Go outside **with some summer projects**

Building **an Alien-inspired cyberdeck**

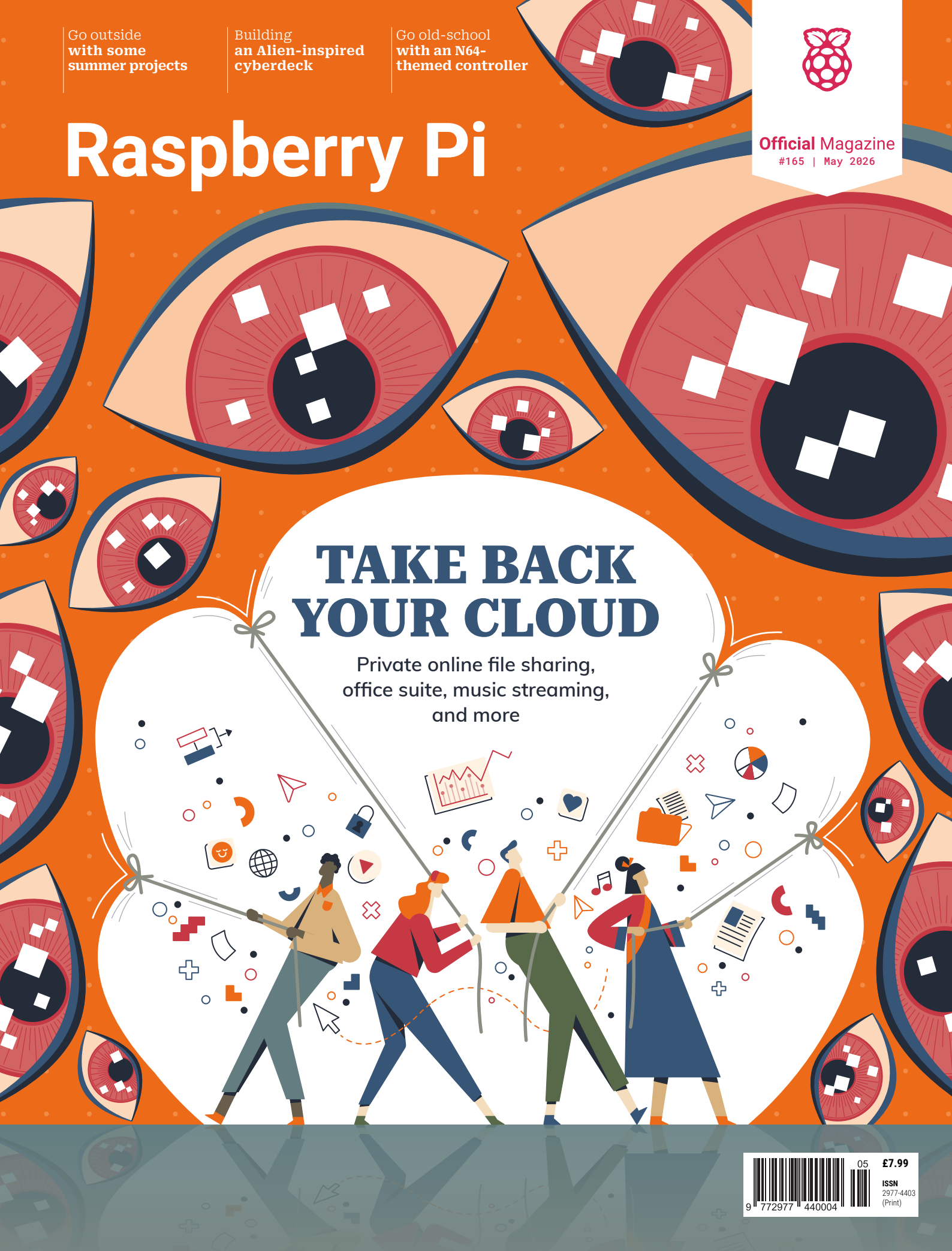
Go old-school **with an N64-themed controller**

Raspberry Pi

Official Magazine
#165 | May 2026

TAKE BACK YOUR CLOUD

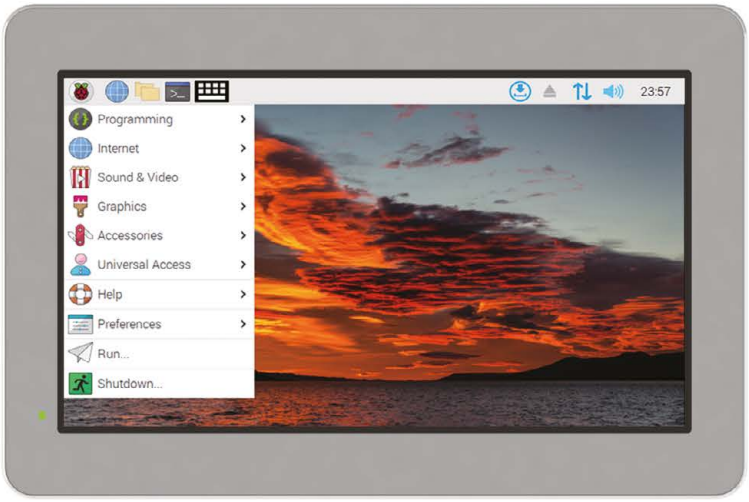
Private online file sharing, office suite, music streaming, and more



05 **£7.99**
ISSN 2977-4403 (Print)
9 772977 440004

Industrial Raspberry Pi

ComfilePi



UL LISTED

Powered by Raspberry Pi

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

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

Visit www.comfiletech.com

COMFILE TECHNOLOGY

Welcome to Raspberry Pi Official Magazine

Editor
Lucy Hattersley
Lucy has been advised not to say "thank you" to LLMs to avoid anthropomorphizing them. If you know her you'll know how hard she is finding this. Thank you!

rpimag.co

It's not paranoia if they really are watching you. And cloud providers really are watching you.

From the words in your email, the images on your phone, your home movies, or even your work files – if it's in the cloud, it's on someone else's computer. And they are sifting through your personal files.

Recent advances in large language models (LLMs) and image recognition models have raised the stakes. Tech companies and governments can crank through every word, image, and video you have ever stored: building a detailed profile of your life, politics, family, and personal affairs.

If you care (and you should), it's time to Take Back Your Cloud. Raspberry Pi is ideal for this kind of home server project. You can replicate all those digital services and serve them up from a computer you own.

As the old saying goes: 'if you're not paying for the product, you are the product.'

Lucy Hattersley – Editor

PCBWay

FULL-SERVICE ELECTRONICS MANUFACTURER

PCB Fabrication / CNC | 3D Printing / PCB Assembly / OEM I EMS

CUSTOMIZED TECHNICAL PARTS & PCB MANUFACTURED

WHY CHOOSE PCBWAY?

PCBWay offers a wide range of services including PCB fabrication, PCB assembly and even CNC machining for over a decade, and has earned a distinguished reputation globally in the industry.

Hassle-free ordering

The digital quote-to-order platform puts you in the back seat. Upload a Gerber file and receive feedback soon.

On-time shipping

We maintain a 99% on-time delivery rate, working in three shifts to ensure your packages arrive fast.

Quality assurance

Our technicians have been working strictly to high standards. All the boards will go through the most stringent tests.

Customer support

The customer service teams work inshifts to provide 24-hour support. You can always contact a live customer service person.

Scan the QR code for more details!

CONTACT US:

www.pcbway.com

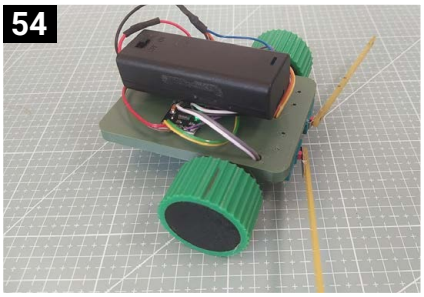
service@pcbway.com

Contents

World of Raspberry Pi

Project Showcase

Raspberry Pi (ISSN No: 2977-4403, USPS No: 25-672) is published 12 times per year by Raspberry Pi Ltd, and distributed in the USA by Asendia USA, 701 Ashland Ave, Folcroft PA. POSTMASTER: send address changes to Raspberry Pi, 701 Ashland Ave, Folcroft, PA. 19032



Top Projects

Feature

Tutorials

Feature

Reviews

Raspberry Pi Community

Competition

Raspberry Pi sets up third-party antenna certification

Developers looking to build projects with custom antennas get a route forward.

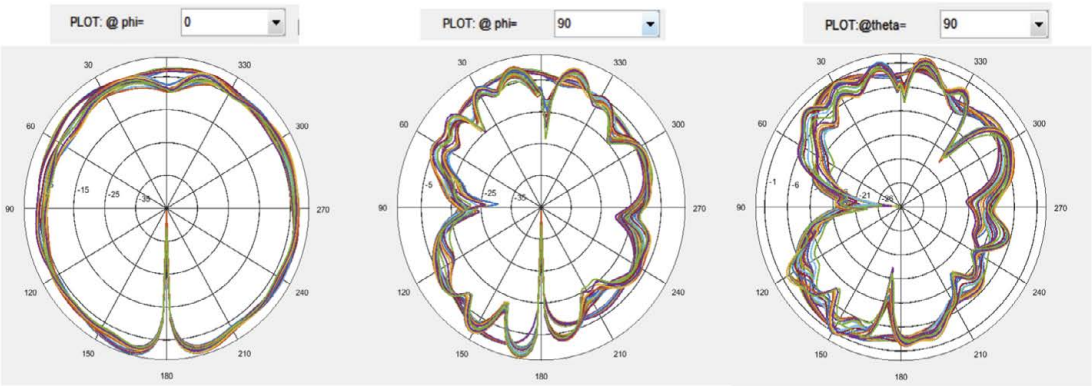
By **Lucy Hattersley**

▶ Raspberry Pi's Antenna Kit already has certification in over 60 countries

Raspberry Pi has published new guidance for industrial developers looking to create their own third-party antennas.

Because radio signals don't respect boundaries, and the radio spectrum is a shared resource, antennas are regulated across markets. Ofcom manages it in the UK, while the FCC is responsible in the US. Other markets have their own regulation agency services.

Raspberry Pi single-board computers ship with on-board PCB antennas while industrial-facing devices, such as the Compute Module 5, have a Raspberry Pi Antenna Kit (**rpimag.co/antennakit**). Both are certified in over 80 countries thanks to Raspberry Pi's in-house compliance department and state-of-the-art test facilities. You can read more about this on the Raspberry Pi product compliance page (**rpimag.co/compliance**).



▲ Antenna gain plot for the external whip antenna included in the Raspberry Pi Antenna Kit

Roll your own

Alex Branton, Global Market Access Lead at Raspberry Pi, said: "Some industrial and commercial customers may need to employ third-party antennas for their applications."

> *The GMA team will review your antenna specifications and advise on compliance*

He provides some examples such as embedding a Compute Module in a metal enclosure where the PCB antenna performs poorly due to the Faraday cage effect, extending the communication distance of the device, and integrating an antenna with a different form factor.

"In such cases, the Compute Module and new antenna may be required to undergo additional testing and certification before the product can be sold," said Branton. These procedures vary depending on the market and device's features.

"Raspberry Pi is well placed to support our customers in meeting these additional requirements," explains Branton, "either by updating our existing certifications or by obtaining new certifications on their behalf."

Compliance requirements

For new antennas, the compliance requirements depend on the specific country's regulations, and whether the antenna gain is less than, equal to, or higher than the approved gain value. Alternative antenna options are therefore split into two categories:

- Antenna gain is equal to or less than the approved antenna gain
- Antenna gain is higher than the approved antenna gain

To help its commercial and industrial customers meet regulatory requirements in either gain scenario, Raspberry Pi has put together a white paper outlining the certifications and testing procedures required in a number of key markets. Download the Third Party Antennas Guide from Raspberry Pi's product information portal (PDF link: **rpimag.co/thirdpartyantennasguide**).

Warning!
Check radio regulations
The information provided here and in Raspberry Pi's white paper is intended to be used as initial guidance only. Customers should always refer to the official regulations and publications issued by relevant local authorities.

Using pre-approved Raspberry Pi antennas

Customers can avoid compliance issues by using Raspberry Pi's existing antenna architecture, which is compliant in all the firm's key markets.

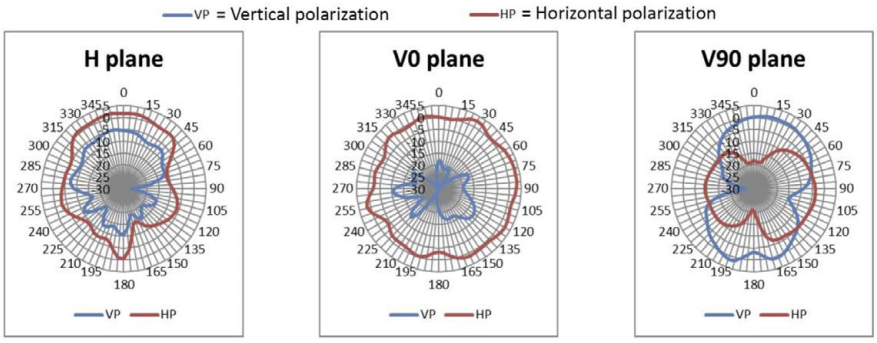
Newer Raspberry Pi single-board computers and microcontrollers include an integrated PCB Niche antenna. This provides on-board Wi-Fi and Bluetooth connectivity as standard. Raspberry Pi Compute Module 4 and 5 also feature PCB Niche antennas, along with a built-in U.FL connector for attaching an external antenna.

The U.FL connector on Compute Module 4 and 5 can be fitted with the omnidirectional external whip antenna included in the pre-approved Raspberry Pi Antenna Kit, or with another compatible third-party antenna.

Next steps: How Raspberry Pi can help

Should you need further assistance with compliance for an alternative antenna, get in touch with Raspberry Pi's in-house Global Market Access (GMA) team: **gma@raspberrypi.com**.

The GMA team will review your antenna specifications and advise on compliance with the relevant market regulations. Once confirmed, the team will update the existing approvals or obtain new ones to include the new antenna, carrying out any additional testing as required.



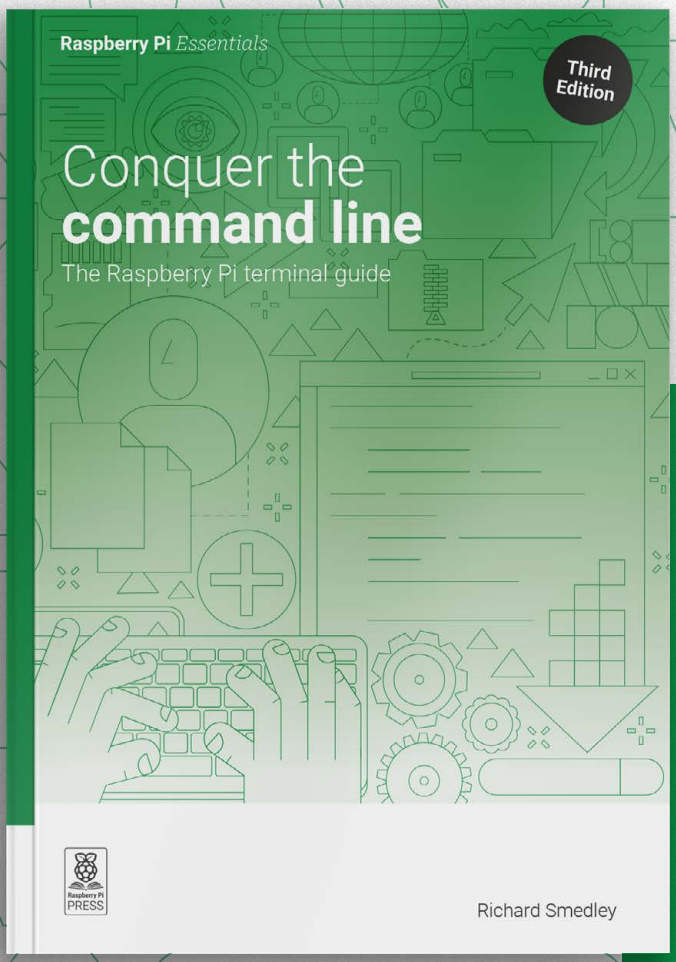
▲ Antenna gain plot for the on-board PCB Niche antennas

Different markets, different regulations

In the UK and EU, for example, integrators can adopt an antenna with a gain less than or equal to the gain of the antenna used for the original certification without needing to carry out any further spectrum usage testing. For antennas with higher gain, this course of action depends on how high the gain of the new antenna is, as this determines whether some or all of the spectrum usage tests need to be repeated. Integrators are, however, encouraged to carry out spurious emissions tests and other electromagnetic compatibility tests on all alternative antennas, regardless of their gain.

In Japan, all antennas must be approved by the country's Ministry of Internal Affairs and Communications (MIC), and all antenna options must be listed, but no additional testing is required. Similarly, in South Korea and Taiwan, all antennas must comply with each country's regulations – but further testing is required for antennas with higher gain. In Vietnam and Mexico, no modifications to the device's existing certifications are required; however, manufacturers must ensure that the radiated output power of the antenna does not exceed the regulatory limits.

For a full list of requirements in several of Raspberry Pi's key markets, refer to the handy table in the white paper.



If you're not comfortable when faced with the $ prompt, don't panic! In this fully updated book, we'll quickly make you feel at home and get you familiar with the terminal on Raspberry Pi (or any Mac and Linux computer). Updated for the latest Raspberry Pi software, this book has everything you need to get started.

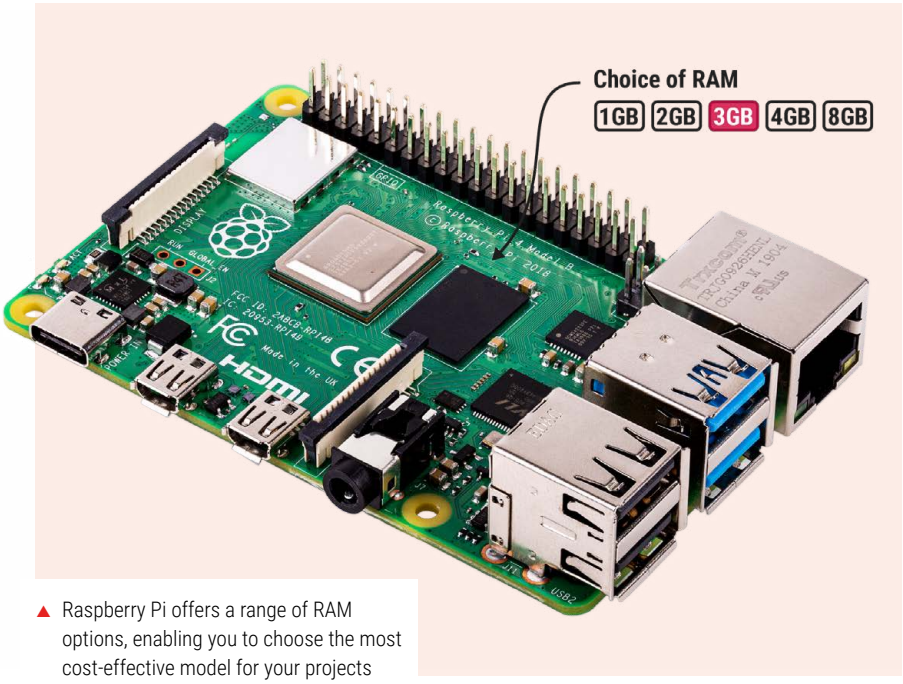
- ***Build essential skills, including:***
  - *Read and write text files*
  - *Find and install software*
  - *Manage removable storage*
  - *Use Secure Shell (SSH) for remote access*
  - *Write disk images to SD cards*
  - *Browse the web from the command line*

BUY ONLINE: **rpimag.co/commandlinebook**

Raspberry Pi 4 with 3GB RAM announced

New option to help users get the right amount of RAM amidst further price rises.

By **Lucy Hattersley**

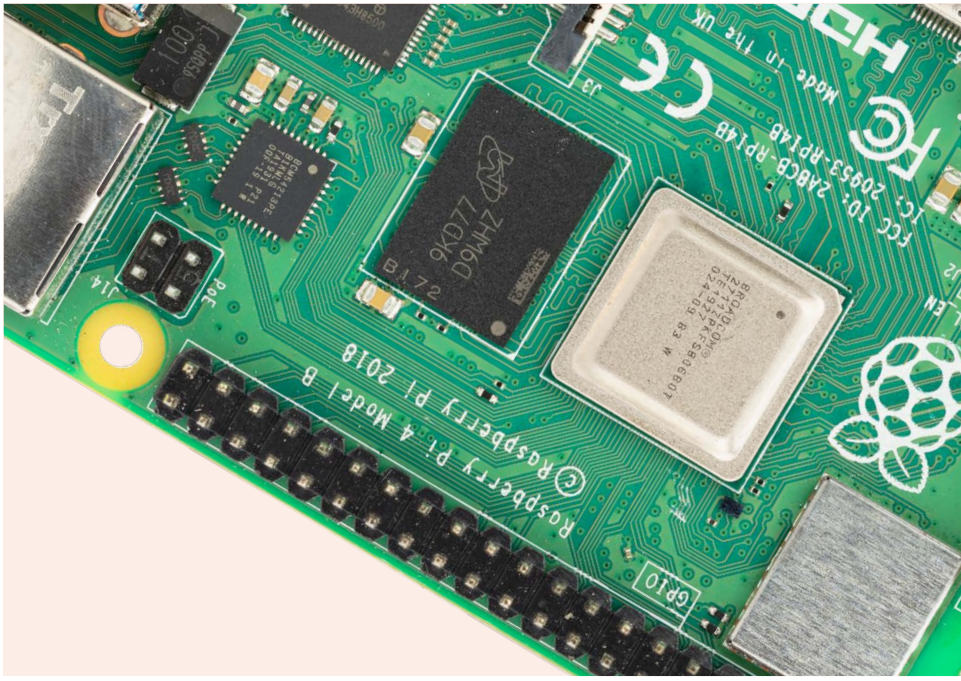


▲ Raspberry Pi offers a range of RAM options, enabling you to choose the most cost-effective model for your projects

Raspberry Pi continues to address the ongoing memory pricing issue via a new Raspberry Pi 4 variant, with 3GB RAM.

Priced at $83.75, the unusual option was announced on 1 April, leading some to think it was an April Fools' joke, but it's real. Eben Upton, CEO and co-founder of Raspberry Pi, said: "Notwithstanding today's date, our new computer is as real as the rest of our products, and you can order it now from Raspberry Pi Approved Resellers around the world.

"In this environment, it's well worth right-sizing both your memory and your overall compute, rather than going for something with more headroom than your application actually needs," continued Upton. "Consider whether these models, or lower-density variants of newer models, will get the job done."



▲ The large black chip next to the silver SoC is the RAM

In this environment, it's well worth right-sizing both your memory and your overall compute

Get the right model

There has been a sevenfold increase in the last year of the price of LPDDR4 DRAM used in Raspberry Pi 4 and 5, leading to price rises for models with 4GB or more of RAM (**rpimag.co/pip**). "As painful as these price rises are, there are some brighter spots in the picture," said Upton. "We've been able to hold the price of Raspberry Pi 400 with 4GB of memory at $60, and the 1GB and 2GB variants of Raspberry Pi 4 and Raspberry Pi 5 at between $35 and $65. These are capable and versatile modern Raspberry Pi computers at affordable price points.

In issue 164 (**rpimag.co/164**) we looked at using htop to investigate and manage memory, and how to run Stable Diffusion to generate images in a 1GB RAM space (along with many other projects such as media players, magic mirrors, and retro games consoles – all of which work fine with 1GB).

Raspberry Pi does not anticipate any price rises for its classic products, including Raspberry Pi Zero, Zero W, and Zero 2 W; Raspberry Pi 1, 3, 3B+, and 3A+; and Compute Module 1 and 3+. "These products use older LPDDR2 DRAM," explains Upton, "of which we currently hold substantial inventory. Our commitment to long product manufacturing lifetimes means that these products are still in production, still supported by software updates, and still used by customers all over the world."

This too shall pass

The price rises will not continue indefinitely and Raspberry Pi will bring prices down when it is possible to do so. "The circumstances in which we find ourselves are challenging," says Upton, "but in the future they will abate. When they do, we will reverse our price increases, and until they do, we will continue to work hard to limit their impact in every way we can."

Learning to code in an AI age

A new Raspberry Pi Foundation course helps young people learn to code, and five reasons why they should. By **Lucy Hattersley**

▲ Kids learning vital programming skills on the Isaac Computer Science programme

Raspberry Pi exists to support the Raspberry Pi Foundation, which helps young people learn to code and become tech creators.

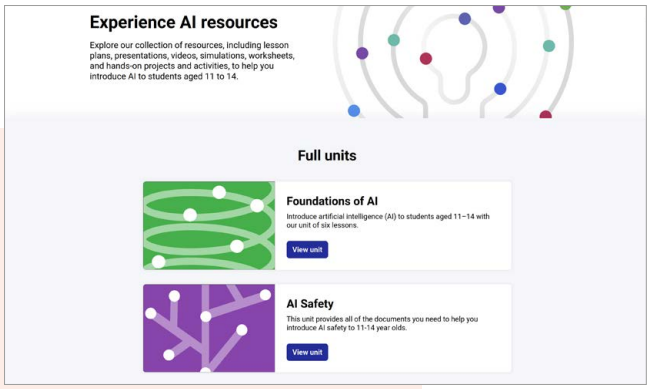
All the recent advances in artificial intelligence, especially in the field of large language models (LLMs), have put a new spin on programming. It's led some to speculate that the art of programming may well disappear. We couldn't disagree more, of course.

Google recently granted $4.6 million to the Raspberry Pi Foundation to bring AI education to 1.25 million students across Latin America (**rpimag.co/eduialatino**).

"Working with education partners across Argentina, Brazil, Chile, Colombia, Dominican Republic, El Salvador, Mexico, Peru, and Uruguay, we will help young people develop a foundational understanding of AI technologies, their social and ethical implications, and the role that AI can play in their lives," says Anna Burton, Director of Global Partnership at the Raspberry Pi Foundation.

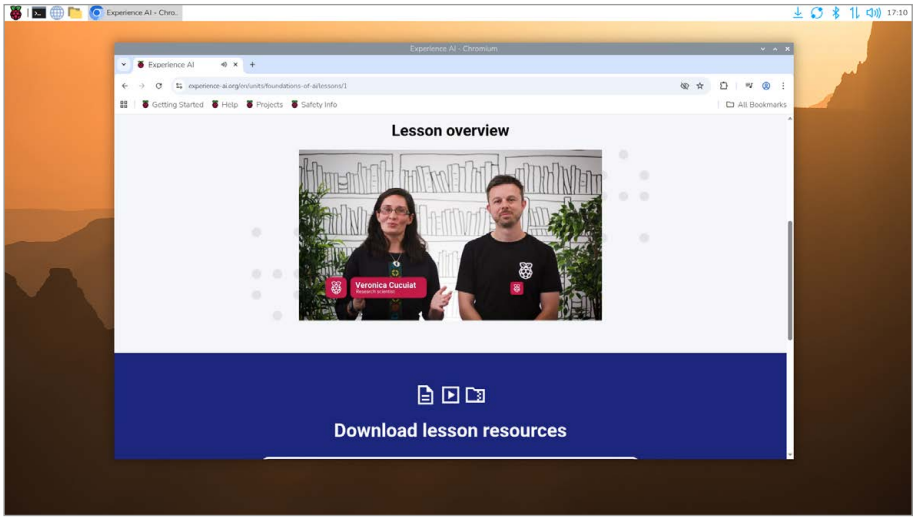
Experience AI

Experience AI (**experience-ai.org**), developed in partnership with Google DeepMind, is a free educational programme that helps teachers and students learn about artificial intelligence



Young people who learn to code will be the ones who ultimately shape the future that we all live in

▲ The new Experience AI website will enable young people to develop AI coding skills
▶ The Experience AI website has video tutorials that help young people develop AI programming skills



(AI). It introduces young people to how AI systems work and how they are used in everyday contexts through lessons, classroom resources, and hands-on activities. The resources give young people opportunities to think critically about the role of AI in society.

"Experience AI is designed not only to build technical understanding, but also to help young people think critically about AI and its impacts," says Burton.

No matter where you live on the globe, a good understanding of programming concepts and coding literacy is going to help you thrive in a world driven by AI.

Philip Colligan, CEO of the Raspberry Pi Foundation, has written an excellent white paper outlining why it is vitally important that kids learn to code in the AI age (**rpimag.co/learncodeaiwp**).

Colligan outlines his five reasons in a Raspberry Pi Foundation blog post, *Why kids still need to learn to code in the age of AI* (**rpimag.co/kidscodeai**):

1. We need humans who are skilled programmers
2. Learning to code is an essential part of learning to program
3. Learning to code will open up even more opportunities in the age of AI
4. Coding is a literacy that helps young people have agency in a digital world
5. The kids who learn to code will shape the future

Colligan draws a distinction between programming and coding. "Programming is perhaps best understood as the process of formulating a problem in a way that it can be solved by a computer," he explains. Whereas coding is "the way that humans give instructions to computers."

Kids need skills

The paper posits that even with AI code generation, we will "need skilled human programmers who can think critically, solve problems, and make ethical decisions."

Learning to code is the most effective way for young people to "develop the mental models and fluency to become skilled human programmers."

It also makes a compelling case that learning to code will open up "more economic opportunities" for young people alongside the advances in technology. It suggests that those young people who learn to code will be "the ones who ultimately shape the future that we all live in."

SUBSCRIBE TODAY FOR JUST £10

Get **3 issues** + **FREE** Pico 2 W

SUBSCRIBER BENEFITS

Free delivery
Get it fast and for free

Exclusive offers
Great gifts, offers, and discounts

Great savings
Save up to 37% compared to stores

SUBSCRIBE FOR £10

Free Pico 2 / Pico 2 W

3 issues of Raspberry Pi Official Magazine

£10 (UK only)

SUBSCRIBE FOR 6 MONTHS

Free Pico 2 / Pico 2 W

6 issues of Raspberry Pi Official Magazine

£30 (UK)	**$43** (USA)
€43 (EU)	**£45** (Rest of World)

Subscribe by phone: 01293 312193

Subscribe online: rpimag.co/subscribe

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

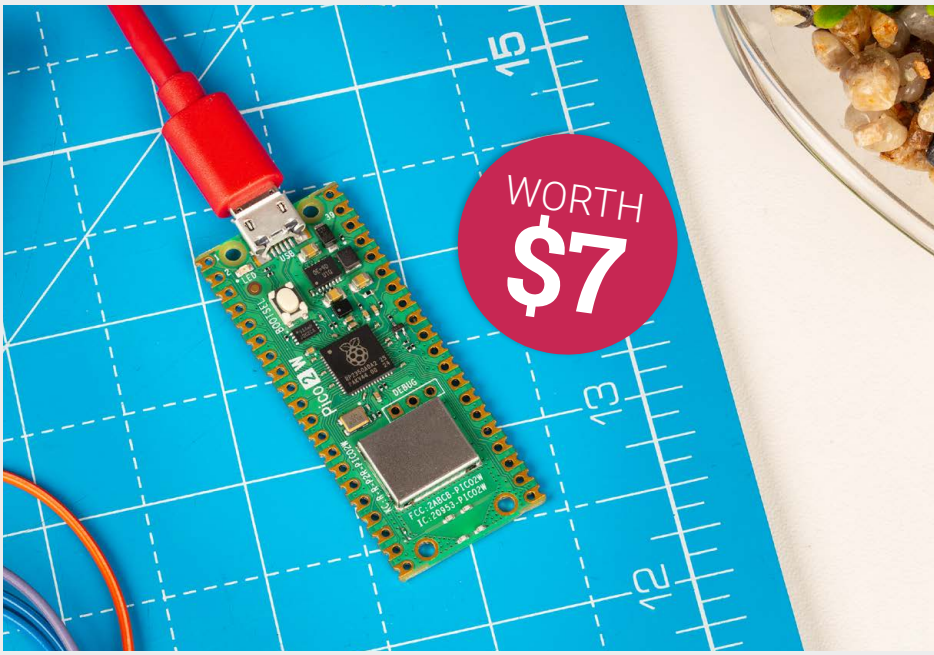
Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A choice of free Raspberry Pi Pico 2 W or Pico 2 is included with all subscriptions.

SUBSCRIBE TODAY AND GET A

FREE Raspberry Pi Pico 2 W (or Pico 2)

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

- A brand new RP2350-based Raspberry Pi Pico 2 W / Pico 2 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



Free Pico 2 or Pico 2 W. Accessories not included. This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

Buy now: **rpimag.co/subscribe**

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

Typeframe PS-85

Inspired by the Alien franchise, this cyberdeck is out of this world. By **David Crookes**

Maker

Jeff Merrick

Jeff is a designer and software developer who builds retro and sci-fi inspired models, props, and cyberdecks in his spare time.

rpimag.co/ps85

Maker Jeff Merrick didn't originally set out to create a cyberdeck inspired by the *Alien* movie universe. Instead, he sought to modify Penk Chen's Penkesu retro-style handheld PC by fitting a hot-swap keyboard PCB, but his attempts at producing a writerdeck hit a wall. "I quickly discovered that a 40 percent keyboard isn't ideal for typing and a Raspberry Pi Zero isn't ideal for browser-based text editors," he says.

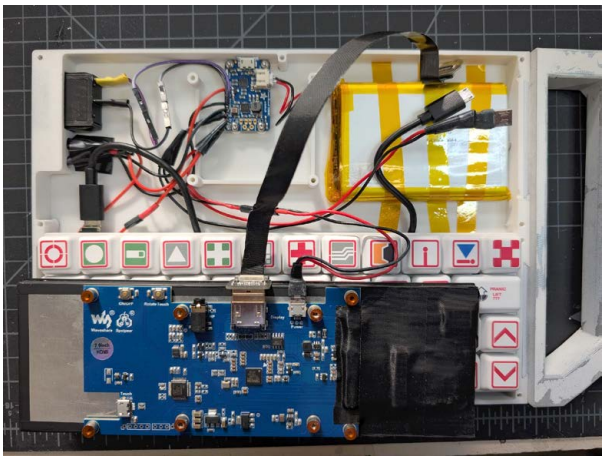
Scrapping the idea, he turned his attention to a new build which he named the Typeframe PX-88. Inspired by the Epson PX-4 computer from 1985 and built around a Raspberry Pi 4, it featured a mechanical 65 percent keyboard as well as a touchscreen. It worked so well that Jeff decided he wanted to develop more projects with Raspberry Pi. "I had not worked with any single-board computers before (or Linux in years), but found Raspberry Pi's large and established ecosystem made figuring things out easier," he tells us.

System ready

Jeff had purchased a set of keycaps imprinted with the Semiotic Standard – futuristic symbols created by Ron Cobb for the *Alien* films. "I didn't have a clear plan for how I was going to use them, but my wife and I are *Alien* fans," he says. "We've built some cosplay and props including the Incinerator Unit from *Alien*, the pulse rifle from *Alien: Romulus*, and a facehugger mask."

Since he still had his Penkesu build, Jeff decided to resurrect the project using the new keycaps. "I had most of the internals already working from my Penkesu build and I just wanted to use them in some way," he explains. "I also knew I wanted to keep it simple after designing my larger writerdeck, so I went with a slate form factor because hinges complicate everything."

As such, his Penkesu evolved into the PS-85 with a new 3D printed case and a Raspberry Pi Zero 2 W inside.



▶ The innards fit neatly within the slate case and the connections are wired for easy external access

01. The Waveshare 7.9-inch HDMI LCD has touch capability, but it's not used in this build

02. The 40 percent mechanical keyboard is fitted with *Alien*-inspired keycaps

The parts and wiring are similar to the Penkesu, but the keycaps set the theme, leading to some fresh ideas. "There's an exposed GPIO, so I knew I wanted some sort of peripheral that could slot in," he says. "I found the Adafruit CharliePlex LED Matrix Bonnet and this led to the addition of a light attachment."

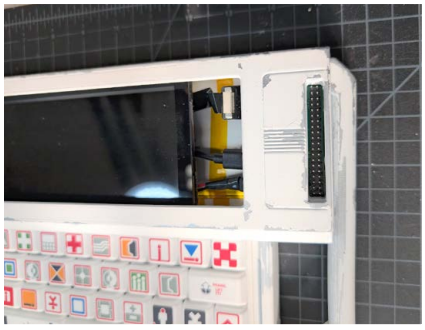
Creating the look

The build, which includes a Waveshare 7.9-inch touchscreen, wasn't entirely smooth. "It was a challenge to learn everything on the electronics/hardware side: CAD and soldering," Jeff says. "But since this was my second project, it went a little smoother than the first." The results exceeded his expectations. "It's actually much sturdier than I expected and it boots up pretty fast," he adds.

▶ A removable LED panel is fitted in the corner, representing the AI Mother System MU/TH/UR

Even so, this build isn't purely about the internals and a lot of time was spent on the overall look. "The idea of adding a handle came after an initial iteration," Jeff says. "I also used the same weathering techniques that I had used on some other *Alien* props: essentially I painted everything silver, then painted the areas I wanted to look chipped with a liquid mask (I used toothpaste), painted everything white, and finally removed the liquid mask to reveal the silver underneath. Then I added some scratches and an acrylic wash to finish it off."

The PS-85 looks so good that Jeff hasn't yet used it for its intended purpose. "My plan was to use it for journaling and there is a nice command-line application, jrnl (**jrnl.sh**), I installed," he says. "But it's mostly a display piece right now."



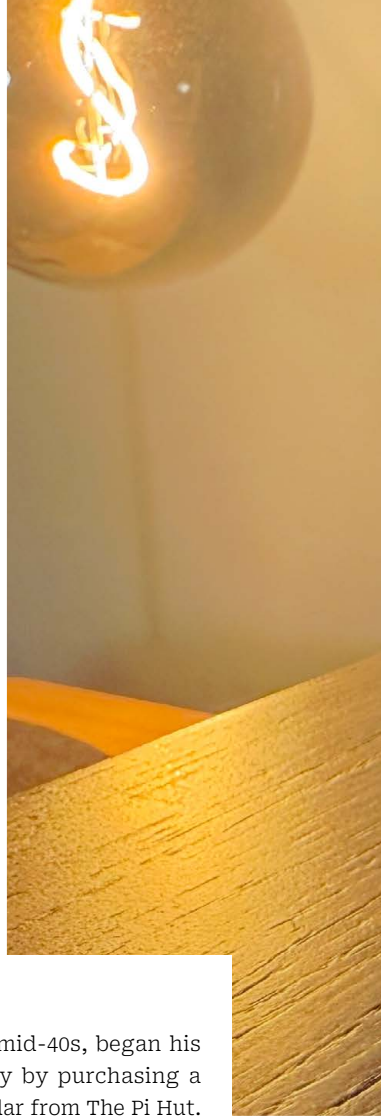
Quick FACTS

- The *Alien*-inspired build is deliberately weathered
- It includes a Raspberry Pi Zero 2 W
- It runs a command-line only interface from Raspberry Pi OS Lite
- The case has been 3D printed
- It's all powered from an Adafruit PowerBoost 1000C

RGB LED Ring Clock

This modern eye-catching clock tick-tocks all the boxes.

By **David Crookes**



Maker

John Dovey

John is a former secondary school science teacher and a current tinkerer and maker.

rpimag.co/ledclock

Although time marches on, John Dovey is proof that it's never too late to learn something new. When he and his children found themselves at a loose end during the Covid lockdowns, they began working on a small model railway and it put John on track to an entirely new hobby.

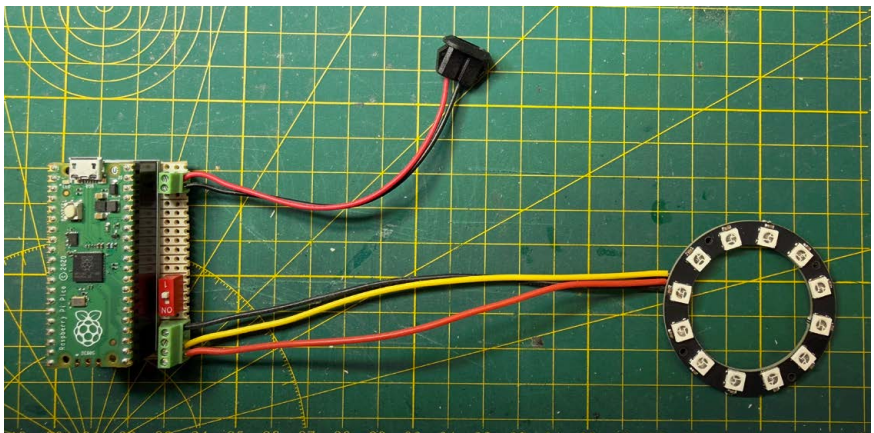
"I particularly enjoyed making simple circuits to make the buildings light up – just an LED, a resistor, and a battery," he says. "But this got me thinking. Could I make devices that allowed the LEDs to do certain things at certain times? Discovering I would need a microcontroller, I decided to go for a Raspberry Pi Pico because it seemed to be the best for a beginner. Most importantly, the guides I saw online seemed simple enough for a complete novice."

Perfect timing

John, who is in his mid-40s, began his Raspberry Pi journey by purchasing a Maker Advent Calendar from The Pi Hut. This gave him a Raspberry Pi Pico H and a bunch of projects to explore. "It has everything I needed to explore all sorts of physical makes and I was quickly amazed at how easy it was. Since then, I've kept learning and making."

Before long, John had an idea to create a fun, eye-catching clock. "I like to make physical items that have a practical function," he tells us. "I had a 12 LED NeoPixel ring and it just occurred to me that making a clock would be an interesting challenge; one that would make a nice addition to my desk."

To turn the ring into a clock, John purchased a Waveshare RTC module for



▶ The clock is fitted with a toggle switch. When it's on, the hour advances for Daylight Saving Time

> *I'd like to make a much larger version with 60 LEDs for my living room wall*

01. The project features a simple 12-hour analogue clock display

02. A blue LED is used to show the hour hand. Red/green is used for the minute hand

Quick FACTS

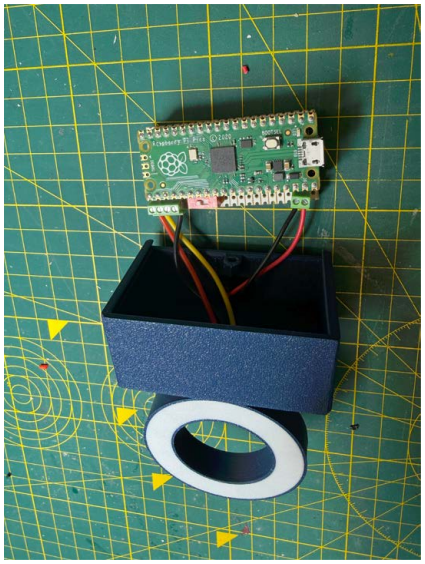
- It uses a Waveshare DS3231 real-time clock module
- The display is a 12 LED WS2812B NeoPixel ring
- The code has been written in MicroPython
- John designed the project to be beginner-friendly
- He has released the code and 3D printing files

Raspberry Pi Pico and he began to work on the code. He decided he would have the LEDs light up in different colours, with the hour hand being blue and the minute hand cycling from red to green as each minute passed. "The whole thing was fun as a puzzle for me to work out," he recalls. "Hours were fairly easy, but I had to work out how I could write simple code that converted the minutes to a specific LED."

Second wind

John also pondered what would happen when both hands were in the same space. He decided to make the LED cycle through a series of colours at this stage. "Once I'd made the basic clock, I then improved it by adding a physical switch to advance the hour by one for daylight savings time," he says. "I also added a USB-C power input so I could power the device from a more convenient place."

Since the Waveshare RTC module also includes a coin cell battery, the clock can keep time when it's unplugged. The clock also looks great thanks to a neat 3D-printed desktop design, but John says there is still room for improvement. "I'd like to make a much larger version with 60 LEDs for my living room wall and I'd also like to have a version with a button to turn it into a Pomodoro timer," he says. "I'd also like the project to inspire other people who don't think of themselves as coders to make things like this."

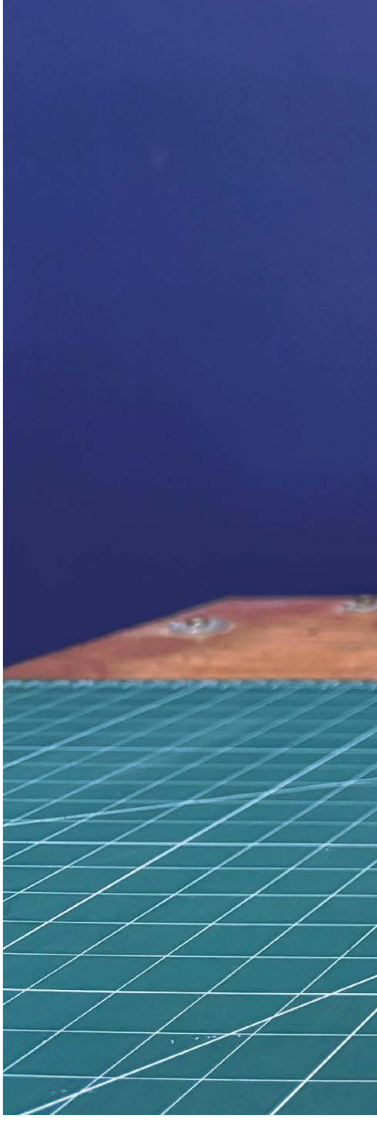


▲ Raspberry Pi Pico and the Waveshare Precision RTC Module fit neatly into a 3D printed case

VAULT-TEC Air Terminal

This Fallout-inspired terminal is a working computer that can also measure air quality.

By **David Crookes**



Maker

Arnov Sharma

Arnov describes himself as "just your average maker", but his talents for electronics, embedded systems, and PCB designing have led to some outstanding projects.

rpimag.co/airterm

Thanks in part to its retro-futuristic aesthetics, the survival video game Fallout has proven to be rather inspirational for many makers. Arnov Sharma is certainly a fan, having recently played Fallout: New Vegas. "In Fallout, there are so many things I want to make, one of them being a computer terminal," he says.

It led to the creation of his robust, industrial-looking VAULT-TEC Air Terminal containing a Raspberry Pi 5 paired with the Raspberry Pi M.2 HAT+. Running the Ubuntu operating system, Arnov can simply use the terminal as a computer. By using an application called Cool-Retro-Term, however, he's been able to mimic the look and feel of old CRT screens, which matches the overall terminal aesthetic.

Futuristic charm

And what a beauty it is. Arnov initially sketched out the concept by hand. Once done, he spent a lot of time using Fusion 360 – now known as Autodesk Fusion – to model the eye-catching 3D case into which he would be able to fit all of the necessary components. "There wasn't any specific feature I was aiming for," he explains. "I mainly wanted the model to resemble, or at least loosely follow, the design language of the terminals from the game. I also added some greeble details to improve the overall appearance."

He printed the case using Creality Hyper PLA, a high-speed 3D printing filament, but there was a lot of tedious work to be carried out. "I started by applying automotive filler all over, then sanded it down and repeated that process twice to get a smooth finish," he says. "After that, I added a few coats of primer and sanded again until the surface was ready for paint." At first, he used several coats of beige to give it a classic 1990s computer look. "But since Fallout terminals are supposed to be more than 200 years old, I wanted to add some weathering and patina, so I sanded the

The number of bars increases as the air-quality readings rise inside the room

edges and a few random areas to create wear, then used a sponge to dab on brown acrylic paint to mimic rust, along with a bit of green for mould. Finally, I sealed everything with a clear coat."

In the end, it turned out looking like an old, worn-out terminal. "I actually took some inspiration from Adam Savage's builds, especially his Blade Runner blaster prop video," Arnov adds. It was then time to get to work putting everything together.

More than a computer

The case needed to have space to fit a Waveshare 4-inch HDMI square display (chosen for its similar aspect ratio to an old CRT TV) as well the Raspberry Pi 5, M.2 HAT+, and an M.2 NVMe SSD. The SSD was flashed with Ubuntu, which significantly sped up the running of the operating system compared with writing it to a microSD card. This aspect of the build was straightforward enough, with all of the parts neatly fitting together, but the more complicated part was yet to come.

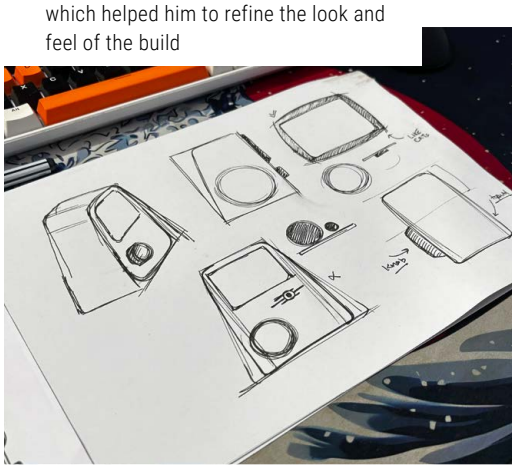
Arnov didn't want his Fallout terminal to simply work as a computer. He also decided to add an indoor air-quality meter. "In Fallout, the Pip-Boy displays player stats, inventory, quests and, most importantly, RADs (radiation levels)," he says. "I wanted to bring that feature into my terminal project but instead of measuring radiation, I used an SGP40 indoor air-quality sensor housed inside a PG7 connector-based probe to display the levels of Total Volatile Organic Compounds (TVOC) in the immediate environment."

Rather than have the information presented on the Waveshare display, Arnov decided he needed another hole in the case for a second display. He chose a TTGO T-Display S3 Long, which is an ESP32-based development board

01. The build is running a customisable terminal emulator called Cool-Retro-Term via Ubuntu

02. This long display is positioned vertically to allow air quality to be displayed using bars

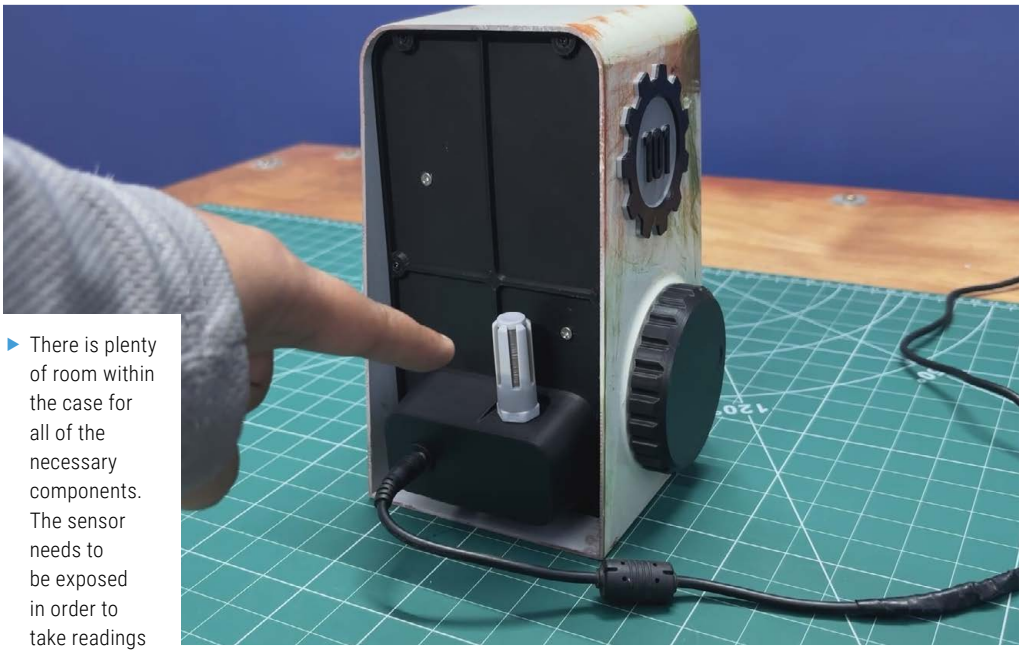
▼ Arnov sketched out his design on paper, which helped him to refine the look and feel of the build



powered by an ESP32-S3R8 dual-core LX7 microprocessor. It has 16MB of flash and 8MB of PSRAM, as well as a 3.4-inch capacitive TFT LCD. By opting for this device, he figured he could connect the sensor directly and not have to bother the Raspberry Pi 5.

As such, the TVOC data is shown on the long, 180 × 640 secondary display and Arnov set it up so that it would appear as a series of on-screen bars. "I implemented ten yellow bars that match the Pip-Boy UI from Fallout: New Vegas," Arnov says. "The number of bars increases as the air-quality readings rise inside the room."

To test the function, he lit an incense stick and placed it close to the sensor before seeing the TVOC reading soar. "By adding this feature, the terminal is no longer just a Raspberry Pi computer," he says. "It's also a functional device that monitors the air quality of the room."



There is plenty of room within the case for all of the necessary components. The sensor needs to be exposed in order to take readings

A new world

For fun, he labelled the readings as RADs, even though the unit is not actually sensing radiation. He also left space in the case for a side speaker which has been pressure-fitted, along with a printed speaker grill, allowing it to sit slightly flush. To add to the aesthetic, he also produced a red front knob which has been positioned on the front of the case. Logos were also created, adding to the authenticity of the design.

Arnov didn't, however, add a keyboard, even though the terminal is rather a large device with more than enough space to fit one. This was due to an early desire to produce a smaller overall unit. Arnov believed a keyboard would have looked out of place, but he's now considering a second version that would incorporate a keyboard and perhaps some other Fallout-inspired features.

He's more than pleased with the first version, though, and it really does look mightily impressive. "The whole project was a challenge, from construction to electronics, but in the end, everything came together really well," he affirms. Arnov also loved creating something from the Fallout universe so much, he's pursuing more game-inspired projects. "I will be posting details of these soon and, like the VAULT-TEC Air Terminal, everything will be documented and open source so that people can try to build their own versions," he says.

Quick FACTS

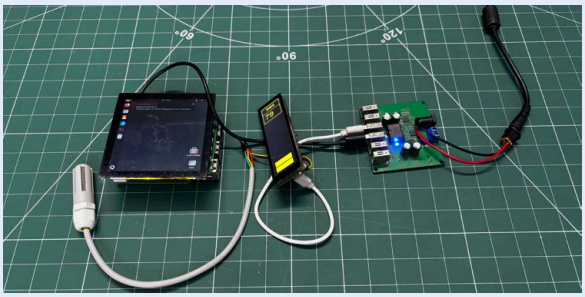
- It's a Fallout-themed functioning computer
- Raspberry Pi 5 is running a retro-style Ubuntu interface
- Real-time air quality is also being measured
- Arnov designed and 3D-printed the case
- It's been designed to be faithful to the game

The case was designed with a bezel surrounding the four-inch screen. This, Arnov says, resembles old CRT monitor bezels

▲ The whole build took more than 20 hours to complete and it required a lot of hard graft to perfect the look

Build a Fallout-style terminal

1. The terminal forms the core of the build. Arnov installed Ubuntu on a Raspberry Pi 5 4GB, then used **pi-apps.io** to install Cool-Retro-Term. This does a great job of making the four-inch screen look like a CRT display, complete with static, flickers, blur, and glows.



2. The Raspberry Pi 5, M.2 HAT+, M.2 NVMe SSD and screen easily fit together. What's more, the screen doesn't require configuring or driver editing. The I2C pins of the SGP40 sensor are connected directly to the I2C connector of the TTGO T-Display board.

3. The ESP32 communicates with the sensor using SDA and SCL lines. Code was prepared and flashed with Arduino IDE to process and display the data on the secondary screen. The entire build is powered using an external power adapter. It's rated at 12V 4A, but a DC buck converter steps it down to a stable 5V 8A.

OVCS – concept electric car

Raspberry Pi helps power this Frankenstein EV made up of different vendor parts. **Rob Zwetsloot** takes it for a spin

Maker

Marc Lainez

Marc and collaborators Loïc and Thibault own an IT consultancy company and have been working together for a decade.

rpimag.co/ovcsgit

The electric car revolution is just about here. With electric car charging points found nearly everywhere and more established car manufacturers introducing EV variants of existing models or straight-up new cars completely, a petrol-free future seems nearer and nearer. Of course, as a reader of this magazine you will be aware of how technology companies can be, and won't be surprised to know that there is a lot of proprietary tech used by each manufacturer.

> *Without this Raspberry Pi, the car wouldn't be driveable*

"Our project, OVCS [Open Vehicle Control System], aims at breaking the traditional vendor lock-in that you see in cars and other vehicles," Marc Lainez tells us. "We want to make it possible to interface parts from different brands together as if they were always meant to be working that way. Most car parts have a universal functionality to perform (braking, steering, showing data…) but the language they speak is different. So we thought we could build such a platform that would allow tinkerers like us to extend or swap a vehicle's functionalities with parts from any brands."

Marc and his team have a prototype of this platform, which uses Raspberry Pi to translate between the different parts.

Cross-compatible

The team had been looking for a larger-scale hobby project they could sink their teeth into which would combine all their various interests, and EV cars ended up being the centre of the Venn diagram.

"[We] were growing concerned with the security and reliability of vehicle software platforms," Marc says. "We thought this was the perfect project to learn a

01. A VW Polo was converted to be an electric vehicle, which the team used as an excuse to overhaul the electronics

02. The team went as far as making the car driveable via remote control

ton about how car parts communicate, how they interact together, and how we could seamlessly integrate parts together using modern languages on off-the-shelf hardware components like Raspberry Pi."

Raspberry Pi itself is used in multiple ways in the concept: first, the vehicle management system, which they describe as the 'brains of the platform'.

"It translates messages from the different communication buses (CAN) and routes them to the appropriate ones," Marc explains. "In total, we have five CAN networks that are being accessed through SPI modules connected to a Raspberry Pi. Without this Raspberry Pi, the car wouldn't be driveable."

It's also used in the infotainment system, something we've seen several folks do with Raspberry Pi in cars in the past. Not only does it show all the usual info about your vehicle, it also includes a touchscreen automatic gear shift as the car is an EV conversion.

Finally, there's the radio bridge: "[It's] a component connected to the CAN bus

Quick FACTS

- The team usually works in web development
- The vehicle is currently not road-legal
- The team found that setting specific milestones helped development
- One of these milestones was simply 'get the wheels spinning'
- Arduino and other microcontrollers are used

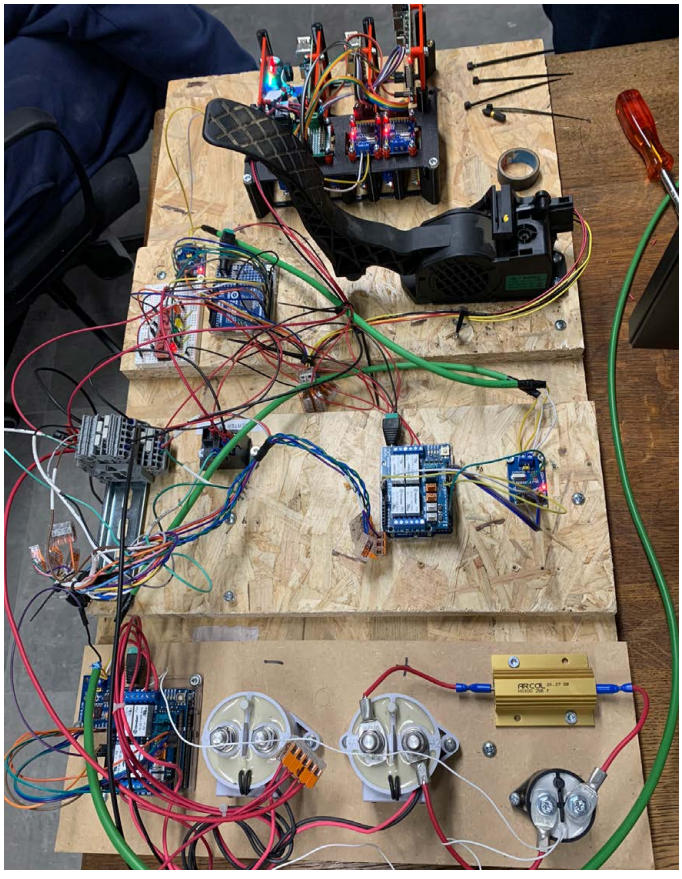
Warning!
Electrical and car safety
A lot of high-powered electronics were used in the making of this project, involving a full-size car. We do not recommend trying to recreate this without training or expertise.
rpimag.co/electricalsafety

◀ A custom steering column was used, so why not use a very serious racing wheel?

▼ The prototype was built on wood before any modifications to a real car happened

and sends instructions to the VMS to accelerate, brake and steer," Marc says. With it, they can control the car remotely.

"From a software perspective, we wanted to have a technology stack that was familiar and at the same time, something 'batteries-included' that would allow us to easily build firmware in a high-level language while at the same time making the firmware updates really easy," Marc continues. "Since we had done quite a lot of Elixir development, we used Nerves. This is an IoT framework built in Elixir and Erlang that relies on Buildroot (Linux build system) and gives you the ability to write your firmware in plain Elixir, a high-level functional language. It made our development cycles much faster/shorter and easier and allowed us to use a language we were already familiar with."



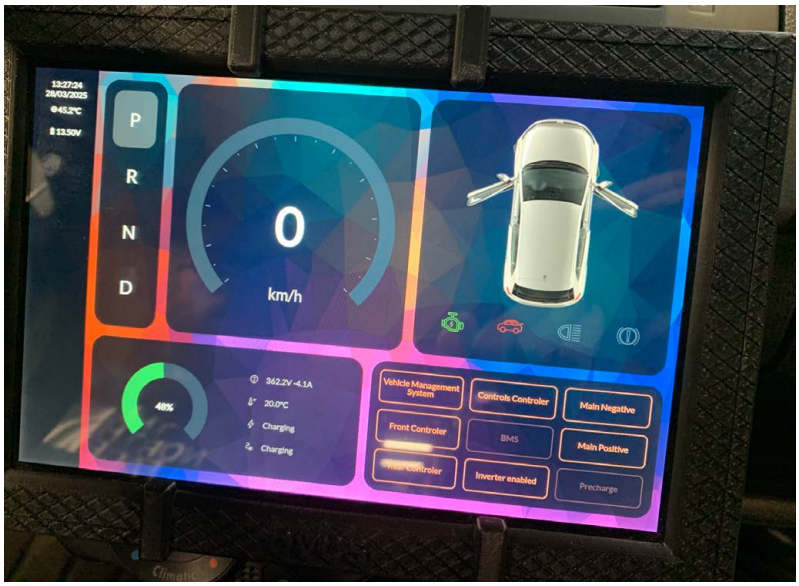
Put it in reverse

Getting the various parts to communicate, such as parts from a VW Polo with a Nissan Leaf electric motor, was one of the harder elements as the manufacturers generally

do not publish documentation on how their components communicate.

"We had to reverse-engineer quite a lot of messages in order to make the car functional," Marc reveals. "To give an example, if you want to know what message gives you the handbrake status (pulled or not), you look at all that is passing on the bus, you pull the handbrake a few times to see what frame is perfectly synchronised with your action to isolate its ID, then you check which bytes change when you pull it… . For more complex components, this is a combination of multiple messages and, fortunately, there is a community of car tinkerers who publish their findings on forums online. Sometimes the work was done; sometimes partially and we had to complete it."

Over the course of 18 months, the team did manage to make their 'Frankencar' driveable, which was their main goal, and then went beyond that by making it remote-controlled. They also wanted to document their build, which they're in the process of completing. After that? "The next goal is for the car to be self-driving," Marc says.

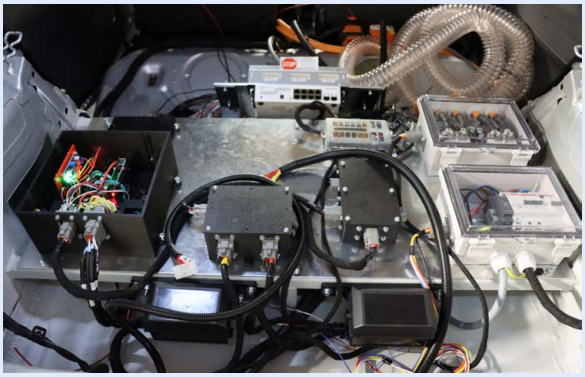


▼ The infotainment system, also powered by a Raspberry Pi, has a touchscreen gear selector

Communication issues

1. Cars have several communication buses dedicated to specific functions of the vehicle. The most common type is CAN.

2. Each bus publishes a 'frame', a series of bytes with an ID representing the relevant information for that bus.

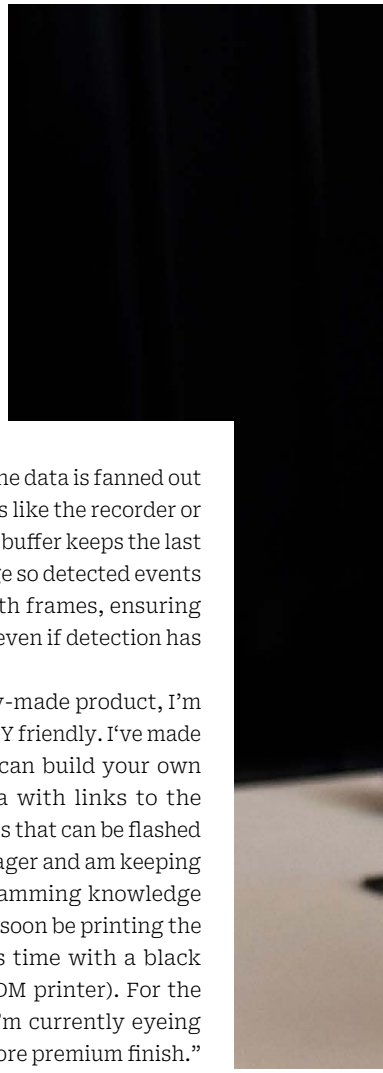


3. OVCS connects to existing buses, but sometimes new ones had to be created to control specific components.

ROOT Observer

By Paul Rhomberg

rpimag.co/Observer



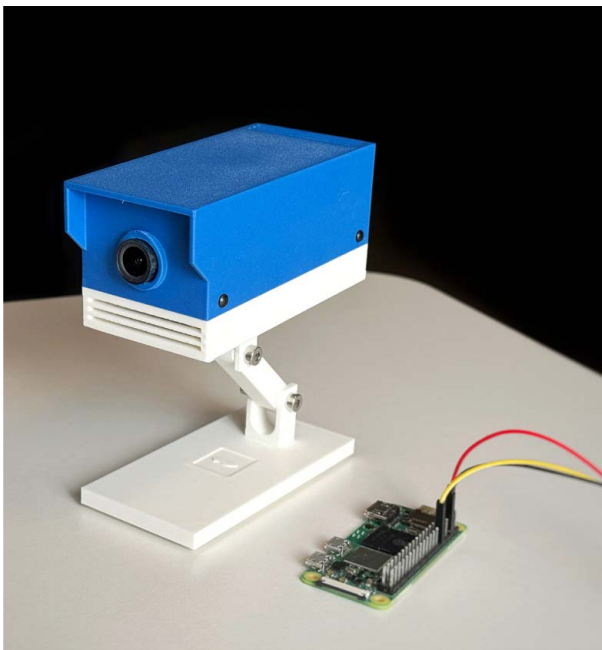
If you've read our cover feature this month, you're doubtless keen to cut down on the amount of your data that ends up in servers halfway around the world in territories you can't influence by the democratic process. That is part of what led the maker of this project to create the ROOT Observer: a privacy-focused security camera.

"The idea came from my parents asking me to find smart security cameras for their house a few years ago," says Paul, "so they could check in while on vacation. They had a local NVR (Network Video Recorder) setup, but it wasn't reliable or remotely accessible, so I ended up going with Google Nest. Not because I like Google's approach to user data, but because the alternatives were similarly bad from a privacy perspective. The few privacy-focused camera companies that exist tend to target enterprise, and even their privacy claims are hard to verify with closed-source software.

"Each camera is a standalone device storing all footage locally. Communication with the app (which is open source, as is the firmware and server code) goes through a relay server that acts as a WebSocket tunnel. Self-hosting is supported and the app makes it easy to swap in your own relay domain. Notification thumbnails are stored in S3, but fully encrypted with keys only the intended device holds, similar to how Signal handles end-to-end encrypted notification images.

"I started ROOT around mid-2025 and am currently focused on getting the firmware as performant and secure as possible, and making sure the app just works. After that, I'll dedicate more time to polishing the hardware and packaging. Raspberry Pi Zero 2 W is a natural choice because it packs enough power for local machine learning (person, pet, vehicle detection) without breaking the bank, and has great camera module support. Most of my time so far went into optimisation. The first iteration used FFmpeg and other heavyweight tools that performed poorly on the Zero 2 W. The latest version idles at around 10% CPU with ML and recording active, and can stream video and audio to up to three clients simultaneously, all fully end-to-end encrypted with unique shared secrets per client. Under that heavy load, the Zero 2 W only hits around 40% CPU and stays cool. The only slight hardware bottleneck is the Wi-Fi chip: rpicam-vid continuously captures from the Arducam IMX290 module, and the data is fanned out to multiple consumers like the recorder or live streams. A rolling buffer keeps the last few seconds of footage so detected events can be prepended with frames, ensuring complete recordings even if detection has slight latency.

"Beyond the ready-made product, I'm also keeping things DIY friendly. I've made a guide on how you can build your own privacy-first camera with links to the latest firmware images that can be flashed with Raspberry Pi Imager and am keeping it updated. No programming knowledge is needed for that. I'll soon be printing the third prototype, this time with a black upper shell (on an FDM printer). For the production version I'm currently eyeing resin printing for a more premium finish."



▶ Get AI features without having to share your data with gigantic corporations

ISS Tracker

By Filip Grace

rpimag.co/ISS-Tracker

◀ Space age is no longer a synonym for futuristic; space age means things that originated in the last millennium

Raspberry Pi is in the business of selling computers, so we really should be telling you to buy the latest and greatest, all-singing all-dancing models. However, this project uses older technology – over ten years old, in fact. Raspberry Pi 3 Model B was first introduced in February 2016. It seems like yesterday, but it was actually over a decade ago and this little machine has been given a new task keeping an eye on an even older piece of hardware: the International Space Station (ISS), which was launched at the back end of 1998.

To give his old Raspberry Pi 3B a purpose, Filip put it to work tracking the ISS. He's added a 3.5-inch Waveshare screen and impressively chunky toggle switch, and a case that's mostly made of 3D printed PLA with a machined aluminium top plate. He painted the aluminium plate and applied water decals so it looks to our earthbound eyes pretty much like the real thing.

Last issue, we featured in these pages a device that uses the open data coming from the ISS to track the levels of sewage held in the space station's tanks. Inwardly we shrink; it's extremely silly (though fun) to turn toilet humour into a science project, but we did hope to see before long something a bit more serious. With this build, Filip has used the same source of open data combined with a couple of manufacturing techniques, an impressive eye for detail, and a computer that if it lived in France would be considered old enough to legally drink alcohol. It's gorgeous and every home should have one.

Paper Console

By Travis Miller

travismiller.design/paper-console

Information overload is the curse of our times. We just aren't programmed to keep on top of so much data, on screens that zap our eyes with blue light and mess with our circadian rhythms. And so much of what data we do consume turns out to be dribble anyway. Fighting back against the constant online distraction is Travis Miller, with his device, the Paper Console. At its heart, this comprises a 58mm thermal printer and a Raspberry Pi Zero 2 W. It prints out information from 16 modules; these include puzzles, RSS feeds, news, a daily quote, shopping lists, weather reports, and more. What it does not do is suck you into the quagmire that is social media and the dopamine-driven search for 'one more click'.

Travis tells us that he "built the Paper Console because I wanted a way to get simple information like weather, headlines, grocery list, etc. without reaching for my phone first thing in the morning. Screens have a way of pulling you into ten other things you didn't plan to look at. A printed slip of paper does one thing and then it's done. You read it and move on with your day.

"I also wanted to build something hackable that other developers can extend with their own functionality, while remaining accessible to less tech-savvy users, like an ageing grandparent, who may just want something set and forget.

"I spent six years as a furniture maker before moving into digital design. This project is a bit of a merger of those two backgrounds. Inside is a Raspberry Pi Zero 2 W running a Python/FastAPI back end with a React settings UI. The code for the project is open source and runs entirely on your local network. No cloud accounts, no subscriptions."

Look at those physical controls! This device is built to be beautiful: it's made out of walnut and brass


WEATHER
MON · MAY 23, 2026
MALDEN
CURRENT WEATHER
33F
Overcast
H 39F · L 31F
Next 24 hours: Mostly overcast, with temperatures between 31F and 39F. Temperatures trend warmer through the day. Expect precipitation a slight chance around Now (up to 23% chance).
12-HOUR FORECAST
Now
33
23%
5 PM
33
17%
6 PM
33
17%
7 PM
32
8 PM


PROJECTS FOR MAKERS & HACKERS
BUILD A CUSTOM KEYBOARD WITH RASPBERRY PI PICO
CUSTOM CONTROLS WITH A 3D PRINTED BIG BUTTON
BOOK OF
MAKING
2026
CONTROL PROGRAMMABLE LEDS FOR DAZZLING LIGHTING
EVERYTHING YOU NEED TO KNOW ABOUT BATTERY POWER
FROM THE MAKERS OF RASPBERRY PI OFFICIAL MAGAZINE

BOOK OF MAKING 2026

STEP INTO THE WORLD OF MAKING!

- DISCOVER YOUR NEW FAVOURITE HOBBY
- LEARN THE DIGITAL SKILLS THAT MAKE THE WORLD GO ROUND
- TRY YOUR HAND AT 3D PRINTING, ELECTRONICS, PROGRAMMING, AND MORE
- DISCOVER PROJECTS THAT YOU CAN DO IN AN HOUR, AFTERNOON, OR WEEKEND

rpimag.co/BookOfMaking2026

3D print

Use a robot to build a robot

rpimag.co/6DOF-robot

Imagine you're an engineering student, using ROS (the Robot Operating System) to learn about robotics. At some point you're going to want to break free of the screen and the keyboard and start applying the knowledge you've learned to real hardware. That's what James Gullberg has done with this glorious, (mostly) 3D printed robot with six degrees of freedom.

The joints and the gears are all 3D printed and are connected by carbon-fibre tubes. This enables James to quickly iterate on the design of each joint without having to reprint the whole arm each time he makes a change. The modular construction and the fact that it's 3D printed offer the opportunity for lots of experimentation: the shoulder and elbow joints use split-ring planetary gearboxes, while the wrist joint uses an inverted belt differential. To sense the position of the joints, there are embedded magnets monitored by a magnetic encoder.

In this robot build, James has used a Raspberry Pi as a CAN bus. The 'CAN' stands for 'controller area network'. Rather than having all of the discrete devices (such as motors, and sensors on each individual joint) talk to each other directly, they instead talk to one separate device. It's a way of making intrasystem communication more efficient, reduces wiring, and gives a degree of central control over which messages are prioritised. For example, in a car, the messages coming out of the engine are of a higher priority than the messages coming out of an electric window. With this in mind, it should be no surprise that the CAN bus protocol was first used in the automotive industry, for the Mercedes W140 in 1991.

As well as the 3D printed body and gears and carbon-fibre tubes, a lot of the electronics were "salvaged from an old 3D printer" that was about to be thrown out. This brought the cost of the bill down to under $300.

TAKE BACK YOUR CLOUD

Ditch faceless multinationals and run your own file-sharing, online office, music streaming, and more!

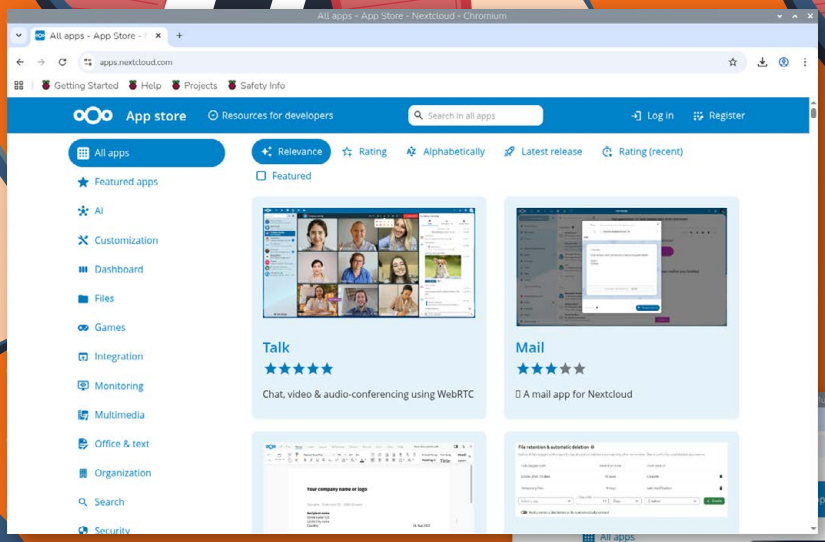
By **Ben Everard**

Maker

Ben Everard

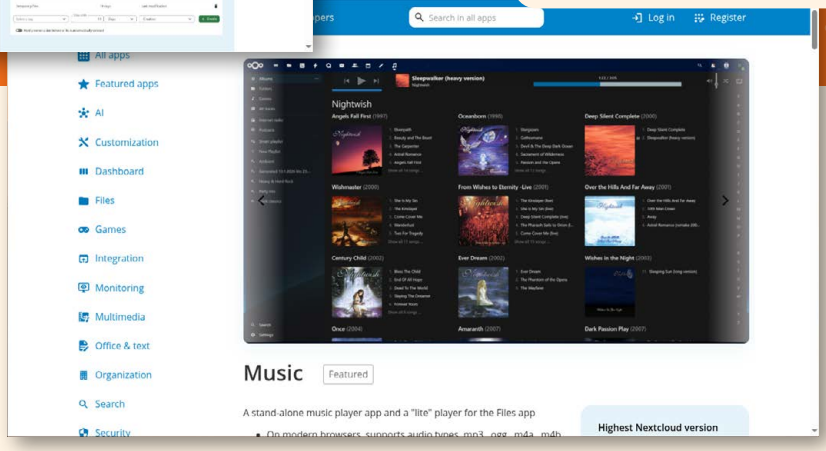
Ben is slowly becoming a cliché as he moves away from the fast-paced, ever-changing world of tech to the gentle embrace of woodworking.

youngwoodworkers.co.uk



◀ Nextcloud isn't a single bit of software, but a framework in which you can install apps. A few are installed by default, but there are many more. Take a look at **apps.nextcloud.com** for a full list

▼ Though not installed by default, the Music app lets you upload your music collection and then stream it to any device connected to your Nextcloud instance



Modern computing is wonderful. As long as you've got a web browser, you've got access to an office suite, tens of gigabytes of network connected storage, almost all the music ever recorded, and much more. It's all there in 'the cloud' and accessible for free (or a modest monthly fee).

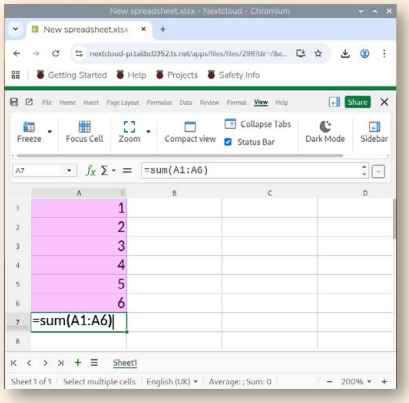
All you have to do is tick accept on some terms and conditions – nothing wrong with that, right? Sure, the firms running these services are some of the largest on the planet and most are prominent in advertising, AI, and other such data-heavy industries – there's probably no data privacy issues there, are there? And yes, many donate vast sums of money to political candidates with whom you may or may not agree. And yes, the data centres powering these systems are consuming vast quantities of power and fresh water while generating huge amounts of e-waste as hardware is regularly replaced...

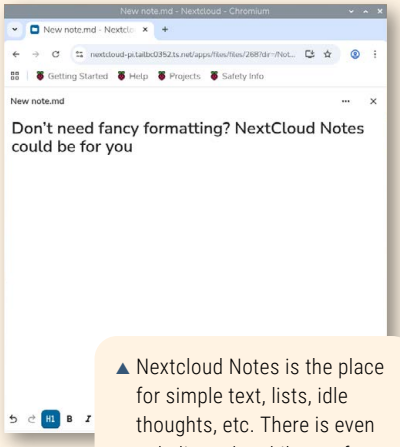
Actually, maybe the cloud isn't such a great deal.

Fortunately, it's possible to get the advantages of 'the cloud' while keeping full control of your data. You just have to run the services yourself! A low-cost, low-power Raspberry Pi is a great platform for this. Let's take a look at how.

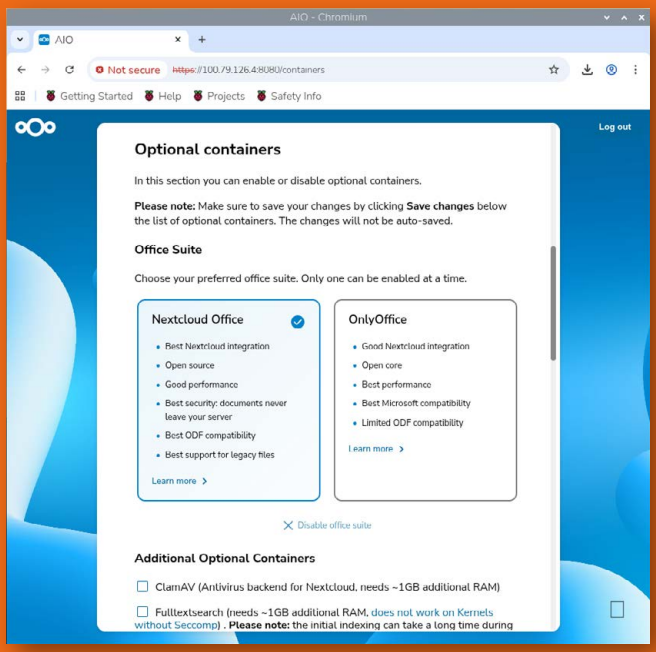
It's possible to get the advantages of 'the cloud' while keeping full control of your data

▼ Spreadsheet jockeys aren't left out. You can create and edit your documents in the cloud





▲ Nextcloud Notes is the place for simple text, lists, idle thoughts, etc. There is even a dedicated mobile app for keeping track of things on the go



◀ While Nextcloud apps can be installed through Nextcloud, if you need additional server-side software, this needs to be installed via the AIO interface

Why Docker?

Docker uses a technology called containers. This basically creates an isolated system within your operating system. It's so isolated that software within the container can't communicate with any software outside the container other than through network interfaces. This is similar to virtual machines, but the isolation happens at a different level.

The advantage of this from our perspective is that it lets us install a preconfigured system. For example, we can run a container that has a database set up exactly how Nextcloud expects it to be set up, and another container with a web server with the correct setup.

Docker helps us with all the configuration of this. It gives us access to a set of containers that are known to work, and it also has a set of tools to easily configure containers.

rpimag.co/docker

Installing the software

We're going to be running quite a bit of software, so ideally you'll have a Raspberry Pi 4 or 5 with quite a bit of RAM (4GB is recommended, though it might be possible to get away with 2GB provided you don't have lots of simultaneous users).

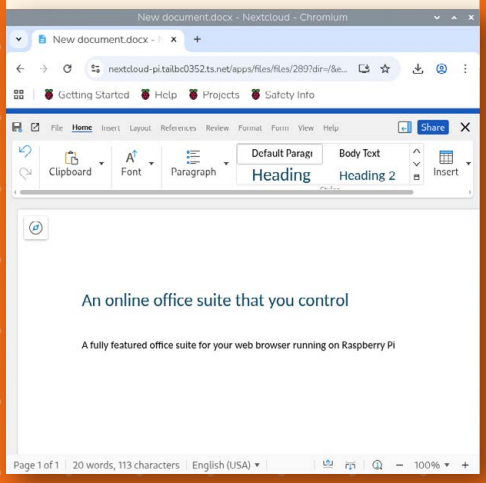
It's best to start with a freshly flashed SD card with the latest version of Raspberry Pi OS (these instructions may not work on older installations). We'd recommend that you enable SSH because once you've got the software up and running, there's no need to attach a keyboard and screen to your Raspberry Pi, but you might need occasional access for maintenance, etc. This can be either with password or certificate, whichever you prefer.

We'll use a lot of disk space, so ideally you should have an SD card of 32GB or more (the official system requirements state 40GB of storage, but 32 worked fine for us and left us with 14GB of space once everything is installed).

Nextcloud isn't a single piece of software – it's a whole system of interconnected bits

In this tutorial, we're going to go through setting up your first instance of Nextcloud. If it's something that you like and want to keep going with, then you'll probably want some storage that's faster and more robust than an SD card. There are a lot of options here, and the best choice depends on a lot of factors including how much space you want, how much budget you have, how important power draw is, and how important reliability is to you. However, before you get into all that, let's get a test system set up and running that you can use, and then migrate to whatever hardware you want to use in the future.

▼ The office suite is based on LibreOffice and should have enough features for all but the most determined power users



Next step: Nextcloud

Nextcloud isn't a single piece of software – it's a whole system of interconnected bits. You'll need a web server, a database or two, an engine to create the dynamic content... there's quite a lot. It's perfectly possible to set it up bit-by-bit but, well, it's a lot of work. Instead, we're going to use a system called Nextcloud All-In-One (AIO). This uses Docker to launch and connect everything.

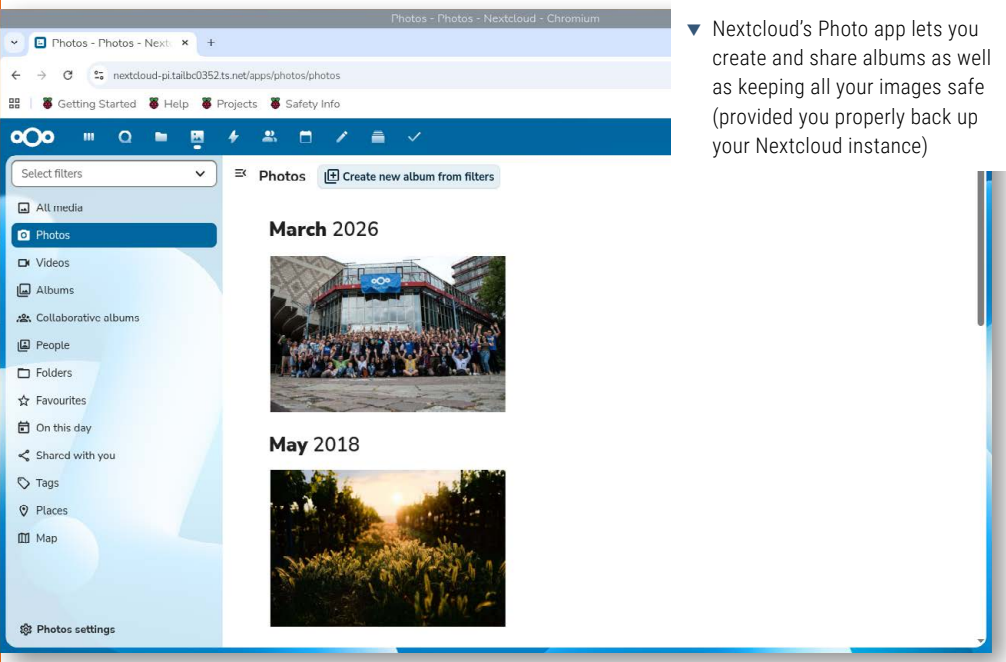
However, before we get to Docker, we do need to consider our networking. Nextcloud AIO is designed to be secure, using HTTPS to communicate. This means that everything we send to and from our Nextcloud server is encrypted. This is a good thing, but it does introduce a problem: certificates. HTTPS requires servers to have a certificate which includes their public key – basically a large number that acts a bit like a password to encrypt data sent to them. HTTPS certificates are linked to domain names, and most computers on home networks don't have domain names. There are a few solutions to this, but the one we'll go with is Tailscale.

This creates a virtual private network (VPN) between devices and can also manage domain names and HTTPS certificates.

Tailscale also handily solves a second problem that we'd hit later on: accessing the Nextcloud server from outside your home network. We won't go into the full networking wizardry involved, but basically any computer that you link to your Tailscale account will be able to talk to any other computer on your account. At the same time, this protects all the traffic between computers.

This is the same technology that other VPN providers use to let you bypass country restrictions on content viewing, but used in an entirely different way.

We'll need to create a Tailscale network with at least two computers (the Raspberry Pi that you'll be installing Nextcloud on, and at least one other – which can be Windows, Mac, Linux, or even a mobile phone – that you want to use to access Nextcloud from).



▼ Nextcloud's Photo app lets you create and share albums as well as keeping all your images safe (provided you properly back up your Nextcloud instance)

Heads or tails?

Much like Docker, Tailscale wraps up a lot of existing technology into a way that's easy to manage. You don't have to use the firm's services, and can manage everything yourself. We've used them here because it's an introductory magazine article, not a detailed book on Linux systems administration.

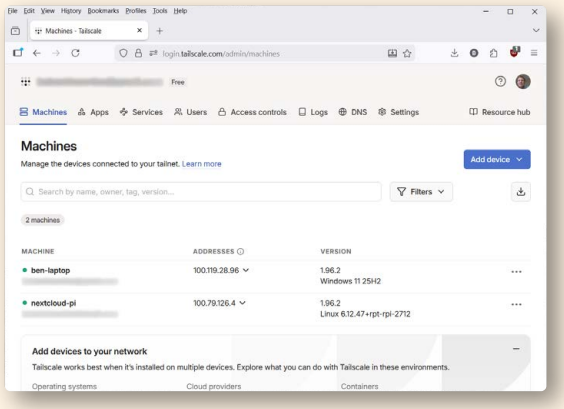
If you're familiar with VPNs, DNS, and HTTPS certificates, there's nothing to stop you going solo and running the whole setup yourself.

In fact, in the spirit of avoiding cloud services, there is even an open-source, self-hostable replacement for Tailscale called Headscale (**headscale.net**). To run this, you'll need a server with a static public IP address. We haven't tested it all out, but it should work, though you might need to manage the HTTPS certificates yourself using a service like Let's Encrypt (**letsencrypt.org**).

Networking made easy

Head to **tailscale.com** and click on 'Get started – it's free!' to create an account. Slightly annoyingly for an article about leaving cloud services, Tailscale accounts are linked to cloud accounts, so you'll need a Google, Microsoft, GitHub, or Apple account. If this is beyond acceptable for you, you can go down a more DIY route (see 'Heads or tails?' box), but it does complicate matters a bit.

Once you've got an account, follow the instructions on **tailscale.com/download** for each machine you want to connect to the network. As soon as you have at least two machines on the network, you should be able to see the machines that have connected and get access to the admin panel.



▲ Two computers attached to the author's Tailscale network. You'll need to attach any computer that you want to access Nextcloud, but you can do that later

In the Tailscale admin panel, select the DNS tab and scroll down the bottom. You need to ensure that both MagicDNS and HTTPS Certificates are enabled (they're enabled when the button says 'Disable').

Tailscale will take care of making sure the data moves safely between machines

This is all we need to do to set up the networking. Tailscale will take care of making sure the data moves safely between these two machines. Now, let's turn our attention to Docker.

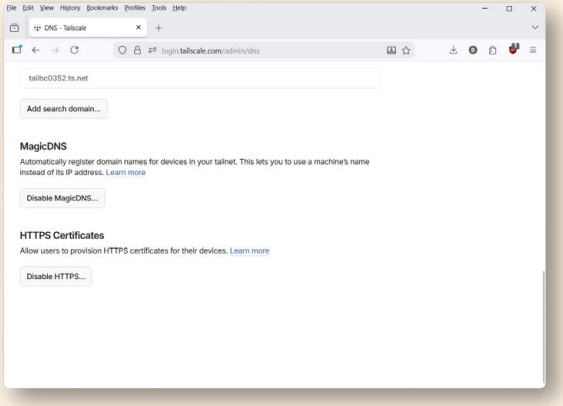
We're going to use Docker Compose, which takes a configuration file and uses it to launch containers and do more networking wizardry to link the various bits of software running in various containers together.

We know we're hand-waving a lot of detail away behind words like 'wizardry' here. There's actually quite a bit of complex setup going on behind the scenes, and we don't want to get bogged down in detail when we've got our own cloud ecosystem to bring online!

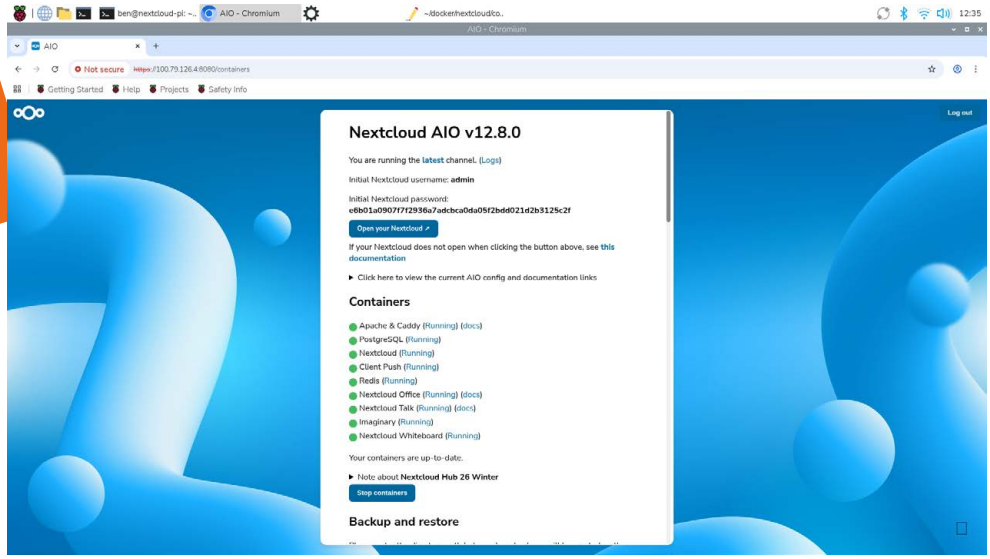
You need to make a directory that will contain the configuration files for your Docker containers. We've called ours **docker/nextcloud/**. Now download the file **compose.yaml** from **rpimag.co/nextcloudcomp** into that directory (keeping the same name – **compose.yaml**).

This is the standard Docker Compose file for Nextcloud AIO. We need to make a couple of changes before we can launch it, so open it in your preferred text editor and immediately under the line:

```
nextcloud-aio-mastercontainer:
```



▲ There are only two settings we need to change – enabling MagicDNS and HTTPS Certificates



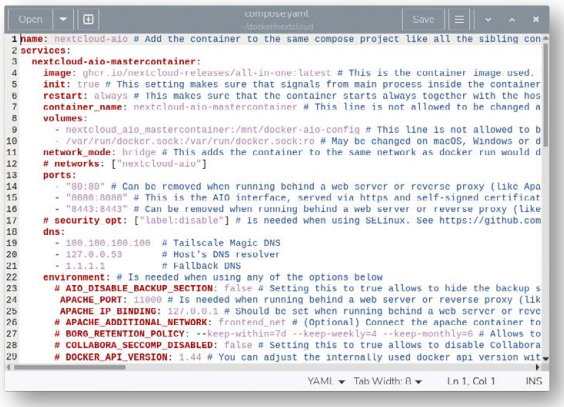
▶ Once you've got a full set of green dots, you're ready to start using Nextcloud!

...add the lines:

```
dns:
  - 100.100.100.100  # Tailscale Magic DNS
  - 127.0.0.53       # Host's DNS resolver
  - 1.1.1.1          # Fallback DNS
```

Note that the number of spaces is important. There should be four on the first line and six on the other lines.

Next, we need to uncomment the following lines. This means remove the # character at the start (only the first # – make sure to leave any subsequent ones). Again, spaces are important, so make sure you only delete the #.



▲ The Docker config file has loads of options you can use to tweak your system

```
APACHE_PORT: 11000
APACHE_IP_BINDING: 127.0.0.1
```

Once you've done that, you can open a terminal, navigate to the folder you've saved the **compose.yaml** file in, and run:

```
$ docker compose up -d
```

This will launch everything and get our software running. However, before we can use it, there's a problem. We need a domain. We've got most of the way there with our Tailscale, but now we just need to link that to our Docker container that we've just launched. We do this with:

```
$ tailscale serve --bg http://127.0.0.1:11000
```

Now we need the IP address of our Raspberry Pi, and we can get this from the Tailscale web page.

Once you have found this, point your web browser to **https://<your-ip-address>:8080**. Notice that it's 'https'.

On this web page, you'll get a passphrase. Make a note of this somewhere, as you'll need to enter it on the next page, so copy it. Now you can log in.

You'll need to enter the domain name of your server on the next page. You can get this by typing the following into a terminal:

```
$ tailscale serve status
```

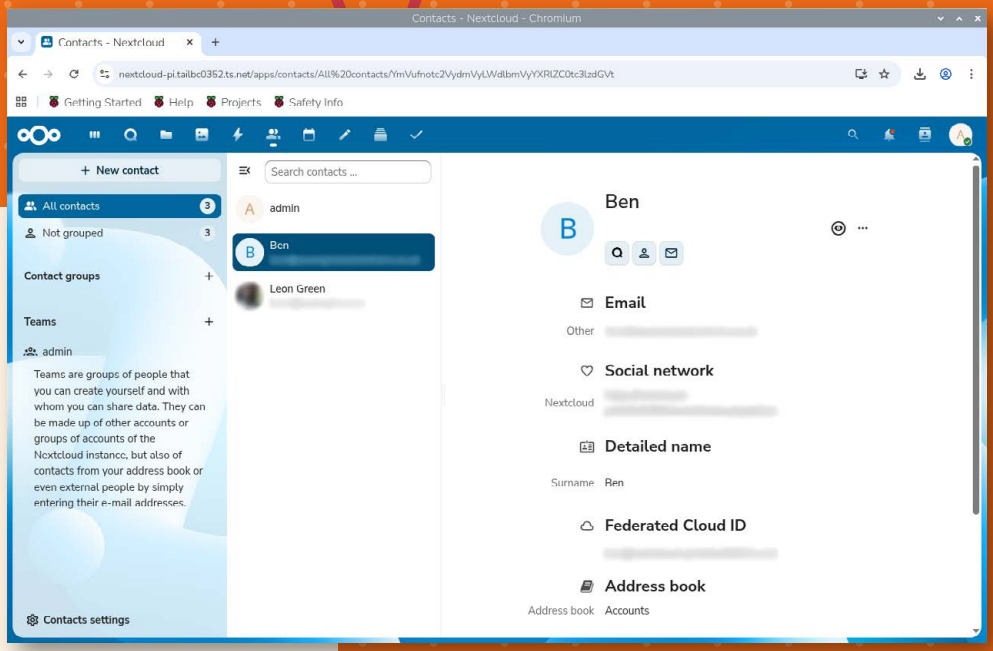
It should be something like:

```
your-machine.tailbc0352.ts.net
```

Now you can select the services that you want running, and we'd recommend adding Collabora Office. This gives you access to a full office suite that runs in your web browser. Once you've done this, it will get everything running. This can take quite a long time as it has to download and configure a lot of software.

When it's finished, you'll have green dots next to everything.

That's the basics of getting everything set up, and now everything's ready to use.



▶ Nextcloud plays nicely with other federated services (such as Mastodon)

You're now logged in and can use Nextcloud. However, you're logged in as the user 'admin'. This has full access to everything. It's a good idea to create a separate user (or indeed, several if you want to share your server with other people).

Click in the 'A' at the top right of the screen and select Accounts, then click New Account where you can enter details including the quota of disk space.

Now everything is all set up, you can just leave it running. However, from time to time you might need to restart your Raspberry Pi. Maybe that will be because of a software update, maybe it'll be because of a power cut, or maybe you just need to unplug your Raspberry Pi.

Inside the Docker configuration, we set the containers to always restart so whatever happens, when you power on the Raspberry Pi, they'll start up. The only thing we need to do is start Tailscale serving the domain.

You can just run the line:

```
$ tailscale serve --bg http://127.0.0.1:11000
```

...every time you turn the computer on. However, we can do it automatically using crontab.

Enter the following command in the terminal:

```
$ crontab -e
```

Then select your preferred text editor (we'll press **1** for nano)

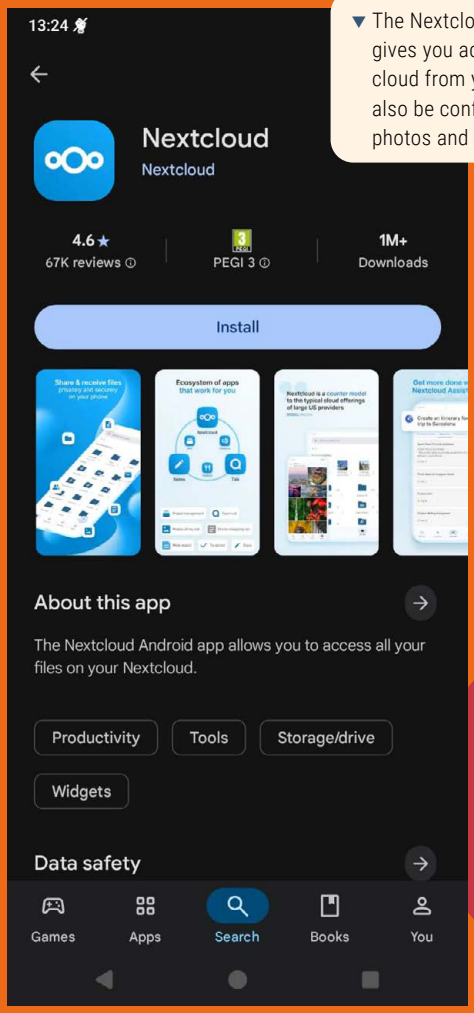
Now scroll to the bottom of the file and enter the line:

```
 @reboot /usr/bin/tailscale serve --bg
http://127.0.0.1:11000
```

And then (if you've used nano), press **CTRL+X** to save (use the same file name) and exit.

Now, whenever you reboot your Raspberry Pi, everything should come back online.

At this point, everything's up and running and you can tuck your Raspberry Pi away somewhere safe.



▼ The Nextcloud mobile app gives you access to your cloud from your phone. It can also be configured to back up photos and files automatically

Backup

If you're using Nextcloud with real data, then it's completely essential that you have a proper, tested and working backup procedure.

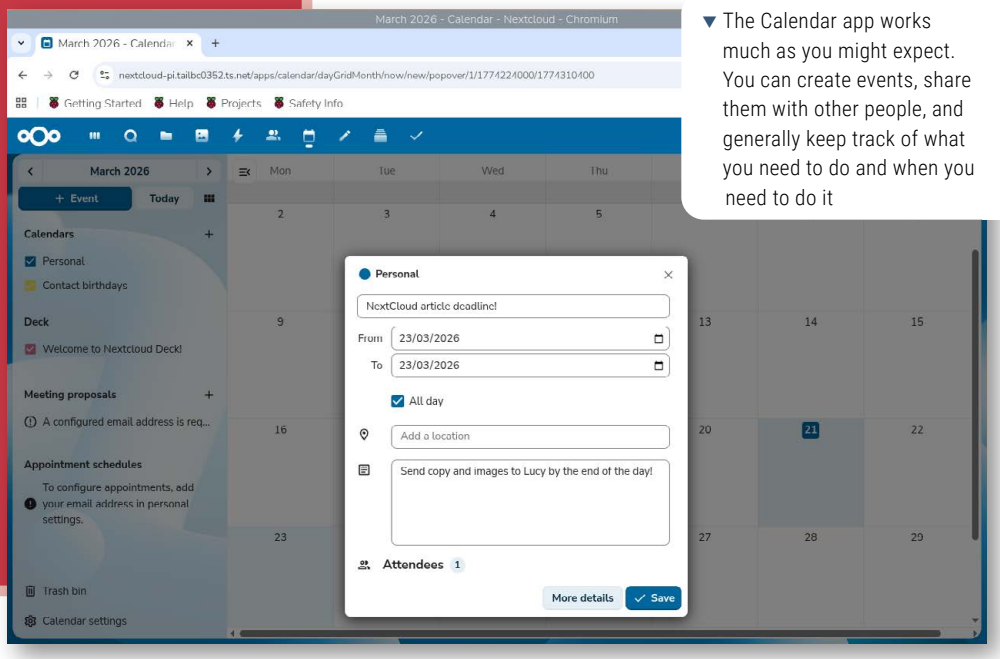
Nextcloud AIO manages backups using BorgBackup, and you can configure this in the AIO admin interface, https://<ip-address>:8080, that you used when first setting up Nextcloud.

You need to first enter a location, which can either be local or remote. Then click on Set Backup Location. Once you have a location, you can create a backup, and once you have a backup, you can schedule additional backups.

There are two important things to consider with backups. Firstly, an untested backup shouldn't really be considered a proper backup. It's not enough to run the backup task. At some point, you should test out the recovery procedure to check that it works and that you are actually backing up what you think you're backing up.

Secondly, there's no such thing as a perfect backup. It's a matter of balancing cost with risk. For example, you could have a separate USB drive attached to your Raspberry Pi and use this as a backup location. This will protect against any damage to your main storage device, but if there is, for example, a fire in your house, it will take out both storage devices and everything will be lost. For important data, an off-site backup should be a minimum requirement. Ideally, your backup should itself be backed up. How far you should take this depends on how lucky or paranoid you feel.

We set the containers to always restart so whatever happens, when you power on Raspberry Pi, they'll start up



▼ The Calendar app works much as you might expect. You can create events, share them with other people, and generally keep track of what you need to do and when you need to do it

E-paper display picture frame

Fuse the old with the new and make something unique!

Maker

Nicola King

Nicola is a freelance writer and sub-editor. Craving bright colours as we edge out of winter and a mindful mission to soothe the soul, this project hits the spot!

@holtonhandmade

QUICK TIP:

Take your time! Just enjoy the mindful process of painting and creating your own art.

In this month's arty-crafty-techy take on life, we'll be combining the classic with the very modern: traditional canal boat folk art blended with an electronic paper (e-paper) display. The beauty of this project is that you don't need to be an artist and you'll have a unique piece of art uniting the traditional with the state-of-the-'art' at the end of it. So, if this idea floats your boat, let's get underway!

Painting a picture

Let's first consider the size of our e-paper screen and the size of box frame that will be required. Our Inky Impression Spectra 4.0" screen (**pimoroni.com/impression40**) measures approximately 7.5cm by 10cm. Compatible with any 40-pin Raspberry Pi computer, it's a super-useful colour display as it's quicker to refresh than earlier colour screens.

It's worth noting that this might not be the cheapest of projects if you don't already own an e-paper display. Ours costs £49.50 ($54.86), but you can get black and white displays for a lot less – check out The Pi Hut for a range of options starting at £13.50 (£17.78): **rpimag.co/pihutepaper**.

Alternatively, you could use a screen from Pimoroni's Inky Frame range, powered by a built-in Pico W or 2 W. Whichever display panel you decide to use, you can repurpose it once you grow tired of your new piece of artwork!

Based on the size of our display, we chose a 21.5 × 21.5cm deep box frame, with an aperture size of 20 × 20cm. We then cut a piece of card to fit the frame, and drew a rectangle in the middle of

- Photos have a painting-like quality when shown on the e-paper screen
- Some of the required tools and consumables needed for this picture – it's likely that you'll already have some of these bits in your stash

You can use any e-paper display you like

the card that represented where the Inky Impression would sit; we cut it out so our screen could sit neatly in the hole. We then knew how much space we had to play with for our artwork.

For the latter, we used some quite fine round brushes (we had sizes 8, 7, 3, and a 1 to hand). We found the other end of the brush was very useful when it came to placing blobs on our paper. You will probably need to have a play around to see what size brush etc. works best for you.

After some practising on scrap paper, the time had come for us to actually paint something on our 'proper' piece of card. Now, when we say 'paint', what we actually mean is placing some blobs of paint on the card, and then either pulling the paint into the shape we wanted, or mixing it with another colour to create a flower. For example, a green leaf is simply two blobs of paint a few millimetres apart, and we then used our brush to pull each

YOU'LL NEED:

- A shadow box frame
- An assortment of bright/colourful acrylic paints (green, red, white, yellow)
- Selection of precision paintbrushes (e.g. **rpimag.co/brushset15**)
- Palette and water
- Practice paper/card
- Quality card (to paint on and insert in frame)
- Pencil, rubber, and ruler
- Scissors
- Glue/masking tape
- An e-paper screen
- Raspberry Pi computer (for our example)

QUICK TIP:

A monochrome e-paper display will refresh far more quickly than a colour one.

Electronic paper: a brief overview

Sometimes also known as intelligent paper, electronic paper, in its most simple terms, is a reflective display technology which is designed to mimic the ink that we would see on paper. These e-paper displays are glare-free as they reflect ambient light and don't emit light, have a high contrast, and they require hardly any power; in our piece of artwork, the screen only uses power when/if the image changes.

Possibly the most famous name in this area of technology is E Ink, the trademarked electronic paper technology. This is a brand of e-paper commercialised by the E Ink Corporation, which was founded by three visionaries from MIT Media Lab who created a prototype of electronic paper back in the late 1990s.

QUICK TIP:

If your frame has glass protecting your artwork and screen, all the better. E-paper screens can be fragile, so a glass-covered frame will help.

blob down slightly to give the shape of a leaf. A flower is also very easy: two contrasting blobs of colour next to each colour, then swirled together to give the impression of a flower. We tried it with red and a very light lilac colour.

Alternatively, you could create a larger rose by painting a circle with strokes going in a spiral direction, and while that dries, creating two green leaves coming out of it. When the circle is dry, using a darker/contrasting colour, start in the centre of the circle and make some small curved strokes, working outwards and making them staggered and slightly larger as you work towards the outer edge. Try it with yellow and orange or red and pink as contrasts, and you'll soon be admiring your own hand-painted roses. Basically, whichever way you decide to paint your roses, adopting a loose stroke, abstract approach is just fine... we don't need to get obsessed with too much detail here.

Basically, cover the blank area of your card or paper with painted flowers and greenery in colours of your choice. Make a random pattern, or something more considered in terms of layout – the choice is yours. We can now think about the tech, creating some code and adding our screen.

E-paper caper

You can use any e-paper display you like and cut out the rectangle (as drawn earlier) in the card for it to fit in the middle. It can be secured with masking tape on the rear to keep it perfectly in place.

Our Inky Impression Spectra 4.0″ screen requires a Raspberry Pi computer to be connected to its female GPIO header on the rear to control it. To save space, we used a Raspberry Pi Zero 2 W.

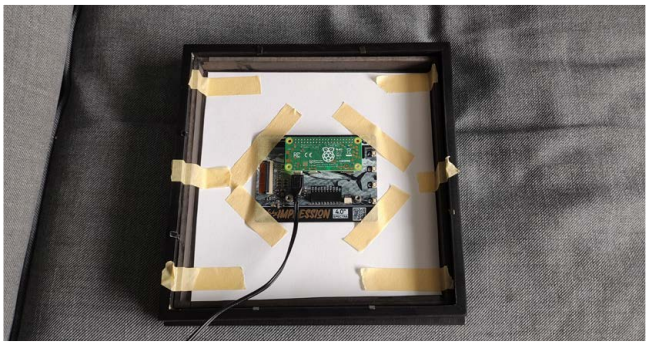
We found some Creative Commons-licensed photos of canal scenes. For best results, you may need to boost the contrast and saturation of your images. Also, for a colour display, look for images with a limited palette of vivid shades.

After installing the Python libraries required for the Inky Impression (see **rpimag.co/inkyguide**), we adapted one of the Inky library code examples to show a series of images...

▲ After a bit of practice, even a non-artist can create some appealing floral designs

```
#!/usr/bin/env python3

import time
from PIL import Image
from inky.auto import auto

# Auto-detect display
inky = auto(ask_user=True, verbose=True)

# Tweak saturation
SATURATION = 0.5

# List of images (edit paths as needed)
pictures = [
    "images/image1.jpg",
    "images/image2.jpg",
    "images/image3.jpg",
    "images/image4.jpg",
    "images/image5.jpg"
]

def show_image(path):
    print(f"Displaying: {path}")
    image = Image.open(path)

    # Resize to match display resolution
    image = image.resize(inky.resolution)

    # Send to display
```

```
    inky.set_image(image, saturation=SATURATION)
    inky.show()

# Main loop
while True:
    for img_path in pictures:
        show_image(img_path)
        time.sleep(180)
```



▲ The masking tape may not be attractive, but it achieves the aim of holding everything in place

In the main loop, the `show_image` function is called with the `img_path` input parameter to select an image from the `pictures` list, resize it if needed (using PIL's `resize` method), and then show it on the display. It then waits 180 seconds – with `time.sleep(180)` – to show the next one (as the `img_path` parameter has been incremented in the `for` loop).

You can call your images whatever you like, so long as the names match in the `pictures` list, and adjust the time delay. The screen will flicker for a few seconds as it's refreshed with an image, as usual with e-paper displays.

We saved our script as **inkyimages.py** in the same folder as the code examples: **~/Pimoroni/inky/examples/spectra6**. We saved our photos in the images subdirectory there.

You'll need to run the script in a virtual environment (we used the `pimoroni` one created when installing the software):

```
  $ source ~/.virtualenvs/pimoroni/bin/activate
  $ python ~/Pimoroni/inky/examples/spectra6/
inkyimages.py
```

To make the script run automatically upon booting up, open the crontab file:

```
  $ crontab -e
```

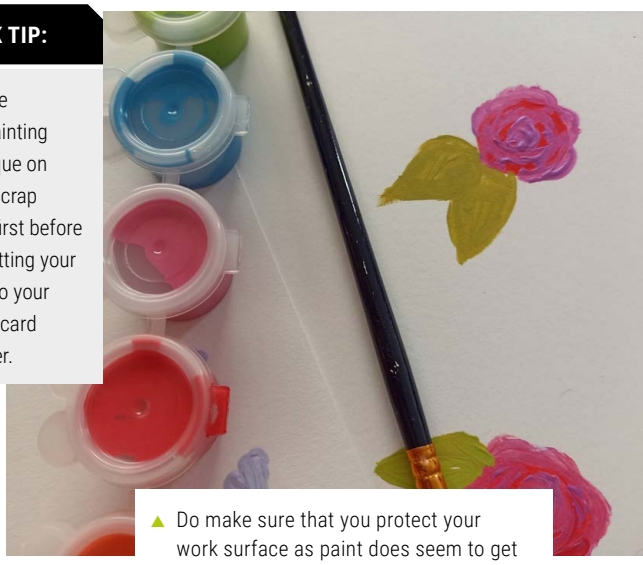
Narrowboat folk art

British canal boat folk art is also known as 'Roses and Castles' art, and is a very colourful and romantic style of painting used to decorate canal (or narrow) boats and the various fixtures and fittings (coal scuttles, jugs, lamps, water cans, fitted items of furniture, kitchen utensils, etc.) aboard them. It's thought that the art-form originated around the 1850s and it was used to decorate the working narrowboats of the time. It really came to the fore at a time when other traditional crafts were on the wane due to the Industrial Revolution. Boatmen would often have had their families with them on the canals, so these boats were their homes, and it became important to them that their tiny living spaces were bright and comfortable, whether through these cheerful paintings, crocheted lace curtains, or highly polished brasses. Individual boat yards and different parts of the canals in Britain also each had their own unique painting techniques and styles. Historians have found clear similarities with folk art from Germany, Holland, and Asia, and these paintings were really a source of pride for the working boatmen and their families.

Solidarity with other boaters also developed, and canal boat art is a colourful form of decoration that is still going strong today amongst the canal boating community. Roses and castles are just two of the 'themes' incorporated in this style of painting, as you can also find other flowers, and many romantic landscape scenes including rivers, cottages, churches, or even animals in some of the paintings.

QUICK TIP:

Practise your painting technique on some scrap paper first before committing your brush to your quality card or paper.



▲ Do make sure that you protect your work surface as paint does seem to get everywhere!

▶ The decorated cardboard mount with a rectangular cut-out, ready for the screen

QUICK TIP:

Remember to use bright colours – that's really important for this style of artwork. Muted colours aren't really going to have the same effect.

Then add this (single) line to the bottom:

```
@reboot ~/.virtualenvs/pimoroni/bin/python
~/Pimoroni/inky/examples/spectra6/inkyimages.py &
```

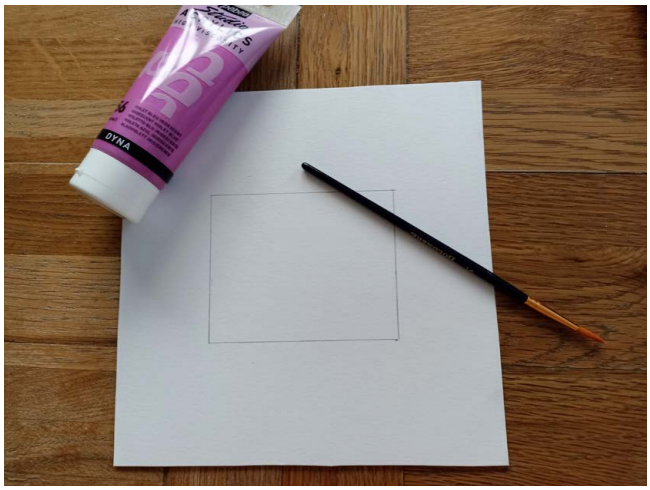
Upon reboot, this activates the virtual environment before running the script. To make it work, however, we found that we needed to copy our photos to a newly created **images** directory in our home folder (as that's where it was looking for them).

We've powered our Raspberry Pi with a standard mains power supply, but you could use a USB power bank, or a LiPo battery pack connected via a LiPo SHIM (**rpimag.co/liposhim**).

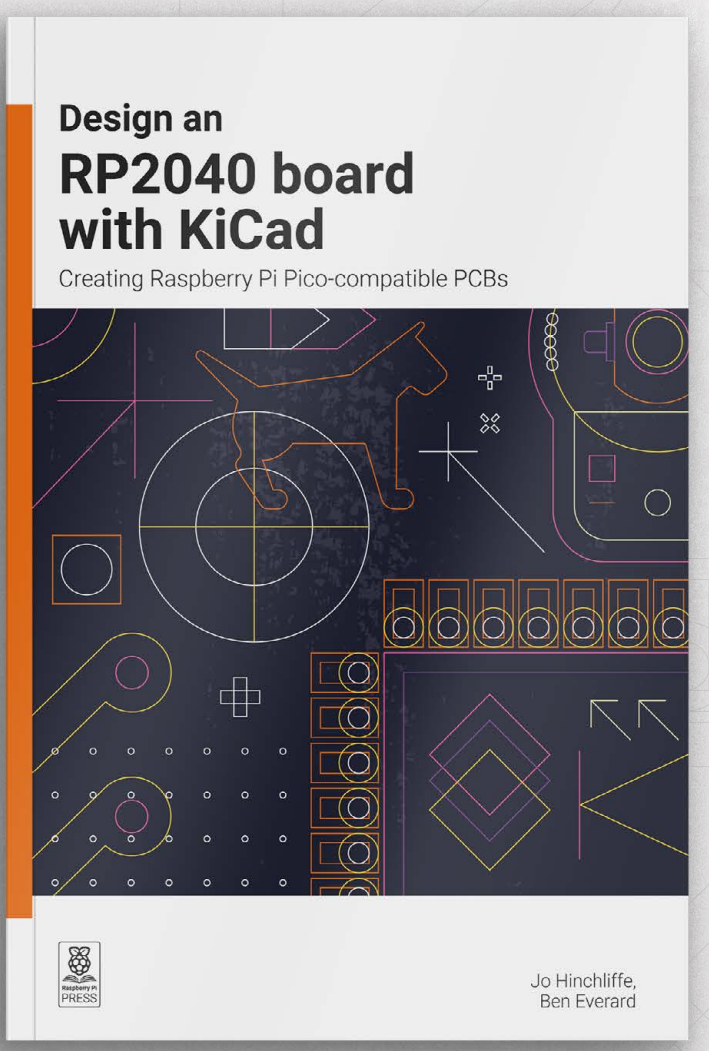
Final thoughts

One of the other images that is famously painted in canal art is the 'castle'. We've not had the space here (or the talent arguably!) to try that, but why not have a go? And don't stop there, as we've seen canal art with animal images such as dogs, people's faces... the key appears to be throw in lots of florals and plenty of colour to whatever you paint.

We've created this project to illustrate how an e-paper screen can be incorporated into a picture frame to display an image that can be changed regularly. One interesting example is the PiArtFrame that shows intricate fractal patterns: **rpimag.co/piframeart**. Another is the Plink digital photo frame with a web interface: **rpimag.co/plink**. You can probably come up with plenty of other ideas for how to use an e-paper display. We'll leave you to use your imagination...



▲ You need to measure the visible area of the screen and draw a rectangle the same size so that you have a perfect fit



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

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

Buy online: rpimag.co/kicad2040

Build a super-simple sensor robot

Affordable robots can still do cool stuff – let's build one from scratch

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

concretedog.blogspot.com

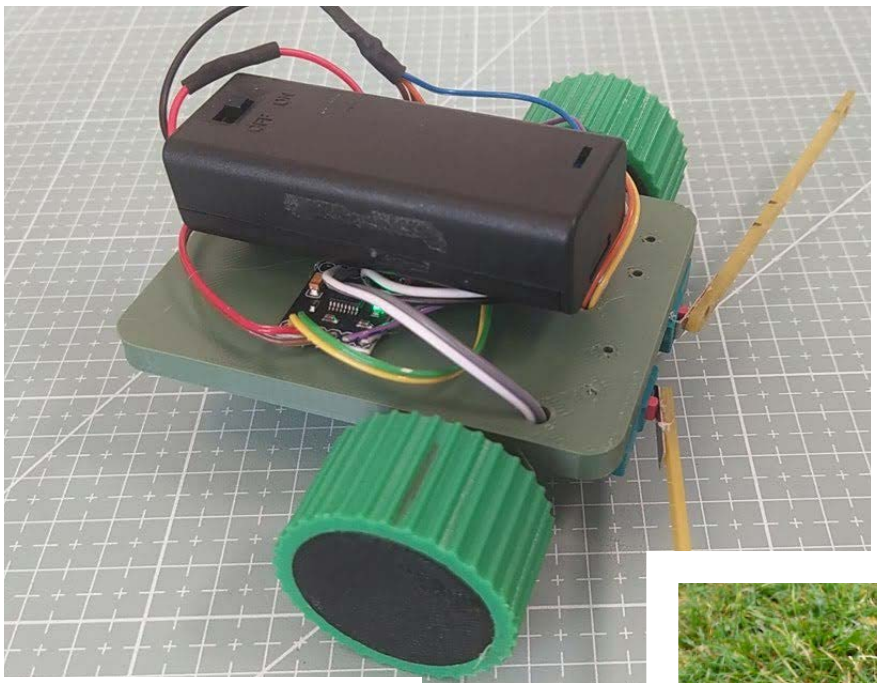
Robots are cool in all shapes and forms, but it's even cooler if you get into building your own. We've featured lots of high-spec robots in these pages over the years, but sometimes the old 'keep it simple' adage is worth revisiting. So we wanted to explore building a simple, minimal component robot with the minimum of 'sensors' to give it some kind of autonomy.

We have some prior experience designing robots. Way back in Hackspace magazine issue 32, we created the Modular Tracked Vehicle (MTV) Robot (**Figure 1**). This used an off-the-shelf tracked chassis system and two pretty powerful brushed geared motors to create a really capable robot platform. In more recent times we published the *Design an RP2040 Board with KiCad* book (**rpimag.co/kicad2040**) which, in one chapter, led to the creation of the open-source StoRPer robot (**Figure 2**).

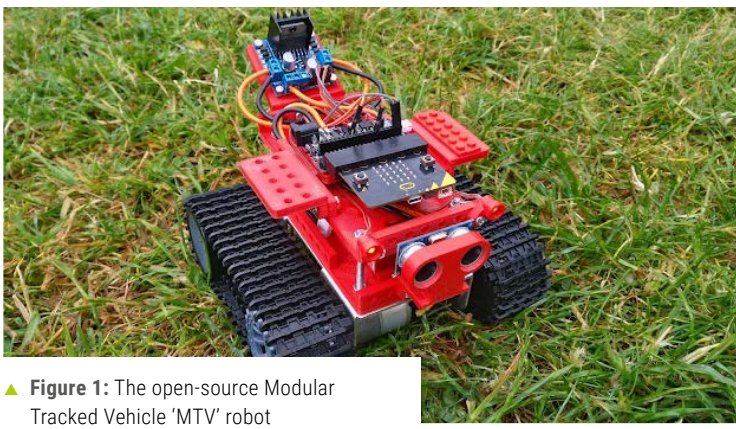
The StoRPer robot is great fun. It was designed with four motor driver systems on board so that you could create an all-wheel-drive system; this is great for Mecanum wheels and Mecanum drive systems. We later added a breakout board (or 'bumper'), which added the ability to easily add ultrasonic sensors and bump switches. Of course, all of this adds complexity, but it serves as some inspiration for a cheaper, simpler build. One area we can simplify from StoRPer is we can build a robot with half the number of motors. Of course, the MTV robot only had two motors, but the tracked system is too complex to make ourselves and requires a higher-rated power supply. The N20 motors have options available that can be powered with the same supply as the Pico, so they are a simpler choice. We can go for a simple configuration of two motors to drive two wheels, with a third passive wheel to help the robot balance. This will make for a fun robot, at reduced cost, complexity, and number of components.

Geared motors

The motors we used on this build, and on the StoRPer robot are called N20 geared motors. They are widely available but you need to take a little care to choose the correct option. They are available in a lot of different gear ratios and in a lot of different voltage ratings. Whilst you can choose the gear ratio yourself, make sure to buy a pair that are the ones rated for up to 6V. This means they will run well whilst testing with a USB cable supplying power to the Pico, but also later when we use a single-cell Li-ion battery.



▲ Our simple two-wheeled robot features two motors and a basic 3D-printed chassis



▲ **Figure 1:** The open-source Modular Tracked Vehicle 'MTV' robot

Look out!

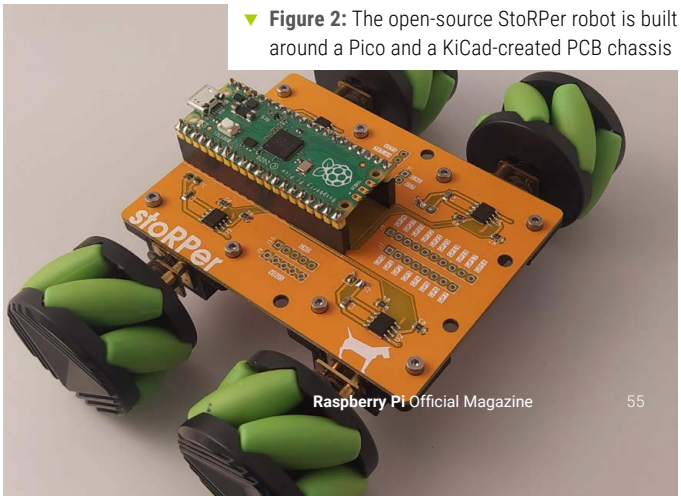
Having played with bump switches with the StoRPer robot, we know that we can make a simple system for collision detection and avoidance which will create a crude form of autonomy in our robot. So with this in mind, we'll make a start.

First of all, we need to think about a chassis. A chassis in its simplest form is simply a base for all the robot components to be attached to. Whilst we have used FreeCAD to create a simple design and then used a 3D printer, it's fair to say that the chassis could be any piece of flat material into which we could drill holes or use hot glue to attach parts. Common materials are plastic boards, or a piece of wood, or even a couple of layers of thick cardboard glued together, got an old CD? ... that could be easily used as a chassis.

It's a great idea to prototype your robot system using a breadboard

It's a great idea to prototype your robot system using a breadboard and indeed, we connected the entire circuit on a breadboard and then tinkered with code on the Pico to help us get to our desired design. However, it's worth noting that with some double-sided sticky foam pads, it's totally possible to attach the motors and bump switches to the underside of a breadboard.

▼ **Figure 2:** The open-source StoRPer robot is built around a Pico and a KiCad-created PCB chassis



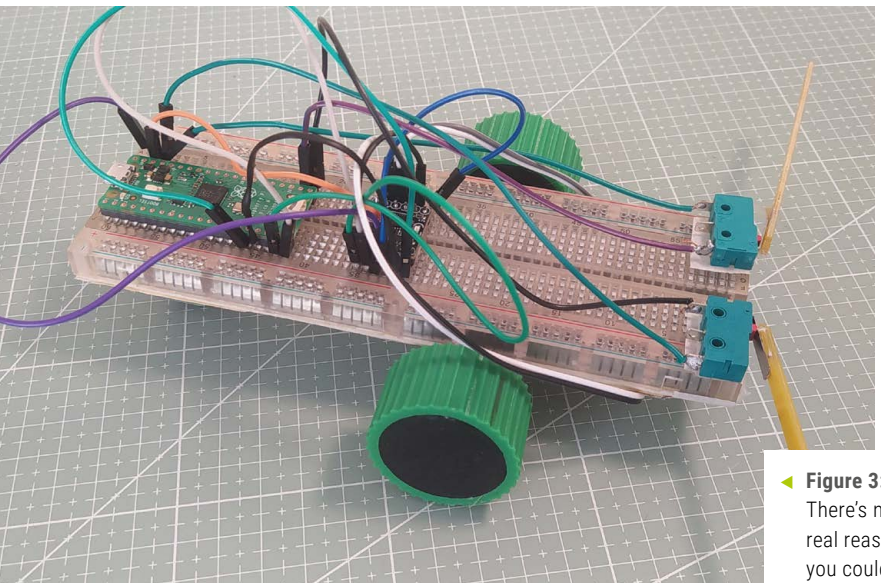


Figure 3: There's no real reason you couldn't build this entire bot on a breadboard!

If you can make or buy a small castor wheel for the back end of the robot, then why not consider making the whole robot on a breadboard? See **Figure 3**.

Using FreeCAD we used the Part Design workbench to layout a basic rectangle which we could 3D-print as a basic chassis that we could iterate on and make decisions about after testing different ideas. As a starter idea, we knew that we'd want to mount some form of rear wheel experiments for the non-driven wheel. A nice thing you can do is create an area where it's easy to glue experiments onto. We achieved this by drawing a 20mm square on our rectangular pad; we then used the pocket tool to cut it into the base by around 1mm. This means that if we make some experiments for the rear unpowered wheel, so long as the base of the idea has a small matching rectangle, we can glue it, exactly centred, into position easily. We added some rectangles where the motors would mount and also added some holes to route wires through (**Figure 4**).

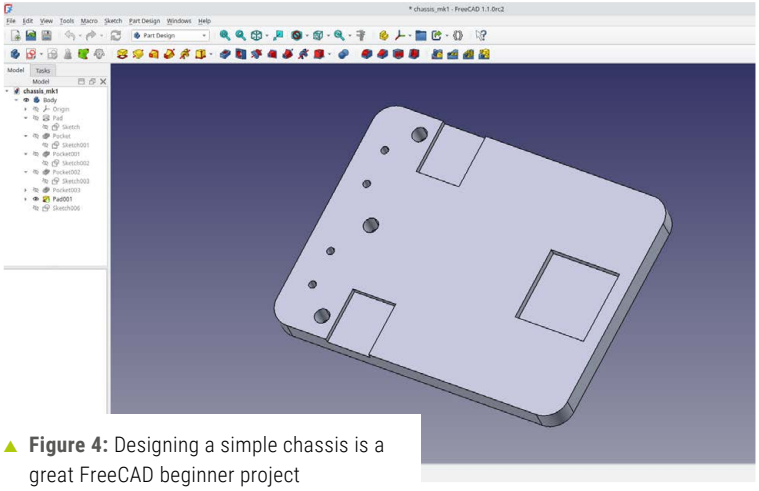
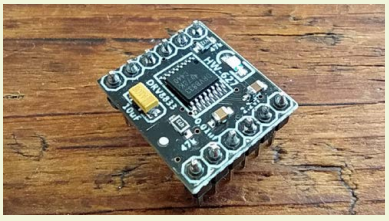


Figure 4: Designing a simple chassis is a great FreeCAD beginner project

QUICK TIP

All the 3D design files and STLs are here: **rpimag.co/sensorbot**.

Motor driver



To drive our motors, we've used a DRV8833 motor controller module. There are numerous similar DRV8833 motor controller modules out there that all operate in a similar fashion. One note is that the DRV8833 has a pin which needs to be tied to a logic high (so a connection to our VSYS, pin 39) that can be called different names on different modules! The pin is often referred to as 'Sleep' or sometimes 'nSLEEP', but on some modules this is labelled on the silkscreen as 'SLP'; on others it's 'EEP'. Whichever you have, the module won't send power to the motor output pins unless this pin is held high.

Mounting momentary microswitches

We also know that we want to add some momentary microswitches to the robot. Almost any microswitches you may already have will do. Most microswitches will have some mount holes. We have a stock of branded Omron SS5GL microswitches as well as some non-branded cheaper clones of this classic switch. The SS5GL has two mount holes that are a little over 2mm in diameter and spaced at 10mm between the centres. This useful dimensional info allows us to quickly add two pairs of holes, slightly under 2mm diameter, so we can self-tap some small M2 bolts into the chassis and hold the switches firmly (**Figure 5**).

Another nice feature of the SS5GL-type microswitches (and others) is that they can be connected in either a 'normally closed' or 'normally open' configuration. Normally closed means that the two connections to the switch are electrically connected until the switch lever arm is pressed and then the disconnect. Vice versa, normally open, means the two connections aren't connected until the switch is pressed. As our Raspberry Pi Pico has internal pull-

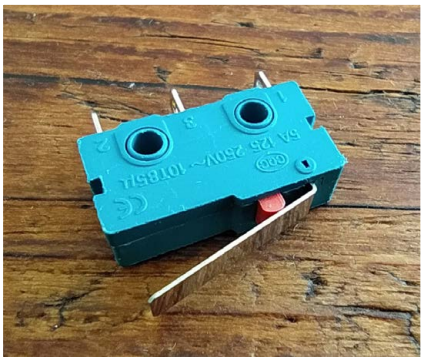


Figure 5: Almost any microswitch can work; we used some affordable SS5GL-style switches

up resistors, it makes sense to use the switch connections for normally closed. This means that we will connect each switch to one GPIO pin on the Pico and ground. We'll define the GPIOs as inputs and we'll set the resistors to pull up. In this configuration, when the switches are not pressed, the GPIO will be connected to ground. When the switches are pressed, and the ground is disconnected, the pull-up resistors ensure the GPIO pin is high. We can then write code to say what happens when the pins are low and what happens if one or more of the pins goes high. Conveniently, using the internal pull-up resistors rather than adding external resistors means we have even fewer components to find, sticking with the theme of this robot build!

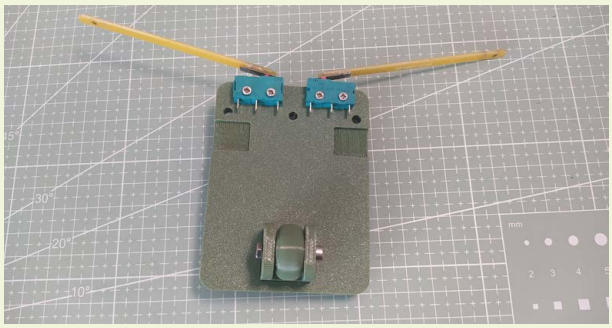
With some ideas and decisions for the chassis and the switches made, it's a good idea to prototype the system and to perhaps iterate on your code. As an example, we began by placing a Pico H onto a breadboard. We connected it up to Thonny and then, as it was a while since we had used this Pico, updated the MicroPython version to the most recent official Raspberry Pi version. We then added our DRV8833 module to the breadboard.

We connected VBUS (pin 40) on the Pico to VCC on our DRV8833 module. We then connected GND (pin 38) on the Pico to the GND pin on the DRV8833. Next, we connected the 'EEP' pin on the DRV8833 (on some modules marked 'SLP' or 'nSLEEP') also to VSYS on the Pico to tie this pin high, enabling the module to supply power to the motors.

We then connected Pico GPIO 14 and GPIO 15 (pins 19 and 20) to the IN1 and IN2 pins on the DRV8833 module and then finally connected one of our 3–6V N20 geared brushed motors to the OUT1 and OUT2 pins on the DRV8833. We are then ready to do our first little code test to check our motor and motor driver are functional using the following code.

Bumper arms

Adding bump switches to a robot it great, but you'll find that, in their standard form, they aren't big enough to detect all objects. The common modification is to attach some kind of lightweight arm to the switch's lever. This in effect makes a system like a cat's whisker! You can experiment with all manner of add-on whiskers for your bump switch mechanism. A popular easy-to-find choice is a length of lollipop (popsicle) stick or a wooden drink stirrer. You could try some small thin stiff wire, or you could even experiment designing and 3D-printing an object to add to your switch. We have a big pile of small offcuts of PCB boards which are the perfect size and length. To attach these to our switch arms, we used a small blob of hot glue, being really careful not to accidentally glue our switch!



```
  from machine import Pin, PWM

  IN1 = PWM(Pin(14))    #set up PWM for GP14
which connects to IN1 on the DRV8833
  IN2 = PWM(Pin(15))   #set up PWM for GP15 which
connects to IN2 on the DRV8833

  IN1.freq(1000)       #set PWM Frequency
  IN2.freq(1000)

  IN2.duty_u16(0)      #setting duty cycle of one
pin to zero fixes the direction of rotation
  IN1.duty_u16(60000)  #setting duty cycle of
this pin sets speed of rotation 0–65535
```

All the code examples in this article are online (**rpimag.co/sensorbot**), so you can download or copy-paste them from there. The above code is **simplest motor_test.py**. Stepping through the above code, you should see that the first line is importing the functions `Pin` and `PWM` from the `machine` library.

The next two lines set up PWM on both GP14 and GP15 and give these the name `IN1` and `IN2`, which are our inputs on the DRV8833 module. Next, we set the frequency for the `IN1` and `IN2` PWMs both of which are set to 1000, which is 1000 times per second or 1kHz. To create a PWM signal, we need to set the duty cycle of the PWM at each of our motor input pins. If we set either (but in the example above, `IN2`) to zero, this then fixes the motor to only rotate in one direction, but we can vary the speed of rotation by adjusting the value of the duty cycle of `IN1`. So by setting the value to `IN1`, you can press the 'Run current script' button in Thonny to check that your motor rotates, then adjust the value of `IN1.duty_u16(45000)` to a value between 0 and 65535 to play with the speed.

There are a couple of things of note at this stage. If you press the 'Stop/Restart back end' button in Thonny, it's likely your motor will keep running! Your only options are to either set `IN1.duty_u16()` to 0 or, the more likely option, pull the USB cable out of your Pico! In the next stage of developing the code, we will add a system to avoid this and we'll also add a reverse function using a small trick in the code.

We aren't going to add the entire code, but rather look at what we will add to the previous motor test example. This code is on the repo, called **basic_n20_motor_test.py**. We import the `time` module for use later in the script, but the main changes begin after setting the `IN1` and `IN2` PWM frequencies. Instead of just setting up the `IN1` and `IN2` duty cycles manually, we will write some small functions to control speed and direction of the motor. The function looks like the code below:

```
  def set_motor(speed):

      if speed > 0:
          # Forward
          IN1.duty_u16(speed)
          IN2.duty_u16(0)

      elif speed < 0:
          # Reverse
          IN1.duty_u16(0)
          IN2.duty_u16(abs(speed))     # absolute
value used

      else:
          # stop
          IN1.duty_u16(0)
          IN2.duty_u16(0)
```

As you should see, we begin by defining a function called `set_motor`, which receives an input called `speed`. We are still going to set the speed of rotation of the motor by setting the `IN1` and `IN2` duty cycle values. In the first part of the function, we set up an `if` conditional statement where if the value of `speed` is set to a number larger than 0, we will create a forward motion by setting the IN1 duty cycle to the value of `speed` and the IN2 duty cycle to zero. In the next `elif` (else if) statement, we set up conditions for reverse. This time, the `elif` is looking for whether the value of `speed` is less than 0. So if the value of `speed` is a negative number, then we will set `IN1` to be zero and `IN2` to be `speed`. However, PWM values are expected to be between 0 and 65535! We are going to allow the speed to be set as either a positive or negative number between 0 and 65535 and then you'll see in the code for reverse that we've used `(abs(speed))`. The `abs` is short for absolute and basically creates an absolute value that is the distance the value is from zero. So if you input -12345 as `speed`, it will actually use the number 12345. A really quick way to see how `abs` works is to use the REPL in Thonny, input `print(abs(-10))`, then press **RETURN** and it should print '10'. Finally, we can finish this function with an `else` statement where we set the duty cycle of both `IN1` and `IN2` to zero, meaning all motors stop any rotation.

We've then added in this script a small `while` loop which allows us to set up a choreographed sequence of motor moves. You should be able to see in the following code snippet how we have set the motor to go forward by setting `speed` in the `set_motor` function to a positive value, then the motor stops with it set to zero, and then reverses when we set it to a negative value. Wrapping the `while True` loop in `try:` and then the `except KeyboardInterrupt:` is a quick way of avoiding the earlier problem

We've added a small while loop which allows us to set up a choreographed sequence of motor moves

where when we stop the script in Thonny, the motors keep turning. When you run this example, you can then press **CTRL+C** on the keyboard and it will interrupt the script and stop the motor.

```
try:

    while True:

        set_motor(30000)  #forward
        time.sleep(2)

        set_motor(0)      # stop motor
        time.sleep(2)

        set_motor(-30000) # backward
        time.sleep(2)

        set_motor(0)      # stop motor
        time.sleep(2)


except KeyboardInterrupt:
    set_motor(0)  # stop motor
```

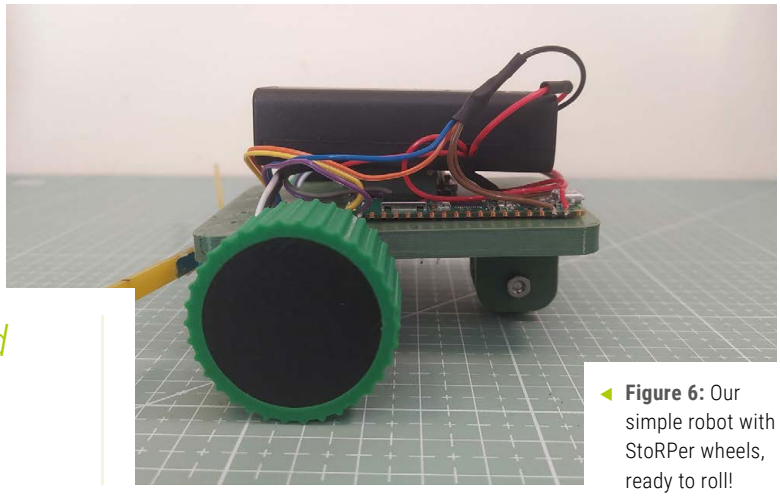
Next, we added one of our microswitches. Again, the code example, called **N20_motor_with_one_button.py**, is on the repo. In this example, you can see we added the line:

```
button = Pin(16, Pin.IN, Pin.PULL_UP)
```

This adds and defines `button` as being attached to GPIO 16 (pin 21), sets it as an input, and makes sure it is using the internal pull-up resistor. The other wire from the switch is connected to the Pico ground/GND (pin 38). This means, as it is a normally closed switch, when the switch is not pressed, the pin is grounded. Whenever the switch is pressed, the pin will be pulled high via the internal pull-up resistor. This means that we can then change our `while` loop to look out for button presses and respond accordingly.

```
FORWARD_SPEED = 30000
REVERSE_SPEED = -30000
try:
    while True:


        set_motor(FORWARD_SPEED)


        if button.value() == 1:


            time.sleep(0.05) #debounce

            if button.value() == 1:

                set_motor(0) #stop
                time.sleep(1)

                set_motor(REVERSE_SPEED)
#reverse
                time.sleep(2)

                set_motor(0) #stop
                time.sleep(1)

                while button.value() == 1:
                    pass

        time.sleep(0.01)

except KeyboardInterrupt:
    set_motor(0)  # Stop motor safely
```

So in the above code snippet, we have set up a couple of variables, `FORWARD_SPEED` and `REVERSE_SPEED`, and given them a value. In the loop we set up the motor to always run forwards when the button is not pressed. We've then set up an `if` statement for what happens when the switch is pressed and returns a high value (1). We've added a little debounce time and check the switch is still high and then it goes into a small choreographed routine with the motor. First, the motor stops for a second, then the motor reverses for a second, then stops again, and then we return out of the `if` statement to the main loop with the motor travelling forward continuously again. This final single motor



◀ **Figure 6:** Our simple robot with StoRPer wheels, ready to roll!

This could be a great and more affordable robot project for a Code Club

Rear castor-wheel

A common approach for a two-wheel drive robot is to deploy some form of castor-wheel system at the back of the robot. This non-driven 'wheel' tends to act more to balance the robot so it isn't dragging the rear of the chassis. A common and affordable off-the-shelf component is the ball-bearing castor. This is a plastic part that can be screwed or glued onto your chassis that contains a free-rotating metal ball bearing. This is a nice solution as the ball can roll in any direction.

As we had access to a 3D printer, we made a couple of designs of rear wheels to fit into the 20mm square area we defined on our chassis earlier. We went for a more standard wheel on a shaft, but made the edges of the wheel very rounded. Whilst this doesn't rotate in the direction of travel like a true castor, the contact point of the rounded wheel on the travelling surface is small, so offers very little friction and allows the robot to turn and rotate freely.

with a button should now give you all the clues as to how we are going to set up a double motor and switch system to form our robot. We won't step through that example, but essentially the idea is that you add a second motor using the IN3 and IN4 on the DRV8833 module and add a second switch to the Pico. You can create a script where the robot drives forwards continuously, but when either switch it pressed, the robot stops, reverses, turns, and then tries to move forward again! If that sounds too complex, we have of course included the script in the examples, called **n20_bump_robot_basic_code.py**.

We went on to build our prototype onto the 3D printed chassis. We used a separate Pico and DRV8833 module for this build as we wanted to directly wire/solder to the devices rather than use header pins. This made it slightly more compact, but again, there's no reason you couldn't mount a Pico with pin headers with the pins pointing upward, etc. You'll notice our chassis has a small fin on the back. This is a little trick for being able to expand a chassis working area: you can glue a small platform on top of the fin to create an upper area. On this we have used a dab of hot glue to mount an AA battery holder (**Figure 6**). Of course, we aren't using an AA battery, but we are using a 14500 Li-ion battery which puts out 4.2V at full charge. We've wired this to our VSYS and GND connections and it acts as a super-neat little power supply for this robot. For wheels, we've 3D-printed some from the StoRPer design, but there are many wheels available that fit the N20 motor shaft. The StoRPer design files are all available here: **rpimag.co/storper**.

This has been a fun project and whilst we've been making it, it's struck us numerous times that this could be a great and more affordable robot project for a Code Club or just for someone looking to only spend a pocket-money level of cash to explore the world of Pico robotics!

Raspberry Pi's GPIO header allows you to connect electronic components and control them with code you've written yourself. Python is the most popular programming language for controlling electronics on a Raspberry Pi, particularly the functions in the GPIO Zero library. With this book, you'll learn how to use GPIO Zero as you build a series of simple electronics projects.

- ***Simple electronics projects including:***
  - *Program some LED lights*
  - *Add a push button to your project*
  - *Build a motion-sensing alarm*
  - *Create your own distance rangefinder*
  - *Make a laser-powered tripwire*
  - *Build a Raspberry Pi robot*

BUY ONLINE: **rpimag.co/gpiozerobook**

Part 01

Bringing LLMs to the edge: Raspberry Pi AI Camera meets large language models

Combine the power of LLMs with the Raspberry Pi AI Camera to connect the physical world to intelligent language-driven systems

Maker
Lucy Hattersley
Lucy is editor of *Raspberry Pi Official Magazine*.

rpimag.co

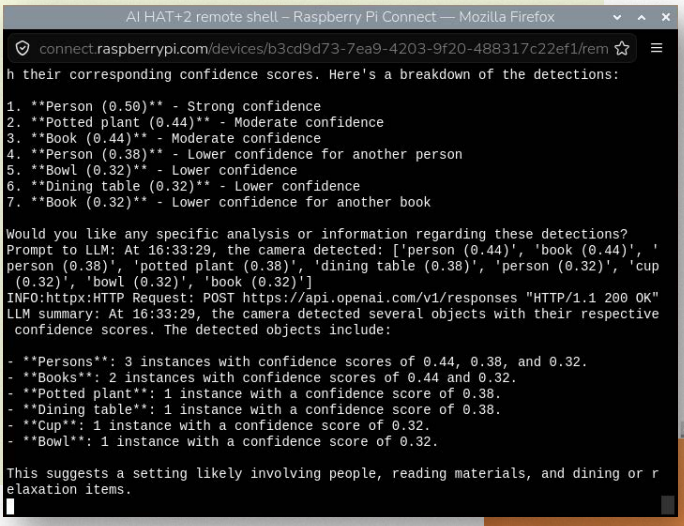
Large language models (LLMs) offer new intuitive ways to interact with technology. From natural conversations with chatbots to summarising long documents, LLMs excel at understanding and generating human-like text.

What happens when we combine the power of LLMs with the Raspberry Pi AI Camera? This pairing opens up new ways to connect the physical world of vision recognition to intelligent language-driven systems.

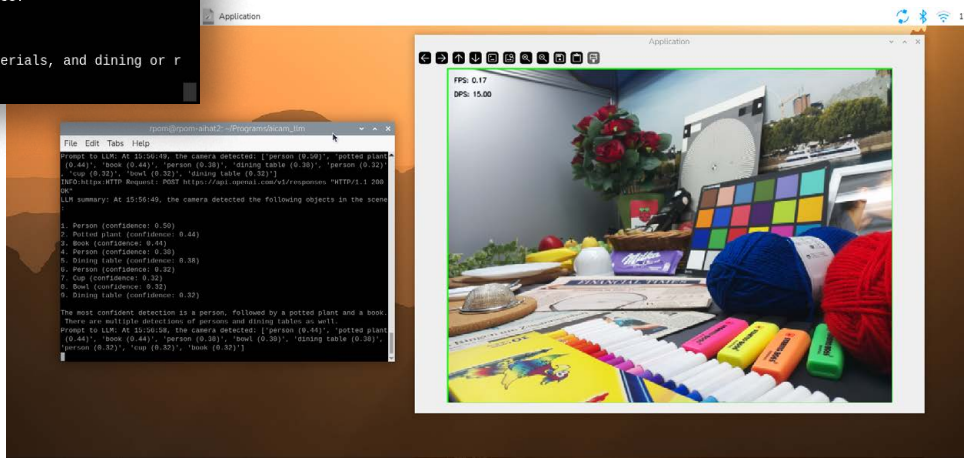
These powerful new systems are being called vision-language models (VLMs). This approach lets you build systems that describe and reason about the physical world using natural language. All without streaming video to the cloud, helping to keep your capture private and reduce the burden of GDPR compliance.

In this tutorial we will consider one way to do this using the Raspberry Pi AI Camera (**rpimag.co/aicam**). Our approach will be where the Raspberry Pi AI Camera constantly sends prompts containing the metadata to the LLM. This approach can be seen in **Figure 1** (overleaf).

Another way to do this is where the LLM is prompted by the user and then the LLM accesses the Raspberry Pi AI Camera to get the latest AI results.



This pairing opens up new ways to connect the physical world of vision recognition to intelligent language-driven systems



- LLM-generated scene summary appears in the terminal, turning raw object detections into a natural-language description of the camera view
- The Raspberry Pi AI Camera detects objects in real time while the LLM interprets the results, combining vision data with language-based insight

This month we will look at the first option. Next month we will take a look at the second approach.

Set up AI Camera

Ensure your Raspberry Pi AI Camera is connected to Raspberry Pi. Before we start, ensure that your Raspberry Pi runs the latest software. Run the following command to update:

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

The AI Camera must download runtime firmware onto the IMX500 sensor during startup. To install these firmware files onto your Raspberry Pi, run the following command:

```
$ sudo apt install imx500-all
```

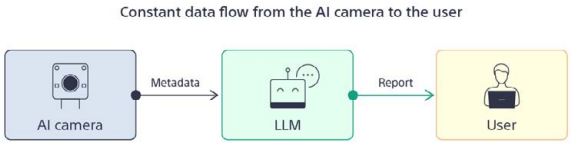
Raspberry Pi's AI Camera does the heavy lifting with the AI model detecting objects, recognising patterns, and generating metadata on the sensor like `{Cat (0.76), Box (0.81)}`.

Instead of streaming raw video to the cloud, the system can output the inference results as metadata, drastically reducing the amount of data transmitted to the cloud or to other systems. This is particularly beneficial in environments with limited bandwidth or expensive data costs. This means the camera provides structured insights as inference results; for example, labels, bounding boxes, and confidence scores. These are then passed to an LLM, which turns structured detection data into human-readable summaries and contextual insights.

The code snippet (**01_aicam_to_llm.py**) at the end of this article can be adapted to your own situations. This sends the metadata from the Raspberry Pi AI Camera to an LLM using OpenAI. To run it, you will need to install modlib and the OpenAI library, then get your own API key for OpenAI.

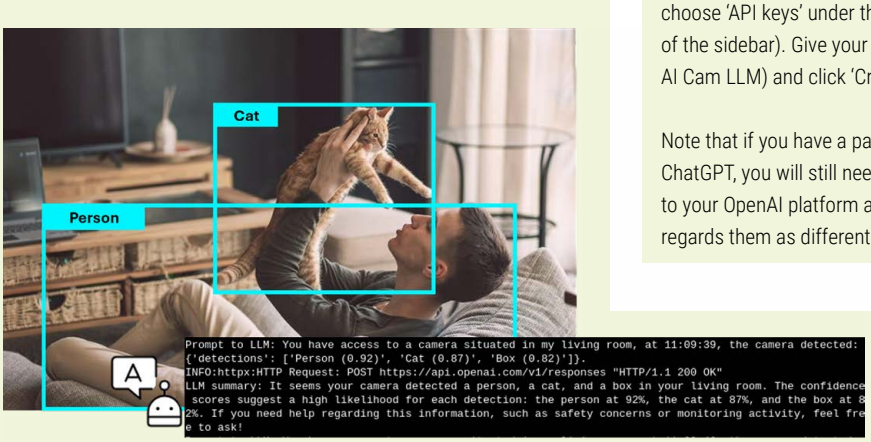
Let's set up the code. First, clone all the files from our GitHub account.

```
$ git clone https://github.com/lucyhattersley/
aicam_llm.git
```



▲ **Figure 1:** Constant data flow from the AI camera to the user

▶ The smart home observer in action, showing person and cat detections with LLM summary



Get an OpenAI API key

To get an API key, go to **platform.openai.com** and sign in (or sign up if you don't have an account). You'll need to add credits to your account, with a minimum of $5. Click 'Create API keys' on the Dashboard window (or choose 'API keys' under the Manage section of the sidebar). Give your key a name (ours is AI Cam LLM) and click 'Create secret key'.

Note that if you have a paid account with ChatGPT, you will still need to add credits to your OpenAI platform account. OpenAI regards them as different things.

Take a look inside with `ls` and you will see example code for all our projects. Many code files contain the same code with different prompts. We expect you to finally use one of the original code files with your own prompt.

We will need to create a virtual environment so we can add the OpenAI and Application Module Library (modlib) packages.

```
$ python -m venv env
```

And activate our virtual environment:

```
$ source env/bin/activate
```

Use pip to install modlib and openai:

```
$ pip install modlib openai
```

Now edit the file and add your API key. We are going to use the Thonny IDE to do this:

```
$ thonny 01_aicam_to_llm.py
```

Add your API key to line 8, replacing `<OPENAI_API_KEY>` with the key inside straight quotes so it looks like:

```
client = OpenAI(api_key="abcde012345")
```

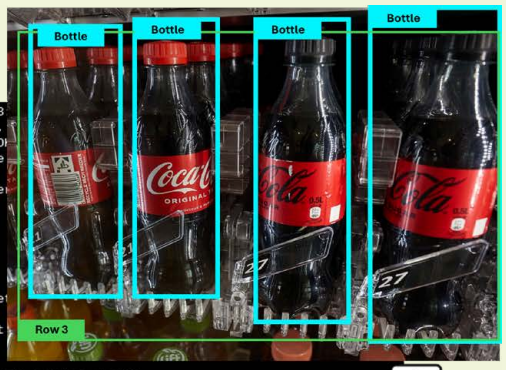
Save the file and exit Thonny.

Now run the file with:

```
$ python 01_aicam_to_llm.py
```

The first time you do this, it will perform a Network Firmware Upload. Wait for the file to upload (around 30 seconds). After this, the terminal will display a text description of what is in the viewfinder:

```
  LLM summary: At 16:33:29,
  The camera detected several objects with their
respective confidence scores.
  The detected objects include:
  **Persons**: 3 instances with confidence scores
of 0.44, 0.38, and 0.32.
```



Retail shelf monitor detecting bottles in row three

```
Prompt to LLM: You have access to a camera monitoring a vending machine, at 11:18
), in Row 3', 'Bottle (0.77), in Row 3', 'Bottle (0.84), in Row 3, Bottle (0.69),
INFO:httpx:HTTP Request: POST https://api.openai.com/v1/responses "HTTP/1.1 200 O
LLM summary: Based on the camera detection at 11:18:45, here's an analysis of the

1. **Item Detection**: The system detected four bottles in Row 3, each with differ
   - Bottle (0.82)
   - Bottle (0.77)
   - Bottle (0.84)
   - Bottle (0.69)

2. **Assumptions**:
   - The differing confidence scores suggest varying degrees of clarity in the de
.
   - Since multiple bottles are detected in a single row, it can be inferred that

3. **Stock Level**:
   - It is likely that Row 3 has a reasonable stock level because there are multiple bottles visible.
   - If Row 3 typically holds more than four bottles, it may suggest a moderate stock level.
   - Conversely, if Row 3 has limited capacity (e.g., only holds five or six items), then this could indicate that it's approaching
w stock level.

4. **Conclusion**: Row 3 appears to have a sufficient stock of bottles at this observation time, but further analysis over time is neede
d to monitor for any depletion or restocking needs.
```

Preview window

If running Raspberry Pi OS with the GUI, you can view a preview window while running the program. Uncomment line 16 to add the `frame.display()` command and activate the preview.

Factory floor watcher detecting compliant and non-compliant workers

```
Prompt to LLM: At 17:23:25, the camera detected: ['person (0.76) matched highvis', 'person (0.83) matched highvis'
, 'person (0.51) not matched highvis']
INFO:httpx:HTTP Request: POST https://api.openai.com/v1/responses "HTTP/1.1 200 OK"
LLM summary: At 17:23:25, the camera detected the following:

- Two individuals matched with high visibility clothing (confidence scores: 0.76 and 0.83).
- One individual not matched with high visibility clothing (confidence score: 0.51).

This information could assist in monitoring compliance with safety regulations or any specific conditions for visi
bility in that area. Would you like any specific analysis or action taken based on this data?
```

```
  **Books**: 2 instances with confidence scores
of 0.44 and 0.32.
  **Potted plant**: 1 instance with a confidence
score of 0.38.
  **Dining table**: 1 instance with a confidence
score of 0.38.
  **Cup**: 1 instance with a confidence score of
0.32.
  **Bowl**: 1 instance with a confidence score
of 0.32.
  This suggests a setting likely involving
people, reading materials, and dining or
relaxation items.
```

We can adjust this program to identify different things by adjusting the prompt on line 23 of our code. The subsequent programs adjust this prompt to perform different tasks.

- 01a_smart_home.py
- 01b_retail_shelf.py
- 01c_factory_floor.py

Inspect these programs with Thonny or an IDE of your choice and look at the prompt on line 23.

Smart Home Observer

On the Raspberry Pi AI Camera, we run an object detection model to detect objects of interest like people and pets, producing results with data containing the class and confidence like:

```
  {"detections": ["Person (0.92)", "Cat (0.87)",
"Box (0.82)"]}
```

Then the Raspberry Pi AI Camera sends this information to the LLM, which processes the results. The prompt on line 23 is:

```
  prompt = f"You have access to a smart camera
in the living room of my home. At {time.
strftime('%H:%M:%S')}, the camera detected:
{labels}"
```

When run, the code produces a friendly update:

```
  At 14:23, one person is in the living room with
the cat. A box is in the room as well.
```

Retail Shelf Monitor

With a Raspberry Pi AI Camera monitoring a shelf, vending machine, or a fridge, we can use an object detection model to detect the items we wish to monitor. Then we can add functionality to check what shelf or row the items are on. We send the LLM the detections with a prompt:

```
  prompt = f"You have access to a smart
camera in a vending machine. At {time.
strftime('%H:%M:%S')}, the camera detected:
{labels} Provide information on the stock levels
of the vending machine."
```

And the LLM generates a report:

```
  "Four soda bottles are left in row three –
stock may need replenishing soon."
```

Factory Floor Watcher

Raspberry Pi AI Camera checks if workers are wearing safety gear. In this situation, we can add some more application logic to match people with high-vis jackets to make sure they are wearing one. The prompt on line 23 of our code is:

```
  prompt = f"You have access to a smart camera in
a warehouse. At {time.strftime('%H:%M:%S')}, the
camera detected: {labels} Provide information if
people are wearing highvis jackets."
```

Then the metadata is forwarded to an LLM, which produces a natural alert:

```
  Warning: one worker is not wearing a high-vis.
```

As we can see, the prompt on line 23 of our code can be adjusted to a wide variety of tasks using natural language.

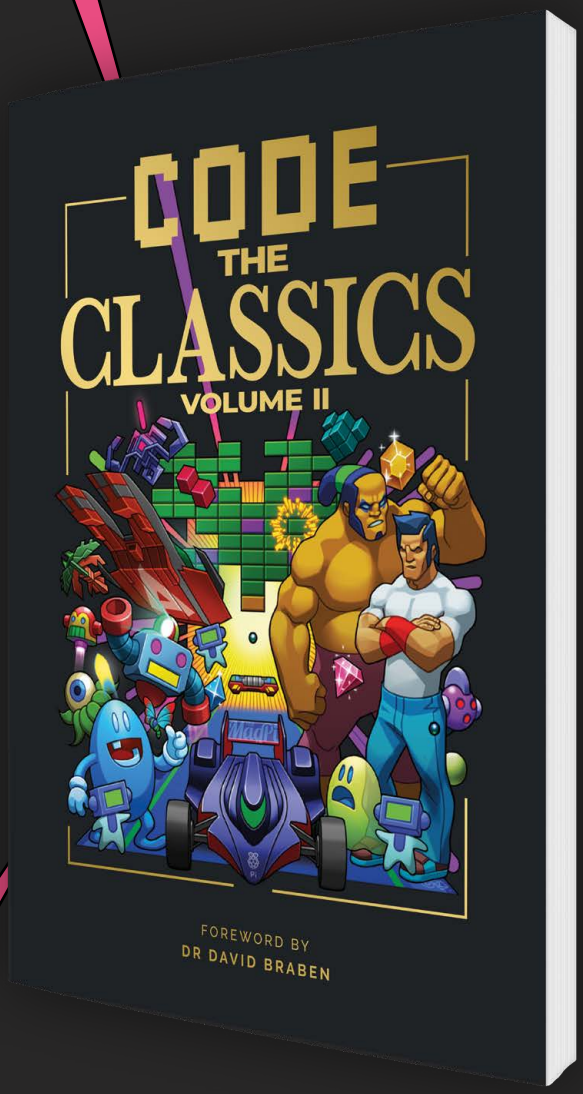
DOWNLOAD THE FULL CODE:

rpimag.co/aicamllm

01_aicam_to_llm.py

> Language: **Python**

```
001. import os
002. import time
003. from openai import OpenAI
004. from modlib.devices import AiCamera
     from modlib.models.zoo import
005. SSDMobileNetV2FPNLite320x320
006.
007. # --- Setup ---
008. client = OpenAI(api_key=<OPENAI_API_KEY>)
009. CONFIDENCE_THRESHOLD = 0.3
010. device = AiCamera()
011. model = SSDMobileNetV2FPNLite320x320()
012. device.deploy(model)
013.
014. with device as stream:
015.     for frame in stream:
016.         # frame.display() # Uncomment to
     display preview window
017.
018.         # Filter confident detections
019.         detections = frame.detections[
     frame.detections.confidence >
     CONFIDENCE_THRESHOLD]
020.         labels = [f"{model.labels[c]}
     ({s:.2f})" for _, s, c, _ in detections]
021.
022.         # Build a natural prompt
023.         prompt = f"At {time.strftime(
     '%H:%M:%S')}, the camera detected: {labels}"
024.         print("Prompt to LLM:", prompt)
025.
026.         # Ask the LLM for a summary
027.         response = client.responses.create(
028.             model="gpt-4o-mini",
029.             input=prompt
030.         )
031.         print("LLM summary:",
     response.output[0].content[0].text)
032.
033.         time.sleep(5)
```



- ***Get game design tips and tricks from the masters***
- *Download and play game examples inspired by classics*
- ***Learn how to code your own games with Pygame Zero***
- *Explore the code listings and find out how they work*

Code the Classics Volume II not only tells the stories of some of the seminal videogames of the 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples co-programmed by Andrew Gillett and Raspberry Pi founder Eben Upton.

Available now: **rpimag.co/store**

Build a wordsearch clock

Show the time in a unique and very customisable way

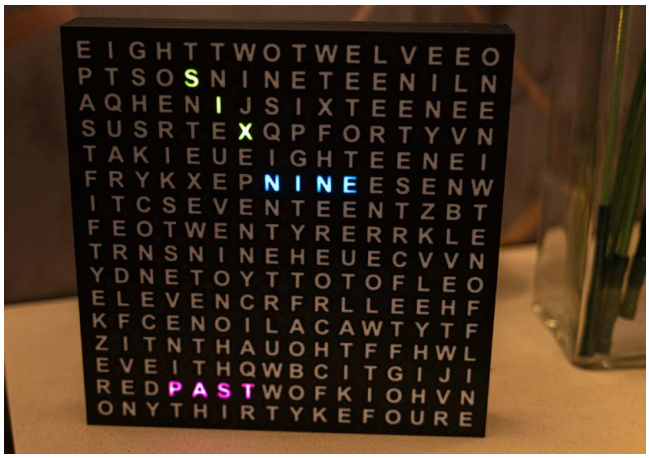
Maker

Rob Miles

Rob has been playing with software and hardware since almost before there was software and hardware.

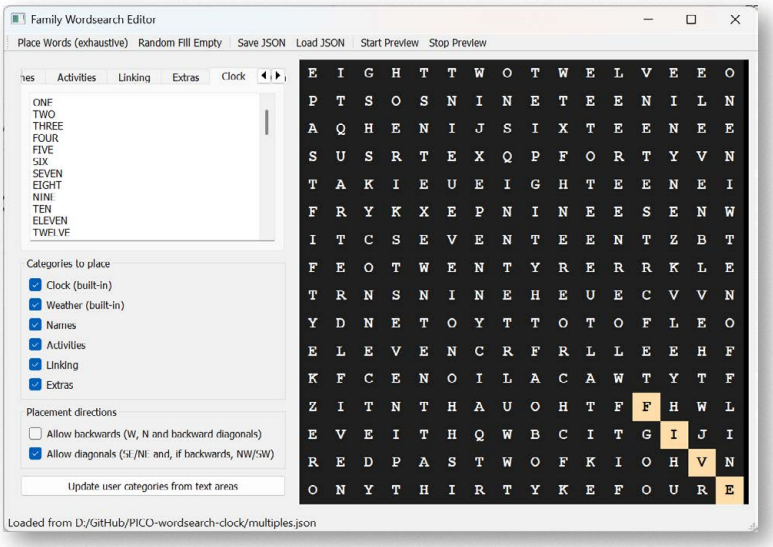
robmiles.com

Create a stylish and ever-changing time display using a Raspberry Pi Pico, a 16 × 16 NeoPixel panel and multicoloured 3D printing. You can extend this to make an alarm clock which plays suitably sarcastic messages to encourage you to get out of bed. You can find all the resources to create and customise your wordsearch clock in the GitHub repository here: **rpimag.co/wordsearchclock**.



▶ **Figure 1:** The time is drawn in sequence, so you know the order of the values

Figure 1 shows a clock created as a wordsearch which shows the time. It uses a 16 × 16 array of NeoPixels with a light behind each letter. A Raspberry Pi Pico in the clock running a MicroPython program lights pixels behind letter shapes in sequence to spell out the time in words. The colours for each word are chosen randomly, producing a very pleasing display. Or at least that is what the author thinks. He has always enjoyed a good wordsearch, so he thought he might have a go at creating his own and putting it in a device. So, he started by writing a Python program to make some wordsearches.



▶ **Figure 2:** You can add your own words to the wordsearch

Searching for words

The Wordsearch Editor shown in **Figure 2** builds wordsearches from lists of words. The words are arranged into categories so that the clock can display other information. The time category contains multiple copies of some of the words to make the time display more interesting. The program builds a wordsearch by placing each word in turn, picking a random location for the first letter and checking that the rest of the word fits. It will also try to fit words inside words if it can. The program will clear everything and start again if it can't place a word. The current arrangement took 15 attempts to find one that worked. It turns out that there is nothing difficult about making complex wordsearches if you are happy to leave your program running for a while. When all the words have been placed, the program generates a JavaScript Object Notation (JSON) file that describes the word arrangement:

```
"grid": {
    "rows": 16,
    "cols": 16,
    "letters": [
      "EIGHTTWOTWELVEEO",
      "PTSOSNINETEENILN",
      "AQHENIJSIXTEENEE",
      "SUSRTEXQPFORTYVN",
      "TAKIEUEIGHTEENEI",
      "FRYKXEPNINEESENW",
      "ITCSEVENTEENTZBT",
      "FEOTWENTYRERRKLE",
      "TRNSNINEHEUECVVN",
      "YDNETOYTTOTOFLEO",
      "ELEVENCRFRLLEEHF",
      "KFCENOILACAWTYTF",
      "ZITNTHAUOHTFFHWL",
      "EVEITHQWBCITGIJI",
      "REDPASTWOFKIOHVN",
      "ONYTHIRTYKEFOURE"
    ]
```

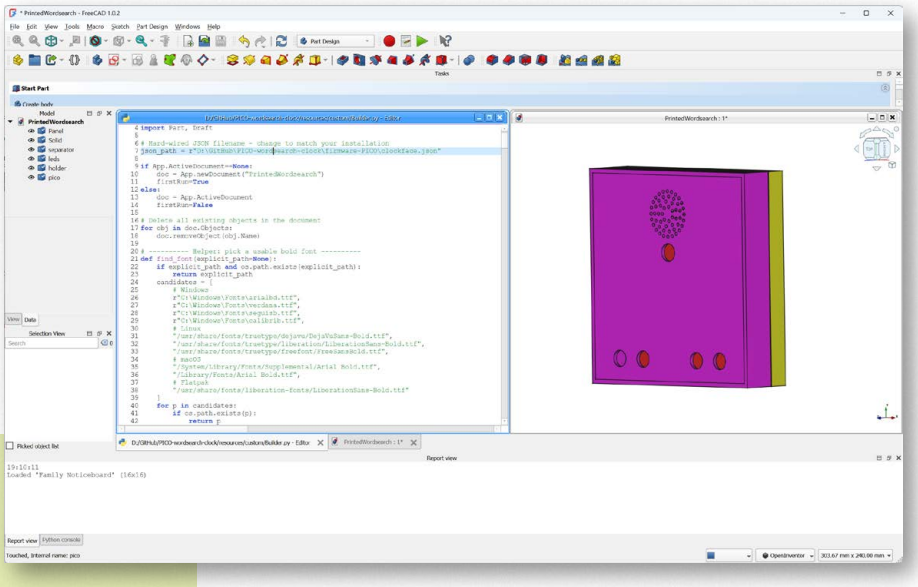
The program builds a wordsearch by placing each word in turn, picking a random location for the first letter and checking that the rest of the word fits

YOU'LL NEED

- Raspberry Pi Pico W or 2 W
- RGB panel WS2812B 16 × 16 pixels flexible module
- DuPont connecting wires, 10cm male-to-male and male-to-female
- Micro USB / USB-C cable for Pico W / 2 W
- M2 Philips pan head screws, 5mm long, to hold the Pico and the speaker in the case
- Two colours of PLA filament (black and white work very well together)

If you want to make the alarm clock version, you will also need:

- DFPlayer – search for 'DFPlayer Mini mp3'
- MicroSD card – between 4GB and 16GB
- Four non-latching push-buttons which fit through a 13mm diameter hole
- 40mm speaker with mounting holes spaced at 32.5mm



▶ **Figure 3:** The program uses fonts on the local machine for the character designs

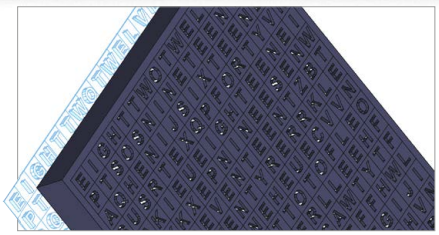
This code above shows the start of the wordsearch description which gives the size of the grid and the letter arrangement. This is followed by an array of word descriptions:

```
    "words": [
        "word": "ONE",
        "category": "clock",
        "direction": "SE",
        "cells": [
          {
            "row": 1,
            "col": 3
          },
          {
            "row": 2,
            "col": 4
          },
          {
            "row": 3,
            "col": 5
          }
        ],
```

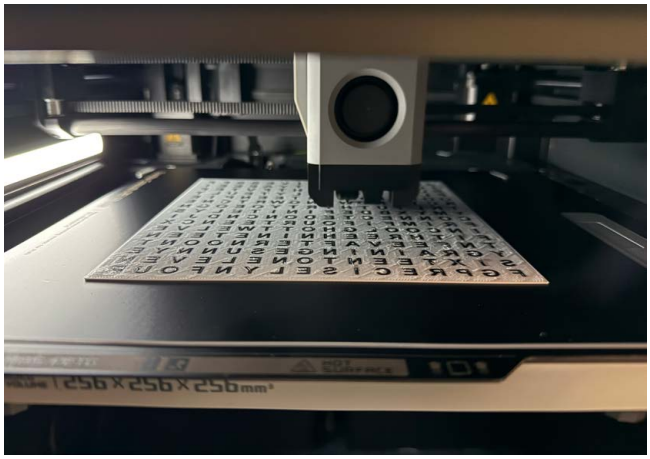
The code above describes the position of the word ‘ONE’ in the wordsearch. It gives the location of each of the letters in the word. The MicroPython program that displays the clock uses this to create an index giving the positions of the words it needs to display the time. The same JSON file is also used by another Python program that runs inside the FreeCAD design tool to create the clock case.

Making a case

Figure 3 shows the **Builder.py** program running inside FreeCAD. The program creates the front and back of the clock along with a separator that fits between the LED panel and the front of the clock. It also generates objects representing the LED panels and the Pico inside the clock. This helps a lot with the design process. The two halves of the case contain grooves and ridges so that they snap together. The alarm clock version of the case contains fittings for the speaker and DFPlayer device as well as the Pico. The front of the case is printed using two different filament colours. The builder program creates two output files which are merged when the object is sliced for 3D printing.



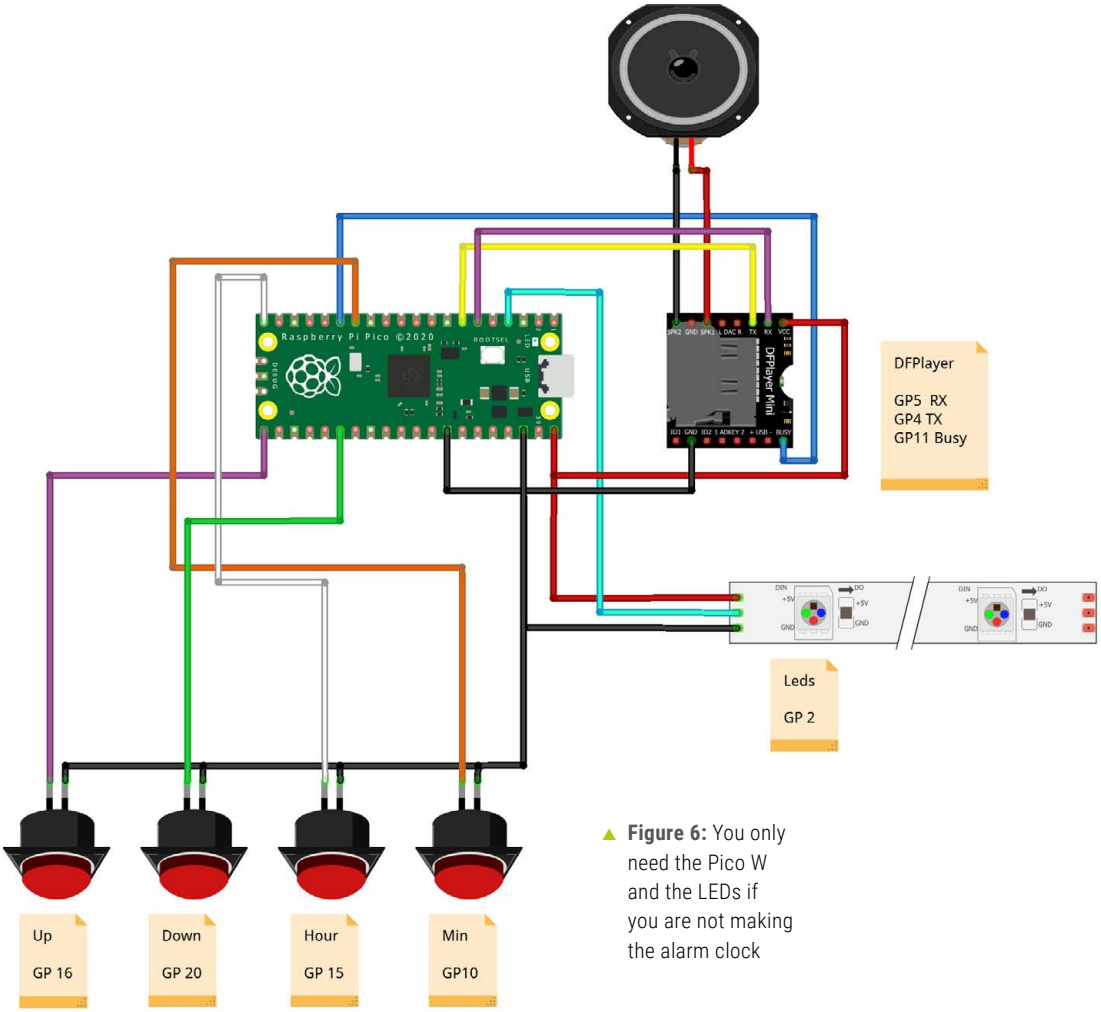
◀ **Figure 4:** The letters have been moved back so that they are visible in the picture



▲ **Figure 5:** The first few layers must be printed very slowly so that the individual letter elements have time to stick to the print bed

Figure 4 shows the two components that are 3D-printed together to make the clock face shown in **Figure 1**. The black outer case contains holes which are filled by the white letters. Light shines through the text to display the letter for that square. The case is printed face-down on a smooth print bed.

Figure 5 shows a clock face being printed. This clock will have black letters on a white background. You can also make a clock with white letters on a black background.



▲ **Figure 6:** You only need the Pico W and the LEDs if you are not making the alarm clock

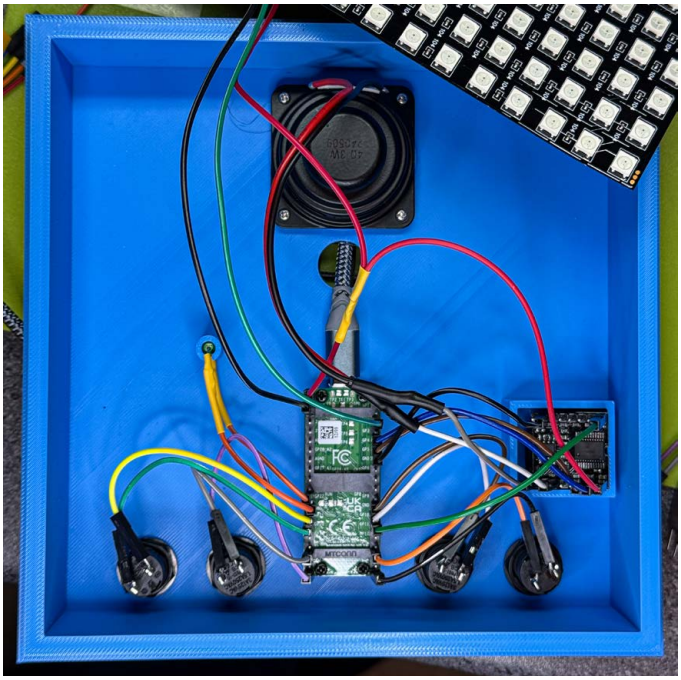
Wiring up

Figure 6 shows the circuit of the clock. The four buttons are used to set the alarm time. The DFPlayer is fitted with a microSD card that contains the alarm sounds encoded as MP3 files. The DFPlayer is controlled via a serial connection to the Pico. There is also a connection between the DFPlayer and Pico which is used to detect when a sound sample has finished playing. This is read by GP11 on the Pico. The LEDs are connected to a single pin and powered from the USB input to the Pico. A standard USB 5V adapter can be used to power the clock.

Figure 7 shows the internals of the completed clock. The connections are made using DuPont male-to-male cables. The pins at the ends of the cables are soldered directly into the Pico and onto the switch connections. The DFPlayer is supplied with pins which are connected using male-to-female DuPont cables. The power supply output from the VBUS pin on the Pico is split between the pixels and the DFPlayer.

Managing managers

The clock is controlled by MicroPython code that is structured as a set of managers. Each manager is a software object that has setup, update, and teardown behaviours. Starting the clock involves reading a settings file to determine which managers are to be loaded, creating the manager objects, and initialising them.



▲ **Figure 7:** This version of the clock also contains a tilt sensor which is not presently used by the software

QUICK TIP

The settings info for the Wi-Fi manager in **settings.json** includes the network SSID and password values.

When the clock is running, each manager is updated in turn. There are managers for the Wi-Fi connection, the network clock, the wordsearch application, the pixels, the MP3 player, and the input buttons. A manager can generate events and provide services to other managers; for example, the button manager will tell the wordsearch application when the user presses a button. The managers are configured in the **settings.json** (JavaScript Object Notation) file stored with the code.

```
    "App_wordsearch_clock": {
        "key_repeat_interval_ms": 500,
        "start_audio_track_no": 3,
        "end_audio_track_no": 20,
        "run_on_power_up": true,
        "enabled": true,
        "wordsearch_file": "clockface.json",
        "wordsearch_letter_delay_ms": 250,
        "alarm_timeout_ms": 30000,
        "key_repeat_delay_ms": 1000,
        "wordsearch_word_delay_ms": 1000,
        "dependencies": [
            "clock",
            "gpio"
        ],
        "alarm_sample_interval_ms": 1000,
        "wordsearch_display_gap_ms": 5000
    }
```

The JSON above configures the wordsearch clock manager. It provides all the settings information for the clock, including which managers the wordsearch is dependent on. There are similar entries for all the managers that are to be used. The device contains a range of managers for different devices. Any manager with a name starting with 'App_' is an application.

```
App_wordsearch_clock_manager.py
base_manager.py
clock_manager.py
dfplayer_manager.py
display_manager.py
gpio_manager.py
hullos_manager.py
mqtt_manager.py
pixel_manager.py
rotary_encoder_manager.py
stepper_manager.py
uart_manager.py
wifi_manager.py
```

The code above shows some of the currently available managers. They make it very easy to use different devices together. You just create a **settings.json** file that brings the managers together and configures them.

QUICK TIP

The settings for the alarm time are stored in a small extra file so that the entire settings file does not have to be rewritten when the alarm time is changed.

Eventful events

Once the managers start running, they can connect to events and services that they provide for each other:

```
evt = self.clb.get_event("clock.minute")
if evt:
    evt.subscribe(self.on_minute_tick)
```

The code above runs when the wordsearch manager starts up. It connects the method `on_minute_tick` in the wordsearch manager to the `minute` event provided by the clock. When the clock is active, it will call this method every minute.

```
def on_minute_tick(self, event, data):
    if self.first_run:
        if self.settings["run_on_power_up"]:
            self.first_run=False
            self.start_show_time()
```

The code above shows the method that runs each minute. It checks a setting to see if the clock should display the time when it is powered up. If `run_on_power_up` is true, the code checks to see if this is the first time that `on_minute_tick` has been called. If it is, the `start_show_time` method is called to start displaying the time. This makes sure that the clock will start showing the time when it has valid clock data.

At your service

A manager can use services provided by another. The `get_service_handle` method returns a reference to an object which exposes all the services provided by the specified manager.

```
  self.dfplayer =
self.get_service_handle("dfplayer")
```

This statement runs inside the wordsearch clock manager when it starts up. The clock needs to use the MP3 player to generate alarm sounds, and this is how it gets access to the player. The variable `self.dfplayer` is set to refer to an object which has all the behaviours of the `dfplayer` manager. Code in the wordsearch clock manager can now call methods on `self.dfplayer` manager (the one that plays audio samples for the alarm). It can then call methods on the player to tell it what to do:

```
  def play_alarm_sample(self):
      self.dfplayer.play(random.randint(
self.start_audio_track_no,
self.end_audio_track_no))
```

The `play_alarm_sample` method above plays a random alarm sound by calling the `play` method on the `dfplayer`. Each manager also exposes all its available methods as commands you can enter via the serial console when the clock is running.

```
  [dfplayer]
    dfplayer.play            - Play track n
    dfplayer.stop            - Stop playback
    dfplayer.volume          - Set volume (0-30)
```

If you want to play a track, you can enter a command at the serial console to do this:

```
  dfplayer.play 3
```

The above command would start track 3 playing. Being able to send commands to managers as the system is running is very useful when creating and debugging a device.

Manager magic

You might be starting to think that using all these managers, service objects, and events makes creating applications rather complicated. Well, it turns out that the author has discovered that not using these techniques makes things even harder. Spreading behaviours across multiple managers and providing well-defined ways that they can communicate makes it much easier to create complex devices.

Data-driven for the win

This is an example of a 'data-driven' application. The starting points for the design and the program behaviours are JSON files of structured data. There are no 'hard-wired' values in the program code: everything is loaded from data when the programs run. The **clockface.json** file is used by the wordsearch clock program and in FreeCAD to make the clock body; the **settings.json** file is used to specify the behaviours that are required and how they should be configured. If we want to make a new application, we just need to create a new manager and bring together the required behaviours for it.

The present version of the clock just displays the time. The next version of the clock will add Message Queue Telemetry Transport (MQTT) support and additional words so that the clock can display other information as well.

QUICK TIP

You can play different ranges of alarm sounds by changing the start and end track numbers.

Reading, reading, and reading

You can make two versions of the clock case, as shown below. On the left-hand clock, the text is displayed by lighting the square around a black letter shape. On the right, the text is displayed by shining a light through a white letter shape. It turns out that when you shine light through letters, you need extra diffusion behind the letter (otherwise light shines through gaps around the letter), so you can't change from one version to another by just swapping the black and white filaments. You tell the **Builder.py** program the version that you want.

▲ You could use other colour combinations, but make sure that one of them is fairly transparent

Part 09

EDVAC, UNIVAC, & Princeton IAS

The computer that never was, and its offspring

Maker

Tim Danton

When not writing books about classic computers, Tim is editor-in-chief of the British technology magazine PC Pro. He has also helped to launch several technology websites, most recently TechFinitive.com, where he is a senior editor.

dantonmedia.com

While wars are never to be celebrated, they have an undeniable uniting effect. There is nothing quite like an existential threat to your country, to your way of life, when it comes to motivation. This can be seen through the story of the Colossus, where University of Cambridge professors formed an unlikely partnership with the General Post Office, but also in the story of ENIAC. Here, the Moore School of Electronics, part of the University of Pennsylvania, marched in step with the US Army to create the world's first large-scale digital computer.

Then war stops. Organisations return to their natural self-serving state, people look to enrich themselves, personal enmities that were set aside for the good of the country resurface.

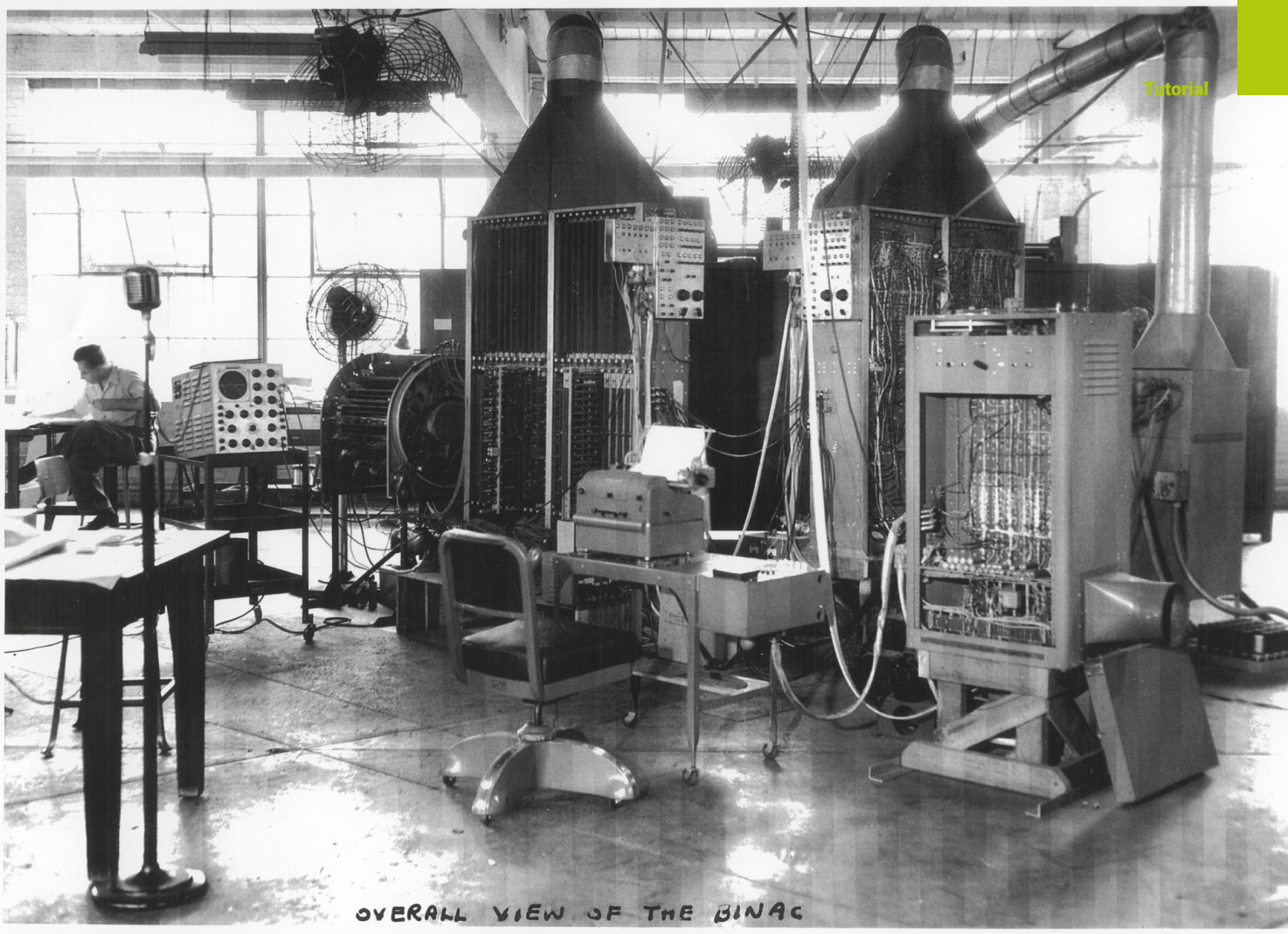
This night-and-day difference between peacetime and wartime is highlighted by the different routes taken by the ENIAC and the EDVAC. Both can consider the Moore School as their spiritual birthplace, but while the ENIAC's progress was marked by how well academic institutions and the army worked together – stimulated, admittedly, by large amounts of money – the EDVAC's path went wrong almost from the outset.

Its story is one of ego, of capitalism, of something approaching chaos. Of an idea born of war that could only survive peace by dividing into three rival projects.

A chance platform meeting

> *My egotism got the better of me*

But John von Neumann knew none of this on the fateful day he bumped into Herman Goldstine on a railroad platform in the summer of 1944. As discussed in part 6 on ENIAC (**rpimag.co/162**), this wasn't a complete coincidence. Both men had good reason to be visiting the Army's Ballistic Research Laboratory (BRL) at its Aberdeen Proving Ground: von Neumann in his role as a consulting scientist for the army; Goldstine as the key liaison between the US Army,



▲ A complete BINAC system; Albert Auerbach, who ran the first test routine on it, is seated at the rear
Image: courtesy of the Computer History Museum, CC BY-NC-SA

which had commissioned the ENIAC, and the Moore School that was building it.

When their paths crossed, von Neumann had been scouring the country for high-speed computational devices for months. The work being done at Los Alamos, groundbreaking research on atomic and hydrogen bombs known collectively as the Manhattan Project, required colossal amounts of calculations. Calculations that weren't feasible on existing equipment, so his task was to hunt out quicker machines. He had already visited Harvard to see the Harvard Mark I in action, but found it wasn't powerful enough and, besides, already booked up by Navy demands.

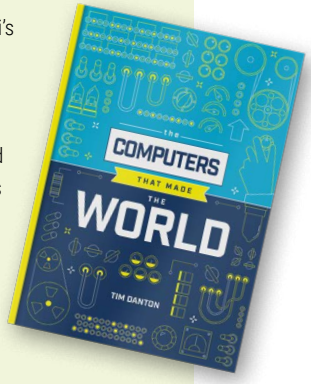
His visit to Bell Labs, which had created its own electromagnetic relay-based computers (see Complex Number Calculator, part 3: **rpimag.co/159**), was far more positive. "I spent the better part of a day with [George] Stibitz," von Neumann wrote in April 1944,[1] "who explained to me in detail the principles and the working of his relay counting mechanisms, and showed me the interpolator as well as the almost-finished anti-aircraft fire-control calculator … Dr Stibitz even suggested, what went

1 Letter from von Neumann to Warren Weaver, 10 April 1944, quoted in William Aspray, *John von Neumann and the Origins of Modern Computing* (MIT Press, ISBN 978-0262518857), p32

The Computers that Made the World

This article is an extract from Raspberry Pi's book, *The Computers that Made the World*. This book tells the story of the birth of the technological world we now live in. It chronicles how computers reshaped World War II. And it does it all through the origins of twelve influential computers built between 1939 and 1950. You can pick up a copy on the Raspberry Pi website.

rpimag.co/tctmtw



far beyond my expectations, that he may seek permission for an experimental computation of the kind I suggested on the big machine in the process of breaking it in".

War's unlikely partnerships

Four months later, he was still keen. In a letter to Robert Oppenheimer, director of the Los Alamos Laboratory, he wrote that Stibitz's big machine "would be well worth having if a future of, say 1½ years or more is being envisaged for the project".[2] By this time he had also met Leland Cunningham, who oversaw the machine-computing section at BRL in Aberdeen. "In Cunningham's opinion the simplicity in planning, the reliability of the elements, the self-checking features, and the ability to run overnight without a crew, should alone make the [Stibitz] machine five times or more faster than any IBM aggregate, quite apart from the other advantages."

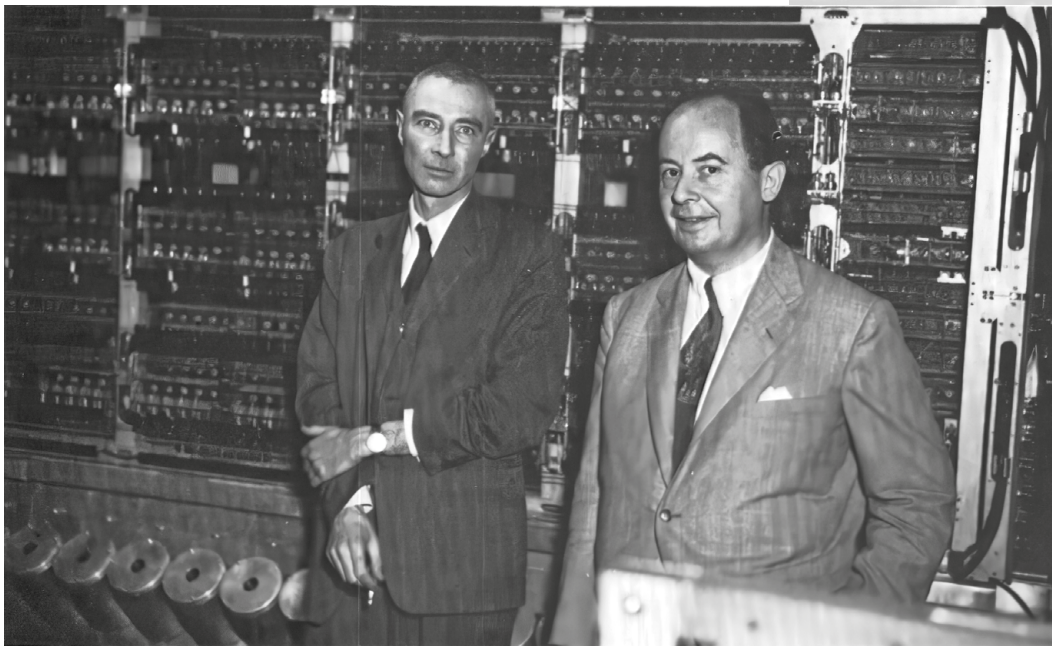
The fact that nothing came of this stems from von Neumann's chance meeting with Goldstine. One that happened – historians believe[3] – soon after von Neumann sent the above letter to Oppenheimer. It was a story Goldstine told many times, which we cover in part 6 on ENIAC, but it bears repeating. "Along came von Neumann with a very abstracted look on his face and he and I are the only two people on the railroad platform," he recalled in 1967.[4]

Goldstine says that he considered leaving the great mathematician alone, but "my egotism got the better of me". Von Neumann, who in all accounts is described as approachable, even funny, was happy to chat. But the mood of the conversation shifted rapidly from "polite chit-chat" to a grilling when Goldstine explained his involvement with the ENIAC. "And he immediately got exceedingly interested. I didn't at that time realise why but he spent the rest of the time of that day and several days subsequent to that in quizzing me about this machine."

2 Letter from von Neumann to Robert Oppenheimer, 1 August 1944, quoted in William Aspray, *John von Neumann and the Origins of Modern Computing*, p33

3 It's a cause of some frustration to historians that there is no written record of the date, with Goldstine's recollections tending to use woolly phrases such as "sometime in the summer" in his retellings.

4 Association for Computing Machinery Meeting, 30 August 1967, Archives Center, National Museum of American History, tape 1



▶ Robert Oppenheimer and John von Neumann in front of the Princeton IAS computer, 1952
Image: Public Domain

Von Neumann arrives

At some point in early August – again, Goldstine is hazy on dates – von Neumann visited the Moore School for the first time. "I recall with amusement Eckert's reaction to the impending visit," wrote Goldstine.[5] "He said that he could tell whether von Neumann was really a genius by his first question. If this was about the logical structure of the machine, he would believe in von Neumann, otherwise not. Of course, this *was* von Neumann's first query."

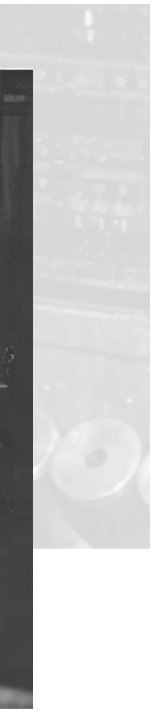
This marked the beginning of von Neumann's many visits to the Moore School, and ultimately the ENIAC would serve the purpose he sought: in December 1945, into early 1946, it was used to calculate thermonuclear equations crucial to the development of the hydrogen bomb. In terms of the EDVAC's story, what matters is that while von Neumann arrived too late to have any notable impact on the ENIAC's design, with building well under way, he would go on to have many discussions with the Moore School team, including Goldstine, on the design of its successor.[6]

Building a dream team

Much of this time is now shrouded in mystery, in part due to a lack of detailed written records but also because the various actors

5 Herman H Goldstine, *The Computer from Pascal to von Neumann* (Princeton University Press, 1972, paperback edition 1993, ISBN 978-0691023670), p182

6 Like most of these early computers, the ENIAC was frequently upgraded during its long life, and von Neumann assisted in this process in the late 1940s. So while he didn't have an impact on the first version of the ENIAC, he is considered to have had an active part in its later development.



– most notably Presper Eckert, John Mauchly, Goldstine, and von Neumann – came to have entrenched positions depending on where they stood on the patents that emerged from the EDVAC's creation. So here we will focus on facts cemented by contemporary documentation; any retrospective quotes from the people involved should be sprinkled with generous helpings of salt.

We know from a three-page memo[7] that in late January 1944, Eckert had considered a "Magnetic Calculating Machine" that replaced many of the costly valves used in the ENIAC with a magnetic "disc or drum" for storage. He also specified that it should use binary, but there was little space in this document for detail on how calculations might be carried out. It is an overview, nothing more.

We also know that on 11 August, Goldstine wrote to Colonel Leslie Simon suggesting that the US Army should grant a second contract to the Moore School "to permit that institution to continue research and development with the object of building ultimately a new ENIAC of improved design".[8] In particular, the new computer would have more storage, fewer valves, and greater capacity in terms of the number of problems it could solve. It would also be truly programmable: reprogramming the ENIAC involved rewiring it using plugboards. A process that could take weeks, depending on the program's complexity.

> *The lightning was out of the bottle*

At some point in August, Goldstine recollects, Eckert "came up with the idea that a mercury delay line could be used for storage of information".[9] This was one of his greatest contributions to early computers, with many of the computers in this book relying on this fussy but ingenious technology. The idea came to him because he had previously worked with mercury delay lines as part of a radar project.[10]

And August 1944 wasn't done yet. Spurred on by Goldstine's earlier memo and perhaps with the added impetus of von Neumann's interest, on the 29th of that month the Firing Table Reviewing Board convened to consider their next move. Attendees included Cunningham and Goldstine, plus five others from the BRL, along with two mathematical consultants and a certain John von Neumann. We don't know what role von Neumann played in the decision, but he must have been pleased with the result: the board recommended entering a contract with the Moore School for a new electronic computer.

One of the few things everyone agrees on is that von Neumann regularly visited the Moore School over the following months to discuss the EDVAC – as it quickly became known, standing for Electronic Discrete Variable Automatic Computer – with Eckert, Mauchly, Goldstine, Arthur Burks, and other engineers involved with the project. And on 31 March 1945, everyone seemed to be getting along just fine, if the project's first summary report is to be believed.

Aside from "problems of logical control", the report states,[11] they had informally discussed "the use of EDVAC, storage capacity, computing speed, sorting speed, the coding of problems, and circuit design". To a lesser extent, they covered input and output systems. But here's the most fateful sentence: "Dr von Neumann plans to submit within the next few weeks a summary of these analyses of the logical control of the EDVAC together with examples showing how certain problems can be set up."

Which he did. In April, he mailed his handwritten notes to Goldstine, who then organised for them to be typed up, copied, bound, and distributed to a select group of 31 people.[12] These included engineers working on the project, the army committee that had commissioned the new computer, and a handful of trusted experts such as Douglas Hartree. It was called a 'First Draft Report on the EDVAC' because that's what it was: there were omissions and mistakes. In places, it's confusing. At the same time, for the first time ever, it set out a logical structure for a computer that we would now recognise.

Another notable point about the report was that it only had John von Neumann's name on it. This was problematic. While the logical interpretation is purely von Neumann's, Mauchly and Eckert claimed that it was built off the back of their ideas, which they had shared freely within the group discussions described above. The report made it look like the work was von Neumann's alone.

This wasn't merely a blow to Eckert and Mauchly's egos, and their place in history, but also threatened any future applications

7 Presper Eckert, 'Disclosure of Magnetic Calculating Machine', 29 January 1944, **rpimag.co/eckertdisclosure**

8 Memo from Goldstine to Simon, 'Further Research and Development on ENIAC', 11 August 1944, reprinted in Herman H Goldstine, *The Computer from Pascal to von Neumann*, p185

9 As above, p186

10 William Aspray, *John von Neumann and the Origins of Modern Computing*, p36

11 As before, p38. The report's authors are listed as Eckert, Mauchly, and S Reid Warren.

12 Thomas Haigh, Mark Priestley, and Crispin Rope provide a detailed timeline in their comprehensive tome, *ENIAC in Action* (The MIT Press, 2016, ISBN 978-0262033985), pp137-139

for patents. They attempted to redress the potential damage – or, in their view, set the record straight – by publishing their own report entitled 'Automatic High Speed Computing: A Progress Report on the EDVAC' in late September.[13]

The most pertinent aspects of this report come at the start, as part of the 'Historical Comments' section. After discussing the creation of the ENIAC they describe the need for a new machine, the EDVAC. "It was clear that this new machine would, with much less equipment, easily handle problems beyond the intended scope of the ENIAC. Therefore, by July, 1944 [so before von Neumann met them] it was agreed that when work on the ENIAC permitted, the development and construction of such a machine should be undertaken."

And what of von Neumann's role? "He has contributed to many discussions on the logical circuits of the EDVAC, has proposed certain instruction codes, and has tested these proposed systems by writing out the coded instructions for specific problems. Dr von Neumann has also written a preliminary report in which most of the results of earlier discussions are summarised. In his report, the physical structures and devices proposed by Eckert and Mauchly are replaced by idealised elements to avoid raising engineering problems which might distract attention from the logical considerations under discussion."

In other words, the ideas were ours, not his. Step away from our patents.

Goldstine was very much in the von Neumann camp. In his view, before the eminent mathematician arrived, "the group at the Moore School concentrated primarily on the *technological* problems, which were very great; after his arrival he took over leadership on the *logical* problems". (The italics are Goldstine's.) To a large extent, this view is reflected in the differing emphases of the two reports, but we know that von Neumann was interested in the practical side of computer development too. For example, in his letters to Goldstine during 1945 he mentions cathodes and voltages; he is well aware of the electrical engineering side of creating a computer.

One anecdote, shared by Arthur Burks,[14] highlights the fight between clean logic and real-world electronics. "I remembered well a discussion of serial adders that took place at one of our meetings [at the Moore School] of March 1945. Pres [Eckert] and John [Mauchly] had designed several serial adders, the simplest of which took ten tubes. Not knowing of these results, von Neumann announced cheerily that he could build an adder with five tubes. We all looked amazed, and Pres said, 'No, it takes at least ten tubes.' Johnny [von Neumann] said, 'I'll prove it to you,' rushed to the board, and drew his adder.

"'No,' we said, 'your first tube can't drive its load in 1 μsec, so an inverter is needed, then another tube to restore the polarity.' And so the argument went. Johnny was finally convinced. But he was not taken aback. 'You are right,' he said. 'It takes ten tubes to add – five tubes for logic, and five tubes for electronics!'"

This story also serves as a reminder to the quiet role played by Arthur Burks in the EDVAC's creation. He had a Philosophy PhD and background in logic, but had earned his engineering spurs the hard way through his work on the ENIAC. His part in the computers' development is often underplayed, but Irven Travis, who played a pivotal role in the Moore School EDVAC's story, was unstinting in his praise: "In terms of logical design and mathematical intuition, if you will, Arthur Burks was one of the most brilliant men we've had in that field – ever," he told historian Nancy Stern.[15] He was also often part of the meetings where the EDVAC's structure took shape over late 1944 and early 1945.

The lightning escapes

Von Neumann summarised the difficulty of ascribing credit during a fraught meeting to discuss patents in 1947: "There are certain items which are clearly one man's," he is quoted as saying in the minutes, "[such as] the application of the acoustic tank to this problem was an idea we heard from Pres Eckert. There are other ideas where the situation was confused. So confused that the man who originated the idea had himself talked out of it and changed his mind two or three times. Many times the man who had the idea first may not be the proponent of it. In these cases it would be practically impossible to settle its apostle."[16]

To an extent, the argument was academic: the lightning was out of the bottle. On his return to England, Douglas Hartree shared his copy of the report with Maurice Wilkes, who read it avidly as we describe in the story of the EDSAC (part 8, **rpimag.co/164**). A copy also found its way to Alan Turing, who referred to the report when setting out his plan for the ACE.

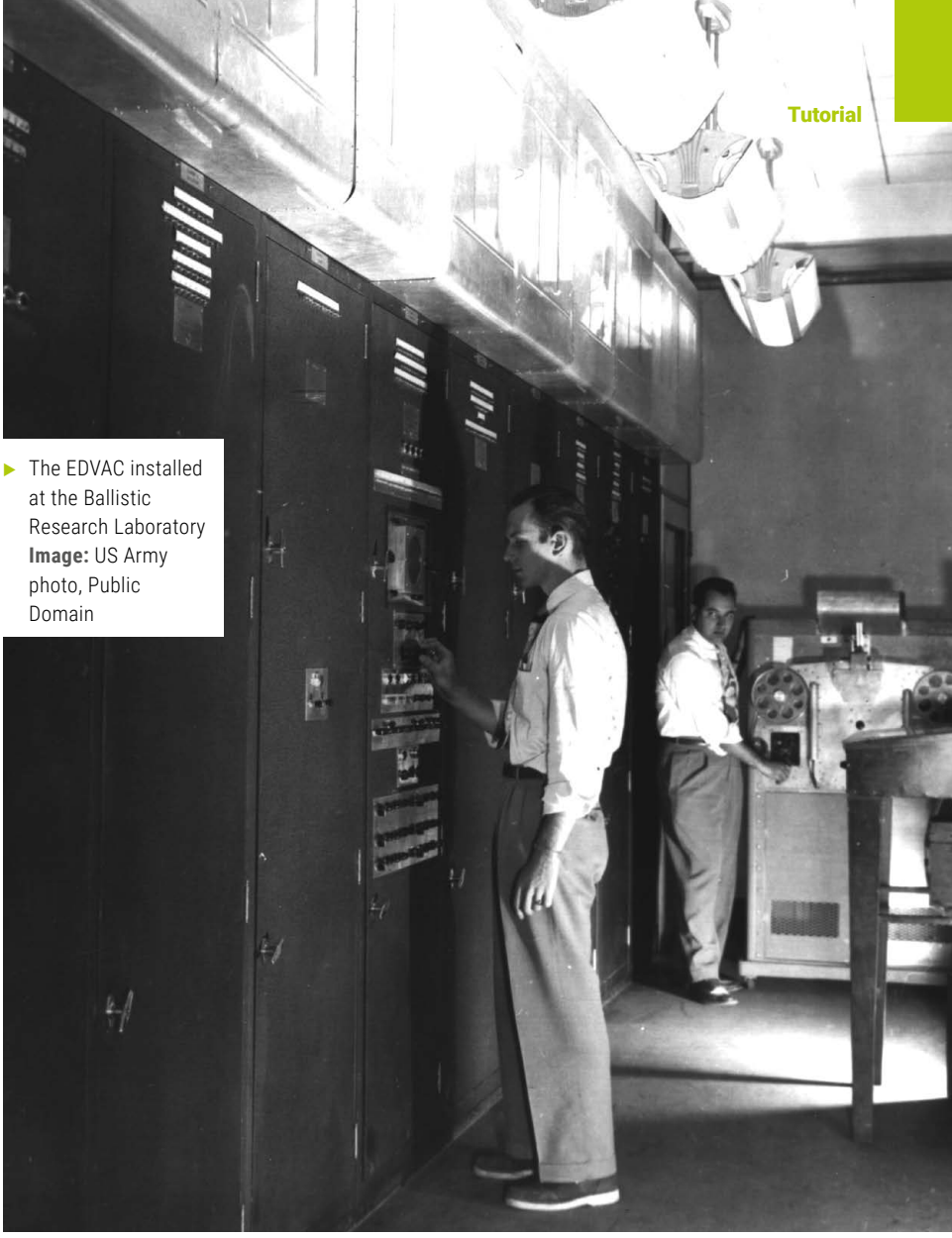
Perhaps the biggest irony is that the EDVAC, as laid out logically by von Neumann and physically by Eckert and Mauchly,

13 John Mauchly and Presper Eckert, 'Automatic High Speed Computing: A Progress Report on the EDVAC', typed manuscript, 30 September 1945. An incomplete copy can be viewed at **rpimag.co/edvacreport**

14 Arthur Burks, 'From ENIAC to the Stored-Program Computer', reprinted in N Metropolis, J Howlett, and Gian-Carlo Rota (eds.), *A History of Computing in the Twentieth Century* (Academic Press, 1980, ISBN 978-0124916500), p341

15 'Oral history interview with Irven Travis' by Nancy Stern, 21 October 1977, Charles Babbage Institute, University of Minnesota, **rpimag.co/irventravisinterview**, p16

16 Remarks from Minutes of Conference held at the Moore School of Electrical Engineering on 8 April 1947 to discuss patent matters, reprinted in Herman H Goldstine, *The Computer from Pascal to von Neumann*, p195



The EDVAC installed at the Ballistic Research Laboratory **Image:** US Army photo, Public Domain

never came into being. From this point, it would split into three rival projects. The first, for the US Army, would be built at the Moore School as per the contract. While the eventual computer would bear the name EDVAC, it was substantially different from these mid-1945 plans.[17] The second line of development would follow Eckert and Mauchly, who as we shall soon see left the Moore School to set up their own company. And the third would be built by von Neumann and friends, the Princeton IAS computer.

Three-way split

The split came gradually, grumbling tremors that acted as precursors to an earthquake. It's surely no coincidence that these first tremors came at the point where the war in Europe had ended and the USA felt that victory in the Pacific was inevitable: everyone was looking at what was to come next. For von Neumann, a return to his mathematical research; for Eckert and Mauchly, perhaps their chance to make a fortune?

By now, von Neumann was hooked on high-speed electronic computers: he wanted to build one dedicated to research, much like his friend and fellow mathematician Max Newman on the other side of the Atlantic (as covered in the story of the Pilot ACE, in a future issue). As early as February 1945, von Neumann discussed the make-up of a team who could build such a computer with Goldstine and Cunningham – his wish list of the time included Eckert, Mauchly, and Stibitz.[18]

In March, Norbert Wiener from MIT approached him with a tempting proposition: join as chair of mathematics and he would have a ready-made lab at his beck and call. This was in stark contrast to Princeton's IAS (Institute for Advanced Study) which, at that time, purely focused on theoretical research. MIT was in many ways the logical choice for a new computer, home as it was to Vannevar Bush's differential analyser and a haven for applied mathematicians. Von Neumann visited in August 1945 and was given a formal job offer the following month. But he didn't take it.

Peacetime ambitions diverge

That's partly because the IAS didn't want to lose such an influential figure from their staff, but also because von Neumann felt loyalty to the institute: it had offered him a position long before his reputation was made. Not that von Neumann was so loyal that he kept quiet about his openness to a move when visiting rival institutions. Following such an occasion, the University of Chicago invited von Neumann to create an Institute of Applied Mathematics, while he received what historian William Aspray describes as "job feelers" from Columbia and Harvard.[19]

If this was von Neumann employing the time-honoured tactic of using other job offers as leverage, it worked. Despite objections

17 This is set out brilliantly in a paper by Michael Godfrey and DF Hendry called 'The Computer as von Neumann Planned It'. It was published in the IEEE Annals of the History of Computing, Vol 15, No.1, 1993, pp11-21, and a copy can be downloaded from **rpimag.co/computervonneumann**

18 Nancy Stern, *From ENIAC to UNIVAC* (Digital Press, 1981, ISBN 978-0133315059), p264, note 47

19 William Aspray, *John von Neumann and the Origins of Modern Computing*, p51

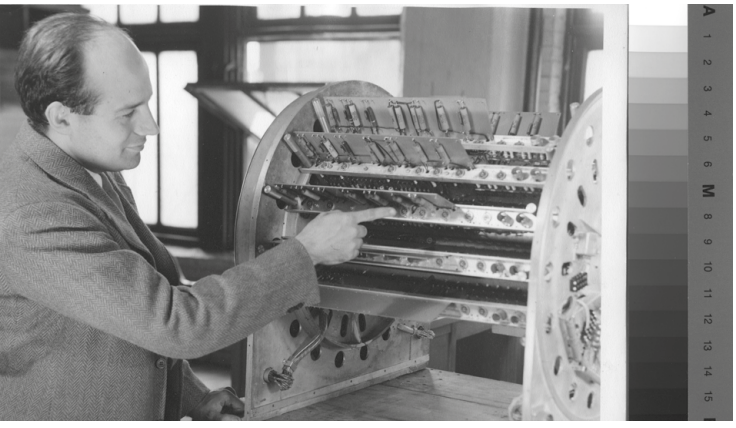
from the more conservative trustees, the IAS director, Frank Aydelotte, added his support to the IAS computer project and von Neumann politely turned down his suitors.

In the meantime, he had been hard at work finding financial backers to build a computer in partnership with the IAS. The main thrust of his argument, particularly to potential military partners already contributing to large-scale computing projects elsewhere, was the need for a dedicated scientific computer. "[It] seems to me that it would be an essentially incomplete policy to develop such devices only for industrial or government laboratories, which have definite, and necessarily relatively narrowly defined, applied problems to which they must devote all or most of the time of their equipment," he explained in a letter to Commander Lewis Strauss of the US Navy.[20]

He concluded: "I have no doubt whatever that we are here on the threshold of very important developments both in pure mathematics and in its applications, and that a pure research institution should spend several years in building a machine and experimenting with it. If we devote in this manner several years to experimentation with such a machine, without a need for immediate applications, we shall be much better off at the end of that period in every respect, including the applications."

20 Miklós Rédei, *John von Neumann: Selected Letters* (American Mathematical Society, 2005, ISBN 978-0821837764), letter from von Neumann to Strauss dated 20 October 1945, p236

> *Von Neumann was stealing ideas*



▲ Presper Eckert demonstrating a BINAC memory unit, c.1948-9
Image: courtesy of the Computer History Museum, CC BY-NC-SA

His argument paid off. On 6 November 1945 the Electronic Computer Project came into being with joint funding from the IAS, Princeton University, the Office of Navy Research, and one private partner: the Radio Corporation of America, better known as RCA.

To partner with the RCA made sense on two fronts. First, much like Bell Labs, it had built computing devices for the military during the war. Second, von Neumann already knew two key figures there: Vladimir Zworykin, who invented an early cathode ray tube called the iconoscope, and Jay Rajchman who was working on a CRT storage technology called Selectron. The RCA funding would cover the cost of the storage for the computer, but they would keep the patents.

First stored-program success

With funding in place, von Neumann started to build his team. His first recruit was Goldstine. Von Neumann offered him the role of deputy director for the computing project in late November. Goldstine took it up once the ENIAC was unveiled in February 1946. Arthur Burks followed a month later, but on a temporary basis: having decided to return to his first love of philosophy on a full-time basis, he had already accepted a position at the University of Michigan. But before the new academic year started, he agreed to work with Goldstine and von Neumann on the computer's logical design.

They also needed engineering talent, and despite their differences of opinion over patents von Neumann still wanted Presper Eckert to head up the team. He offered Eckert the job on 27 November 1945, but Eckert took time weighing up his options. "Eckert was being torn by Mauchly, who of course wanted to stay associated with Eckert, because Mauchly really needed Eckert," said Goldstine.[21] "He was torn by his wife, who wanted to stay in Philadelphia close to her family. He was torn by his parents, who wanted him to stay in Philadelphia. And he was torn by his own desire to make a lot of money."

But von Neumann didn't help matters in January 1946. Francis Reichelderfer, Chief of the US Weather Bureau, wanted to know how electronic computers might help advance meteorology, and asked the RCA's Zworykin to meet with his representatives.[22] Zworykin invited von Neumann along, and they met on 9 January to discuss the exciting new opportunities that lay ahead. This meeting wasn't the problem: that came the next day, when The New York Times published an article entitled 'Electronics to Aid Weather Forecasting'.

21 'Oral history interview with Herman Goldstine' by Nancy Stern, 14 March 1977, Niels Bohr Library & Archives, **rpimag.co/goldstineinterview**, p37
22 Nancy Stern, *From ENIAC to UNIVAC*, p83

▲ James Pomerene working on the IAS machine, holding a Williams tube **Image:** Wikimedia Commons, CC BY-SA 4.0

The story told of "a new electronic calculator, reported to have astounding potentialities, which, in time, might have a revolutionary effect in solving the mysteries of long-range weather forecasting". It went on to name von Neumann and Zworykin as the inventors of this new machine, with the clear implication to any readers being that the RCA and von Neumann were the big forces in this new space.

Von Neumann quickly apologised to the Moore School team and the US Army, who had also been annoyed by the coverage: the ENIAC's official unveiling was due the following month, so this was terrible timing. Eckert was still fuming almost 30 years later. "You know, we finally regarded von Neumann as a huckster of other people's ideas with Goldstine as his principal mission salesman," he told Nancy Stern in 1977,[23] in relation to the First Draft report. He soon added: "Von Neumann was stealing ideas and trying to pretend work done at the Moore School was work he had done." Stern asked, "Over matters like The New York Times leak and things like that, I assume you mean?" A terse, one-word reply: "Yes".

The only surprise, then, is that it was von Neumann who withdrew the offer – in March 1946 – before Eckert refused it. By this time, the ENIAC had been unveiled to the world, and the Moore School realised it could be sitting on a money-making pot of patents. And this would lead to one final tremor, rising to a team-destroying earthquake, that resulted in Eckert and Mauchly both abruptly leaving the Moore School.

It came in the form of Irven Travis, who had been an assistant professor at the Moore School before being drafted to the US Navy. He'd spent the final four years of the war in the Bureau of Ordnance in Washington DC in charge of the contracts with organisations such as Bell Labs, RCA, MIT, and the Moore School. On his return in January 1946, he was the obvious person to take up a role as director of research for the organisation.

Right from the start, Travis knew that one of his biggest priorities was to sort out patents. "The university was in an untenable position," he said in 1977.[24] "It had an obligation under a contract with United States Government that it couldn't fulfil. It had no contract with any employees. So one of the first things I did was to write up a patent agreement, more or less along the lines that I had known about at MIT and Bell Labs."

This agreement boiled down to signing away any past and future patent rights to the university, or leaving with immediate effect. Most of the Moore School engineers chose to sign, but it forced Eckert and Mauchly – the two men pursuing patents for their work on the computer projects – to leave at the end of March. Eckert, as he was wont to do, put it bluntly: "I have a letter from Pender in the safety deposit box which says I resigned; but I was fired, by Irv Travis."[25] The Pender here refers to Harold Pender, dean of the Moore School.

Burks, who had already decided to leave, took a more pragmatic view. "I think that's too extreme to say that they were forced out," he said to Nancy Stern in a 1980 interview, when she put forward Eckert's point of view. "[Every] engineer was told 'if you want to stay in this role you will have to agree to abide by this long-standing, but previously unenforced, university policy on patents'."[26]

The story of the EDVAC, UNIVAC, and Princeton IAS will be continued in issue 166.

23 'Oral history interview with J Presper Eckert' by Nancy Stern, 28 October 1977, Charles Babbage Institute, **rpimag.co/eckertinterview**

24 'Oral history interview with Irven Travis' by Nancy Stern, 21 October 1977, Charles Babbage Institute, University of Minnesota, **rpimag.co/irventravisinterview**, p29

25 As above, p52

26 'Oral history interview with Arthur W and Alice R Burks' by Nancy Stern, 20 June 1980, Charles Babbage Institute, University of Minnesota, **rpimag.co/burksinterview**, p69

Part 10

Conquer the command line: downloading and installing

We look at downloading and unpacking software, and show you how to write images to SD cards

Maker
Richard Smedley
A tech writer, programmer, and web developer with a long history in computers, who is also in music and art.

about.me/RichardSmedley

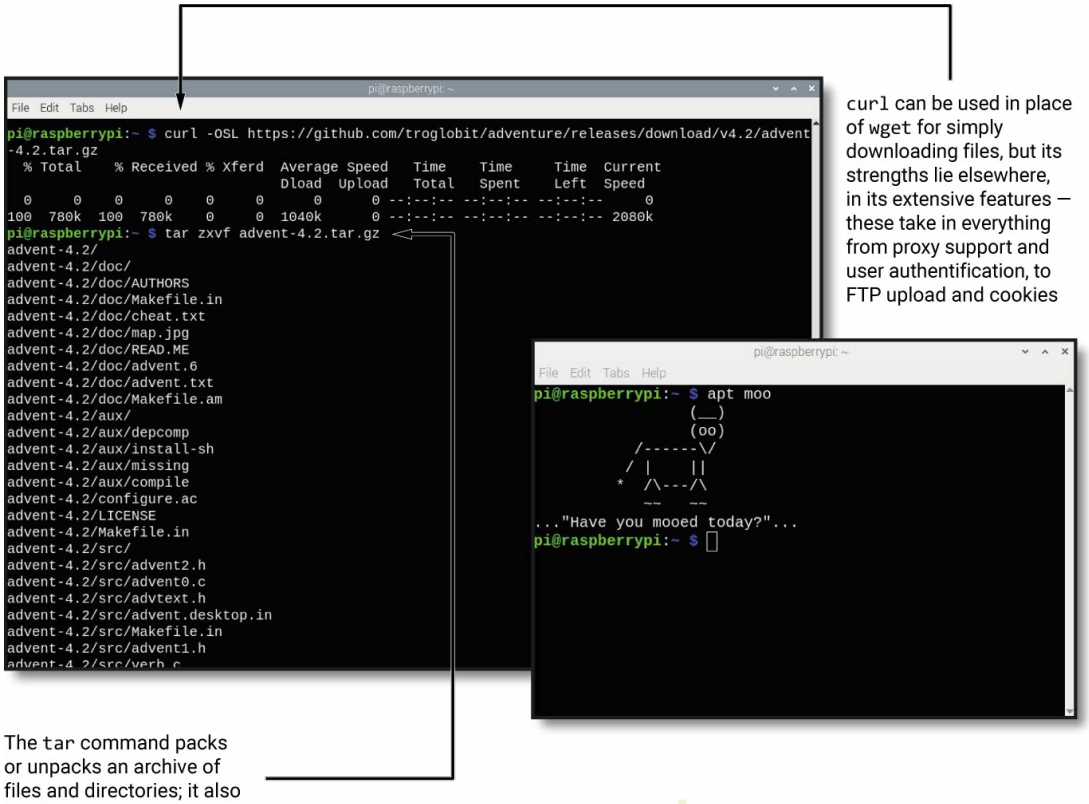
Running an `apt` command (see part 3: rpimag.co/157) allows access to a huge collection of software – several thousands of packages in the main Raspberry Pi OS repository – but sometimes we need to add software from outside the main repository.

If we are lucky, we find that someone has packaged up the software in the .deb format used by Raspberry Pi OS, or even created a whole repository to take care of the dependencies. We'll look briefly both at adding repositories, and dealing with other kinds of downloads.

Adding a repository

Information about software repositories is kept in the **/etc/apt/sources.list** file, which on a new install just contains the Raspberry Pi OS repository. Rather than editing this file to add other repositories, you are advised to add them in a .list file to the **/etc/apt/sources.list.d** directory. To add a new repository, use `sudo nano` to create a .list file there (e.g., **extrasources.list**) and inside it, add a source like the one below (combine this all into one line with a space between `debian/` and `trixie-backports`):

```
deb https://deb.debian.org/debian/
trixie-backports main contrib non-free
```



▲ Downloading and extracting files on the Raspberry Pi OS command line

All Debian releases have been named after characters in Toy Story

Here, `trixie` in `trixie-backports` is the Debian (and Raspberry Pi OS) release name: all Debian releases have been named after characters in the *Toy Story* series of films since 1996 (former Debian project leader Bruce Perens was involved in the early development of Debian while working at Pixar). Bullseye (Woody's horse) was the name of a previous version.

Most software is in the `main` repository, which can be freely copied or mirrored anywhere. Other components, like `non-free`, allow repositories to contain software you may not be free to pass on, keeping it separate from the main repository.

Once you've added the source, you should run `sudo apt update`. This updates Raspberry Pi OS with the list of the backports packages, a collection of backported software from the upcoming version of Debian. Be careful though – they may not work properly. For a full list of what's in that repository, run these commands:

```
  $ cd /var/lib/apt/lists/
  $ for d in
*trixie-backports*binary-arm64_Packages; do grep
^Package $d
  $ done
```

This changes to the **/var/lib/apt/lists/** directory and runs a `for` loop over any file matching the wildcard expression. For each matching file, it runs the commands between `do` and `done`, searching each file for lines that begin with 'Package'. You'll notice that the prompt changes from `$` to `>` after you type the line beginning with for. The shell is waiting until you complete the `do` command with the `done` keyword.

If you want to see details of a specific file, you can run a command like `apt show -a kicad`. If you have the backports repository enabled, you'll see two versions.

EASTER EGG

Read `man apt`, and you may see: "This APT has Super Cow Powers." If it's there, try typing `apt moo` to see what happens.

Here is the output, truncated to show the version and the repository:

```
  Package: kicad
  Version: 9.0.1+dfsg-1~bpo12+1
  APT-Sources: https://deb.debian.org/debian
trixie-backport...

  Package: kicad
  Version: 6.0.11+dfsg-1
  APT-Sources: http://deb.debian.org/debian
trixie/main...
```

If you run `sudo apt install kicad`, you'll get the older version, as backported packages are always inactive by default. To get the latest package, run `sudo apt install kicad/trixie-backports`.

Trusting a repository

When you add third-party repository sources to **sources.list.d**, you need to get a key for it and use `gpg` to install it. This is because third-party sources are considered untrusted by default. For example, an apt repository for the R programming language is available by creating a .list (e.g., **/etc/apt/sources.list.d/R.list**) with the following line:

```
  $ deb http://cloud.r-project.org/bin/linux/
debian trixie-cran40/
```

Then try to run `sudo apt update`; you'll get a warning that the repository signature could not be verified. You can add a key with `gpg` to trust the repository – here is the example for the R programming language source...

```
  $ gpg --keyserver keyserver.ubuntu.com \
      --recv-key
95C0FAF38DB3CCAD0C080A7BDC78B2DDEABC47B7
  $ gpg --armor --export
95C0FAF38DB3CCAD0C080A7BDC78B2DDEABC47B7 \
      | sudo tee /etc/apt/trusted.gpg.d/cran_
debian_key.asc
```

The specific commands (and key signatures) will be listed with any repository you need to add, so you don't need to remember the alphanumerical code for each one!

wget & curl

Wget is a command to download from a given URL. For example, you can download the Raspberry Pi logo with `wget https://rpimag.co/logo.png`.

Wget is a simple but robust download tool, with a powerful recursive feature that helps fetch entire websites, but it does have mild security risks, so be careful using it to fetch scripts. An alternative is curl, a file transfer tool that works with many protocols and can be used for simple downloads. It dumps to stdout by default; to save as a file with the same name as the resource in the URL, use the `-O` switch. For instance:

```
  $ curl -OSL https://rpimag.co/robotfindskitten-
euler.tar.gz
```

Here, the `-S` switch will show any errors, while the `-L` switch will enable curl to reattempt to fetch the requested file if the server reports that it has a different location.

Unzip

The game downloaded above is compressed with gzip, which losslessly reduces the size of files, and can be decompressed with `gunzip`. The contents here are files rolled into a tar archive (instead of .tar.gz, you'll sometimes find they end with .tgz), and the `tar` command can do the decompression and untarring in one:

```
  $ tar zxvf robotfindskitten-euler.tar.gz
```

Note that the dash is not needed for single letter options in tar. The first switch, `z`, calls gzip to decompress the archive, then `x` extracts the contents. `v` is verbose, informing you of the process as it happens, and `f` tells tar to work with the named file(s), rather than stdin. Miss out the `z` and tar should automatically detect the necessary compression operation.

The result in this case is a folder containing, among other things, a configure script to prepare the source code for compilation (read the **INSTALL** file first). Before you try these commands, make sure that the C header files for ncurses (a programming library for creating text-based GUIs) are installed with `sudo apt install libncurses-dev`.

To copy our image, we will need to unmount the secondary microSD card we've plugged in via a USB card reader

The following will change to the source directory, configure the source code, then compile and install it:

```
$ cd robotfindskitten-2.7182818.701
$./configure
$ make
$ sudo make install
```

After you've installed it, you can run the game by running the command `robotfindskitten`. Find kitten, robot!

While gzip is more efficient than zip (and even more efficient options like bzip2 are available), sometimes you'll get a plain old zip file, in which case `unzip` is the command you want.

```
$ unzip compressedfile.zip
```

Disk image

Disk images for Raspberry Pi OS are available to download manually from **rpimag.co/download**; however, you cannot copy it across to a second microSD card (connected to Raspberry Pi via a USB card reader in this case) with regular `cp`, which would simply put a copy of it as a file on the card. We need something to replace the SD card's file system with the file system and contents that exist inside the Raspberry Pi OS disk image, byte-for-byte, and for this we can use a handy little utility called dd.

`dd` converts and copies files – any files, even special devices like **/dev/zero** or **/dev/random** (you can make a file full of zeroes or random noise) – precisely copying blocks of bytes. To copy our image, we will need to unmount the secondary microSD card we've plugged in via a USB card reader. Use `sudo fdisk -l` both before and after plugging in the card (you can also use `df` to see what's mounted) to see attached devices. If, say, a `/dev/sdb` appears, with the size equal to the SD card, then unmount with `umount /dev/sdb1`. Now copy the disk image with:

```
$ sudo dd if=~/Downloads/2024-11-19-raspios-
trixie-armhf.img of=/dev/sdb bs=1M
```

Development of Raspberry Pi OS's ancestor UNIX started in 1969, so we've covered a few utilities with a long heritage in this book, but that `if=` in place of the usual dashes for command-line options indicates a lineage stretching back to the early 1960s, and IBM's Data Definition (DD) statement from the OS/360 Job Control Language (JCL).

Be very careful that the destination matches the correct disk, or you will lose the contents of another storage device! The `bs=1M` is a block size default; `4M` would be another safe option. Now put the card in another Raspberry Pi and go and have fun!

Part 06

Make a motion-sensing alarm

Stop people from sneaking up by creating an alarm that buzzes when it 'sees' them

Maker
Phil King
A long-time Raspberry Pi user and tinkerer, Phil is a freelance writer and editor with a focus on technology.

philkingeditor.com

YOU'LL NEED

- 1× solderless breadboard
- 1× HC-SR501 PIR sensor
- 1× Mini piezo buzzer
- Jumper wires

Need to protect your room or precious items from miscreants or nosy family members? With just a PIR motion sensor and a buzzer wired up to your Raspberry Pi, it's very simple to create an intruder alert. Whenever movement is detected in the area, a loud beeping noise will raise the alarm. To take things further, you could add a flashing LED, an external speaker to play a message, or even a hidden Camera Module to record footage of intruders.

Attach components

First, we need to wire the PIR (passive infrared) sensor to Raspberry Pi. While it could be hooked to the GPIO pins directly using socket-to-socket jumper wires, we're doing it via a breadboard. The sensor has three pins: VCC (voltage supply), OUT (output), and GND (ground). Use socket-to-pin jumpers to connect VCC to the '+' rail of the breadboard, and GND to the '–' (ground) rail. Connect OUT to a numbered row, then use another jumper to connect that row to GPIO 4 (you could also use a socket-to-socket jumper to connect the OUT pin directly to GPIO 4).

Next, we'll hook up the mini buzzer. Place its two legs across the central groove in the breadboard. Note that the longer leg is the positive pin; wire its numbered row to GPIO 3 on the Raspberry Pi to connect it. Wire the row of the buzzer's shorter leg to the '–' rail.

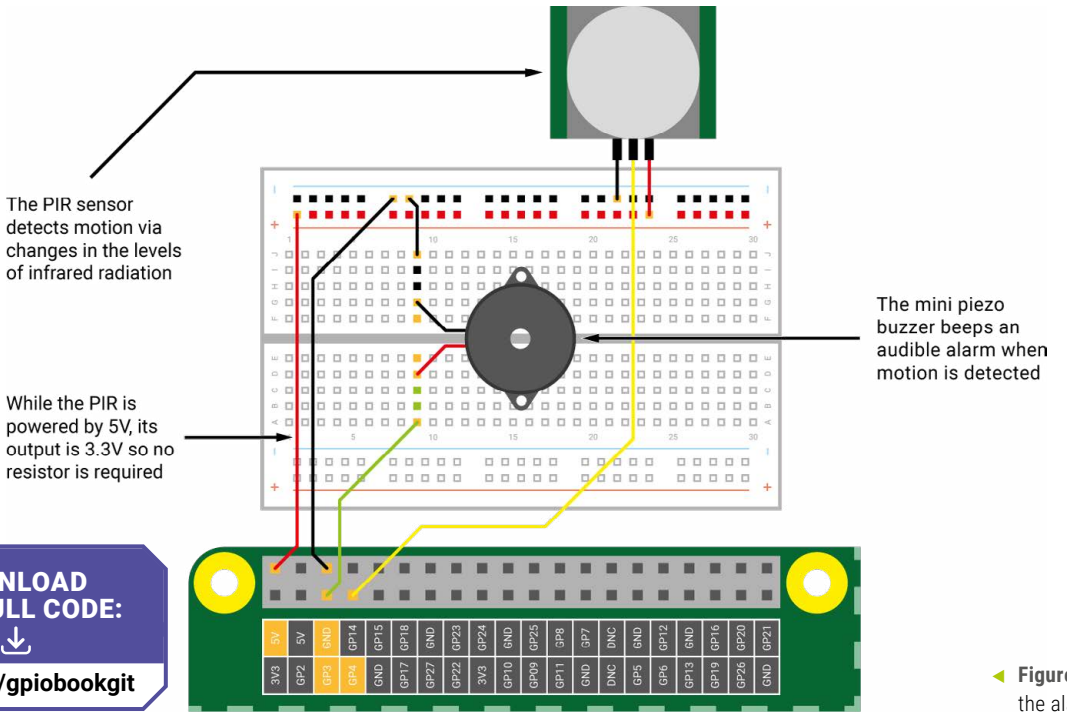
Finally, connect the '–' rail to a GND pin on the GPIO header, and the + rail to the 5V pin. Your circuit should resemble **Figure 1**.

Work on the code

Create a new file, enter the following code, then save it as **motion_alarm.py** and run it (see part two of this series in issue 159 (**rpimag.co/159**) for an overview of editing and running code).

```
from gpiozero import MotionSensor, Buzzer
from time import sleep

pir = MotionSensor(4)
bz = Buzzer(3)

print("Waiting for PIR to settle")
pir.wait_for_no_motion()

while True:
    print("Ready")
    pir.wait_for_motion()
    print("Motion detected!")
    bz.beep(0.5, 0.25, n=8)
    sleep(3)
```

At the start, we import the `MotionSensor` and `Buzzer` classes from GPIO Zero, each of which contains numerous useful methods; we'll need a few of them for our intruder alarm. We also import the `sleep` function from the `time` library so that we can add a delay to the detection loop. Next, we assign the relevant GPIO pins for the PIR sensor and buzzer; we've used



DOWNLOAD THE FULL CODE:

rpimag.co/gpiobookgit

◀ **Figure 1:** Wiring up the alarm

GPIO 4 and 3 respectively in this example, but you could use alternatives if you prefer.

Setting things up

Before starting our motion detection `while` loop, we make use of the GPIO Zero library's `wait_for_no_motion` function to wait for the PIR to sense no motion. This gives you time to leave the area, so that it doesn't immediately sense your presence and raise the alarm when you run the code! Once the PIR has sensed no motion in its field of view, it will print 'Ready' on the screen and the motion detection loop can then commence.

Motion detection loop

Using `while True:` means this is an infinite loop that will run continually, until you stop the program by pressing **CTRL+C**. Whenever motion is detected by the PIR sensor, we get the buzzer to beep repeatedly eight times: 0.5 seconds on, 0.25 seconds off, but you can alter the timings. We then use `sleep(3)` to wait 3 seconds before restarting the loop.

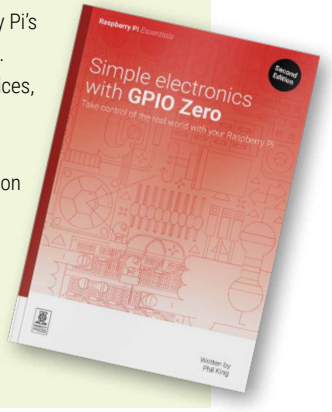
Adjust the sensitivity

If you find that the alarm is going off too easily or not at all, you may need to adjust the sensitivity of the PIR sensor. This is achieved by using a small screwdriver to adjust the plastic screw of the left potentiometer, usually labelled Sx; turn it anticlockwise to increase sensitivity. The other potentiometer, Tx, alters the length of time the signal is sent after detection; we found it best to turn it fully anticlockwise, for the shortest delay of 1 second.

Whenever movement is detected in the area, a loud beeping noise will raise the alarm

Simple electronics with GPIO Zero

This article is an extract from Raspberry Pi's book, *Simple electronics with GPIO Zero*. Updated for the latest Raspberry Pi devices, this book has all the info you need to start creating electronic projects using Raspberry Pi's GPIO pins. Coded in Python with the GPIO Zero library, projects include LED lights, a motion-sensing alarm, rangefinder, laser-powered tripwire, and Raspberry Pi robot.



rpimag.co/gpiozerobook

Remote updates on Raspberry Pi Connect

Create artefacts to perform remote updates on Raspberry Pi computers via Raspberry Pi Connect

Maker
Phil Elwell
Phil is a Senior Principal Software Engineer at Raspberry Pi.

rpimag.co

Raspberry Pi Connect is a secure and convenient way to access your Raspberry Pi remotely from anywhere in the world. I like it because it works behind firewalls and doesn't get confused when the device is given a different IP address, which is a common occurrence here at Raspberry Pi Towers. The remote desktop functionality is great, but a CLI guy like me secretly prefers the more recent remote shell feature; it's much snappier and doesn't take up loads of screen space on your host computer. And best of all, for both kinds of connection, the client software can be whatever browser you have available.

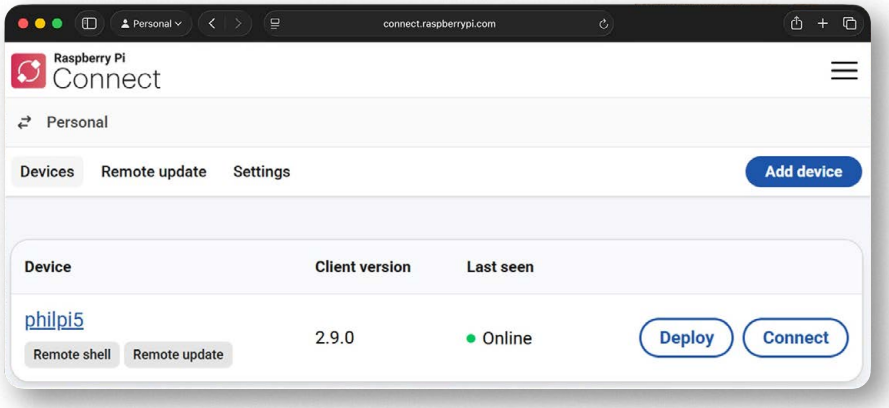
Now it's time to take the next step in the evolution of Raspberry Pi Connect. For the last six months, we've been working away at a brand-new capability: remote updates. The goal is simple enough: make updating the software on your devices as straightforward and convenient as possible.

> *The goal is simple enough: make updating the software on your devices as straightforward and convenient as possible*

An update about updates

Raspberry Pi Connect users can already update their devices in the same way they update any Raspberry Pi: either by using the 'Software Updates' icon in the toolbar or by running one of the update commands (`apt upgrade`, etc.). Now, with remote, 'over-the-air' updates, users can trigger software updates without opening a Connect session. And, because the update is managed through the Connect servers, the device doesn't even need to be switched on at the time – Connect will pick up the update when it is next online.

Before going any further, here is some terminology we use when talking about the remote update process:



▲ Updates can be deployed to Raspberry Pi devices via the Connect interface

Warning!
Experimental feature
Remote update is an experimental feature. For more info, visit **rpimag.co/connectab**.

- **Artefact**: The package that contains the software update or the instruction to perform the update. Also called 'artifact' in some varieties of English.
- **Deployment**: The instruction to install an artefact; the same artefact may be installed repeatedly, but each installation will be considered a unique deployment.

Follow these steps to perform a remote update via Raspberry Pi Connect (you must be running a Trixie version of Raspberry Pi OS)...

1. Install packages

First, install the latest rpi-connect package and the new rpi-connect-ota package:

```
$ sudo apt update
$ sudo apt install rpi-connect rpi-connect-ota
```

Or, if you're using rpi-connect-lite:

```
$ sudo apt update
$ sudo apt install rpi-connect-lite
rpi-connect-ota
```

2. Enable experimental remote updates

Use the rpi-connect CLI to turn on OTA updates:

```
$ rpi-connect ota on
```

Enter your Raspberry Pi OS admin password when prompted. You will see:

```
Experimental remote update enabled
Raspberry Pi Connect restarted
```

3. Create an artefact

Now create the artefact or find one created by somebody you trust. If you need guidance on creating artefacts, see the 'Just do it' section (overleaf).

4. Place the artefact

Put the artefact somewhere accessible to the device(s), which will install it via a URL (it doesn't need to be accessible to the Connect servers). For a public artefact, steps 3 and 4 will have already been done for you.

5. Register the artefact

Register the artefact on the Connect website, along with its SHA-256 checksum, to ensure that it hasn't been corrupted or tampered with.

6. Deploy to your device

Use the 'Deploy' button to send a deployment to your device(s); the artefact can be used repeatedly and for multiple devices.

At this point, the deployment will show as 'Pending'. If the device is online, it will immediately change to 'In progress', rebooting at the end if needed. Once completed, the deployment should show as having 'Succeeded' (a page refresh may be necessary). In the event that something goes wrong, the status will read 'Failed', and clicking on it will reveal an associated error message. Pending deployments can be cancelled manually, or automatically when another deployment is queued up.

From A to B

Some remote devices are more remote than others – sat in the loft, stuck up a tree, or installed at a relative's house – and you don't want them to fail to come back online after an update. This is especially true for appliances and other devices with minimal user interfaces, none of which are about being a computer. Scenarios like these call for a means to try an update, as well as a way to automatically retreat back to safety if it doesn't work out.

The only sure-fire way to do this is to have space for two versions of the software on every device: the active version, and the space (or slot) where the other will go. This is commonly referred to as an A/B scheme, in which every update swaps the roles of the two slots – slot A goes from active to other (or vice versa), and slot B goes in the opposite direction. The clever part of this system, the part that protects against bad updates, is the device's ability to reboot into the other slot without permanently flipping the switch. If the boot is successful, the device can commit to making the new slot active. If, instead, the software crashes or reboots spontaneously, this can be detected and the swap can be abandoned.

An A/B update scheme relies on some simple hardware features, including a small amount of state that can survive a reboot (but not a power cycle), and a watchdog that can trigger a reboot if the software fails to give it attention. All Raspberry Pi devices have these features. Unfortunately, OS images also need to be created with these slots, and with extra partitions for data that should persist across updates. This has not been the case for any released OS that we know about to date (though Dave Jones at Canonical has discussed his plans for future Ubuntu releases).

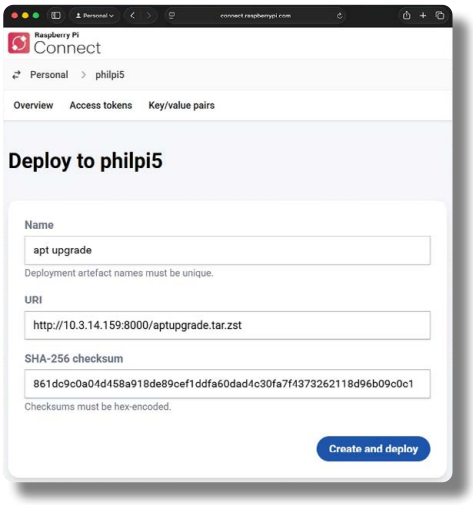
For developers basing their products around Raspberry Pi hardware, our rpi-image-gen tool makes it relatively easy to create images with this A/B structure. A forum post (**rpimag.co/connectab**), written by our web guru Paul Mucur, describes the process and how to use the result to deploy remote/over-the-air updates on Raspberry Pi Connect.

Just do it

Up until this point, artefacts and deployments have been described in terms of software updates, but they can also include arbitrary tasks to be performed on your devices. An artefact can contain one or more scripts to execute, along with data files that these scripts can use. These scripts are run as root so that they can access all of the file systems. I've already referred to running `apt upgrade`, but for custom software updates, you could create an artefact containing a .zip file and a script to extract its contents. Alternatively, it could just do something simple, like triggering music playback or turning on an LED.

To demonstrate how simple it can be to create your own artefacts, here's a walkthrough of the process...

An `apt upgrade` artefact may look like the code below.



▶ **Figure 1:** A deployment artefact using the URI of your host Raspberry Pi and a path to the file. The SHA-256 checksum ensures that it has not been tampered with

aptupgradescript

DOWNLOAD THE FULL CODE: rpimag.co/github

> Language: **Bash**

```
#!/bin/sh
export DEBIAN_FRONTEND=noninteractive
apt update
if apt -y -o DPKG::Options::="--force-confnew" upgrade >
output.txt; then
    if [ -r /var/run/reboot-required ]; then
        echo Rebooting to finish the upgrade
        exit 2 # EXIT_REBOOT
    fi
else
    echo Upgrade failed:
    echo
    cat output.txt
    exit 1 # EXIT_FAILURE
fi
echo Upgrade complete
exit 0 # EXIT_SUCCESS
```

The main upgrade command needs some persuasion to run non-interactively. Place it in a file called **aptupgradescript**. Note: do not include a .sh script extension to the filename.

Create a YAML control file called **aptupgrade.yaml** to go with it, containing the code here.

aptupgrade.yaml

DOWNLOAD THE FULL CODE: rpimag.co/github

> Language: **YAML**

```
001.  # Run apt upgrade on a remote device
002.  artefact:
003.    name: aptupgrade
004.    version: 1.0
005.    device_type: rpi
006.
007.    payloads:
008.    - name: aptupgradescript
009.      type: script
```

Run the otamaker utility to turn this into a Zstandard-compressed artefact:

```
$ otamaker aptupgrade.yaml
```

Terminal will output the following:

```
Contents:
  _contents_.yaml
  aptupgradescript

Artefact: aptupgrade.tar.zst
SHA256:
861dc9c0a04d458a918de89cef1ddfa60dad4c30fa7f4…
```

Find your host computer's IP address on your local network:

```
$ hostname -I
```

Run an HTTP server on your host – this wouldn't normally be the device targeted for update, but it will work if you only have one Raspberry Pi:

```
$ python3 -m http.server 8000 --directory .
```

Press the 'Create and deploy' button to create a new artefact on the Connect server, using your own IP address and SHA-256 values (as in **Figure 1**).

The new artefact will be automatically deployed to your device (**Figure 2**). Give it some time to run – it may need to reboot at the end.

Just the facts, ma'am

The otamaker utility will ship as part of the rpi-connect-ota package. However, for anyone building artefacts on a non–Raspberry Pi platform, it is also available to download from our 'utils' repository on GitHub. There, you will also find some preliminary documentation on the syntax and features of the artefact description files and scripts.

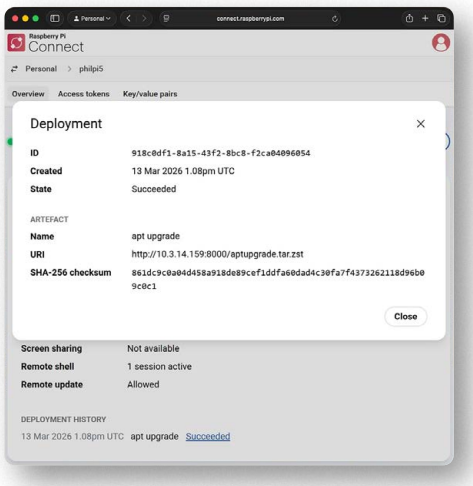


Figure 2: The artefact is deployed to Raspberry Pi

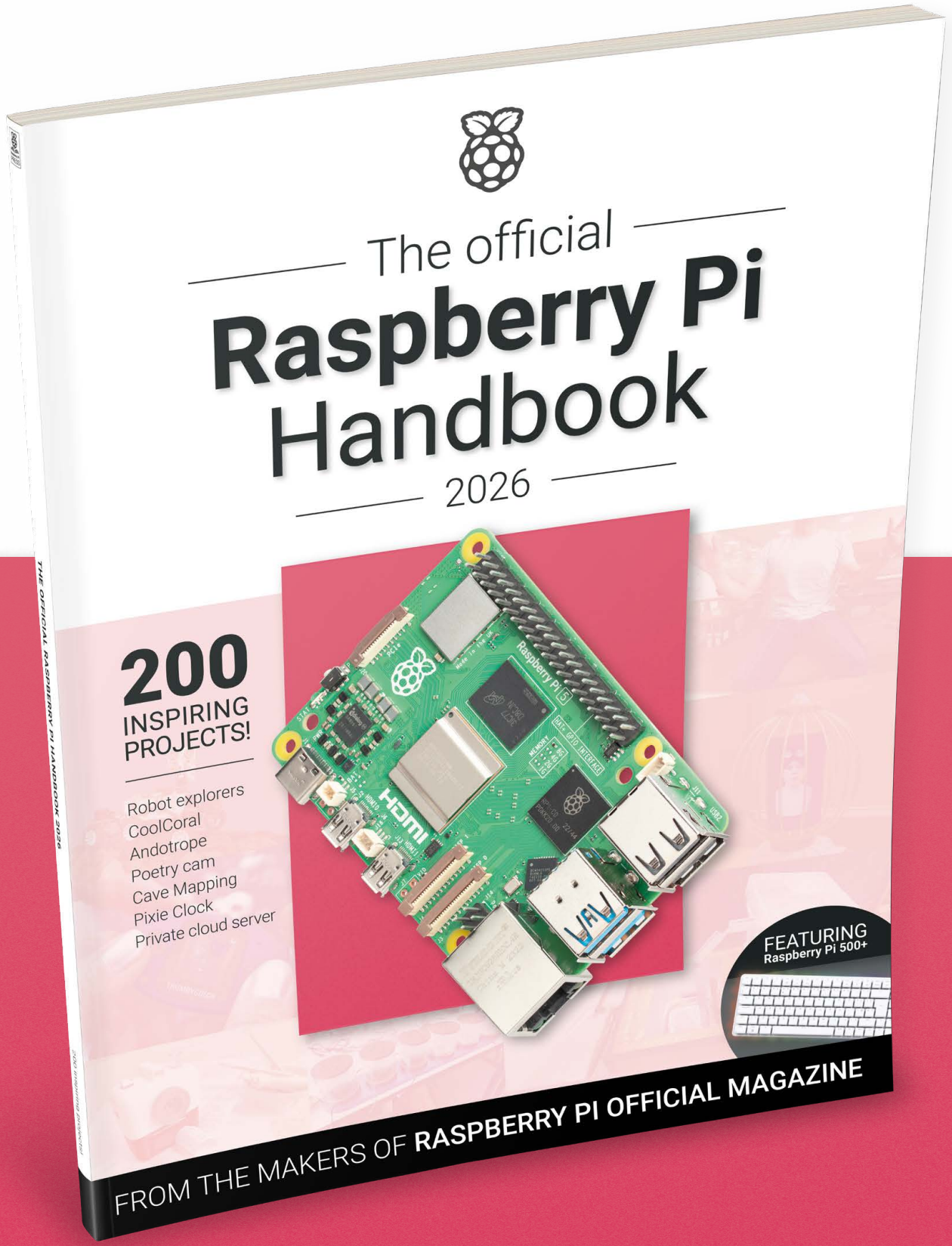
And finally

I hope this has given you a feel for what Raspberry Pi Connect's remote updates feature can do. It's still in beta, and there are a number of obvious limitations, the largest being the lack of an easy way to retrieve outputs from any scripts that run. An interim workaround is to use remote shell to log in, and then:

```
$ journalctl -t rpi-ota-connector
```

… to read the script output.

Ideas for the future include creating and hosting some common public updates, such as an official `apt upgrade`. There will also be careful integration with Connect for Organisations, along with its device-filtering capabilities; being able to update a fleet of devices in just a few clicks is a superpower I need.


The official
Raspberry Pi
Handbook
2026
200
INSPIRING
PROJECTS!
Robot explorers
CoolCoral
Andotrope
Poetry cam
Cave Mapping
Pixie Clock
Private cloud server
FEATURING
Raspberry Pi 500+
FROM THE MAKERS OF RASPBERRY PI OFFICIAL MAGAZINE
THE OFFICIAL RASPBERRY PI HANDBOOK 2026

The official Raspberry Pi Handbook 2026

200 PAGES OF RASPBERRY PI

Get started guide covering every Raspberry Pi

Everything you need to know about the brand new Raspberry Pi 500+

Inspiring projects to give you your next project idea

Build a Raspberry Pi 5-powered media player

Explore the world around you with roving robots

Play retro horror games on Raspberry Pi 5

Buy online: rpimag.co/handbook2026

SUMMER PROJECTS

STAY COOL THIS SUMMER BY TAKING YOUR PICK OF THESE AMAZING PROJECTS.

BY DAVID CROOKES

As the days stretch lazily into golden evenings, our thoughts naturally turn to the great outdoors and all that it has to offer. But whether you intend to explore or enjoy your own personal outside space, you can make even more of the summer months with the aid of Raspberry Pi.

As we're about to see, there are many wonderful projects that will not only help you to make the most of your surroundings but also take away a lot of the hard work involved in maintaining a pleasant environment. There are projects that will monitor nature, those that will help you get out and about, and many that will help you keep an eye on the weather. You can also take the strain out of tending to your garden.

So grab a refreshing drink (**rpimag.co/drinksmachine**) and join us as we take a look at projects that could help make this summer your best yet.

WILDLIFE

- There's a kit which comes with all of the components that you need
- It's a great project for both beginners and advanced enthusiasts

NATUREBYTES WILDLIFE CAMERA

naturebytes.org

If you have access to land teeming with wildlife, Naturebytes can effortlessly capture some stunning close-up shots. The full kit includes a Raspberry Pi Model A+, a Camera Module, a PIR sensor, an SD card with preloaded software, and the all-important weatherproof case fitted with a Fresnel IR lens. You can, however, also buy the case alone and fit it with your own components, perhaps adding a NoIR Camera Module and IR LEDs for night-time footage.

Once it's assembled and fastened to a tree, you can use the device to detect and capture anything that is moving in view. After you've amassed a few photos and videos, you can then begin to dive deeper, perhaps by opting for a more advanced Raspberry Pi computer, coding your own features, or using machine learning and computer vision to more easily identify the many visitors to your space.

GETTING OUT AND ABOUT

If you're looking to get active this summer, you don't need to leave your beloved Raspberry Pi behind. Cyclists may want to make a speedometer and tracker for their bike. There are a number of projects you can use for inspiration, such as Bike Dashboard (**rpimag.co/bikedash**) and Shareable Smartbike (**rpimag.co/sharebike**). The latter even turns the lights on when it gets dark (which, blissfully, is now getting later and later).

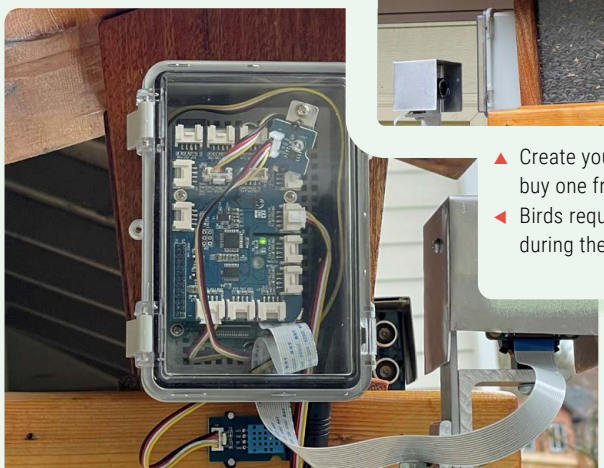
If you prefer to get around on foot, you can have some fun tracking down geocaches, which are hidden containers stuffed with trinkets. Pi Hiker (**rpimag.co/pihiker**) will point you in the right direction. Worried about getting lost? Then make your own low-power live GPS tracker (**rpimag.co/livegpspi**). You can also work on a project to measure altitude with a BME280 sensor and Raspberry Pi Pico, which is ideal if you're looking to hike in hilly or mountainous areas.

A BIRD IN THE HAND

rpimag.co/birdinhand

This smart bird feeder is not only likely to welcome many feathered friends, it's also ready to do its bit for science. Inspired by the Great Backyard Bird Count initiative which runs every February, the project's primary aim is to automatically identify and keep a tally of every visitor. But it's also ready to help feed birds and take motion-triggered photographs, so it's a great device to make and use all year round.

Maker Jeff Stockman has used a Raspberry Pi 3B connected to a Camera Module. He has also included a GrovePi+ HAT to add temperature, humidity, and light sensors. Images are analysed using the free tier of Microsoft's Azure Custom Vision machine learning service, but there are plenty of alternative methods. You may even want to check out the Smart Bird AI project (**rpimag.co/smartbirdai**), which uses a Raspberry Pi 5 computer connected to a Camera Module 3 and sends email notifications when a bird flies by.



- Create your own bird feeder or buy one from a shop
- Birds require high-protein food during the summer months

WEATHER

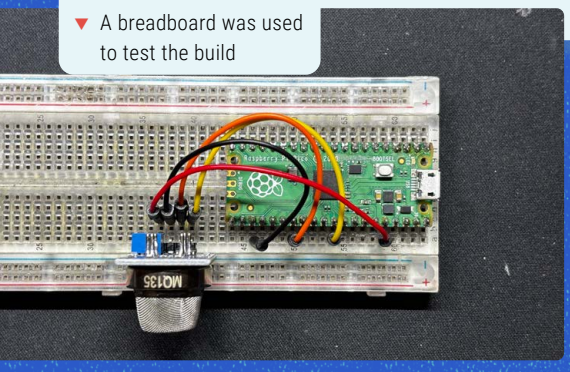
PORTABLE AIR QUALITY MONITOR

rpimag.co/portairqual

Want to get out and about in the fresh air over summer? Then you may want to check you're not filling your lungs with pollutants. Arnov Sharma's handheld air quality monitor is capable of detecting smoke, carbon dioxide ammonia, benzene, nitrogen oxide, and alcohol. This means it can display the levels of gases produced by the incomplete combustion of fossil fuels, wood, and coal, as well as some of those pumped out by vehicles and emitted from hazardous waste sites. It can also detect the amount of livestock waste, decaying organic matter, and fertiliser production.

▼ Air quality is displayed as a number together with an explanation

This is achieved by combining a Raspberry Pi Pico 2 with an MQ135 air quality sensor and a 0.96-inch OLED screen. It makes for a low-cost, relatively simple build and while it's not quite on a par with official Air Quality Index measurements, it's nevertheless a fun and potentially eye-opening project to pop in your pocket while on your travels.



▼ A breadboard was used to test the build

▶ Personalise the project with your own case design

▼ The project is open source, so feel free to tweak it

FLIGHT TRACKER WITH WEATHER

rpimag.co/flighttracker

This cool-looking device will not only tell you how hot it is outside over the coming days, it'll also let you check details of any overhead flights so you can watch with envy as plane-loads of passengers make their way to exotic locations. Created by Adam Paulson, it uses a 64 × 32 RGB matrix panel instead of a screen to give it an eye-catching appearance. The other main component is a Raspberry Pi 3A+ computer, with Adam's software handling the rest.

Pulling in-flight information from Flightradar24 API, the device displays the airports that planes are travelling between as well as the name of the airline, flight number, airplane type, and distance. If there is no flight overhead, the screen switches to the time, temperature, and a three-day weather forecast. The project also makes good use of colour to add further information. The temperature changes colour based on current humidity levels, for instance, and the time changes colour at sunrise and sunset.

OFF-GRID WEATHER STATION

rpimag.co/offgridweather

This is a more advanced weather station project: one that doesn't rely on data in the cloud or an internet connection. Instead, it gathers data using an Ecowitt WS90 outdoor sensor which communicates with a Raspberry Pi computer via an RTL-SDR – a software-defined radio dongle.

The all-on-one sensor monitors the temperature, humidity, wind speed and direction, solar radiation, UV, and rainfall. It then broadcasts raw RF packets of data which the RTL-SDR can pick up. By using the rtl_433 tool to decode the 915MHz radio transmissions into clean JSON packets, maker Vinnie Moscaritolo is able to ensure the Raspberry Pi can receive and make use of sensor updates every five seconds. His program can then display the data in a readable form using a local JSON API to generate a simple dashboard.

▶ Ensure the components are in a weather-proof box

▼ Vinnie connected the sensor to a PVC pipe and stuck it in the ground

▶ VBIT-Pi makes use of a Raspberry Pi 3B+ running customised Raspi-teletext software

NMS CEEFAX

rpimag.co/nmsceefax

The BBC's Ceefax service ran from 1974 to 2012, giving millions of people access to news and sport headlines, TV listings, showbiz tittle-tattle, and financial information. It also allowed anyone with a compatible television to check the latest weather forecasts nationally, internationally, and regionally via page 400. And, as it happens, much of this is still available thanks to a group of enthusiasts keen to keep the memory of teletext services alive.

Key developments over the years have included Peter Kwan's VBIT-Pi project and Alistair Buxton's Raspi-teletext software, but Nathan Dane brought of those both together by creating software which scrapes the BBC website, parsing the links and extracting text from each page while stripping away photos and videos. The result is NMS Ceefax, which also uses weather data taken from the Met Office's API, and the output can be viewed by going to **nmsceefax.co.uk**.

If you'd rather get more hands-on, you can look to create your own teletext service (**rpimag.co/teletextpi**): PJ Evans explains how you can use Peter Kwan's updated VBIT 2 teletext streaming project (**rpimag.co/vbit2**) and Alistair Buxton's raspi-teletext (**rpimag.co/raspitele**).

▶ NMS Ceefax has a full up-to-date weather service

GARDENING

POLLINATOR CAMERA

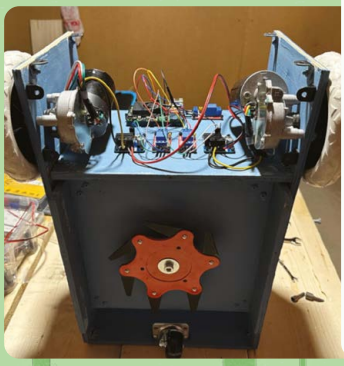
rpimag.co/intelligarden

Created as part of the Avanade Intelligent Garden which won a Gold award for Best Construction at Chelsea Flower Show in 2025, Pete Gallagher's Pollinator Camera is a project that's certainly worth considering. Grab a Raspberry Pi 5 computer and a Raspberry Pi Camera Module 3 along with an SD card, PSU, ribbon cable, a 3D printed camera case, and the awesome Raspberry Pi AI HAT+ (26 TOPS version) and you're ready to go.

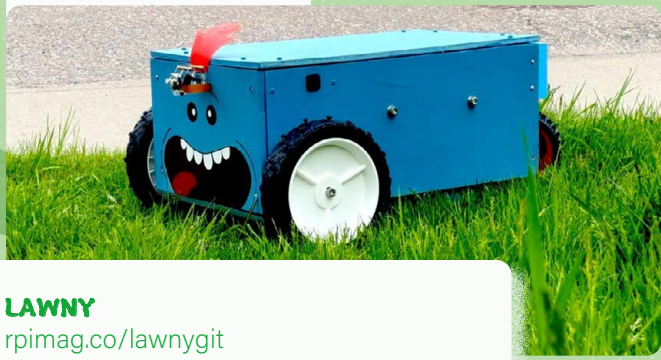
The Pollinator Camera can detect flowers, bees, and butterflies, keeping a running total of the number of insects while capturing images and videos. Bounding boxes are drawn around detections, while footage and stills are saved in Azure Storage where they can be used for further machine learning training of a YOLO model. It's a great way to engage in citizen science and monitor the hard work your flying guests are carrying out in your garden.

▲ A Pollinator Camera was also placed in a bird box

◀ Peter Gallagher set up the Pollinator Camera hardware in the Intelligent Garden



▼ Windscreen wiper motors are used to drive this mower



LAWNY

rpimag.co/lawnygit

It looks like the lawnmower equivalent of the Henry vacuum cleaner thanks to its expressive cutesy face. But this is arguably more special because it has smarts powered by a Raspberry Pi 5 computer and, given it'll cost around £250 to make, it's also something of a bargain build.

The main components are a Raspberry Pi Camera Module 3, a windscreen wipers motor, a grass cutter electric motor, relays, and a steel razor strimmer head. But what also makes Eugene Tkachenko's build special is the fact it can be controlled using a phone or desktop computer, because it's operated by a Node.js server. Putting the components together is the tricky bit, but the case should be easy enough to build because it's created out of plywood. You can then add any images you want to make the mower your own before sitting back and relaxing, allowing your new robot friend to put in the hard work while you enjoy some lazy summer days in the garden. What's not to love?

Warning!

Sharp blades

Be careful when handling the Lawny project. It has moving blades. Children should be supervised.

◀ Lawny can also be made using a Raspberry Pi Zero W

The herbicide jug is mounted to Herbie_Bot

HERBIE_BOT

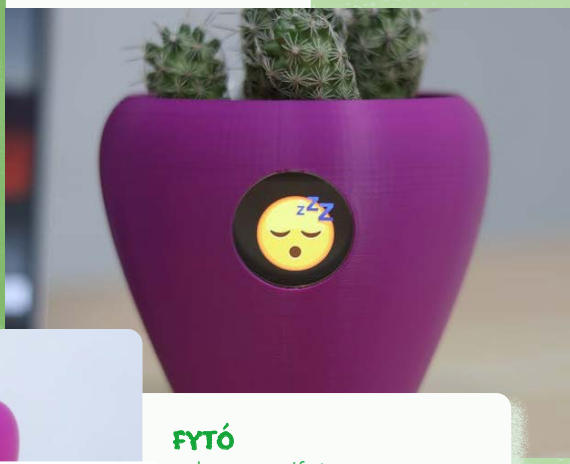
rpimag.co/herbie

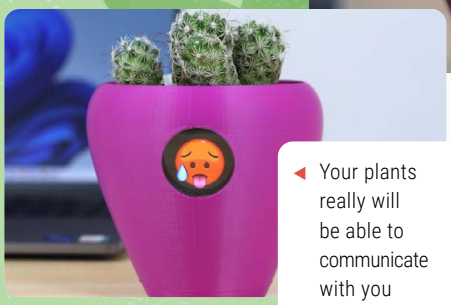
Nobody relishes the task of weeding a lawn, least of all maker Russ Hall. And that's why he decided to level up a robot he'd previously built to do the job for him. By fitting it with gallon jugs of lawn-safe herbicide and a battery-powered spray wand, he's been able to automate the task, using the OpenCV OAK-D Camera and AI to detect and tackle only the areas which need attention.

Russ shows how it's done on his GitHub, explaining how a Raspberry Pi 4 computer with 4GB RAM is used to run the software and interact with the camera. The Movidius chip in the camera makes light work of object detection while an Arduino Mega, working with wheel encoders, detects the device's speed to ensure greater accuracy when spraying.

Keep that lawn looking lush and weed-free

You could look to use a shop-bought flower pot





Your plants really will be able to communicate with you

FYTÓ

rpimag.co/fyto

Don't have a garden? You can still get stuck into a fun gardening-related project courtesy of Fytó, a smart 3D printed planter with added personality. By connecting sensors capable of detecting soil moisture, temperature, and light exposure to a Raspberry Pi Zero 2 W, it's possible to work out the requirements for your plants. These can be then displayed on a Waveshare 2-inch LCD screen (or Waveshare's 1.28-inch round LCD module) as one of six emoji facial expressions – and that will tell you if the plant is thirsty, hot, cold, lacking light, well-watered or happy.

It's a genius, eye-catching idea by makers Shebin Jose Jacob and Nekhil, who are the co-CEOs of Coders Cafe, and it would look amazing both inside the home or outside on a patio or balcony. Full build instructions are available along with design files for the flower pot, while the code can be found at **rpimag.co/fytogit**.

ONLY THE **BEST**

Educational kits

Raspberry Pi-based kits to help you learn electronics, coding, and more

By **Phil King**

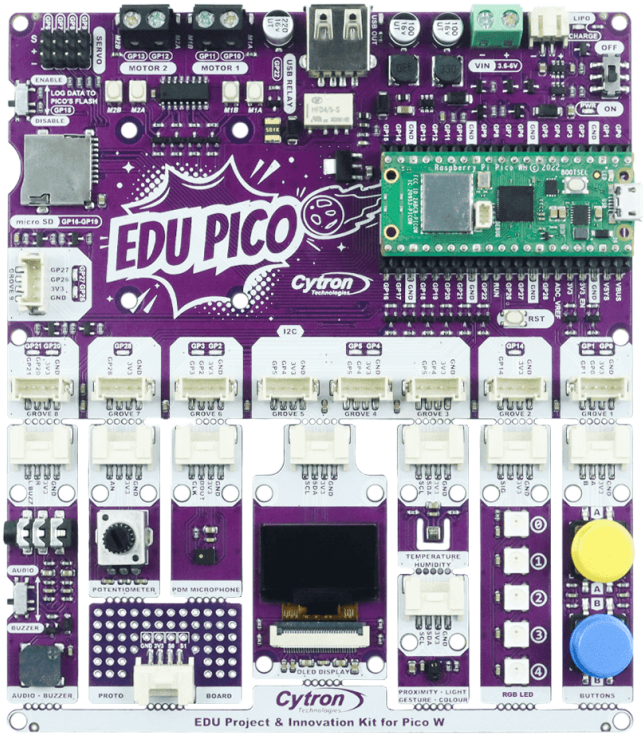
You could just go and buy some individual electronics components and learn about wiring circuits and coding from online guides or books, but dedicated educational kits can make the process a lot more convenient. As a quick perusal of popular online retail sites shows, there are certainly plenty of them to choose from, ranging from small to very large with hundreds of components.

Size isn't everything, however, and the ease (or difficulty) of the learning process depends greatly on the quality of the supporting documentation and tutorials – whether in printed form, downloadable PDFs, websites, or video guides.

The sheer amount of choice can be daunting. To help narrow the field, we've rounded up some of the best educational kits for Raspberry Pi computers and the Pico series of microcontrollers. Some are aimed at complete newbies, or STEM educators, while others are more advanced and branch into the topics of robotics and AI. Let's get learning...

EDU PICO

Cytron / The Pi Hut | £51 / $68 | cytron.io / thepihut.com



Leave the components in place on the PCB or snap them off and connect them manually

This Pico-based learning kit comprises a motherboard, 188-page guidebook, plus a few other bits. You can buy it with a Pico W included or use your own, plugging it into the board's female GPIO headers – which are doubled to break out all the pins.

The board has numerous components on it, including a couple of sensors (temperature/humidity and four-in-one light, colour, proximity, gesture), two push-buttons, a piezo buzzer, microphone, RGB LEDs, potentiometer, and mini OLED. You can leave these in situ, pre-wired to Pico's GPIO pins via the PCB, or snap them off and connect them via nine Grove ports. In addition, the kit includes a servo and DC motor (with attachable fan), which connect to other dedicated ports on the board. Bonus features include a USB relay (to switch a connected device), microSD slot, and LiPo charging port.

Aided by quick reference cards, the detailed guidebook takes you step by step through CircuitPython-coded projects such as a gesture reaction game, automated waste bin, noise pollution monitor, and a climate control greenhouse made from the fold-out clear plastic supplied in the box.

Verdict

An easy-to-use kit with an excellent guidebook to aid learning.

AI Fusion Lab Kit

SunFounder | £74 / $99 | sunfounder.com

A seriously large electronics kit, it's based around the Fusion HAT+ we reviewed in issue 162 (rpimag.co/162). Compatible with any 40-pin Raspberry Pi, this robotics board features twelve PWM channels (for servos), four motor ports, four digital and four analogue inputs, and a portable power option using the supplied twin 18650 battery pack. For adding voice control and sound to projects, there's a mic and speaker.

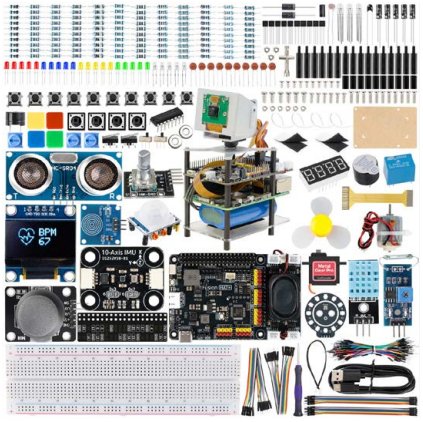
The kit expands on this with a big box of components, including ten sensor modules: photoresistor, thermistor, tilt, touch, reed switch, PIR, ultrasonic, rotary encoder, temperature/humidity, and ten-axis IMU. Other inputs include buttons, switches, a potentiometer, and joystick. You also get a SunFounder camera module and pan-tilt mount. There are tons of LEDs and resistors, along with a mini OLED, seven-segment display, RGB LED, WS2812 LED ring, buzzer, and jumper wires. Plus a DC motor, relay switch, and two servos.

Extensive online documentation (**rpimag.co/aifusionlabdoc**) takes you step by step through a host of projects, taking in electronics (with Python coding), LLMs, and computer vision.

Verdict

A big box of components, plus the Fusion HAT+ and learning resources.

▼ That's a whole load of components to get you started!



CamJam EduKit

The Pi Hut | From £6 / $8 | thepihut.com

Verdict

Three low-cost kits to get you started with electronics and robotics.

▲ All about sensors, the EduKit #2 packs its components into a handy tin

Created by the founders of Cambridge Raspberry Jam (CamJam for short), this range of pocket-money kits is ideal for learners on a budget. The EduKit #1 takes you through the basics of connecting electronics to a Raspberry Pi. In its metal tin you'll find a 400-point breadboard, three single-colour LEDs (red, yellow, green), push-button, piezo buzzer, resistors, and jumper wires.

With these, you can use the series of downloadable worksheets to learn how to blink LEDs, detect user input, make sounds, and more. It's all explained well, with Python code examples using GPIO Zero.

Moving onward, the EduKit #2 is all about sensors and, along with other components, includes three of them: PIR, light, and temperature. Projects detailed in the worksheets include a basic burglar alarm and night light.

Finally, the EduKit #3 enables you to build a robotic car using a dual H-bridge driver board, two DC motors, wheels, battery box, and ultrasonic sensor. You can even use the cardboard box as a makeshift chassis (or supply your own).

PicoBricks Base Kit

PicoBricks | £53 / $71 | picobricks.com

Aimed mainly at STEM educators, the PicoBricks Base Kit comes with a Pico W slotted into the female headers (with extra breakout rows) on its large PCB. As with the similar EDU PICO, the board has numerous component modules that are wired to Pico, but can be detached and connected manually using the Grove ports if needed.

On-board components include an RGB LED, button, sensors (temperature/humidity, light, and IR), buzzer, OLED, motor driver, and relay. In the box, you also get cables, a battery holder, mini breadboard, and IR remote.

To help you on your learning journey, there are three e-books. One of these takes you through 25 electronics projects of increasing complexity, from blinking an LED to setting up a mini smart greenhouse. Coding can be done using MicroPython or a block-based IDE.

You may need to supply extra components for some projects. Or you can buy the expanded 'Zero to Hero' PicoBricks kit to add robotics projects.

Verdict

A solid all-in-one STEM learning kit based around Pico.

◀ A Pico-based educational kit with multiple coding options

Project Box 1 for Raspberry Pi

MonkMakes / The Pi Hut | £13 / $17 | monkmakes.com / thepihut.com

▼ Just enough components to make ten projects

The Project Box 1 has all the components needed to make ten simple projects. It's been put together by Simon Monk (author of the *Raspberry Pi Cookbook* and *Programming Raspberry Pi*), along with a 45-page downloadable tutorial booklet to help you learn the basics of Python coding and electronics.

Projects covered in the booklet range from blinking one or two LEDs to making a reaction timer, light harp (using a phototransistor and buzzer), and proximity detector. It also includes a basic setup guide and info on resistor colour codes and the GPIO Zero, GUI Zero, and Pi Analog Python libraries.

In the box, you get a 400-point breadboard, a few resistors of differing values, a couple of red LEDs, an RGB LED, two push-buttons, a phototransistor, thermistor, capacitor, and buzzer, plus a bunch of jumper wires. While not as extensive as some kits, it may well prove less daunting and confusing for beginners. A handy Raspberry Leaf card helps you to identify GPIO pins, too.

Verdict

A smaller learning kit that's ideal for absolute beginners.

CrowPi 3

Elecrow | £172 / $229 | elecrow.com

This electronics lab in a briefcase is crammed with components. First, you need to slot a Raspberry Pi 5 into an adapter board hidden by a panel on the underside. This reroutes the USB ports and sends video output to the kit's built-in 4.3-inch 800 × 600 touchscreen with a camera on the top.

A raft of on-board sensors include gyroscope/accelerometer, temperature/humidity, light, sound, PIR, ultrasonic, touch, tilt, Hall effect, RFID, and flame. In addition, extra components (supplied in a bag) can be inserted into a breadboard and connected to a GPIO breakout header. You can even plug in a Pico (or other microcontroller). There's also an 8 × 8 RGB LED matrix, seven-segment display, small LCD, four buttons, and joystick, plus interfaces for I2C, UART, servo, stepper, and motor.

The hands-on learning support is superb, with over 150 lessons with fun projects to try out, grouped into the topics of AI, Python, Node-RED, Pico, micro:bit, Arduino, and Minecraft.

Verdict

A slick way to learn about electronics, coding, and AI.

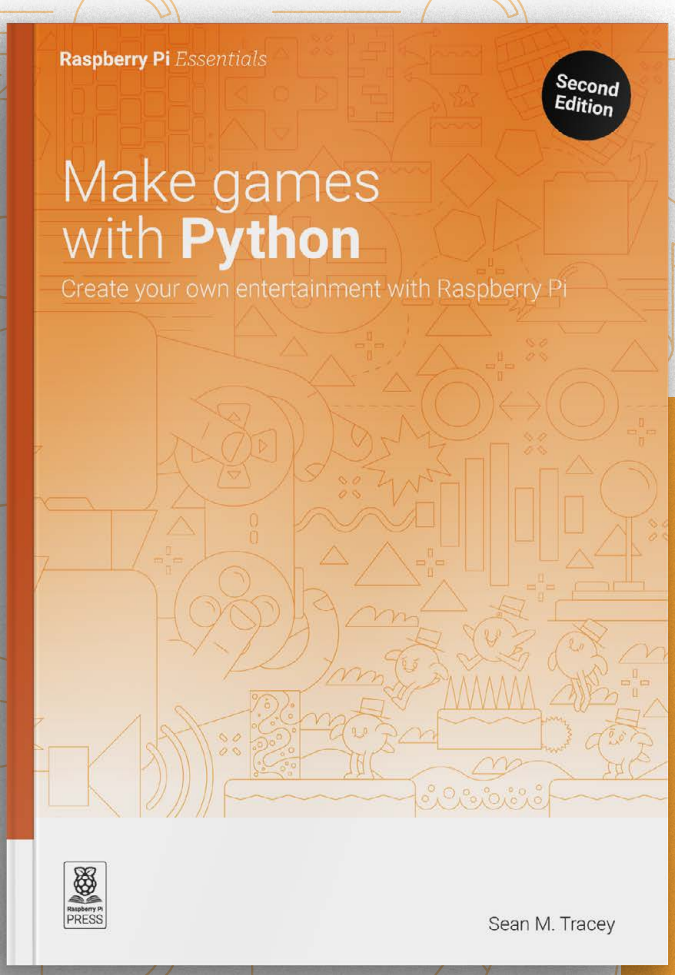
▲ A Raspberry Pi 5-powered electronics lab in a case – what's not to like?

▼ The kit includes pin headers to solder to your Pico

ELECTRONICS KIT 1 FOR PICO

MonkMakes | £16 / $21 | monkmakes.com

Another neat beginner's kit put together by Simon Monk, this one is for Raspberry Pi Pico. In the box are all the components – including a servo motor – needed for the ten MicroPython-coded projects covered in a downloadable 68-page booklet.



While millions of us enjoy nothing more than spending hours racking up high scores on our favourite video games, too few are exposed to an even more gratifying way to spend time — making them.

This book teaches Python and Pygame development, helping you to understand the games you play and create almost anything your imagination can come up with.

- ***As you work your way up to creating your own shoot-'em-up game, you'll learn how to:***
  - *Create shapes and paths*
  - *Move sprites and detect collisions*
  - *Handle keyboard, mouse, and gamepad input*
  - *Add sound and music*
  - *Simulate physics and forces*

BUY ONLINE: **rpimag.co/makegamesbook**

PiPower 5

Keep your Raspberry Pi powered up with this versatile UPS. By **Phil King**

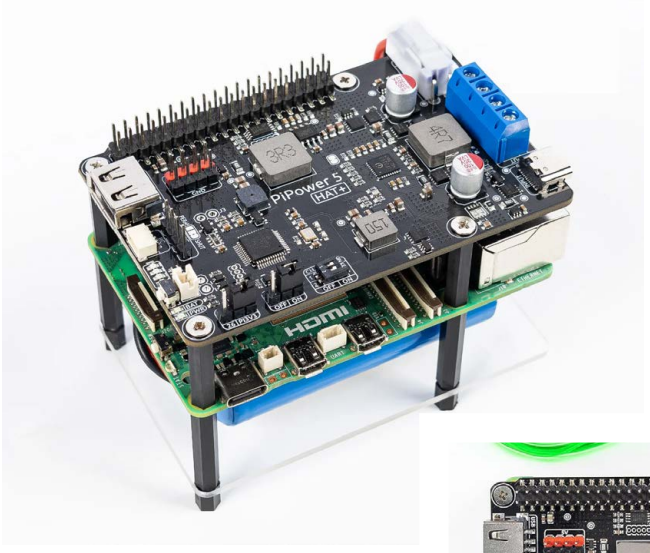
SunFounder **rpimag.co/pipower5** £35/$35

SPECS

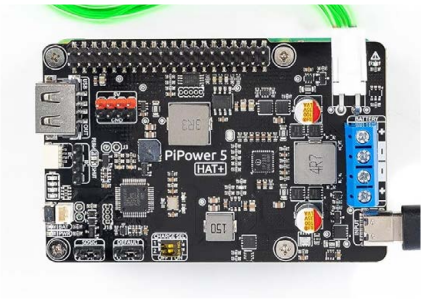
DIMENSIONS:
85 × 56 × 65mm (stack height)

POWER
7.4V, 2000mAh two-cell 18650 Li-ion battery pack; input: 5–15V, 45W via USB-C or screw terminals; output: 5V/5A via GPIO, USB-A, and 2× 4P 2.54mm pin header

FEATURES
32-bit Arm Cortex-M23 microcontroller, battery status LEDs, power button, output power indicator, reverse battery warning LEDs, 2× I2C interfaces (four-pin header and Qwiic/ STEMMA QT)



▶ The battery pack sits on a base plate below Raspberry Pi with the PiPower 5 HAT on top



▼ The HAT is packed with features, including extra power inputs/ outputs, status LEDs, and a couple of I2C headers

Dealing with unscheduled power outages has always been an issue for computer users. Not only is it inconvenient when you're working on something, it may cause data loss and the corruption of system files. If you're using your Raspberry Pi as a server, NAS, or any other project that needs to stay up and running, an uninterruptible power supply (UPS) is essential, instantly switching the power supply to a battery when needed.

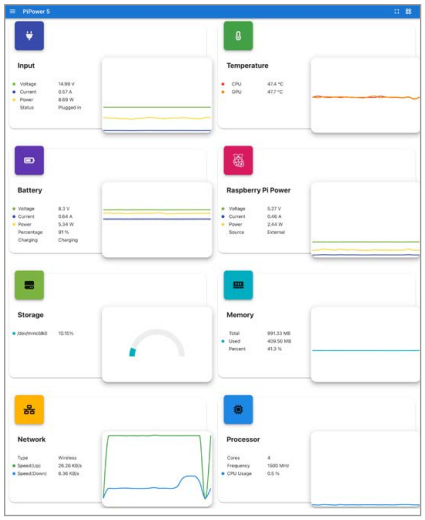
We've seen a few UPS solutions for Raspberry Pi before, but perhaps none quite as sophisticated and versatile as SunFounder's new PiPower 5. Building on previous iterations, its two-cell rechargeable battery pack can now supply enough current (5A at 5V) to power a Raspberry Pi 5, along with most other 40-pin models (except the keyboard computers). Thanks to an on-board microcontroller chip, it also boasts some smart power management features and can even send you email notifications for specific events.

Stacking up

Upon unboxing the product, you'll find the PiPower 5 HAT along with the battery pack, a few other bits, and assembly instructions. The battery pack sits on an acrylic base plate, stuck to it with an adhesive strip. You then attach four stand-offs to secure Raspberry Pi 5 (or other standard-size model) above – if using a Raspberry Pi Zero 2 W, there are slightly longer stand-offs for the empty side. You then add four more stand-offs for the HAT, which connects to Raspberry Pi via an extended female header and has a pass-through header to extend all the GPIO pins – the HAT only uses the I2C pins and GPIO 26. Finally, plug the cable from the battery pack into a port on the HAT.

Before using it for the first time, you're advised to do a full battery charge to ensure optimal performance. Just connect a standard Raspberry Pi 5 power supply to the USB-C port on the HAT. During charging, four LEDs blink to show the charge level of the battery; they also act as status indicators for certain conditions, such as the two middle ones flashing while waiting for a system shutdown signal. Once the battery is fully charged, it should have enough juice to keep a Raspberry Pi 5 powered for up to around five hours, depending on the workload.

- You can even use the HAT with any of SunFounder's Pironman enclosures, although it sits on the exterior
- A web dashboard shows all the vital system stats, along with historical data, and gives access to settings



Power tools

To make the most of the setup, you should install the special PiPower 5 Tool by cloning the GitHub repository and running an installation script. You can then access a web dashboard from any device on the network by navigating to your Raspberry Pi's IP address with the port number 34001. Here you can view all sorts of detailed system info, such as battery level, power drain, CPU/GPU temperature, storage, memory, network, and processor usage. You can also set the 'shutdown percentage': the battery level at which the system will automatically shut down.

> *Notifications can be set via the web dashboard or terminal and may result in a buzzer alert and/or an email being sent*

Usefully, the PiPower 5 supports event notifications for conditions such as when battery power is activated or the battery level is low. Notifications can be set via the web dashboard or terminal and may result in a buzzer alert and/or an email being sent. For the latter, you'll need an email provider that uses an SMTP server; for Gmail, you just need to generate an App Password.

A nice bonus is that you can interact with the PiPower 5 in your Python scripts, using SunFounder's SPC library that's pre-installed in the virtual environment, to read data and handle shutdowns.

Verdict

A versatile UPS solution that works well and has some neat bonus features.

9/10

8BitDo 64 Controller

Taking it back to 1996, with a very modern and welcome twist. By the very millennial **Rob Zwetsloot**

8BitDo **rpimag.co/8bitdo64** £35/$44

SPECS

DIMENSIONS:
147 × 103 × 61.4mm

WEIGHT
344g

INPUTS
10× face buttons, Hall effect analogue stick with digital click, four-way directional pad, 4× shoulder buttons

▲ The controller is familiar yet new thanks to its blending of the original style with modern layout. The stick even has an 'L3' push-in button function

Let's just put this to rest: the N64 controller is not the worst controller of all time. That being said, human input technology for video games has vastly improved since 1996, with form factor perfection being obtained in 2005 with the Xbox 360. Yes, 21 years ago. Once you dust your ageing bones up off the floor, we'd love to tell you about 8BitDo's 64 Controller, a lovingly crafted modern interpretation of the controller that popularised analogue control schemes.

Housed inside the 8BitDo Ultimate form factor (derived from that 360 perfection), the two-prong controller rearranges the N64's complement of buttons to a more standard form employed by even Nintendo themselves in their Pro Controller designs. The once separate D-pad sits below and to the right of the analogue stick, and there's nary a second analogue in sight in favour of its digital ancestor: the C-buttons. There are now a pair of Z buttons, taking up the trigger slots of today, and some extra hot keys are included – great for using emulation functions not required by original pads, and kept accessible yet nicely out of the way so you don't accidentally press them.

Controlling stick

The weight and form factor make it a very comfortable controller to hold. While N64 games usually weren't designed for you to be able to use the D-pad and the stick at the same time, the layout of this controller makes it no problem to access them both if needed – useful for menu navigation or even just using it for non-N64 games.

▼ The shoulder buttons are digital, and you may need to play with the configuration depending on how it feels to use one shoulder and one Z-button/trigger during play

▲ Stark white and black alternatives are available, but we do like the nostalgic Classic Grey version

It still does feel very familiar, though. Great lengths have been made to replicate the look of the buttons on an N64 controller, with that offset A button very well known from the thumb imprints it caused after particularly intensive *Mario Kart 64* sessions.

The stick has the correct, and very firm, ridges of the original. The original controller's analogue had a lot of throw in it: due to the length of the stick, it did move quite far from neutral to the edge. We had some slight concerns that the Hall effect, lower-profile joystick might rob some of that nostalgic feel, but either 25 years of lower-profile sticks or some great tweaking to its resistance meant it had absolutely no effect on the games we played. And the Hall effect stick won't have the same drifting or breaking issues as the original, either. It even has the little octagonal impression around the edge, perfect for specific *Super Mario 64* speed run setups.

The only thing we had a slight issue with was the raised C-buttons. Initially our thumbs were fumbling with them; however, by the end of our *Lylat Wars* testing (three medals and the good ending), we were looping and U-turning with ease.

> *Great lengths have been made to replicate the look of the buttons on an N64 controller*

Hook it up

The controller connects via Bluetooth and cable – not only to Raspberry Pi, but computers and consoles too. Connecting to Raspberry Pi is extremely easy via dedicated emulator operating system installs such as Recalbox, although it took us some tweaking to understand how to configure each button for optimal play. The version we reviewed was the Classic Grey, but alternate versions with a more modern, sleek colour scheme are available (and a few bob cheaper too).

After much laborious testing, we came away forgetting the purpose of why we were playing in the first place. Making the tricky *Super Mario 64* jumps and barrel-rolling our way through *Lylat Wars* felt no different, albeit with slightly more comfort than in the last 30 years of playing them. Honestly, that's probably the highest praise it can get.

Verdict

A near perfect replica of the original N64 controller, but laid out for modern sensibilities and comfort. A must-buy for anyone misty-eyed over *GoldenEye 007*.

10/10

Makers of out-of-this-world products are embracing Powered by Pi. **Rosie Hattersley** hails newcomers to the accreditation scheme

Watching four astronauts blast off into orbit on an exploratory mission to the far side of the Moon has entranced millions worldwide while pushing the boundaries of our knowledge of Earth's nearest celestial body. Such endeavours also serve to inspire existing and future scientists and explorers, and it's a safe bet that interest in astronomy and space technology will increase as a direct result.

Raspberry Pi has been on a similar mission for more than a decade, persuading British astronaut Tim Peake to use an Astro Pi – a Raspberry Pi, Camera Module (later, an HQ Camera), and Sense HAT combo – on the ISS, where its sensors could provide real reportage on temperature, velocity, air quality, and more, and send data back to us. The Raspberry Pi Foundation continues its partnership with the ESA (European Space Agency), sending our single-board computers into orbit on several subsequent missions, as well as partnering on the Astro Pi (**astro-pi.org**) coding challenge. Meanwhile, Raspberry Pi has also been used in astrophysics labs and by university projects to develop CubeSats that orbit Earth, providing communications, weather, and other critical data. In fact, there's a whole strand of products that use Raspberry Pi and are certified for use in space – see **rpimag.co/raspberrypispace**.

While the rigours involved in such accreditation are necessarily more extensive than those for our own quality assurance scheme, it's certainly well worthwhile consulting the ever-expanding roster of Powered by Pi products when researching your own projects or kitting out your smart home. The Powered by Pi logo denotes a product accredited by Raspberry Pi and one that has gone through rigorous testing, with nearly 400 products now bearing the scheme's badge. The database is organised by product type, with new entrants such as DynamicDeepSky's AR camera, Aqua Libra's smart drinks dispenser, and Framework's modular laptop showing the breadth of entries.

Find detailed information about the compliance regulations and testing procedures for every Raspberry Pi product at **pip.raspberrypi.com**.

APPLY TO POWERED BY RASPBERRY PI

Our Powered by Raspberry Pi logo shows your customers that your product is powered by our high-quality Raspberry Pi computers and microcontrollers. All Powered by Raspberry Pi products are eligible to appear in our online gallery.
rpimag.co/poweredbypiapply

This Is Not Rocket Science Bopp & Steve

Netherlands | rpimag.co/boppsteve

With NASA's recent Moon mission enthralling so many people, it seemed churlish of us not to draw your attention to this range of music-making modules called This Is Not Rocket Science. The latest is the Bopp & Steve, a deconstructed reverb unit with added effects. The entertaining blog explains its origins in creators Stijn and Priscilla's habit of doodling little planets on the packaging for their earlier Fenix IV modular synth. Once the planetary entities had been given names, the duo began pulling in phat sounds from existing synths, including a reverb that Stijn had been working on for 20 years and which they soon named 'space mode'. Additional diffusions, delays, and 'tails' were added to the effects options. After some great feedback on the prototype at audio shows, production based on Raspberry Pi RP2350 commenced, with Bopp & Steve making its debut at the start of 2025.

XOGO Mini Max

USA | xogo.io

XOGO is one of those companies that understands the importance of focusing on a specific task and providing a solid solution. The seemingly simple puck-shaped controller fits in the palm of an adult hand and connects to a screen via HDMI. Once its remote control is paired over Wi-Fi or Ethernet, you can serve up digital messaging and content from your XOGO account. Clips and media stored in your digital signage account can be accessed and shared to dozens of screens, even called up via a XOGO Player phone app. XOGO also has a dedicated version for Raspberry Pi, so a Raspberry Pi 3, 4, or 5 can be used as a XOGO hub. Full instructions can be found at **rpimag.co/xogosetup**.

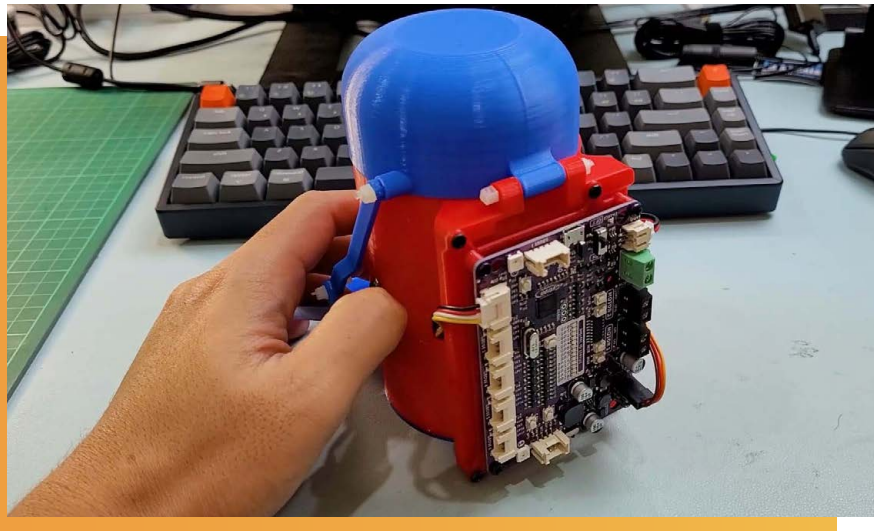
Aqua Libra

UK | rpimag.co/aquahospitality

Smart drinks and flavoured waters are a familiar sight on supermarket shelves and in office chiller cabinets. Canned versions are convenient, but leave something to be desired in environmental terms. In places where guests or staff can just as easily fill up a reusable glass or flask, an Aqua Libra tap offers a greener option. A smart dispenser serves up subtle fruit tastes, with Raspberry Pi Pico controlling the Flavour Tap's pump and solenoids to ensure a perfect mix of peach and raspberry, orange and yuzu, or simply still or sparkling water.

Cytron Robo Pico

Malaysia | rpimag.co/robopico



Cytron's dedicated board for Raspberry Pi Pico 2 or Pico W features four servo motor ports, seven Grove I/O connectors, and a dual-channel DC motor driver. The Malaysian firm describes Robo Pico as "a canvas for your creative masterpieces". By way of example, it has provided several fun projects, including one to wirelessly control a robot vehicle. The web page also features a simple but effective Hungry Robot build that encourages you to toss scrap paper into a desktop bin for easy recycling. The setup involves a 3D-printed bin with a sensor and servo that lifts the lid, ready to receive a deftly thrown ball of paper.

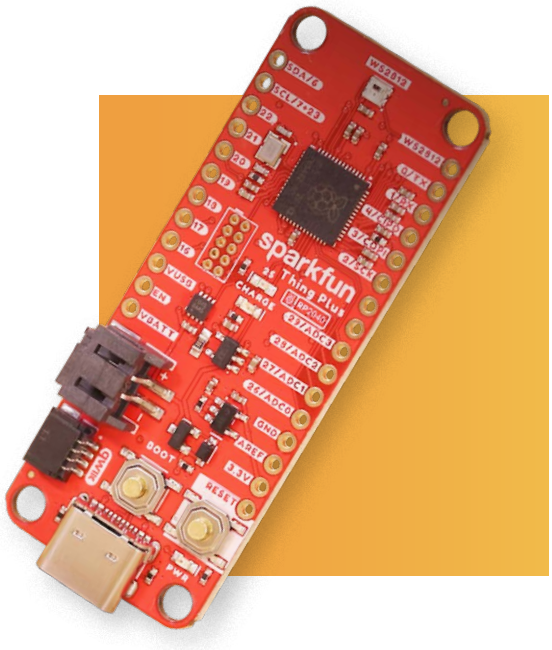
Dynamic Deep Sky Astroid

Australia | rpimag.co/dynamicastro

With all the excitement about NASA's recent Moon mission, and our own recent astrophotography coverage (rpimag.co/162), we were delighted to see Australia's Dynamic Deep Sky pop up with an innovative product that aims to make sense of celestial sights. The firm has developed both augmented reality and AI products for astronomers and astronomy-focused manufacturers. Its Astroid software platform offers an interactive way of exploring the stars that can be installed on OEM astronomy products, for stacking and deep-sky tracking, or in a dedicated AR camera. The Astroid OS can also run natively on Raspberry Pi, making it a great option if you want to build a Raspberry Pi-based astronomy rig.

SparkFun Thing Plus RP2040

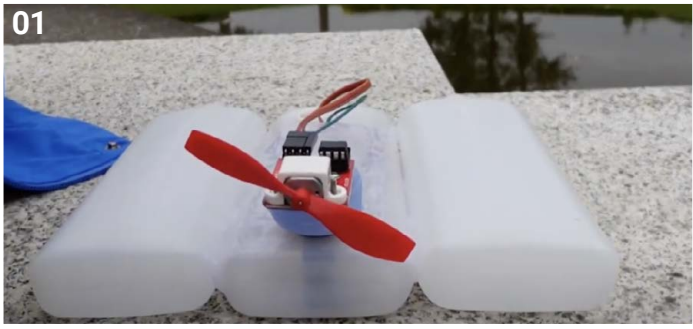
USA | rpimag.co/thingplus



SparkFun has an extensive inventory of products that work with Raspberry Pi, including its own Qwiic range of accessories that make it easy to prototype using the I2C protocol. The Thing Plus RP2040 development board is another great option, offering a plug-and-play Qwiic connector for its range of sensors, actuators, and displays. Based around the Raspberry Pi RP2040 dual-core microcontroller (as used on Pico), it has the convenience of USB-C power, offers 30 I/O pins (including 18 multifunctional GPIOs), addressable LEDs, additional RAM, a microSD card slot and, of course, PIO (RP2040's Programmable Input/Output). The board also has a two-pin JST connector so it can connect to a LiPo battery for portable use. There's now an RP2350 version of Thing Plus too.

10 amazing: waterborne projects

Raspberry Pi can safely get wet, as a treat



Electronics and water don't mix – that's one of the core rules of making. However, our planet's surface is mostly covered in water, which means some intrepid makers have decided to bend that rule and put Raspberry Pi on and beneath water.

01. Roboat

Pico-powered boat

rpimag.co/roboat

Using three empty contact lens solution bottles, a Raspberry Pi Pico, and many prayers, this little robot boat zooms across water with the help of a smartphone app.

02. Nemo-Pi

Coral weather reports

rpimag.co/nemopi

One of the major threats to coral reefs is climate change, and Nemo-Pi is able to monitor the local 'weather' of the ocean surrounding the coral reefs while underwater.

03. Antarctic Picam

Ice-cold photography

rpimag.co/icepicam

Inspired by a project in an old issue of The MagPi, this simple watertight PVC tube built by a young maker has been used to study algae living on ice in the Antarctic.

04. BlueROV2

Serious exploration

rpimag.co/bluerov2

From £3,649/$4,900

This professional-grade ROV is very affordable for what it can do, and it can do a lot – including going down as deep as 300 metres, which is actually a big deal.

05. Submersible drone

Diving tube

rpimag.co/subpi

The term for an underwater drone is ROUV (Remotely Operated Underwater Vehicle), and this basic yet sturdy build is completely home-made.

06. LEGO submarine

Bricks don't sink?

rpimag.co/legosub

This LEGO submarine is a fairly simple submersible using a Raspberry Pi Zero 2 W and a syringe as a ballast tank. The LEGO provides the structure of the interior components, as well as a little seat for the LEGO Minifigure captain.

07. SailBot

Automated Atlantic navigation

rpimag.co/sailbot

Tired of constantly winning robotic sailing regattas, students at the University of British Columbia decided to create Ada, an autonomous sailing boat that sailed from America to Ireland.

08. Maka Niu

Deep-sea photography

rpimag.co/makaniu

The deeper you go in the ocean, the harder it is to keep tech going. Maka Niu can descend to at least 1500 metres, with a theoretical maximum of 6000 metres. James Cameron can only dream.

09. Raftberry

Portable dock

rpimag.co/raftberry

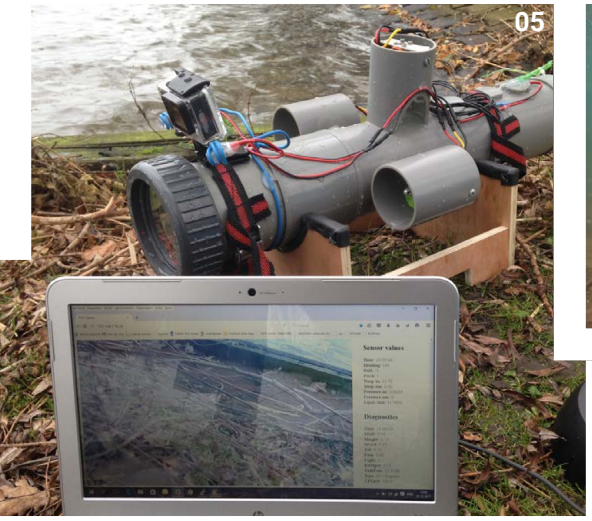
If you're lucky enough to be able to take friends out on a lake, why not create a powered dock they can just chill on, eat lunch, and dip their toes in the water? All powered by a Raspberry Pi and accessories.

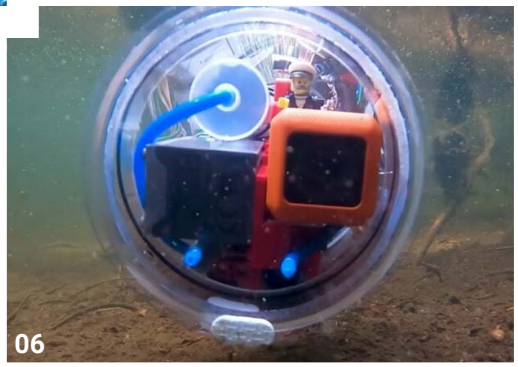
10. Yacht navigation

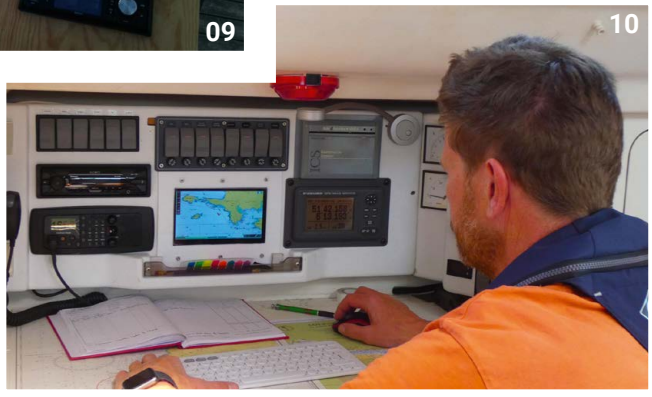
Rebuilding from scratch

rpimag.co/yachtnav

Buying a 30-year-old boat can apparently come with the issue of severely worn and damaged electrical systems. Nothing a Raspberry Pi and a bit of know-how can't fix.







Warning!

Water and electricity

Water and electricity do not mix, so be extra careful with waterproofing and handling such a project if it has just been removed from water.

rpimag.co/electricalsafety

V Hunter Adams

An ever-curious lecturer helps his students embrace their creativity with RP2040 and RP2350

Name V Hunter Adams
Occupation University lecturer
Community role Maker
URL vanhunteradams.com

Hunter Adams tells us he has always been an "unfocused" person... "Rather than developing an obsession with a particular specialty, I have a tendency to accumulate and maintain my interests."

Hunter is a professor at Cornell University, teaching a class on microcontrollers that features Raspberry Pi RP2040 and RP2350. His students come up with amazing builds that use them, including robot dogs, Street Fighter clones, and an interactive lightsaber.

"To some extent, I used school to explore those interests," Hunter says. "I studied physics in college, and then went to more school to learn spacecraft engineering, did a brief post-doctoral appointment in Cornell's astronomy department, and I now teach in the electrical and computer engineering department."

When did you learn about Raspberry Pi?

I learned about Raspberry Pi in college! Probably in 2011 or 2012. With no previous experience in making, Raspberry Pi excited me because it offered a lower activation energy to productivity than more industry-oriented alternatives. The datasheets are spectacular, and the getting started guides are second to none.

As I've gained a bit more experience in embedded systems engineering, I've

▶ Hopefully it can read higher than 3.6 roentgens

▶ It's like a pianola, only slightly more shrill

remained deeply enthusiastic about Raspberry Pi, but the source of my enthusiasm has evolved. I still love the datasheets and user guides, but I now also deeply appreciate the open-source philosophy that Raspberry Pi employs for so many of its products. I love that I can go read the boot ROM of the RP2040 to learn more about how the process works, for instance!

How did you get into teaching?

If you really like learning things, you've probably figured out the learning mechanisms that work best for you. If you're interested in learning some new math, for instance, you might consider watching a video about it. That works OK for me, but I find that I learn that topic more deeply if I instead (or additionally) read a good book about it. If I want to learn that content even better, I'll work some problems independently. And if I want an even deeper understanding than that, I'll generate written documents that explain my solution.

I got into teaching because I found that, for me, it sits on the very top of this learning hierarchy.

If I really, really want to learn something I will go through all of activities above, and then I'll prepare and deliver a lecture on the topic. For me, I find that this last step of teaching highlights gaps in my understanding that had previously been hidden.

What kinds of things do you teach in the microcontrollers class?

In the microcontrollers class, we use engineering as the mechanism by which we learn about interesting topics separate from engineering. One of those topics is birdsongs!

The first laboratory assignment for the class is to build a real-time birdsong synthesizer. In particular, the students are tasked with synthesizing the song of the Northern Cardinal. I chose this particular bird for a couple reasons. The first is that it's [a local bird], so students will hear it singing as they walk across campus. And the second is that it's a songbird! Songbirds tend to 'whistle'. They generate a single tone, and they modulate that tone in complicated and beautiful ways… The students are finished with this lab when they build a synthesizer that is good enough to trick Merlin, the app developed by Cornell's Lab of Ornithology that identifies birds by their song. After having completed the lab, a fascinating thing happens for almost every student. They hear birds singing! Of course they had always heard birds singing, but their brains had long-ago stopped noticing those songs.

> *I love that I can go read the boot ROM of the RP2040 to learn more about how the process works*

One of the amazing consequences of doing a project to explore an interdisciplinary interest is that it removes a filter from your consciousness.

What are some of your favourite things that students have built?

Pico Pasture Cow Herding Simulation: These two students, Kirti and Francesca, really liked cows. To explore that interest, they found a paper that provided a mathematical model that describes cow herding behaviour. They implemented this algorithm on a Raspberry Pi Pico, hand-crafted bitmapped pixel art for cows, barns, trees, etc., and implemented a video game that allows for the user to interact with a herd of cows.

Nuclear Fission Simulator: Andrew and Tyler were interested in learning how nuclear reactors work. To that end, they implemented a simple model of a nuclear reactor on the Raspberry Pi Pico, and built themselves a control panel that allowed them to adjust parameters and control rods. This allowed for them to safely 'experiment' with controlling a nuclear reactor.

The Self-Playing Xylophone: These students, Harris, Karina, and Zoe, were amused by the idea of getting a simple kids toy to do something complicated. So, they interfaced eight servo motors with a Raspberry Pi Pico and used them to actuate mallets to play a children's xylophone. The user could select among a wide variety of songs, including (of course) the Cornell Alma Mater and *Never Gonna Give You Up* by Rick Astley.

Maker 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.

Follow along to #MakerMonday each week over on our various social media platforms!

01. We may have to follow this video ourselves for our home server!

02. Make their friend's day: glance at the README

03. Affordable and able to be done *right now*? Sign us up

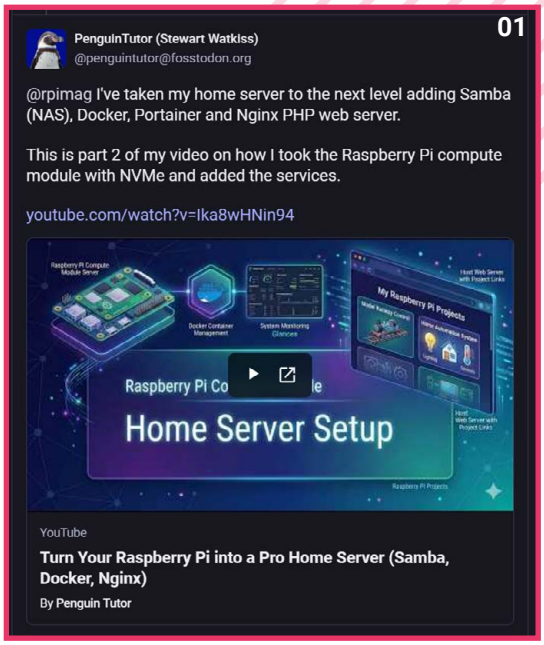
04. Testing the networking limits of a Raspberry Pi 5 with Akkie

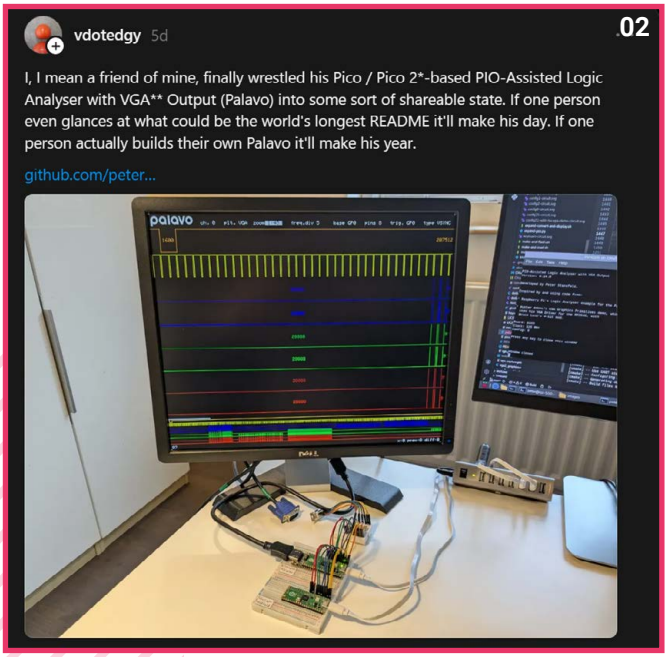
05. Raspberry Pi 1 Model Bs are still excellent pieces of kit even today

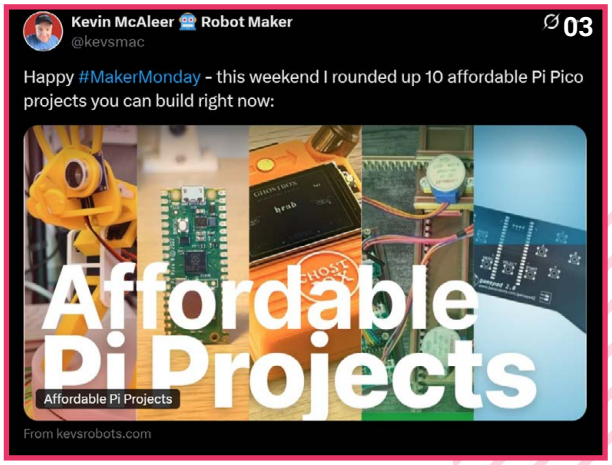
06. We're back in the model train mines with Pater Practicus

07. Theatre techs everywhere take note

08. The purest form of vibe coding







04

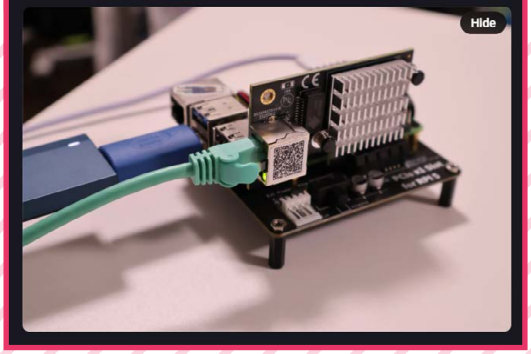
あっきぃ
@akkiesoft@h.kokuda.org

@rpimag Hello!!

Last week, I tried running a Realtek RTL8127-based 10GbE NIC on the Pi 5 running Linux 6.18!

Although standard drivers became available in Linux 6.18, it was necessary to compile the source code for the vendor-provided drivers in order to achieve optimal performance.

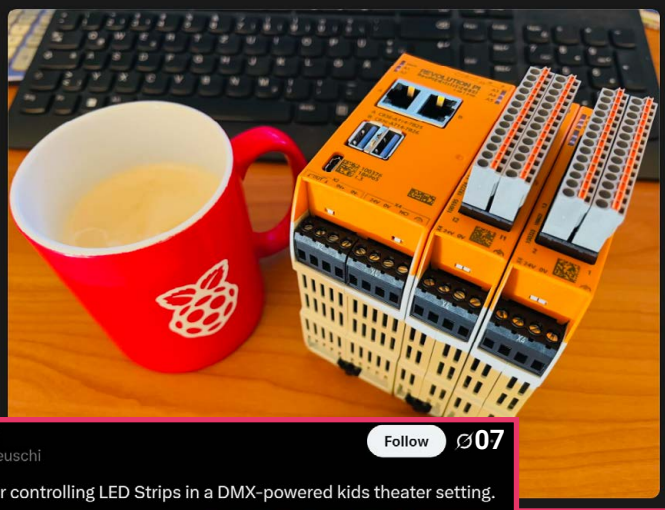
When I ran iperf3 with PCIe 2.0 enabled, it recorded a speed of 3.6 Gbps. However, I enabled PCIe 3.0, the operating system failed to boot...



05

r_schulz_maker 23/03/2026

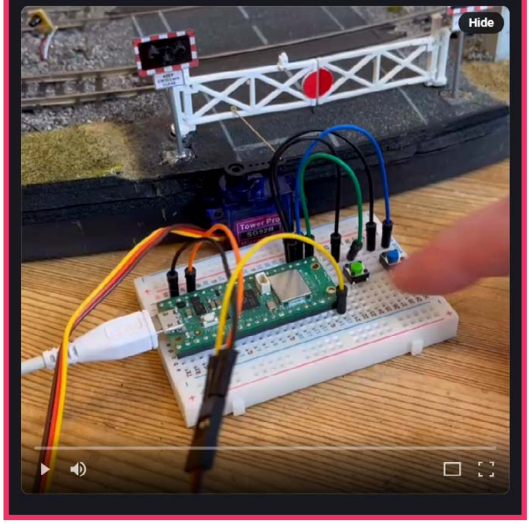
Hi @rpimag , happy #MakerMonday. I bought my first @raspberrypi model B about 13 years ago. After countless #maker projects, I will now experience new adventures to integrate #RaspberryPi into my professional life for new tasks like #RevolutionPi for PLC applications .



06

Pater Practicus
@paterpracticus@mastodon.world

@rpimag First efforts on servo powered crossing gates, with a second Raspberry Pi Pico. Eventually buttons will be replaced by the signals from the break-beam sensors that control the lights (youtu.be/fLOKcf58j8k?si=3UDjf3...). #MakerMonday

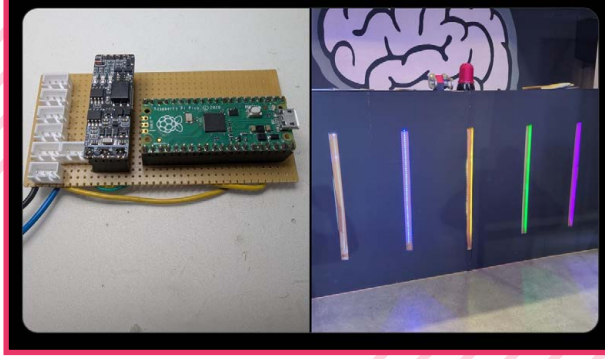


07

RN
@neuschi

Follow

Used it for controlling LED Strips in a DMX-powered kids theater setting.



08

thecarolinedunn › OpenClaw 01/04/2026

I made a timelapse Raspberry Pi without coding using #openclaw - youtu.be/bl7Ec...

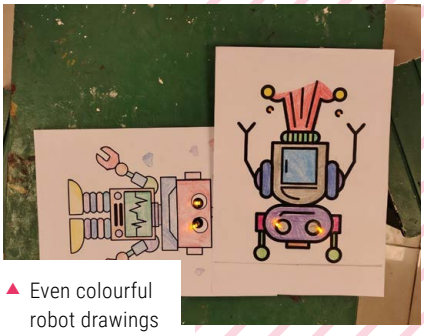
Raspberry Jam 2026 – Santiago del Estero

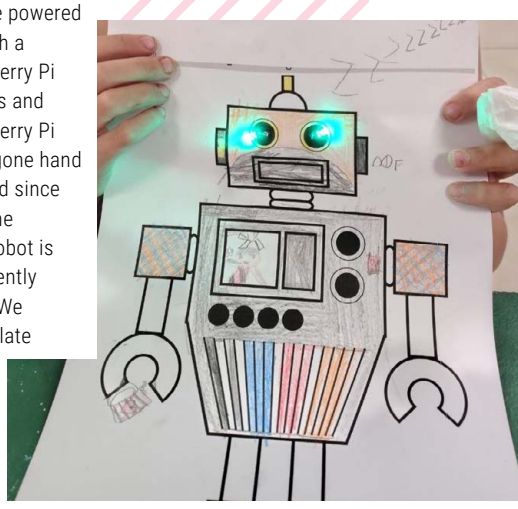
Community members around the world celebrate the birthday of Raspberry Pi

Over email, Eliana Otrera tells us, "It's a pleasure to share our experience with you on 'Aniversario Pi Day'." She runs a Raspberry Jam in Northern Argentina.

"This time, we worked with children between the ages of five and ten," she says, "introducing them to the world of programming and robotics in a fun and educational way. During the activity, we provided basic concepts adapted to their age, fostering curiosity, creativity, and logical thinking. We also introduced them to the Raspberry Pi as an example of a different technological device, explaining its features and potential uses, with the aim of expanding their knowledge of the various tools available in the field of technology."

We loved the robot theme of the projects in the photos, and we hope you will too.





▲ Even colourful robot drawings can be powered up with a Raspberry Pi

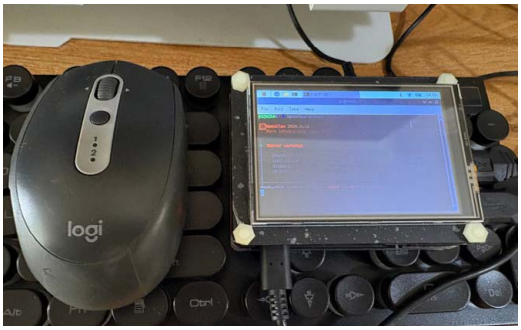
◀ Robots and Raspberry Pi have gone hand in hand since day one

▶ This robot is apparently tired. We can relate

Crowdfund this

Crowdfunding campaigns to keep an eye on

ClawCore One



Run a private AI assistant directly from Raspberry Pi with this plug-and-play solution that gets to work in no time. The team reckons it will help to save you time during your day; your mileage may vary, however.

▶ **rpimag.co/clawcoreone**

Touchscreen Data Logger

This soon-to-be-launched Kickstarter is for an industrial-use Raspberry Pi that includes a touchscreen and many serial ports and environmental sensors. The page has a laundry list of other features, but seems well suited for measuring voltages as low as 0.006V.

▶ **rpimag.co/touchscreenlogger**

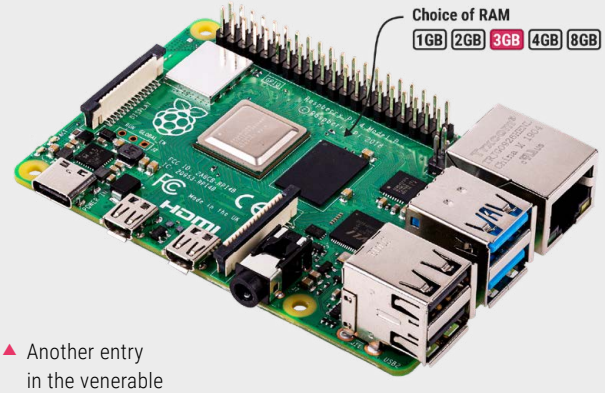
Your Letters

April Foolin'

Doth my eyes deceive me, or is there an actual Raspberry Pi with 3GB of RAM? Being announced on the 1st of April threw me for a bit of a loop, but Pimoroni was more than happy to potentially take my money for it. Why does an odd number of RAM sound so, well, odd?

Piotr via email

Despite the date, it is indeed very real, much like the frustrating circumstances leading to all these measures to try and make the current RAM shortage as bearable as possible. As for the odd number of gigabytes, we always assumed that due to the way binary systems work, it's easier to do stuff in multiples of two. Plus with PCs you're either using one RAM stick or two for dual-channel stuff. 3GB in this case is very real, though.



▲ Another entry in the venerable Raspberry Pi 4 line

Odd time

I was reading the list of weird clocks and the concept of how telling the time is veritably ancient – it made me look up the history of timekeeping and had me wondering about other ways you could create strange clocks with a Raspberry Pi. A sundial with a rotating LED light, a cylinder of LEDs reducing down like a candle, a Pico hourglass, etc. Not sure how an incense clock could be made unless you had plenty of scratch-and-sniff cards available for a robot to rub away. I feel like my mind has expanded!

Karen via email

Sometimes innovation doesn't have to be logical; sometimes it can be a robot hand that moves a minute hand. We love how people can sometimes get truly creative with a project, not doing something for utility but sometimes just because they can.

Anyway, perhaps we should 3D-print a model of Big Ben (or rather, its tower) and record ourselves yelling "BONG!" so it can be played every hour...

◀ This project always reminds us of the robot arm someone programmed to tick the 'I am not a robot' CAPTCHAs

Beige is beautiful

Any chance of getting one of those beige, BBC Micro-style Raspberry Pi 500+s in a shop sometime? The effort of painting sounds like a lot of hassle, and I applaud Mr Roberts for taking the time; however, I don't quite have the patience for that. I do like that you can just buy the keycaps, though. I might actually get some nice red ones for my current keyboard either way.

Ant via email

We too are coveting the very pretty beige 500+. We have a lovely Famicom colours keyboard from 8BitDo, so we're clearly predisposed to having old-looking clacky keyboards. Unfortunately, the beige version is not for sale just yet, and we don't think the 500+ plastic is like the stuff in the 1990s that would yellow with a bit of sun exposure either, so you may need to find your local Warhammer painter and throw them some cans (not literally, mind).

▲ Sometimes, but very rarely, we think some aspects of the past were a bit better. Like this colour scheme

Contact us!

- @rpimagazine
- @rpimag
- rpimag.co/facebook
- magazine@raspberrypi.com
- forums.raspberrypi.com

Community Events Calendar

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

01. Raspberry Jam @ Wikimedia Hackathon Milan 2026

- **Saturday 2 May**
- **Voco Milan-Fiere Hotel, Milan, Italy**
- **rpimag.co/wikihack26**

This isn't your average Jam; it's a dedicated brainstorming and ideation workshop which will bring together two powerful worlds: the versatile Raspberry Pi ecosystem and the open knowledge mission of the Wikimedia movement.

02. CoderDojo Kingston University

- **Saturday 16 May**
- **Kingston University, Kingston, UK**
- **rpimag.co/cdku165**

A free computer coding and digital skill session for youngsters aged between 7 and 17. Activities can include programming with Scratch, Python, or Unity; the Coolest Projects AI Toolkit; robotics with Raspberry Pi RP2040; web page creation (HTML/CSS/JS); and micro:bit digital electronics.

03. Raspberry Pi Jam Mombasa

- **Saturday 23 May**
- **I.O.ME001 Innovation Centre, Mombasa, Kenya**
- **rpimag.co/rpjm165**

Raspberry Pi Jam Mombasa is a one-day event for students, teachers, and enthusiasts of robotics and AI. The Jam will feature workshops, demonstrations, and hands-on activities.

04. Norwich Games Festival

- **Saturday 23 May to Saturday 30 May**
- **The Forum, Norwich, UK**
- **rpimag.co/rpjamwich165**

Have fun with Raspberry Pi computers at the Norwich Games Festival. They will have the ever popular Minecraft, and this time with a coding challenge if you'd like to take it on. You can also have a go at designing your own Lego set with open-source software, or prepare to disappear behind an invisibility cloak!

FULL CALENDAR
Get a full list of upcoming community events here:
rpimag.co/events

05. International Drone Show 2026

Official Raspberry Pi Event

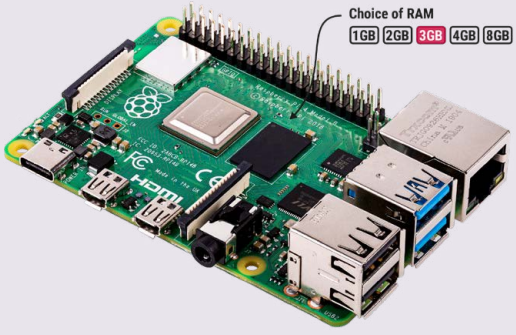
INTERNATIONAL DRONE SHOW

- **Wednesday June 3 to Thursday 4 June**
- **HCA Airport, Odense, Denmark**
- **rpimag.co/ids26**

The Raspberry Pi team is delighted to be exhibiting at the International Drone Show in Odense, Denmark, in June. There, you'll be able to meet the team and see a range of Raspberry Pi technology. You'll see how individuals and businesses in the UK and around the world use Raspberry Pi to support their projects and applications, and discover how Raspberry Pi can help you with your own solutions.

Win 1 of 3 Raspberry Pi 4 with 3GB RAM

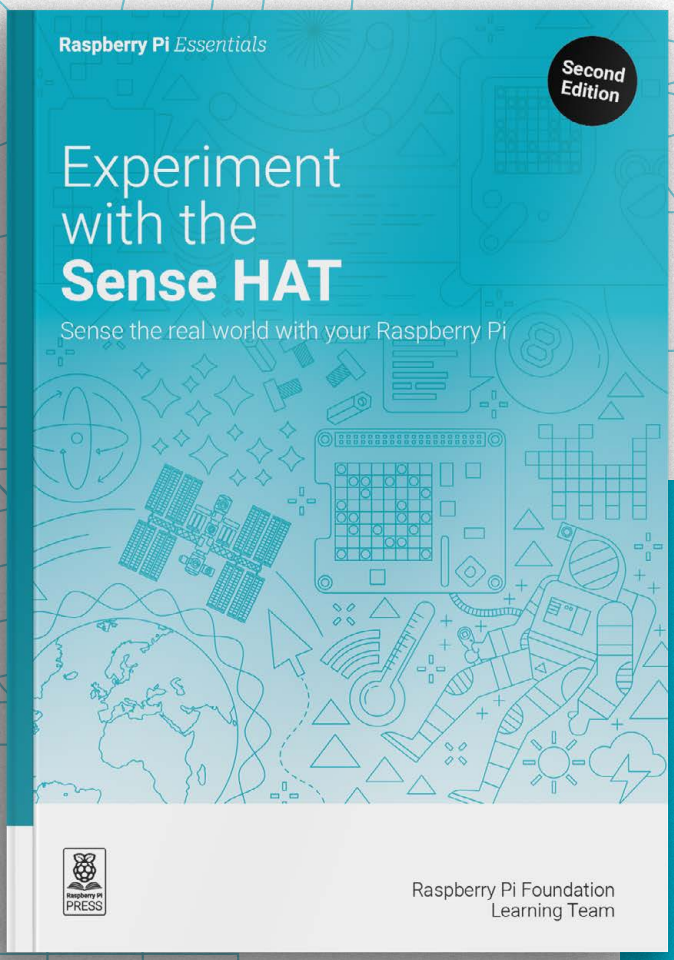
Discover a new variant of the classic Raspberry Pi 4 board with 3GB RAM. It's perfect for maker projects that need low cost and moderate headroom. It's also an interesting product with a specific amount of RAM and is ideal for learning to rightsize your memory. As Eben Upton says: "Providing low-cost general-purpose computing remains a non-negotiable priority for us at Raspberry Pi."



Head here to enter:
rpimag.co/win
Learn more:
rpimag.co/raspberrypi4

Terms & Conditions

Competition opens on **29 April 2026** and closes on **28 May 2026**. 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 Raspberry Pi Official magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter (X) or any other companies used to promote the service.



The Sense HAT is an incredibly versatile and flexible bit of kit with plenty of obvious uses, along with a huge number of less obvious ones, that you'll love to make and share. Updated for the latest Raspberry Pi devices and hardware, this book has everything you need to get started.

- ***Getting started with Sense HAT***
- ***Learn by building:***
  - *A digital twist on the Magic 8 Ball*
  - *Your own interactive pixel pet*
  - *A sparkly light show*
  - *An environmental data logger*
  - *Flappy Astronaut, a low-res, high-fun video game*

BUY ONLINE: **rpimag.co/sensehatbook**

REACH A GLOBAL COMMUNITY

Advertise in **Raspberry Pi Official Magazine**

Our readers are passionate about technology and the Raspberry Pi ecosystem. From DIY enthusiasts to professional engineers.

Your advertisement can sit alongside our cutting-edge tutorials, features, and reviews. And we have flexible advertising options for a range of different businesses.

Take the next step –
advertise with us today!

Email Charlie Milligan at: **charlotte.milligan@raspberrypi.com**

Official Magazine
#166

Audio & Music Special

Make some noise with Raspberry Pi

On sale 28 May

PLUS!

OpenClaw & PicoClaw agents

Raspberry Pi do's & don'ts

Automating model bus movements

Editorial

Editor
Lucy Hattersley
lucy@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Features Editor
Rob Zwetsloot
rob@raspberrypi.com

Sub Editor
Phil King

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

Design

Head of Design
Jack Willis

Designers
Sara Parodi, Natalie Turner

Illustrator
Sam Alder

Brand Manager
Brian O Halloran

Contributors

David Crookes, Tim Danton, Phil Elwell, Ben Everard, Rosemary Hattersley, Jo Hinchliffe, Nicola King, Phil King, Rob Miles, Richard Smedley

Publishing

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications
Helen Lynn

CEO
Eben Upton

Distribution

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

Subscriptions

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

ISSN: 2977-4403 (Print)
ISSN: 2977-4411 (Digital)

The kids get it

It's still program or be programmed in the brave new world. By **Lucy Hattersley**

My parents used to watch the snooker on black and white television. My grandparents saw the arrival of electricity. Every generation encounters advances.

Douglas Adams captured this perfectly with his three rules: "Anything that is in the world when you're born is normal and ordinary and is just a natural part of the way the world works. Anything that's invented between when you're fifteen and thirty-five is new and exciting and revolutionary and you can probably get a career in it. Anything invented after you're thirty-five is against the natural order of things."

I've never known a time without computers. However, I was lucky enough to be born into a world where the internet was being rolled out (even if it was invented slightly earlier). I remember the excitement of surfing the World Wide Web.

Exciting and revolutionary

My niece asked me how an office in the past worked without computers. It's baffling to her. I had to explain that most of the desktop programs were digital recreations of physical things: inboxes on desktops, mailbags, calculators, filing cabinets, typewriters, and so on. The generation being born now will never know a world without generative AI. It's here to stay.

Philip Colligan, CEO of the Raspberry Pi Foundation, has argued comprehensively why kids still need to learn to program computers (**rpimag.co/kidscodeai**). "We need to challenge the false narrative that AI is removing the need for kids to learn to code, and redouble our efforts to ensure that all young people are equipped to take advantage of the opportunities in a world where AI is ubiquitous," says Colligan.

> *The generation being born now will never know a world without generative AI*

The ability to read, write, and understand code is how young people will develop the mental models to understand the systems shaping their world. "The hard cognitive work of reading, modifying, writing, explaining, and testing code is precisely how young people develop a deep understanding of programming and computational thinking."

AI doesn't make learning to code redundant. If anything, it makes it even more essential. A generation can only use AI-created code if it understands it.

Raspberry Pi is right at the heart of this. This is because Raspberry Pi sits at what is called 'the edge'. This is at the edge of industry where computing is processed locally instead of in the cloud.

Raspberry Pi runs at the edge and eases up a lot of those issues. The kids who learn to manage AI systems at the edge will have a huge advantage in a world where AI is just part of the way the world works. Just as hobbyist programmers of the 1980s had a huge boost in life. As Douglas Rushkoff said: "Program, or be programmed."

Previous generations had exciting new technology to play with. And even though AI is challenging, it's also fun to see a new generation with a new toy. Let's hope they build something extraordinary with it.

Lucy Hattersley – Author
Lucy is editor of *Raspberry Pi Official Magazine* and is looking forward to a world of AI-based irrelevance where computers do the work so she can throw a party on the beach. She asked AI and it was pretty frank that she wasn't going to get to do that for quite a while yet.

rpimag.co

HiPi.io

HighPi 5S

The new case from the HiPi team

Reliable case for complex projects:

- Built for Raspberry Pi 5
- Rapid tool-free assembly and disassembly
- Large internal volume for HATs even with the active cooler installed
- Camera cable slot to connect to external cameras
- Power button and external status LED
- Passive & active cooling options
- Secure microSD cover
- VESA mount support

Customization options for volume orders:

- Output ports molded to your exact specs
- Logo printing for a branded finish

Available at these great Pi stores:

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

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

PiKVM

Remote control **redefined**

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

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

PiKVM V4 Plus

The most feature-rich edition

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

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

Available at the main Raspberry Pi resellers

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

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBII · Cytron Marketplace

ROBU.IN Your ideas, Our Parts · ANTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

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

List of official resellers by country: