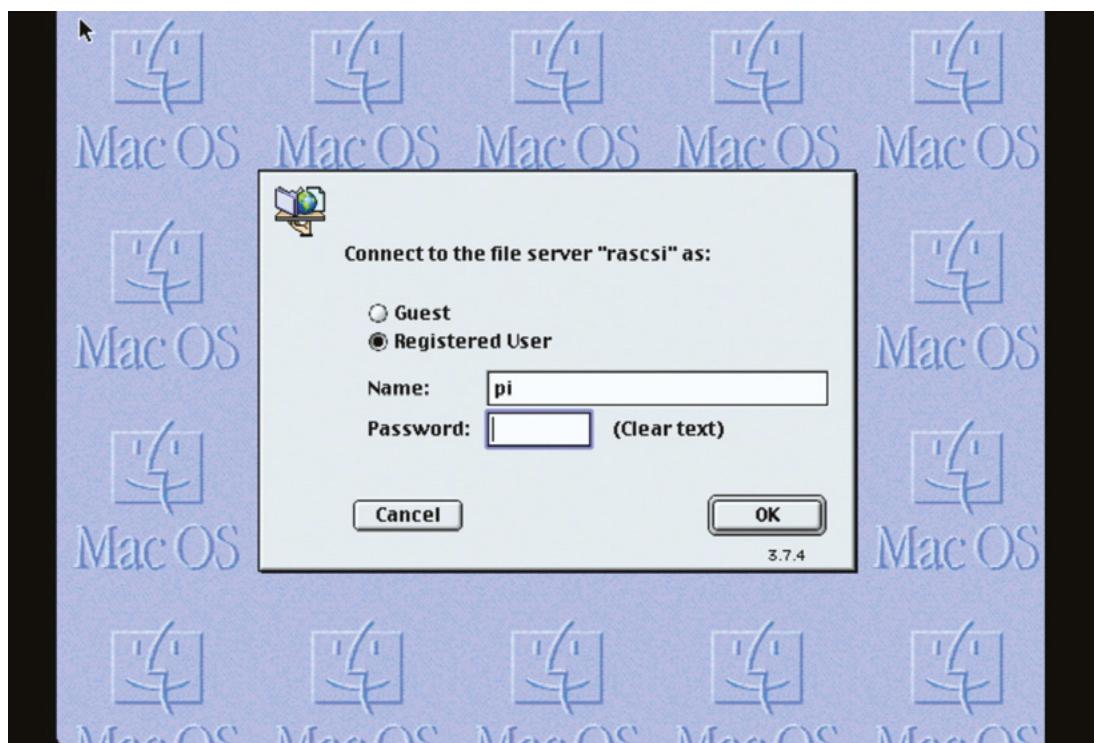
There is sufficient interest in SCSI peripherals to justify producing kits for them. Brian bought his first one from the now defunct Vintage Computer Centre, and a second from LandoGriffin on Tindie (magpi.cc/landogriffin).

It was a challenge to source the correct parts, particularly the SCSI cable. "You'll definitely want to make sure you understand what connector your computer uses; there are 25-, 50-, and 68-pin variants in different configurations, so make sure you have something that plugs into the PiSCSI on one end and something that plugs into your computer on the other," he advises.

Setting up PiSCSI for his MacQuadra "worked pretty much out of the box", says Brian, he



▲ Brian enjoyed old style web browsing on his Quadra too



▶ Logging into an AppleShare server

tweaked it over time “configuring Raspberry Pi to serve up AFP [Apple Filing Protocol] using netatalk [magpi.cc/netatalk] to let the Mac mount directories over the network,” as well as adding the proxy web browser “to allow older web browsers to make sense of the modern web”.

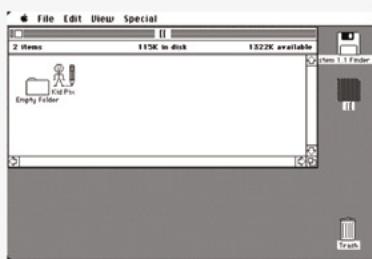
Getting PiSCSI to work with an old ThinkPad was a little more tricky (as his blog details: magpi.cc/rs6000). “The PowerPC ThinkPad runs

“Raspberry Pi is going to run faster, more quietly, and be much more reliable than some old SCSI drive”

IBM’s AIX, and I was using PiSCSI to emulate a boot CD-ROM, so I could install AIX on the internal hard drive. I couldn’t get it to work until I realised I needed to tell PiSCSI to present the CD-ROM image with 512-byte blocks.” He also warns that RaSCSI/PiSCSI serves up disk images, which you can, in theory, mount from within Raspberry Pi OS. “I learned the hard way not to do this while the Mac has mounted the disk image. It will corrupt the image.”

Having successfully revived a brace of computers using PiSCSI, Brian has learned there are “shelves and shelves of vintage computers at the Rhode Island Computer Museum that talk SCSI,” which should keep him busy for the next few months. M

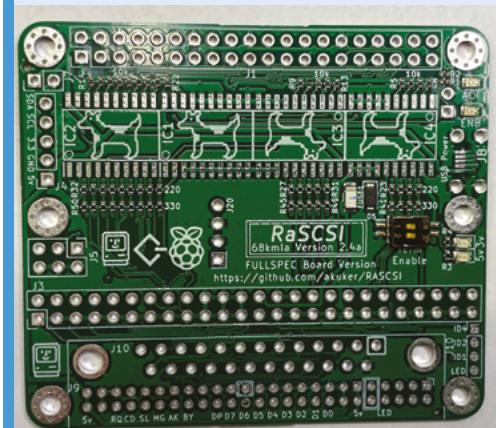
▼ James Friend’s site (jamesfriend.com.au/pce-js) runs Mac OS 11 in any browser



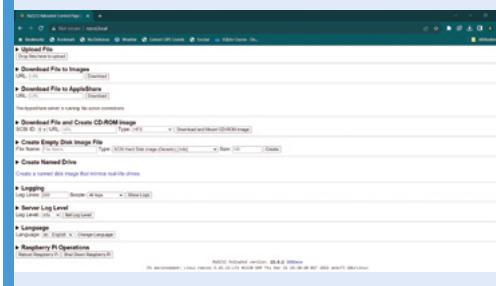
Only connect



01 The easiest way to add SCSI connectivity via Raspberry Pi is to buy a ready-made kit and source appropriate cables for the peripherals you want to use with your Mac or other old computer.



02 Solder headers onto the board and plug in the Raspberry Pi. Use the SCSI cable to connect the Mac to the RaSCSI board.



03 Use the instructions at magpi.cc/piscsigit to configure the web interface. You might also want to install a vintage web proxy: magpi.cc/macprox.

POV Candle

This RP2040-based persistence-of-vision project has **Phil King** in a spin



MAKER

Tim Jacobs

Tim Alex Jacobs, aka mitxela, is an inventor, artist, and hardware hacker based in London, UK.

mitxela.com

Tea-light candles are ideal for mood lighting, but naked flames can be dangerous – and those off-the-shelf artificial LED versions aren't very convincing.

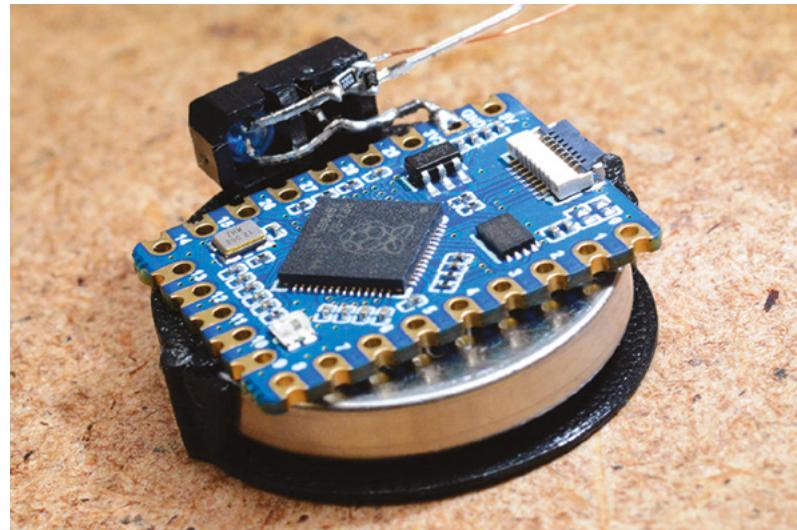
Taking a different approach, Tim Jacobs's POV candle spins an LED matrix at high speed to create the illusion of a hovering 3D flame.

"There was a discussion at a pub meet-up about electronic candles, of which there are many different types," recalls Tim. "I was specifically trying to think of a design that would look right from any angle. I'm not sure if I've achieved that yet, but this is only the first prototype!"

His solution was to make use of persistence of vision (POV) – the brief retention of a visual impression on the eye's retina that creates the illusion of a moving image in film and television.

Shine a light

While the general consensus was that POV displays require too much in the way of



supporting machinery to make them work, such as bearings and slip rings, Tim had the bright idea of getting a whole tiny device, including battery and motor, to spin.

"Getting the software to work and generating those animations took maybe a couple of days of experimenting"

He quickly threw together an LED matrix board design. "It took about a week for the matrix PCB to arrive," he says, "which gave me a little time to think about how it was going to go together, but once I had all the parts building the whole thing only took a couple of hours."

For the brains of the device, he looked at using Raspberry Pi Pico, but eventually opted for a Waveshare RP2040-tiny partly due to its smaller size (about half that of Pico).

"Aside from the low cost and easy availability, one of the nice things about the RP2040 is that you can set, or get, all of the GPIO pins in a single clock cycle," he notes. "This makes implementing a custom matrix very easy. The software [for the project] does very little at all: it just steps through an array of numbers to send to the GPIO port."

The flame effect is an animation based on volumetric data and rendered in the open-source Blender 3D modelling application. Tim first experimented with a rotating cube and fluid simulation. "Getting the software to work and generating those animations took maybe a couple of days of experimenting."

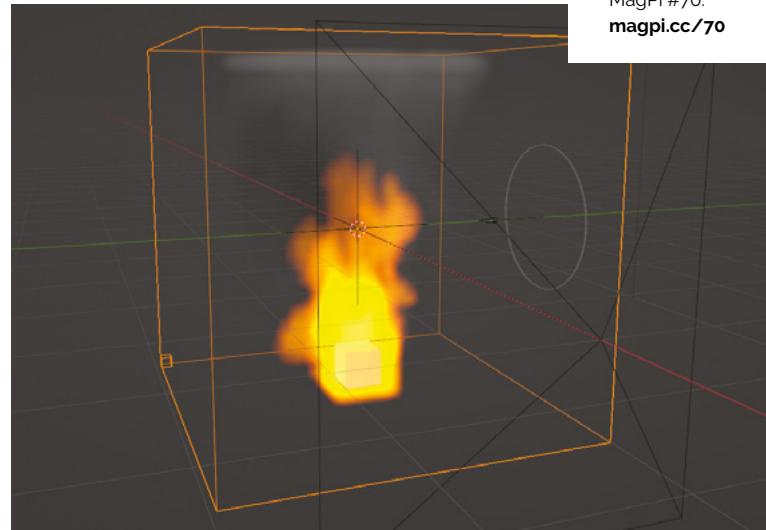


Spinning around

The candle is spun around by a Mabuchi RF-410CA motor Tim had to hand. This can rotate at up to 5900rpm, but for a 30fps animation he used PWM to reduce it to around 1800rpm.

Power for the project is supplied by a LIR2450 coin cell housed in a custom 3D-printed battery holder. “The amount of current a battery can deliver is related to its capacity,” says Tim, “so it’s not just about making the battery last as long as possible, it’s also that a smaller battery might not be able to drive the motor at all.”

From this initial prototype, “the next step is to make a circuit board out of the hand-wired mess it is currently, and then increase the resolution of the display. We can roughly double the number of LEDs without changing the design, but beyond that we’ll need to rethink how the display is driven.” **M**



Quick FACTS

- ▶ Any volumetric video can be displayed by the candle
- ▶ The matrix's low resolution means video files are quite small
- ▶ The candle can also show scrolling text messages
- ▶ Tim has created numerous Raspberry Pi and Pico projects
- ▶ His Thermal Printer Polaroid was featured in The MagPi #70: magpi.cc/70

OneInchEye

Will Whang designed his own lens for his Raspberry Pi solar photography endeavours.

Rosie Hattersley eyes it up



MAKER

Will Whang

Will is an atmospheric scientist who enjoys making and tinkering with all kinds of fascinating hardware in his projects.

willwhang.dev

You may have read about OneInchEye in issue #137 of The MagPi: it was a lynchpin in the development of Csaba Nagy's CinePi XL filmography project. Sensing there was a great story behind the creation of the dedicated 20.3Mp sensor, we got in touch with its maker, Will Whang, to hear how OneInchEye (named because of the size of the CMOS) came about – and to admire the resulting photography.

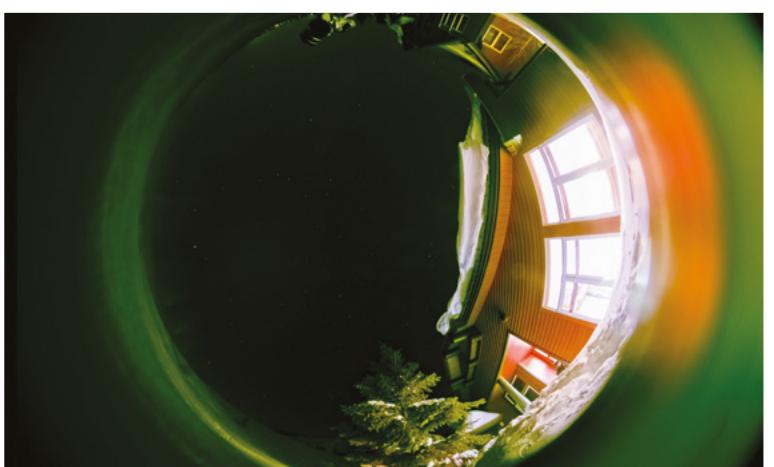
Will has always been interested in solar imaging and says “the sight of the mighty sun temporarily obscured, and events like the Venus transit,” have always left him in awe. He studied atmospheric science where he had plenty of opportunities to use measurement tools and environmental sensors. “With the advent of Arduino and Raspberry Pi, I realised I could build atmospheric science-related projects, ranging from a compact weather station to high-altitude ballooning, both for personal exploration and for my undergraduate lab.”

Computing and electronic engineering were natural next academic steps, and he has been tinkering and making ever since.

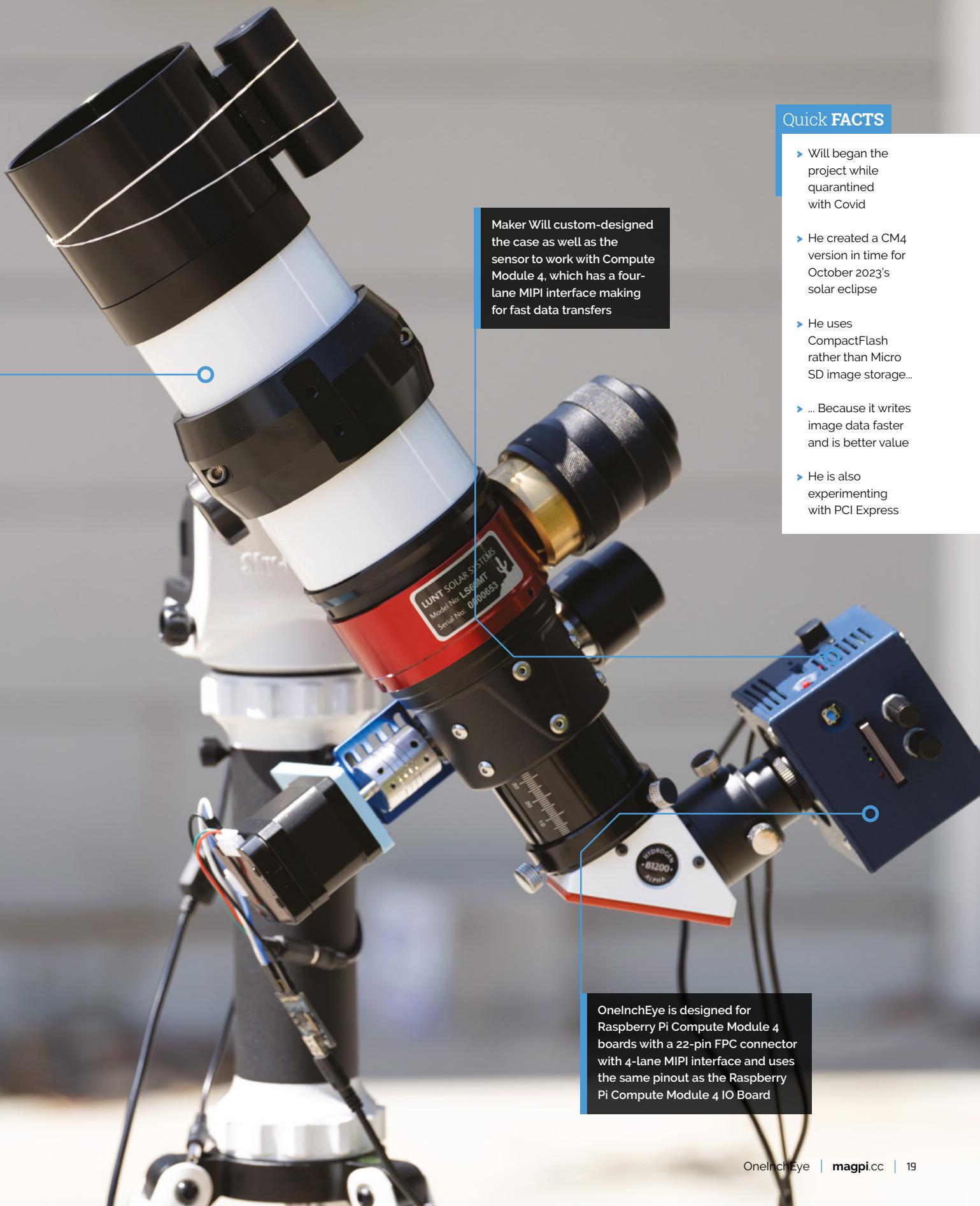
Development lab

With a number of Raspberry Pi projects, from CM4 clusters to Nixie clocks under his belt, he returned to his love of the sky in 2020. Will’s solar photography project used Raspberry Pi 4 and Raspberry Pi High

Photos captured by the telescope and camera are saved to CompactFlash, which Will rates as faster and more efficient than all but the most expensive microSD cards



► The OneInchEye can be used to amazing effect!





Will's solar photography project used Raspberry Pi 4 and a Raspberry Pi High Quality camera to track the sun

▲ The Waveshare display provides invaluable image previews and information



Warning! Solar retinopathy

Looking directly at the sun, even for a short period of time, can permanently damage your retina. Use solar glasses that meet ISO 12312-2 standards.

magpi.cc/solarsafety

► Timelapses and sun tracking create memorable images

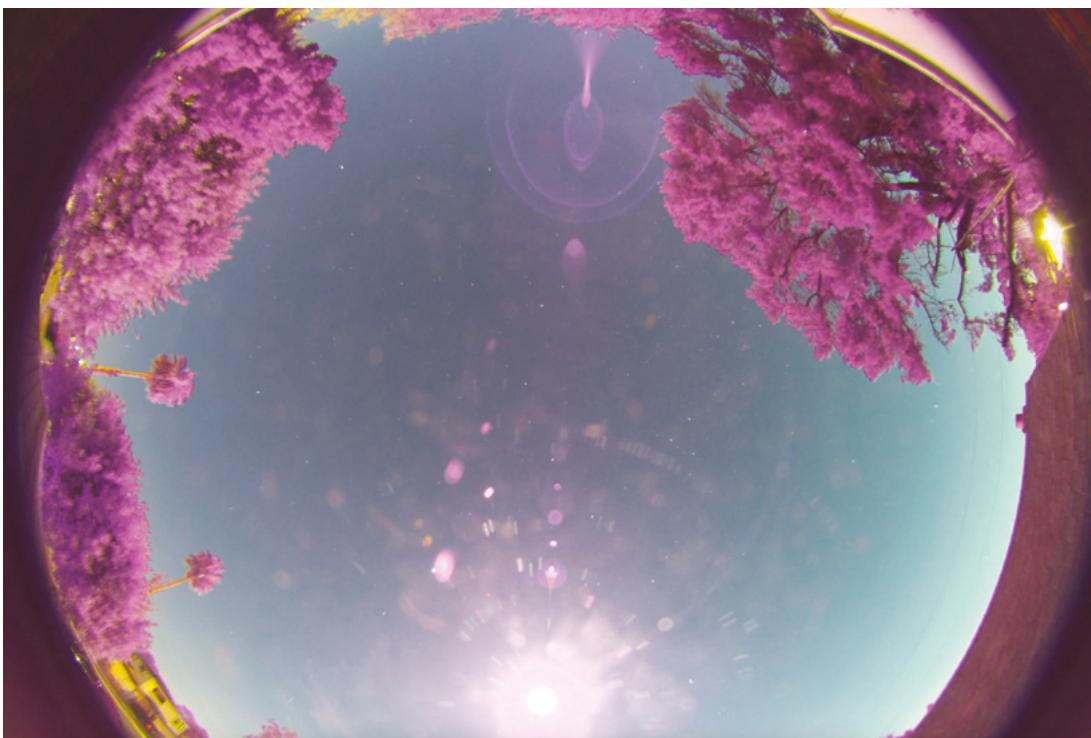
Quality camera to track the sun. "The results were astounding," he tells us.

In 2023 Will felt he had pushed the Sony IMX camera module found in various smartphones to its limit, and set about developing his own image sensor using the detailed documentation of Raspberry Pi's camera modules. The goal was to capture the solar eclipse in October 2023.

These in turn helped Will develop a modular camera system based on the image sensor boards he had developed. His aim was versatility, "easily adapting to various functions such as a video recording camera for travel or an astrophotography camera for use with my telescope, all without the need for a complete redesign".

He says "Raspberry Pi's comprehensive documentation on camera-related topics, such as their camera tuning guide, was instrumental in helping me write code for image sensors that are not natively supported by Raspberry Pi." He notes that many other single-board computers are equipped with a MIPI CSI-2 camera connector, "yet they often lack detailed documentation, offering only basic guidance for a limited range of camera sensors".

Will wanted to make maximum use of Raspberry Pi Compute Module 4 including its four-lane CSI and DSI connectors, as well as PCIe storage





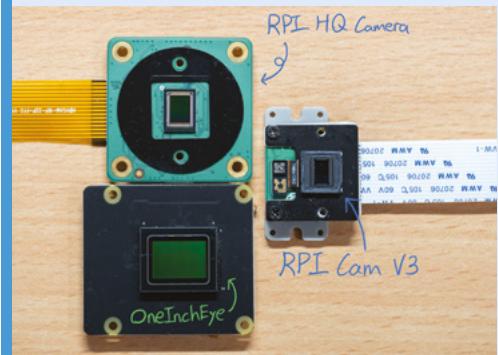
using CFexpress and a write speed of 400MBps. It can capture 4K images at 30fps (frames per second) or 20MP at 12fps for RAW recording. He describes CM4 as being at the heart of his project, processing output from the camera sensor, storing it on an NVMe drive, and enabling remote control and preview of the camera. However, he's looking forward to the greater RAW recording possibilities of the next iteration and also working on integrating the RP2040 into the system to digitally control an EF-mount lens.

Striving for more

Will is delighted that Raspberry Pi supports libcamera with such aplomb, and believes it opens the door for more photography enthusiasts to experiment with their own camera designs. Having developed OneInchEye for CM4, he is already planning ahead to use it to capture the solar eclipse in Spring 2024. *M*

▲ Once built, Will recommends 'tuning' the camera's focal and object recognition abilities

A larger lens



01 As well as Compute Module 4 and Raspberry Pi HQ camera lens with mount, you will need a Waveshare 2.8in LCD display. The full hardware list is at magpi.cc/oneincheyegit.



02 The OneInchEye CMOS, custom board and other hardware fit perfectly inside the custom case Will designed for the project, and which he found the most challenging aspect.



03 A limited number of Will's hardware kits can be bought via Tindie, including the camera mount (lens not supplied). You will also need to install CinePi RAW to optimise the camera setup: magpi.cc/cinepirawgit.

CarbonaraGPT

When you're having trouble cooking and the world isn't quite futuristic enough to have robot chefs, it's time to turn to a chatbot. **Rob Zwetsloot** samples the menu



Michele Ippoliti

MAKER

A Roman computer engineer in a 'large and boring company'. His friend Luciano 'Lucianone' Remo helped him with some of the preliminary design.

The key to a lot of Italian cuisine is simple, high-quality ingredients combined with precision. The carbonara in Rome tends to be a bit better than the one out of your kitchen, after all. Not all Italians are master chefs themselves though, so Michele Ippoliti has come up with a solution: CarbonaraGPT

"Its goal is to become the Alexa for Romans who are a total mess at cooking," Michele explains to us. "Essentially, it's a Python script that records your voice, translates it into text, and asks ChatGPT to provide you with [recipes]. The fact that ChatGPT can respond in any dialect or language makes it all the more fun! For now, I've focused on Roman dialect because in Rome, if you ask for any recipe, they explain it to you in a strong dialect, especially in certain neighbourhoods like Trastevere and Torbellamonaca... After producing the response, the text-to-speech plays the answer through the speakers, listing the ingredients and cooking steps. I've even made it repeat the steps upon

"The fact that ChatGPT can respond in any dialect or language makes it all the more fun!"

voice request. It's simple and fun. If you want, it can also send the recipe to you on Telegram."

Torta di lamponi

Michele loves Raspberry Pi, starting off using it as a retro game console, then a Kodi box, before using it for other projects.

"After a small briefing with [my friend] Lucianone, I started with the software." Michele tells us. "I realized there were many Python libraries and examples for speech-to-text. I

► A Bluetooth mic and speaker carry the instructions to and from the user

began putting together the script, first testing it on a Windows computer and then on my personal one with Linux. I tried various libraries, and in the end, I bought a used Raspberry Pi 3B... and began testing and modifying the code directly on Raspberry Pi OS, initially with a USB microphone and speakers connected via [the 3.5mm jack]. At one point, I convinced myself that it had to work with Bluetooth speakers with a built-in microphone purchased on Amazon. That's where the real struggles began! But in the end, I made it work!"

The technical side of the operation is hampered a little by Wi-Fi reception quality and raw



CarbonaraGPT
bot

9 ottobre

Ecco la tua ricetta: Ecco la ricetta d'li rigatoni alla carbonara in stretto dialetto romano:

Ingredienti:

- 320g d'li rigatoni
- Ojo d'oliva
- 150g d'guanciale o pancetta, tagliata a dadini
- 2 uova
- 100g d'pecorino romano grattugiato
- Pepe nero q.b.
- Sale (se serve)

Istruzioni:

1. Cuoci li rigatoni in abbondante acqua salata finché so' al dente. Ricorda de mantene' un po' d'acqua de cottura pe' la salsa.
2. Nel frattempo, in 'na padella, fa' scaldà un po' d'oj d'oliva e aggiungi er guanciale o la pancetta tagliata a dadini. Fa' soffriggere finché diventa croccante.
3. In 'na ciotola, sbatti le uova e aggiungi er pecorino romano grattugiato. Amalgama bene.
4. Scola li rigatoni e mettili nella padella co' er guanciale o la pancetta. Mescola bene.
5. Togli la padella dal fuoco e versa subito la mistura d'uova e formaggio sui rigatoni. Mescola rapidamente pe' distribui la salsa in modo omoogeneo. Se serve, puoi

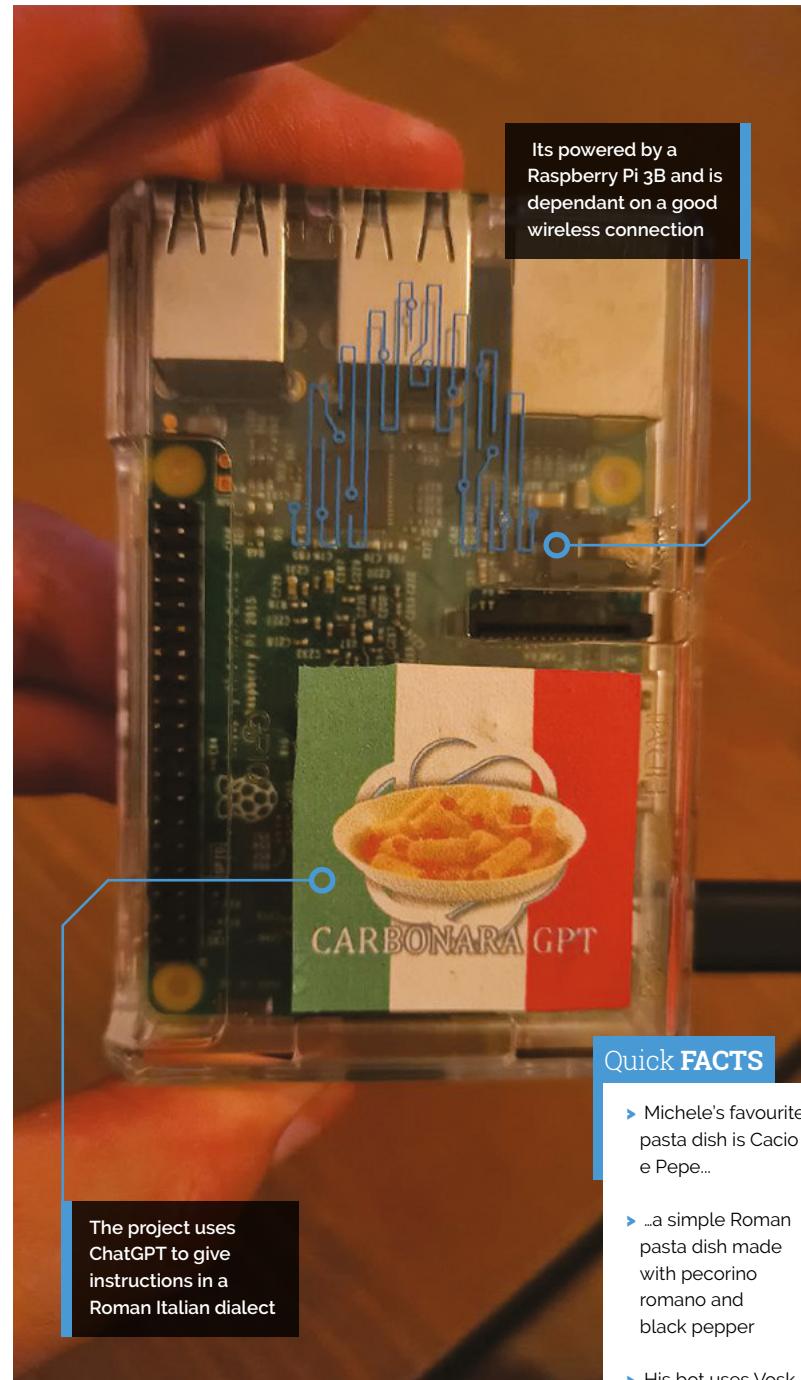
A recipe in action – complete with Roman Italian

power – Michele reckons it would run just fine on a Raspberry Pi 4 as long as your internet connection is strong.

Cooking sins

"As a cook, CarbonaraGPT is not great, but still better than those who put cream in carbonara." Michele admits. "The AI gets confused and believes she's authorised to add typical Roman ingredients to any dish, just because you asked her to speak in [the] Roman dialect."

It sounds like progress, but still nowhere near the level of an Italian grandma. As for cream in carbonara as we like to do in the UK, Michele has some parting words: "Sometimes,



we're not entirely aware of our mistakes. Some Italians think that Gigi D'Alessio [a popular Italian singer] is better than Paul McCartney. You folks insist on putting cream in carbonara. Both great mistakes. However, there is room for improvement as long as Gino D'Acampo is broadcast on British TV."

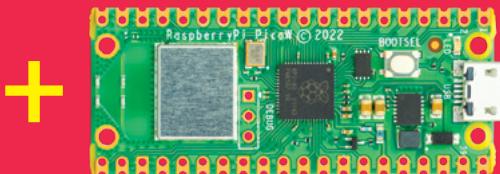
We agree. And while we're at it, switch out the spaghetti for tagliatelle or fettuccini with your bolognese too.

Quick FACTS

- ▶ Michele's favourite pasta dish is Cacio e Pepe...
- ▶ ...a simple Roman pasta dish made with pecorino romano and black pepper
- ▶ His bot uses Vosk for speech-to-text, connected to an Italian dictionary
- ▶ The response is made with Google's text-to-speech API
- ▶ According to Michele, ChatGPT doesn't know how to cook sea bream

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**

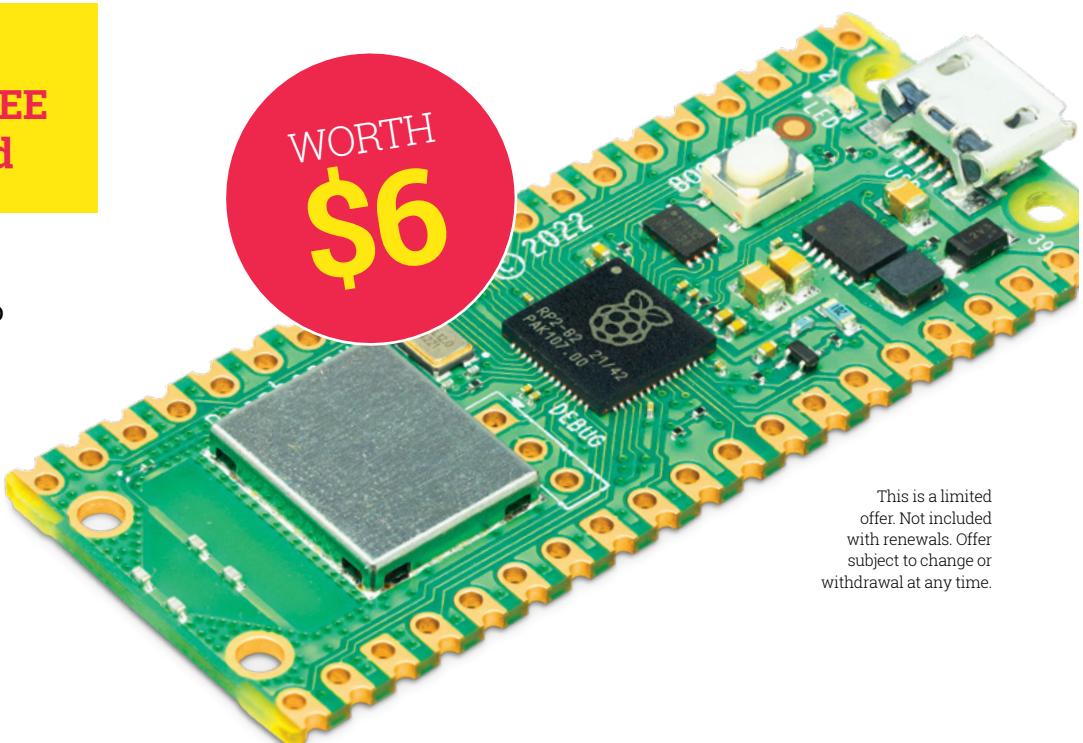
Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A free Pico W is included with a 6-month subscription in USA, Europe and Rest of World.

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

WORTH
\$6



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**



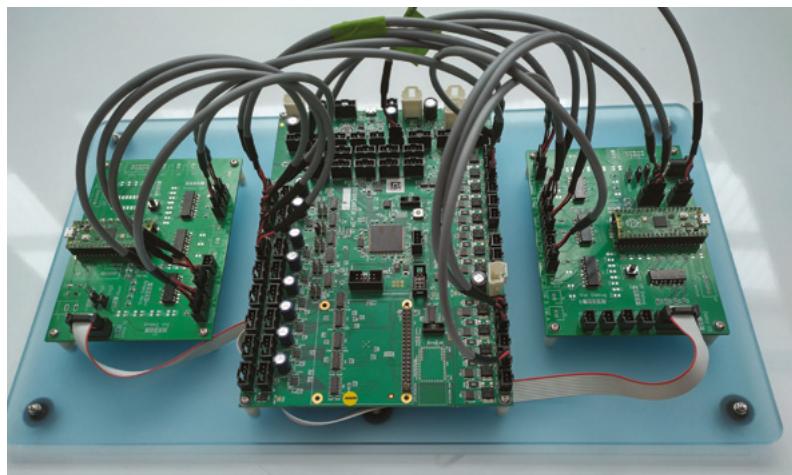
SUCCESS STORY magpi.cc/success

Hardware-in-the-loop testing by Green Custard

Raspberry Pi Pico supports powerful and robust automated testing. By **Neil Bryan**, Principal Consultant, Green Custard

Britvic-owned Aqua Libra Co's sustainability plans depend on robust, reliable dispensing equipment. Green Custard designed a test solution that takes advantage of the flexibility of Raspberry Pi Pico to provide rigorous testing that would be difficult to achieve by other means.

Under its "Healthy People, Healthy Planet" mission, Aqua Libra Co, owned by UK drinks giant Britvic, is working towards a sustainable future for workplace and retail drink dispensing. The company worked with Amazon Web Services (AWS) and Cambridge-based professional services company Green Custard to develop the Aqua Libra Flavour Tap, which eliminates the need for single-use packaging by delivering beverages straight to a reusable vessel. The tap can dispense still or sparkling water and a variety of flavours.



Within the tap are several control systems, one of which reacts to user input and relays flavour cartridge information to AWS. Another directly controls the dispense, switching valves and solenoids for precisely controlled periods as well as driving pumps for the exact quantity of flavour for a specific drink recipe. As part of Green Custard's work to help develop the tap, a solution for testing the software and hardware running on the dispense control board was needed.

THE CHALLENGE

The cost of finding bugs rises exponentially throughout the development and deployment process. Several studies have revealed the cost of resolving a bug, which can range from \$100 for an early change of requirements to many tens of thousands of dollars if fixed after the software is deployed. Reproducing an intermittent bug seen in the field can be very difficult, and unless the bug can be observed, it may well be marked as "unable to reproduce" and ignored.

Reviewing code can uncover implementation errors which could manifest as a bug, but testing is the cornerstone of modern software development. It takes many forms, including unit tests, integration tests, and so on. Testing software running on an embedded system can be particularly difficult, because often the unit under test requires inputs from external sensors which provide signals which change over time. A solution is to simulate these external signals, which allows the embedded software to function as if it were running within a complete system, not just an isolated test setup.

The inputs can be simulated using a technique called hardware-in-the-loop (HIL). With HIL, sensor inputs can be generated automatically, without user interaction. This helps to test how the software behaves under different conditions.

The Aqua Libra dispense controller consists of a microcontroller running a real-time operating system, and is used to monitor weight and fluid flow, as well as to drive stepper motors and electromechanical valves. Manual testing requires placing a vessel on a weight-measuring device to enable a dispense to be made. The subsequent valve switching and pump activity need to be monitored to confirm that the expected drink is dispensed. Along with this is the need to simulate the flow of liquid by generating a sequence of pulses like those produced by an electronic flow sensor.

THE SOLUTION

Raspberry Pi Pico is based on Raspberry Pi's RP2040 microcontroller, and has a peripheral which is very useful for HIL testing. In the test tool for Aqua Libra's dispense controller, the programmable input/output (PIO) block is used to generate very fast serial data, as if derived from a 24-bit ADC, to simulate a strain gauge. The ADC output can be set arbitrarily, and triggers events on the system under test. The PIO is also used to simulate the output from a flow sensor, and the simulated output is indistinguishable from that of the real hardware. The unit under test drives many stepper motors, and the PIO is used to read the two-phase motor drive outputs, conditioned through optocouplers. This approach can measure every single step which would be applied to the motor, even when the step rate is in the kHz range. It provides unprecedented levels of instrumentation for a low-cost test tool.

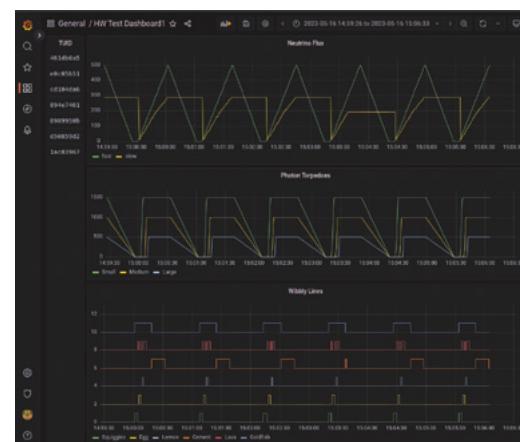
Similarly, solenoid and valve states are monitored as the test progresses. RP2040 contains two PIO blocks, and a single stepper channel uses all the program for a single PIO. It is, therefore, necessary to daisy-chain multiple RP2040s to enable monitoring of all the I/O of the board under test. This requires a novel communication technique over serial that can address a unique board or all boards to configure for at the onset of a test, and recover the data as fast as possible during the test. The states of the I/O on the unit under test are sampled at 20Hz, and the collected data is written to a time series database together with a unique test identifier. Post-test, the collected data can be plotted using Grafana with a dashboard that is configured to represent a logic analyser.

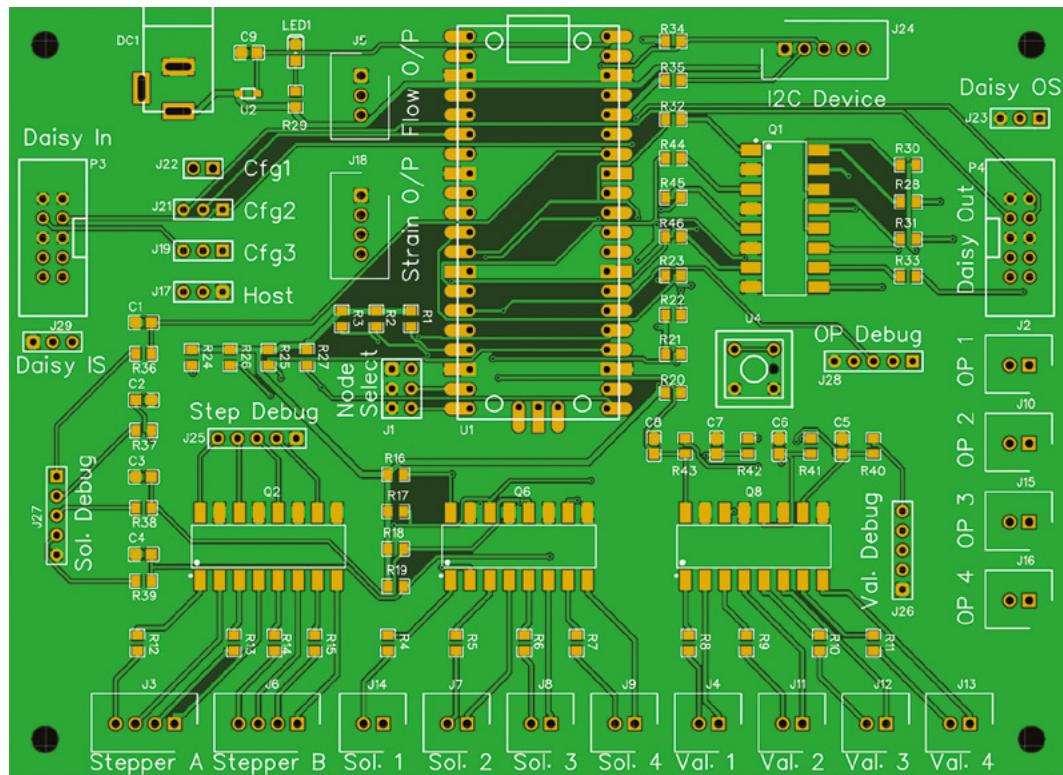


This is very useful for system development by the hardware and mechanical team.

The database can also be queried and, depending on the type of test, the data may be processed to signify a pass or fail condition. The whole system is fully automated and can run thousands of standalone tests without human interaction.

The software running on RP2040 is identical for each test node, and the GPIO is used via jumper leads to provide a unique node identification which is read at runtime by the RP2040 application. This tailors the node functionality, loads the PIO programs for that node, and means that only a single image needs to be maintained. A PCB was





designed and fabricated with a Raspberry Pi Pico on Node 0 acting as the main controller, with the option to connect multiple nodes to provide more input monitoring.

WHY RASPBERRY PI?

Raspberry Pi Pico is attractive as a basis for this kind of testing for several reasons:

- Cost:** the entire PCB with on-board power supply and flash storage can be purchased for less than £4. This is beneficial since it means several testing setups can be made for different sites, and if a Pico is damaged during development, the replacement cost is not prohibitive.
- Form factor:** Pico uses 40-pin DIL headers, which can be soldered directly to a PCB or fitted with low-profile turned pin headers to allow for easy replacement.
- PIO:** PIO, Green Custard reports, is “a superb idea, and gives such flexibility that it cannot be overstated how useful it is”.
- Ecosystem:** The whole Raspberry Pi ecosystem – from blog posts to GitHub repos, YouTube tutorials, and the Raspberry Pi forums – provides a lot of useful information which reduces the development time significantly.

- Availability:** Raspberry Pi Pico remained readily available for immediate shipping even during the global silicon shortage, so orders could be placed with confidence.

Programmable I/O (PIO) is an incredibly flexible feature of Raspberry Pi Pico. It allows for the development of hardware interfaces that would be difficult without a full-blown FPGA

The Raspberry Pi ecosystem reduces the development time significantly

implementation. The parts cost and the power provided by PIO made Pico a compelling choice. The SDK is well documented and there are many useful code examples in the accompanying GitHub repo. Pico is my preferred part for low-cost, low-volume development. I am pleased with the HIL testing framework. It gives me confidence that any change made to the software running on the target hardware has not introduced any regressions and performs as expected.

THE RESULTS

The flexibility provided by the HIL testing tool allows for many combinations of dispensed

```

2023-05-16 16:59:34 INFO      [TEST] starting: test 7 with tuid: cfb401ca
2023-05-16 17:00:40 INFO      [TEST] writing 461 test records to the database
2023-05-16 17:00:40 INFO      [TEST] tuid: cfb401ca start_time: 2023-05-16T15:59:50.366079Z
2023-05-16 17:00:40 INFO      [PASS] test case: cfb401ca start/stop conditions
2023-05-16 17:00:40 INFO      [PASS] controller1: 1499
2023-05-16 17:00:40 INFO      [FAIL] controller2: 818
2023-05-16 17:00:40 INFO      [FAIL] controller3: 0
2023-05-16 17:00:40 INFO      [PASS] abc total: 287
2023-05-16 17:00:40 INFO      [PASS] xyz total: 499
2023-05-16 17:00:40 INFO      actuator1 opened at 0:00:00          for 0:00:03.613210
2023-05-16 17:00:40 INFO      actuator2 opened at 0:00:04.412389  for 0:00:01.598692
2023-05-16 17:00:40 INFO      switch2 opened at 0:00:03.613210  for 0:00:00.799179
2023-05-16 17:00:40 INFO      switch3 opened at 0:00:00          for 0:00:06.090997
2023-05-16 17:00:40 INFO      [FAIL] turbo transitions: 1
2023-05-16 17:00:40 INFO      [FAIL] test cfb401ca failed

```

recipes to be run in an automated environment. A single test can request many thousands of dispenses which take hours to complete, checking each and every pump step and valve open-and-closure cycle. This would be almost impossible to accomplish manually. The test results can then be viewed either as text-based results, which are ultimately pass or fail conditions, or as a graphical display, similar to a logic analyser.

In Green Custard's system, the test results are converted to records and written in bulk to an instance of an Influx time-series database. The test input JSON file contains metrics which are used to determine whether the test has passed or failed. A Python script parses the file, extracts the relevant metrics for that test, and stores them ready for use. When the test is complete, the Influx database is queried, and checks for specific features can be made; for example, the number of steps of the pump motors, or the valve opening time and duration to millisecond precision.

A second, graphical, way of displaying data is provided by Grafana, an open source data analytics and visualisation platform. This view allows teams not involved in software development – for example, the people who develop flavours, and liquid specialists – to use the test data to determine whether the switching times and sequences are as expected. The image below, for example, shows three distinct charts that were generated sequentially during a dispense of four separate drinks. The top chart shows the data around the dispensing progress; the middle chart displays more data related to the dispense which changes over time; and the bottom chart shows the opening and closing of control elements within the tap.

This level of detail, together with the ability to test many input conditions, makes for a very powerful testing tool.

In this application, Raspberry Pi Pico enables Green Custard to provide a low-cost solution to simulate different types of sensors and measure the resulting change in outputs. It's possible to monitor the states of different components in real time, and write the collected data to a time-series database. Lastly, and crucially, the data can be plotted on a dashboard for further analysis. The whole process is automated and can run thousands of tests without human interaction.

Green Custard's hardware-in-the-loop system supports test cases that inject failure conditions that it would be very hard, or even entirely unachievable, to recreate on a real system, making it possible to assess the robustness of the software under test.

Without Raspberry Pi Pico, it would be very difficult – if not impossible – to identify intermittent bugs that could affect performance in the field. Using hardware-in-the-loop, Britvic can be confident that the Aqua Libra Co Flavour Tap will perform as expected.

To learn more about how Green Custard helped develop the Aqua Libra Co Flavour Tap, read their full case study (magpi.cc/greencustard). 

```

2023-05-16 17:00:40 INFO      [TEST] starting: test 8 with tuid: 69e2744f
2023-05-16 17:01:46 INFO      [TEST] writing 464 test records to the database
2023-05-16 17:01:46 INFO      [TEST] tuid: 69e2744f start_time: 2023-05-16T16:00:56.000000Z
2023-05-16 17:01:46 INFO      [PASS] test case: 69e2744f start/stop conditions
2023-05-16 17:01:46 INFO      [PASS] controller1: 1499
2023-05-16 17:01:46 INFO      [PASS] controller2: 1001
2023-05-16 17:01:46 INFO      [PASS] controller3: 501
2023-05-16 17:01:46 INFO      [PASS] mdb weight total: 287
2023-05-16 17:01:46 INFO      [PASS] htb weight total: 499
2023-05-16 17:01:46 INFO      actuator1  opened at 0:00:00          for 0:00:03.372355
2023-05-16 17:01:46 INFO      actuator2  opened at 0:00:04.412035  for 0:00:02.238088
2023-05-16 17:01:46 INFO      actuator3  opened at 0:00:07.689258  for 0:00:01.119000
2023-05-16 17:01:46 INFO      switch1  opened at 0:00:17.361552  for 0:00:13.764388
2023-05-16 17:01:46 INFO      switch2  opened at 0:00:03.372997  for 0:00:01.039038
2023-05-16 17:01:46 INFO      switch2  opened at 0:00:06.650102  for 0:00:01.039156
2023-05-16 17:01:46 INFO      switch2  opened at 0:00:08.808391  for 0:00:02.638089
2023-05-16 17:01:46 INFO      switch3  opened at 0:00:00          for 0:00:17.361552
2023-05-16 17:01:46 INFO      [PASS] turbo transitions: 3
2023-05-16 17:01:46 INFO      [PASS] test 69e2744f successful

```

NEXT GEN RETRO

on Raspberry Pi 5

FEATURES
LEGAL
ROMS!

Smoothly emulate fifth- and sixth-generation consoles, including the original PlayStation, Dreamcast, Game Boy Advance and Wii

By KG Orphanides



Warning!
Copyright!

This feature discusses the use of legally distributed games and system ROM images for a variety of vintage computers. However, please note that many other titles are distributed online without their rights holders' permission, and are not legal to use under UK law.

magpi.cc/legalroms

Raspberry Pi 5 is significantly more powerful than its predecessors, and that translates to upgraded emulation performance. While Raspberry Pi 4 could emulate fourth-generation consoles like the Sega Mega Drive and Super Nintendo without breaking a sweat, and could even manage fifth-generation PlayStation games at CRT resolutions, Raspberry Pi 5 opens the door to emulating fifth- and sixth-generation console hardware, like the Dreamcast, GameCube and Wii.

When it comes to desktop emulation, there are a couple of things to be aware of. Raspberry Pi 5 is supported by a new version of Debian-based Raspberry Pi OS, Bookworm. This new OS has moved from using the X Window system (Xorg, X11, or just X) to Wayfire as the default

compositor – the display system that renders your graphical desktop for you. Wayfire uses the efficient new Wayland protocol, developed by freedesktop.org, the same project behind Xorg. (See “Where there’s a Wayland” boxout).

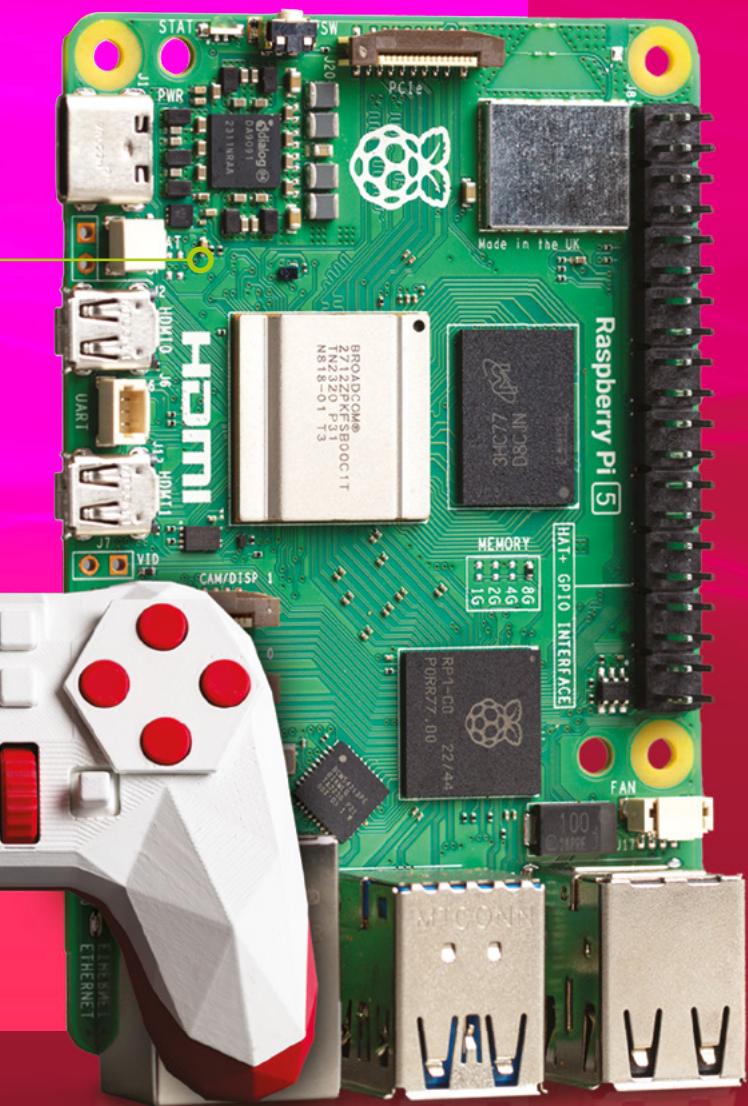
At the time of writing, the mesa 23.3 graphics stack, although released, hadn’t yet made it into the Raspberry Pi OS repositories, but should result in improved Vulkan support on Raspberry Pi 5.

YOU’LL NEED

- ▶ Raspberry Pi 5
- ▶ USB or Bluetooth gaming controller (such as 8BitDo Pro. magpi.cc/8bitdo)
- ▶ Raspberry Pi OS Bookworm
- ▶ An x86 computer (optional). Required to obtain the official PlayStation BIOS image

Raspberry Pi 5's faster hardware has opened up a whole new range of emulatable consoles

You'll want a controller, like this 3D-printed Alpakka (magpi.cc/alpakka), to play retro games



WHERE THERE'S A WAYLAND

Wayland is less mature than X11, and some software hasn't been updated to work with it, although Raspberry Pi OS does ship with XWayland, a compatibility tool that allows you to run individual programs in X11, which it'll automatically fall back to. You can also use the

`WAYLAND_DISPLAY="" <your-app-name>` command to force its use.

It's possible, via `raspi-config`, to switch Bookworm over to the Openfire window manager with a legacy X11 backend, used by Raspberry Pi 3 and earlier systems, but we're going to stick with

Wayland as this is where most active development is happening.

Raspberry Pi OS is also moving away from the OpenGL graphics rendering API to Vulkan. Legacy support is built in and most programs will fail over to either software rendering or OpenGL.

EMULATING WII AND GAMECUBE CONSOLES



▼ There's some fantastic Wii homebrew out there, like synthwave racer *Newo Zero*



Top Tip

Build your own GameCube game

Retro League, a car football game for GameCube using a novel game engine, is open source and ready to build or modify (on x86 architecture) at magpi.cc/retroleague.

01 Dolphins go Wii

The Dolphin emulator allows you to run GameCube and Wii games. Sadly, you can't directly play Nintendo optical discs in Dolphin and it's illegal to copy console game discs in the UK, but there's a vibrant Wii homebrew scene. Although there's a copy of dolphin-emu in the Raspberry Pi OS repositories, there's a newer one available, so we're going to build from source. Open terminal and enter the following:

```
$ sudo apt install ca-certificates qt6-base-dev qt6-base-private-dev libqt6svg6-dev git
cmake make gcc g++ pkg-config libavcodec-dev
libavformat-dev libavutil-dev libswscale-dev
libxi-dev libxrandr-dev libudev-dev libevdev-dev
libsfml-dev libminiupnpc-dev libmbedtls-dev
libcurl4-openssl-dev libhidapi-dev libsystemd-dev
libbluetooth-dev libasound2-dev libpulse-dev
libpugixml-dev libbz2-dev libzstd-dev
liblz2-dev libpng-dev libusb-1.0-0-dev gettext
$ git clone https://github.com/dolphin-emu/dolphin.git
$ cd dolphin
$ git submodule update --init --recursive
$ mkdir Build && cd Build
$ cmake .. -DUSE_SHARED_ENET=ON
$ make -j$(nproc)
$ sudo make install
```

Now enter:

```
$ dolphin-emu
```

...to start the Dolphin emulation software.

02 Connect a controller

If you're using a USB controller, plug it in. If you're using a Bluetooth controller, open Raspberry Pi's Bluetooth menu towards the top right of the desktop taskbar and click Add Device. Put your controller in pairing mode and select it from the Add New Device menu when it appears, most likely as "Wireless Controller". We used a Sony DualShock 4 controller, but you can use any USB or Bluetooth controller with Dolphin, including original Wii wireless controllers. We'll be using the same controller with all our emulators.

03 Configure a GameCube controller

In Dolphin, open the Options menu and select Controller Settings. Here, ensure that the Port 1 GameCube controller is set to Standard Controllers, Wii Remote 1 is set to Emulated Wii Remote, and

make sure Background Input is ticked at the bottom. Click the Configure button for the GameCube controller. In the GameCube Controller at Port 1 window, under Device, select your controller, then calibrate each button to your preferred layout by clicking on the button assignment and then pressing the controller button or movement you want to associate it with. The Modifier for each thumbstick should generally be set to that stick's click. The GameCube controller doesn't map entirely cleanly to most modern controllers. The Z button should be mapped to your controller's right shoulder buttons. This diagram will help with mapping: magpi.cc/gcconwiki. Give the profile a name and click Save before closing the window.

04 Configure a Wii controller

Next, we'll do the same mapping for an emulated Wii remote. This is an even more unusual controller, so we're just going to have to do our best here. At least there aren't many buttons to map. We mapped the D-pad to our D-pad, AB12 to our front buttons, and + and - to the left and right shoulder buttons. Open the motion emulation tab and map pointer controls to your left analogue stick. Name your file, save and then close the configuration window.

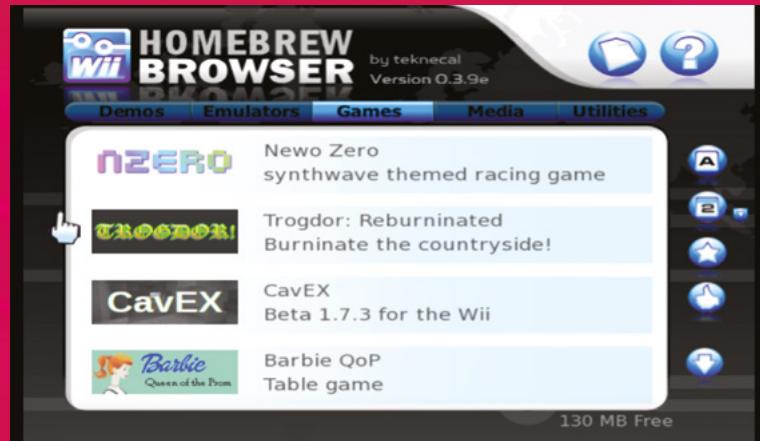
05 Getting started with Wii homebrew

There are plenty of homebrew games for the Wii but getting them running on Dolphin can be fiddly. Using the Open Shop Channel (oscwii.org) to install homebrew gives you a much higher success rate.

The OSC provides a similar experience to the software download channels that Nintendo used to offer for the Wii. You'll need a virtual SD card to be able to download games from the channel, but Dolphin will create this for you.

First, we'll create a directory where you'll store your games and download the open homebrew browser. Open a terminal and type:

```
$ mkdir -p Games/Wii
$ cd Games/Wii
$ wget https://hbb1.oscwii.org/unzipped_apps/
homebrew_browser/apps/homebrew_browser/boot.dol
```



06 Access the Open Shop Channel

In Dolphin, double-click the main pane to select your games directory – browse to the Games/Wii folder we just made. Click the Config button on the toolbar, select the Paths tab, and tick the Search Subfolders box. With that done, a file called `boot.dol` should now appear on Dolphin's main screen.

Double click it to launch the browser and, if this is your first launch, give it a minute to attempt to contact the server for updates. Press B to skip the update. You can press A to install it, which inevitably fails, as there's not been a new version released since 2012, although new software has been added to the Open Shop Channel itself.

Either way, the Homebrew Browser will then open. You'll find numerous freeware and open-source games and programs here. Try *Newo Zero*, a chill synthwave-themed racing game, and music utility Harmonium.

▲ The best way to access Wii homebrew is to install the Homebrew Browser and access the Wii Open Shop Channel

There's a vibrant Wii homebrew scene

07 Configure your storage

In Dolphin, open the Options menu and select Configuration. Select the Wii tab. Under SD card settings, you'll see that Insert SD card is already enabled. Below it, tick Automatically Sync with Folder – this will give us an editable version of the SD card that we can browse and modify as needed, and will sync changes every time you run Dolphin.

Dolphin automatically creates a sync folder in your `~/.local/share` tree, but this won't be indexed by default. Instead, create a `WiiSD` folder in the `~/Games/Wii` directory we made earlier and select it.

Click Convert File to Folder Now, and the games you downloaded to your SD card in the previous step will appear in Dolphin's main game menu, easily available for you to run. Not every homebrew game runs smoothly under emulation, but you'll have fun finding those that do.

EMULATING GAME BOY ADVANCE

01 GameCube's pocket pal

While the Wii is well supported, you'd struggle to find GameCube homebrew titles. Released in the same year as the GameCube, the Game Boy Advance was its more successful, pocket-sized sibling, and has one of the most vibrant dev communities around. We're going to install the mGBA emulator as a Flatpak package, so we'll start by installing the package manager and repository. The Flathub app store (flathub.org) only supports 64-bit operating systems and processors, tagged aarch64 in its index. Open a terminal and type:

```
$ sudo apt install flatpak
$ flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
```

Now reboot Raspberry Pi to ensure that Flatpak's desktop integrations can set themselves up correctly. With that done, you'll be able to search for software from the command line or at flathub.org, although Flatpak installation from the browser isn't currently supported by the Raspberry Pi OS software centre.

With a Game Boy Advance emulator like mGBA, you can buy and run new digital releases like the excellent adventure platformer *Goodboy Galaxy*.
GalaxyChannel



02 GBA: Online

Open a terminal window and type:

```
flatpak install
```

```
flathub io.mgba.mGBA
```

Choose "yes" when prompted to confirm the package and its dependencies, and it'll be added to your Games menu. If you open it from there, though, you'll get an error



message telling you that it's failed back to software rendering mode and Wayland won't correctly render the window, although the emulator will be usable. This is likely to be resolved when Mesa 23.3 comes to Raspberry Pi.

Until that's fixed, we'll run mGBA with a terminal command:

```
WAYLAND_DISPLAY="" flatpak run io.mgba.mGBA
```

We're still on software rendering, but now the mGBA desktop application can be moved and resized.

Beyond these teething problems, mGBA works out of the box. Just load a ROM in .gba format from the File menu. You can play in fullscreen mode by pressing **CTRL+F**.

03 Get some GBA games

There's been a flurry of recent GBA releases, largely distributed on the itch.io digital store. We've been playing the fantastic *Goodboy Galaxy* (magpi.cc/goodboygalaxy), and you can download the prologue/demo for free at **magpi.cc/goodboygalaxydemo**. Help a cat rescue their family in free platformer *Feline* (magpi.cc/feline), tower-defend your oozy kingdom in *Slimelord Siege* (magpi.cc/slimelordseige) or go dungeon delving in *Inheritors of the Oubliette* (magpi.cc/oubliette).

You can check out over 100 more GBA games in our Itch collection at magpi.cc/itchgba.



EMULATING DREAMCAST

01 Dream on

We'll install the Flycast emulator to play Dreamcast games. Flycast is also available as a flatpak, but at the time of writing this wasn't working reliably on Raspberry Pi OS. This is likely to be fixed by the time you read this. Open a terminal and type:

```
$ git clone https://github.com/flyinghead/flycast.git
$ cd flycast
$ git submodule update --init --recursive
$ mkdir build && cd build
$ cmake ..
$ make -j4
$ sudo ln -s ~/Software/flycast/build/flycast
/usr/local/bin
$ flycast
```

02 Configure Flycast

Click the Settings button at the top right of the Flycast window and, in the General tab, click the add button in the Content Location bar. This should direct you to the directory where you'll be keeping your Dreamcast games. We've opted to put them in `~/Games/Dreamcast`. Put each game in its own directory – Flycast will check all subdirectories automatically. Now click on the Advanced tab, and under Other, tick the HLE BIOS box. You can optionally keep each game's save files on its own virtual VMU (Visual Memory Unit, also known as a Visual Memory System) by ticking the per-game VMU A1 box at the bottom of the Controls tab. Now click Done.

03 Get some Dreamcast homebrew

Now we'll need some games. As with many classic consoles, hobbyists and indie developers are still making both free and commercial software for the Dreamcast. Crazy Viking Studios granted Marc Hall permission for their 2013 platformer *Volgarr the Viking* to be ported to the Dreamcast and distributed for free. Download a copy from volgarr.rkd.zone and unzip the file into a subdirectory of your Dreamcast folder. If you've got Flycast open, the game should immediately appear as a playable disc image.

For a feel of what DC homebrew was like back in the day, download the Inducer collection from magpi.cc/inducer and extract it into your Dreamcast games directory, but be aware that the games in this collection don't always emulate smoothly. You'll find more Dreamcast homebrew on the Dreamcast Talk forum's new releases pages (magpi.cc/dreamcasttalk) and in our Itch collection magpi.cc/itchdreamcast). The Dreamcast Junkyard (thedreamcastjunkyard.co.uk) regularly covers new releases.

▼ Arcade platformer *Volgarr the Viking* was ported to Dreamcast in 2015 with the blessing of the original developers



WHAT'S A VMU?

Visual Memory Unit (VMU) devices were the Dreamcast's external flash memory card modules, which plugged into the controllers, allowing players to take their saved data around if they wanted to visit friends. They had mono LCD screens, integrated file managers and the ability to double as a handheld console. A number of Dreamcast games allowed you to load extra mini-games onto the VMU. Flycast doesn't allow you to play these directly, but it can load the Dream Explorer (magpi.cc/dreamexplorer) homebrew, which allows you to manage your VMUs.



EMULATING PLAYSTATION

01 PlayStation 1 emulation

PlayStation emulation is mature, but you get different features from different emulators. If you primarily want to play homebrew, then Mednaffe – a front end for the Mednafen multi-emulator – is an excellent choice, as it upscales well. However, it doesn't include an emulated BIOS and can't play games from disc.

There are others, including PCSXR, but as we're interested in new games, we'll stick with Mednaffe here.

Open a terminal and type:

```
$ sudo apt install mednaffe
```

02 Getting an official PS1 BIOS

You can currently get a PS1 BIOS from Sony, as it's included in a PS3 BIOS that's available. However, you'll need to install the RPCS (rpcs3.net/download) emulator to open it, and that's most effectively done on an x86 PC.

Go to magpi.cc/pssystemsoftware then click the + sign next to Reinstall using a computer, then click the Download PS3 Update button.

Install and run RPCS. Select Install Firmware from its File menu and select the PS3UPDAT.PUP file you downloaded and allow it to install.

With this done, assuming you're running RPCS3 on Linux, you'll find a file called `ps1_rom.bin` in `~/.config/rpcs3/dev_flash/ps1emu/`

More detailed instructions are available at magpi.cc/psibios. To work with Mednafen, you'll have to rename the file. `ps1_rom.bin` is a region-free universal ROM, so can be renamed to reflect any of the three PlayStation region. Just copy it

three times, and name the copies `scph5500.bin`, `scph5501.bin` and `scph5502.bin`.

03 PS1 Homebrew

The original Net Yaroze development kit is still the gold standard for homebrew making on PlayStation, and you'll find plenty of games and demos to download at magpi.cc/netyaroze, as well as a few at PSX Palace (magpi.cc/psxpalace). Download Kaiga's action-RPG *Magic Castle* from magpi.cc/magiccastle – we'll use this in the next step, so unzip it in a suitable directory – ours is in `~/Games/PS1/Magic_Castle`.

04 Configuring Mednaffe for PS1

First, you'll need to copy your renamed BIOS files over to Raspberry Pi and put it into `~/.mednafen/firmware/` – you may need to run Mednaffe once to create this directory. Start Mednaffe from the menu or a terminal window.

Before we play, we'll need to configure a few things:

Select the Global Setting tab, then Sound, and set the driver to sdl. Go to Systems, select the Input, choose Sony PlayStation and then select the Input tab from the pane on the right. Select each original PlayStation controller button and map a keyboard or controller key to it.

Finally, open the File menu > Open, browse to `Magic_Castle_2021_07_May.cue` and open it. Note that Mednaffe uses bin/cue file pairs rather than iso images. Press **ALT+ENTER** to fullscreen your game, and enjoy *Magic Castle*.

Top Tip

PlayStation 2

PlayStation 2 games are playable using the Play! emulator (purei.org) but performance on Raspberry Pi 5 is limited for many titles.

▼ Recalbox automatically configures its PlayStation emulator for your controller and gives you a nice on-screen bezel, shown here around free action RPG *Magic Castle*



EASY PLAYSTATION EMULATION WITH RECALBOX

While running emulators from your desktop gives you more control over your downloads and settings, there's nothing like a dedicated emulation distro for convenience. Recalbox was first off the mark with dedicated support for Raspberry Pi 5. At the time of writing, Recalbox 9.2-Experimental for Raspberry Pi 5 wasn't yet available for direct installation from within Raspberry Pi Imager.

Download the image file manually from magpi.cc/recalboxrpis5, open Raspberry Pi Imager, and from "Choose OS", scroll down and select "Use custom". Select the image file you just downloaded and the Storage menu to select your microSD card, click "write", have a cup of tea, then pop the microSD card into Raspberry Pi 5. Recalbox supports external storage, but if you want a self-contained unit, use a large microSD card (over 64GB), as CD games can take up a fair bit of space.

01 Essential files

Press the **START** button on your controller, go to Network Settings, enable Wi-Fi, and enter your Wi-Fi password. When connected to a network, Recalbox presents SMB shares and a web interface (accessible by default via a browser at <http://recalbox.local> if you're on Linux or <http://recalbox/> if you're on Windows). The SMB shares allow you to easily upload games and BIOS images, while the web interface gives you easy remote access to Recalbox's settings and at-a-glance views of installed games and BIOSes. The easiest way to add new BIOS images is to access <smb://recalbox.local/bios/> from another computer and copy files over. Recalbox ships with a number of freeware and emulated BIOS files, and all of the legally available BIOS files for various systems

that we've discussed here and in previous tutorials such as our Emulate Everything guide in issue 133 (magpi.cc/133) should also be compatible, making this an excellent all-in-one emulation solution.

02 Add your games collection

Recalbox ships with a variety of open-source and freeware games, and you'll find a wide range of additional platforms ready to go. After copying your game and BIOS ROMs over, you'll want to reboot to force Recalbox to re-scan everything. You can always exit a game by pressing the controller hotkey and **START** buttons.

03 You get different features from different emulators

03 Add more storage

Connect an external drive and boot Recalbox, press **START** and go to System Settings. Go down to the Storage Device entry and select your USB stick. Recalbox will reboot and create a `/recalbox` directory tree on the device for your files. **M**



Top Tip

Music, maestro

If you're using a Sony DualShock 4 controller, it can end up being the default audio device. In Recalbox, fix that in Settings > Sound Settings > Output Device.

You can configure Recalbox to use a USB drive as its main hard disk for game and BIOS storage



RETRO GAMING WITH RASPBERRY PI

3RD EDITION

Retro Gaming with Raspberry Pi shows you how to set up Raspberry Pi 5 to play a new generation of classic games. Build your gaming console and full-size arcade cabinet, install emulation software and download original games with our step-by-step guides. You'll discover a vibrant homebrew scene packed with new games for original consoles and legal access to all those retro games you remember!

- *Set up Raspberry Pi for retro gaming*
- *Emulate classic computers and consoles*
- *Learn to code retro-style games*
- *Build a console, handheld, and full-size arcade machine*



BUY ONLINE: magpi.cc/store



Alasdair Allan

MAKER

Alasdair Allan is a scientist, author, hacker and journalist. In the past he has mesh networked the Moscone Center, caused a US Senate hearing, and contributed to the detection of what was – at the time – the most distant object yet discovered in the Universe.

magpi.cc/docs

Real Time Clock (RTC)

Using the real time clock functionality in Raspberry Pi 5

Raspberry Pi 5 includes an RTC module. This can be battery powered via the J5 (BAT) connector on the board located to the right of the USB-C power connector.

You can set a wake alarm which will switch the board to a very low-power state (approximately 3mA). When the alarm time is reached, the board will power back on. This can be useful for periodic jobs like time-lapse imagery.

To support the low-power mode for wake alarms you should edit the bootloader config:

```
$ sudo -E rpi-eeprom-config --edit
```

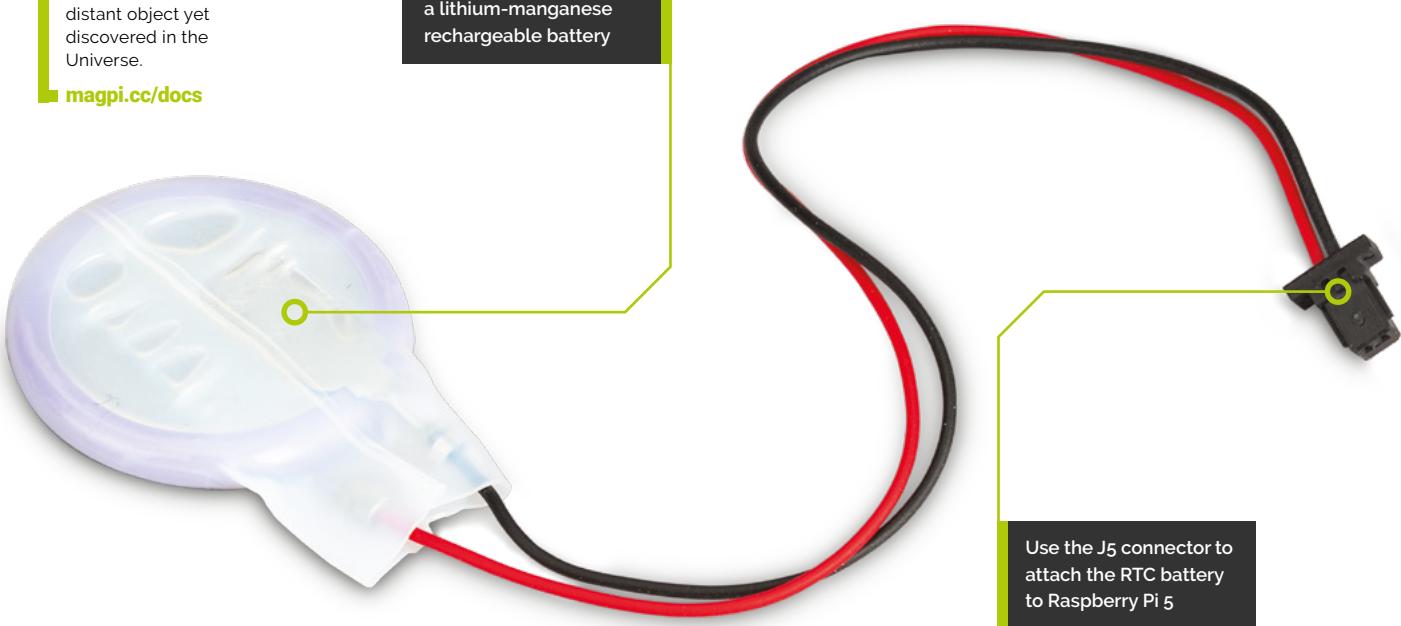
... adding the following two lines.

```
POWER_OFF_ON_HALT=1  
WAKE_ON_GPIO=0
```

Reboot Raspberry Pi so the updated eeprom takes effect:

```
$ sudo reboot
```

The RTC battery features a lithium-manganese rechargeable battery

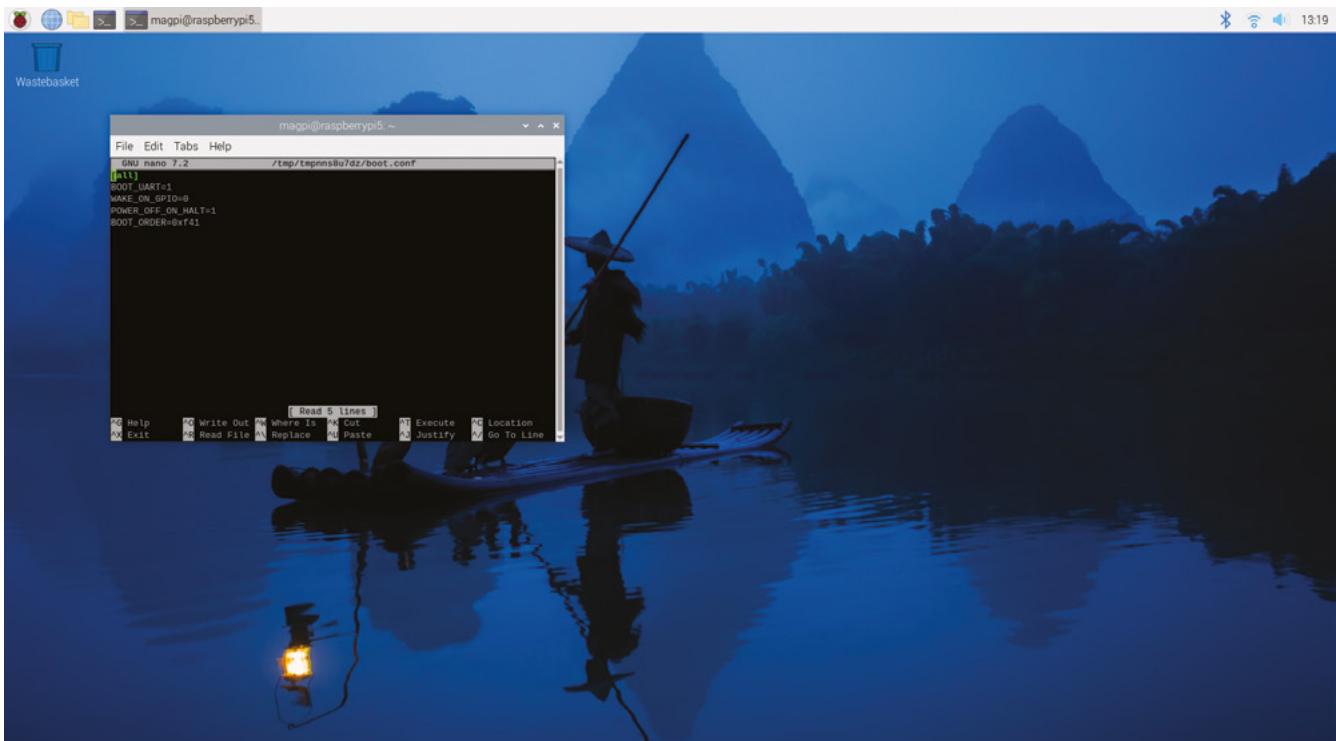


Warning!

Lithium batteries

We do not recommend using a primary lithium cell for the RTC, as it has a trickle charge circuit which is disabled by default. If enabled, this will kill the cell quickly.

magpi.cc/batterysafety



You can test the functionality with:

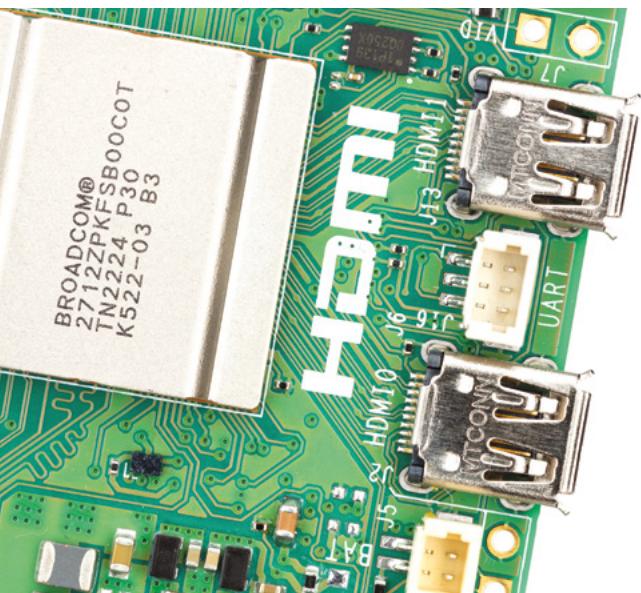
```
$ echo +600 | sudo tee /sys/class/rtc/rtc0/wakealarm
$ sudo halt
```

That will halt the board into a very low-power state, then wake and restart after 10 minutes.

The RTC also provides the time on boot e.g. in dmesg:

```
[ 1.295799] rpi-rtc soc:rpi_rtc: setting system clock to 2023-08-16T15:58:50 UTC
```

▼ The J5 connection (bottom) is used to connect RTC Battery to Raspberry Pi 5



(1692201530)

...which may be useful if you don't have an internet connection for getting time through NTP.

Adding a backup battery

The official battery part is a rechargeable lithium manganese coin cell, with a pre-fitted two-pin JST plug and an adhesive mounting pad. This is suitable for powering the Raspberry

" You can set a wake alarm which will switch the board to a low-power state "

Pi 5 RTC when the main power supply for the board is disconnected, with a power-off current draw measuring in single-digit μ A, giving a retention time that can be measured in months.

Enabling trickle charging

Trickle charging of the battery is disabled by default. There are sysfs files that show the current trickle charging voltage and limits:

```
/sys/devices/platform/soc/soc:rpi_rtc/rtc0/charging_voltage:0
/sys/devices/platform/soc/soc:rpi_rtc/rtc/
```

▲ Edit the eeprom configuration file to enable RTC support

You'll Need

- ▶ Raspberry Pi 5
magpi.cc/raspberrypi5
- ▶ Raspberry Pi RTC Battery
magpi.cc/rtcbattery
- ▶ Raspberry Pi OS

- The J5 socket may have a small plastic cover. Remove this to attach the RTC Battery to Raspberry Pi 5

```
rtc0/charging_
voltage_max:4400000
/sys/devices/
platform/soc/soc:rpi_
rtc/rtc0/charging_
voltage_min:1300000
```

You can check these values with `cat`, for example:

```
cat /sys/devices/platform/
soc/soc:rpi_rtc/rtc/rtc0/
charging_voltage
```

...returns: “0”, the current charging value. If you add `rtc_bbattery_vchg` to `/boot/firmware/config.txt` using:

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

... and add:

```
dtparam=rtc_bbattery_vchg=3000000
```

...and reboot. Use `cat` to investigate the values and you'll see:

```
/sys/devices/platform/soc/soc:rpi_rtc/rtc/
rtc0/charging_voltage:3000000
/sys/devices/platform/soc/soc:rpi_rtc/rtc/
rtc0/charging_voltage_max:4400000
/sys/devices/platform/soc/soc:rpi_rtc/rtc/
rtc0/charging_voltage_min:1300000
```

The battery will be trickle charging. Remove the `dtparam` line from `config.txt` to stop the trickle charging. ☺

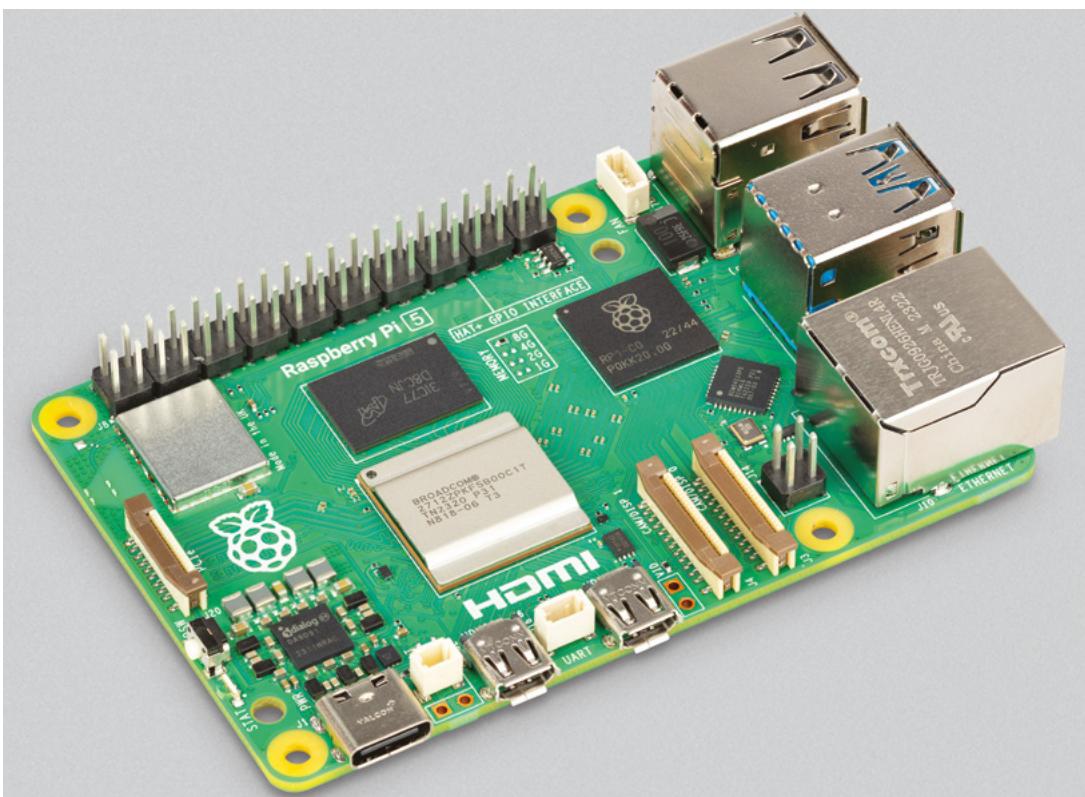
Top Tip



No battery?

The RTC is still usable even when there is no backup battery attached to the J5 connector.

- The RTC Battery connects to the UART connector sitting between the two micro-HDMI ports





TURNING IDEAS INTO REALITY

PCBWAY EMPOWERS
INNOVATORS WORLDWIDE

Join the **PCBWay revolution**
today and turn your
imagination into reality!

WWW.PCBWAY.COM

SERVICE@PCBWAY.COM



ENGINEERED
TO PERFECTION

Our state-of-the-art facilities and expert team of engineers ensure that your designs are brought to life with the utmost precision and efficiency, from a small-scale prototype to a large-scale production run.



QUALITY
ASSURANCE

Illustrate PCBWay's rigorous quality control processes, emphasizing that every product undergoes meticulous testing to ensure it meets the highest industry standards.



GLOBAL
REACH

Highlight PCBWay's extensive network, reaching innovators across the globe. Mention our fast and reliable international shipping options.



CUSTOMER
SUPPORT

Emphasize PCBWay's commitment to outstanding customer service, featuring a dedicated team of experts available 24/7 to assist with any inquiries or support needs.

Part 12

Create your own PCB


Warning!
Soldering

Soldering is relatively safe, but does use a very hot soldering iron. Take care to avoid getting burned.

magpi.cc/soldering

Make your own printed circuit board to make your circuits more professional. Use the finished PCB to control simple home automation



MAKER
**Stewart
Watkiss**

Also known as Penguin Tutor. A Maker and YouTuber who loves all things Raspberry Pi and Pico. Author of Learn Electronics with Raspberry Pi.

penguintutor.com
@stewartwatkiss

You'll Need

- ▶ PCB
- ▶ Soldering iron and solder
- ▶ Bill of materials
- [magpi.cc/
picolightsbom](http://magpi.cc/picolightsbom)**

Previously, we looked at designing a printed circuit board (PCB) in KiCad. This time around, we will go from a schematic in KiCad to a printed circuit board design which can be sent off to manufacture. Add the components and solder it up to create a complete circuit. The design can be used for 12V lights, including bright light strips and model railway lighting, or the 5V COB lights used in the earlier tutorial.

Using Pico W allows wireless access, so you can control your lights using a smart phone or to integrate with home automation.

01 KiCad schematic

This is going to start with the schematic diagram from the previous tutorial. The diagram is shown in **Figure 1**. The circuit is designed and annotated but still needs some changes before turning it into a printed circuit board.

Our Raspberry Pi Pico documentation (magpi.cc/hwdesignrp2040) recommends connecting all the grounds on the Pico to ground. This should be set on pins 3, 8, 13, 18, 23, 28 and 38. All the rest of the pins should be set to unconnected, using the Add No Connect Flag from the Place menu.

02 Electrical rules checker

Next it is useful to run the electrical rules checker from the Inspect menu. This can be handy to identify pins that you've forgotten to connect, or are wired incorrectly.

Running this will bring up some errors and warnings. They will be listed in the Rules Checker

display as well as indicated by arrows on the schematic diagram.

The errors all relate to not driving power pins. This is because we are using the terminal connections to provide power and KiCad is not aware of that. To remove the three errors add a new power symbol, choosing PWR_FLAG. Add this to each of 12V, 5V and Gnd as shown in **Figure 3**.

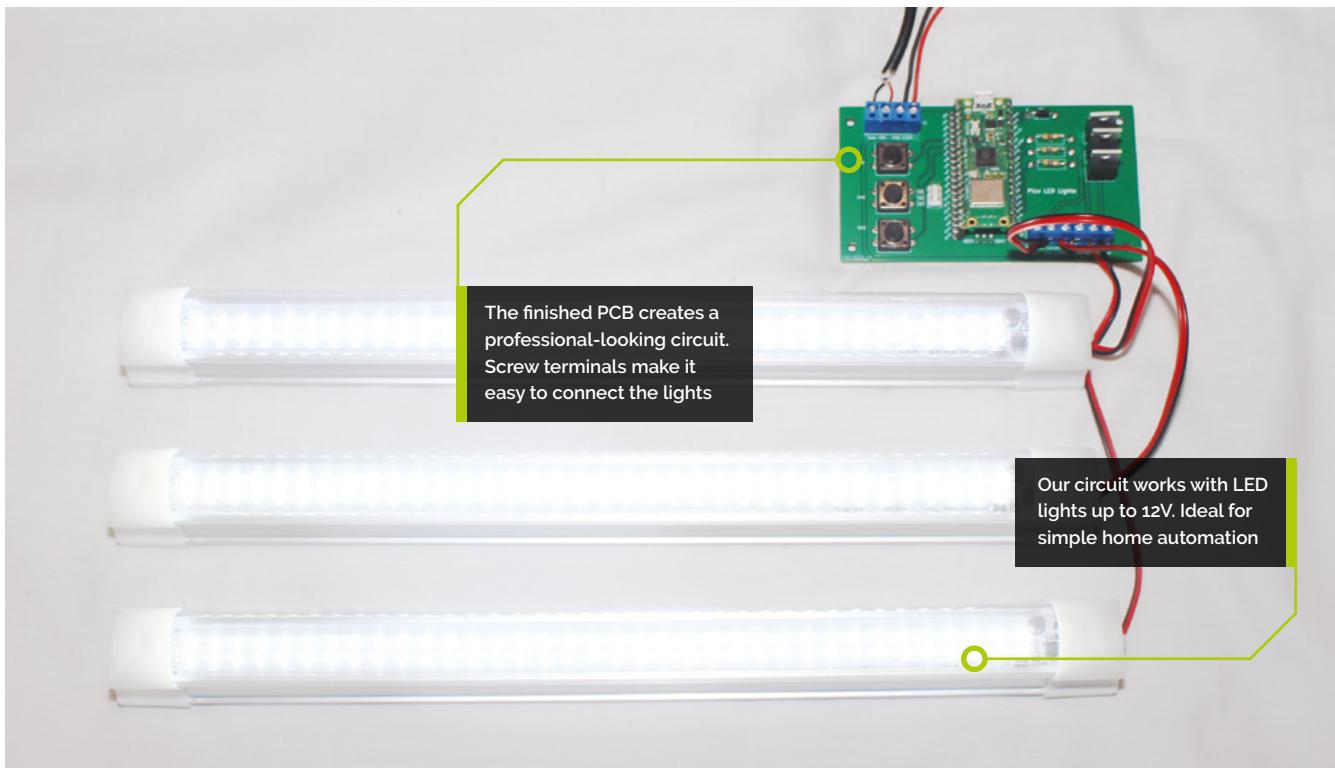
03 Update the Pico symbol

The rest of the violations are warnings which relate to some of the pins on the symbol of the Pico. For example, Pin 39 is listed as unspecified, but we are driving it as a power input. These could be safely ignored, but if you want to remove the messages edit the Pico symbol using the symbol editor.

Choose the pin table and change 38 to Power Input and 39 to Input. This is shown in **Figure 4**. Save the updated symbol and return to the schematic to run the electrical rules checker again. This will now just have the one warning saying that the symbol Pico has been modified. You can right click and exclude the violation.

04 Assigning footprints

Once the circuit passes the tests you can move on to assigning the circuit symbols to their physical symbols. This needs to be done because there are different components which need different pad positioning on the PCB. For example there are different resistor sizes and you need to tell KiCad which ones you intend to use.



Click on Assign Footprints from the Tools menu. You will see that some are already filled in based on their model number, including the MOSFETs and the diode.

For the rest, you will need to manually choose an appropriate symbol. There are three footprint filters across the top which can be useful, as well as clicking on the appropriate Footprint Libraries on the left side.

05 Choosing appropriate footprints

Then select the terminal screw connectors (labelled power input and LED outputs). Choose "TerminalBlock_Phoenix" and ensure the number of pins is selected in the footprint filters. Choose the appropriate terminal block based on the correct spacing, for example 5mm matches TerminalBlock_Phoenix_MKDS-1,5-4_1x04_P5.00mm_Horizontal. To ensure it looks correct, right click, and use View Selected Footprint. If you have installed the 3D package then you can also click on the 3D icon on the top of the view to see how it looks in 3D. See **Figure 5**. Double click on the footprint to associate it with the circuit symbol.

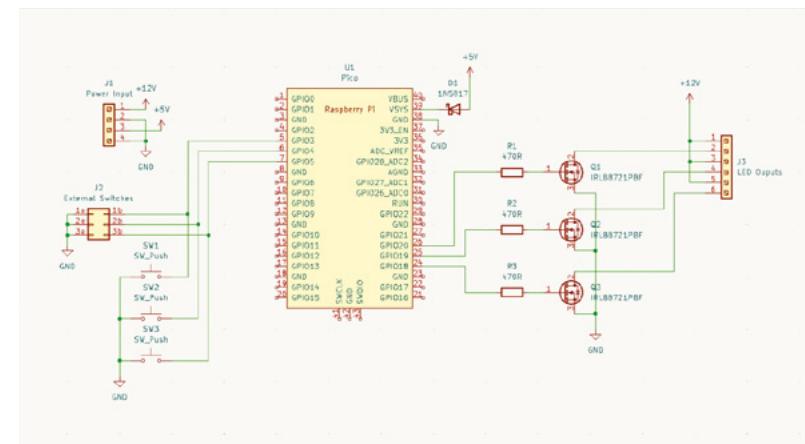
For the switch connection block use the library Connector_PinHeader_2.54mm (the same distance as the pins on a breadboard) and choose the one with 2x03 Vertical.

06 Different footprint options

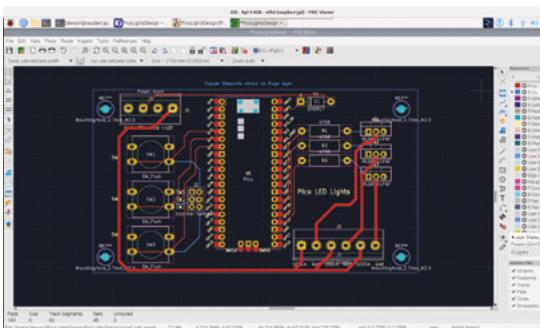
For the resistors then there are number of different resistor sizes under the Resistor_THT category (through-hole-technology). As well as representing different sized resistors they provide for different lengths of wires which can make it easier to solder. A typical 0.5W resistor has dimensions of approximately 9.9×3.6mm. Under Axial L.9.9mm D.3.6mm you can choose the one with P15.24mm which allows additional space for the resistor legs. Look at the 3D image to see how the resistor legs are bent a short distance from the resistor body.

The final components are the push buttons which you can use SW_PUSH-12mm.

▼ Figure 1. Here is our schematic diagram from the previous tutorial. The diagram shows the logical circuit design and how each of the components connect together



► **Figure 2.** The PCB design. The same components as the schematic diagram are instead positioned as they will on the final PCB



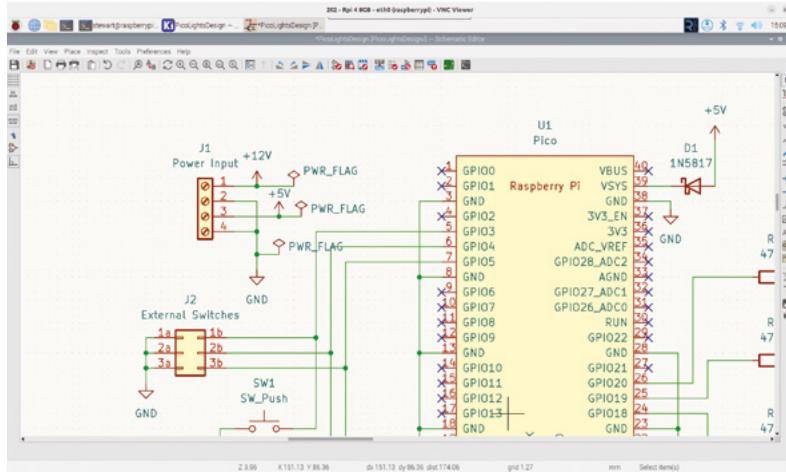
Top Tip



Different component packages

Electronic components are often available in different packages and sizes. Check that you use the KiCAD footprint to match your components

▼ **Figure 3.** Adding PWR_FLAG symbols will fix the errors created from the electrical rules checker



07 Adding the footprint library

The footprint for the RPi_Pico is listed but cannot be used yet as it has not yet been added to the footprint libraries. From the Preferences menu choose Manage Footprint Libraries and add the kicad mod file **RPi_Pico_SMD_TH.kicad_mod**, which was downloaded in the previous tutorial.

You can check that it is loaded using the footprint viewer. There is no 3D image for either the 12mm push buttons or the Pico footprint, so if you choose 3D viewer it will just show the position of the copper pads.

08 PCB board editor

After choosing the footprints run the PCB editor, which can be launched from within the schematic designer or by going back to the project window. From the Tools menu click Update PCB from Schematic to import all the component footprints. The components will be imported, but all positioned close together. Choose each component individually and move them around to the positions you'd like them to be on the PCB. The switches should be on the left and the MOSFETs with their resistors on the right near to the GPIO pins assigned. Usually, you will want to put the components close together to reduce the size of

the PCB, but not too close that it makes routing wires more difficult. A suggested layout is shown in **Figure 6**.

09 Adding mounting holes

Before starting routing it is useful to ensure that all features are included in the layout. The one thing that is useful for many PCBs is to add mounting holes to make it easier to mount in an enclosure. This could have been done in the schematic but it's not relevant there, so it can be added just to the PCB.

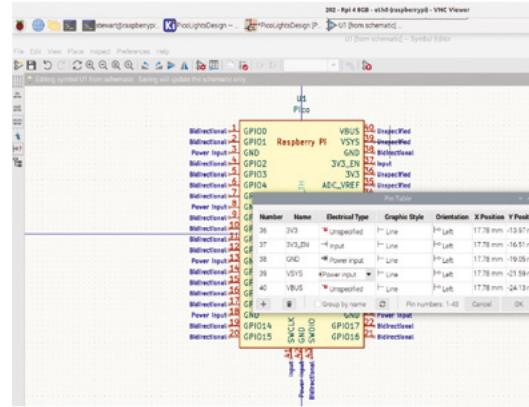
Choose Add footprint from the Place menu and search for 'mounting hole'. You can choose any appropriate size. M2.5 is a common size for PCB mounts, so you could use MountingHole_2.7mm_M2.5.

Once you have placed all the components you can draw a rectangle to show where the edge of the board will be. Choose the Edge.Cuts layer before drawing the rectangle. This is shown in **Figure 7** (see the blue arrow indicating the Edge.Cuts layer is selected).

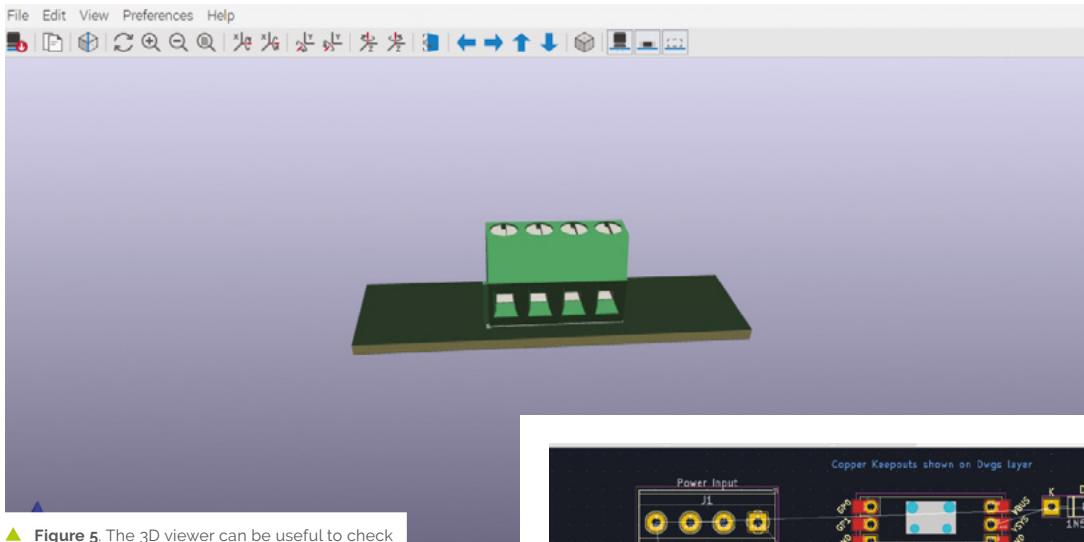
10 Laying the 12V tracks

When designing PCBs you can normally use the default track width. For this project the current from the power input could be as high as 1.8A so will need wider tracks for those. Click on the Track pull-down menu in the top left and choose Edit pre-defined sizes.

Add a track width of 0.25mm (standard size for KiCad and useful for most purposes) and one for



▲ **Figure 4.** Some of the warnings from the electrical rules checker can be fixed by selecting the correct input or output in the pin table



▲ Figure 5. The 3D viewer can be useful to check that you have selected the correct component

1mm (for high current connections). Use the pull-down menu to switch between those as required.

You can now start by routing the connections around the board. First select the top copper layer (F.Cu) and start drawing the wide tracks for the 12V power connections and connections to the MOSFET drain. Ignore Gnd as that will be done last.

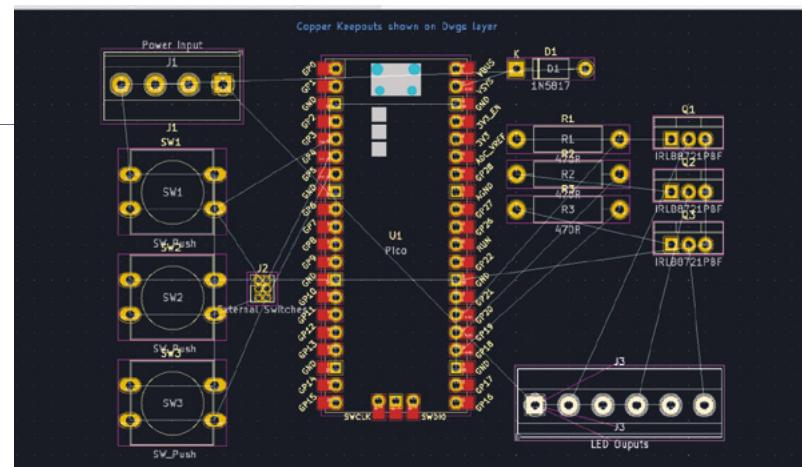
11 Completing the tracks

Switch to the 0.25mm track size and continue laying out the rest of the tracks. KiCad shows the required connections (called nets) using thin lines. You may find it useful to rotate some of the components if that will make routing easier. If it is not possible to run a connection without crossing another track then you can switch to the back copper layer B.Cu. The front layers are coloured red and the back layers are blue to make them easier to identify.

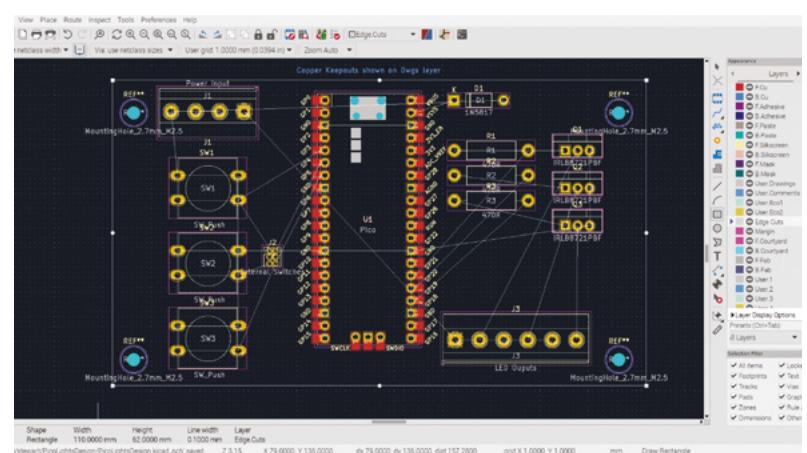
The ground connections can be handled using a ground fill which can be placed on the front and back layers. This will fill all the unused space on the board with copper and connect it to ground, which can help reduce electrical noise. Use Add Filled Zone and draw a rectangle along the entire PCB enabling both layers and GND. There will be small lines around the PCB to indicate there is a copper fill.

12 Inspect the PCB

Choose Fill all zones from the Edit menu to connect the ground connections to the ground plane. You can turn the display of the fill on and



▼ Figure 6. Suggested initial layout for the components. The final positions can be adjusted later



▼ Figure 7. The Edge Cut layer is used to mark where the edge of the PCB will be cut

off using the ‘show filled zone’ and ‘show zone’ boundaries icons on the left.

The status bar on the bottom should now show ‘unrouted’ as 0 to indicate that all the appropriate pins are connected. If not, then add any missing connections and re-run Fill all zones to update.

You can then run the Design Rules Checker to test the PCB for potential errors. There will be warnings about drilled holes being co-located (which relates

Top Tip**Check the PCB**

It is worth taking time to check all the routing and component positioning on the PCB. This can help reduce delays and the cost later

to the mounting holes) and silkscreen being clipped by solder mask. These messages can be ignored. Any warnings about silkscreen overlap can be fixed by re-arranging the labels from the components.

13 Labels and final checks

The final part of the board design is to add any additional text or labels which should normally be on the F.Silkscreen layer. Only the text shown on that layer (yellow) will be printed on the PCB.

As a final test you should also print a paper copy of the PCB to check that the components fit correctly. At a minimum, print the front Silkscreen layer and compare it with the size of your components.

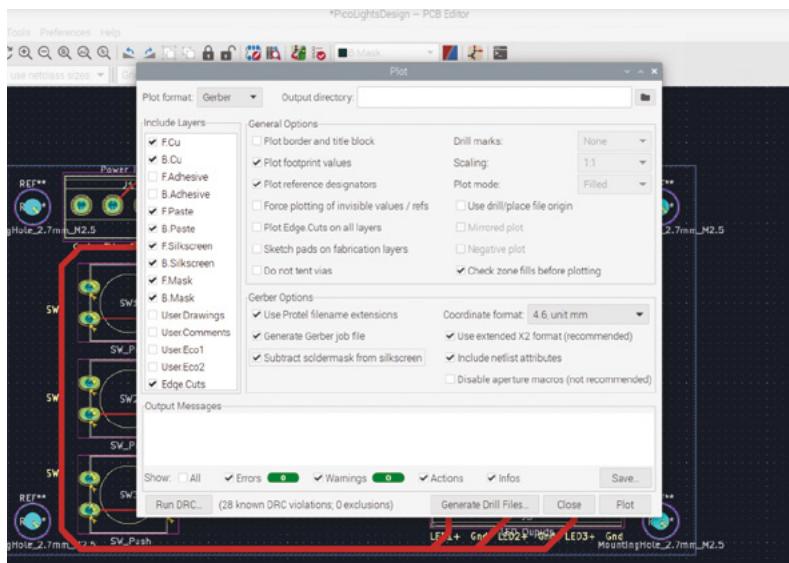
14 Export for manufacturer

The final stage in KiCAD is to export the PCB and send it to a manufacturer. The exact process depends upon your PCB manufacturer, the example used here is for JLCPCB and the process is explained here: magpi.cc/jlcpcbprocess.

Choose Fabrication Outputs and Gerber. Set Output directory to **picolights-gerbers**. Select Protel filename extensions and Subtract soldermask from silkscreen (this will fix some of the warnings in the design rules checker). Then click plot. See **Figure 8**.

In the same dialog box, click Generate Drill Files. Choose Alternate drill mode for Oval Holes. Use mm

Figure 8 Use the plot option to export ready for manufacture. You may need to adjust settings for different manufacturers



for drill units. Then Generate Drill File and then Generate Map File.

You can then use the Gerber Viewer to check the files. Look to ensure that all the layers line up correctly and that the outline is shown without any breaks.

Finally you can zip the file, submit it to your chosen manufacturer, and wait for the results.

15 The PCB arrives

Depending upon your the manufacturer and shipping times it may take a week or two before you get the PCB back. This should provide time to buy any parts you need ready to solder it together when it arrives. A link to the bill of materials (BOM) is included, which lists all the components. You will also need a soldering iron and some solder. The list is based on 12V light strips, but you could also use the 5V COB LED strips and appropriate power supply which were used in the previous tutorial.

16 Soldering

The components listed are all through hole components which should make soldering fairly easy. For best results, hold the soldering iron so that it is in contact with both the pad on the PCB and the lead of the component (see **Figure 9**). Then, feed the solder in: it will melt and make a metal joint between the component and the PCB. After removing the soldering iron it should cool down quickly to form a permanent electrical connection. It's usually easier to start with the smaller components, such as resistors, as they fit flush to the PCB. If a component won't stay still then you can use Blu Tack (sticky putty) or something similar to hold the component in place whilst it is soldered.

17 Connecting to the lights

Once soldered, mount Pico into the headers with the USB connector towards the top of the board. Then connect suitable power supplies to the terminals at the top. The 12V connection may be a different voltage if using LED lights designed for other voltages. The 5V supply is used for powering Pico.

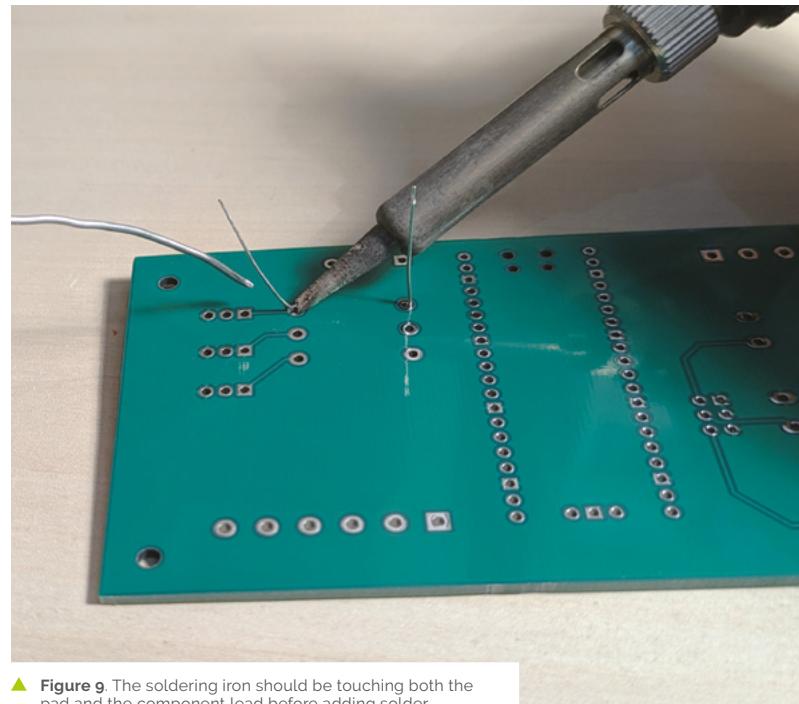
Connect an LED light to each of the three outputs at the bottom of the PCB and it just needs the code to be uploaded to your Pico.

18 Upload the code

The code is provided for a basic web server which needs to be installed on a Pico W. You will need to use the network-enabled version of the UF2 file, which can be installed using the installer within the Thonny editor. The code file is named **web-lights.py**, which can be saved on your Pico as **main.py** to run automatically. A file called **secrets.py** also needs to be saved onto your Pico with entries for the SSID and PASSWORD for your local Wi-Fi network.

When the code is running, you can connect to the web server using a web browser. You will need the IP address of the Pico which you can see through the Thonny console or by looking at the address allocation on your Wi-Fi router.

As well as using the web interface you can also turn the lights on and off using the three button switches. 



▲ Figure 9. The soldering iron should be touching both the pad and the component lead before adding solder

web-lights.py

► Language: MicroPython

**DOWNLOAD
THE FULL CODE:**

 magpi.cc/picolights

```

001. from machine import Pin
002. from utime import sleep
003. import network
004. import socket
005. import uasyncio as asyncio
006. import secrets
007. import re
008.
009. # Indexed at 0 (board labelling is 1)
010. # These must be the same length (ie 3)
011. outputs = (20, 19, 18)
012. switches = (3, 4, 5)
013.
014. # shortened version for the pin objects
015. out = []
016. sw = []
017.
018. html = """<!DOCTYPE html>
019. <html>
020. <head> <title>Pico Lights Plus</title> </head>
021. <body> <h1>Pico Lights Plus</h1>
022. <ul>
023. <li><a href="/lights?light=1&action=toggle">LED
024. 1</a></li>
025. <li><a href="/lights?light=2&action=toggle">LED
026. 2</a></li>
027. <li><a href="/lights?light=3&action=toggle">LED
028. 3</a></li>
029. </ul>
030. </body>
031. </html>
032. """
033. # Uses out to toggle - sets led to same
034. def toggle_out (pin):
035.     new_state = 1 - out[pin].value()
036.     print ("Setting out {} to {}".format(pin,
037. new_state))
038.     out[pin].value(new_state)
039.
040. def connect():
041.     #Connect to WLAN
042.     wlan = network.WLAN(network.STA_IF)
043.     wlan.active(True)
044.     wlan.config(pm = 0xa11140) # Disable power
saving mode
045.     wlan.connect(secrets.SSID, secrets.PASSWORD)
046.     while wlan.isconnected() == False:
047.         print('Waiting for connection...')
048.         sleep(1)
049.     ip = wlan.ifconfig()[0]
050.     print(f'Connected on {ip}')
051.     return ip

```

web-lights.py cont...

► Language: **MicroPython**

```

052.
053. def setup_pins():
054.     # uses length of outputs
055.     for i in range (0, len(outputs)):
056.         out.append(Pin(outputs[i], Pin.OUT))
057.         sw.append(Pin(switches[i], Pin.IN,
058.             Pin.PULL_UP))
059.
060.     async def serve_client(reader, writer):
061.         print("Client connected")
062.         request_line = await reader.readline()
063.         print("Request:", request_line)
064.         # We are not interested in HTTP request
065.         # headers, skip them
066.         while await reader.readline() != b"\r\n":
067.             pass
068.
069.         # Regular expressing Looking for toggle
070.         # request
071.         m = re.search ('light=(\d)&action=toggle',
072.             request_line)
073.         if m != None:
074.             led_selected = int(m.group(1))-1
075.             # check valid number
076.             if (led_selected >=0 and led_selected <
077.                 2) :
078.                 print ("LED selected "+str(
079.                     led_selected))
080.                 toggle_out (led_selected)
081.
082.                 writer.write('HTTP/1.0 200 OK\r\nContent-
083. type: text/html\r\n\r\n')
084.                 writer.write(html)
085.
086.                 await writer.drain()
087.                 await writer.wait_closed()
088.                 print("Client disconnected")
089.
090.                 # Initialise Wifi
091.                 async def main():
092.                     setup_pins()
093.                     print ("Connecting to network")
094.                     try:
095.                         ip = connect()
096.                     except KeyboardInterrupt:
097.                         machine.reset()
098.                         print ("IP address", ip)
099.                         asyncio.create_task(asyncio.start_server(
100.                             serve_client, "0.0.0.0", 80))
101.                         print ("Web server listening on", ip)
102.
103.                         while True:
104.                             #onboard.on()
105.                             # Enable following line for heartbeat
106.                             # debug messages
107.                             #print ("heartbeat")
108.                             await asyncio.sleep(0.25)
109.                             # Check gpio pins 10 times between
110.                             # checks for webpage (5 secs)
111.                             for i in range (0, 5):
112.                                 check_gpio_buttons()
113.
114.                             # Main loop
115.                             def check_gpio_buttons():
116.                                 for i in range (0, len(sw)):
117.                                     if sw[i].value() == 0:
118.                                         toggle_out(i)
119.                                         sleep (0.5)
120.
121.                             if __name__ == '__main__':
122.                                 try:
123.                                     asyncio.run(main())
124.                                 finally:
125.                                     asyncio.new_event_loop()
126.                                     except KeyboardInterrupt:
127.                                         machine.reset()
128.                                         print ("IP address", ip)
129.                                         asyncio.create_task(asyncio.start_server(
130.                                             serve_client, "0.0.0.0", 80))
131.                                         print ("Web server listening on", ip)
132.
133.                                         while True:
134.                                             #onboard.on()
135.                                             # Enable following line for heartbeat
136.                                             # debug messages
137.                                             #print ("heartbeat")
138.                                             await asyncio.sleep(0.25)
139.                                             # Check gpio pins 10 times between
140.                                             # checks for webpage (5 secs)
141.                                             for i in range (0, 5):
142.                                                 check_gpio_buttons()
143.
144.                                             # Main loop
145.                                             def check_gpio_buttons():
146.                                                 for i in range (0, len(sw)):
147.                                                     if sw[i].value() == 0:
148.                                                         toggle_out(i)
149.                                                         sleep (0.5)
150.
151.                                                 if __name__ == '__main__':
152.                                                     try:
153.                                                         asyncio.run(main())
154.                                                     finally:
155.                                                         asyncio.new_event_loop()

```

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE
FOR THE MODERN MAKER



ISSUE #75

OUT NOW

hsmag.cc



Part 02

Beepy: Play games on your palmtop computer

Who needs Candy Crush when you can play Colossal Cave Adventure with a full keyboard?



MAKER

**K.G.
Orphanides**

KG is a developer and writer who has been waiting over a decade to play text adventures at the bus stop without having to fiddle with an on-screen keyboard.

twoot.space/@owlbear

You'll Need

- ▶ SQFMI Beepy kit or equivalent
beepy.sqfmi.com

- ▶ 3D printed case (optional)
magpi.cc/slimflat

Last tutorial, we set up our handheld computer with the basics, including a word processor, spreadsheet, encrypted instant messaging and more. That's great, but handheld devices can be idle distractions and sources of stimulation, too. While it's not going to replace either the Game Boy or the Steam Deck, Beepy is extraordinarily well suited to keyboard-driven activities, from parser text games to extremely online social media posting. This month, we'll look at books, games and other distractions for Beepy. Remember that you can turn **ESWD** into arrow keys by pressing the mod key (the touch button in the middle of the bar below Beepy's screen), or activate the touchpad itself by double-tapping it.

01 Read a book

Terminal ebook reader **epy** (magpi.cc/epy) supports the EPUB, EPUB3, FB2, MOBI, AZW and AZW3 formats. Last month, we used Python's pip package management system within a venv to install a Python 3 application, matrix-commander, and keep it discrete from other Python applications on our system. This time, we'll use the pipx installer, which automates a lot of this for you. Via an SSH session, type:

```
$ pipx install epy-reader
```

Make a directory for your ebooks, and grab a public domain epub to read – we'll use a terminal browser to download a book from Project

Gutenberg, which doesn't allow wget or curl access to individual books.

```
$ mkdir Ebooks
$ cd Ebooks
$ w3m https://www.gutenberg.org/ebooks/11.epub.noimages
```

Once the file has been downloaded, press **Q** to quit, then we'll rename it:

```
$ mv pg11.epub AlicesAdventuresInWonderland-NoImages.epub
$ epy AlicesAdventuresInWonderland-NoImages.epub
```

Navigate the book with **EWSD**, press **Q** to quit.

02 Toot on Mastodon

If you thought building a handheld PC might help you ditch your social media habit, you're out of luck, but interacting with it in the terminal might at least curb your worst excesses. You'll need to create an account on a Mastodon server before you begin this step. If you don't have one, there's a list at joinmastodon.org/servers. SSH to Beepy and then type:

```
$ sudo apt install toot
$ toot login
```

Say No when asked if you'd like to open the authentication link in your default browser.



03 Authenticate and toot

Instead, copy the supplied login URL from your SSH terminal, open it in a browser, make sure you're logged into the correct account, and click Authorize.

Copy the authorisation code, return to your SSH session to Beepy and paste in the code.

You should see a message telling you that you've successfully logged in.

Now, grab Beepy, open a new Tmux session if you want to keep the app open in the background, and type:

```
$ toot tui
```

Press the trackpad button to switch into ESWD cursor navigation or use vi-style HJKOL, and enjoy your very, very Usenet-like fediverse browsing experience. Quit with **SHIFT+Q**. Read the full docs at magpi.cc/tui.

04 RSS never died

Whether you use a synchronised service or just want to subscribe to a clutch of feeds locally, Newsboat (newsboat.org) is an outstanding RSS reader. We'll install and run Newsboat once to create its config directory, which by default is **.newsboat** in your home.

```
sudo apt install newsboat
newsboat
nano ~/.newsboat/urls
```

Now add the URLs of any RSS feeds you wish to subscribe to. For example, we added <https://newsboat.org/news.atom> and <https://www.raspberrypi.org/blog/feed/>. Press **CTRL+X** to save and exit and pick up Beepy.

05 All aboard the newsboat

At the terminal, type:

```
$ newsboat
```

You'll notice that your feeds are listed, but that there's nothing in them. Press **SHIFT+R** to load all feeds, or highlight an individual feed and press **R** to reload only that feed. You can

Top Tip

Find more games

The awesome-ttygames project indexes terminal games around the web. Find it at magpi.cc/awettygames



▲ While not every roguelike lends itself to Beepy's 400x240 display, Nethack and its relations work a treat

navigate by pressing the number of each feed or by using **N** (next) and **P** (previous). You can scroll through messages with the trackpad or by putting Beepy into ESWD directional key mode. Enter opens a feed or a message, while **Q** moves up a level, and quits Newsboat if you're at the top level/

Alternatively, you can create a `~/.newsboat/` config file to sync to an external RSS feed reader service, such as a self-deployed Nextcloud News installation, The Old Reader, or FreshRSS. See magpi.cc/newsboatclient to find the appropriate settings for your service.

▼ With newsboat, you can subscribe to all your favourite blogs and periodicals, download them when you have a Wi-Fi connection, and read them at your leisure



06 Get on IRC

IRC (Internet Relay Chat) is the Ur online messaging protocol, pre-dating and inspiring the likes of XMPP, Slack and Discord. It remains active, with networks including Libera.Chat, QuakeNet, Undernet and EFnet. Find more at magpi.cc/ircnetworks.

```
$ sudo apt install irssi
$ irssi
$ /network add LiberaChat
$ /server add -network LiberaChat -tls -tls_
verify irc.libera.chat 6697
$ /connect LiberaChat
$ /join #raspberrypi
```

This unofficial Raspberry Pi enthusiasts' channel tends to be quiet, but works for this example. Many servers, including Libera Chat, require you to register an account and nickname for full access to all channels, particularly busier general `#chat` channels. Libera Chat provides detailed instructions at magpi.cc/liberachatreg.

07 Terminal games

There are plenty of good (and less good, but nonetheless interesting) games that you can play in your terminal, but not all of them adapt well to Beepy's 400x240 mono display, while others are too fiddly to play with the ESWD directional key layout. Thus, our recommendations don't include the terminal port of trading simulation strategy classic *Taipan*, or roguelike RPG classic *Angband*. It's a good idea to create a games directory and ensure that the ncurses development libraries are installed. Where relevant, we'll also symlink games to `/usr/local/games`, so you don't have to type the full path to play them.

```
$ sudo apt install libncurses5-dev
$ mkdir Games
```

08 Tiny Moon Runner

The only game so far in Ben Busby's ncurses arcade (magpi.cc/ncursesarcade), it's like that T-rex cactus jumping game you can play when Chrome isn't working, and it's great. Press **SPACE** to jump over rocks as an ASCII art landscape scrolls by, or be smashed to your doom.



▲ Internet Relay Chat is long past its heyday, but remains an excellent way to chat, without the vast system overhead of Slack or Discord

```
$ cd Games
$ git clone https://github.com/benbusby/
ncurses-arcade.git
$ cd ncurses-arcade
make
./ncurses-arcade
$ sudo ln -s /home/YOUR_USER_NAME/Games/
ncurses-arcade/ncurses-arcade /usr/local/
games
```

You'll now be able to run it from anywhere by typing:

```
$ ncurses-arcade
```

09 Debian's terminal game pack

You can bulk install Debian's collection of terminal games via the games-console meta-package, but the majority don't work properly on the hardware available, most often due to Beepy's petite display size. We've tested all of them, and can recommend a number that actually work, from RPGs to procedural plant generators. Enter:

```
$ sudo apt install freesweep
```

Freesweep is an ncurses-based Minesweeper clone. All the default settings – except colour – are Beepy compatible. You can navigate with ESWD in arrow key mode, or vi-style with HJLK. Press **F** to place a flag and **SPACE** to clear a square. Quit with **Q**.

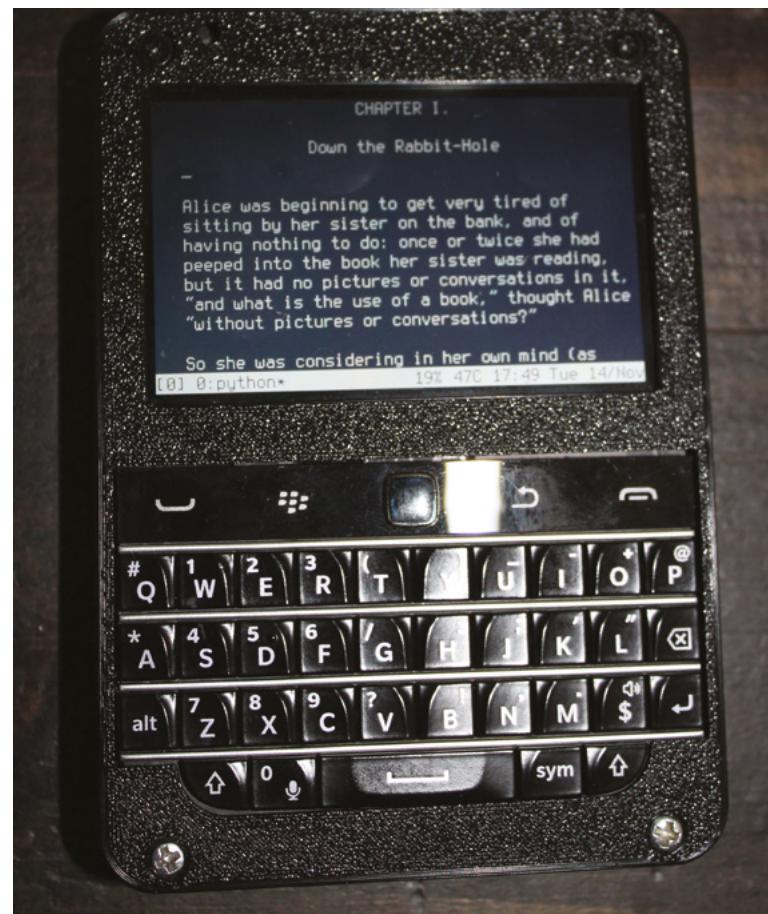
10 Dungeon delving

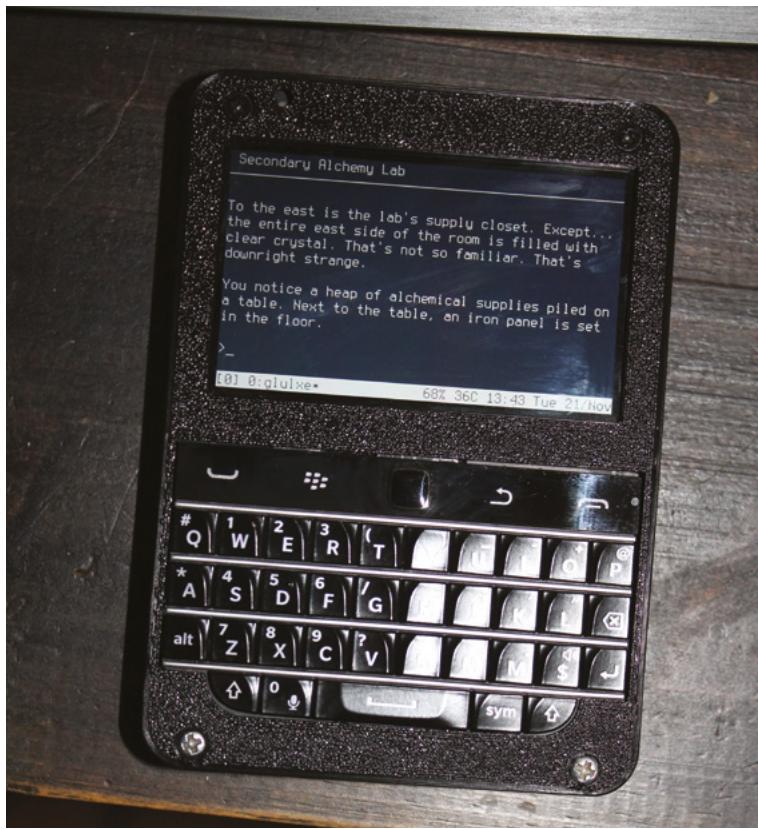
To play Nethack, enter:

```
$ sudo apt install nethack-console
```

While most roguelikes in the Debian games bundle require far too much screen real estate, this console version of *Nethack* works well, and you can control your movement with the vi keys, HJKL, although the limited number of keys means it's easiest to use # commands for tasks such as looting chests. You'll find the command reference at magpi.cc/nethackcommands.

▼ Terminal ebook reader epy allows you to carry a library with you on Beepy. You'll need to use DRM-free ebooks with it, though





▲ Support for Z-Machine and Glulx files opens up a world of interactive fiction. Seen here, Andrew Plotkin's commercial Glulx release, Hadean Lands

```
$ sudo apt install slashem
```

Nethack variant *Slashem* (which stands for Super Lotsa Added Stuff Hack – Extended Magic) is also fully functional. *Slashem* gets extra player character races, including vampires, hobbits and drow, as well as new monsters, spells and more. However, it's generally regarded as more difficult than the original, largely due to game balance issues with the new levels and monsters.

11 Classic spelunking

If you fancy a text adventure, enter:

```
$ sudo apt install colossal-cave-adventure
```

This is a faithful port of Will Crowther's early text adventure, inspired by caver Crowther's exploration of Mammoth Cave in Kentucky, and it's where familiar phrases like "you are in a little maze of twisty passages, all alike" come from. This port is pretty faithful to the 1976 version, in which you can earn a maximum of 350 points by exploring and collecting items.

```
$ sudo apt install open-adventure
```

Top Tip



Adventure online

Install *tintin++* to access multiplayer dungeons and other online text games. Learn more at magpi.cc/tintinmud.

This is another port of *Colossal Cave Adventure*, this time expanded upon by Don Woods in 1977. It has quality life features including not yelling at you in capital letters all the time. This is probably the one you want to play, as it pays less attention to the accurate reproduction of the cave complex and more to being a game. Launch it by typing **advent**. There are 430 points to be won in this iteration.

Both versions are pretty tough, so you may want some hints. Try magpi.cc/colcavehints.

12 The world of interactive fiction

The text adventures of the 1970s gave birth to a rich culture of interactive fiction. Beepy's keyboard makes it the ideal device for portable parser gaming, because you can easily type in commands such as GET LAMP. In that spirit, we're going to install Glulxe, an IF engine virtual

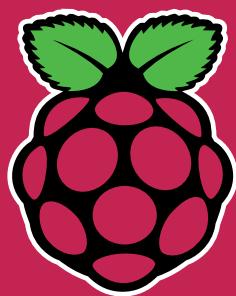
"Beepy's keyboard makes it the ideal device for portable parser gaming"

machine that lets you play parser games written in the Glulx format supported by Inform 6.30 and above, enter:

```
$ sudo apt install glulxe fizmo-ncursesw
```

We're going to install a couple of classics: the original free version of Michael Gentry's Lovecraftian horror game *Anchorhead* (magpi.cc/anchorhead), later developed for commercial release, and Emily Short's *Counterfeit Monkey*, a word-bending puzzle adventure. You can find more free games at the Interactive Fiction Database (ifdb.org).

```
$ cd Games
$ wget https://ifarchive.org/if-archive/games/zcode/anchor.z8
$ fizmo-ncurses anchor.z8
$ wget https://github.com/i7/counterfeit-monkey/releases/download/r11.1/CounterfeitMonkey-11.gblorb
$ glulxe CounterfeitMonkey-11.gblorb M
```



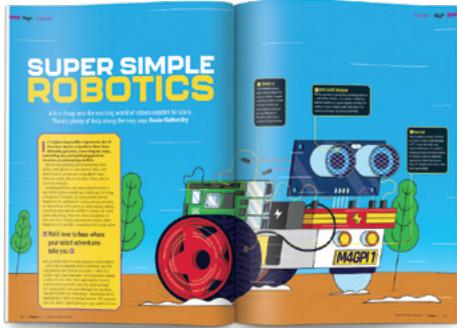
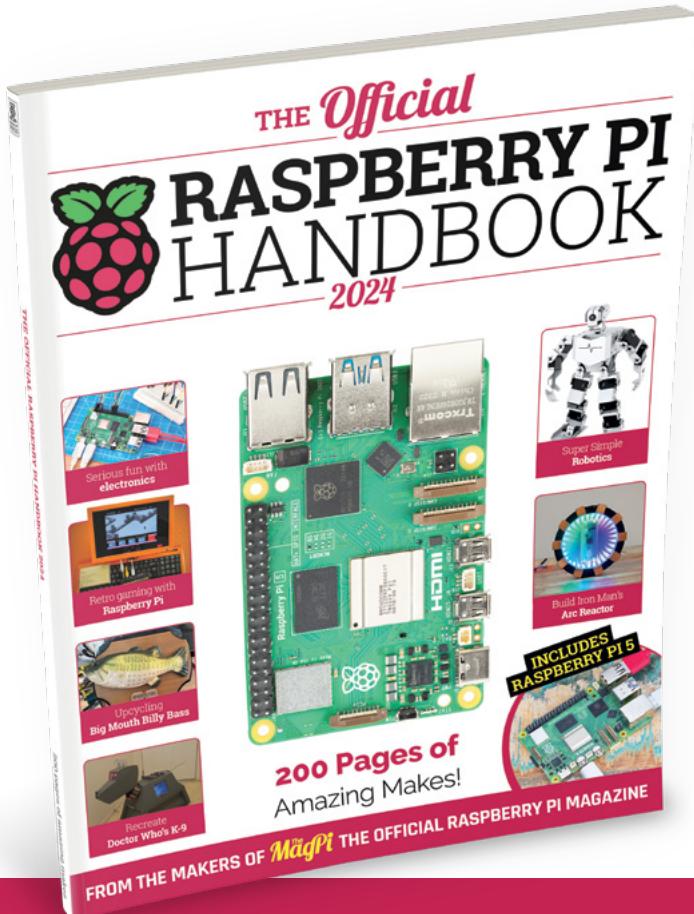
THE *Official*

RASPBERRY PI HANDBOOK

2024

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects



Buy online: magpi.cc/store

Turn an old phone into a robotic personal assistant

Bring an old phone into the 21st century by adding a Raspberry Pi to turn it into a networked assistant



Rob Miles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at robmiles.com.

Rotary dial phones are fashionable again. You can even buy brand new 'old' ones based on the original design shown in **Figure 1** below. These have dials, but not the weight or the authentic bell sound. The author was completely unaware of this trend when he picked up his red phone. The idea was to retain the external appearance and behaviours but bring the device up to date with all-new internals and some fun behaviours. It seemed to him that there should be room inside for a reasonable amount of computing power, and he was keen to hear the old telephone bell sound again. He wanted to create a web-controlled device that could be used to receive messages and alerts. The telephone that was built contains a Raspberry Pi Zero 2 running JavaScript code inside the node environment, using Express to host a telephone website. You can find all the code, 3D files for the mounting plate, and a setup sequence for a Raspberry Pi in the GitHub repository for the project here: hsmag.cc/DialTelephone.



Figure 1 ◆
The phone still looks good nearly 50 years after it was made

DELVING INTO HISTORY

The first task was to open the telephone and check the amount of space available for the new innards. **Figure 2** shows the printed circuit board (PCB) inside the phone. This design was one of the first times a PCB had been used in a UK telephone.

The author would have liked to have kept the internal components in place so that the phone could be returned to its original state if required. Unfortunately, this turned out to be impossible.

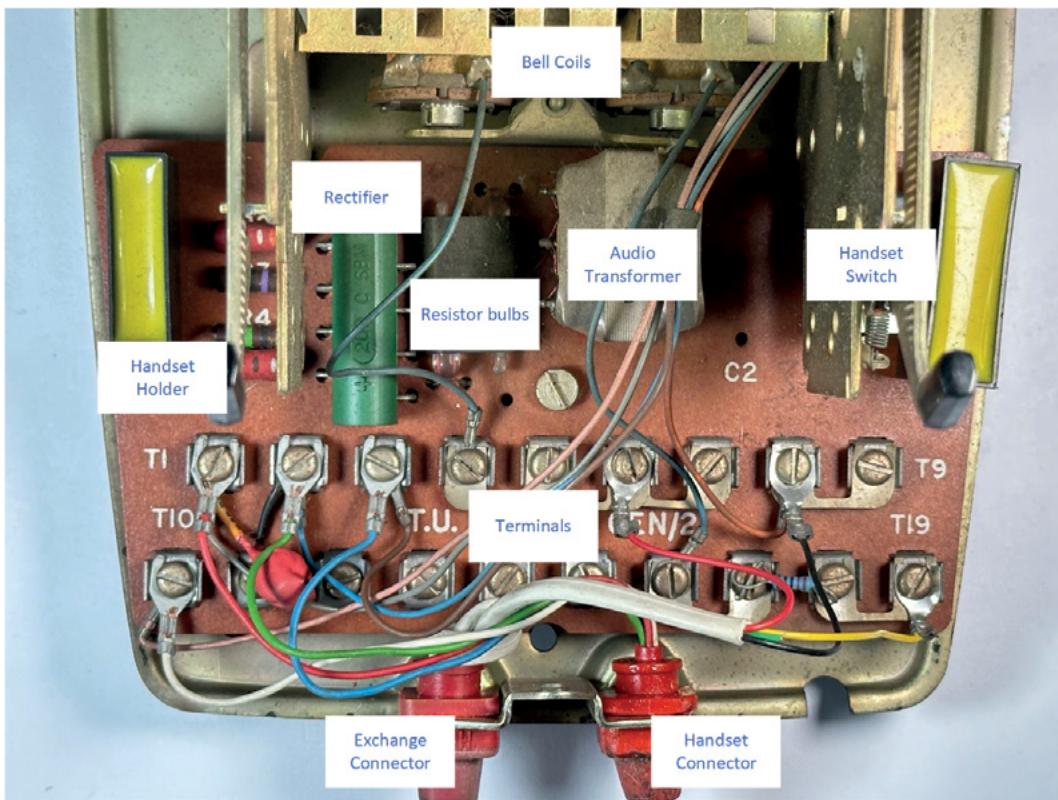
WIPING THE SLATE CLEAN

The circuit board was cleared of components, and a 3D-printed holder for the Raspberry Pi Zero with prototyping board was inserted into the space, as shown in **Figure 3**, overleaf. The phone will use a 12-volt power supply, and it was found that a power supply socket fits into the cord holder for the exchange connection with an appropriate washer.

Figure 4, overleaf, shows the circuit for the telephone. The two devices in the centre are two 'buck converters'. The one on the left converts the 12-volt power input into 35 volts to power the bell. The second converts 12 volts into 5 volts to power the Raspberry Pi. The handset switch is connected to the handset cradle and indicates whether the handset is on the phone. The dial pulse and dial active switches are in the telephone dial, of which more later.

RING THE BELL

The bell in the telephone was originally driven by two coils powered by a 75-volt alternating current signal with a frequency of around 18Hz. The coils move a bell-clapper left and right between two metal bells tuned to different musical notes. The author was very keen to retain the distinctive ring, but less keen on getting 75 volts up his armpits when assembling the phone. So, rather than using 75-volt AC, he opted to



use a much less tingly 35-volt supply, using software to drive each coil in turn. Two MOSFET controllers were used, one for each bell. These are connected to general-purpose input/output (GPIO) pins on the Raspberry Pi which are controlled by JavaScript running the phone.

```
async ding() {
    this.bell1GPIO.on();
    await this.delay(25);
    this.bell1GPIO.off();
    this.bell2GPIO.on();
    await this.delay(25);
    this.bell2GPIO.off();
    return;
}
```

A single 'ding' is produced when the handset is lifted or replaced, just like the old phones do. The JavaScript above makes the bell go 'ding' when the `ding` method is called. The code moves the clapper towards each bell in turn. The delay values of 25 milliseconds between the movements of the clapper were determined by trial and error.

```
async repeatRing(length) {
    for (let i = 0; i < length; i++) {
        await this.ding();
        if (!this.ringing) {
```

```
        return;
    }
}
```

The `repeatRing` method produces a longer ringing sound by repeatedly calling the `ding` method the requested number of times. It checks the `ringing`

TELEPHONE ETIQUETTE

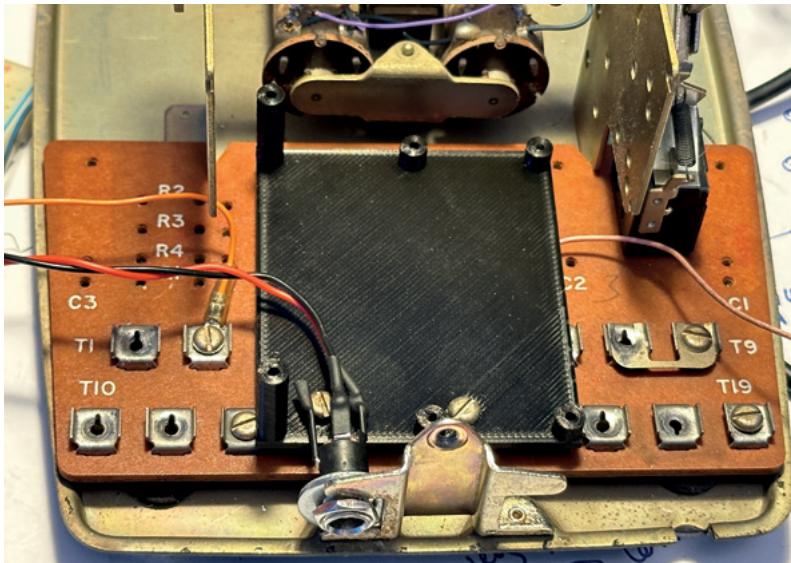
If you've only ever seen mobile phones up to now, you might be wondering how a dial phone is used. If the handset is on the phone, as seen in **Figure 1**, the phone is waiting for an incoming call. The phone is connected to a telephone exchange which makes and maintains the connections between telephones. To make a call you lift the handset, at which point the exchange produces a dial tone you can hear from the handset speaker. You use the dial to enter the digits of the number of the phone you want to ring. After you have dialled the last digit of the number, the exchange connects to the destination phone and makes it ring. When the receiver of the destination phone is picked up, the exchange connects the microphone and speakers of the phones together so that the phone users can have a conversation. When either handset is placed back on the phone, the exchange ends the call. The software inside the Raspberry Pi emulates this process.

Figure 2 ◇

The circuit sends audio signals through the resistor bulbs if it detects the phone has a connection which is close to the telephone exchange. This reduces the sound volume and makes the bulbs light up in time with your speech

YOU'LL NEED

- ◆ An old-style telephone with a dial. The author used a 746 model he picked up in a second-hand shop. The author thinks the phone should be red, like the original 'batphone', but the software will work with other colours
- ◆ A Raspberry Pi Zero 2
- ◆ 2 x power switches for the bell. The author used one with dual D4184 MOSFETs which can be driven by the GPIO signals from the Raspberry Pi
- ◆ 2 x 1N4007 1A 1000V silicon rectifier diodes
- ◆ A power converter to convert 12 volts to 35 volts for the bell
- ◆ A power converter to convert 12 volts to 5 volts for the Raspberry Pi
- ◆ A 12-volt power supply for the above
- ◆ The author used a Raspberry Pi GPIO breakout board to mount all the components
- ◆ A USB audio adapter and micro USB 'on-the-go' cable to connect it

**Figure 3**

The remaining two connections (orange wire and pink wire) are for the switch that the Raspberry Pi will use to detect when the handset is lifted. The switch that detects the handset remains on the circuit board

flag after each ding and will stop ringing if this flag becomes false.

```
async ukRing() {
    while (this.ringing) {
        await this.repeatRing(10);
        await this.delay(100);
        await this.repeatRing(10);
        await this.delay(1400);
    }
}
```

The **ukRing** method produces the characteristic ‘brring-brring’ sound of a UK telephone, which is two rings separated by a silence of around one and a half seconds. The length of each ring and the intervals between them were also determined by trial and error. In fact, the completion of this project seems to have been accomplished with a lot of trial and an awful lot of error.

MULTITHREADING WITH AWAIT

The statements in the **ding** and **ukRing** methods above use the JavaScript **await** keyword to ensure that the phone application can still respond to events (for example, the handset being lifted) while the bell is ringing. When a JavaScript program reaches an **await** in a method or function, it creates another thread to run from that point and returns, allowing the calling function to continue. In other words, in the **ding** method above, the first call of **delay** (which pauses for 25 milliseconds) will not pause execution of the program that called **ding**. Instead, the **ding** function will return to the caller at that point. The **ding** function is flagged as **async**, which means that it returns a **Promise** object to be used by the caller to trigger other actions when the ding has completed. The **Promise**

QUICK TIP

There is a useful guide to old telephones here: hsmag.cc/phoneT746. This includes some illustrations that the author thinks would look great on T-shirts.

READ THAT DIAL

The telephone dial is used to enter phone numbers. The user puts their finger into the required number hole (see **Figure 1**) and turns the dial clockwise until their finger hits the metal stop at the bottom of the dial. The dial is spring-loaded. When the user releases the dial, it turns back to its home position. As the dial rotates back to its home position, it sends a series of ‘dial pulses’ to the exchange. If a larger number is entered, the dial will rotate further. Telephone numbers contain a particular number of digits; once the last digit has been entered, the exchange will connect the call.

can be given event methods to be called when the promise is resolved (i.e. when the ding has finished). This form of multithreading is very flexible.

If it seems confusing, imagine that you could create new versions of yourself at will. If you need to queue to buy something, you could create a new version of yourself, put that in the queue, and then go about the rest of your business. At some point the ‘new you’ will reach the head of the queue, get what you want, and then call you and say it has finished before vanishing in a puff of smoke. That’s how awaits and promises work.

Figure 5, overleaf, shows the contacts inside the dial. The white plastic cog in the centre has ten teeth for each number. The plastic follower at the bottom right-hand side of the dial swings out of the way when the user is moving the dial clockwise and then engages with the metal contacts when the dial is returning, opening and closing a contact as it does. The further the dial is turned, the more teeth will hit the follower, and the more pulses will be sent. It is an ingenious piece of mechanical design. The contacts near to the centre of the cog in the centre are closed when the user moves the dial from its ‘home’ position so that the exchange can be told that the user is dialling a number.

Figure 6, overleaf, shows the signals produced by the dial when the user dials a three. The telephone program in the Raspberry Pi must read these signals to obtain this value so that it can be used to control the phone. The decoding software uses functions which are bound to events generated when the dial signals change state.

```
startDialing(){
    this.pulseCount = 0;
    this.dialing = true;
}
```

```

dialPulse(){
    if(this.dialing){
        this.pulseCount++;
    }
}

endDialing(){
    console.log('Dialed a:${this.pulseCount}');
    this.owner.numberDialed(this.pulseCount);
    this.dialing = false;
}

```

The **startDialing** method above is called when the software detects a rising edge (a change from low to high) on the 'Dial Active' signal. It sets **pulseCount** to zero and sets dialling to true, which indicates that a number is being dialled. The **dialPulse** method is called on the rising edge of the 'Dial Pulse' signal. It checks to see if a number is being dialled and increments **pulseCount** if it is. The **endDialing** method is called when there is a falling edge on the 'Dial Active' signal. This calls the owner of the dial and delivers the pulse count to the dial owner by calling the function **numberDialed**. It then turns dialling off by setting the dialling flag to false.

“The further the dial is turned, the more teeth will hit the follower, and the more pulses will be sent”

ADDING AUDIO

The Raspberry Pi Zero in the phone uses a USB audio hardware interface to produce sounds. The output is quite capable of driving the speaker in the handset. The JavaScript program uses the **play-sound** library to play sound files and the eSpeak program to convert text to speech. Presently, the phone doesn't support audio input. This is because the microphone in the telephone is implemented using a little foil box of carbon granules which change in resistance when vibrated by sounds. This change in resistance is used to drive a coil in a transformer to generate the audio signal to be sent over the phone line. This microphone cannot be connected directly to the microphone input on the USB sound interface. The author intends to investigate using the transformer removed from the phone to see if this could create a usable signal. However, the phone is still great fun to use, even if you can't speak into it just yet.

BUILDING THE PHONE

Figure 7, overleaf, shows the completed phone hardware. The two MOSFET switches are mounted on a Raspberry Pi prototyping board which is plugged into the Raspberry Pi Zero. The USB sound interface is plugged into a micro USB 'on-the-go' adapter at the back of the phone. The author was careful with the 35-volt signals, especially after he destroyed a Raspberry Pi Zero by accidentally touching one of the ringer coil terminals with a GPIO input.

PHONE SOFTWARE

The phone software is implemented as several cooperating JavaScript classes, each of which looks after one part of the phone. The **Phone** object acts as a container for these.

```

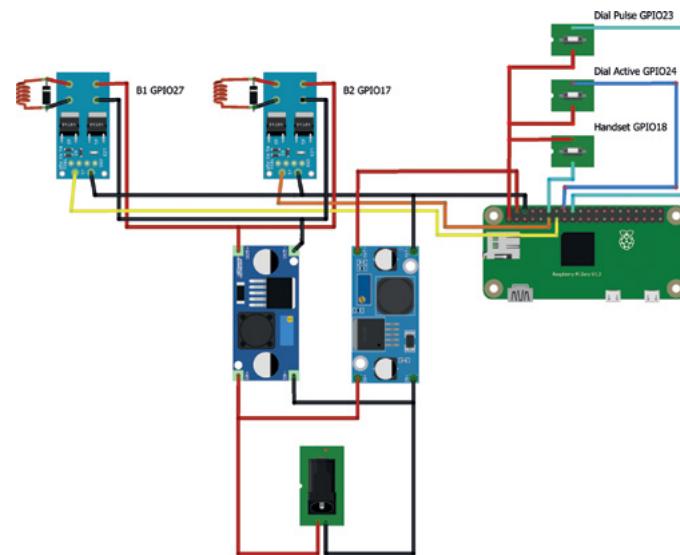
constructor(owner) {
    this.owner = owner;
    this.ringer = new Ringer();
    this.handsetSwitch = new HandsetSwitch(this);
    this.dial = new Dial(this);
    this.soundOutput = new SoundOutput(this);
    this.speech = new Speech(this);
    this.ringing = false;
    this.ringStart = null;
    this.ringLengthMillis = 10000;
    this.dialing=false;
    this.messages = null;
    setInterval(() => {
        this.update();
    }, 500);
}

```

QUICK TIP

The diodes across the bell coils protect the MOSFET switches from the reverse voltage induced in the bell coil when the bell current is turned off. The diodes are not expensive, but the author has discovered that missing them out can be.

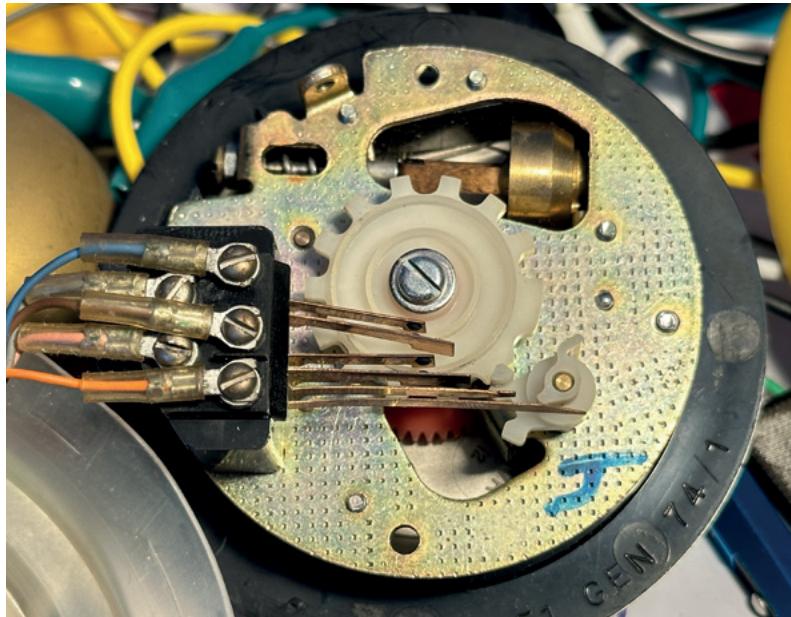
Figure 4 There is also USB audio device plugged into the Raspberry Pi to provide the sound output from the phone



QUICK TIP

The story that the first telephone dials were invented by an undertaker to stop telephone operators (people who connected telephone calls for a living) from learning about local deaths and passing on the details to his competition is not true. Almon Brown Strowger was an undertaker, but he just wanted to improve the accuracy of his telephone connections.

Figure 5 ◊
The little brass cup at the top right-hand side of the dial contains a mechanical regulator to limit the speed at which the dial turns back



The code above is the constructor for the **Phone** class. It creates all the different phone component objects and speech and sound output services, sets up some initial values, and then starts an update timer ticking which can do things such as time out the ringer. The component objects trigger actions in the phone by calling methods in the **Phone** instance. For example, the **Phone** class contains a method called **numberDialed** which is called by the **Dial** class.

```
numberDialed(number){
    if(this.handsetSwitch.handsetOffPhone()){
        this.delay(600).then(()=>
    {
        switch (number){
            case 1:
                this.startRinging();
                break;
            case 2:
                this.randomCall();
                break;
        }
    });
}
}
```

The code above shows **numberDialed**. It waits for around half a second to simulate the exchange connection delay, and then if a 1 was dialled, the phone starts ringing. If 2 was dialled, the handset makes a random call.

```
randomMessages = [
    "I know what you did last summer.",
    "Is that you, Boris?",
    "Look out of the window.",
    "They are on to you.",
    "Look behind you."
];

randomCall(){
    let messageDelayMillis = this.getRandom(2000,5000);
    this.delay(messageDelayMillis).then(()=>{
        let messageNo = this.getRandom(0,this.randomMessages.length);
        this.acceptMessage(this.randomMessages[messageNo]);
    });
}
```

It is fun to play with the telephone itself, but the phone is even more fun when controlled remotely

The **randomCall** method waits a random time between 2 and 5 seconds, picks a message from the **randomMessages** array, and then calls the **acceptMessage** to play it.

```
acceptMessage(message){
    this.message = message;
    this.startRinging();
}
```

The **acceptMessage** stores the message in the **Phone** class and then starts the phone ringing. When the handset is picked up, the message is played.

```
handsetPickedUp(){
    if(this.ringer.ringing){
        this.stopRinging();
        if(this.message){
            this.delay(1000).then(()=> {
                console.log(`Saying
message:${this.message}`);
                this.speech.say(this.message);
                this.message = null;
            });
        }
    }
}
```

**Figure 6**

The 'inout' library is used to allow a Raspberry Pi JavaScript application to interact with the GPIO pins

```
else{
    this.ringer.ding().then(()=>
    {
        this.doDial();
    });
}
```

The `handsetPickedUp` method is called when the handset is picked up. Which just goes to show how good the author is at picking names for methods. If the ringer is ringing, it turns it off and then checks to see if the phone has a message to say. If it does, the message is spoken after a short delay to give the user time to get the handset to their ear. If the ringer is not ringing when the handset is picked up, it calls the `doDial` method to start dialling a number.

```
doDial(){

    if(this.dialing){
        console.log("doDial called when already
dialing");
        return;
    }

    console.log("Dialing");

    this.dialing = true;

    this.soundOutput.playFile('dialTone');
}
```

The `doDial` method starts the dialling process. It sets a flag to indicate that the phone is dialling (having first checked that the phone is not already dialling) and then plays the dial tone sound effect.

From the code above, you can see that the phone works as a series of methods which are called in response to events and modify values held in the phone object. You can use this to make the phone respond to dialled numbers in any way you like.



More advanced phone code could assemble longer sequences of dialled numbers. The author thinks it might be fun to create a 'Phone powered' mystery game where the phone rings every now and then and invites the player to perform an action and dial responses to move through the game.

MAKING A TELEPHONE INTO A WEB SERVER

It is fun to play with the telephone itself, but the phone is even more fun when controlled remotely via the local network. The RedServer.js program hosts a website which can be used to remotely control the phone.

The Raspberry Pi in the telephone hosts the web page shown in **Figure 8**, overleaf, on your local home network. The author has set the Raspberry Pi machine name as 'theredphone' and the site is hosted on port 3000. This means the site can be found on a home network as **theredphone.local:3000**. The web server is powered by Express, a popular library for hosting web pages.

The Express library allows you to create 'routes' which are pieces of code that are executed when the client browser is used to access a particular web address.

Figure 7

The two voltage converters are mounted on holders which are then stuck to the base of the phone. You can see them on each side just above the dial

Figure 8 When ‘Send the message’ is clicked, the phone will ring. When the receiver is picked up, the phone will speak the message that was entered into the form

QUICK TIP

The code for zero generates ten dial pulses, since sending zero pulses would make the dialling process vulnerable to noise (a noise pulse on the dial active line would be interpreted as dialling the value zero). Remember that in a real telephone, these signals are sent over cables to the telephone exchange.



HackSpace

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

A screenshot of a web browser window titled 'Red Phone Personal Assistant'. The URL is 'theredphone.local:3000/sendMe...'. The page has a dark header with the title. Below it is a blue button labeled 'Ring the phone now'. Underneath is another blue button labeled 'Stop the phone ringing'. A section labeled 'Message:' contains a text input field with the text 'Mr Watson, come here. I want to see you.'. Below the input is a blue button labeled 'Send the message'. At the bottom, the text 'Message sent' is displayed. The browser's address bar shows 'Not secure | theredphone.local:3000/sendMe...' and the status bar shows 'theredphone.local:3000'.

```
app.get('/', (req, res) => {
  res.render("index.ejs", {message:''});
});
```

The code above is performed when the user browses the site. The Raspberry Pi in the telephone web server uses the **ejs** library which allows us to create web pages that contain JavaScript elements. When the root is accessed, the server displays the index page you can see in **Figure 8** with a message value of an empty string. The page layout is described in the **index.ejs** file:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Red Phone Personal Assistant</title>
</head>

<body>
  <h1 class="mb-4">Red Phone Personal Assistant</h1>
  <a href="/ring">Ring the phone now</a>
  <a href="/stopRing">Stop the phone ringing</a>
  <form action="/sendMessage" method="POST">
    <label for="email">Message:</label>
    <input type="text" id="message" name="message" required>
    <button type="submit">Send the message</button>
  </form>
  <p> <%= message %> </p>
</body>
</html>
```

It would be interesting to add speech decoding so that the phones can recognise what the user says

```
</body>
</html>
```

When the user fills in a message and clicks ‘Send the message’, the POST action sends the message text back to the **sendMessage** server which runs a handler that gets the message out of the body of the web request and asks the phone to play it.

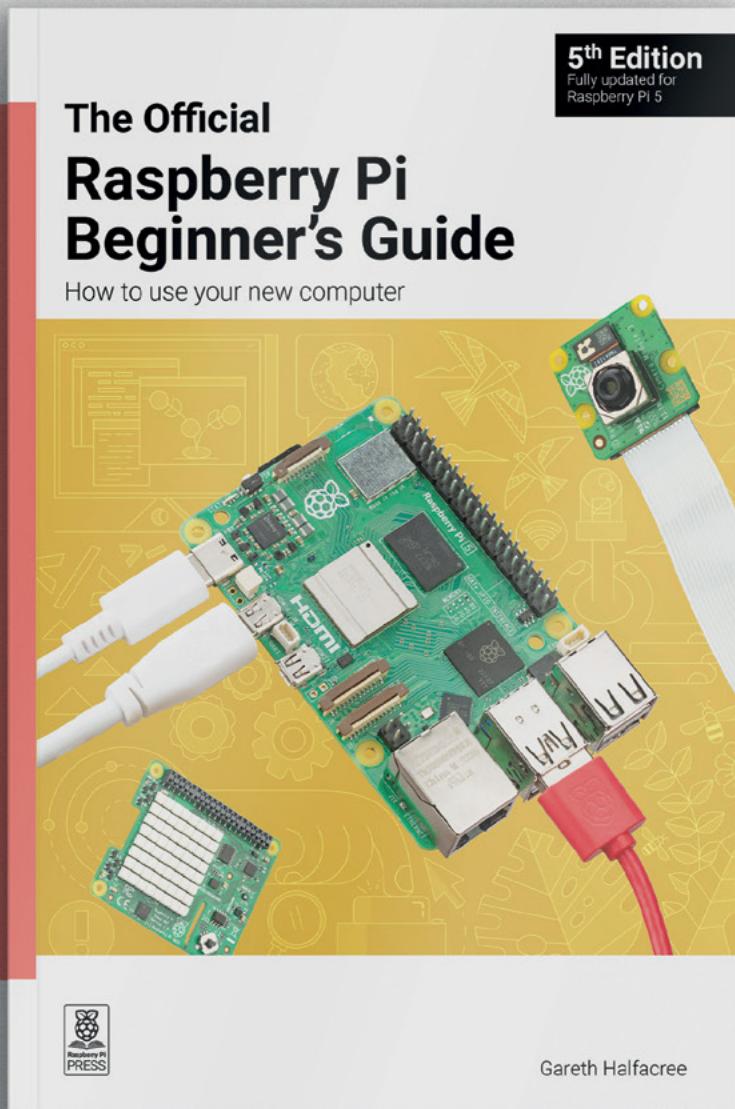
```
app.post('/sendMessage', (req, res) => {
  phone.acceptMessage(req.body.message);
  res.render('index.ejs', {message: 'Message sent'});
});
```

You can add more remote commands by adding routes to the index page and then creating the JavaScript handlers to deal with them.

FURTHER DEVELOPMENT

The author is very pleased with the phone and it works well. He has even bought a second one with a view to connecting them together – once he has figured out how to make their microphones work. It would be interesting to add speech decoding so that the phones can recognise what the user says. The Raspberry Pi inside should be able to do this. **M**

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



magpi.cc/beginnersguide

Arts & Craft

With
Raspberry Pi

Technology and creativity are
inclusive and Raspberry Pi
brings the two together.

By David

M

any Raspberry Pi projects offer solutions to practical problems. Many builds also perform specific tasks that are either informative or seek to help make

life a little easier. Equally, however, there are loads of makers unleashing their creativity and putting Raspberry Pi at the heart of imaginative endeavours: simply because they can.

These makers are producing appealing projects that stop people in their tracks because they either look good or perform tasks that are just sheer, plain fun. These are our kind of makers.

In this feature, we take a look at arts and crafts projects that inspire us. In some cases, they're straightforward but they will hopefully unlock the inventive part of your brain so that you come up with more ingenious ways of taking projects further. In other cases, a project may be difficult to replicate like-for-like, but that doesn't mean you can't be flexible and pursue an alternative path. You may even decide to mash up two or more projects.

Whatever you do: be bold and innovative. And remember, with Raspberry Pi powering your projects, you already have a head start. The possibilities are endless.

THE PI HUT

► thepihut.com

Into making wearables? Snap up conductive thread, woven conductive fabric and more from The Pi Hut.



WHERE TO BUY KIT

PISHOP.US

► pishop.us

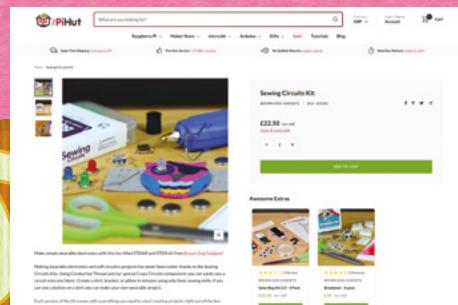
Light up your projects with NeoPixels, screens, and LEDs. You'll find loads on sale at PiShop so take your pick.



PIMORONI

► pimoroni.com

Find great creative kits such as the Bearables Fox LED Badges and LED circuit stickers at Pimoroni



Maker Motivation

DIFFICULTY GUIDE

BEGINNER



Projects for everyone

ADVANCED



Projects that require some experience

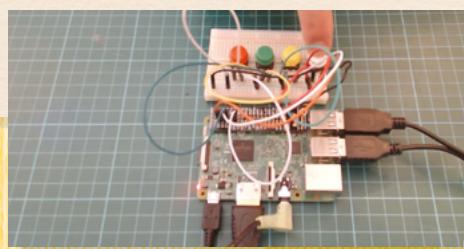
EXPERT



Projects that require skill

TOOLS NEEDED

- ▶ Raspberry Pi
- ▶ Breadboard
- ▶ Four tactile switches
- ▶ Five pin-to-socket jumper leads
- ▶ Four pin-to-pin jumper leads
- ▶ Speakers/ headphones



Don't be afraid to make a noise about your arts and crafts project. In fact: if you try your hand at making this GPIO music box, that's exactly what you'll be doing. Connect a set of buttons to a Raspberry Pi computer and you can begin to play sounds using Python. The more buttons you add, the more sounds you can enjoy, potentially turning the project into a fully-fledged instrument. Either that or a sound box that can make rude noises on demand. The choice is yours.

GPIO MUSIC BOX

► magpi.cc/gpiomusicbox



HONEYCOMB WALL SHELF

► magpi.cc/112



Fancy creating an artistic talking point for your home without going over the top? These honeycomb wall shelves may do the trick. Adopting a hexagonal design, the shelves were designed with a channel, through which 12V RGB light strips could be threaded. Once

outputted to a Creality Ender 5 Pro 3D printer, the shelves could then be connected up to a Raspberry Pi 4 computer from which they are controlled, allowing different levels of red, green or blue to be displayed.

TOOLS NEEDED

- ▶ Raspberry Pi 4
- ▶ 3D printer
- ▶ Four tactile switches
- ▶ RGB light strips
- ▶ Three IRFZ44N MOSFETs
- ▶ Three NPN transistors



RASPBERRY PI BERET



TOOLS NEEDED

- ▶ Raspberry Pi Pico
- ▶ NeoPixel LEDs
- ▶ MicroPython

► magpi.cc/raspberrypiberet

If you want to get ahead in fashion then you need to get a hat, and this Prince-inspired headwear is sure to inspire. As well as allowing the ring of colourful LED lights to animate at random intervals, you can connect a phone and control the lights from a web browser thanks to Raspberry Pi Pico W's wireless LAN. Any hat would work for this project (and you could even decide to make your own). Just make sure there's enough room above your head for the several feet of wire required for the project.



SPEECH-TO-TEXT BADGE



TOOLS NEEDED

- ▶ Raspberry Pi Zero 2 W
- ▶ Touchscreen display
- ▶ Mini HDMI – HDMI Adapter
- ▶ Battery pack and cable
- ▶ USB Microphone

► magpi.cc/speechtotextbadge

So, you're at a party or a conference, and the person you're talking to can't hear anything you're saying. Rather than resort to pointing, you can make yourself a badge that picks up your words via a USB lapel mic, processes them on a Raspberry Pi Zero with the deep learning speech recognition system Deepgram, and transcribes your speech on a small screen.



SCREEN DRESS



► magpi.cc/screendress

Let's be honest here, creating something that could be worn on the catwalk is certainly no cakewalk. But a Dutch FashionTech designer has shown what is possible if you let your imagination run wild. Her ScreenDress makes use of Hyperpixel 2.1 Round LCD screens costing around £50 each, and they've been designed to look like eyes which change according to the wearer's thoughts (gathered by an EEG sensor strapped to the head). They show how you can bring tech together in a creative way.

TOOLS NEEDED

- ▶ Raspberry Pi Zero 2 W
- ▶ Hyperpixel 2.1 Round LCD screens
- ▶ Unicorn Headband
- ▶ 3D printer



Visual

arresting projects

These projects are sure to catch the eye

MATRIX ALBUM ART DISPLAY



► magpi.cc/artledmatrix

As much as we love streaming services, we can't help but think we've lost the art of the album cover. With this fab project, you can bring those covers back to life by displaying them on Adafruit's LED matrix. It's a simple enough project which ties in with the music you're playing so check out the maker's GitHub page (magpi.cc/ftartwork) to learn how to make it.

TOOLS NEEDED

- Raspberry Pi 3B+
- Adafruit 64x64 RGB LED Matrix
- Adafruit matrix bonnet
- Power supply
- 5v 10a switching PSU
- 3D printed matrix feet
- Pi mount



SUNRISE LAMP



► magpi.cc/sunriselamp

The Sunrise lamp marries technology with an origami technique called snapology. It requires a lot of patience but, if you have the time to learn how to use strips of paper to create shapes from rigid triangles, then you too could create a rather special light. A Raspberry Pi Zero W is connected to a set of NeoPixels and the lights have been programmed to dim at night before brightening in the morning to help ease sleepers from their slumber. There's lots of scope to create some amazing designs here.

TOOLS NEEDED

- Raspberry Pi Zero W
- Plastic case
- NeoPixels
- Paper



RUHACAM

► magpi.cc/ruhacam

Photographers don't need to spend a fortune on a cutting-edge brand-name camera, nor do they need to reach for the latest smartphones. With the High Quality Camera for Raspberry Pi, it's possible to produce your own device in any style you want – allowing you to flex your artistic muscles in the creation of a cool case before showcasing your flair for a good photo. This particular project opts for a retro look, but it comes complete with a modern viewfinder. As creator Penk Chen says, "it's literally technology married with liberal arts".

TOOLS NEEDED

- Raspberry Pi Zero W
- High Quality Camera
- 2.2-inch TFT display
- 2000 mAh lithium polymer battery
- Case



Art
Advice!

Why not make yourself a light meter too? magpi.cc/picolightmeter



This visually arresting project works by rotating a strip of LEDs at high speed. By syncing their blinking patterns (and working on the principle that humans continue to briefly see an image that has stopped entering the eye), it's possible to create the illusion of a moving image. The strips are connected to the Pico, mounted on a rotating arm and spun at high speed. A wireless charging module prevents the wires from becoming a tangled mess.



AI ART

► magpi.cc/creatingaiart



Thanks to artificial intelligence, you can enter a prompt, click generate, and see a picture appear before your eyes. If you fancy experimenting with the technology, try OnnxStream, a small inference library that runs Stable Diffusion 1.5 on a Raspberry Pi Zero 2 W. Sure, it can be slow, but it's super impressive. Check how it works and learn how to set it up at github.com/vitoplantamura/OnnxStream

TOOLS NEEDED

- Raspberry Pi Pico
- Adafruit DotStar Digital LED Strip
- QTR-1A Reflectance Sensor
- DFRobot Wireless Charging Module 5V
- DC Motor RS-540SH
- Microsoft Visual Studio 2017



TOOLS NEEDED

- Raspberry Pi Zero 2 W
- OnnxStream

Bullfrog

► Erica Synths ► magpi.cc/bullfrog ► £430 / \$540

SPECS

FEATURES:

Analogue design, 8-octave voltage controlled oscillator (VCO), voltage controlled amplifier (VCA), voltage controlled waveshapes with pulse width modulation (PWM, voltage controlled amplifier (VCA)

I/O:

DIN5 MIDI input, USB connector, CV (controlled voltage), phones out, audio out

VOICECARDS:

Acid bassline, sampler-looper, sequencer, 3 x black voicecards are included

Learn to build a synthesizer, and how electronic sounds are created, with this RP2040-based music kit. By **Lucy Hattersley**



Bullfrog is a collaboration between Erica Synths and Richie Hawtin (the legendary DJ and music producer from Detroit) and features a Raspberry Pi RP2040 microcontroller at its heart.

Raspberry Pi is used to help power this subtractive synthesizer (a device in which tones are created by removing unwanted frequencies). RP2040 is used to take care of MIDI implementation and in a Sampler/Looper voice card.

The Bullfrog itself an analogue synthesizer that uses Eurorack patch cables to connect various sound modules together. It can be used as a professional sound tool, but its primary purpose is educational, to learn how synthesized sounds are created: “Our goal is to both nurture a passion

for electronically produced sounds and promote fun,” says Richie.

The front of Bullfrog is a smorgasbord of dials with instant appeal to anybody who loves tweaking and the feel of hands-on analogue technology. There are three main sections: VCO, VCF, and Delay (corresponding to the three elements of sound: pitch, timbre, and amplitude). To the right of this are envelope generators and a Sample&Hold section, while at the top sits a blue cartridge socket. This is where the (included) voicecards slot in. Voicecards patch the internals of Bullfrog and quickly expand the sounds to create a variety of noises. The kit comes with three voicecards: an acid bassline, sampler-loopers (that can record and playback any sound), and a sequencer. There are also three blank voicecards that you



▲ Around the back of Bullfrog we see input and output including CV and MIDI control inputs

can patch yourself by soldering the points together with wires.

To the rear are CV (control voltage) and MIDI (Musical Instrument Digital Interface) ports, phone and audio out, plus power sockets and config buttons. There's a speaker set into the device itself, or you can use headphones.

Wired for sound

The 77-page manual is where things come to life (magpi.cc/bullfroglearn). It walks you through sound generation, pitch, waveforms, overtones and harmonics, plus virtually every aspect of

■ The 77-page manual is where things come to life ■

sound synthesis. Far more than just how to use the equipment, it covers the science behind sound. If there's any criticism, it's that it gets a little stuck in the weeds before getting you to patch together the components and start making noises. But this is nitpicking on what is a wonderful educational resource. Girts Ozolins from Erica Synths has made a YouTube video that explains the Bullfrog project that also includes a patching guide (magpi.cc/bullfrogyt).

Taking things further

Bullfrog is more fun with a CV (controlled voltage) keyboard, and the manual mentions an Arturia



▲ Girts Ozolins running a workshop in Hamburg

Keystep (magpi.cc/keystep) or a MIDI keyboard. These enable you to turn the synthesized sounds into notes. It's also possible to use Raspberry Pi to expand on the music abilities and learning. Either by using Pico to create a CV generator or by attaching a MIDI HAT to Raspberry Pi (see this OSA tutorial: magpi.cc/osamidi). Both of which could add programming aspects to this sound generator.

Erica Synths is using Bullfrog as an educational tool, and to that end has been running workshops using an XL version of the kit that also features an oscilloscope. They are hoping to get it into educational environments around the world. ■

Verdict

An innovative educational resource that takes you through sound creation and is a fully working subtractive synthesizer to boot. We loved testing this one out

10 / 10

Argon NEO 5

► Argon 40 ► argon40.com ► From £19

SPECS

COOLING:

Passive and active, air intake vents

FAN:

30mm PWM

MATERIALS:

Aluminium, plastic

A new Raspberry Pi means new cases. So what's the well-dressed computer wearing this season? **PJ Evans** suits up



► Strong, dark and handsome, the NEO 5 case looks good

A new form factor for Raspberry Pi 5 means existing cases need updating, especially if you want efficient cooling. Thankfully, the official case was available from day one and it does a great job of protecting your Raspberry Pi 5 from the elements. However, if you need something different, offering more robustness, cooling options and its own style, Argon 40 has always brought something to the table, from their original NEO cases to the iconic ONE. Now it's Raspberry Pi 5's turn.

Argon has launched its entry-level case, the NEO, for our new member of the family. For about £20 you receive an aluminium enclosure with a funky red plastic base. More interesting is the built-in active cooling system that connects to the new on-board fan header. In addition to the fan are heatsinks for the CPU and a power management circuit. All of this, along with good labelling and thoughtful exposure of the various

headers and GPIO, make for a solid package.

The enclosure is

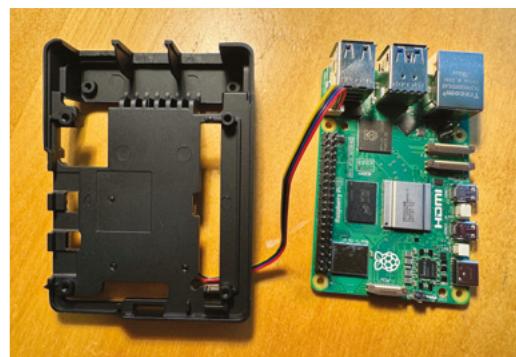
provided with instructions. These are straightforward, although we will confess to some confusion with wiring in the fan cable. It took four attempts before the 'ah ha!' moment when we realised what was intended. The instructions could cover this better. The rest of the installation was painless and thermal pads are included for good contact with the heatsink. Once you have Raspberry Pi inserted, you can add the aluminium cover which gives the overall package a solid, strong feel making it more suitable for environments such as factories or classrooms. The case is held together by screws, adding to the strength of the overall package. You can choose between rubber feet (supplied) or wall-mounting using the built-in screw points.



▲ The NEO combines strength and looks

The case in use

The aluminium cover looks great, but it does mean that the GPIO, PCIE, camera ports and other headers become inaccessible. You are not required to have the cover on to use your Raspberry Pi but it would be great to see alternative covers just as with the official cases. The SD card slot is exposed (which is not the case with many third-party cases) and a nice touch is a cover which can be screwed into place to protect the card from accidental removal. A welcome feature is the combo power light and switch, which links to the new on-board power switch.



▲ Assembly was straightforward but the fan wiring was confusing

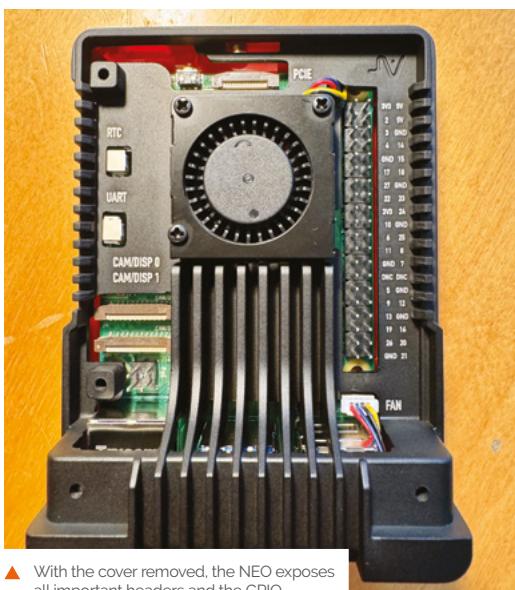
Being actively controlled by Raspberry Pi OS means that the fan is silent for the majority of the time and you've got to get the CPU nice and busy to get any noise out of the case, which is minimal. As the body is nearly all aluminium, there is plenty

“ Argon's build quality is some of the best in Raspberry Pi's space ”

of area for soaking up heat, so this would be a great choice for Raspberry Pis under heavy load.

Argon's build quality is some of the best in Raspberry Pi's space and this case is no exception. You get the impression that a fall off a desk (normally due to a curious cat) would do it no harm whatsoever. The plastic base has a cheaper feel, but the red accents it provides look the part.

The NEO 5 is another excellent product from Argon that combines value and features into a hard-to-beat package. ■



▲ With the cover removed, the NEO exposes all important headers and the GPIO

Verdict

A low-cost, good-looking case with excellent resilience and cooling options. A solid choice for any Raspberry Pi project, although some may wish to wait for the next generation of the Argon ONE.

8/10

8BitDo Retro Mechanical Keyboard

SPECS

KEYS:

87 dye-sublimated PBT keys with Kailh Box White Switches V2, along with two programmable Super Keys connected by 3.5mm jack

CONNECTIONS:

Bluetooth, 2.4GHz (via included dongle), wired (USB-C, cable included)

BATTERY:

2000 mAh Li-ion, rated at 200 hours of use

► 8BitDo ► magpi.cc/8bdkeyb ► £79 / \$100

Retro in aesthetics as well as technology, this gaming focussed keyboard from retro-specialists 8BitDo is very unique as **Rob Zwetsloot** finds out

Interest in mechanical keyboards for serious computer users is not new; finding wireless mechanical keyboards online is easy and allows you to get the tactile feel of the old days (and that ASMRY clickety-clacking as well). While there are definitely brands that focus on modern PC gaming, with brightly backlit keyboards that wouldn't look out of place on the set of *The Fast and the Furious*, not many have old computers or gaming in mind, which is slightly ironic.

8BitDo's latest offering bucks this trend. A pair of beautiful wireless mechanical keyboards with colour

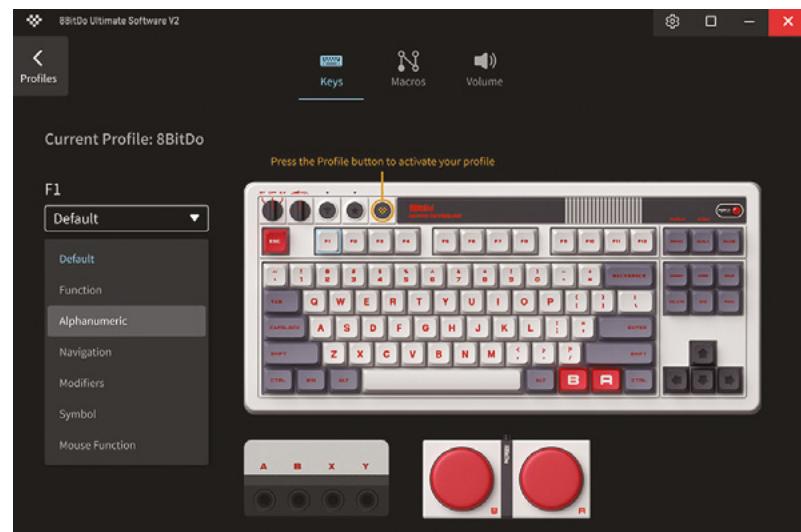
schemes based on the original NES and Famicom, the most noticeable thing about them is an included pair of gigantic 'super buttons', arcade-esque add-ons that plug into the keyboard and can be programmed to run certain keys or macros.

Playing with power

If you've used any 8BitDo input device before, you'll know that you can press a specific macro button to assign a keypress manually to these buttons, along with very conspicuous A and B buttons next to the space bar. In our mind, the intended use of this is



► It's styled after the NES and Famicom, although the Famicom version uses the console as a colour scheme instead of the official Famicom keyboard



either shortcuts on the desktop, or to assign custom keys for a more accurate experience on some retro games via RetroPie. Unlike other controllers and keyboards, you can't set up these extra super buttons inside RetroPie but it will recognise the inputs you've set to it from using the macro button, or via a custom 'profile' that's set up on another machine.

The customisation software, called 8BitDo Ultimate Software V2 is very powerful, allowing you to not only assign keys and/or macros to the super buttons, but also completely remap the keyboard as well – including adding special navigation or media control functions, which is very fancy. Creating macros is very deep as well, allowing you to pull off long combos with custom delay between button presses too. It's an astounding amount of customisation, although unfortunately the software is only available on Windows PCs right now. Still, you can turn the customisation on and off with a simple press of the keyboards profile button, so once you have set it up on a PC you don't need to remain connected to it.

Hook it up

The connection options are great – as well as a classic wired connection via the USB-C charging port, you can connect via Bluetooth or a 2.4GHz RF stick stored away on the underside. We found initially connecting the keyboard via Bluetooth had some issues, but once connected it worked just fine. The 2.4GHz stick worked right away and is currently our preferred way to use it. There's a satisfying clunky switch that goes between the different radio types which perfectly complements the comfy and clicky keys on the keyboard. We had no lag on any

"There's a satisfying clunky switch that goes between the different radio types which perfectly compliment the comfy and clicky keys on the keyboard **"**

games we played on RetroPie, and the super buttons were very fun to smash for that.

The switches and keys are standard PCB/PBT types so you can swap them out if you have preferred keys or want a quieter type (we have been banned from using them at Pi Towers due to the noise, oops).

There is currently only a US layout, and despite the Japanese symbols on the Famicom version, it's missing one or two keys to be a full Japanese keyboard. It is just an aesthetic choice though, and the design of the keys and labelling elsewhere is authentic and very pretty. **M**

▲ The software is fairly easy to use although it does allow for a lot of customisation



▲ The super buttons connect via a 3.5mm jack just like peripherals on accessible game controllers, and are very satisfying to use

Verdict

A very good retro keyboard for retro gaming, although you can't unlock its full features with just a Raspberry Pi

9/10

Revolution Pi Connect 4

► Kunbus ► magpi.cc/rpconnect4 ► From €329 / £286 / \$364 (ex. tax)

SPECS

FEATURES:

CM4 with 8GB, 16GB, or 32GB eMMC storage; 1GB, 2GB, 4GB, or 8GB RAM; optional Wi-Fi/Bluetooth

PORTS:

2 x USB 3.2, 2 x Gigabit Ethernet, RS485, micro-USB, micro-HDMI, 2 x PiBridge headers, X2 input/relay, X4 power input (10.8 to 28.8V DC)

DIMENSIONS:

96 x 45 x 110.5mm

► With its robust polycarbonate housing, the RevPi Connect 4 is designed for industrial settings

Verdict

This CM4-based, ruggedised industrial computer benefits greatly from its modular expansion system.

8/10



Based around a Raspberry Pi Compute Module 4, the Revolution Pi Connect 4 is a powerful, ruggedised unit designed for industrial settings. Its robust polycarbonate housing with IP20 protection can be mounted on a DIN rail, while passive cooling from a heatsink enables a wide operating temperature range of -25°C to +55°C.

Available with up to 8GB RAM and 32GB on-board eMMC storage, the Connect 4 features two Gigabit Ethernet ports, along with a couple of USB 3.2 ports and optional on-board Bluetooth and Wi-Fi (with external antenna connector). A custom version of Raspberry Pi OS is pre-installed (with useful apps), but the eMMC can be re-flashed via a micro-USB port. You can also connect it to a monitor via the micro-HDMI port.

Modular marvel

While there's no GPIO header, a screw terminal-type RS485 connector enables the connection of serial devices such as sensors, while an X2 port adds a single digital input and relay output. If you need more I/O, you can add extra modules – such as the RevPi DIO with 14 digital inputs and outputs, or AIO with two

"This modular system offers a lot of versatility for a wide range of use cases"

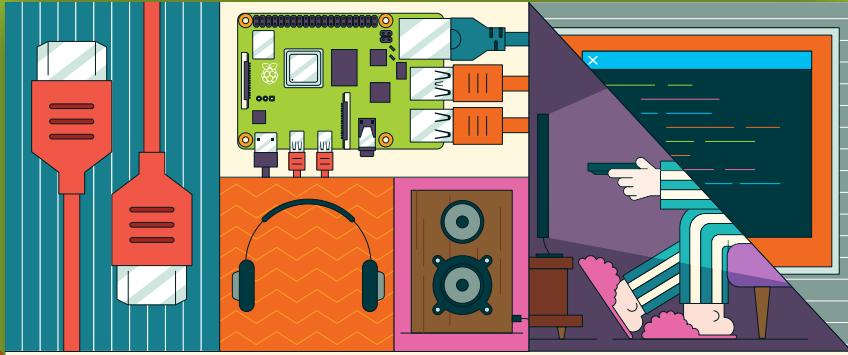
analogue inputs, four outputs, plus two temperature input channels. U-shaped 'PiBridges' make it easy to connect a series of modules side by side – up to ten in total.

This modular system offers a lot of versatility for a wide range of use cases. A good level of support is available on the website, including detailed user manuals and videos, along with a user forum. The price may seem high, but is about par for industrial embedded computers. **M**



► The RevPi Connect 4 can be mounted on a standard DIN rail, along with optional I/O modules

Your FREE guide to making a smart TV



The illustration shows a central green Raspberry Pi board connected to various components: a blue power adapter, a red Ethernet cable, a black HDMI cable, a purple USB stick, and a blue microSD card. To the right, a television screen displays a terminal window with code. Below the TV is a wooden speaker cabinet with a blue speaker. In front of the TV, a person's hand holds a white remote control pointing at it. The background features abstract orange and purple geometric shapes.

BUILD A RASPBERRY PI MEDIA PLAYER

Power up your TV and music system



A collage of five icons arranged in a grid. Top-left: a large orange play button on lined paper. Top-right: a television screen showing a colorful solar system with planets and stars. Bottom-left: a white cloud with a blue download arrow pointing down. Bottom-center: a hand holding a blue remote control. Bottom-right: a green eye icon next to a red and white striped popcorn bucket.

FROM THE MAKERS OF *magPi* THE OFFICIAL RASPBERRY PI MAGAZINE

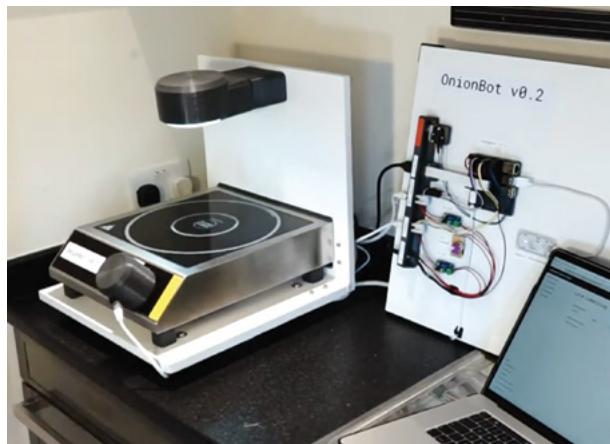
magpi.cc/mediaplayer

10 Amazing:

Home automation projects

Let a computer take some of the work off your plate

Home automation is an incredibly popular way to use a Raspberry Pi due to its size, customisability, and low-power requirements. If you've always wondered if you could upgrade your house with a bit of Raspberry Pi-powered tech, here is some inspiration to maybe get you started.. 

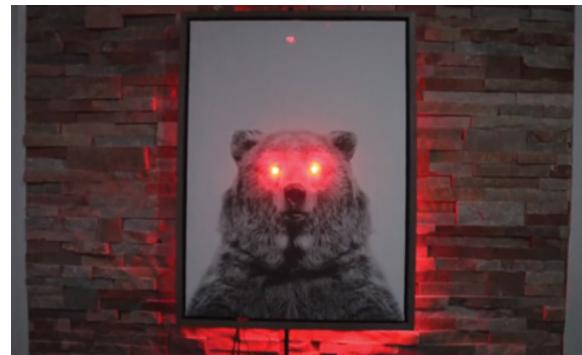


▲ OnionBot

Robotic sous-chef

Softening or caramelising onions can be a bit tricky for some cooks, so let a Raspberry Pi use computer vision to make sure they're cooked perfectly.

magpi.cc/onionbot



▲ Mr Bear

Voice-controlled lights

What was once a way to have AI settle arguments is now a voice-controlled bear painting that controls the lights in your home, naturally.

magpi.cc/mrbear



◀ Coffee maker

Robotic java

Raspberry Pi is used to control the coffee machine via UART, with inputs from a touchscreen and also an NFC reader to make sure the coffee wanter is authorised to brew it.

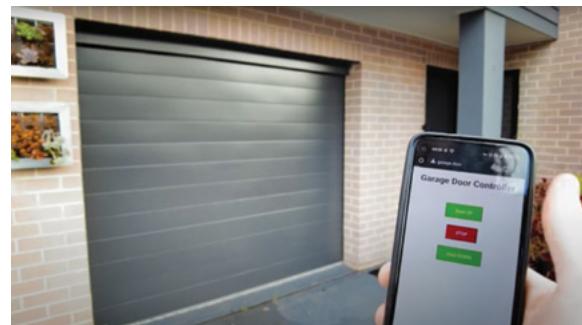
magpi.cc/jurapi

▼ Pico garage door

Low power opening

Remotely open your garage door with a smart phone and a Pico, allowing for greater customisation and timing of how the door works.

magpi.cc/garagepico





► Raspberry Pi garage door

Voice controlled opening

This project connects Siri on an iPhone to a Raspberry Pi to open and close your garage door – it can even check whether it's currently opened or closed if you get that anxiety.

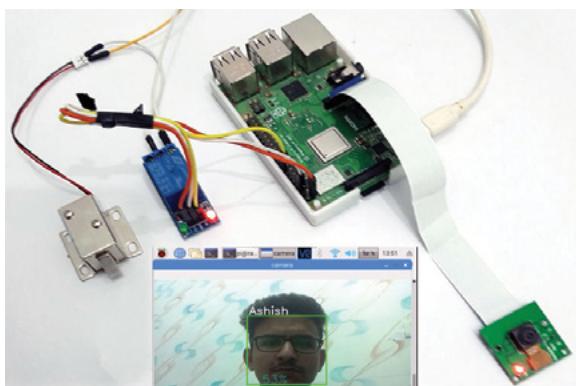
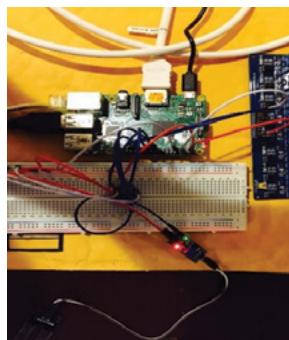
magpi.cc/sirigarage

► IoT community sprinkler

Water-saving irrigation

Young maker Adarsh Ambati created this system to help his local community save money by using weather predictions and soil moisture samples to better irrigate their lawns.

magpi.cc/iotsprinkler



► Face recognition door lock

House Face ID

With machine learning and Raspberry Pi, you can create an app that will automatically open a door if it detects a learned face.

magpi.cc/facelock

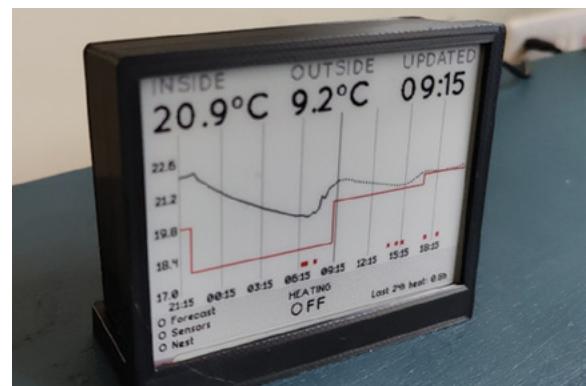


► Raspberry Pi plant watering

Automated gardening

Never let your plants die from thirst again with this automatic watering solution that also has a timelapse function for cool school projects.

magpi.cc/plantwater



► Thermostat

Predictive under-floor heating

Using weather forecasts and some human logic, this automatically activates heating so the residents can be comfy while also saving on energy bills.

magpi.cc/ufhtherm

► PiHue

Colourful lights

If you have the popular Philips Hue light system, you can ditch the standard controller and use a Raspberry Pi instead.

magpi.cc/pihue



Learn algorithms with Raspberry Pi

These resources will help you to learn about this key computing concept. By **Phil King**

A Visual Introduction to Algorithms

AUTHOR	Educative
Price:	Free
	magpi.cc/visualalgo



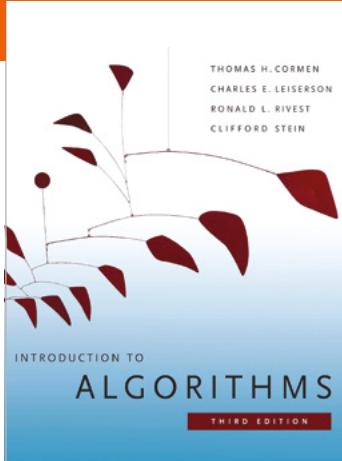
An algorithm comprises a finite sequence of well-defined steps to perform a computation or solve a problem. The concept was first postulated by 9th century Persian mathematician Muhammed al-Khwārizmī, from whose name the word ‘algorithm’ is derived.

Even a cookery recipe can be considered an algorithm, as it comprises a set of logical steps, but the concept is now more commonly associated with computer programs.

A good starting point for newcomers to the field is Educative’s ‘A Visual Introduction to Algorithms’ course. Taking around 14 hours to complete, it is packed with diagrams and illustrations to

help you visualise the concepts. Following an introduction, the course moves on to the binary search method, then asymptotic analysis, selection and insertion sort methods, recursion algorithms, the ‘Towers of Hanoi’ puzzle, and further sort methods (merge, quick, breadth-first). Pseudocode is used throughout, along with JavaScript and Python, depending on the situation. ■

Further reading



Books to help you learn more about algorithms

INTRODUCTION TO ALGORITHMS

Considered the ‘bible’ of algorithms, Cormen and co’s 1292-page classic is rather heavy-going, but still a must-read for anyone who’s serious about the subject. The ideal companion guide to dip into during a course.

magpi.cc/introalgo

PYTHON ALGORITHMS

This 352-page book gives a good overview of implementing algorithms in Python. While focused on classical algorithms, it also gives a solid

understanding of fundamental algorithmic problem-solving techniques.

magpi.cc/pythonalgo

ALGORITHMS

Sedgewick and Wayne’s comprehensive textbook is another must-read. It covers the most important computer algorithms in use today and teaches fundamental techniques, making it both a useful introduction and valuable reference.

magpi.cc/algorithmssw

Grokking Algorithms

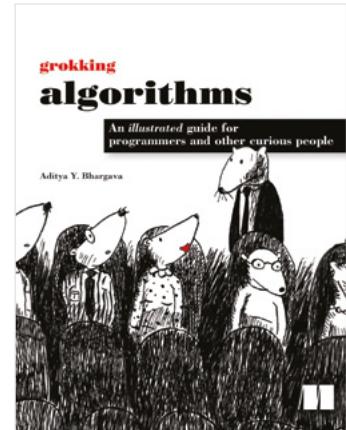
AUTHOR
Aditya Bhargava

Price:
£39 / \$50
magpi.cc/grokalgo2

Soon to be available as a fully updated second edition with fresh coverage of trees, NP-complete problems, and code updates to Python 3, this easy-to-follow illustrated guide is aimed at those with basic coding skills. It teaches you how to apply common algorithms to the sorts of real-world problems you may face as a programmer. Each algorithm example is accompanied by diagrams and fully annotated Python code samples that can be

downloaded, so it's a hands-on style of learning.

Starting off with an introduction to algorithms and the basics of binary search and other methods, it moves on to cover selection sort with arrays and linked lists, recursion, divide-and-conquer quicksort, hash tables, breadth-first search, trees and balanced trees, Dijkstra's algorithm, greedy algorithms, dynamic programming, and K-nearest neighbours. ■



Online courses



Enrol in an algorithms web course

ALGORITHMS PART 1 & 2

Created by Princeton University, this highly rated two-part free course is available on Coursera. It has an emphasis on applications and scientific performance analysis of Java implementations. Modules comprise videos, written materials, and quizzes..

► magpi.cc/algocoursera

in the browser and covers everything from nodes and linked lists to graph searches and path-finding algorithms.

► magpi.cc/algopythonica

MASTERING DATA STRUCTURES & ALGORITHMS USING C AND C++

This paid Udemy course includes 58.5 hours of on-demand video and 270 downloadable resources. You'll need a C/C++ coding environment set up and experience with the language. It's a very comprehensive course, though.

► magpi.cc/algocudemy

LEARN DATA STRUCTURES AND ALGORITHMS WITH PYTHON

This paid Codecademy course takes around 26 hours to complete. The 19 modules include live coding examples

Pironman Mini PC Case for Raspberry Pi 4



Aluminum Tower Case

Ice Tower Cooler

Support M.2 SATA SSD

Programmable OLED Display

IR Receiver

Safe Shutdown



Adam Cohen-Rose

Code Club organiser and Astro Pi challenger,
Adam is helping young folk get ahead in computing

- ▶ Name **Adam Cohen-Rose** | ▶ Occupation **Software engineer**
- ▶ Community role **Community leader** | ▶ URL **blog.cohen-rose.org**

Astro Pi is probably one of the most popular events that our colleagues over at the Raspberry Pi Foundation have ever put on – a chance to get your code running on a Raspberry Pi on the International Space Station for real space experiments.

“I heard about the Astro Pi launch through Code Club and

followed the projects through the first couple of years,” Adam Cohen-Rose told us. He runs a couple of Code Clubs and is a software engineer for Tesco the rest of the time. “I was really excited to hear about the Mission Zero challenge that lets younger students write a simple program that will actually run in the space station – and made

sure we entered teams from the first year in 2017. Since then I’ve run the Mission Zero challenge for multiple groups – the year six club at Fleetville Junior School each year, both Tesco Code Clubs, and a Woodcraft Folk Pioneers group in 2021.”

What is your history with making?

I’ve always enjoyed building things – I had a big chest of LEGO as a kid, and so did my wife, and now we’ve only expanded it as our children got into model building too!

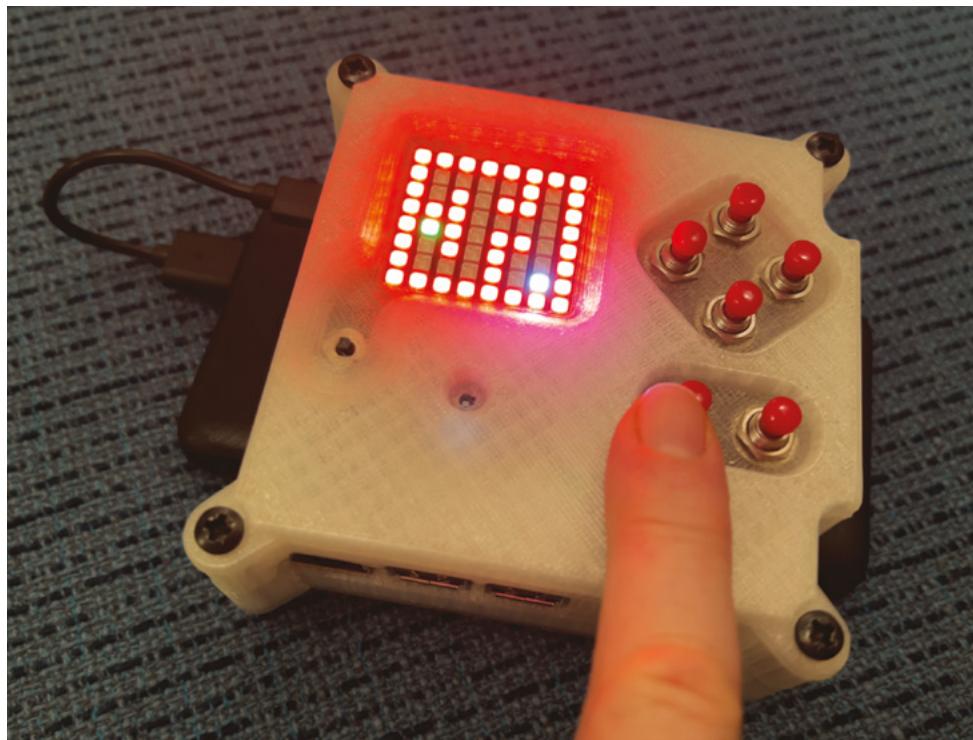
I’ve been going to BarCamps and hackathons since 2007 and have loved the opportunity to work with other people to put something together in a short space of time. I was even featured on BBC Click as an ‘Inventor’ back in 2009 for building a LEGO Dalek controlled from my phone

When did you learn about Raspberry Pi?

Pretty early on. My then 12-year-old (now 20!) was one of the judges at the Code Club Pi-hack back in December 2012.

I’ve three active Raspberry Pi [boards] in the house, plus

▼ The maze game on the Astro Pi – it's a bit small, but so is the screen it's displayed on





◀ Adam shows off his homemade Astro Pi with some Cub Scouts as part of his demo

about seven or eight others, connected or embedded in various projects.

How did you start with Code Club?

I started running a programming club in my child's primary school back in February

running the year 6 club, even though neither of my kids go there any more! And we also run lunchtime clubs for years 4 and 5 – using Code Club material as well as Minecraft Education, micro:bits and Machine Learning for Kids.

I've also started up Code

and a bar graph for the humidity and temperature sensors. The kids then have to try different inputs to figure out what they are responding to. See the code and some pictures here: magpi.cc/astrodemo

Seeing the actual Astro Pi hardware at Raspberry Pi Big Birthday Bash events and at Richard Hayler's talk at EMF Camp – and then seeing videos of their twins in the space station. M

▼ This displays temperature and humidity, so during the demo people can figure out how the Sense HAT works

■ The children really enjoyed making their own games ■

2012 – just before Code Club was founded. I approached the head teacher to offer a free lunchtime club for year 4s and he jumped at the idea!

Once the Code Club Scratch projects came out, I switched to using them pretty quickly as they were great fun and the children really enjoyed making their own games.

We've now had Code Clubs at Fleetville Junior School for nearly 12 years. I'm still

Clubs at work: Tesco Technology supports two clubs – one lunchtime club by our Welwyn office, and one after-school club by our London office.

What are some of your favourite Astro Pi moments?

Getting the kids to guess what the sensors are on my 3D-printed Astro Pi mockup – I run a small program that displays a maze for the gyroscope and accelerometer,



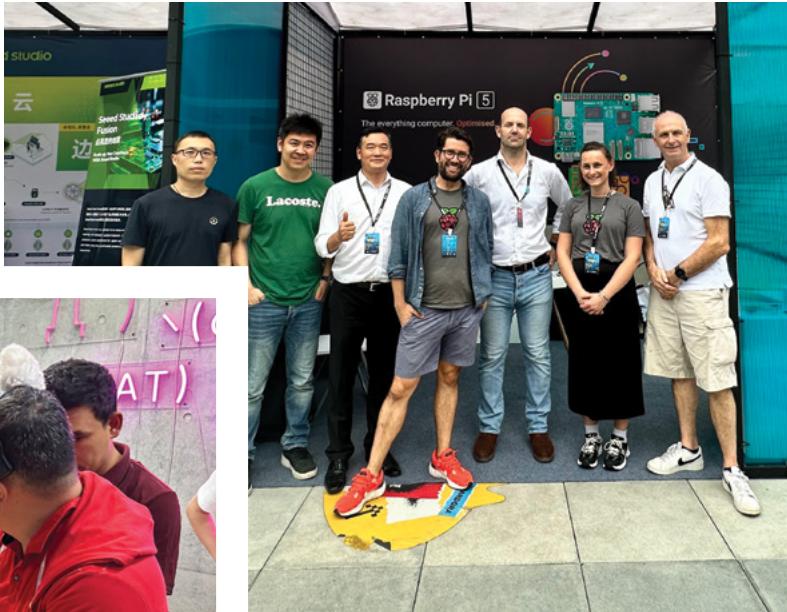
Raspberry Pi at Maker Faire Shenzhen

One of China's biggest tech hubs celebrates makers and making, with help from Raspberry Pi

At the end of 2023, the Raspberry Pi community team and Raspberry Pi CEO Eben Upton attended one last big event before Christmas: Maker Faire Shenzhen. Joining them was Seeed Studios, Raspberry Pi Approved Reseller, showing off their wares and hunting down cool Raspberry Pi projects in the process. It was a busy event full of talks and cool stalls. ↗



▲ Showing off your project to Eben feels very special



▲ Raspberry Pi could be found next to the Seeed Studio tent



▲ A Raspberry Pi meetup was organised to show off what people were working on



▲ Eben joined talks on industry topics



Raspberry Pi projects on display

Some of the cool projects that Seeed Studios found at Maker Faire Shenzhen

ROBOT DOGS BY YANG LI



"I am a chip design engineer," says Yang, "and playing with Raspberry Pi and building robots has been a long-term hobby of mine. When I first saw a video of Boston Dynamics' robot dog, it impressed me. Before this I had tried building a robot dog using 996 servos, but they were not responsive and torquey enough for my requirements. However, when MIT released the design scheme for the Cheetah mini robot dog's geared-down motors, smaller and more affordable FOC motors with gears quickly became available. This made it more feasible for me to build this robot dog."

SOUND-TRACKING GOPRO STABILIZER BY WEIJIE YU



After looking at functions in a modern vlogging camera, Weijie came up with the idea of making use of the three-axis movement of the lens to track the speaker instead of just being used for face tracking, which can be a bit jittery.

"First of all, there are four MEMS microphones on the ReSpeaker mic array that pick up sound from different angles, convert them into multi-dimensional audio signals, and transmit them to Raspberry Pi through the USB cable," Weijie says. "Then, Raspberry Pi executes the sound source tracking algorithm, an introlab ODAS, to process the raw audio signals, calculating the elevation, azimuth, and active level of several potential sound sources."

RASPBERRY PI RETRO COMPUTER BY CHANGLIANG LI



"The project was inspired by the Commodore HX-64, a non-existent handheld computer that can only be found in 3D renderings on the web. I decided I had to make it," says ChangLiang. "I spent a week doing 3D modelling, repeatedly adjusting the slot position, and I felt cool being able to change the tape. I used a 3D printer to create the desired look, and despite a lot of polishing the shell is still not very smooth. Raspberry Pi CM4 has a strong ability to adapt, so even if my PCB design is not perfect, it still works very well... I even designed a cartridge in the shape of a separate tape cartridge with an SD card inside, just to be retro. I think it would be cool to put different operating systems into different tapes."

MagPi Monday

Amazing projects direct from social media!

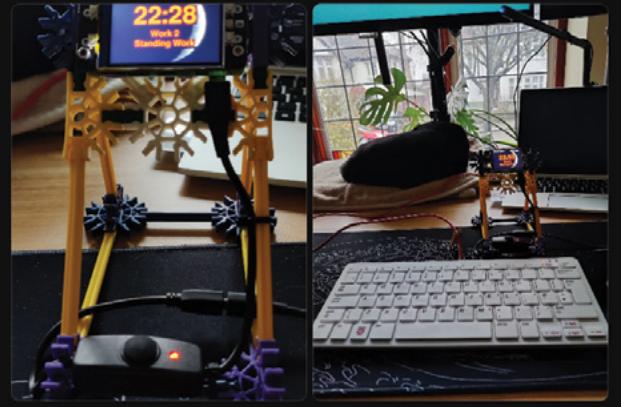
Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made.

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. Once again thinking we need to make some Pomodoro stuff for ourselves
02. A mini synthesizer you can plug into your laptop via Raspberry Pi Zero? Love it
03. Very cute robot, and ten axes is a lot of movement
04. A robot that maps its environment is very smart
05. Adding Raspberry Pi to your other hobbies is a rite of passage
06. This is a cool idea to add VGA output to Pico
07. We've done that before! Although not the screen part. Good luck

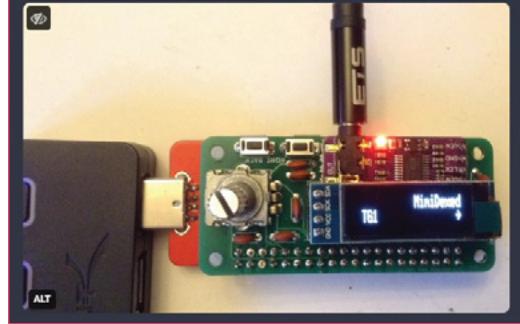
richard.kirby351 5 01

I made a Raspidoro timer that implements the Pomodoro technique, but allows custom timers and prompts to help keep me focussed and taking suitable breaks. This includes stretch and exercise breaks as well as standing work sessions at my standing desk. Helps my focus immensely, but also ensures I look after myself. Built with a Pi ZeroW2, Pimoroni Display Hat mini, and Kin'x building toy. Cat for scale. Moon phases indicates progression of the timer.



dielelectromusic @dielelectromusic@mastodon.social 02

@themagpi I've now got a #MiniDexed IO board for a Pi zero, which is great for using with the USB Gadget mode "DX7 on a dongle".
(I've a MIDI board on the way... watch this space!)
#MagPiMonday
[diylectromusic.wordpress.com/...](http://diylectromusic.wordpress.com/)





이윤재

Our robot uses Raspberry Pi as the main controller. It is a 10-axis robot for educational/home use.



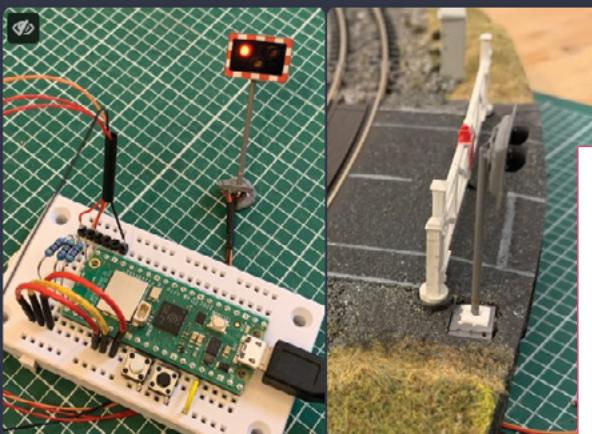
03



Pater Practicus

@paterpracticus@mastodon.world

@themagpi Pico-powered scale model level crossing lights up and running. First one installed on the layout. Only three more to make! #MagPiMonday



05

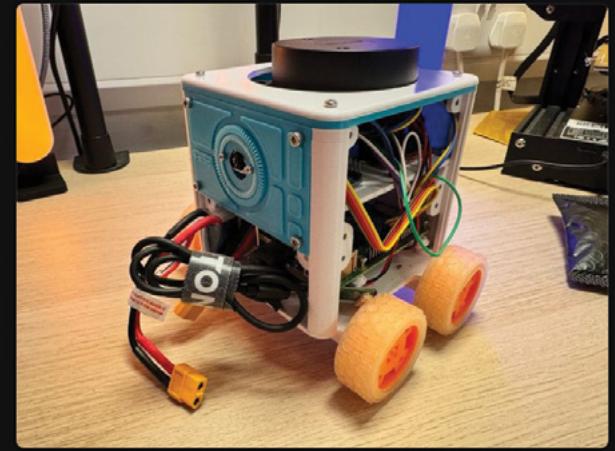


kevinmcaleer

Happy #MagPiMonday This weekend I upgraded my @raspberrypi 4 powered Cubie-1 robot so it can move around and map its environment!

Learn more at kevsrobots.com/blog...

Tech Threads #STEM



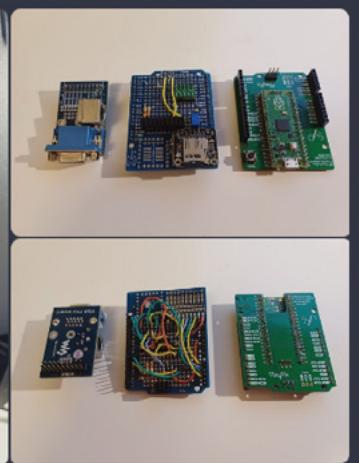
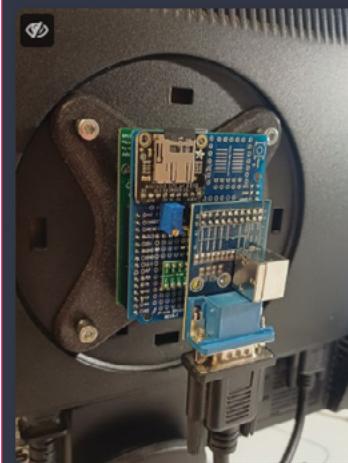
04



Szymon Jakubiak

@szymonjakubiak@fosstodon.org

@themagpi I have built myself a PicoMite VGA, which is an all in one, BASIC programmable computer based on the Pico. More info here: geoffg.net/picomitevga.html



06



Ray Kelly

Trying to squeeze all that in to a "snes" controller. 😊



07

Raspberry Pi pumpkin

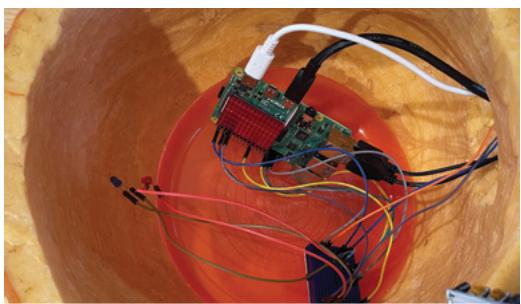
Missing Halloween? Get spooked by this project from Elliot

Normally we'd have Christmas projects to show you around now, however we didn't see any this year, and we forgot to publish this cool Raspberry Pi pumpkin from reader Elliot who is nine years old.

"I used the GPIO pins to make LEDs flicker like candles, and a motion sensor and Bluetooth speaker to make it flash blue and play spooky music when the trick-or-treaters come too close!" Elliot tells us. "I got the idea from someone at the October Coder Dojo at Raspberry Pi HQ in Cambridge, UK." 



▲ The Raspberry Pi logo lit up, albeit spooky. Maybe.



▲ Keep your Raspberry Pi clean by putting it on plate inside.

Raspberry Pi Pumpkin

DOWNLOAD
THE FULL CODE:

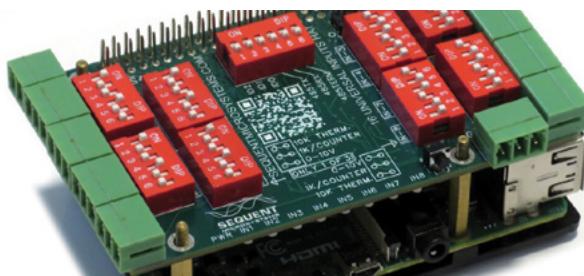
 magpi.cc/github

```
001. from gpiozero import LED,          029. pir = MotionSensor(4)
002. MotionSensor                      030.
003. from time import sleep           031. def pumpkin():
004. from random import randint       032.     while True:
005. randint                           033.         flame_red()
006. from signal import pause        034.
007. from pygame import *             035.     sleep(randint(20,50)/300)
008.                                     036.         turnOff()
009. #initialise pygame's            037.
010. mixer.init()                     038.     sleep(randint(20,50)/300)
011. mixer.music.load("Tww.          039.
012. mixer.music.set_volume(1)        040.     def detected():
013.                                     041.         mixer.music.play()
014. #set red and blue pins         042.         for i in range(0,150):
015. bluePin = LED(12)               043.             spiritual_blue()
016. redPin = LED(19)                044.             sleep(0.1)
017.                                     045.             turnOff()
018.     def turnOff():              046.             sleep(0.1)
019.         redPin.off()           047.             print("Something
020.         bluePin.off()           048. moved")
021.                                     049.     pir.when_motion = detected
022.     def spiritual_blue():      050.     pir.when_no_motion =
023.         redPin.off()           051.         pumpkin()
024.         bluePin.on()           052.
025.                                     053.         pause()
026.     def flame_red():           054.
027.         redPin.on()           055.
028.         bluePin.off()          056.
029.                                     057.
030.                                     058.
```

Crowdfund this

Great crowdfunding projects this month

UNIVERSAL ANALOG/DIGITAL INPUTS HAT



This HAT for Raspberry Pi 5 allows it to process analogue signals much more easily with 16 inputs that can be stacked quite high on Raspberry Pi 5. Each input can be changed to take a different signal too, included thermistors, so it will aid in IoT applications.

► [kck.st/3RJ4HOK](https://kickstarter.com/kck.st/3RJ4HOK)

REBOI



There were a few Game Boy games that tried to turn your 8-bit handheld into a PC with varying levels of non-success, but this project will overhaul the classic console so that you can put a Raspberry Pi Zero inside it – although mainly to run RetroPie. It should be able to play more games than the Game Boy could as well.

► [kck.st/48bd5gj](https://kickstarter.com/kck.st/48bd5gj)



A wireless hydrometer and thermometer that connects to your Raspberry Pi

Free international shipping with checkout code:

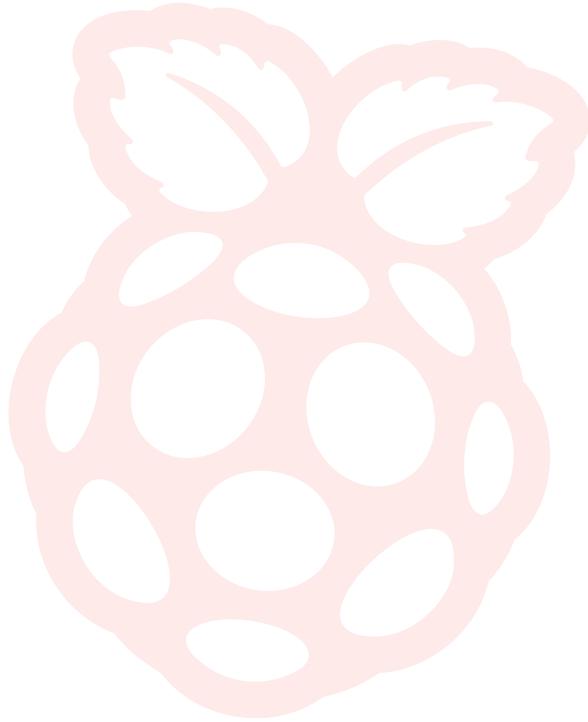
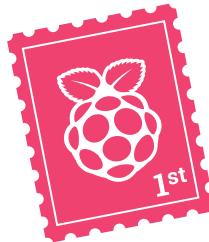
SHIP-FREE-INTL

shop now at
tilthydrometer.com

Download SD Card Image here
QR code



Your Letters



Gadget types

I am following *The MagPi* tutorial “Make a Pi Zero W Smart USB flash drive” and using a Raspberry Pi 4B to test. I’ve connected it to laptop but cannot see the USB mass storage. I followed the tutorial until `sudo modprobe g_mass_storage file=/piusb.bin stall=0 ro=1` then attached to laptop USB. How to debug or know what’s wrong?

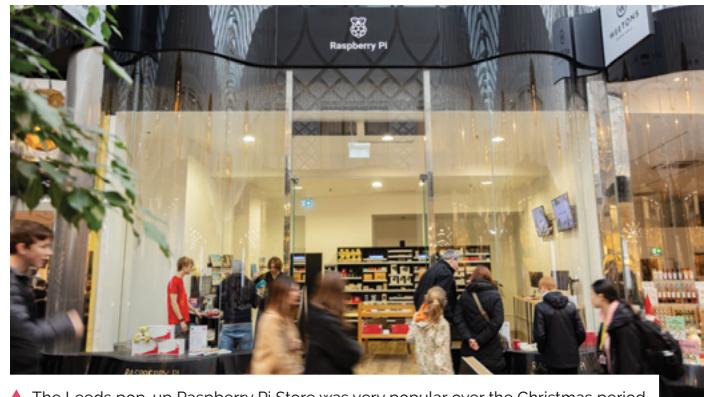
Nelson via email

This is a bit of an older tutorial that we will probably revisit and revise in the future, however the main issue would be that you’re doing it on a Raspberry Pi 4. The USB gadget mode that allows for this project to work is different on a Raspberry Pi 4 compared to Zero, so the tutorial will not be one-to-one compatible.

We’ve not tested it, but if you look up gadget mode tutorials for Raspberry Pi 4, you may be able to get something similar working, although the point of using a Raspberry Pi Zero is that it uses a lot less power.



If you want to try out this tutorial, check out magpi.cc/usbflash



▲ The Leeds pop-up Raspberry Pi Store was very popular over the Christmas period

Leeds pop-up

I am writing to say a big “thank you” to the young man who served my wife in the pop-up shop in Leeds recently. My wife isn’t very clued up on the Raspberry Pi – she worked in IT for years but doesn’t care much for computers these days!

The gentleman in question spent some time with my wife, explaining everything she needed to know, and more. They also had a chat about Scotch whisky as well for some unknown reason!

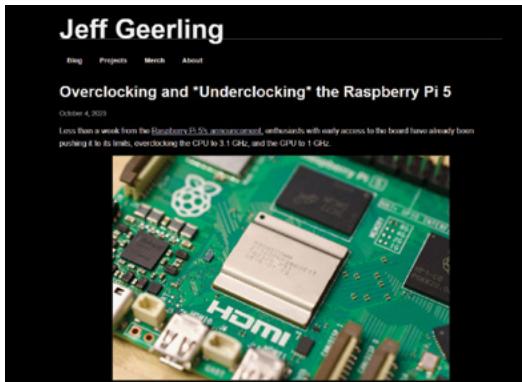
She came away with a bag of Raspberry Pi goodies, including an 8GB Raspberry Pi 5 which I am assured is for my Christmas present – lucky me – as long as I use it to “automate the entire house” which apparently the young man in question told her the Raspberry Pi 5 could do, easily!

Thanks again to whoever you are.

Norman via email

We’ve relayed this message to the shop team who were delighted to hear your feedback! And yes, home automation is an excellent use for a Raspberry Pi 5 – maybe even Scotch distilling if you know your stuff?

▼ The blog covers some basics on how to change the clock speed on Raspberry Pi 5 – but be careful if you plan to copy it



Link issues

In *The MagPi* #135 on page 81, when you click the link to the Lost Sounds Projects it won't load all of the page – an icon spins and doesn't do much.

Also in issue 136 on page 42, the link to Jeff Geerling's overclocking blog doesn't seem to work.

Ted via email

For the Lost Sounds Project, the website can be a little slow at times so we suggest visiting it another time to get the full experience – we think it's worth it!

The link to Jeff Geerling's blog looks to be a typo – sorry! It should be (and point to on the PDF) magpi.cc/jeffoverclock. We've made it so the original link will now work though. Sorry for the inconvenience.

Contact us!

- Mastodon [@themagpimag](https://magpi.cc/mastodon)
- Threads [@themagpimag](https://magpi.cc/threads)
- Facebook magpi.cc/facebook
- Email magpi@raspberrypi.com
- Online forums.raspberrypi.com

USA SPECIAL!

6 ISSUES

FOR \$43



FREE
**RASPBERRY PI
PICO W**



Subscribe online:
magpi.cc/subscribe

Continuous credit card orders will auto-renew at the same price unless cancelled. A free Pico W is included with all subscriptions. This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

Community Events Calendar

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

01. HYBRID Raspberry Pint Experience Raspberry Pi 5

- Tuesday 30 January
- 📍 Online
- ▶ magpi.cc/hrp138

This is Raspberry Pint, where we share our digital making experiences via informal presentations and discussions.

Most of our presentations are about building personal or professional projects with Raspberry Pi and other maker tech. We also welcome presentations about skills and techniques such as website design, PCB design, software development, 3D printing, soldering, etc. Occasionally, we have had presentations about artificial intelligence, big data, IoT, and things like that, and would love to hear about what you do in your hobby or professional life. All makers and all digital making welcome. The quirkier, the better.

02. Cornwall Tech Jam

- Saturday 3 February
- 📍 Fraddon Village Hall, Saint Columb, UK
- ▶ magpi.cc/ctj138

Cornwall Tech Jams are run by volunteers working in IT and education throughout Cornwall. They are supported by Software Cornwall, its members and other local businesses. Our volunteers give their own time and expertise to plan each Cornwall Tech Jam and to the regular maintenance of all our equipment.

03. Newham Citizen's Science Fair

- Saturday 17 Feb
- 📍 University of East London, London, UK
- ▶ magpi.cc/ncsf

We're organising a "Raspberry Pi Jam-like" tech event at the University of East London, Stratford, Water Lane on Saturday 17 Feb 2024. There's a larger space for demonstrations, accessible via stairs and lift, and a smaller space downstairs for talks, including AI bias and use of AI in healthcare. There will be five-minute lightning talks at the end too.



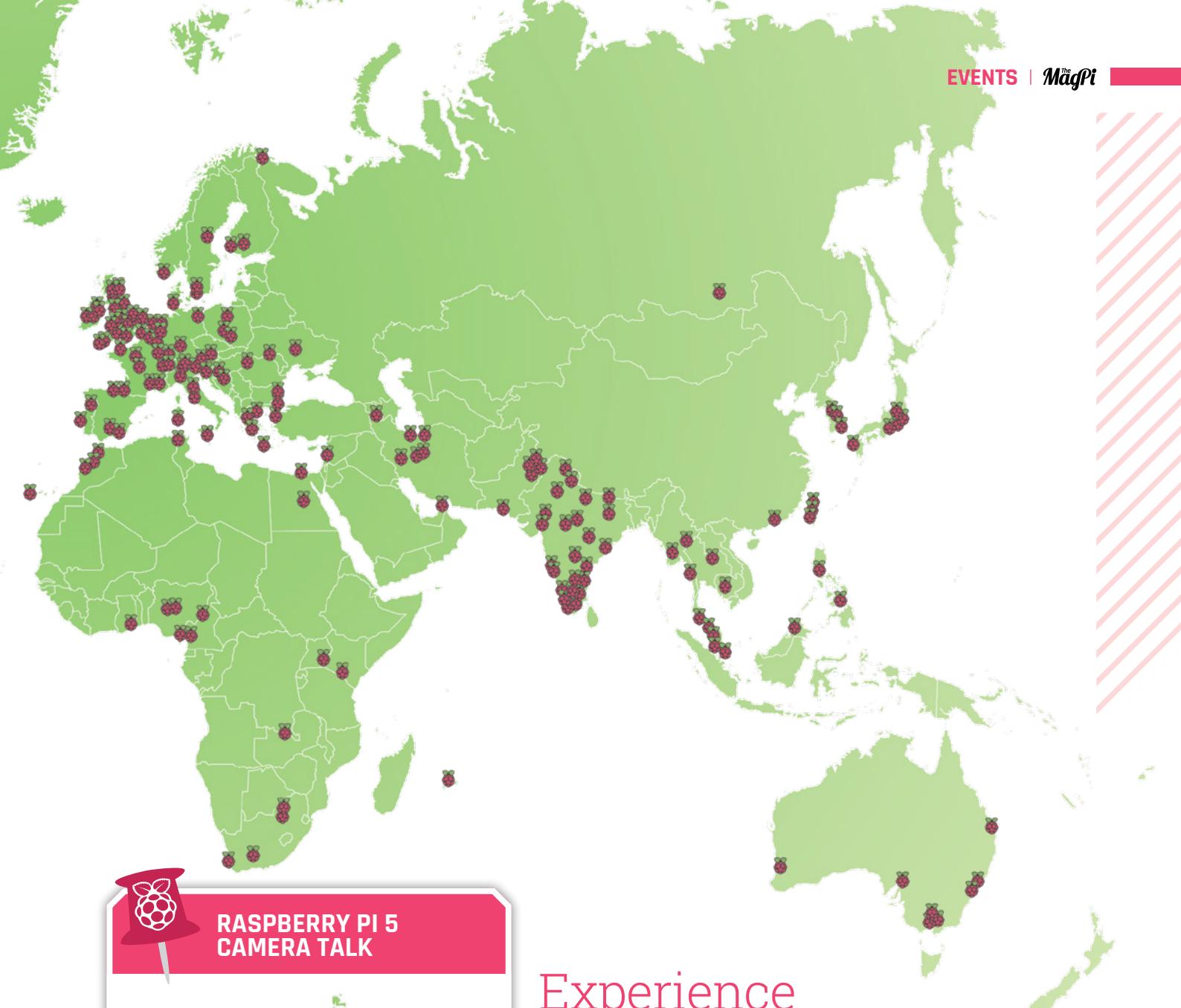
04. Roanoke Raspberry Jam

- Saturday 2 March
- 📍 South County Library, Roanoke, VA, USA
- ▶ magpi.cc/rrj138

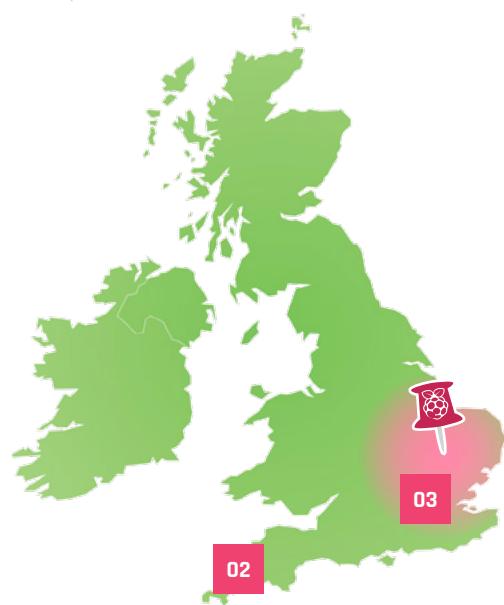
Come to the 2024 Raspberry Pi Jam to see and learn about Raspberry Pi projects, Raspberry Pi Pico projects, Linux operating systems, Python coding, maker tech, and 3D printing projects. There will be exhibitors and technical experts on hand for collaboration and informal Q&A.

FULL CALENDAR

Get a full list of upcoming community events here:
magpi.cc/events



RASPBERRY PI 5 CAMERA TALK

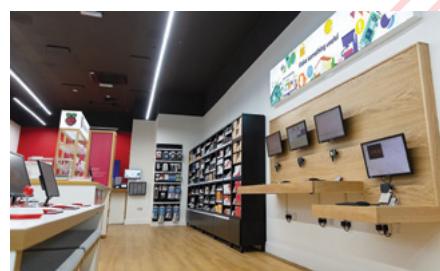


Experience Raspberry Pi 5

- Where [Raspberry Pi Store, Cambridge, UK](#)
- When [Ending Wednesday 31 January](#)

Get hands-on with the all-new Raspberry Pi 5 in our flagship store on the first floor of the Grand Arcade, a shopping centre in Cambridge, England. Experience Raspberry Pi 5's blazing performance and check out the newest accessories. While you're there, reserve a Raspberry Pi 5 for purchase so that you can be among the very first to have it. Speak to one of our store colleagues for more details.

magpi.cc/rpsexp



WIN ONE OF THREE CROWVISION DISPLAYS!

These 11.6" portable displays are very slim and practical, and are designed for use with Raspberry Pi as well. They include a capacitive touchscreen at a resolution of 1366×768 for fun projects in and out of the house.



In association with



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

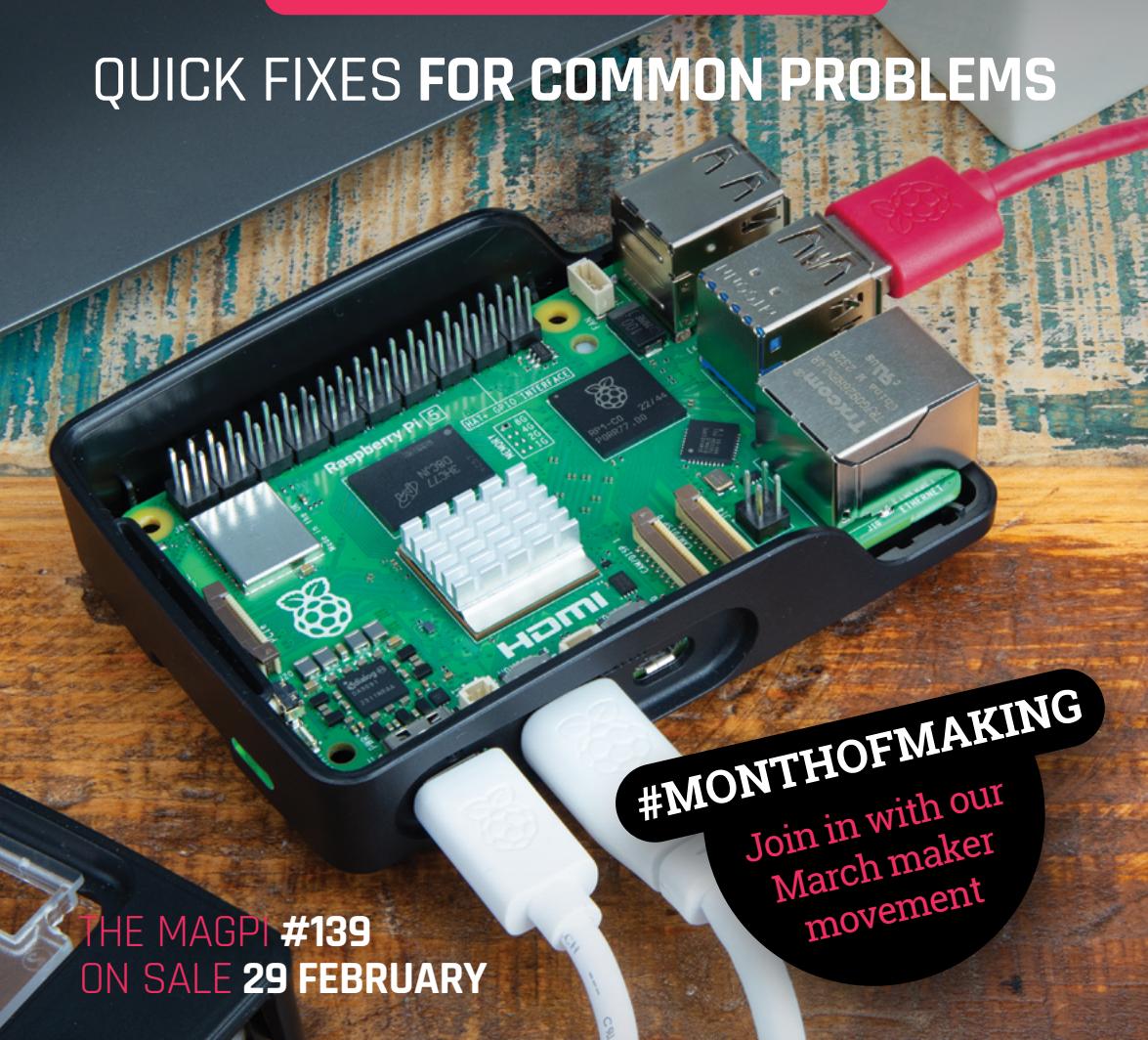
Terms & Conditions

Competition opens on **24 January 2024** and closes on **29 February 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 or any other companies used to promote the service.

TROUBLESHOOTING

Raspberry Pi 5

QUICK FIXES FOR COMMON PROBLEMS



#MONTHOFMAKING
Join in with our
March maker
movement

THE MAGPI #139
ON SALE 29 FEBRUARY

Plus!

Testing the super
fast M.2 HAT+
hardware

**Transform a
Sonos system
with SomaFM**

Give THEA500 Mini
a working keyboard

DON'T MISS OUT! magpi.cc/subscribe

MASTODON magpi.cc/mastodon

THREADS [@themagpimag](http://magpi.cc/thread)

FACEBOOK magpi.cc/facebook

EMAIL magpi@raspberrypi.com

ONLINE forums.raspberrypi.com

EDITORIAL

Editor

Lucy Hattersley
lucy@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

criticalmedia.co.uk

Head of Design

Lee Allen

Designer

Sam Ribbits

Illustrator

Sam Alder

CONTRIBUTORS

James Adams, Alasdair Allan, Neil Bryan, David Crookes, PJ Evans, Rosie Hattersley, Phil King, Simon Long, Rob Miles, KG Orphanides, Stewart Watkiss

PUBLISHING

Publishing Director

Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications

Liz Upton

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



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 Science Park, Cambridge, 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).



NonCommercial-ShareAlike
3.0 Unported
(CC BY-NC-SA 3.0).
ISSN: 2051-9982.



DOS ain't dead

You can bring new life (and games) to dead computers

The last stand-alone version of MS-DOS was released in 1994, so you might be surprised to know that a commercial DOS game came out on Steam in 2023. Hadrosaurus Software's *The Aching* (magpi.cc/theaching), is an otherworldly horror adventure with 16-colour graphics, an arrow key and text parser interface, and hardware support for the Tandy 1000, a PC released in 1984. You can play it on Raspberry Pi, or anything else that'll run the DOSBox emulator.

No one really uses MS-DOS any more, but the modern, open-source FreeDOS ships with every copy of DOSBox, and you've quite probably used that. Most modern DOS developers use DOSBox and its forks for testing, so they can rapidly spot bugs and iterate solutions.

The year 2023 in DOS also saw the release of Damien "Cyningstan" Walker's stylish *Barren Planet* (magpi.cc/barrenplanet), a turn-based, space exploitation-themed strategy game in which rival mining corporations battle for control of resources, with some of the best four-colour CGA graphics we've ever seen. Cyningstan has also released a range of tools and libraries (magpi.cc/cyningstan) to

support DOS games development in C, as well as open-sourcing his older games.

Juan J. "Reidrac" Martinez, developed *Gold Mine Run!* in C and cross-compiled from Linux to DOS, using DJGPP to target 32-bit (i386) DOS. He also open-sourced (magpi.cc/goldminerungit) the game's code to help other developers.

But you don't have to use C. Tiny DOS city-builder *Demografyx*

"A commercial DOS game came out on Steam in 2023"

(magpi.cc/demografyx) was developed in Microsoft QuickBasic 4.5, an IDE released in 1990, which you can run on Raspberry Pi in DOSBox if you can find a copy. Microsoft's more common QBASIC and GW-BASIC languages are no longer available, but PC-BASIC (magpi.cc/pbasic) is a fully-compatible GW-BASIC interpreter you can install on Raspberry Pi, and there's even a GW-BASIC extension for Visual Studio Code if you want an IDE.

There's an entire community of developers making wildly

distinct games based on ZZT (museumofzzt.com), a 1991 game creation system by Tim Sweeny, now CEO of Epic Games. ZZT spawned a vast living ecosystem of DOS games like Wil's Galactic Foodtruck Simulator (stale-meme.itch.io/gfs), development tools like KevEdit (magpi.cc/kevedit), and modding tools such as Weave (meangirls.itch.io/weave-3).

There are multiple DOS game jams to encourage would-be developers. In 2023, we saw the DOS COM jam (magpi.cc/doscomjam), the DOS Games June Jam (magpi.cc/2023jam), and the DOS Games End of Year Jam (magpi.cc/dosgoty).

The DOS renaissance still has a way to go before it catches up to the C64, ZX Spectrum, and Game Boy development scenes, but the sheer range of tools available makes it a very approachable space to experiment in. If you want some inspiration, check out this DOS games we've created at magpi.cc/itchdos. ■

K.G. Orphanides

K.G. fell in love with games writing and the command line in DOS and decamped to Linux in protest over Windows Me. They regret nothing.

tweot.space/@owlbear

AUTHOR

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:



Welectron.

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

PiKVM

Remote control **redefined**

Manage your
servers or PCs
remotely!



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 USB 3.0
- 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



Welectron.



HiPi.io

No reseller in your country?
Check shop.hipi.io (import fees might apply).

List of official
resellers by country:

