Train a **rock, paper, scissors AI model**

Build a **working Aliens motion tracker**

Make and **publish your own zine**

Raspberry Pi

Official Magazine
#160 | December 2025

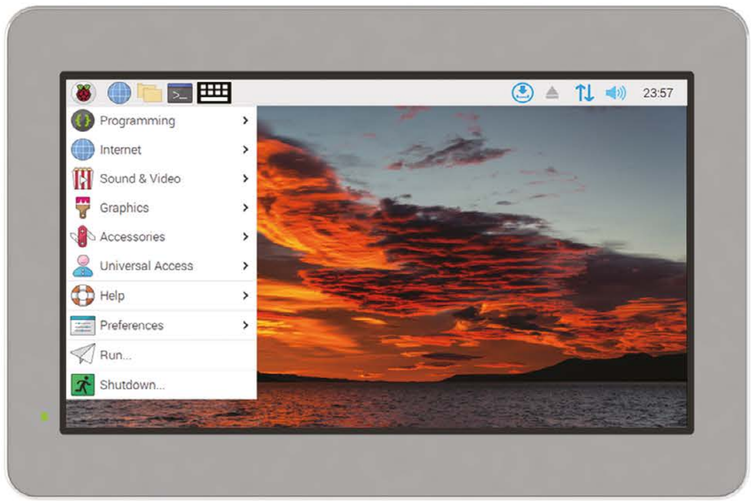
CHRISTMAS GIFT GUIDE

Get the perfect present for makers, hobbyists, and beginners alike

12 £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

© copyright COMFILE Technology, Inc. ALL RIGHTS RESERVED

COMFILE TECHNOLOGY

Welcome to Raspberry Pi Official Magazine

Editor
Lucy Hattersley
This month Lucy has been running, as promised. Thanks to the magic of making magazines way in advance, she's also about to find a big fire for Bonfire Night and watch fireworks go boom! See you in January.

rpimag.co

It's that time of year when the snow starts to settle, the days are short, and the nights are mostly spent putting up gaudy decorations, watching television, playing video games, and getting ready to eat far too much food.

Before that though, there is the annual buying of the presents. We love Christmas here at Raspberry Pi. Rob loves Christmas more than all of us combined, so he and Rosie have put together a massive Christmas shopping and projects feature. Give somebody the gift of a better, and more fun, computer or treat yourself to a new project.

Ben has finished his Christmas lights tutorial so you can make the winter nights look a bit warmer with NeoPixels. There's also a quiz of the year for you to put your Raspberry Pi and tech knowledge to the test. It's a great issue.

Happy Christmas. I hope you celebrate it in style and enjoy whatever treats it brings.

Lucy Hattersley – Editor

Innovation demands selection

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

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

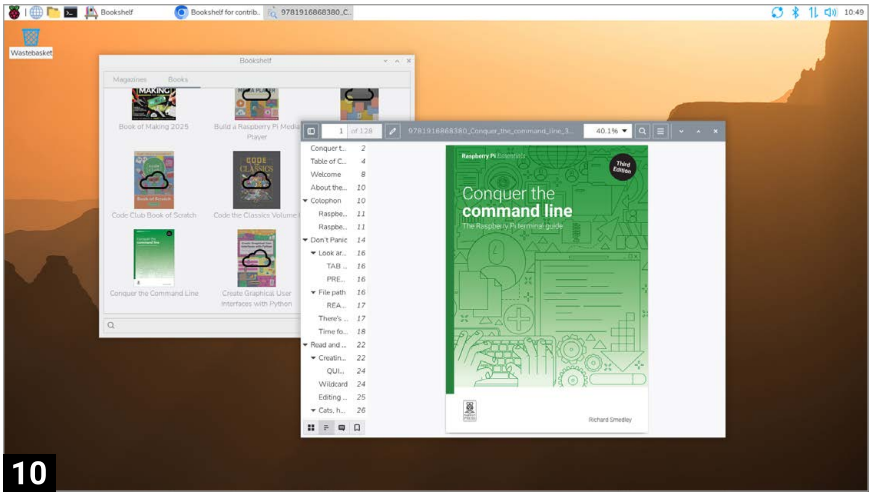
DigiKey

we get technical

DigiKey is a franchised distributor for all supplier partners. New products added daily. DigiKey and DigiKey Electronics are registered trademarks of DigiKey Electronics in the U.S. and other countries. © 2025 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

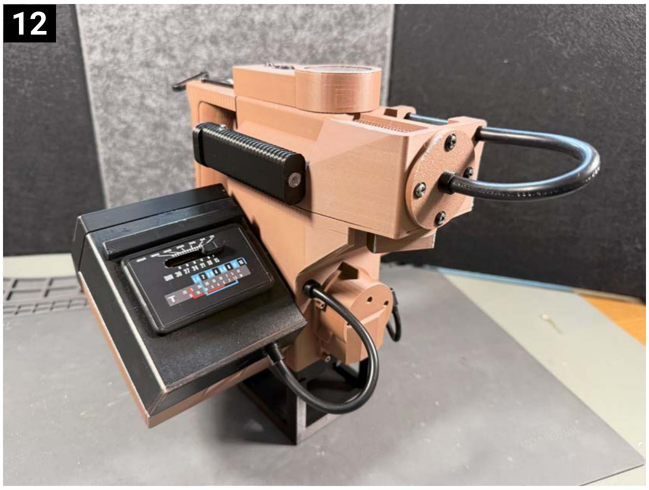
ECIA MEMBER
Supporting The Authorized Channel

Contents



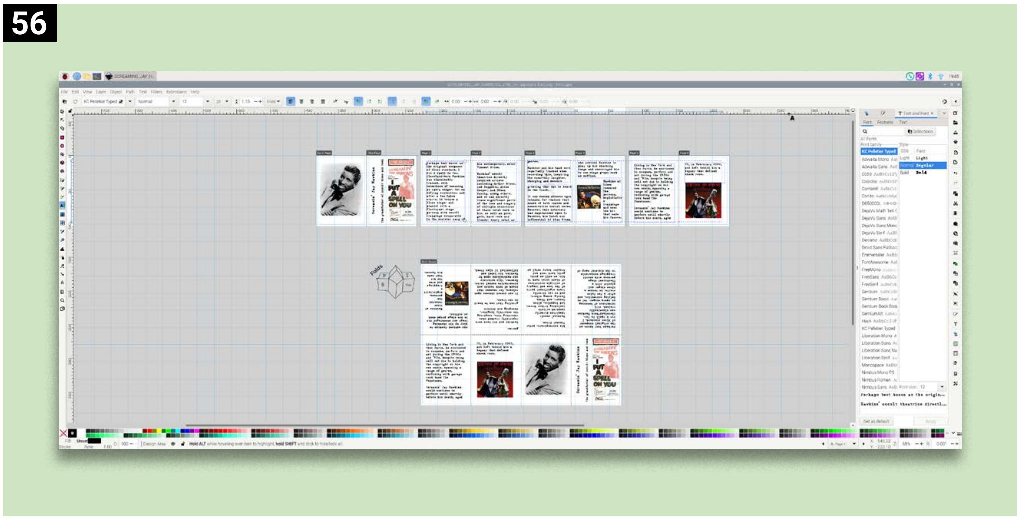
10

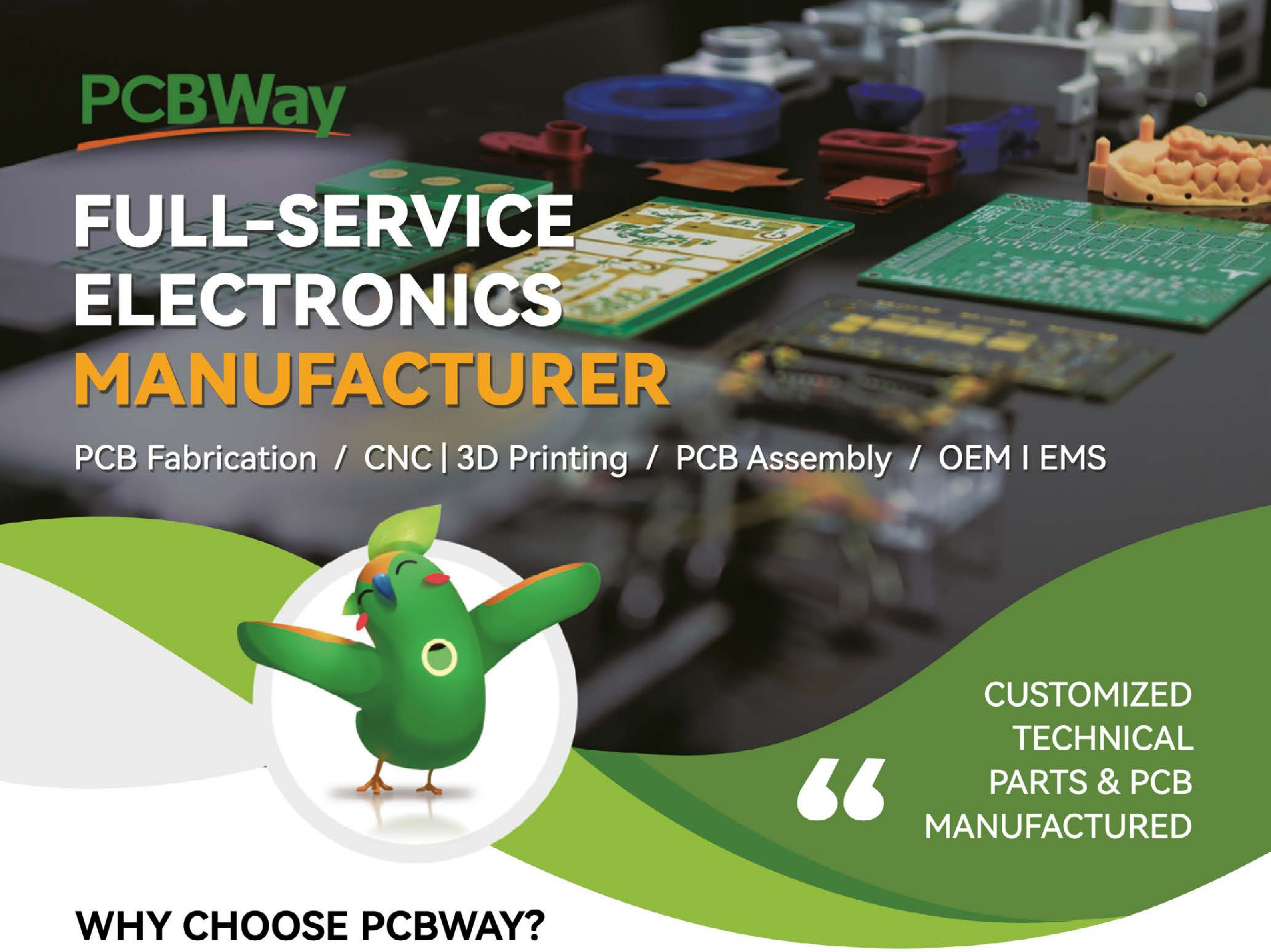
16



12

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. Application to Mail at Periodicals Postage. Prices is pending at Philadelphia, PA and additional mailing offices. POSTMASTER: send address changes to Raspberry Pi, 701 Ashland Ave, Folcroft, PA. 19032.





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

Reviews

Raspberry Pi Community

Competition

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

W6300

The Ethernet Solution for Faster, Smarter Networking!

Need ultra-fast, stable Ethernet? W6300 delivers up to 90 Mbps with powerful IPv4/IPv6 dual stack support!

Key Benefits

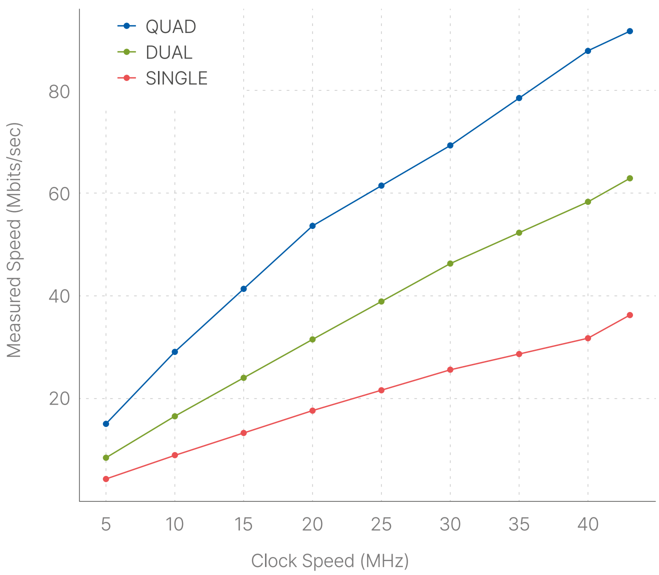
- **High-speed QSPI Interface**
  SINGLE, DUAL, QUAD modes supported
- **8 Independent Sockets**
  Multi-device connectivity made easy
- **Low Power & Wake on LAN**
  Efficient and remote-friendly
- **Seamless IoT & Industrial Integration**
  POS, cameras, automation & more

Where to Use W6300?

- **Smart Home & IoT**
  Network sensors, smart APs, home automation
- **Industrial & Factory Automation**
  POS, network printers, LED displays
- **Security Systems**
  DVRs, CCTV, access control systems
- **Embedded Servers & Cloud Devices**

Performance Test Results

- iPerf3 Performance
  Up to 90 Mbps on RP2040/RP2350 in 43MHz QUAD mode
- Optimized speed across different QSPI modes



WIZ630io 

W6300-EVB-Pico (W6300 + RP2040) 

W6300-EVB-Pico2 (W6300 + RP2350) 

Bookshelf application revamped for Trixie

New Bookshelf app for magazines and books.

By **Simon Long**

At the start of October, we released the Trixie version of Raspberry Pi OS. One of the changes we made was to revamp the Bookshelf application to give contributors access to subscriber-only titles, so we thought we'd talk a bit more about that decision.

Distribution change

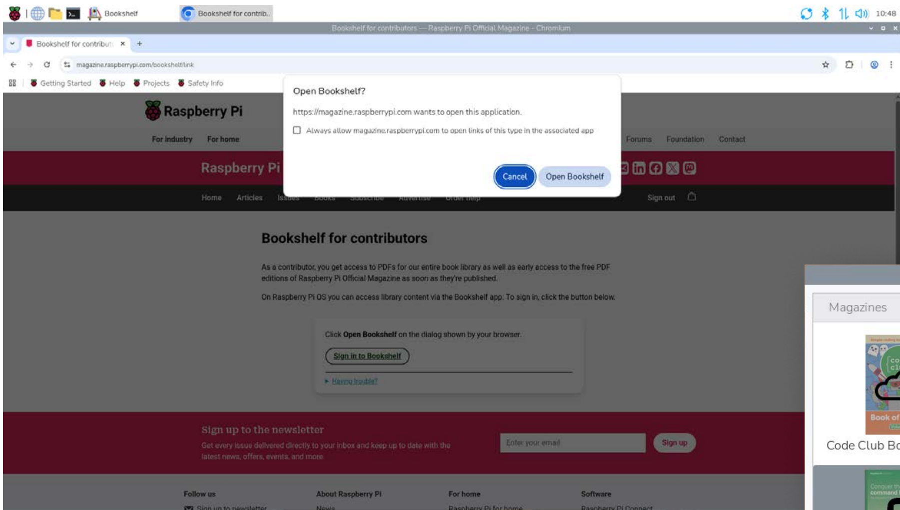
Some time back, we made a change to how our e-books were distributed. Originally, we made all the PDFs of our books available for free on the day they were released, but in 2023 we began expanding the distribution of our print and electronic titles. Before then, our only third-party distribution was through resellers and booksellers in the UK and Ireland. We were entirely absent from e-book channels like Kindle, Google Play Books, and Apple Books, and our print titles were not reliably available from online or retail bookstore channels; when they were, they were typically marked up substantially by third-party sellers.

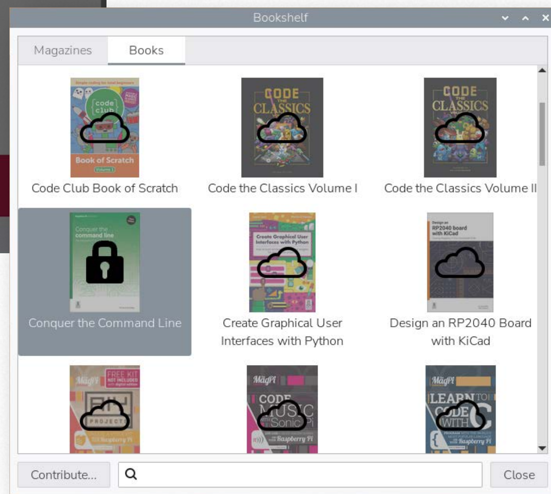
When we started selling e-books, free versions would only become available after a new edition was released. However, we maintained the same Creative Commons licence (CC BY-NC-SA), so in theory, anyone who obtained an e-book could share it and read it on any device they desired. Unfortunately, certain e-book platforms began making it more difficult to extract assets from books you've purchased. To address that problem while avoiding conflict with the channels where we sell e-books, we've instituted two ways you can get DRM-free e-books.

> *We're improving the accessibility of our e-books by gradually making all of our titles free to download in Bookshelf*

We'll get to the second way in a bit, but first, anyone who makes a contribution of £5/€5/$5 a month is eligible to download not only PDFs from our books website (**rpimag.co/books**) but also ePub versions, which are more suitable for e-book readers. Recurring contributors will also get early access to *Raspberry Pi Official Magazine* PDFs, three weeks before they are made freely available to all. If you signed up for early access to

▼ All our books and magazines can be read in the Bookshelf app in Raspberry Pi OS





◄ Sign in to Bookshelf using the same email address as you contribute with to access the books

▼ Magazines and some books are accessible to all readers, with contributor-only books marked with a padlock

the magazine at £3/€3/$3, you'll have access to both book and magazine PDFs at that rate – unless you cancel your recurring contribution.

What does this mean for Bookshelf?

Previously, we only showed free titles in Bookshelf, all of which could be downloaded by anyone. Now we show both free titles – which are still available to everyone – and contributor-only titles, which can only be downloaded by contributors.

When you first start Bookshelf, free titles are indicated with a cloud icon, as before, and can be downloaded and viewed by double-clicking. Contributor-only titles are indicated with a padlock icon, and if you try to double-click one of these, you will be notified that it is only for contributors.

In order to unlock these titles, you will need to set up a contributor account. Click the 'Contribute…' button at the bottom left of the window to open the contributor pages on the Raspberry Pi website.

Contributions are linked to a Raspberry Pi ID, so you will need to set up one of these – if you are already using Raspberry Pi Connect, you can use the same ID, in which case you can simply log in with your existing username and password.

Once you have set up your ID, you will be asked to set up contributions associated with that ID.

After you have associated your contributions with your Raspberry Pi ID, the page will prompt you to 'Sign in to Bookshelf'.

Clicking this link will open Bookshelf and store your ID, identifying you as a contributor. You will now see that the 'Contribute…' button has been removed, and titles which were previously marked with padlocks can now be downloaded.

Even if you stop your contribution, you retain access to the titles that you have already downloaded – the PDFs are stored in the Bookshelf folder of your home directory, and will continue to appear in Bookshelf as titles you can double-click to open. (But you won't be able to download them again without a new contribution, so don't lose them.)

If you move to a new Raspberry Pi or flash a new OS image, you can just log in again with your contributor details and your titles will be unlocked on that Raspberry Pi as well. As long as you remain a contributor, you can access your contributor-only titles on as many Raspberry Pi computers as you want.

The second way we're improving the accessibility of our e-books is by gradually making all of our titles free to download in Bookshelf. Just as the latest issue of the magazine is contributor-only for the first month and thereafter becomes free to all users, contributor-only books will also become free for all Bookshelf users several months after the e-book goes on sale worldwide. For those who download our e-books directly from our website, older editions will become free after a new edition is published, but all e-books will be accessible to those who make a qualifying recurring contribution.

M314 Alien Motion Tracker

Game over, man. Game over. It's the ultimate tracker from *Aliens*. By **David Crookes**

Maker
Rob Smith
Rob is a software engineer, electronics tinkerer, retro gamer, and YouTuber who enjoys building fun things that spark nostalgia.

rpimag.co/alientracker

As fans of the *Alien* movies will no doubt agree, the M314 Motion Detector is a cool piece of kit. Aside from being super-useful – the Colonial Marines would certainly have struggled to detect the presence of aliens without its ultrasound capabilities – the handheld device, with its rugged military looks, was also something of a design masterpiece.

Over the years, many people have replicated the prop, right down to the iconic, suspense-building, beeping sound made as it locks on to movement. Among those enthusiasts is Rob Smith, who first toyed with the idea of making such a device ten years ago. He'd stumbled across the RCWL-0516 Arduino radar module and instantly thought of *Aliens*. "It would just react to nearby movement," he says. And so began a quest to create a tracker that actually works, unlike replicas (and the movie version) that just play back video.

Although the plan was parked for some time, the recent launch of the *Alien: Earth* series on the FX television channel motivated Rob to get cracking.

> *The project had been on my to-do list for a long time and I thought now would be a good time to actually get moving on it*

"The project had been on my to-do list for a long time and I thought now would be a good time to actually get moving on it," he says. As luck would have it, the project would also prove less expensive. "Consumer-based radar modules have developed a lot over the past ten years," he adds. "Some of them were very expensive and are now relatively cheaper and more accessible."

Signal locked

At first, Rob mocked up plans based on an older radar module. He needed it to pan left and right to scan the environment. "I still don't know if that would have worked, because I received an email about a new radar module and the demo on the website made me instantly think it was the right part for me.

The module he ended up using was the DreamHAT+ radar costing £100. "It was specially designed for Raspberry Pi 4 Model B and Raspberry Pi 5 computers," says Rob, who liked the fact it had a 60GHz mm-wave radar with a range of up to 15 metres. The challenge was to make use of the data to generate a realistic picture and sound on a TFT display and speakers that would be integrated into the project. "I was originally tempted to try to incorporate a thermal camera into the design too, but in the end I decided

01. A 2.4-inch Waveshare LCD module is used to display the radar images, which are made to look like those seen in the *Alien* film series

02. A Raspberry Pi 4 Model B is used to process information from the radar to display images and play sound

to keep the prop as close to the original as possible."

Having a Raspberry Pi computer as part of the set-up allowed Rob to achieve his aims. "A lot of off-the-shelf radar modules do the processing for you, so you don't get any control," he explains. "They will tell you the distance usually, but not always the angle. I wanted more control and DreamHAT+ radar gives the raw data from the sensor. The amount of processing required was quite a bit more than I'd expected, so Raspberry Pi made more sense in this case."

It also helped that the radar came with example programs coded in Python and that it was set up to run in the Linux environment. "It was easier to progress along this route," Rob says. From that point, the project began to evolve quickly. "I didn't need many components, although as the project evolved, a few were added," he reveals. These included an accelerometer originally added to move the display as the device is rotated. "I didn't anticipate using it to help reject the user's motion from the radar whilst holding the device."

Pulse rising

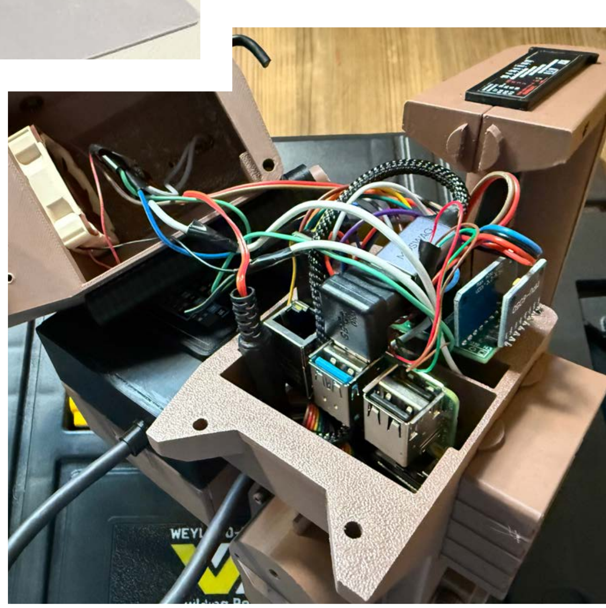
The components needed to fit inside the tracker's case, but this was trickier than first imagined. Rob opted to use 3D-printable files of the M314 Motion Detector available on Thingiverse, but he noted space would be tight. "I hadn't realised how small the tracker actually was and fitting Raspberry Pi, the radar module, and

Quick FACTS

- The Motion Tracker was introduced in the film *Aliens*
- This replica detects motion many metres away
- It uses a Raspberry Pi 60GHz mm-wave radar HAT
- There's audio as well as visual feedback
- It cost around £250 in total to build

◀ The device has been made to look as authentic as possible, although Rob is continuing to make tweaks to both the hardware and software

▼ There's not a lot of room inside the device. Rob says inserting the fan was tricky because wires would catch on it



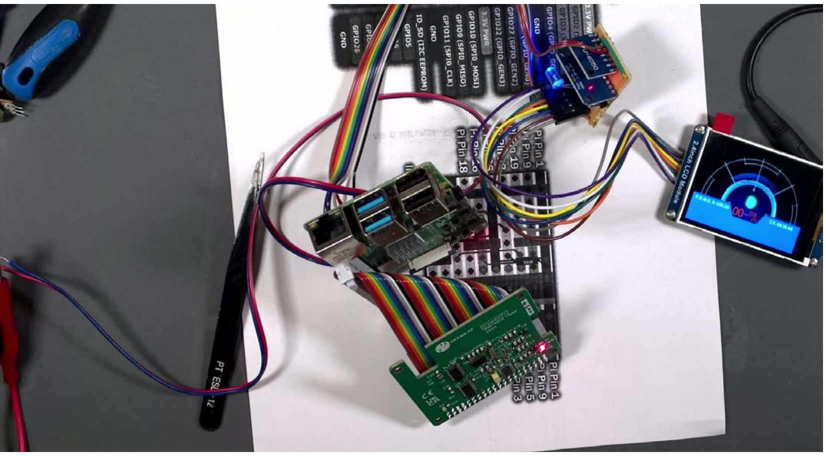
a suitable battery power supply inside was actually quite difficult," he says.

"I also had problems sourcing an LCD screen that was about the right size. Due to the limitations of the case, I couldn't use an HDMI screen either as there simply wasn't room in the case for the cable to plug in, so instead I opted for one controlled via SPI. I also spent many hours in Tinkercad and Blender adjusting the model to fit my requirements and I had to slightly enlarge the viewer part to fit the LCD panel, including its PCB. This meant having to redesign other parts and work out how wires got from the viewer into the main body of the tracker."

Target acquired

It took a fair bit of trial and error to get right, with Rob reprinting the main body and top part about eight times before he was happy with it. But that's not to say the build is complete. Rob has already identified several areas where it could be improved, yet overall it's working well.

The radar is picking up the required data, although the range is less than expected. The detections are also appearing as a series of blue dots on the screen just as they do in the film. Rob worked on making the screen like a CRT display. "The big blue dots are blurred to simulate the softer glow of an old CRT,"



▲ The first full assembly of a project which has been in the planning for ten years!

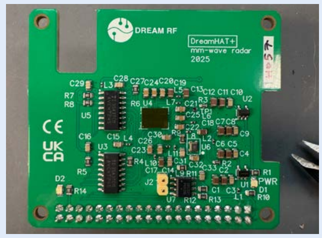
he says. What's more, the sound is faithful with pop noises and a tone recorded from the *Aliens* film. The latter can be synthesised at any frequency.

So what could be done? "I've already experimented with different parameters for the radar," he says. "It doesn't really affect the results it gives, but it reduces power consumption, which is always handy. I'd also like the signal processing – which takes the radar information and works out where the movements are – to be further refined, making it more accurate by reducing false positives."

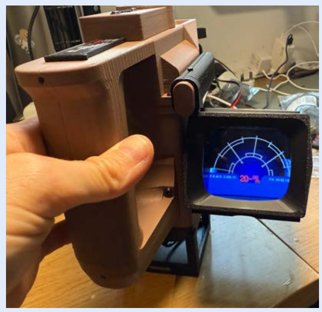
Rob is also pondering whether or not to add a headphone jack and he's considering moving the internal fan to make more space. But even without these additions, it's still an impressive build. "It was the first real Raspberry Pi project I've worked on, so everything was a learning curve," Rob says. "It also took me so long to create, I didn't actually get around to watching *Alien: Earth* until I'd finished writing the build guide."

How It Works

1. The DreamHAT+ radar module can sense movement in two dimensions. Rob Smith uses two of the module's three receivers and contained in that data are reflections of the transmitted signal, at around 60GHz. The distance and angle of moving targets is calculated.



2. Detections are captured several times per second and they are merged together. A program takes into account any hand movements which could affect the signal and the detections are plotted on the screen. An accelerometer rotates the picture on the display as you turn around.



3. Older detections from the previous second are slowly faded out and there are two sounds. There's a pop sound once per second, accompanied by an animation. This plays even when nothing is detected. A tone is also played at different pitches as the detections come closer.

Electronic Drum Business Card

This is a project that is certainly worth making a noise about. By **David Crookes**

Maker

Sergey Antonovich

Sergey is an embedded systems engineer specialising in real-time system software and sensor integration for self-driving cars and delivery robots.

rpimag.co/drumcard

Want to drum your name into someone's memory at a network event? Then Sergey Antonovich has you covered. The embedded systems engineer has reinvented the age-old business card by turning it into a playable electronic drum kit – and we'd hazard a guess that this one won't end up languishing forgotten in someone's pocket.

Sergey got the idea for the project some months ago. He enjoys building digital musical instruments including ultra-portable all-in-one digital accordions and he was inspired by business cards which fit an entire Linux system on a tiny PCB. Producing a business card which could be played felt like a natural fit. "It does three jobs at once," he says. "It instantly grabs attention when you hand it over, it communicates exactly what I do, and it invites people to learn by reproducing the design."

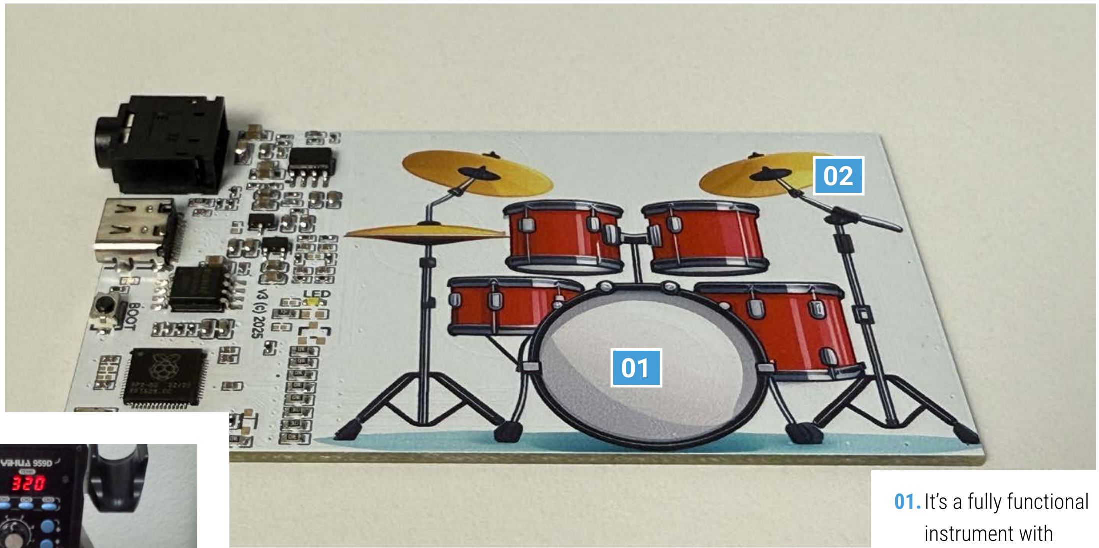
> *It instantly grabs attention when you hand it over*

This led him to create a card that incorporated a F1C100s system-on-chip running Linux as well as a TTP229 capacitive touch sensor. Recipients simply needed to power it up and listen via headphones connected to a 3.5mm TRS audio jack while tapping on an image of a drum kit printed on the card. "Touchpads were hidden under the silkscreen art so you could tap drums right on the print," Sergey says. "It worked, looked great, and felt like a real instrument."

▶ The cards are being prepared for hand-soldering. They are normally created in batches of ten

Digital Drumming

Yet Sergey wasn't entirely happy because the project proved costly and complex.



01. It's a fully functional instrument with touchpads hidden under the colour UV silkscreen

02. Sergey designed it to be fun and intuitive: tap the printed cymbal to hear a crash

▼ The RP2040 was soldered manually onto the PCB with the use of a hot plate and hot air

"A Linux image/toolchain is doable, but heavy for beginners or classrooms," he notes. "And even with trimming, a boot time of a few seconds was noticeable. I wanted instant-on."

To resolve these issues, he turned to the Raspberry Pi RP2040 microcontroller. "It's inexpensive and it uses external QSPI flash which is large enough for multiple stereo drum samples," he says. "It can also be programmed using CircuitPython and this takes you from idea-to-music in an evening. There's no heavy toolchain."

Indeed, as Sergey points out, he really did have a build ready within hours. "I started from a CircuitPython build for a 16MB RP2040 board, prepared a tiny drum sample set, and wrote a short script that scans pads and plays stereo WAVs via 'audiomixer' and 'audiopwmio'.

Feel the beat

The result has been a fresh RP2040 + CircuitPython version optimised for instant power-on via USB-C that is ideal for beginners while also being cost-effective and quick to create. The touchpads are read directly by RP2040, so there is no need for an external touch integrated circuit, and 16MB of external flash memory is more than sufficient for the code and sound samples.

Thanks to CircuitPython's touchio module, RP2040 senses the pad capacitance directly. "A finger increases the capacitance, which leads to changes in the charge/discharge time that results in a reliable 'touched' flag," Sergey explains. An LED gives instant visual feedback and the sound is output via two of RP2040's PWM channels.

Sergey now wants other people to make the project their own, which is why he has made it open source. But even though he has printed his own details on the reverse, including a QR code pointing to his LinkedIn profile, he says its use goes beyond sharing contacts.

"It's a memorable handout that grabs attention the moment you tap it," he says. "But the goal isn't just to hand someone a cool card – it's to give you a tiny, hackable instrument you can learn from and extend."

Quick FACTS

- Each card costs around $6 to make with DIY assembly
- Tap the printed drums on the card to hear the sounds
- Sound samples are stored in the memory
- It feels instant – latency isn't perceptible in play
- The audio is playable via plug-in headphones

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

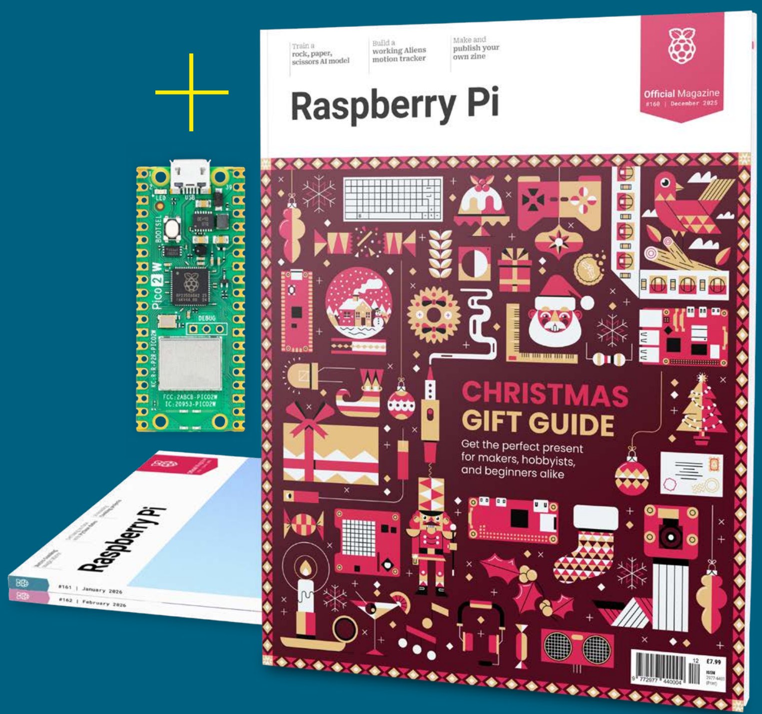
£10 (UK only)

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**

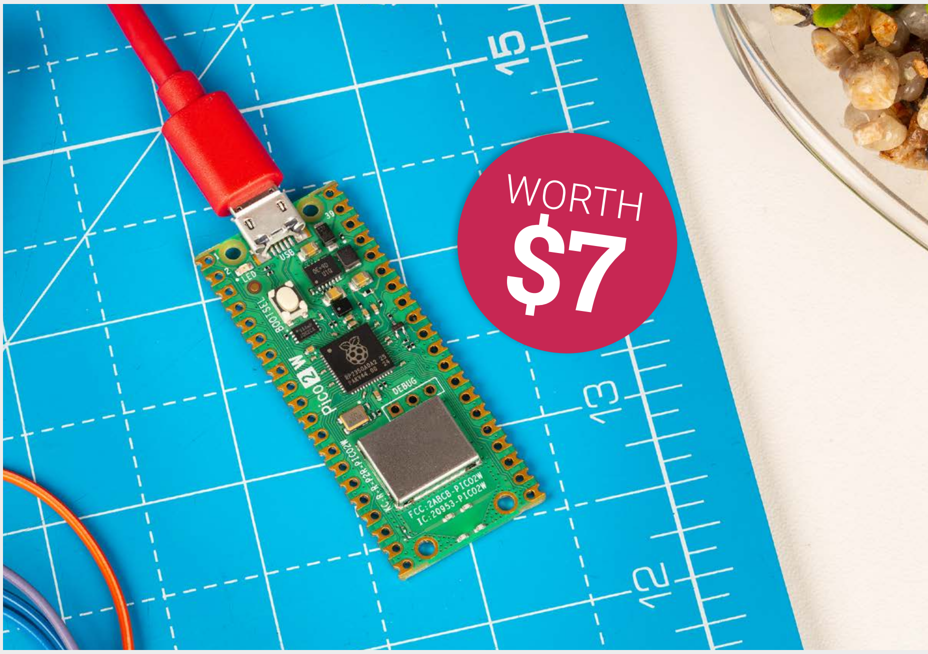
Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A choice of free Raspberry Pi Pico 2 W or Pico 2 is included with all subscriptions.

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.

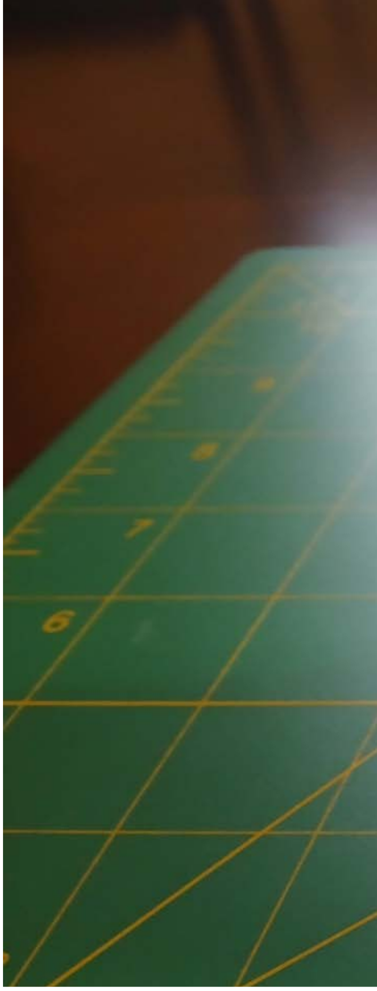
Buy now: **rpimag.co/subscribe**

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play



Banamera

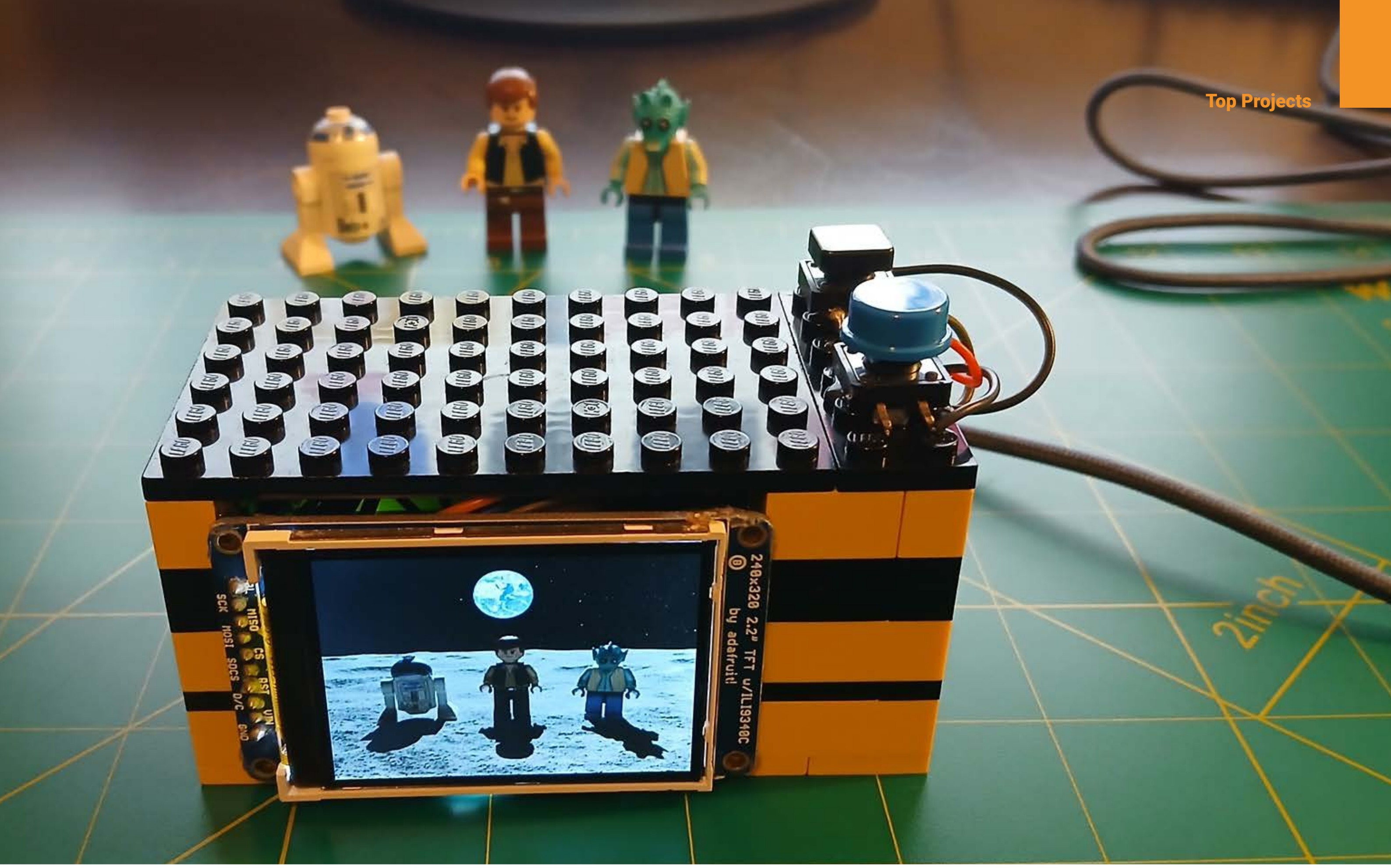
By Nick Bild

rpimag.co/Banamera

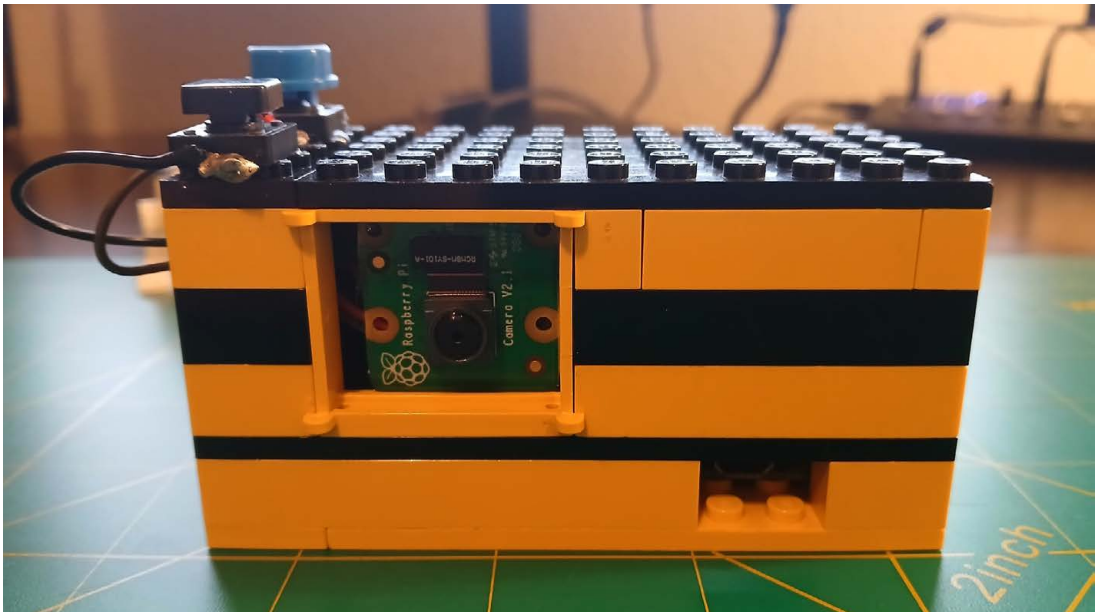
Despite the constant errors in AI generated images, the software behind AI is getting better all the time. Actually, that's wrong – it's not despite the errors, it's *because* of the errors. AI isn't actually intelligent; it's just recognising patterns. More data equals better results.

One of the more recent image editors is Google's Nano Banana, the abilities of which have pleasantly surprised maker Nick Bild. And in true maker fashion, Nick has designed and built his own machine to take the best of Nano Banana and improve it.

The problem that this build solves is Nano Banana's clunky interface. In Nick's own words, "the web interface is less than ideal. Upload an image. Type out a request. Wait. Download the result. Repeat for each image." Banamera streamlines this process. It uses a Raspberry Pi Zero 2 W with a Camera Module 2 and a 2.2-inch LCD screen to show the images. But there's also an Adafruit I2S MEMS microphone that enables the user to speak editing instructions, which are turned into text and forwarded to the Nano Banana API. The results appear on the screen within seconds.



▶ Lego bricks are perfect for prototyping – not everything has to be 3D printed



Project Gigapixel

By Yannick (Gigawipf)

rpimag.co/Gigapixel

One of this author's first jobs was scanning film transparencies for a publishing company, and using an early version of Photoshop to edit out the scratches that had been caused by the transparencies having been kept in a shoebox seemingly full of grit. The scanner in question was a large box containing, among other things, a CCD (Charge-Coupled Device) optical sensor. It was slow and fiddly, but in the early 2000s it was still better than the digital cameras of the time.

In the quest for ever greater image resolution, some enterprising hardware hackers have tried taking the CCDs out of scanners and repackaging them into high-resolution, though bulky cameras. With this project Yannick is going several steps further than this, by reverse-engineering the CCD so that a Raspberry Pi 5 can interface with the sensor assembly directly, replacing all other components of the original scanner with a custom implementation – including a snazzy 3D printed body.

▶ With a medium format 6×7 lens, this camera could theoretically be capable of producing a 3.3 gigapixel image


ASAHI OPT.
8472084

Single Digit Nixie Clock

By Daniel Knezevic

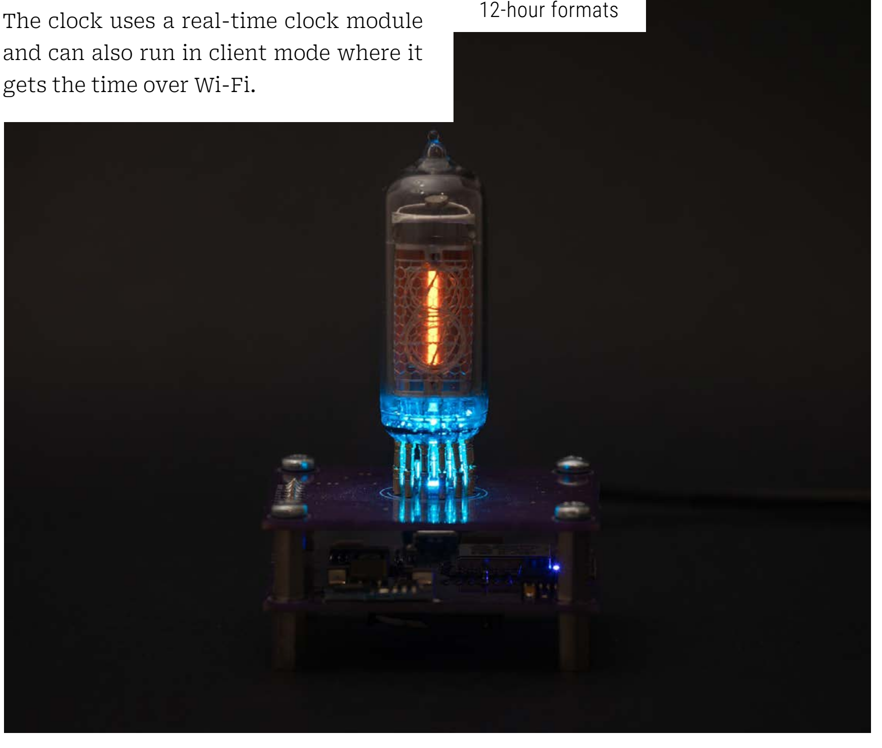
rpimag.co/SingleDigitNixieClock

The glowing screens of our devices are many things, but you'd never call them beautiful. Functional, flexible, yes; but stare at them too long and you'll get square eyes, your circadian rhythms will go awry, and you'll find yourself yearning to go outside and live in nature. Not so with Nixie tubes – these gorgeous, glowing filaments take a bit of working with, but they are beautiful enough that we want more of them in our homes, not fewer. These digits are made to admire.

So it was with great joy that we discovered a new (to us, at least) digital clock based on the IN-14 Nixie tube, by Daniel Knezevic. Daniel got round the main drawback of Nixie tubes (they're relatively expensive) by only using one of them: instead of using four tubes in HH:MM format, Daniel's clock rapidly flashes just the one tube with four digits, one after the other. This means that you can't tell the time with a glance – you actually have to look at the clock for long enough that you catch all four digits, but that's no hardship when it looks as good as this.

Daniel's clock uses an ESP32 controller and two custom PCBs: one to host the ESP and the power supply to the Nixie tube, and one to host the Nixie and the RGB LED. The clock uses a real-time clock module and can also run in client mode where it gets the time over Wi-Fi.

▼ Daniel's single-digit clock can display the time in 24-hour or 12-hour formats



TVArgenta

By Ricardo Sappia

rpimag.co/TVArgenta

There's something really weird about the nostalgia we get from television. These days, streaming services allow us to watch whatever we want , whenever we want it. Whereas TV used to be broadcast on a set schedule. Even today, the *EastEnders* theme tune flicks a mental switch telling us that it's time to go to bed because we have to get up for school in the morning; the theme from *Antiques Roadshow* means it's Sunday night, and the weekend has drawn to a close. The television marks time in a way that online entertainment just doesn't. A programme from your youth can take you right back there just as effectively as one of Proust's madeleines.

That feeling of nostalgia is what Ricardo Sappia wanted to recreate with this build, the TVArgenta. He was raised in Argentina, and now lives in Germany, and to give his kids a taste of Argentinian TV from his youth, he's created a fully offline nostalgia machine. This TV uses a 4.3-inch DSI display, a rotary controller for a dial, and an I2S audio interface, all held together by a Raspberry Pi 4.

The project's Raspberry Pi 4 handles content management, channel creation and tagging, video synchronisation and thumbnail generation, and an admin interface for editing and configuration. But possibly our favourite aspect of this build is the 3D printed enclosure, which is shaped like the bulky television sets that used to house CRT tubes, rather than the more modern flat screens of today.

▶ "For me they're not just ads," says Ricardo "… they're small pieces of memory with the smell of home"


TVA
TVARGENTA


RTC
C.7

3D print

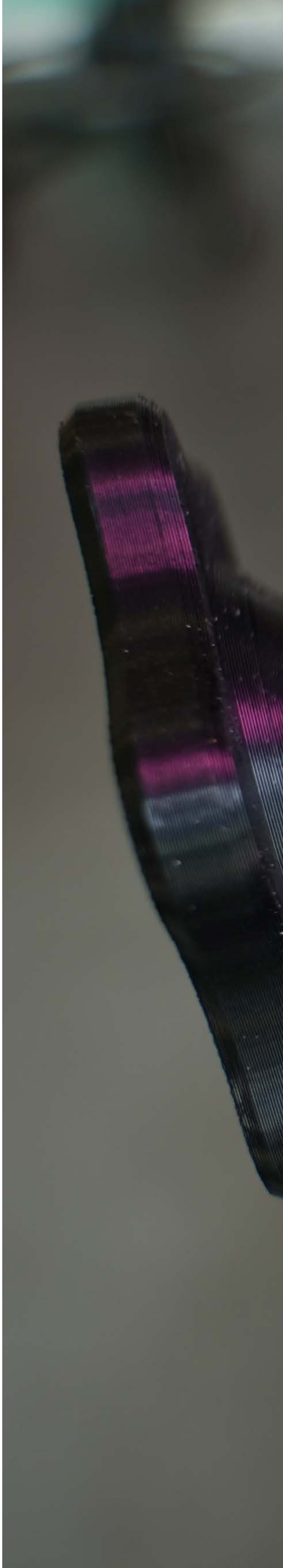
Need an extra pair of hands? This mini bench vice is ideal for simple jobs. By **Toby Roberts**

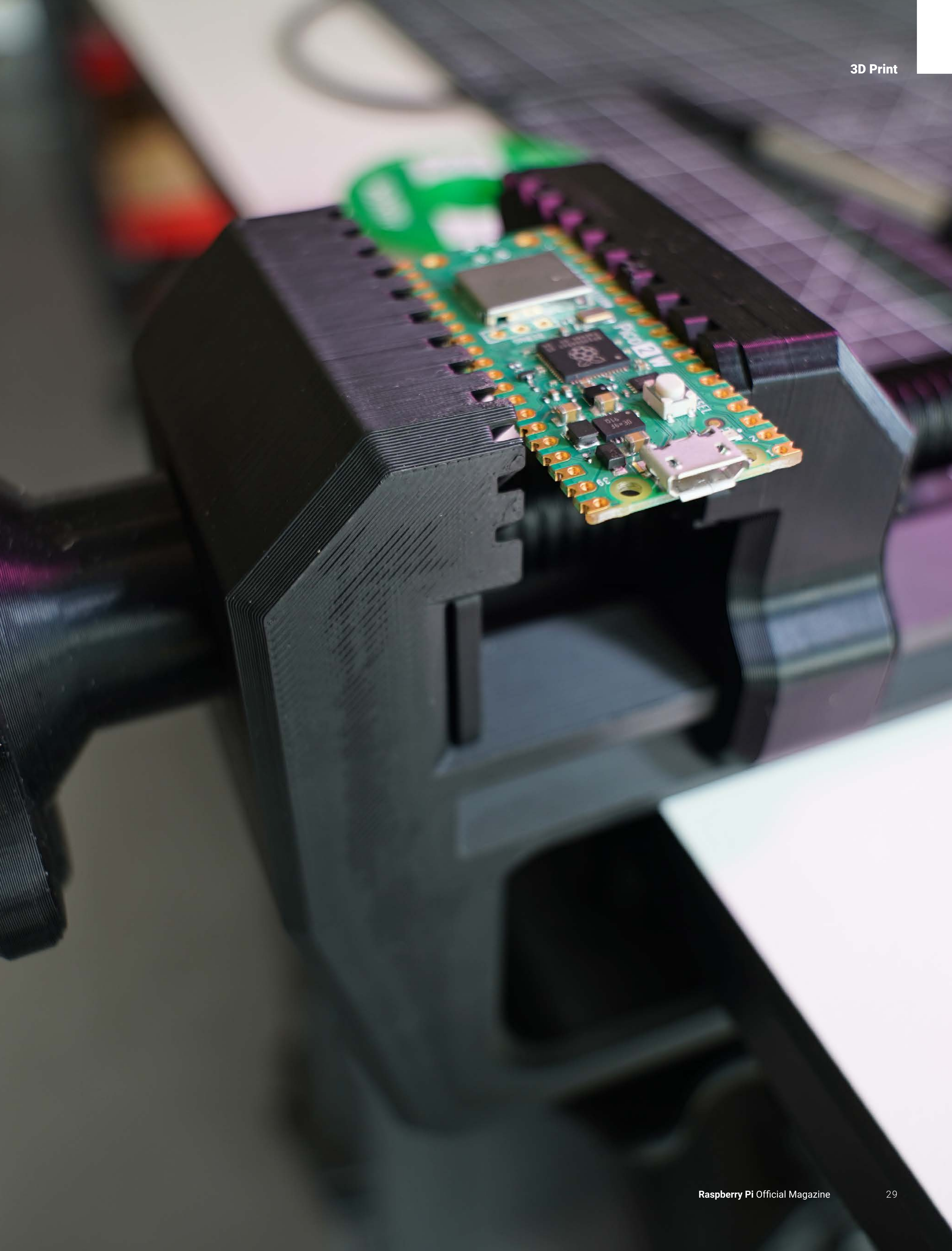
rpimag.co/3D-Print-Vice

We've printed plenty of fun and games recently, from simple card kit aeroplanes to a propeller launcher that used planetary gears to generate enough force to launch a spinning toy high into the air.

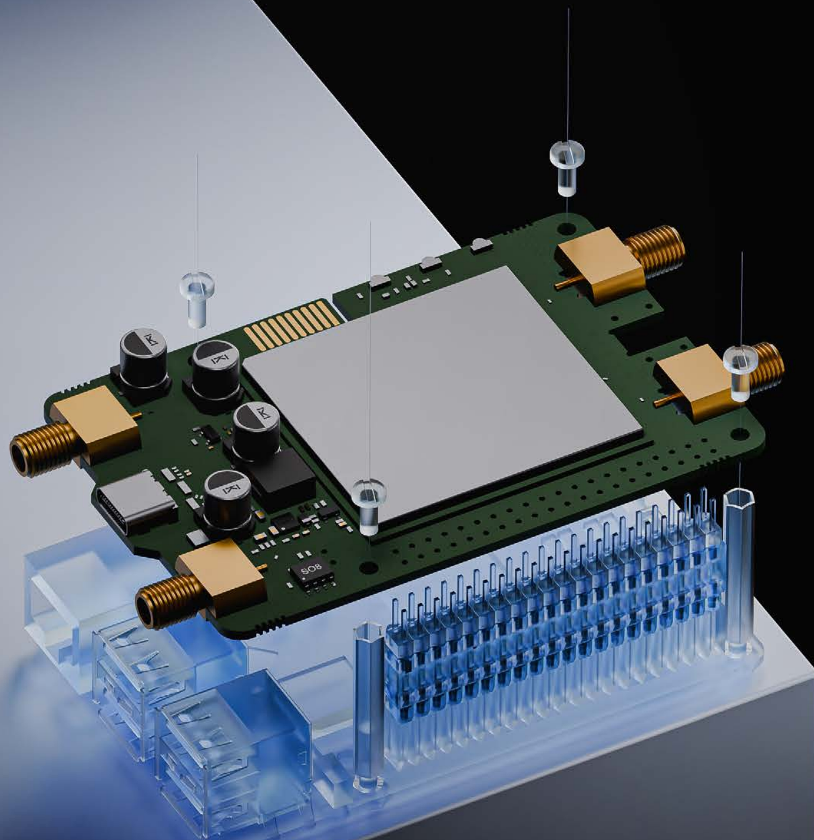
Sadly though, work gets in the way, and this month we needed an extra pair of hands to hold a PCB we were working on. So enough with the fun and games, and time for something practical: a genuinely handy little bench vice. Sure, this 3D-printed version won't replace a heavy-duty steel one, but for simple jobs like gluing or soldering it does the trick, holding pieces firmly enough that you'll soon wonder how you managed without it.

Designed by @ms520, it's a compact unit (80 × 90 × 40mm) with a 0–40mm clamping range. I first printed mine in PLA at the suggested settings, but over-tightening cracked the main thread. Switching to PETG for a touch more flexibility, and upping the infill percentage, solved the problem perfectly.





TELTONIKA



CELLULAR RASPBERRY PI HAT+

CALYX

Instant mobile connection
to your Raspberry Pi

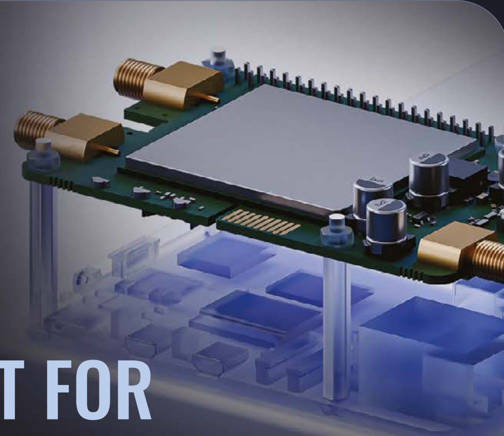
5G / 5G Redcap / 4G

Designed for
Raspberry Pi 4 & 5

Backwards compatible connectivity

AT Commands

Plug-n-**play**

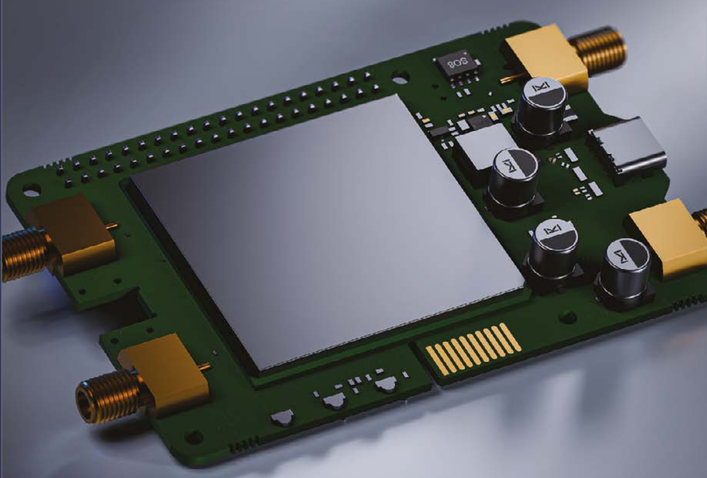


BUILT FOR RASPBERRY PI 4/5

Mount seamlessly with the compact HAT form factor

Wide temp **range**

1 x **4FF SIM**



CALYX

Cellular Raspberry Pi HAT
5G | 5G Redcap | 4G Cat 4

PCM Enhanced

Integrate voice calls
and audio streams

SBC form factor

Optimized for Single
Board Computers

GPIO CONTROL

Wake, reset, and manage
modem via Pi pins

CHRISTMAS GIFT GUIDE

Calling all Santas – here's a helping hand for getting makers of all ages the perfect gift this holiday season

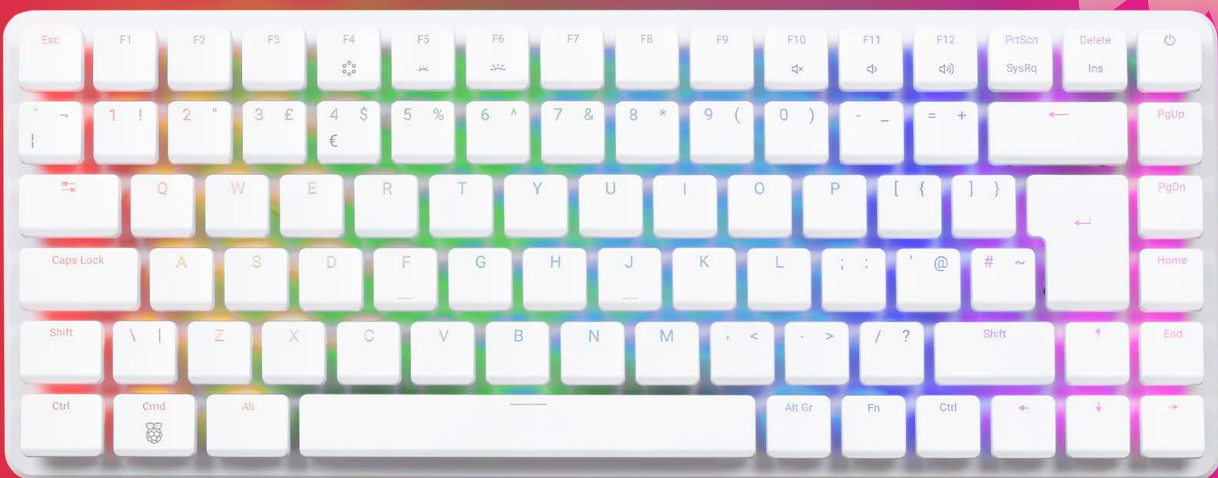
By Rosie Hattersley

Inspiration for makers, coders, inventors, smart home converts, gamers

Christmas is the perfect time to gift someone you love – possibly even yourself – some tech goodies to make life go with a swing and divert you with tinkering and creative possibilities. If family members have been hinting hard about the latest and greatest Raspberry Pi goodness, these very pages should serve as inspiration for makers, coders, inventors, smart home converts, and gamers of all stripes.

GET A RASPBERRY PI FOR CHRISTMAS

Who doesn't need more Raspberry Pi in their lives? Here are our top choices



Raspberry Pi 500+

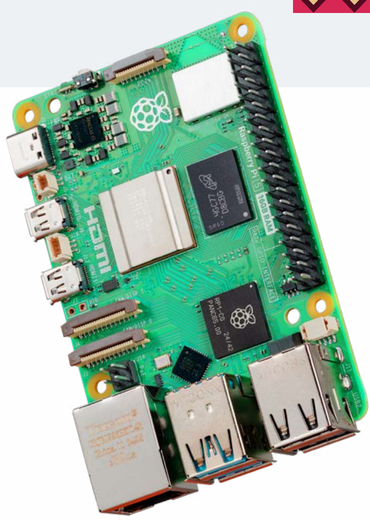
£192 / $200
rpimag.co/500plus

Crammed with 16GB RAM and a 256GB SSD, Raspberry Pi 500+ has both retro computing appeal and plenty of processing power. A special version of Raspberry Pi 5, equipped with 16GB RAM, is concealed beneath the low-profile RGB backlit mechanical keyboard. It also houses a 256GB NVMe SSD for storage. An aluminium heatsink keeps everything cool, ensuring optimal computing performance, even when serving up 4K video. Raspberry Pi 500+ is also available as a Desktop Kit with USB-C power supply, mouse, HDMI cable, and the *Official Raspberry Pi Beginner's Guide* book.

Raspberry Pi 500

£96 / $99
rpimag.co/rpi500

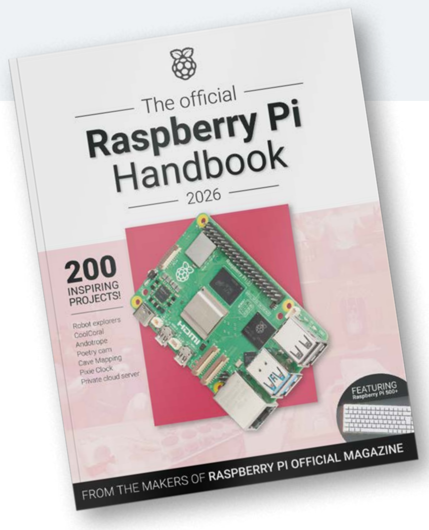
A smart keyboard conceals the self-same 2.4GHz 64-bit Arm processor and RP1 I/O controller found in Raspberry Pi 5, along with a performance optimising heatsink, Bluetooth, USB 3.0 ports, and dual HDMI outputs for HD entertainment.



Raspberry Pi 5

£114 / $120 for 16GB model [£76.80 / $80 for 8GB]
rpimag.co/raspberrypi5

The flagship Raspberry Pi, the 16GB variant offers a significant leap forward in single-board computing with two-to-three times the performance of the already powerful Raspberry Pi 4. The 2.4GHz quad-core Broadcom Cortex A76 processor and dedicated video GPU make Raspberry Pi 5 an excellent choice for gaming, multimedia tasks, and 3D rendering, and perfect for high-definition video playback thanks to HDMI support for two 4K screens.



Official Raspberry Pi Handbook 2026

£14 / $25
rpimag.co/handbook2026

If you're giving someone special a shiny new Raspberry Pi, the perfect accompaniment is surely the official handbook to help them make the most of their brand-new computer. Pair this excellent tome packed with advice, tutorials, and product details with a subscription to *Raspberry Pi Official Magazine* and they'll get even more inspiration throughout the year!

GIFTS TO GIVE AND RECEIVE

We won't tell if you don't if these gifts just happen to end up in your own stocking

Pi Hut Advent Calendar Let It Glow

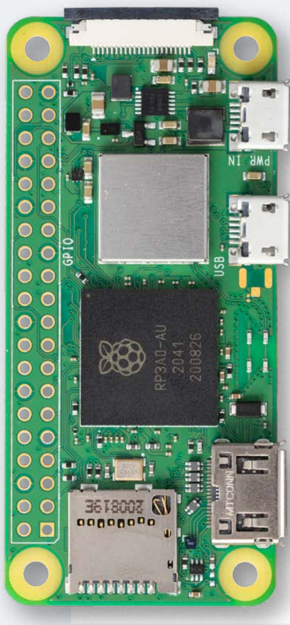
£45 / $59
rpimag.co/pihutadvent

If you love all things that flash and light up, the projects in the Let It Glow advent calendar are bound to appeal. Each daily surprise works with the bundled Raspberry Pi Pico H and breadboard, and combine to create festive lights and other glorious RGB-based displays.

Raspberry Pi 1TB SSD

£67 / $88
rpimag.co/ssd

Raspberry Pi's operating system and code to run additional hardware and apps all work fine from a microSD card. However, photography, multimedia, AI, and other memory- or graphics-intensive tasks will really benefit from the much more capacious storage offered by a far faster-to-access SSD. Solid-state memory has the further benefit of boot-up times "so fast it's crazy" in the words of one happy customer. Unless you're using a Raspberry Pi 500+, you'll need an M.2 HAT+ board to connect it.



Raspberry Pi Zero 2 W

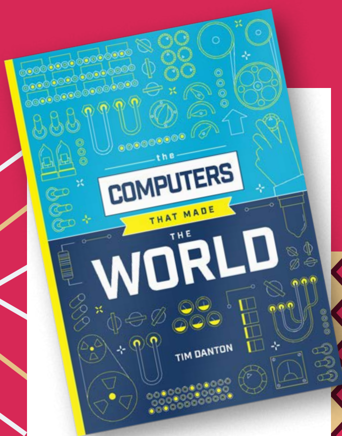
£15 / $15
rpimag.co/zero2w

We've secretly named the incredibly popular Raspberry Pi Zero 2 W our little genius. It works with a huge range of hardware, making it an obvious choice for maker projects, gaming, and anything handheld. From Bluetooth to discreet sentinel duties, the adaptable but eminently affordable Zero 2 W is one to snap up, safe in the knowledge it will be exactly the right Raspberry Pi for your very next make.

Pimoroni Inky Impression

£80 / $88
rpimag.co/inky73

We love the pointillist effect of the Inky Impression's e-ink screen which comes in an 800×600-pixel 7.3in and a 13.3in (£229) version. It manages to have both a matte, muted effect as well as being vibrant and sharp. Pimoroni clearly spent a long time perfecting the display code.



The Computers That Made The World

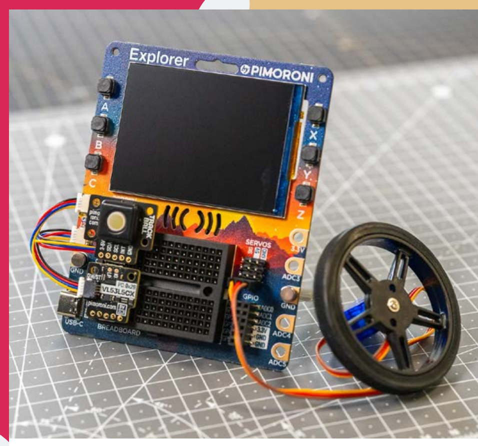
£20 / $22
rpimag.co/ctmtw

Charting the development of influential computers built between 1939 and 1950, this fascinating book tells the hidden stories of celebrated and lesser-known inventors, as well as revealing the secretive circumstances in which many of these ground-breaking machines came about.

Nanoleaf Pegboard

£60 / $78
rpimag.co/nanoleaf

Keep desk clutter to a minimum while brightening up your computing and gaming space. Nanoleaf's Pegboard provides a light-up stand from which to dangle gaming handset, mouse, headphones, your glasses, or even a Raspberry Pi-based console if that isn't putting temptation quite too close to hand.



Pimoroni Explorer

£34 / $38 board only; Starter Kit £60 / $67
rpimag.co/explorer

Based around the RP2350 microcontroller (as used in Pico 2), it provides an engaging introduction to coding and prototyping electronics projects. Features include a 2.8in LCD, speaker, mini breadboard, servo headers, and analogue inputs. The Starter Kit adds a multi-sensing stick, potentiometer, LEDs, push-buttons, two continuous rotation servos, two 60mm wheels, jumper wires, and an AAA battery pack with Velcro to attach it.

GET GAMING

Load up the latest retro games and puzzles or code your own

Thumby Color

£39 / $49
color.thumby.us

More than 100 free and open-source games can be enjoyed on this tiny games console based around RP2350 which functions as a development board for further MicroPython projects. There's even haptic feedback via a minute-but-mighty rumble motor. Add one to your keychain this Christmas.

Raspberry Pi Touch Display 2

£58 / $60
rpimag.co/touchdisplay2

This 7in (or 5in) touch-sensitive screen is ideal as a dashboard for smart home and multimedia projects as well as an information board. The 800×480-pixel screen works with Raspberry Pi 3B, 3B+, 4, or 5 via the DSI port. Use it to access menus, control IoT devices, or to play retro games using an emulator such as RetroPie (**retropie.org.uk**).

Retroflag NESPi 4 Case for Raspberry Pi

£28 / $37
rpimag.co/nespi4

Retroflag's NESPi case is a great choice if the lucky recipient is an ardent retro gamer. Though it no longer comes supplied with a heatsink, the Raspberry Pi 4 (not included) setup offers both internal and external USB 2.0 ports, a microSD storage slot, 10/100 Ethernet port, plus power and reset buttons on the unmistakably 1980s-style hard plastic case. Open the front flap and you can slot in a 'game cartridge' which is actually a case for a 2.5in SSD (not supplied).

MAKERS GONNA MAKE

Begin or extend your retinue of project-creating tools with a printer, camera, case or consumables

Put a 3D printer to work on these baubly nice Christmas decorations: rpimag.co/printdecs

Bambu A1 Mini

£169 / $179
rpimag.co/a1mini

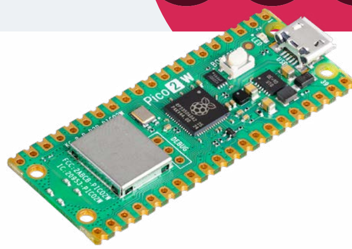
Bambu has become one of the go-to maker brands, helping 3D printing become increasingly affordable as well as sophisticated. The four-colour A1 lets you print multi-coloured designs (albeit more slowly than single-colour prints), but it's the unbelievably affordable A1 Mini that really impresses us. It works out of the box, the software is intuitive, it's quiet – altogether it's a great introduction to 3D printing.

CrowPi 3 AI Learning and Development Station

£168 / $229

rpimag.co/crowpi3

CrowPi 3 turns Raspberry Pi 5 into a portable lab in which to experiment with everything from engineering and advanced tinkering to making sense of the world via machine learning and OpenCV AI processing. Among a host of onboard modules to play with, this exciting and adaptable educational hub features a built-in camera and microphone, plus a host of lessons to follow.

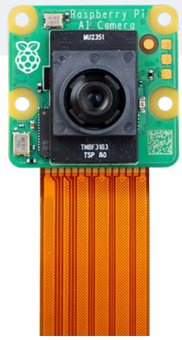


Raspberry Pi Pico 2

From £5 / $5

rpimag.co/pico2

Powered by the high-performance RP2350 microcontroller, Pico 2 is a great option for makers who need something tiny and powerful for portable projects. Like the original Pico, it features plenty of GPIO pins (including analogue inputs) to connect electronic components and sensors. The 2 W model adds wireless connectivity in the form of 802.11n 2.4GHz WiFi and Bluetooth 5.2, making it ideal for IoT projects.



Raspberry Pi AI Camera

£63 / $70

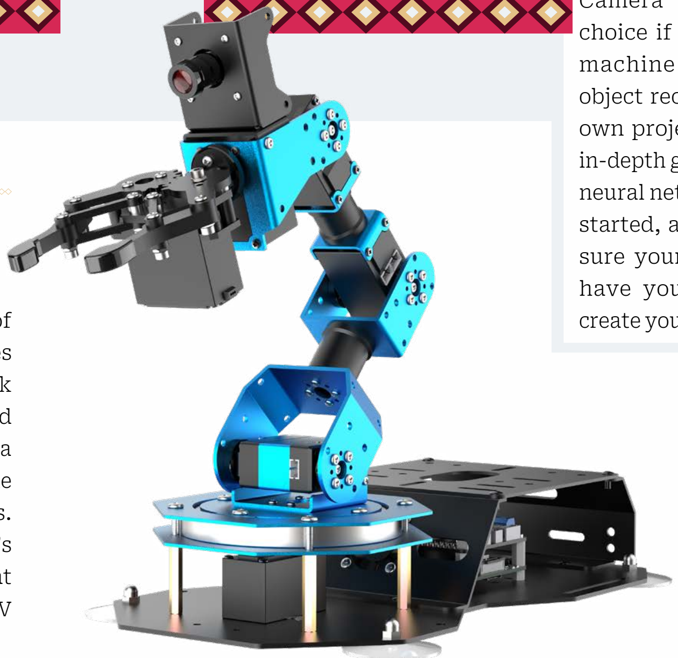
rpimag.co/aicam

Based on Sony's 12MP IMX500 intelligent vision sensor, our very own AI Camera is an excellent choice if you want to try machine learning and object recognition in your own projects. We provide in-depth guides and sample neural networks to get you started, after which we're sure your own ideas will have you clamouring to create your own AI projects.

ArmPi AI Vision Robot

£228 / $300

rpimag.co/armpifpv



Our reviewer was amazed by the quality of this excellent robot arm, which manoeuvres through six degrees of freedom, flexing back and forth to catch balls, grab objects, and follow lines that the user dictates. Thanks to a top-mounted HD camera and AI, it's also able to recognise colours, faces, and obstacles. Bristling with tutorials and possibilities, it's a great learning and making tool and a great advert for Raspberry Pi paired with OpenCV machine learning.

SOUND AND VISION

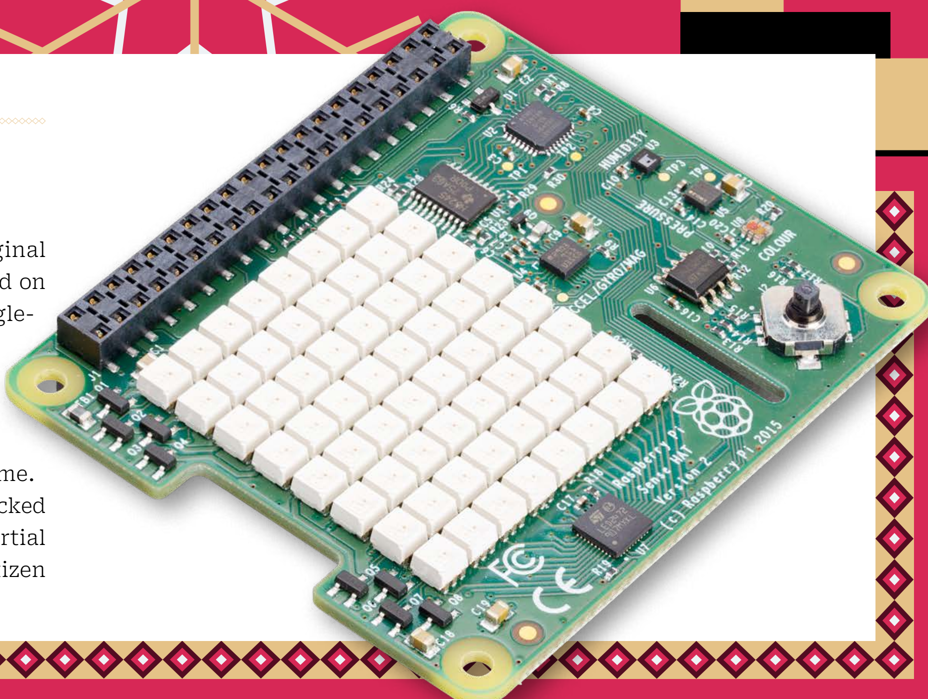
Music to your ears and beautiful visuals too

Sense HAT enabled our single-board computer to make its space debut

Sense HAT v2

£29 / $33
rpimag.co/sensehat

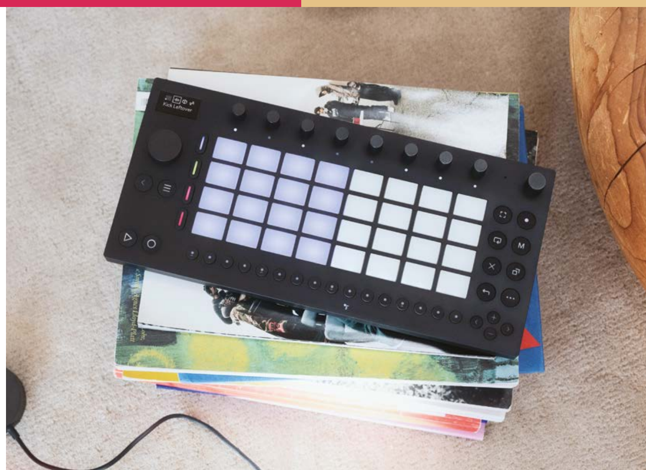
The Sense HAT was both the original Raspberry Pi HAT (Hardware Attached on Top) accessory and enabled our single-board computer to make its space debut, beaming back visual and atmospheric details from beyond the Earth's atmosphere, and launching the Astro Pi school education scheme. Along with an RGB LED matrix, it's packed full of sensors including an IMU (inertial measurement unit). It's ideal for citizen science as well as coding experiments.

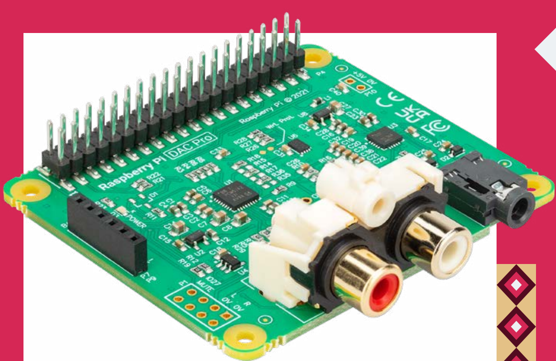


Ableton Move

£399 / $449
ableton.com

Ableton Move is a portable instrument with a library of more than 1500 sounds and presets. It has a 16-step sequencer, 32 polyphonic pads, reverb, delay and saturator effects, and its own Ableton Live software suite. All this glorious sound is based on Raspberry Pi hardware (a Compute Module 4) of course, so you just know it's going to be good.





Raspberry Pi DAC Pro

£24 / $36
rpimag.co/dacpro

This high-fidelity audio HAT is based on Texas Instruments' PCM5242 DAC, has its own microphone, works with Raspberry Pi Zero 2, 3B, 4 and 5 (connecting via GPIO pins with no external power needed), and boasts an impressively high signal-to-noise ratio.

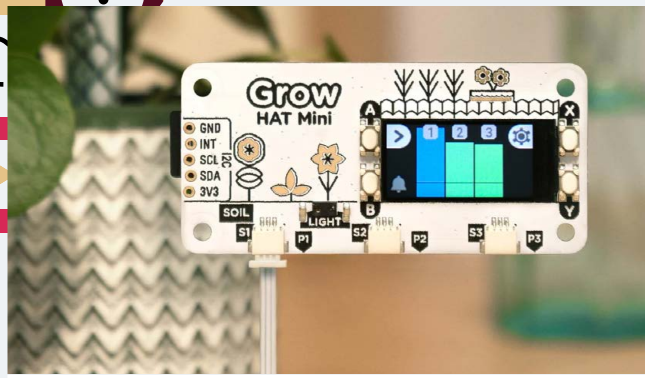
Raspberry Pi Monitor

£96 / $129
rpimag.co/monitor

This 15.6in IPS display is designed to pair beautifully with Raspberry Pi 400 or 500/500+ and comes in distinctive red and white livery or a smart black version. With an HDMI connection and pair of front-facing speakers, this crisp screen can be VESA wall-mounted for multimedia enjoyment. The monitor works with Raspberry Pi 3B, 4, or 5.

HOME COMFORTS

Make downtime cosier, simpler and run smoother with smart tech tricks



Pimoroni Grow HAT Mini

£33 / $43
rpimag.co/growhatmini

Grow your own herbs, chillies, and vegetables with the aid of this moisture-sensing kit that will alert you should your plants be crying out to be watered. The Grow HAT comes with three capacitive moisture sensors that act like smart plant labels. (Fret not, the Grow HAT can be set not to trigger Raspberry Pi's alarm in the middle of the night.) It can also control low-energy pumps, motors, and grow lights, and you can use its input to automatically message you when plants are thirsty.

DreamHAT+

£100 / $135
dreamboards.co.uk

Personal radar is a relatively new means of location tracking that lends itself well to indoor scenarios such as Supersense's Sens2 hub to benignly monitor dementia patients (**rpimag.co/sens2**) or to tracking pets and other creatures in a defined area. Setup involves installing the preconfigured microSD card, then running through tutorial videos on Dream RF's YouTube channel so you understand the classic Doppler and azimuth graphs that the 60GHz millimetre-wave radar has uncovered.

Tado Smart Thermostat X Starter Kit

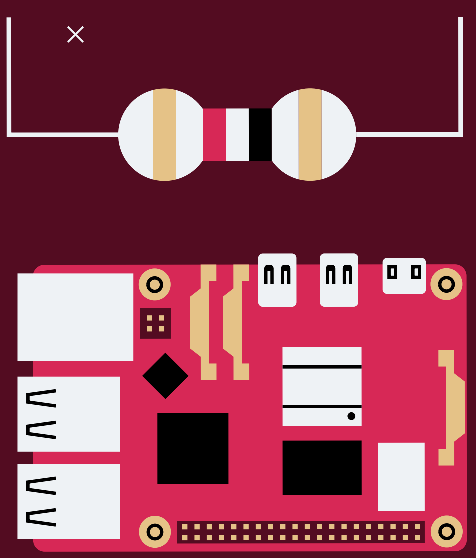
£109 / $143
rpimag.co/tadothermostat

Goldilocks knew just how important it was to get the temperature just right. Tado's Smart Thermostat uses Matter (or another Internet of Things standard) to connect Raspberry Pi and cloud services so you can adjust settings to a tee. Add-ons for underfloor heating and additional hubs extend the cosiness across larger homes.

PROJECTS FOR YOUR CHRISTMAS GIFTS

Got a Raspberry Pi and some accessories from Santa? Here's how to use them…

Master of gift-giving: Rob Zwetsloot



Depending on how highly on Santa's Naughty or Nice list you are, you may have received a whole host of Raspberry Pi and related goodies for Christmas. This may be leading you to ask a question that is as old as Raspberry Pi itself: what do I make with it?

Fear not, there are a million amazing projects you can do with Raspberry Pi, and here are just some to get you started with your brand new microcomputer – or inspire you to do something else of your own!

MEDIA PC AND SERVER

Chill and relax with the most convenient media setup ever

▲ You can also use your home media centre to play retro games if you desire

For gifts: Raspberry Pi 5, Raspberry Pi Monitor, DAC Pro HAT

rpimag.co/155

With a tidy case, Raspberry Pi 5 is the perfect way to power up your telly. With an intuitive and easy UI that easily connects to network shares, you can have all your home media right at your fingertips. You can also connect streaming services so they're all linked to one input if you so wish.

In this guide available in issue 155, we go over all the essential hardware and software you'll need for this project, as well as an example of how to set it (and network storage) up. All you need to do is provide the popcorn.

Magic Mirror

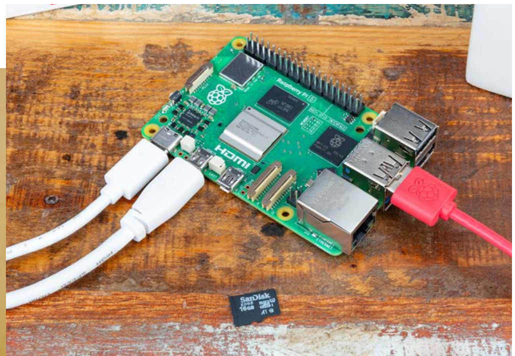
magicmirror.builders

Use your Raspberry Pi 5 and monitor (or old TV!) to make a smart mirror that compliments you every morning.

► Smart mirrors are easy to make, you just need to do a little bit of carpentry beforehand

FULL DESKTOP

Unleash the full power of Raspberry Pi while sitting at a desk



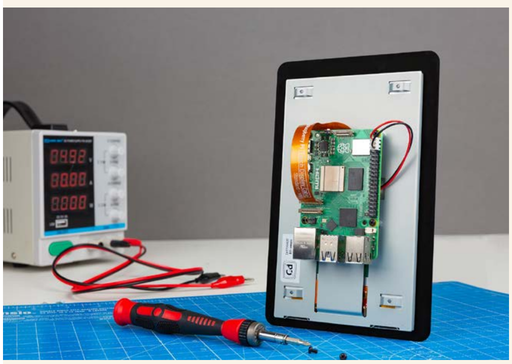
▲ A Raspberry Pi 5 can be connected to a keyboard and monitor to make a desktop computer

Kitchen computer

rpimag.co/kitchenpc

A kitchen computer can do wonders for your baking, and can be easily set up with just a Raspberry Pi and a Raspberry Pi Touch Display 2 – you can even go all out and mount it under a cabinet!

▼ The Touch Display 2 is a great little monitor you can place in fairly small spaces



For gifts: Raspberry Pi 500+, Raspberry Pi 500, Raspberry Pi Monitor, Raspberry Pi 1TB SSD

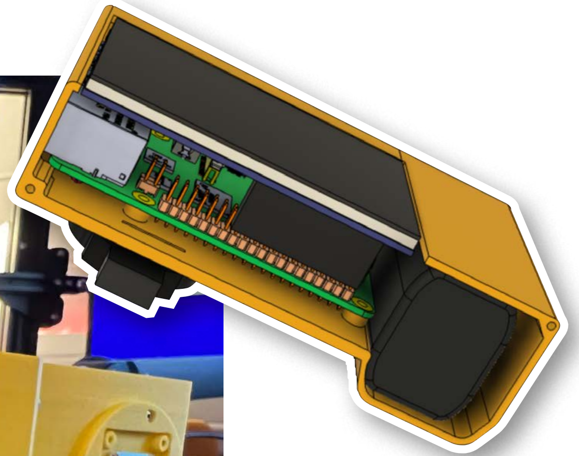
rpimag.co/156

When we say you can use Raspberry Pi as a desktop computer, there are no caveats to that. With a blazing fast CPU, native dual-screen 4K video, and a highly optimised operating system with excellent compatibility features, it can easily be your desktop-away-from-your-desktop.

For many users, just installing Raspberry Pi OS will be all they need to do. However, for those a little more intimidated by the change, we have a full guide on the ins-and-outs of completely replacing your desktop with Raspberry Pi. While this is definitely better/easier with Raspberry Pi 500 series computers, a standard Raspberry Pi 5 can also be used if you have even tighter space requirements.

POINT-AND-SHOOT CAMERA

Make a very customisable camera with Raspberry Pi



▲ The parts are squeezed in thanks to clever 3D design

For gifts: Raspberry Pi 5, Raspberry Pi Zero 2 W, Raspberry Pi AI Camera, Bambu A1 Mini

rpimag.co/pikon2

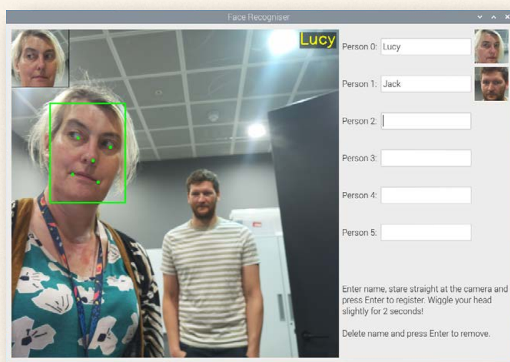
The Camera Modules for Raspberry Pi are powerful and very customisable thanks to all their operations being powered by Python code. While you can easily set up a simple way to modify settings with rotary dials and take a photo with a shutter button, you can also give it smart features where it reacts to the environment.

People love to retrofit old cameras (**rpimag.co/leicampi**), but if you don't have access to one, you can easily 3D-print your own version. This tutorial for the upgraded Pikon II by regular in the mag Kevin McAleer guides you through how to program and assemble the camera using his excellently designed case.

Object recognition camera

rpimag.co/157

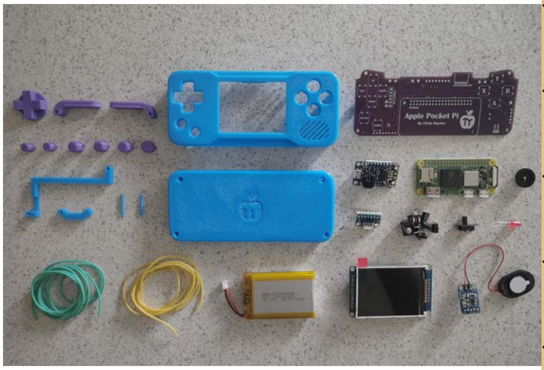
With an AI Camera installed, you can do some smart analysis, with object or even face recognition built-in.



▲ Find out who is in your photos with face recognition

HANDHELD GAME CONSOLE

Portable retro gaming with no limitations



▲ All the parts for the build – there's plenty of 3D printing to get on with

For gifts: Raspberry Pi Zero 2 W, Bambu A1 Mini

rpimag.co/pocketpi

Raspberry Pi Zero 2 W is as powerful as a Raspberry Pi 3, albeit in a much smaller package. That smaller package means you don't need as big a build to use it, so a pocketable handheld console (we're looking at you, Game Gear) is an easy proposition.

This Apple Pocket Pi is a great handheld build, borrowing the sideways form factor that fits nicer in the hands and allowing for a backlit screen, rechargeable battery, Wi-Fi capabilities, and the added bonus of being able to plug in external controllers and hook it up to a TV. All you need to do is supply your own games.

Warning!
Legal ROMs
Find out about legal ROMs for emulation at **rpimag.co/legalroms**

Make your own game

color.thumby.us

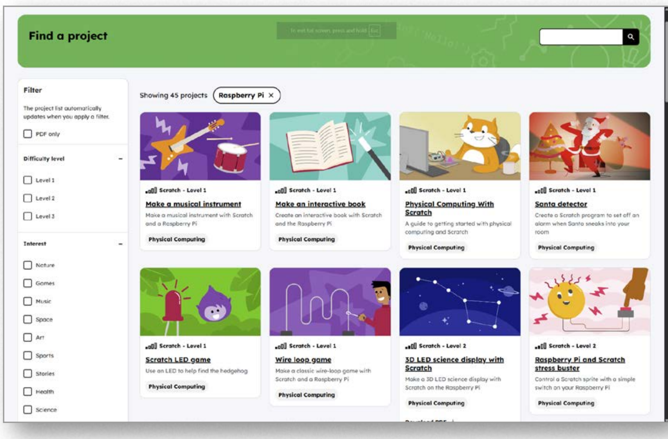
Got a Thumby Color for Christmas? It runs on RP2350, making it slightly different than emulation on Raspberry Pi, and ripe for making your own video game dreams a reality.

▲ A great little games console and a secret development platform

FUN ELECTRONICS

Get started with electronics with a Raspberry Pi Pico (or Raspberry Pi) and a few components

Keep an eye out for our next issue which will include a whole feature on coding and programming!



For gifts: Raspberry Pi Pico 2, Raspberry Pi 5, CrowPi 3, Sense HAT v2

rpimag.co/projects

Raspberry Pi was originally created to get folks back into computing, and even though many people don't even use it for that, it's still the best way to learn about digital making, electronics, and coding.

We love using Raspberry Pi Pico (and Pico 2) in small electronics projects – it's very easy to program, has very low power requirements, and can be slipped into the smallest space. The Raspberry Pi Foundation has some excellent tutorials that use Pico (and a full Raspberry Pi) that cover a wide variety of interests.

> Start off with a simple LED lighting project

We suggest starting off with a simple LED lighting project and go from there – it's amazing how quickly you'll be inspired to do something yourself

▲ Perhaps the appropriately themed Santa detector should be your project?

Electronics kit

rpimag.co/picolitekit

For electronics projects, you'll need to get an electronics kit – these come filled with different components to get you started with tons of projects.

▲ Resistors, capacitors, buttons, LEDs, buzzers, and much more to play around with

Part 02

Christmas lights: designing patterns

Let's make the most of our hardware and illuminate the night

Maker

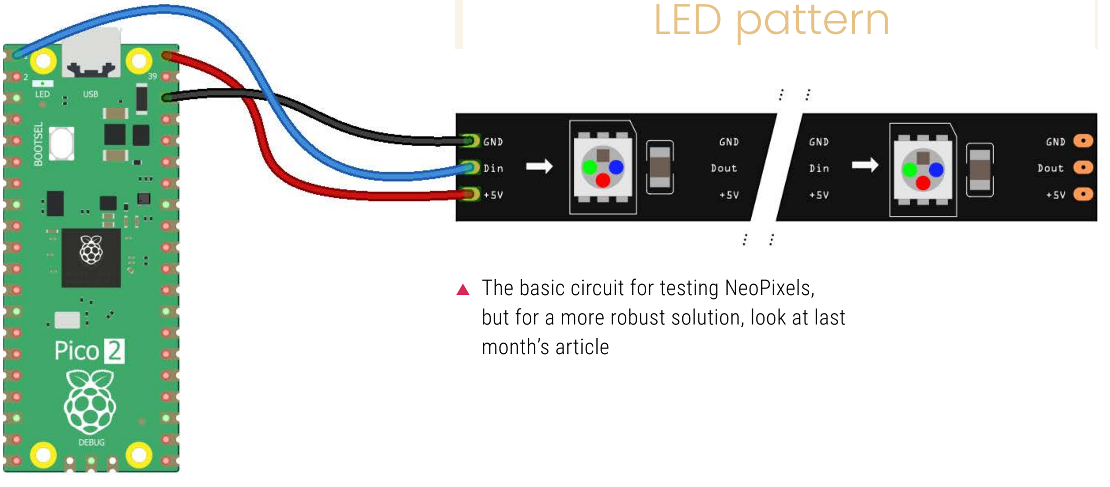
Ben Everard

Ben is a light artist who has shown NeoPixel-based art at light festivals around the UK.

glowingart.co.uk

Last month we looked at different forms of WS2812B LEDs (also known as NeoPixels) that you can get to make Christmas lights. Now let's take a look at how to use them. We're not going to dwell on the particular software too much – there are libraries available for just about every language you're going to come across, and you can control them on Raspberry Pi, Pico, or most other platforms that you can physically access a GPIO on (we'll use CircuitPython on Pico, but you could translate it into another language or platform fairly easily). Instead, we're going to think about what we want them to do. In other words, we're not going to look too much at how to make them work, we're going to focus on how to make them look good.

> Black is one of the most powerful colours in an LED pattern



▲ The basic circuit for testing NeoPixels, but for a more robust solution, look at last month's article

▲ Our pattern in action on some NeoPixel fairy lights

Monochromatic

There is a separate take on colour, and that's not to vary the hue, but the saturation. The palette would then all have the same hue, but you'd pick out a range of saturations. This can have a powerful effect. For example, a monochromatic blue palette feels very cold, while green might give a feeling of nature.

Balancing act

In our experience, getting a good-looking Christmas light pattern is about creating balance in three different things:

- **Movement** – there's a common joke in the maker community that things need more blinking lights. However, harsh on-and-off blinks can be really garish, distracting, and rarely look good. Instead, we usually want our movement to be more fluid. There is a balance between fluidity of movement and speed.
- **Colour** – you can create almost any colour with WS2812B LEDs. However, that doesn't mean that you can just mash every colour together and end up with something looking good. A well thought-out colour palette is probably the single thing that has the most impact on making a good-looking Christmas light display.
- **Brightness** – WS2812B LEDs often come in large blocks. Strings of hundreds of them. Each one may not be that bright, but lots of them quickly get very bright. Just because you have all the LEDs doesn't mean that they have to all be on at the same time. Black is one of the most powerful colours in an LED pattern. You can move patches of colour around, or simply get a pattern that users can look at without hurting their eyes.

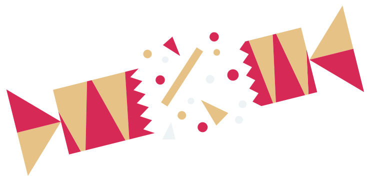
Colourify

First, let's take a look at colour. Obviously, the most classic Christmas combination is red and green. And it's a classic for a reason. Colours are complementary if, when you combine them, you get grey. They contrast strongly with each other, but in a way that seems to balance.

Colour theory is often shown using a colour wheel. This has all the hues around the outside. Complementary colours are opposite. The hue is often given as the number of degrees around the outside of the colour wheel you have to travel.

You can continue the basic principle of building a palette out of colours that combine to make grey with more colours. With three colours, you're looking for colours that are 120 and 240 degrees apart. You can continue to work upwards with more colours, ensuring that they are evenly spaced around the wheel.

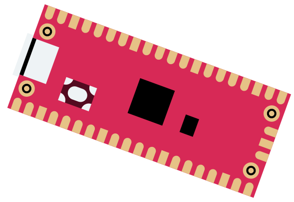
However, the distance around the outside of the colour wheel only defines the hue, and this is only one part of colour. For example, you won't find pink on the outside of the colour wheel, and this is because pink is the same hue as red, but with less saturation. Saturation defines how 'pure' a colour is. You reduce



test_pattern.py

> Language: **CircuitPython**

```python
import time
import board
import neopixel
import adafruit_fancyled.adafruit_fancyled as fancy
import math
import random

#change based on your hardware
pixel_pin = board.GP0
num_pixels = 100

#change to
analogous = True # set to False for a balanced palette

#Palette settings
root=0.01
num_colours = 6
saturation = 0.8
variation = 0.05


#movement settings
steps=5
sleep_time = 0.01

#brightness settings
new_colour_prob = 2

pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.3, auto_write=False)
colours = []

def build_analogous_palette(root, number, saturation = 1.0, value = 1.0, variation=0.1):
    colours = []
    iseven = True
    if number // 2 == 1:
        iseven = False
        colours.append(fancy.CHSV(root, saturation, value))
    for i in range(number // 2):
        if iseven:
            colours.append(fancy.CHSV(root+(variation/2)+variation*i, saturation, value))
            colours.append(fancy.CHSV(root-(variation/2)-variation*i, saturation, value))
        else:
            colours.append(fancy.CHSV(root+variation*(i+1), saturation, value))
            colours.append(fancy.CHSV(root-variation*(i+1), saturation, value))
    return colours

def build_balanced_palette(root, number, saturation = 1.0, value=0.5):
    colours = []

    for i in range(number):
        colours.append(fancy.CHSV(root+i*(1/number), saturation, value))

    return colours

if analogous == True:
    colours = build_analogous_palette(root, num_colours, saturation=saturation, variation=variation)
else:
    colours = build_balanced_palette(root, num_colours, saturation=saturation)

```



DOWNLOAD THE FULL CODE:

rpimag.co/github

```python
hsv_pixels = [fancy.CHSV(0.0, 0.0, 0.0)]*num_pixels

while True:
    pixel = random.randint(0,num_pixels-1)
    if random.randint(0,new_colour_prob) == 0:
        new_colour = colours[random.randint(0,num_colours-1)]
        black = False
    else:
        black = True
        new_colour = fancy.CHSV(hsv_pixels[pixel].hue, hsv_pixels[pixel].saturation,0.0)

    for i in range(steps):
        #need to deal with the previous colour being black
        if hsv_pixels[pixel].saturation == 0 :
            #set the hue and saturation now?
            hsv_pixels[pixel].hue = new_colour.hue
            hsv_pixels[pixel].saturation = new_colour.saturation
        pixels[pixel] = fancy.mix(hsv_pixels[pixel], new_colour, i/steps).pack()
        if black:
            hsv_pixels[pixel] = fancy.CHSV(0.0,0.0,0.0)
        pixels.show()
        time.sleep(sleep_time)

    hsv_pixels[pixel] = new_colour
    pixels[pixel] = hsv_pixels[pixel].pack()
    pixels.show()
```

saturation by mixing in white. If we're defining our palette by hues spaced around the colour wheel, we usually want to do it so they all have the same saturation.

There is also a slightly unusual take on this that can work: infinite range colour palette. If you pick hues at random, but always have the same saturation, you can get a balanced palette. This works best at lower saturations, otherwise it can get a bit too garish.

You don't have to balance your colours like this. Another option is to use a string of colours that are close to each other on the colour wheel. This is known as an analogous palette.

There are many ways of building a palette that you can use, and we've only looked at a few here. They're ones we like to use, but that's partly down to personal preference. We should end this by saying that there is no perfect theory of colour (despite there being many things called a theory of colour). There are lots of ideas – like the ones we've given above – many of which have a track record of looking great. However, these are just

> There are many ways of building a palette

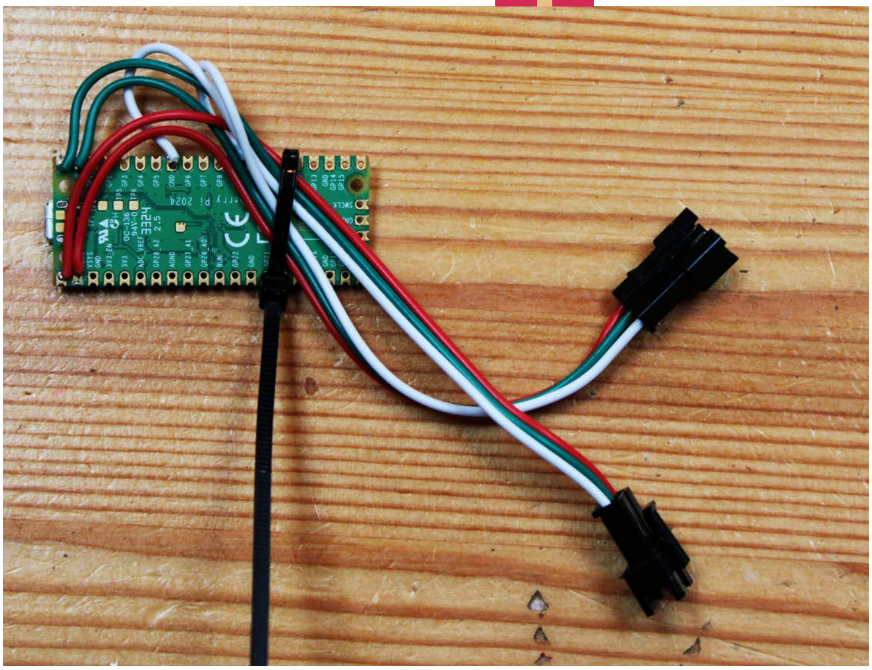
things that people have found that tend to look good. If you prefer something that doesn't fit into anyone else's theory of colour, then it doesn't matter. They're your lights and what you like matters more than what some theory says.

Perhaps the key thing that we'd like you to take away from this is that there are methods for building palettes to create different effects in the pattern, and if you build your pattern with a changeable palette, it lets you experiment and find one that works for you.

Theory into practice

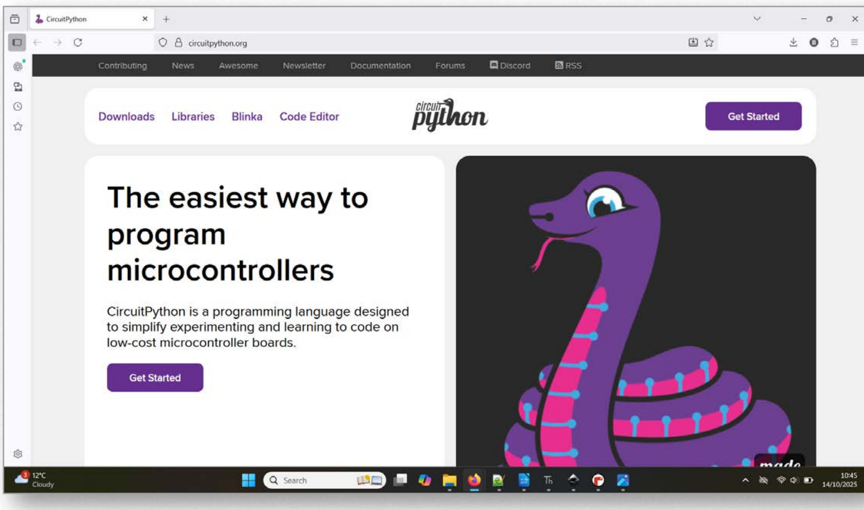
Let's now take a look at all this in action. Last month we looked at the range of options for connecting LEDs to Pico, so we won't go into it in much detail again. We've connected the string to GPIO 0, but that's easy to change if you want to connect it in a different way.

We've written this in CircuitPython, so you'll need to install that. You'll also need the neopixel and adafruit_fancyled



Our NeoPixel Pico has two connectors (though we're only using one on this project). The cable tie helps relieve strain on the soldered connections

modules, both of which are in the CircuitPython Library Bundle. You can get all of this from **circuitpython.org**. You'll also need a computer with the Mu code editor installed, which you can download from **codewith.mu**. If you've not used CircuitPython before, we'd recommend having a quick look at the beginner guide so you know how it all works: **rpimag.co/cpoverview**.



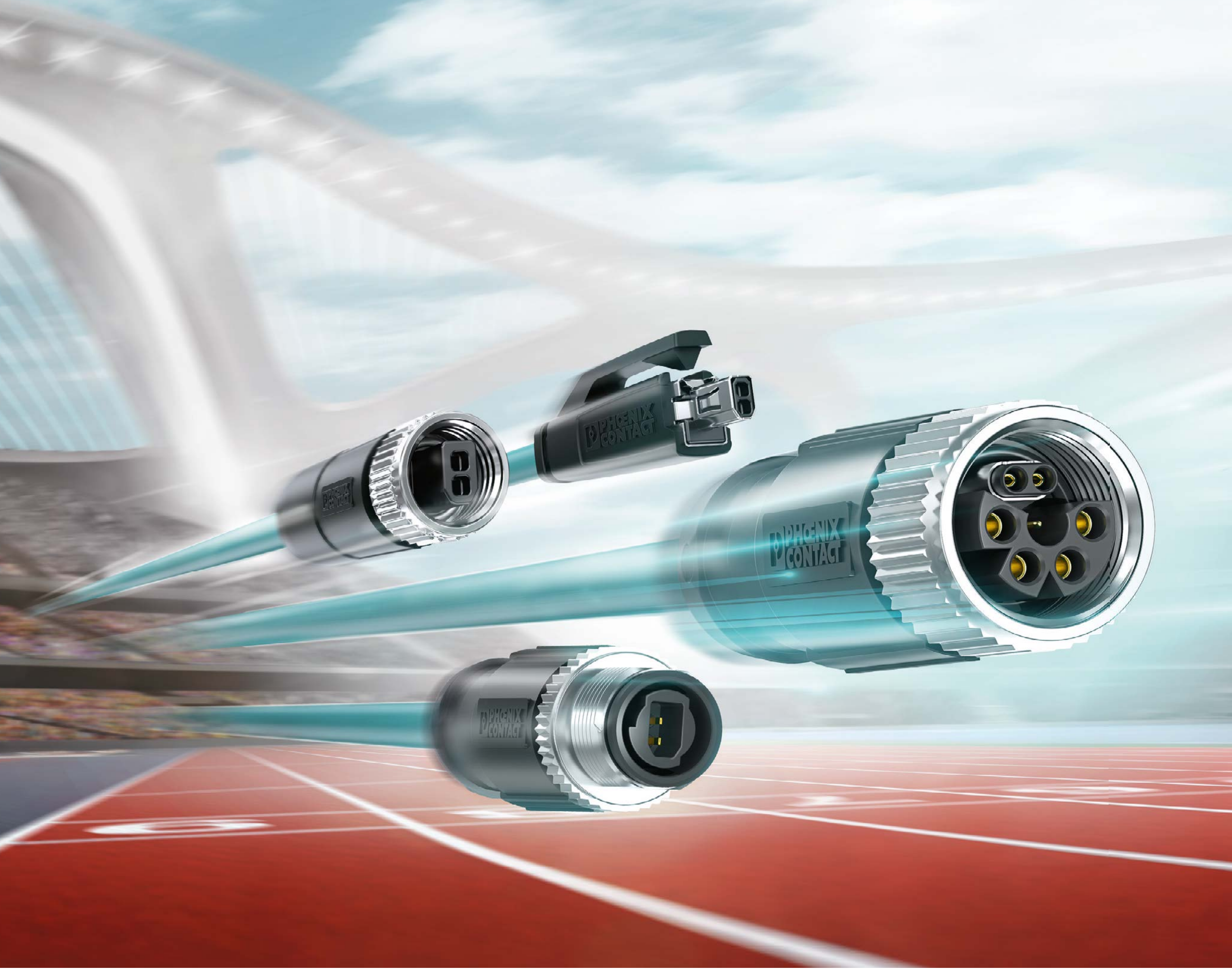
CircuitPython is really well documented. If you need any help, head to **circuitpython.org**

We've devised a test pattern designed to let you play with all of these variables to see how they affect the look of the pattern – see the **test_pattern.py** code listing. Once you've figured out where your preferences for the balances of movement, colour, and brightness are, then you're in a great place to design your own pattern.

This will build a palette in two different ways. Either a balanced palette (evenly spaced around the colour wheel) or an analogous one (clustered around one point). These aren't the only ways of building a palette, and you could write your own function to build a palette, or create one by hand and paste in the values if you like). You can select the number of colours, the root colour, and (in the case of analogous) the variance (i.e. the distance between adjacent colours in your palette).

You can also adjust the movement in the pattern in two ways: the number of steps (i.e. the number of brightness shades between fully off and fully on) and the sleep time. Increase either or both numbers for slower and smoother animations. Decrease them for faster, more blink-y animations. There is also the `new_colour_prob` variable that changes both the movement and the brightness. In each loop, it picks a pixel at random and either lights it in a new colour or turns it off. Increasing this number increases the chances of the pixel going black. A higher number means a darker pattern – but within that darker pattern, there's more scope for movement.

Exploring the various options on this pattern should give you an idea of the sort of things you like to see in patterns, and that will put you in a really good place to build your own.



Continuous data transmission finally goes the distance!

Discover the advantages of the ONEPAIR series

Experience seamless and efficient data transmission with the ONEPAIR series for Single Pair Ethernet (SPE). The compact device and cable connectors are designed to meet the highest standards, ensuring reliable performance in factory, process, and building automation applications.

Upgrade to the ONEPAIR series and take your data transmission to the next level. Future-proof your network today!

For additional information, scan the QR code

PHOENIX CONTACT

Make and publish your own zine

Get into on- and offline DIY publishing with Inkscape

Maker

K.G. Orphanides

K.G. has been making digital and print zines for fun and (remarkably little) profit since the 1990s.

twoot.space/@owlbear

Zines are DIY magazines that you can make yourself, or with friends. They can be anything from a single folded page of A4, to a photocopied magazine, or an ornate hand-constructed work of papercraft art. We used Inkscape to create our zine, and it's our preferred choice. There are a few different ways to get hold of this popular open-source vector illustration program, which tidily serves double duty as a flexible DTP program, but only the Flatpak version is currently updated to the latest version 1.4.2, which we've been using and has all the features and templates we need to make this task easy.

Installing Inkscape

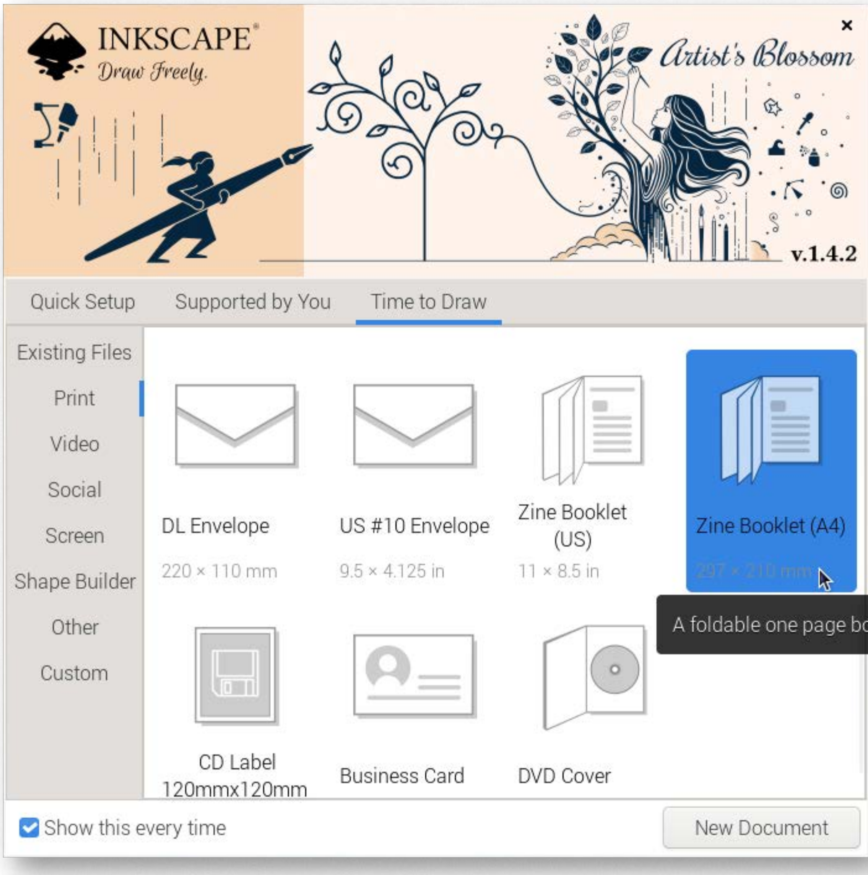
We prefer to install Inkscape from the Flathub repository (**rpimag.co/flathubinkscape**), which provides a convenient way to install the latest stable release and keep it up to date.

To install the Flatpak version, open a terminal and type:

```
 $ flatpak remote-add --user --if-not-exists
flathub https://flathub.org/repo/flathub.
flatpakrepo
 $ flatpak install --user flathub org.inkscape.
Inkscape
```

You can then run Inkscape directly from the Graphics section of Raspberry Pi OS's application menu, or from the terminal by typing:

```
 $ flatpak run org.inkscape.Inkscape
```



Inkscape 1.4.2 has a built-in template for foldable mini zines

Warning!

Sharp blades

This project can involve the use of cutting blades or scissors. Always use sharps responsibly. Children should be supervised if blades are used.

Formatting your zine

A zine can look however you want it to, from a broadsheet, to chapbooks or classic folded mini zines.

We've distributed [zines] at art galleries, protests, pubs, on the Internet Archive...

If you're publishing for online reading, then the easiest way is by setting up A4 pages and exporting them as a PDF. Readers can then easily download it and flip through its pages using their favourite PDF or e-book reader, and you can upload it for easy online reading using an embedded reader with realistic page flip mechanics.

If you'd rather make a tiny zine, recent versions of Inkscape come with a template that makes it easier to produce a folded zine from a single sheet of A4, which you'll find under the Print section of the Time to Draw tab in Inkscape's startup interface. These look great, save paper, and are great for giving to friends and putting out at public events.

Of course, the sky – and your imagination – is the limit, not only when it comes to form factor, but also design.

Zines are often as much about your DIY artistic instincts as they are about the content. Whether you're using clip art, making collages, or illustrating your own comic strips, you can give your zines tons of personality. Think about what you find engaging on the page, rather than making something look professional and polished. If unevenly hand-defined text flow boxes and wonky images and photos stacked on top of each other bring you joy, then there's no reason to seek 'professional' polish.

Layout and navigation

If you're making an A4 zine, particularly if you plan to distribute it on the web, then you'll appreciate one of Inkscape's most useful innovations: the multi-page SVG file. While you can only guarantee that current versions of Inkscape will handle these files without first exporting them to PDF, it's a great format to work with, allowing you to easily see, add, and flow content between multiple pages.

The page tool is the bottom icon on the left-hand toolbar. Select it, and another icon, a page with a plus sign on it, will appear above the left-hand toolbar. Click this to add pages to your document. To the right of that, you'll find options that allow you to define page sizes, add, delete, reorder, and extend a selection to include the entire content of pages. Clicking on a page will select it, and you can delete a page from the toolbar or by pressing **DELETE**.

You can move around the workspace, no matter how many pages you're working with, by holding **CTRL** and using the arrow keys, or by scrolling with the mouse wheel (with **SHIFT** to pan horizontally). Hold **CTRL** and scroll the mouse wheel to zoom in and out. There are multiple keyboard and mouse shortcuts for every tool in Inkscape, and you'll find them all listed at **rpimag.co/inkskeys**.

A critical feature of any DTP tool – and that's what we're using Inkscape as right now – is the ability to flow text between pages so it automatically adjusts when you edit your words. Make text flow into boxes by drawing one or more rectangles and pasting your text in a text box. Now use **CTRL** + left click to select all your rectangles and your text box, and either press **ALT+W** or go to the Text menu and select the 'Flow into frame' option. If you're working with text that you've previously flowed and want to put it into different frames, you may need to select it and use the 'Unflow' or 'Convert to text' options from the Text menu before reflowing it.

In smaller font sizes, called for in mini-zines in particular, you might find that there's too much space between each line of text. To change the line spacing, select the text tool (the icon in the left-hand toolbar that looks like a letter A followed by a caret), select the text block you need

Free images

Creative Commons Media Search:
rpimag.co/ccmedia

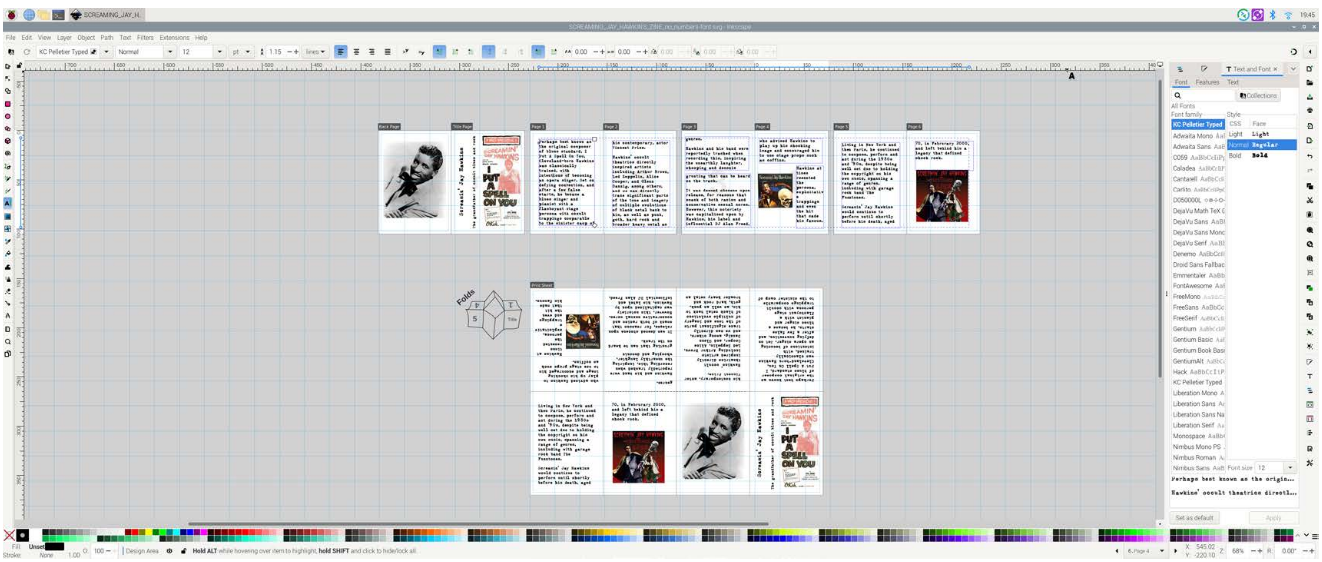
National Portrait Gallery: **rpimag.co/npg**

New Old Stock: **nos.twnsnd.co**
Vintage photos in the public domain.

Smithsonian Open Access:
rpimag.co/smithsonian
Many other museums, state-funded scientific organisations and art galleries have released images under Creative Commons licenses.

Wikimedia Commons:
rpimag.co/wikicommons

Understanding Creative Commons licences
rpimag.co/cclicences

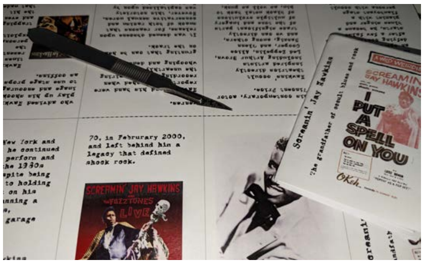


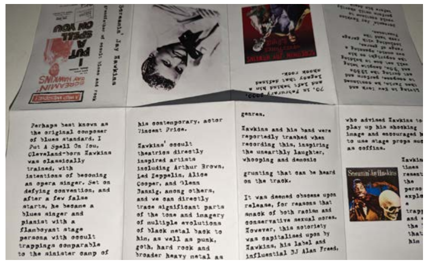
▼ Inkscape was designed for vector graphics, but like Corel Draw before it, it makes a great desktop publishing tool to adjust, and then look towards the centre of the tool ribbon above your editing area, where you'll be able to select your font, its style, size, and then spacing between lines.

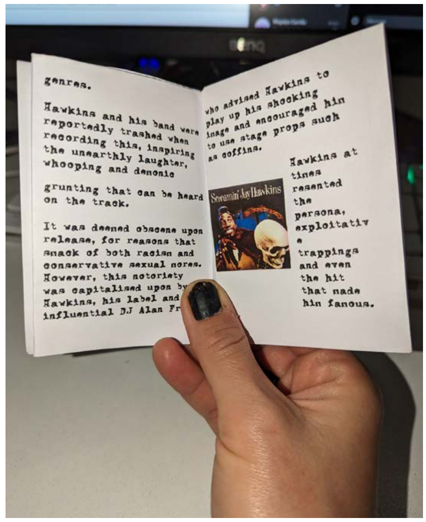
Once you've aligned your text where you want it, you'll probably wish to dispense with the coloured rectangles you used to lay it out. To render these invisible, select them. The easiest way to do this in a busy document is to go to the right-hand tool bar where you'll also find your text editing tools, select the Layers and Objects tab and **CTRL**-select all your rect objects. Now select the Fill and Stroke tab, and the Fill tab below it, and then click the 'No paint' option.

To add graphics, you can just drag PNG or JPEG images into your SVG document, and embed them, which means that they're imported from the external files. Keep the files together with your SVG to avoid losing them. You can also import your images into the SVG if you don't mind increasing its file size. You'll probably need to resize them: the easiest way to do this is to select the image, look at the tool ribbon above the working area, click the lock icon between the W and H measurements, switch their unit to percent, and adjust your image size down that way. You can also resize to millimetres or pixels if you need something more precise.

To export a multi-page A4 zine for easy sharing, save it as a PDF. If you're using Inkscape's one-page zine template, you should move its watermarks – the numbers indicating how you should fold it – to the 'Guides (do not print)' level of the Layers and Objects tab and use the Print (**CTRL**+**P**) dialogue to either send its Print Sheet to your printer or print to file to produce a PDF you can use elsewhere. Once printed, cut along the dotted line and fold it as shown.







▲ Fold and cut and refold your mini zine per the template, and share it with your friends

Publish and be damned

Now that you've created your zine, where to publish it? We've distributed ours at art galleries, protests, pubs, on the Internet Archive (**archive.org**), and on strange little 2000s throwback web pages.

If you want to sell your zine, then Itch (**itch.io**) is your best bet. Creating an account to distribute zines (or games, music, or almost anything else) for free is quick and easy. If you want to accept donations or sell your zines, setting up the kind of account that can sell digital downloads – without having to handle your own VAT payments – isn't too much harder. Note that, due to a tightening of restrictions by payment processors Visa and Mastercard, Itch can no longer make payouts on some adult content.

If you want to sell anything on Itch, you'll have to provide a Taxpayer Identification Number (TIN) and fill out a US withholding tax form for sales to countries under their local VAT systems, such as VAT MOSS in the EU. The fees for this will be taken from your first payment, and the form will be processed when you request your first Itch payout, which means that this can take longer to be processed than subsequent payouts – we had to wait nearly a month for our initial payout.

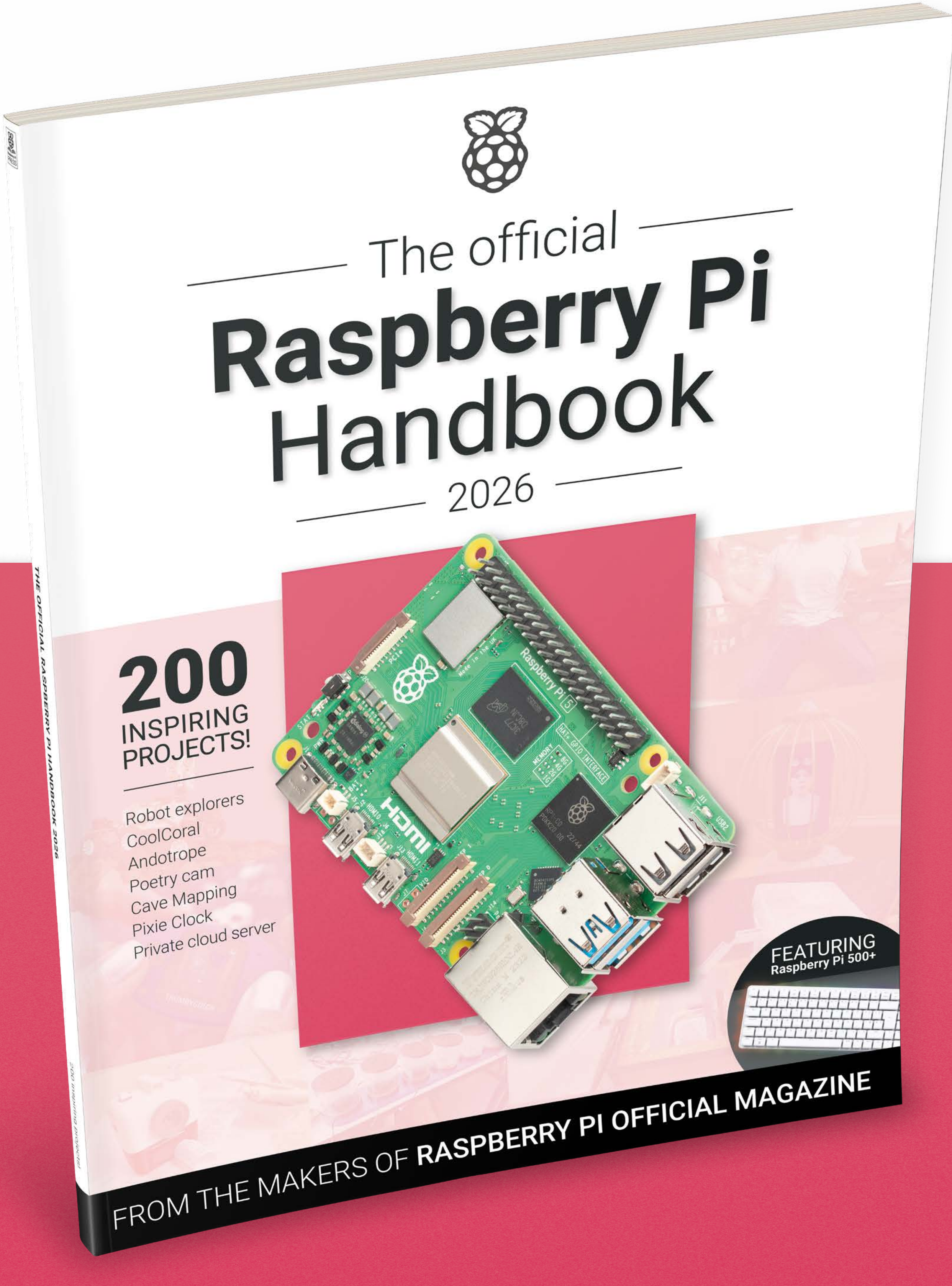
Free fonts

Fontsource: **fontsource.org**

Kernclub: **rpimag.co/kernclub**
Rotating selection of fonts available for $0.00

Uncut: **uncut.wtf**

Velvetyne: **velvetyne.fr**



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

Train a rock, paper, scissors AI model

Teach your Raspberry Pi and AI Camera to recognise simple hand gestures

Maker

Lucy Hattersley

Lucy is editor of *Raspberry Pi Official Magazine* and loves a game of rock, paper, scissors.

rpimag.co

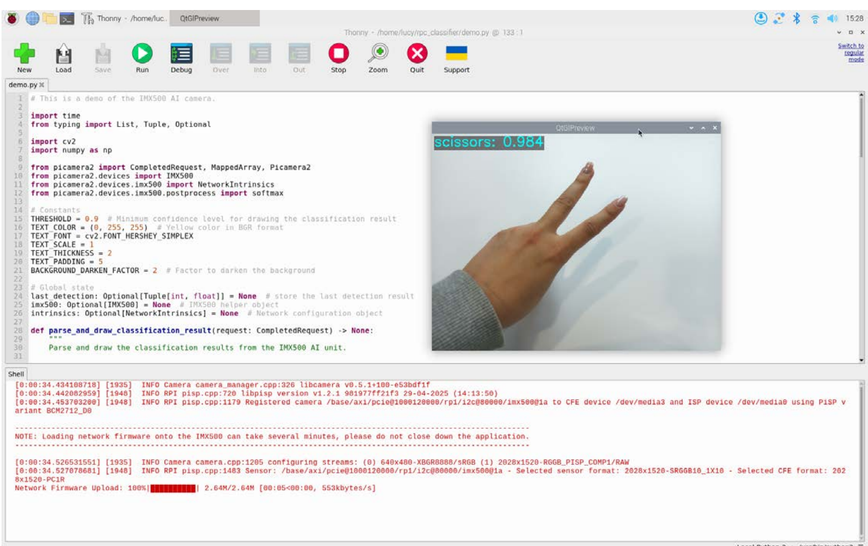
At the heart of each AI system is its model. This is the engine that makes the AI work. There are different types of models for different tasks. Convolutional Neural Networks (CNN) are great for images; Large Language Models (LLMs) are great for handling text, and Reinforcement Learning (RL) models are great for environmental interaction, such as teaching a robot to walk.

Typically, when using AI in a project you will use a model that has been pretrained by somebody else. A language model developed by OpenAI or an image classifier developed by Sony, for example. Or you can take a model and adapt it to your needs.

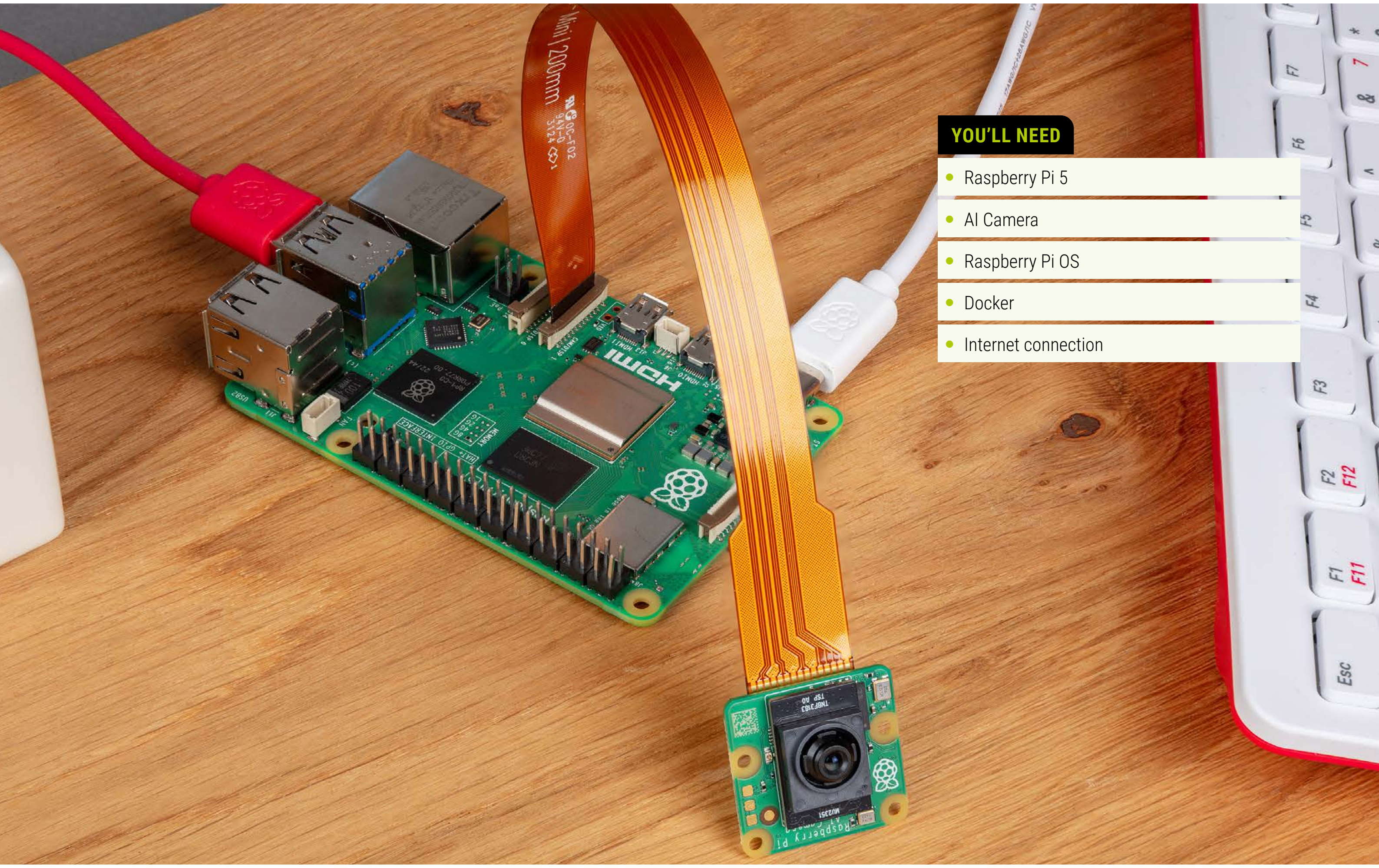
In this tutorial, however, we are going to train our own model using images and TensorFlow code. We will then compress the model using a process called quantization. As a bonus extra, you will convert this model into a package that can be used by Raspberry Pi AI Camera.

Loading the package file onto Raspberry Pi AI Camera enables us to play a game of rock, paper, scissors (rps) against our computer. Our rps model is incredibly lightweight, but fully functional. On an educational level, it is a great way to understand how the AI models you are using are created.

Our trained model is practical, too. You can use this code to create custom detection models for all kinds of industrial and real-world purposes. You could train it to detect faulty/working items on a production line, or stock levels on shelves; weeds vs crops in a field. Or just about anything where you need a camera to look at something and answer "is it this, or this, or this?"



A program in Thonny that detects rock, paper, or scissors hand shapes



YOU'LL NEED

- Raspberry Pi 5
- AI Camera
- Raspberry Pi OS
- Docker
- Internet connection

Let's work through the educational process first. This tutorial helps us differentiate between the two key stages of an AI model:

- **Training.** This is the computational heavy lifting where you adjust the parameters in a model until it can reliably map inputs (images of hands, in this instance) to outputs (labels of 'rock', 'paper', or 'scissors' depending on the shape of the fingers).
- **Inference.** This is where you deploy the model to make predictions or decisions. Computationally it's much lighter than training and inference is easily run locally on smaller devices, such as Raspberry Pi.

Despite us training on a Raspberry Pi, we are leaning on the shoulders of giants, in this case the mighty MobileNet architecture. Rather than train all of the thousands of layers at once, we slice off the final categorisation layer and train this layer.

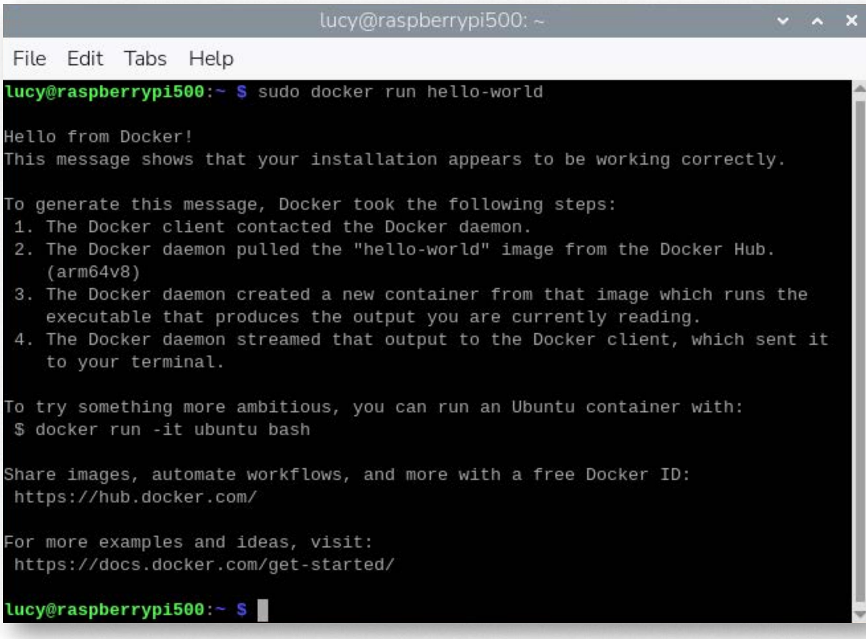
Thanks to this clever approach, it is perfectly possible to train a model like this on Raspberry Pi (it helps to have a 16GB model).

It is still a bit time consuming. Expect to spend around 30 minutes waiting for the model to train on a Raspberry Pi 5 with 16GB of RAM. It took us less than five minutes on an x86_64 computer with an NVIDIA graphics card. If you're lucky enough to have access to both we have provided a separate GitHub branch for you to test each out.

Colab

Sony has also provided a version of this code in Google Colab. This virtual environment enables you to bypass Docker and run the whole program in a web interface with GPU support. Be sure to change the runtime version to 2025.07.

rpimag.co/mobilenetcolab



▶ Docker successfully installed

Hardware requirements

For this tutorial, you will need a Raspberry Pi computer to train the model and an AI Camera to run the converted package. The AI Camera requires any Raspberry Pi with a CSI (Camera Serial Interface) connector. Because the inference is being handled at a hardware level on AI Camera, it works with good results even on a Raspberry Pi 3B or Raspberry Pi Zero 2 W. Using later models allows higher resolution streaming at higher frame rates.

Our program has complex software requirements and runs in a Jupyter Notebook environment running inside a Docker container.

Jupyter Notebook

Project Jupyter is a web-based interactive environment for data science and machine learning. It enables developers to combine Python code with Markdown cells. This combines code with detailed documentation, which explains the code and makes for a powerful educational environment.

Code in Jupyter Notebook is split into cells. These can contain Python code, shell commands, images, audio, video, and even interactive widgets and plots. It's a powerful way to combine code with descriptive material that makes it tremendously useful for explaining how code works.

At its heart, though, is Python code and you can train the model with just Python and the appropriate frameworks.

Docker vs Venv

Docker is similar to Venv, but more powerful. While building a virtual environment enables you to import specific packages, Docker uses OS-level virtualisation that shares the kernel but operates in an isolated environment. This is faster than Venv and more resource-efficient.

Docker

Docker is an open-source platform that enables developers to build containerized applications. Code is built on specific versions of packages, running on specific operating systems.

Our Jupyter Notebook requires specific packages:

- TensorFlow 2.14.0
- Model Compression Toolkit 2.2.0
- IMX500 Converter
- NumPy 1.26.4
- Java 17

These are all installed in the Dockerfile, allowing us to concentrate on running the code.

Install Docker

Start by installing Docker in Raspberry Pi OS. Open a terminal window and enter:

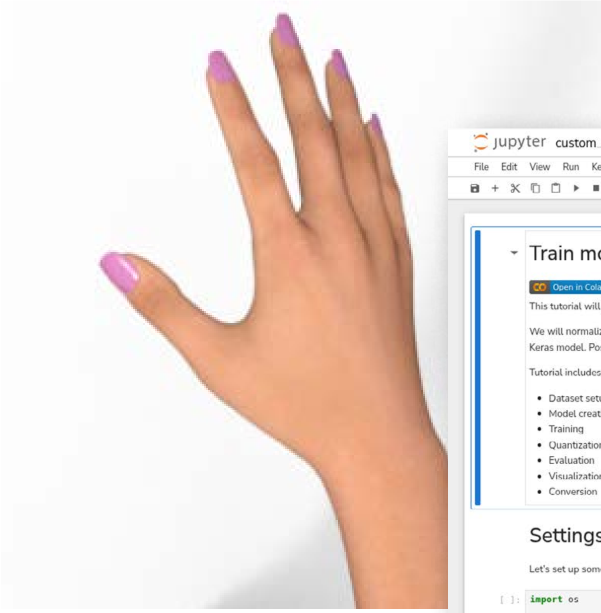
```
$ curl -sSL https://get.docker.com | sh
```

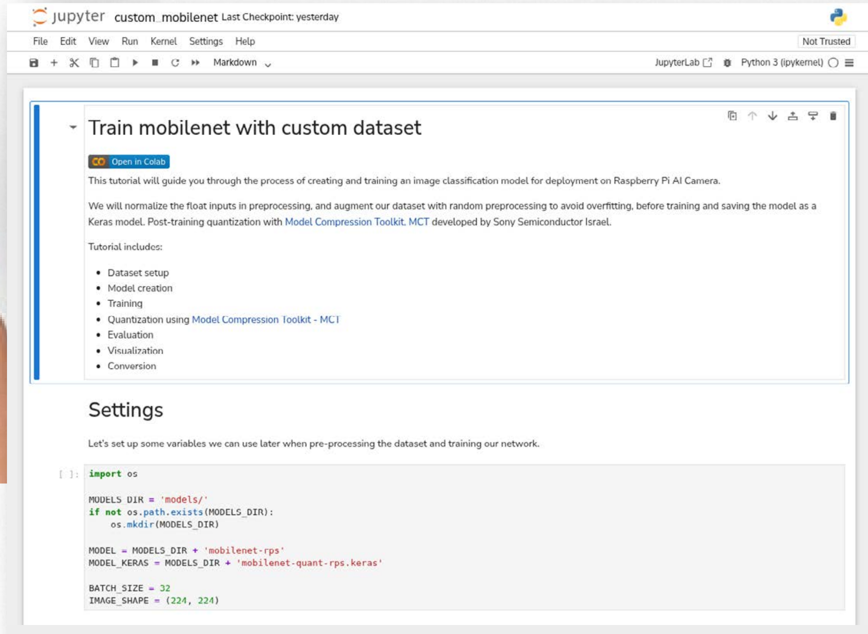
Test Docker is working with the hello-world script:

```
$ sudo docker run hello-world
```

You should see the following message:

```
 Hello from Docker!
 This message shows that your installation
appears to be working correctly.
```





Clone the GitHub repository

Next, we need to clone the code from the GitHub repo:

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

Navigate into directory:

```
$ cd custom_mobilenet
```

And take a look at the **README.md** file:

```
$ cat README.md
```

Run the Jupyter Notebook instance

Now that we have everything installed, we should `cd` into the correct folder and run the Jupyter Notebook in a local instance:

```
$ cd notebooks/mobilenet-rps
$ sudo make jupyter-local
```

Wait for Docker to create the container and start the Jupyter Notebook. Open a web browser and enter the following URL to access the notebook:

```
http://localhost:8888
```

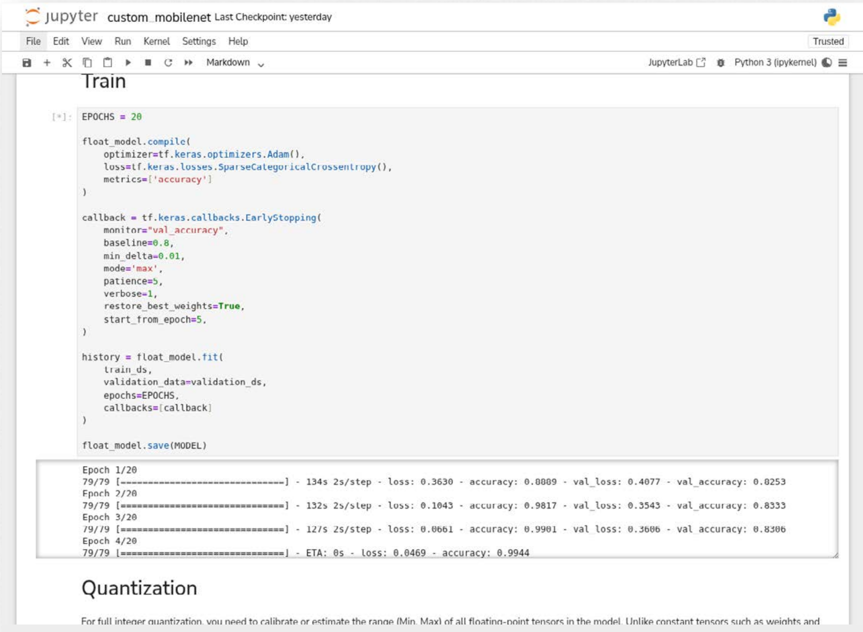
You will be presented with a web interface displaying the files in the container. Double-click **custom_mobilenet.ipynb** to open the notebook. You should see a code file titled **Train mobilenet with custom dataset**.

Take a look at the code

The code is split into several stages. One of the handy things about Jupyter Notebooks is that code and text are placed into cells, known as Code blocks and Text blocks.

The code contains the following cells:

- **Settings**. Initial setup determining some constants and file locations.
- **Dataset**. We will use the RPS (rock, paper, scissors) dataset from TensorFlow datasets. This comes in two parts: a training set and testing test.
- **Pre-processing and augmentation**. Images are sent through in batches. Here we set the range, adjust the image sizes, and slightly adjust each image to improve the performance.
- **Keras Mobilenetv2 model for transfer learning**. We bring in the MobileNetV2 model and freeze its base layers. This saves us from training an entire model and enables us to concentrate on the head (the final classification layer).
- **Train**. This is the computationally intensive process of training a model on the dataset. We go through the data up to 20 times, with each pass known as an 'epoch'.
- **Quantization**. This is a process of optimisation that maps the larger values into a smaller set to reduce the complexity. This enables the trained image to run on



▲ The code running the training process

lighter hardware such as the IMX500 camera inside Raspberry Pi AI Camera. It sacrifices a small amount of accuracy for substantial speed gains.

- **Evaluation**. Here we check the accuracy of the model.
- **Visualize detections**. This section of code displays a selection of test images and inference results.
- **Conversion**. Output the quantized Keras model to a zip file. We can then convert this into a **network.rpk** file for use on AI Camera.
- **Next step**. The final cell gives us information on saving the **packerOut.zip** file and how to package it for the IMX500 sensor inside Raspberry Pi AI Camera.

The code is somewhat lengthy and, like most things in AI-world, quite gnarly on first pass. But it is tremendously valuable to learn how this process works, both for the educational value and for the ability to train models on custom datasets with real-world implications.

Run the code

Now we've had a brief overview of the code, we can start running it. You can run each cell individually by highlighting it and pressing **SHIFT+ENTER**. Next to each cell is a marker displayed with square brackets: **[ ]**. As the code runs, it will appear with an asterisk inside [*****] and when it is finished, it will display a number **[1]**. This is the code execution prompt and will let you know which code is running, and what ran in which order.

You can run all the code by selecting Run and Run All Cells from the Jupyter Notebook menu. We prefer to go through it one cell at a time and read the notes as we go.

Conversion

Our final cell is a relatively simple conversion and output to a **packerOut.zip** file.

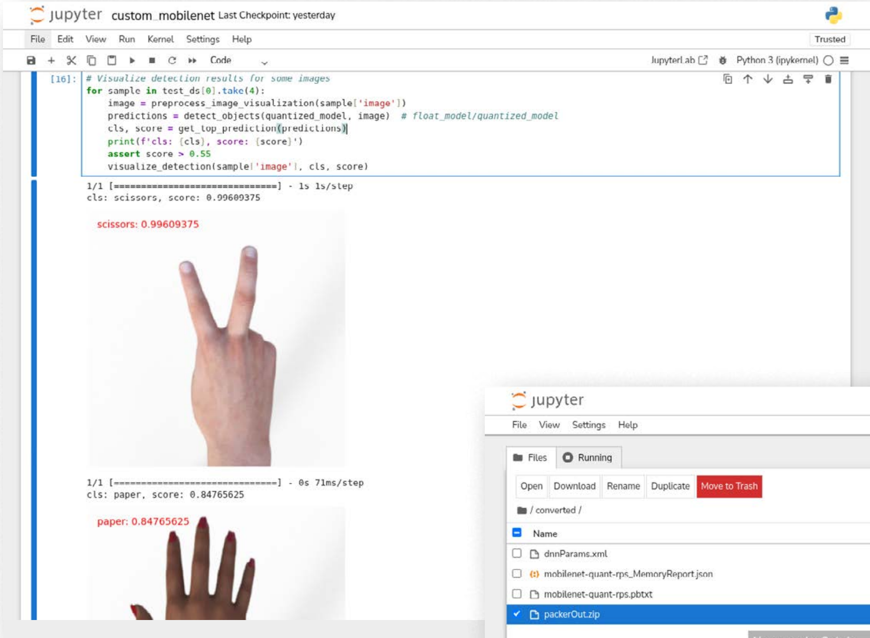
```
!imxconv-tf -i {MODEL_KERAS} -o converted
```

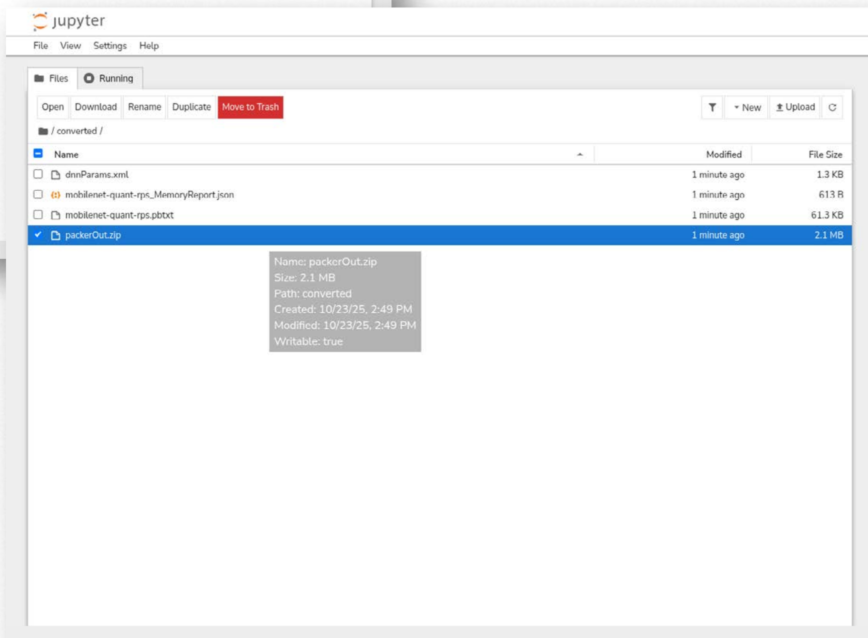
The `!` at the start lets us know that this is a system shell command, instead of the Python we have been using to train, test, and evaluate our data. The `imxconv-tf` program was installed inside the Docker installation. It takes the Keras model we saved earlier and produces a **packerOut.zip** file which is compatible with our IMX500.

Finally, we use `!ls converted` to check all the files in our converted folder and an `assert` command to check that the **packerOut.zip** file is present.

Get the file

When the code has finished running, grab the **packerOut.zip** file. Click View and File Browser to open a new web page displaying the file directory.





The File Browser enables you to download files to your computer

Click the converted folder, right-click **packerOut.zip** and choose Download. You will then be able to find the file in your **Downloads** folder.

Package the file

The final step packages the model into an RPK file. When running the neural network model, we'll upload this file to AI Camera.

You do this on Raspberry Pi in terminal with imx500-tools. Install with this command:

```
$ sudo apt install imx500-tools
```

To package the model into an RPK file, you need to run the following command:

```
$ imx500-package -i <path to packerOut.zip> -o
<output folder>
```

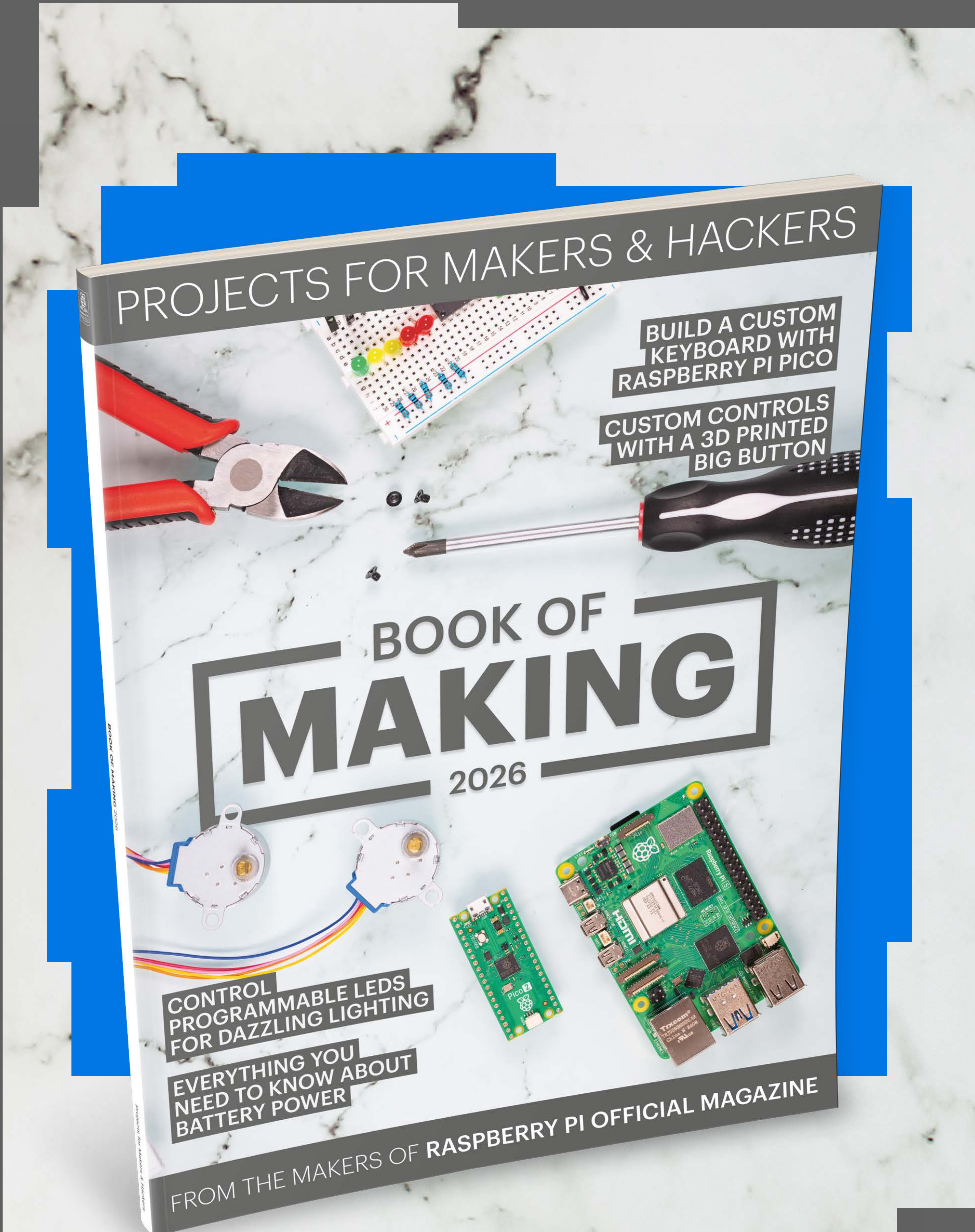
This command should create a file named **network.rpk** in the **output** folder. You'll pass the name of this file to your IMX500 camera applications.

In the next tutorial, we shall set up a code review to go over each cell and explain what is going on. After that we are going to show you how to load the **network.rpk** into Raspberry Pi AI Camera and customise the code. Perhaps we'll even build a robot to play against. Join us on this AI model training journey.

IMX500 Packager

Raspberry Pi's documentation covers packaging files for Raspberry Pi AI Camera: **rpimag.co/aicampack**.

Sony's AITRIOS website has an IMX500 Packager User Manual that outlines the packaging process in more detail: **rpimag.co/imx500pack**.


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

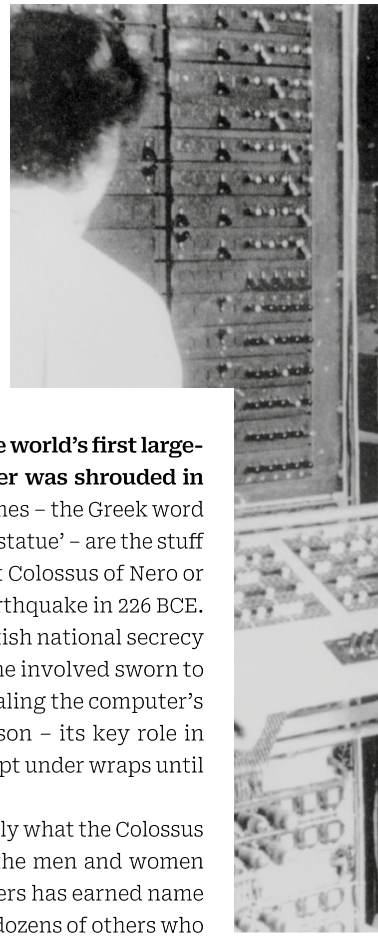
STEP INTO THE WORLD OF MAKING!

- 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

Colossus

“Like magic and science combined!” – Joanna Chorley, Wren



Author

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

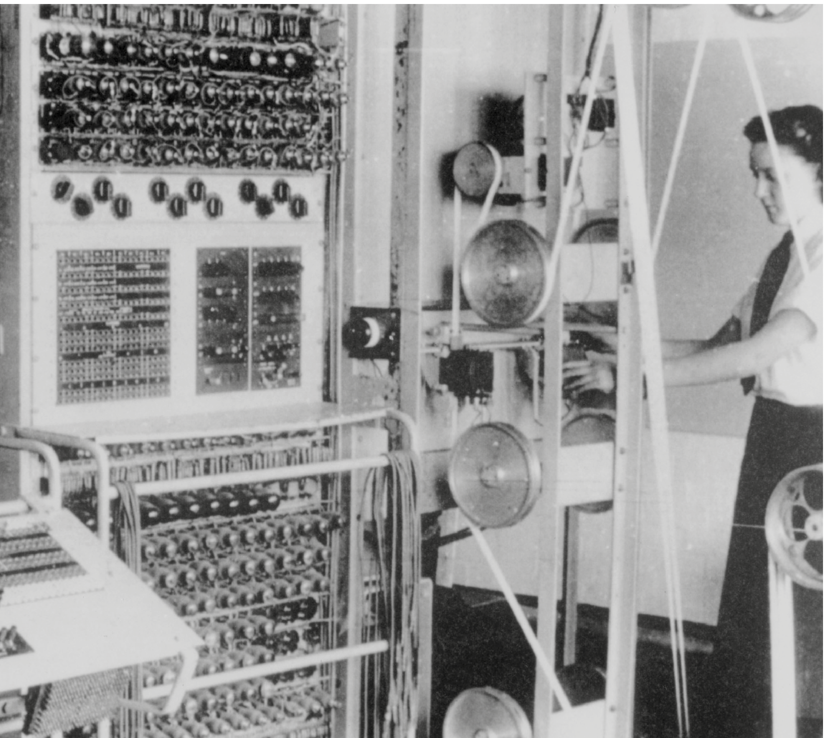
Perhaps it is poetic justice that the world’s first large-scale electronic digital computer was shrouded in secrecy. The Colossi of ancient times – the Greek word κολοσσός (kolossós) literally means ‘gigantic statue’ – are the stuff of myth and legend, whether that’s the lost Colossus of Nero or the Colossus of Rhodes destroyed by an earthquake in 226 BCE.

The Colossus in this story fell foul of British national secrecy rather than natural elements. With everyone involved sworn to silence under the Official Secrets Act – revealing the computer’s existence would have been an act of treason – its key role in breaking Nazi High Command codes was kept under wraps until the 21st century.

Even today, few people understand exactly what the Colossus did, how it worked, or the role played by the men and women behind the machine. Perhaps Tommy Flowers has earned name recognition – and rightly so – but there are dozens of others who were key to its success, whether that’s code breakers such as Bill Tutte and Max Newman, or electrical engineers such as Sid Broadhurst and Bill Chandler.

Then there are the hundreds of unsung heroes whose names are rarely recorded in history books. The dozens of people who worked under Tommy Flowers, executing his plans. The factory workers who built the Colossus – more accurately, the Colossi, as ten were in operation by the end of the war. The hundreds of Wrens (the women’s branch of the Royal Navy) and their army counterparts, or the radio signal interceptors who painstakingly recorded the German transmissions. As they had to: one mistake could render the whole message useless.

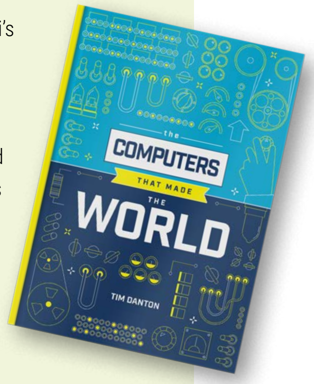
There is also one giant hero, which is Britain’s wartime General Post Office. It was the GPO rather than Bletchley Park that took the risk to build the first Colossus, and this insufficiently honoured organisation built the infrastructure on which all wartime communications relied.



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



▲ A Colossus Mk 2, 1943. The slanted control panel is on the left, the 'bedstead' paper tape transport on the right
Image: The National Archives (UK), Public Domain

Desperate to modernise, the Post Office needed engineers and they needed them fast

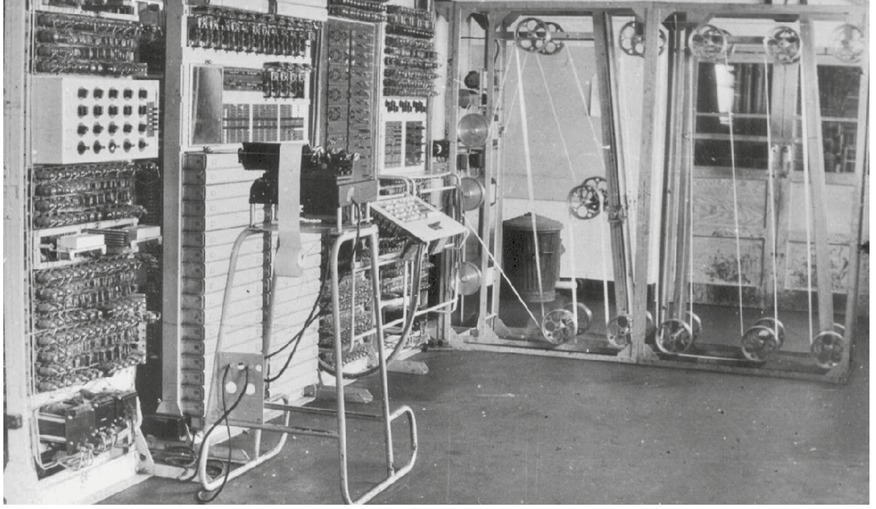
But let's start the story of the Colossus on 1 September 1939. Tommy Flowers was then a 33-year-old employee of the Post Office's research department and had just arrived in Berlin, of all places, to take part in a European conference about telephone systems. Germany invaded Poland the same day.

Flowers and a fellow British delegate went to the British embassy to let them know where they were staying. "The young man at the embassy … looked at us and said, 'You people must be mad, there's going to be a war in a few days'," Flowers said in a far-reaching interview in 1998.[1] "Which surprised us a bit. But we couldn't go back."

The British duo spent the next day with their German counterparts on the committee, trying to play it cool, but the following morning the embassy called to say they had to catch the next train out or else. Flowers said of the tense eight-hour journey through Germany: "We hardly saw anything, there was hardly anyone on the stations, but when we finally got to Holland, which was after dark, everywhere was in a blaze of light. The Dutch army was mobilising, the same in Belgium."

After catching the boat train across to England, they arrived at Liverpool Street station at 8am on 3 September 1939. Flowers sensibly exchanged his ten days' worth of German Marks into Pounds at the first opportunity, while his fellow traveller decided to wait until 11am. At which point the UK had severed diplomatic ties with Germany and the banks stopped taking its currency. "He never got his money back," said Flowers.

1 Thomas Harold Flowers interview with Peter Hart, 18 May 1998, Imperial War Museum Collections, **rpimag.co/thfinterview**. All quotes from Tommy Flowers are taken from this interview unless otherwise stated.



▲ The tenth Colossus computer with its extended bedstead in Block H at Bletchley Park in 1945
Image: UK Public Record Office, Public Domain

Still, it had been a lucky escape for both men. Many British citizens unfortunate enough to be in Germany on the day Britain declared war were detained at internment camps until VE Day in May 1945. It was an even luckier escape for the British war effort, because without Tommy Flowers there would have been no Colossus – and without Colossus the Allies' path to Germany would have cost many more lives, on both sides.

To understand why Tommy Flowers was so intrinsic to Colossus, we now need to go back to 1922 when he left school at the age of 16. He had been a star pupil and "was determined to be an engineer", just like two of his uncles; their advice was to "go through the workshops" rather than into further study. So he took up an apprenticeship at the Royal Arsenal in Woolwich, but also enrolled in engineering evening courses at Woolwich Polytechnic[2] (now the University of Greenwich).

In 1926, by which point Flowers was still only 20, the General Post Office[3] decided to accelerate its move to automatic telephone exchanges. While automatic exchanges were by no means a new concept, and had been rolled out in several British cities, many calls still relied on physical connections being made by operators sitting at a switchboard. When a call came through, they had to manually connect callers with cords.

Desperate to modernise, the Post Office needed engineers and they needed them fast. So they invited school leavers to take an examination, with the best performers invited to join the scheme. And Tommy Flowers, it turns out, was the best of the lot, beating hundreds of others to top the exam scores.

Within three months of starting, Flowers said he was "yanked out of" general training and sent to the GPO's Circuit Laboratory. It was Flowers's job – along with everyone else in the lab – to design new circuits based on electromagnetic relays that could be rolled out in exchanges. Much of his work was akin to programming today, but rather than coding in software Flowers was coding in hardware. First it was a matter of designing the circuits, then testing them to see if they worked as desired. Again, a familiar scenario for today's programmers, and excellent training when it came time to build the Colossus.

2 See 'Thomas "Tommy" Flowers MBE, 1905-1998', University of Greenwich alumni, **rpimag.co/uofgtflowers**

3 The General Post Office took ownership of the National Telephone Company in 1912, because the ruling Liberal government had decided the NTC had become a monopoly (having bought up the local, private telephone companies in the 30 years between 1881 and 1911).

The Post Office installed London's first automatic exchange in Holborn in 1927, and by the time Flowers left the Circuit Laboratory in 1930 "we were well advanced, we got a lot of exchanges working, and they were working reasonably well." He left to join the Post Office Research Station in Dollis Hill, where he was assigned to work with a senior engineer who was studying the problem of long-distance dialling.

"With ordinary electricity in wires, the limit to the distance is about 40 miles," said Flowers. Voices would become too faint to hear. "This chap I was put with was saddled with the problem [but] wasn't getting anywhere because he wasn't any good … but after about six months, during which time I almost went barmy, he was then promoted. If you had someone on your staff who wasn't any good, the only way to get rid of him was to get him promoted."

When Flowers was given the job to solve, he turned to thermionic valves. He realised that not only could these replace electromechanical switches, and work thousands of times faster, but they could be used to amplify people's voices. This wasn't a unique insight – unbeknownst to Flowers, Bell Labs had almost exactly the same idea in the USA – but he was the first person to introduce valves for this purpose in Britain. The trials took place between London and Bristol, roughly 120 miles, and Flowers remembered that one of the first 'tests' was for him to call his then fiancée and future wife, Eileen Green.

By the outbreak of war in 1939, his valve-based system had moved out of the lab and into production. The Research Station also had a new boss in the form of Gordon Radley, who Flowers described as a "warped genius". Radley had an astounding memory for detail and he needed it, as the Dollis Hill team stopped all civil tasks and focused instead on the war effort.

Flowers spent the early part of the war working out the best way to transmit data from radar stations on the coast to Fighter Command in Stanmore, Greater London. It wasn't until early 1942 that he signed the Official Secrets Act and was informed of the code-breaking work being done at Bletchley Park. From that point on, revealing anything about his work with the Government Code and Cipher School (GC&CS) would be an act of treason.

Soon after putting pen to paper, an act that effectively stopped him even talking about Colossus for decades, Flowers met Alan Turing for the first time. "[Turing] explained the technology of code breaking," said Flowers. "They needed the equivalent of the [Enigma] coding machine … they wanted something that the girls could operate with a sort of typewriter keyboard."

Flowers went away and designed a machine, but it was much bigger than the Bletchley Park team was expecting – based on large electromagnetic switches, as that provided reliability – and meanwhile the German Enigma machines had evolved. "When they got it they were disappointed, they didn't want it," said Flowers. Despite this, Turing was impressed by Flowers's engineering skills, and Radley's wartime diary[4] reveals that Flowers would sometimes accompany his boss to Bletchley Park from that point on.

> In a year the war could be over and lost

We should briefly pause to mention a side of Britain's code-breaking weaponry that rarely gets a mention: the radio listening stations, called Y stations, that intercepted enemy communications. "They were kind of the front wall, they were picking up the messages, and if it hadn't been for them, there would have been nothing for the code breakers to break," says Jack Copeland, Professor of Logic at the University of Canterbury in New Zealand, and a world authority on the Colossus.

One of those visits concerned a different code-breaking challenge: a new type of signal had been picked up by one of the Y stations in the latter half of 1940, with more following in early 1941. Quite unlike Enigma, which transmitted in Morse code, messages of this new type were sent as teleprinter code. Here, each character is represented by a five-strong pattern of pulses and non-pulses. Y, for instance, is pulse-gap-pulse-gap-pulse. Or 10101 in binary notation. At Bletchley Park, they denoted pulses as crosses and gaps as dots, so Y becomes x • x • x.

4 The diary is available for download from **rpimag.co/dollishill**

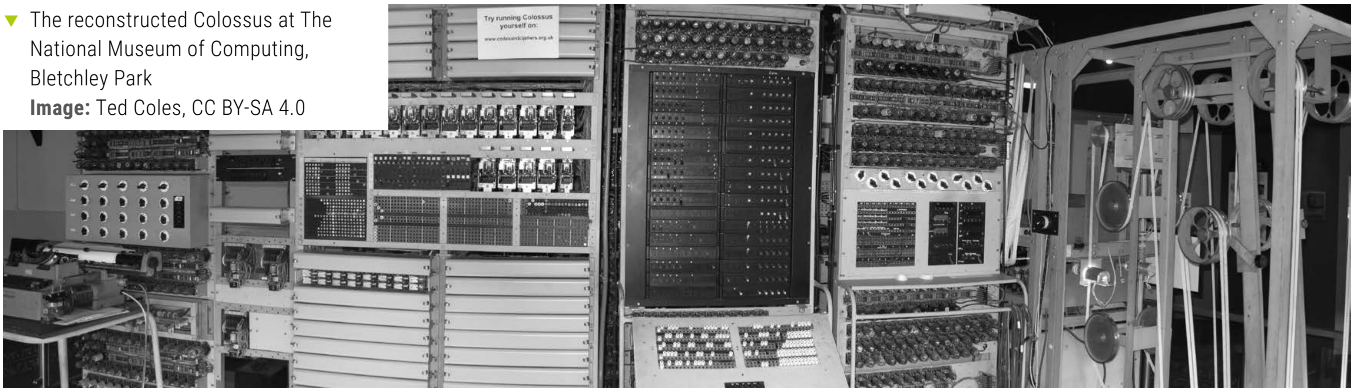
Sophisticated automatic teleprinters capable of reading and transmitting at high speeds had been around since the start of the 20th century, so when these radio messages started appearing in 1940, people had no problem recognising their distinctive tones. Initially, however, their existence was merely noted. There was too much else to do.

It was only in June 1941, when an RAF station near Folkestone intercepted an unencrypted message revealing military details, that Britain's network of intercept stations began to record them. Naturally, these messages – now encrypted with an unfamiliar cipher – were sent to Bletchley Park for examination.

While Bletchley Park had no idea of the cipher being used at this point, nor even the machinery, analysis of radio signals provided useful information. "Some of the bigger Y stations carried out radio fingerprinting," says John Pether, one of the volunteers who looks after the Colossus replica at The National Museum of Computing, adjacent to Bletchley Park. "They used a specialist oscilloscope and passed a 35mm film over the screen at high speed. They could determine whether the transmitter was running off batteries, what the frequency domains were, whether the key contacts were dirty to identify individual keys."

But the real prize was to decipher the messages' contents. Within two months, this rising challenge was being met by Bletchley's Research Section, reporting to the already legendary Colonel John Tiltman. Tiltman found his way into code breaking through the army. A World War I veteran, his ability to pick up languages became clear when he learned Russian without formal training whilst serving in a British contingent with Belarusian soldiers. Originally seconded to work for the newly created Government Code and Cipher School as a Russian translator in 1919, the move was rapidly made permanent when his incredible ability to spot patterns became apparent.

Alongside Dilly Knox, another famed cryptologist who would play a crucial role in breaking the Enigma ciphers, Tiltman



▼ The reconstructed Colossus at The National Museum of Computing, Bletchley Park
Image: Ted Coles, CC BY-SA 4.0

broke Russian Comintern signals[5] in the early 1920s. In 1933 he turned his attention to Japanese military codes, breaking six of their cipher-systems, so by the time he was put in charge of the Research Section his legend was already made. As was his reputation as one of the nicest men at Bletchley.

"My arrival was unforgettable," recalled William Filby of his first encounter with Tiltman.[6] "As I saluted, I stamped the wooden floor in my Army boots and came to attention with another shattering noise. Tiltman turned, looked at my feet, and exclaimed: 'I say old boy. Must you wear those damned boots?'" From that point on, added Filby, he would wear his full army uniform with a pair of white running shoes, much to the exasperation of more formal army officers.

By the end of August 1941, through Tiltman's work, Bletchley Park knew several things about these non-Morse broadcasts. Crucially, they knew they were encrypted using the Vernam cipher, which added a "random" value to each of the five dots or crosses (called impulses) that made up a character in the teleprinter code. As a reminder, Y is represented by five impulses: x • x • x. If the encrypting key was J, represented by x x • x •, then the new enciphered character would be the sum of its parts.

To understand this process, it may be easier to use binary notation. Here, Y is equivalent to 10101 (x • x • x) and in our example the encrypting key is J, or 11010 (x x • x •). Adding the first impulses together means adding x to x. Or 1 to 1. The answer is •, or 0 (note there is no carryover to worry about here, which is why 1 plus 1 is 0). For the second component, we're adding x to •. That sum is x or 1. Going through all five components, Y + J is 10101 plus 11010. So the broadcast letter is represented by 01111, otherwise known as the pulses • x x x x . That is, the letter T.

One more crucial thing to understand about the Vernam cipher the Nazis were using is that you decrypt by applying the same key. So, adding T to the encrypting key J (breaking this down to impulses, adding • x x x x to x x • x •) gives you Y (x • x • x).

▲ An Enigma I machine in the Museo Nazionale Scienza e Tecnologia Leonardo da Vinci, Milan **Image:** Alessandro Nassiri, CC BY-SA 4.0

While at this point Bletchley Park had no idea what the encrypted messages said, they knew they were being sent from Berlin to a German military outpost in Greece. This alone made them interesting. They called the radio link Tunny, then a common name for tuna, and would continue the fish-based code names as other connections sprang up. For example, the Berlin to Rome connection was called Bream, distinguishing it from Berlin to Copenhagen (Turbot) or Rome to Tunis (Herring).

"Each message was preceded by a string of twelve letters, which Bletchley Park called the indicator, which functioned to tell the recipient of the message the positions to which they should turn the wheels of their machine in order to decrypt the message," explains Copeland.[7]

However, this information was initially of little use to the code breakers as they had no idea of what ciphering machinery was being used to generate the encrypting key. How was it wired? How did the wheels turn? Were there any weaknesses the Bletchley Park team could exploit? They needed a piece of luck.

That came on 30 August 1941 when the same twelve-letter indicator, HQIBPEXEZMUG, appeared at the beginning of two separate messages. Two or more messages with the same indicator were called 'depths' and, as we shall see later, depths played a crucial role in Bletchley Park's attack on Tunny messages.

If the operator had simply resent the same message, it would have been no help. Here, however, the messages started the same but soon differed, with the second message around 100 characters shorter in total. This suggested it was intended as a resend of the first message, but that the operator had begun abbreviating words to save himself time. Which it no doubt did: but it also meant Bletchley Park could add the messages together and eliminate the encrypting key. As explained above, this key effectively cancels itself out when two characters are added together.

This meant the British code breakers had a stream of deciphered letters that were the result of two combined messages. Now all they had to do was untangle them. Tiltman set to work, and after several days had not only revealed the

5 Lenin established the Comintern, short for Communist International, in 1919. Its aim: to promote global communist revolution and unite communist parties under a single banner.

6 This quote is taken from Michael Smith's excellent book, *Station X: The Codebreakers of Bletchley Park* (Pan Books, 2004, ISBN 978-0330419291), p84

7 Interview with author, as are all direct quotes in this article from Jack Copeland unless stated.

plaintext[8] of the two messages, but also the whole encrypting key – almost 4000 characters long – that could surely help them work out the structure of the machine.[9]

Were that it was so easy. Three frustrating months followed in which the Research Section failed to make progress. Enter Bill Tutte, who shared his part in the story in *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, edited by Jack Copeland. "Was it a gesture of despair that Captain [Gerry] Morgan, that day in October, handed me the Tunny key, with associated documents, and said 'See what you can do with this?'"[10]

Tutte, it turned out, could do a lot. He started with the hypothesis that twelve indicators meant twelve wheels. And since eleven of the twelve indicators used all letters in the German alphabet (aside from J), but one only used 23 letters, might one of the wheels have 23 settings? We won't go into detail here, for it's enough to know that this supposition was right and that, combined with many other suppositions – and a huge amount of trial and error – Tutte discovered patterns. And if there's one thing cryptographers love, it's patterns.

With the help of other members of the Research Section, who joined in once Tutte's method of attack bore fruit, they brilliantly deduced the inner workings of the machine. From those 4000 characters of encrypting key, they worked out that the wheels came in two sets: what they called psi-wheels and chi-wheels (mathematicians commonly use Greek letters). Both sets (psi and chi) included five wheels, with two motor wheels in the centre.

They also knew that the wheels featured movable 'cams'; that is, mechanical components that could be slid by hand from an active to an inactive orientation by pushing them sideways. More than that, they knew exactly how many cams each wheel contained. And that while the starting positions for each wheel were changed with every message, the cam orientations would stay fixed for weeks on end.

At Bletchley Park, they distinguished between the starting position of the wheel and the orientations of the cams on the wheel by describing the first as the wheel *settings* and the second as wheel *patterns*. Once the wheel patterns were changed, the team was locked out until a German operator went against protocol and sent two messages using the same indicator. That is, a depth.

It's here, in July 1941, that Alan Turing makes his solitary but pivotal appearance in the story of Colossus. In Tutte's simple words, "Turing became interested in the problem of breaking a true Tunny depth and he found a method of doing so."[11]

The key to Turingery, as the method became known, was to examine differences determined by the number of cams on a wheel. For example, Turing knew that the first chi-wheel included 41 cams and so whatever position each cam was in – on or off – would repeat every 41 characters. There would be a pattern, albeit a deeply hidden one.

Fortunately for the code breakers, the German design added a 'stagger' where a psi-wheel might stay in the same position. The idea was to add complexity, but it actually introduced a weakness. By examining the differences between encoded characters, once broken into a series of dots and crosses, Turing showed that it was possible – through a mix of luck, a giant sheet of graph paper, much crossing out, and sheer repetition – to uncover each cam's orientation on one wheel, and then to determine the orientations of all the cams on all the wheels.

Turingery was a laborious process, but, at this stage of the war, the wheel patterns were only changed once a month or longer (to be accurate, the chi-wheels were changed monthly, the psi-wheels every three months).

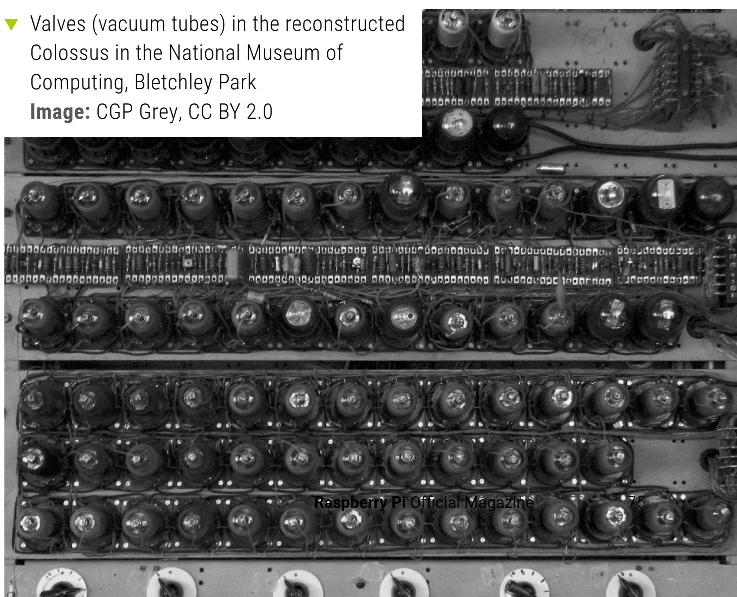
"Once Turing had shown the Research Section how to extract the wheel patterns, then they were in the game," says Copeland. "They'd got exactly the same information the German guy who was decrypting the message with his Tunny machine had got.

8 That is, unenciphered text. We'll use the term plaintext to distinguish it from enciphered text.

9 Tiltman's work is set out in detail in 'The Tiltman Break' by Friedrich L Bauer, in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers* (Oxford University Press, 2006, ISBN 978-0192840554), Appendix 5, p370

10 William T Tutte, 'My Work at Bletchley Park', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Appendix 4, p356

11 William T Tutte, 'My Work at Bletchley Park', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Appendix 4, p360. Jack Copeland gives an overview of how Turingery works from pp380-382



▼ Valves (vacuum tubes) in the reconstructed Colossus in the National Museum of Computing, Bletchley Park
Image: CGP Grey, CC BY 2.0

They knew the wheel patterns and the message indicator was telling them what the wheel settings were, so they could simply set up their replica machine and decrypt the traffic, just like the German recipient."

With a procedure laid out, the day-to-day job of breaking Tunny messages moved to a section run by Major Ralph Tester. The Testery, as it was called, could now decipher a torrent of messages encrypted using Tunny until the wheel patterns changed, so long as the Nazis were kind enough to keep sending the indicators at the start of each message.

This generosity continued until November 1942, when the German military switched instead to numbers found in a book issued to Tunny operators. These numbers told the operators the starting positions for the wheels. At which point, the Testery was plunged into near-darkness as the torrent became a trickle. All it could hope for was a depth (two messages sent with the same indicator), which would allow them to apply Turingery, but this time 'in reverse' to extract the wheel settings.

Crucially, however, the Nazis had tightened up other security measures too. "It was a double whammy," says Copeland. "The code breakers lost the twelve-letter indicators, and the depths were getting rarer, so the number of messages they could break using Turing's method was becoming smaller and smaller."

Bill Tutte was not to be deterred. Building upon the same ideas as Turingery, but introducing statistical methods, he devised a way to work out the starting positions of the first and second chi-wheels for any message. His so-called '1+2 break' meant that with enough time the code breakers could detect the most likely chi-wheel start positions simply by counting (and adding) dots and crosses.

The idea was to add the teleprinter message to the stream of chi (that is, every possible combination of encrypting key from the chi-wheels), then count up the dots. Then slide it along by one and repeat the process. And so on until you had tried all the different possible chi streams. Tutte worked out that if around 54% of the individual pulses were zeroes when totalled up, the wheel settings were almost certainly correct.[12]

But the sharper-eyed reader would have spotted the phrase 'enough time' above, which hides a large problem. 'Enough' here, according to Newman's own estimate, could mean centuries for some messages. Even though they only needed to deduce a pair of wheel settings to begin the process, that meant 1271 (41×31) different scans of the text to spot likely candidates. And as this was a statistical method, they needed long messages, not short ones, adding to the time.

All of which meant they needed a much quicker way to count and add – and that meant some form of automatic adding machine. Especially as, even after the first two chi-wheel positions were known, you would have to repeat the method to reveal the three remaining chi-wheel settings. Only at this point would experienced code breakers have enough information to work out the remaining wheel positions by hand.

Tutte took his 1+2 break theory to Captain Gerry Morgan and Max Newman. "They began to tell me, enthusiastically, about the current state of their own investigations," wrote Tutte.[13] "When I had an opportunity to speak I said, rather brashly, 'Now my method is much simpler.' They demanded a description. I must say they were rapidly converted."

In what historian Jack Copeland describes as "a moment of inspiration",[14] Newman realised that one of his Cambridge colleagues, Eryl Wynn-Williams, had already designed an electronic counter based on valves that he thought could be adapted for this crucial part of the job. He called in Wynn-Williams to take control of that side of the design, with the GPO in charge of creating the drive mechanism for the tapes, the photoelectric readers to detect dots and crosses as they passed, and a combining unit to add up the numbers.

But when we say the GPO was put in charge, we should emphasise that it wasn't given to Tommy Flowers: instead it passed to Francis Morrell, head of the teleprinter group. Morrell commissioned GPO engineers Eric Speight and Arnold Lynch to produce the photoelectric readers (without telling them why).[15] They did an excellent job, but the combining unit – which used electromagnetic relays – proved more challenging for Morrell and his team. It was time to call in an expert in relays, Tommy Flowers.

12 Interested readers should read in-depth descriptions found in the technical appendices of *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*. But here's a quick summary. In teleprinter code, as we detailed above, each letter can be represented by a five-long string of dots and crosses, or zeroes and ones. Each letter in the encrypting key is also represented by five dots and crosses. These are added together to form a new character, the ciphertext, which is what is transmitted. Tutte's breakthrough was to realise that the correct settings for the first pair of wheels would be revealed (probably but not every time) by a very slightly higher number of dots – or zeroes – in the resulting stream of five-digit code. This is why it became a counting game.

13 William T Tutte, 'My Work at Bletchley Park', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Appendix 4, p364

14 Jack Copeland, 'Machine against Machine', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Chapter 5, page 64

15 Brian Randell, 'Of Men and Machines', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Chapter 11, p146

"I quickly came to the conclusion that the [machine] would never be any good, that we were wasting time on it," said Flowers. In particular, he was extremely sceptical about keeping two tapes in perfect synchronisation. One run with a long message, he pointed out, could take several hours and that would surely be too much for paper tape to stand. "So I thought up something different."

Flowers's key idea, one he was uniquely placed to come up with, was that by using thermionic valves as switches "an analogue of the coding machine could be made". So no need for tape, as the data would be stored within the machine.

While Flowers was convinced his solution would work, he struggled to convince the Bletchley Park hierarchy. This was partly due to thermionic valves' poor reputation for reliability – used in radio sets and early TVs, they were vulnerable if moved and were well known for their limited operational life – but Flowers had built automatic exchanges with hundreds of valves in them, and knew that they would prove extremely reliable so long as they were kept switched on.

Newman appears to have been convinced on this point, but there was a second problem. When Newman asked Flowers how long it would take to build, the Post Office engineer gave the honest reply of at least a year. "[Newman] said, well, they just couldn't wait a year; in a year the war could be over and lost," said Flowers.

Despite this, Newman cautiously welcomed Flowers's ideas. In a report to Commander Edward Travis, dated 1 March 1943, he wrote:[16] "Flowers, of the PO, has produced a suggestion for an entirely different machine, in which the message, and the wheels to be compared with it, would be set up on valves, by means of relays. This would involve 5000 or so valves, and about 2500 relays. (The heat the valves would generate would be equal to that of eight two-bar electric fires!)"

While Newman describes this as an "ambitious scheme", he isn't against it. "I feel that this is basically the right sort of approach, and that it is very much to our advantage to try out these techniques, and if possible get a step ahead with them. At the same time, since there is a risk of hold-ups along these new paths, I emphasised that the simpler tape-machine (which also has the advantage of easy adaptability to all sorts of purposes) should be gone on with also, at full speed."

One of Newman's criticisms of Flowers's proposal was that it would take too long to convert each new message into electronic form and that some messages would be too long to be stored. Flowers understood and accepted this point, quickly revising his proposal to one where only the chi streams need be stored electronically. Within two weeks Newman wrote a handwritten note to Travis stating:[17] "For the more ambitious machine they [the Post Office] now propose to use tape for the message and valves only for the fixed wheels. This does away with the main objection to their first scheme (lack of flexibility for us)."

Nevertheless, Bletchley Park decided to back the two-tape machine rather than gambling on Flowers's design. By June, the first two-tape machine – featuring a combining unit that Flowers and his team developed as per the original request – was delivered to Bletchley Park.

It was quite something to behold. So much so that it was named the 'Heath Robinson' by the Wrens who operated it due to its elaborate, cumbersome design (William Heath Robinson was a famous cartoonist of the time who created comically complicated machines to perform a simple job; cartoonist Rube Goldberg earned similar fame in the USA). It worked, but only after weeks of effort by Newmanry section members Jack Good and Donald Michie.

Despite the best efforts of the British government, the very modern legend of Colossus lives on

Bletchley Park ordered more, and an improved, finessed version arrived in November. This slicker machine lost the 'Heath' and was simply called a Robinson. And Flowers's initial reaction that the design wouldn't work proved incorrect: in fact, their flexibility meant they were still being built and improved upon until the end of the war. The later versions were even christened Super Robinsons.

While Bletchley Park wouldn't commit money to the Colossus, they also said that the Post Office could go ahead with the design at its own risk and cost. Fortunately, Flowers had the backing of Gordon Radley, who understood the importance of the work and gave his chief engineer access to all the materials and people he required.

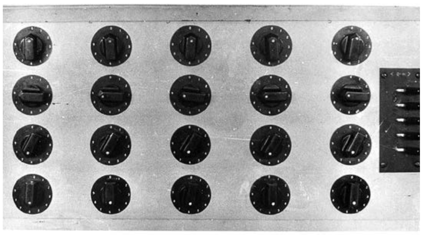
16 James A Reeds, Whitfield Diffie, J V Field (eds.), *Breaking Teleprinter Ciphers at Bletchley Park* (Wiley IEEE Press, ISBN 978-0470465899), Appendix D: Initial conception of Colossus, p535. Also available at 'GCHQ celebrates 80 years of Colossus', **rpimag.co/colossus80** 18 January 2024

17 As before, p536

"I knew what it would do," said Flowers. "I drew a lot of square boxes, with lines joining up to these, and I specified what each box had to do." For example, how each box would simulate one of the code wheels. Flowers had to work alone until others could be let into the secret, but was soon joined by talented colleagues.

"Sid Broadhurst was a master of system design, and of the logic and timing of complicated relay circuits," wrote Harry Fensom, another member of the team at Dollis Hill, in his article 'How Colossus was Built and Operated – One of its Engineers Reveals its Secrets'.[18] "Bill Chandler was the expert on digital valve design, and could produce a system that worked straight away when implemented from the drawing board."

In fact, while some histories simplify the computer's story and say that Tommy Flowers "built" the Colossus, he had a large team of engineers working alongside him in the lab, plus the considerable manufacturing power of the Post Office to draw upon. Fensom, who contributed to later Colossi's design and worked on-site at Bletchley Park once the machines were installed, added that other engineers helped draw "the detailed layouts and thought through the fine print of the design" while "dedicated wiremen" soldered it all together.



▲ The 'set total' switch panel on Colossus
Image: from *General Report on Tunny: With Emphasis on Statistical Methods* (1945), UK Public Record Office, Public Domain

Flowers started designing the first Colossus in February 1943, with the machine delivered to Bletchley on 18 January 1944 and operational on 5 February.[19] So, a year later, just as Flowers had predicted[20]. It filled a room, more than justifying its name. While the machine was loud – thanks to the use of many electromechanical parts, not the valves – it worked from the get-go. Not only was it quicker than the existing Robinson machines, when they ran a message twice they obtained the same result. This wasn't always the case with the Robinsons, even the improved versions.

In his seminal report 'The Colossus', computer historian Brian Randell quotes Flowers directly: "They just couldn't believe it when we brought this string and sealing wax sort of thing in and it actually did a job. They were on their beam ends at the time, Robinson just hadn't got enough output, they wouldn't go fast enough, and suddenly this bit of string and sealing wax, in about ten minutes … and then they started to take notice!"[21]

Gordon Radley's wartime diary[22] is more measured in its analysis, but still the delight at its success rings through. "Since February 22nd Colossus has been in full daily operation for 16 hours per day, and, although the operating staff are not yet sufficiently familiar with it to utilise it to the best advantage, it has dealt with eight messages in eleven hours. For the remaining eight hours of its day, Colossus is turned over for the attention of Research Staff who are endeavouring to graft on to it equipment for providing still further facilities."

Flowers and Radley expected Bletchley Park to immediately place orders for more machines, so were surprised when they didn't hear anything. Still, they decided to press ahead with the manufacture of the more time-consuming components so that they would be ready when the call came.

It's a good thing they had a head start, because the following month they were visited by "some man in uniform, straight from the War Cabinet, [who said] that we had to make twelve of these machines by June the 1st," remembered Flowers. By then it was obvious to all that it was only a matter of time before the Allies invaded Europe, and their visitor's next statement made Flowers suspect that his machines might just play an important role. "We were told that unless you can make these machines and have them working by June the 1st then it would be too late."

While the Post Office couldn't build a dozen machines in such a short time, Flowers promised that the first new machine would be working by 1 June with the next two following in July and August. So began an intense four months of effort to build the Colossus II, this time with 2400 valves. "We worked day and night, six days a week," said Flowers. "We worked ourselves almost to exhaustion."

That work continued right up until the last moment. Only at 1am on the morning of 1 June 1944 did Flowers and his team

18 Harry Fensom, 'How Colossus was Built and Operated – One of its Engineers Reveals its Secrets', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Chapter 24, p301

19 This is according to Gordon Radley's war diary, available for download at **rpimag.co/dollishill**. See War Diary Vol 2, p227

20 Some histories state that it was running at Bletchley in December 1943, and this is also what Flowers says in his 1998 interview. However, we know from official records that it was only installed and working at Bletchley Park in February 1944. Copeland suggests that December 1943 is likely to be the date when Flowers successfully completed a test run on the machine, which was built at Dollis Hill.

21 Brian Randell, 'The Colossus', University of Newcastle upon Tyne, 1976, **rpimag.co/collohist**

22 Gordon Radley's war diary, **rpimag.co/dollishill**, p227

work out that the reason the Colossus II obstinately refused to work was due to misbehaving valves. This was a problem best left to valve expert William Chandler, so they left him there. "And then Flowers came back the next morning and Colossus II was buzzing away like a top," says Copeland. "Chandler was looking very pleased with himself, but also standing in a pool of water because the radiator had leaked during the night." This, incidentally, is why some reports incorrectly state that the Colossus was water-cooled; ignore all such articles.

"Between [5am] and 8pm this evening, Colossus II was running continuously and has very successfully dealt with no less than 20 separate tasks," wrote Radley in his diary that day. "This is roughly equivalent to a full 24-hour output from all the other machines at present operated by the Group concerned, and has given very great satisfaction to Professor Newman and Commander Travis."

The Colossus II was so quick thanks to the advent of what can best be described as parallel processing. This involved having the equivalent of five processors working on the same tape, effectively bringing the speed up to 25,000 characters per second. "It meant more gates and counters, of course," wrote Fensom,[23] "but we didn't mind a thousand extra valves."

Nor did Bletchley Park mind throwing hundreds of women at the problem. For while the Colossi revealed the settings for the five chi-wheels, there was still much work to be done before a message could be decrypted, and much of this burden fell to Wrens and women from the army's Auxiliary Territorial Service.

"There were so many messages being sent in, when one didn't come good you were sent another one," Cora Jarman told Tessa Dunlop, author of *The Bletchley Girls*,[24] of her time in the Newmanry. "Checking checking. It was very boring." The tedium wasn't helped by the fact that the women weren't privy to the methods, they were merely told what to look for.

After a Colossus had successfully revealed a likely set of starting positions for the chi-wheels, the message needed to be 'de-chi'ed'. This meant processing it through a machine to remove the chi encryption, which one of the women would do. The de-chi'ed message tape would then be taken to the Testery, where the code breakers would set to work. When a complete set of wheel positions arrived, it was women who would set up a Tunny machine and check the output.

So it was hard work for everyone involved, but also vital work. According to legend, on 5 June 1944, General Eisenhower was handed a decrypt fresh from Bletchley Park. He read it, then declared: "We go tomorrow." This may be too neat a story, but it's certainly known that Eisenhower was a regular visitor to Bletchley Park in the run-up to D-Day and an admirer of its work.

We should not oversimplify history to say that the Colossi alone won or shortened the war. For one, we can't know if Eisenhower would have still invaded the following day if he hadn't read that message. But giving all the credit to the computer ignores the months of work done by a network of double agents who had been feeding the Nazis with misinformation.[25] Not to mention other ruses, such as creating fake barracks near Dover, or dropping aluminium foil to create the illusion of incoming planes on German radar. All these deceptions were to convince the German military that the Allies were going to land in Pas-de-Calais rather than their real target of Normandy.

The deception worked. The Nazis held back reinforcements from Normandy, allowing the Allies to establish a strong foothold there. Over the coming months, as the Allies battled their way across Europe, more Colossi were installed. By the end of 1944, seven whirred away, spread across two vast, bomb-proof concrete buildings. Three more Colossi arrived in 1945. One mathematician might be in charge of two Colossi, and they would give Wrens plugging diagrams to follow. If something went wrong, Post Office engineers were on hand to fix the problem.

"I saw this astonishing machine the size of a room," said Joanna Chorley.[26] "It was ticking away, and the tapes were going round and all the valves, and I thought what an amazing machine… Like magic and science combined!"

What's more, the machines became more magical over time. For example, Bill Tutte created a technique called rectangling that could determine the wheel patterns. However, it required a quite legendary amount of effort by hand and one tiny mistake could ruin the calculations. Donald Michie, a talented member of the Testery, came up with a proposal that meant a Colossus could become a high-speed rectangling machine through the addition of a "rectangling gadget".[27] A good thing, as the Germans started to change the wheel patterns on a daily basis as they became more concerned about security.

23 Harry Fensom, 'How Colossus was Built and Operated – One of its Engineers Reveals its Secrets', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Chapter 24, p301

24 Tessa Dunlop, *The Bletchley Girls* (Hodder & Stoughton, 2015, ISBN 978-1444795745), p124

25 This was Operation Fortitude South, part of a wider plan called Operation Bodyguard, geared at fooling the German High Command into expecting an attack at Pas-de-Calais.

26 Tessa Dunlop, *The Bletchley Girls*, p125

27 Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, p244

"We were in no doubt about how important [Colossus] was," said Captain Jerry Roberts in 2010,[28] a member of the Testery. "We saw a number of messages signed by Adolf Hitler himself." The operation became slick, with Roberts adding that "I suspect sometimes we genuinely saw the messages before the blessed Germans."

The day they installed Colossus II marked the moment when Flowers moved away from day-to-day charge of the computers, as he took on a different project to design a computer to help target anti-aircraft guns. In this way, unknown to both men, Flowers and George Stibitz's war careers echoed each other. It made sense to employ Flowers's talents elsewhere, as the template for each new machine was now established and his skills were too valuable to be wasted.[29] From this point on, Allen Coombs took control.

For the rest of the war in Europe, the Colossi continued to break codes, with the tenth and final operational machine installed in April 1945. A month later, the teleprinters relaying the Tunny messages would fall silent – an eerie experience for those working in the huts – with the final message sent shortly before VE Day on 8 May 1945, when Nazi Germany unconditionally surrendered to the Allied forces.

A lack of new messages didn't spell the end for the Colossi. There was still much to be learnt from the backlog of messages still to be deciphered, including clues to the identities of British agents giving material to the Soviet Union. For while the USSR was an ally of sorts, it was always an uneasy relationship.

Moscow must have known about the British success decrypting Tunny, because British civil servant John Cairncross – later unmasked as the fifth member of the infamous Cambridge spy ring that included Kim Philby and Guy Burgess – gave them decrypts that he smuggled out of Bletchley Park by stuffing them down his trousers. But the USSR clearly didn't realise how comprehensively Tunny had been broken, as it grabbed the Nazi cipher machines and used them (with enhancements) to encrypt messages after the war.

With this in mind, it's easy to understand why the British government wanted to keep its secrets safe. We don't know if Cairncross knew of the Colossi's existence, such was the level of secrecy within Bletchley Park, and so it could be that the Soviets had no idea about the existence of Britain's high-speed computers.

However, there was no need to keep all ten active Colossi in service, so the command came to dismantle them. In a 2012 short film to celebrate the 70th anniversary of Colossus's birth,[30] Google interviewed engineers and wirers who worked on the machines. Their most poignant remarks came when describing the Colossi's demise.

"When we had the order to dismantle all the equipment, the idea was to smash everything. Smash all the valves and everything else," remembered Roy Robinson.[31] "Some of [the Colossi] were thrown down coal mines," added Albert Bareham. "The whole piece of equipment was dumped down a disused coal mine and things like that."

Even 70 years later, you can hear the disappointment in the voice of Margaret Bullen (née Boulton), who like other women "with nimbler fingers" helped with wiring and soldering, when she remembers the end of her time at Bletchley Park. "We carried on … until the war was over, and immediately we were told we could go," she said. "You felt cheated in a way that you'd worked so hard on it and that it was going to be no more."

None of them knew at the time that at least two Colossi survived the massacre. These were quietly transported to the new headquarters of CG&GS – renamed Government Communications Headquarters, GCHQ, the following year – in northwest London. The Colossi then moved with GCHQ to Cheltenham in 1951, where they continued to provide valuable code-breaking service for another decade.

"I worked as an engineer on Colossus for a year during the 1960s," wrote Bill Marshall, a former GCHQ engineer, in an article on GCHQ's website to mark Colossus's 80th anniversary.[32] "I was told very little about the machine I was working on – what the machine was actually doing was not for me to know. My job was to repair it as necessary, using just a few circuit diagrams and no detailed user handbook."

It's likely that Marshall was one of the final engineers to tend to the machines, with the same article stating that Colossi stayed in use "until the early 1960s". Despite the computers' retirement, the government refused to declassify the Colossus

28 'Bletchley's code-cracking Colossus', BBC News, 2 February 2010, **rpimag.co/codecracking**

29 After the war, Flowers stayed in position at Dollis Hill and helped the General Post Office in its push to a modern telephone network. His only computing involvement was in the early stages of the Pilot ACE, as described in Chapter 10, which eventually resulted in the MOSAIC. This was designed by Allen Coombs and William Chandler, who worked under Flowers, and is considered the truest implementation of Turing's ACE design. MOSAIC is said to have provided good service to the Ministry of Supply from 1955 until its retirement in the early 1960s.

30 'Colossus: Creating a Giant', **youtu.be/knXWMjIA59c**, 8 March 2012

31 While this may have been the fate of some of the Colossi at Bletchley Park, we know that a truck's worth of components were shipped to Manchester at the request of Max Newman

32 'GCHQ celebrates 80 years of Colossus', **rpimag.co/colossus80**, 18 January 2024

project. It was only through the efforts of historians, particularly Brian Randell, that more information came to light in the mid-1970s. It took another quarter of a century for the government to release a 500-page report written by members of the Bletchley Park team, 'General Report on Tunny',[33] that filled in many of the missing details.

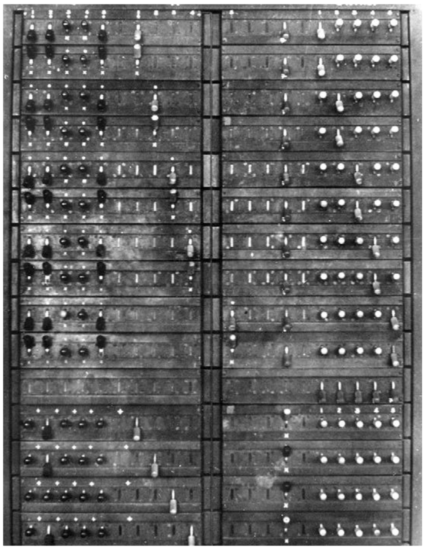
"ENIAC demonstrated very publicly the viability of using large numbers of electronic tubes," said Randell in an interview now found on YouTube,[34] "just as Colossus had done on a smaller scale, but privately. Colossus, I think, had an effect within Britain, but a rather limited effect, whereas ENIAC brought 'giant electronic brain' onto front pages."

Which brings us to the big question of the impact of Colossus. Randell doubted that it had much effect on Alan Turing – "One of his major characteristics is his wish to start everything from scratch" – but there is little doubt that it inspired Newman to start work on a computer at the University of Manchester. This would give birth to the Manchester Baby.

Even though some in American intelligence knew about the Colossus, we can also state with some confidence that it had no direct impact on the growth of computing in America. They had the ENIAC and detailed plans for the EDVAC, and unlike the reticent British the Americans were happy to share their wartime advancements.

This does not mean the Colossus should be ignored in computing history, as it marked many firsts. The Colossus I was the first computer to use electronic binary circuitry on a large scale. The first to use a clock pulse to synchronise operations. And while it was a special-purpose computer, it and its successors were programmable – to an extent.

Timing naturally becomes important when claiming world firsts, so it should be noted that the first Colossus was operational in February 1944, pre-dating the ENIAC by almost two years. This was why, when Randell – with the assistance of Coombs – gave a guarded presentation on the Colossus to the International Research Conference on the History of Computing, Los Alamos, in 1976 it caused quite a stir.



◄ Colossus K2 switch panel showing switches for specifying the algorithm (left) and the counters to be selected (right)
Image: UK Public Record Office, Public Domain

Bob Bemer, who helped develop the ASCII character code, wrote a short but vivid article describing the proceedings.[35] "My decision to keep everyone in view paid off. I looked at Mauchly [John Mauchly of ENIAC fame], who had thought up until that moment that he was involved in inventing the world's first electronic computer," he wrote. "I have heard the expression many times about jaws dropping, but I had really never seen it happen before. And Zuse – with a facial expression that could have been anguish."

All of which begs the question, what might have been different if the Colossi's creators hadn't been sworn to secrecy? Could it have given extra momentum to Britain's efforts? Might it have changed computing history?

What we know for certain is that Tommy Flowers, and others involved in the creation of the Colossus, never received the recognition they deserved in their lifetimes. Flowers was given a one-off gift of £1000 (equivalent to roughly £35,000 in 2025) after the war, but is said to have shared that with others involved in the project.[36] He also collected an MBE in 1943 for his early war efforts, before he even started work on the first Colossus. Standing for Member of the British Empire, this is a national honour, but among the lowest-ranking.

It was only through the effort of Randell that Flowers gained greater recognition in later life. This included an honorary degree from Newcastle University (where Randell was then a professor and remains a professor emeritus) in 1977,[37] but when asked how he felt about recently winning the Charles Babbage award in the late 1990s Flowers's response was muted.

33 This can be read in full here: **https://rpimag.co/tunnyreport**

34 The Computer Pioneers: Interview footage of Brian Randell, **youtu.be/9HFyrJq3e1E**

35 A copy of the document can be found at rpimag.co/colossusbemer, and it is quoted extensively in 'The First Public Discussion of the Secret Colossus Project' by Michael Williams, in IEEE Annals of the History of Computing, Vol 40, Issue 1, Jan–Mar 2018

36 Some online articles state that Flowers paid for parts of the Colossus himself, but this is unlikely to be true. "I think Flowers meant he paid the personal cost," says Copeland, referring to an interview where Flowers mentioned being "out of pocket" for Colossus. "He's in the Dollis Hill Research Station. They've got relays and vacuum tubes piled up behind every door. What would he have to pay for?"

37 Ignore sources that suggest it was given in 1973. The date 1977 is taken from Newcastle University's official listing, found at **rpimag.co/nuhongrads**

"I appreciate it," he said, "but it isn't the same as at the time. It would have been fine if it had happened in 1946, but in 1998 it doesn't really mean much. I'm quite honoured but I'm not over-enthusiastic."

Perhaps he would have appreciated the BBC documentary *Code-breakers: Bletchley Park's Lost Heroes*, which shone a light on the efforts of Tutte and Flowers in breaking Tunny. Broadcast in 2011, the hour-long film is still available to watch online.[38]

More accolades also followed Flowers's death. There is now a blue plaque commemorating the Dollis Hill Research Station, while British Telecom – the organisation that inherited the Post Office's national telephone duties in 1981 – created a statue of Flowers at its research facility in 2013. If anything, Tommy Flowers's fame and repute continues to grow, with streets and even a pub now named after him.[39]

Colossus itself lives on in two forms. The first is a fully working recreation of Colossus II, created through the hard work of dozens of engineers but the dogged determination of one man: Tony Sale. "In 1991, some colleagues and I started the campaign to save Bletchley Park from demolition by property developers," he wrote.[40] "At the same time I was working at the Science Museum in London restoring some early British computers. I believed it would be possible to rebuild Colossus, but nobody else believed me."

With the help of still-classified documents, circuit diagrams that engineers had illegally kept, and Arnold Lynch, one of the men behind the photoelectric reader, Sale first recreated the design. Then, this time helped by a mix of "current and retired Post Office and radio engineers", they built a limited version of the first Colossus – switching it on in time for the 50th anniversary of the ENIAC in 1996. A ceremony Tommy Flowers attended.

By 2003, after what Sale described as "over 6000 man-days of effort", the replica of the first Colossus was working almost exactly as designed. But, not content with this, the team decided to "upgrade" to Colossus II. Their self-imposed deadline was 1 June 2004, exactly 60 years after the second computer became operational. This time, there would be no misbehaving valves.

38 It can be found directly at **rpimag.co/timebreakers**, but it's easier to visit **clp.bbcrewind.co.uk** and search for 'Timewatch code-breakers'.

39 Tommy Flowers Drive is in Kesgrave, near the BT Laboratories, while the Tommy Flowers Mews is in North London. The community-run Tommy Flowers Pub is based in Poplar, London, where Flowers was born.

40 Tony Sale, 'The Colossus Rebuild', in Jack Copeland (ed.), *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*, Chapter 12, p150

"[On] Thursday 20th May I filmed Colossus Mk 2 setting all five K wheels [psi-wheels] on the Bream cipher text and after editing, this video was shown to 120 people at our commemoration event at the Science Museum in London on 1st June 2004," wrote Sale.[41]

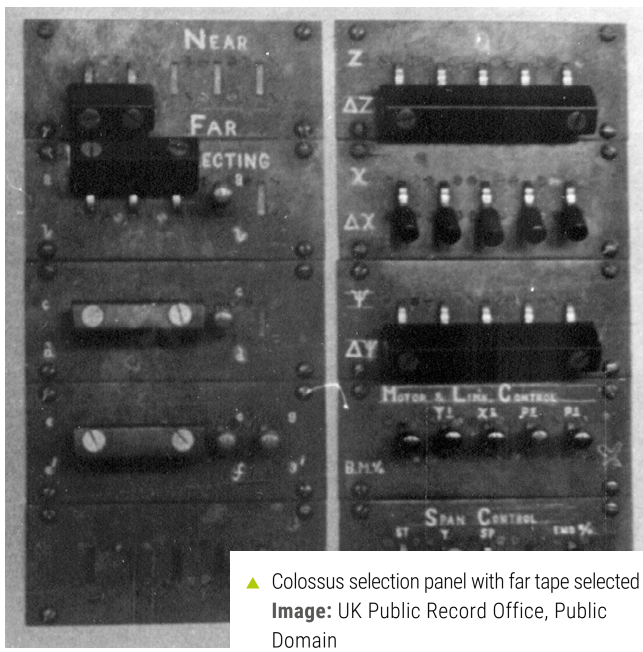
The rebuilt Colossus II remains on show at The National Museum of Computing, which is on the same site as Bletchley Park but is run separately.[42] The museum also houses what it claims is the world's largest collection of historic computers and, unlike in most museums, visitors are encouraged to get hands-on with them. The Colossus is very much 'look but don't touch', unless you're one of the volunteers who keep it going.

The Colossus is also kept alive by a virtual version – search for 'virtual Colossus' online – where you can see a 3D representation of the computer and even set it to work. While the original program was written by Tony Sale, it was Martin Gillow who recreated it in 3D and brought it up to date.

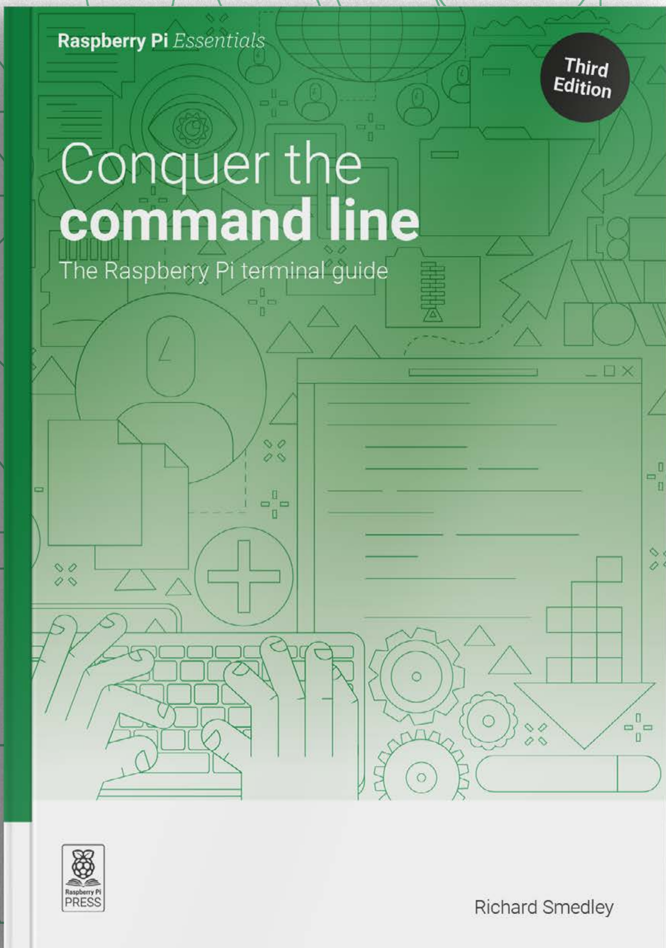
All of which means that despite the best efforts of the British government, the very modern legend of Colossus lives on.

41 'Rebuilding Colossus' by Tony Sale, The National Museum of Computing website, undated, **tnmoc.org/rebuilding-colossus**

42 Visit Bletchley Park's website at bletchleypark.org.uk and The National Museum of Computing's website at **tnmoc.org**



▲ Colossus selection panel with far tape selected **Image:** UK Public Record Office, Public Domain



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**

Part 03

Make games with Python: take control

Write some code to get to grips with the keyboard, mouse, and gamepad in Python and Pygame

Maker

Sean M Tracey

Sean calls himself a technologist, which is a fancy way of saying he still hasn't decided what he wants to do with technology – other than everything.

smt.codes

In the first two parts of this series, you got to grips with the core concepts of drawing and moving shapes of all types, sizes and colours with Pygame. Now that we know our way around Pygame, we're going to start making things that we can play with that are a bit more interactive. This time, we're going to make two simple programs to learn how to use our keyboard and mouse. The examples in this tutorial are a bit longer than in previous instalments, so you may want to download and follow along with the example code from GitHub repository (**rpimag.co/makegamescode**).

For our first program, we will use the keyboard; with it, we'll draw a red square and give it some code so it can move left and right and jump, which may conjure memories of a certain heroic plumber. Our second program will use the mouse. Again, we'll create a square which we can pick up, drag around and which, when we let go of our mouse button, will drop to the floor with the help of a little Pygame-programmed gravity. Finally, we'll learn how to modify the keyboard example to support a gamepad. We're focusing on game dynamics at this point, but don't worry: later parts of this series will explore the more aesthetic aspects of game design!

Pygame keyboard input

On to our first program – **keyboard.py**. Unlike in previous parts, we're not going to chop and change bits of code to show off Pygame's range of capabilities. Instead, we're going to walk through the code to understand what each bit does. Like a lot of things in computing, we are going to start at the top. The first several lines of code should look familiar to you by now; these are the statements we've used previously to import libraries, initialise Pygame, and define our window. The next lines are constants and variables that determine how our keyboard-controlled square should look and where it should be.

> *We're focusing on game dynamics at this point*

```
import pygame

# Pygame Variables
pygame.init()
clock = pygame.time.Clock()
FPS = 60

WIN_WIDTH = 800
WIN_HEIGHT = 800
surface = pygame.display.set_mode((WIN_WIDTH,
WIN_HEIGHT))
pygame.display.set_caption('Pygame Keyboard!')

# Constants
PLAYER_SIZE = 20
MAXJUMP_VY = 25.0
MOVE_SPEED = 1.0
MAX_VX = 10.0

# Variables
player_x = (WIN_WIDTH / 2) - (PLAYER_SIZE / 2)
player_y = WIN_HEIGHT - PLAYER_SIZE
player_vx = 1.0
player_vy = 0.0
gravity = 1.0
```

Following that, we have two functions, `move()` and `quit_game()`, which we'll use to move the square and quit the game. We also have the main loop where we draw all our pixels, including the square, and update the display.

Make games with Python 2nd Edition out now

This tutorial is part of a series from the latest revision of *Make games with Python*. Grab it today at **rpimag.co/makegamesbook**

Moving the square

Before now, almost all the code we've written has been inside our main loop, which becomes a little hard to follow when the code gets long. To make things easier, we've put the code for moving our square into its own function, `move()`, which expects you to supply it with a `direction` and `jump` argument. Let's look at the code (shown overleaf) one chunk at a time.

The first line is a `global` statement. Code inside the `move()` function no longer has the same scope as the main loop: although we can *look* at the values of variables defined outside our function, we can't *reassign* their values unless we mark them as `global`. See **rpimag.co/PyScope** for details.

The `move()` function first checks to see whether the player switched direction. In other words, did `move()` receive either of:

1. A positive value for direction while the player was already moving to the left (is the X velocity, `player_vx`, less than 0?)
2. A negative value for direction while the player was already moving to the right (`player_vx > 0`)

If so, we'll stop and change direction. Think about it: if you're running in a straight line, you can't turn right around and keep running at the same speed. You need to stop, turn, and build the speed up again. We do this by setting the X velocity to the base `MOVE_SPEED` (`1.0`) multiplied by the direction.

So long as direction is not zero, we then add `player_vx` to `player_x`. If `player_vx` is negative, the square moves left; if positive, the square moves right. We don't want our square to run off the screen either; the next few lines stop our square moving if it's at the left or right edge of our window.

```
def move(direction, jump):
    global player_x, player_y, player_vx,
player_vy, gravity

    # Did we switch direction along the x axis?
    if (direction > 0 and player_vx < 0) or
(direction < 0 and

player_vx > 0):
        player_vx = MOVE_SPEED * direction

    # Move the player along the x axis
    if direction != 0:
        player_x += player_vx

    # Keep the player within the screen bounds
along the x axis
    if player_x > WIN_WIDTH - PLAYER_SIZE:
        player_x = WIN_WIDTH - PLAYER_SIZE
    if player_x < 0:
        player_x = 0
```

Next, the code checks to see if `jump` is `True`, and also confirms that the player's square isn't already in the middle of a jump (if the `player_y` coordinate is equal to the window height less the square's height, the square is on the ground). If that's the case, the player's Y velocity (`player_vy`) is set to the maximum. The line `if player_vy > 1.0` checks whether our square is travelling upwards at a speed greater than 1 pixel per frame. If it is, we multiply that value by 0.9 so it will eventually travel less than 1 pixel per frame; when that happens, we set the value to 0 so that the square can start falling back to the ground.

Next, our code checks whether our square is in the air: if it is, it will need to come back down (**Figure 1**). If the square is in the air, we start adding the `gravity` value to the `player_vy` value; this will make our square move back down. Each time we add `gravity` to the `player_vy` value, we multiply the former by 1.1; this makes the square speed up as it falls back to the bottom of the screen, just as it would if you threw a ball in the air. The code sets `gravity` to 1.0 when the square lands on the ground.

```
    # If we're not already jumping, max out the
y velocity
    if jump and player_y == WIN_HEIGHT - PLAYER_
SIZE:
        player_vy = MAXJUMP_VY

    if player_vy > 1.0:
        # Decrease player_vy throughout the jump
        player_vy = player_vy * 0.9
    else:
        player_vy = 0.0

    # Is our square in the air?
    # Better add some gravity to bring it back
down!
    if player_y < WIN_HEIGHT - PLAYER_SIZE:
        player_y += gravity
        gravity = gravity * 1.1
    else: # Reset gravity so it starts at 1.0
next time we jump
        gravity = 1.0
```

Because Pygame Y coordinates decrease towards the top of the window, the code subtracts `player_vy` from `player_y`. We then use the `min()` function to keep the square from falling through the floor: we make sure that the Y coordinate is never greater than `WIN_HEIGHT - PLAYER_SIZE`, which is the square's Y coordinate at the bottom of the window.

The last few lines of code stop the square from moving any faster left or right once our square has jumped in the air. **Figure 2** shows the effects of X velocity on your jump distance.

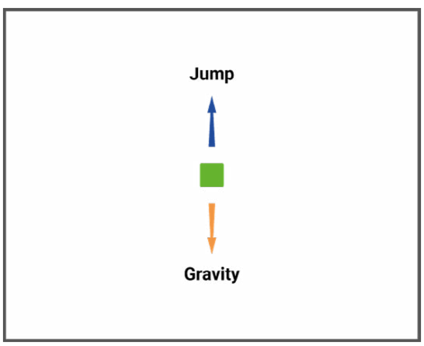


Figure 1: Gravity working against the square's Y velocity

```
    # Move the player along the y axis
    player_y -= player_vy

    # Don't let the player fall through the
floor
    player_y = min(player_y, WIN_HEIGHT -
PLAYER_SIZE)

    # Increase x velocity if we're moving but
not at maximum.
    if direction and abs(player_vx) < MAX_VX:
        # But only if we're not in the air!
        if player_y >= WIN_HEIGHT - PLAYER_SIZE:
            player_vx = player_vx * 1.1
```

How do we know which key our player pressed? Every Pygame key event has a `key` property that describes which key it represents. If we were to print out the `event.key` property, we would see a lot of numbers, but these aren't the keys that the player pressed. The numbers we would see are key codes; they're numbers that are uniquely tied to each key on your keyboard, and programmers can use them to check which keys they represent. For example, the **ESC** key on your keyboard is 27, the **A** key is 97, and **RETURN** is 13. Does this mean that we have to remember a seemingly disconnected bunch of numbers when we're writing keyboard code? Fortunately, the answer is no. Pygame has a ton of constants for checking key codes, which are easier to read and remember when we're writing code.

If the player presses the **ESC** key (`pygame.K_ESCAPE`), we quit the game. If they press **UP ARROW** (`pygame.K_UP`), we set the `jump` variable to `True`. The events are arranged in the list in the order that Pygame received them. So, for example, if we wanted to use the keyboard events to type in our player's name, we could trust that we would get all of the letters in the right order and not just a random jumble of characters.

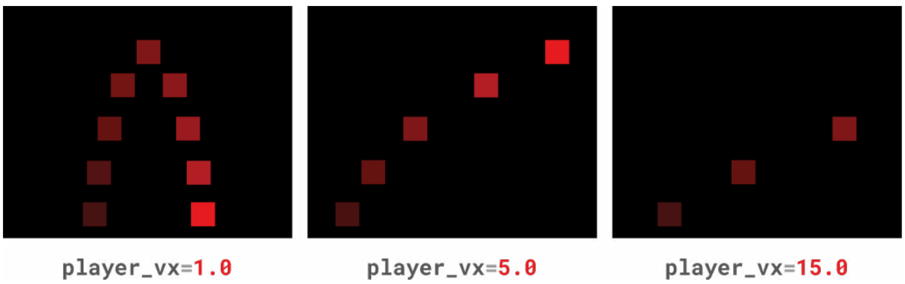


Figure 2: The varying effects of the X velocity when jumping

Processing events

The main loop of every Pygame program in this series is one big `while True` loop that keeps on running forever or until we exit the program. Each time our `while` loop runs, we call `pygame.event.get()` to get a list of events that have occurred since the last time the `while` loop ran. This includes system events, like a `QUIT` signal; mouse events, such as a left button click; and keyboard events, like when a button is pressed or released. Once we have the list of events that Pygame received, we iterate over it using a `for` loop and can decide how our program should respond to those events.

> *Pygame has a ton of constants for checking key codes*

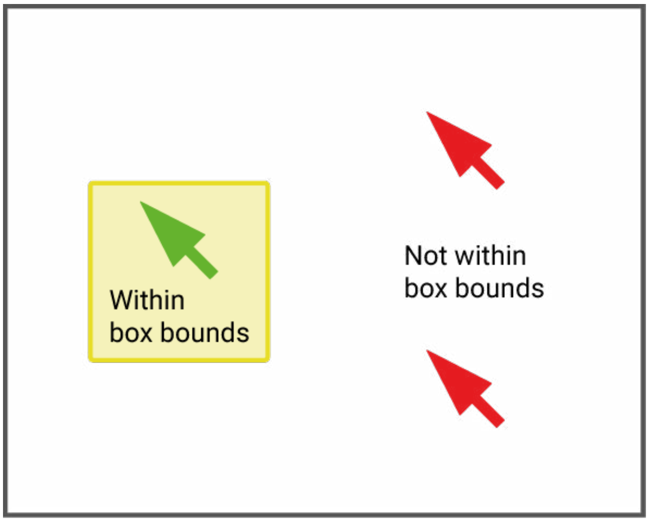
QUICK TIP

Pygame has a set of handy built-in variables for checking which keys are pressed. We've only used a couple, but you can find the complete list at **rpimag.co/pgkeyvars**.

Manic Miner is an early example of a game with a jumping character that follows gravity

Right after that loop, the code gets a list of pressed keys: if you press **LEFT ARROW**, it calls the `move()` function with a direction of -1 and the `jump` variable. If you press **RIGHT ARROW**, it sends a direction of 1 instead. And if neither are pressed, it sends a direction of 0. Why two ways of checking for key presses? The answer is somewhat simple: if we had used `pressed_keys` to check whether **UP ARROW** was pressed, the player would jump repeatedly while you hold that key down. By looking only for the `KEYDOWN` event, you get just one jump per key press. In contrast, you want the player to keep moving while you hold down the **RIGHT ARROW** or **LEFT ARROW** keys, and if you look in `pressed_keys` for a key, it will keep returning `True` as long as you hold the key down.

```
# How to quit our program
def quit_game():
    pygame.quit()
    raise SystemExit

while True:
    surface.fill((0,0,0))

    jump = False

    # Get all events since the last redraw
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                quit_game()
            if event.key == pygame.K_UP:
                jump = True

        if event.type == pygame.QUIT:
            quit_game()

    pressed_keys = pygame.key.get_pressed()
    if pressed_keys[pygame.K_LEFT]:
        move(-1, jump)
    elif pressed_keys[pygame.K_RIGHT]:
        move(1, jump)
    else:
        move(0, jump)

    pygame.draw.rect(surface, (255,0,0),
                     (player_x, player_y,
                      PLAYER_SIZE, PLAYER_SIZE))
    pygame.display.update()
    clock.tick(FPS)
```



▲ **Figure 3:** Checking the box bounds against the cursor coordinates

Pygame mouse input

That's enough of the keyboard for now; it's time for the mouse to shine. The mouse is a simple bit of kit, so the code for it is far less complicated than our keyboard code. If you run **mouse.py**, you'll see a familiar red square sitting at the bottom of the screen. Pressing your keyboard keys will do nothing this time, for this square is different. If you want to move it, you've got to use the mouse to pick it up. Drag your mouse over the square, hold down the left mouse button and drag up. Our square moves with our mouse. If you let go of your mouse button, the square will fall back to the bottom of the window. Nice and simple, but how does it work? We start with our usual setup and initialisation followed by constants and variables.

```
import pygame

pygame.init()
clock = pygame.time.Clock()
FPS = 60

WIN_WIDTH = 800
WIN_HEIGHT = 800
window = pygame.display.set_mode((WIN_WIDTH,
WIN_HEIGHT))
pygame.display.set_caption('Pygame Mouse!')

SQUARE_SIZE = 40
square_x = WIN_WIDTH / 2
square_y = WIN_HEIGHT - SQUARE_SIZE
gravity = 2.0
pressed = False
is_dragging = False
```

This time, we have hardly any code at all in our main `for` loop. Most of the work is handled by four functions.

Checking the square

The purpose of `check_bounds()` is to check whether or not our mouse position is within the bounds (edges) of our square (see **Figure 3**), and if the left button is pressed, to let other functions know that we're dragging the square. The `is_dragging` global variable indicates whether this is the case. If we were making a fully-fledged game, this function would probably check the position of every game object against the mouse coordinates, but in this example, we're only interested in our red square.

If the mouse is currently pressed, the function first defines a rectangle, `box`, that corresponds to the box's current location. Next, it uses `box`'s `collidepoint()` function to determine whether the box overlaps with the result of `pygame.mouse`'s `get_pos()` function. When we call `get_pos()` we get a tuple back with two values: the current X and Y value of the tip of the mouse pointer inside the window. The `collidepoint()` function tells us whether there's a collision between the mouse and box. In later parts of this series, you'll see other ways to determine whether objects on screen are touching each other.

QUICK TIP

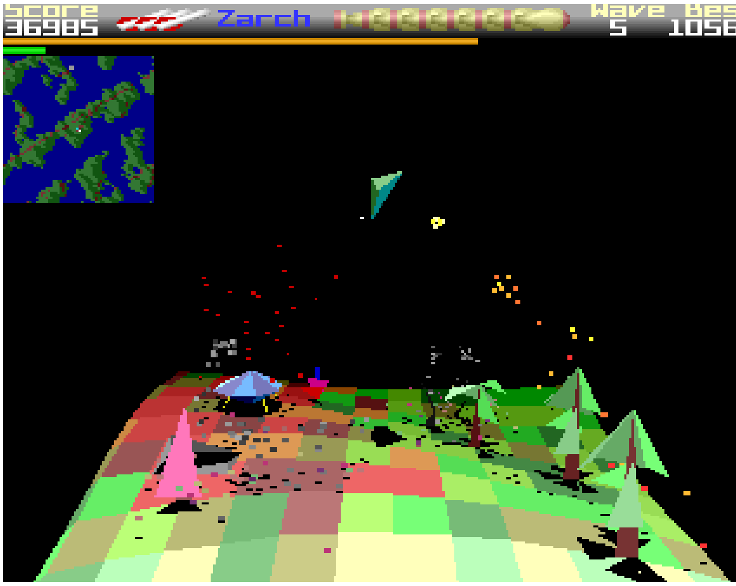
The X and Y coordinates of a mouse are relative to the left and top of the window, not the computer screen.

Now that we know our mouse is positioned within our square and that we've pressed our mouse button, we can set our `is_dragging` variable to `True` and hide the mouse cursor. Once we stop dragging it, we set `is_dragging` to `False` and show the cursor.

```
def check_bounds():
    global is_dragging

    if pressed:
        box = pygame.Rect((square_x, square_y,
                       SQUARE_SIZE, SQUARE_SIZE))
        if box.collidepoint(pygame.mouse.get_
pos()):
            is_dragging = True
            pygame.mouse.set_visible(0)
    else:
        is_dragging = False
        pygame.mouse.set_visible(1)
```

Once `check_bounds()` has done its job, `check_gravity()` gets to work. Just as we did in **keyboard.py**, `check_gravity()` looks at where our square is in the window: if it's not on the bottom, it will accelerate our square to get there. However, it will only do this if we've let go of our mouse button, because we don't want our shape to fall to the ground when we're holding onto it.

The next function is `draw_square()`: its purpose is easy enough to guess. Based on the adjustments of `check_bounds()` and `check_gravity()`, `draw_square()` will draw the square for us. If our square is being moved around by our mouse, it will draw the square at the mouse coordinates. But if we aren't dragging the square around, it will draw a graceful gravity-driven descent back to the bottom of our window. `draw_square()` has one little trick up its sleeve: as well as affecting the position of our square, it changes its colour: red when not being dragged and green when being dragged. This code could be useful if, instead of a square, we had a character and we wanted to change its graphic to make it look like it was holding onto our cursor. After that, we come to the `quit_game()` function that exits the game.



▲ David Braben's Zarch was one of the first games to feature mouse control

Congratulations! You now have the beginnings of a real game

```
def check_gravity():
    global gravity, square_y

    # Is our square in the air
    if square_y < WIN_HEIGHT - SQUARE_SIZE:
        if not is_dragging: # have we let go of
it?
            square_y += gravity
            gravity = gravity * 1.05
    else:
        square_y = WIN_HEIGHT - SQUARE_SIZE
        gravity = 2.0

def draw_square():
    global square_x, square_y

    if is_dragging:
        square_colour = (0, 255, 0)
        mouse_pos = pygame.mouse.get_pos()
        square_x = mouse_pos[0] - SQUARE_SIZE / 2
        square_y = mouse_pos[1] - SQUARE_SIZE / 2
    else:
        square_colour = (255,0,0)

    pygame.draw.rect(window, square_colour,
                     (square_x, square_y,
                      SQUARE_SIZE, SQUARE_SIZE))

def quit_game():
    pygame.quit()
    raise SystemExit
```

There's not a lot going on in the main loop; we fill the window with a black background, check to see whether the user wants to quit, then find out whether the mouse button has been pressed. After that, we call three functions before updating the screen: `check_bounds()`, `check_gravity()`, and `draw_square()`. Then we update the display and tick the clock.

```
while True:
    window.fill((0,0,0))

    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                quit_game()
        if event.type == pygame.QUIT:
            quit_game()

    pressed = pygame.mouse.get_pressed()[0]

    check_bounds()
    check_gravity()
    draw_square()

    pygame.display.update()
    clock.tick(FPS)
```

The two important things we need to know when using a mouse are where it is and which buttons, if any, have been pressed. Once we know these two things, we can begin to make things happen. In the main loop, we need to determine whether any of the buttons have been pressed; we do this on with `pygame.mouse.get_pressed()`, which returns a tuple of three values: the first is for the left mouse button, the second for the middle mouse button, and the third for the right mouse button. If the button is pressed down, then the value is `True`, otherwise it's `False`. We're not doing anything with the middle or right mouse button, so we can simply check the first value (the left mouse button) with `pygame.mouse.get_pressed()[0]`. If `pygame.mouse.get_pressed()[0]` is `True`, then our player has clicked the button, and we can proceed. In this case we set `pressed` to `True`. We declared it at the top of the program, so we can read its value in any function, and read (and set) its value in the main program. Give the game a try. Can you drop the square and catch it again before it hits the floor?

Pygame gamepad input

It's not difficult at all to add support for gamepad/joystick input. Open **keyboard.py**, and save a copy as **joystick.py**. You'll need to check whether a joystick exists and create a variable to represent it (as well a helper variable). Add the following before your main loop (just before `while True:`):

```
joystick = None
joy_threshold = 0.05
pygame.joystick.init()
if pygame.joystick.get_count() > 0:
    joystick = pygame.joystick.Joystick(0)
```

Next, replace everything between `jump = False` and the call to `pygame.draw.rect()` with the following. This code will use the joystick if one is detected (otherwise, it uses the keyboard):

```
    # Get all events since the last redraw
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                quit_game()
            if event.key == pygame.K_UP:
                jump = True

        if event.type == pygame.JOYBUTTONDOWN:
            jump = True

        if event.type == pygame.QUIT:
            quit_game()

    if joystick:
        x_axis = joystick.get_axis(0)
        if abs(x_axis) <= joy_threshold:
            move(0,jump)
        elif x_axis > joy_threshold:
            move(1, jump)
        elif x_axis <= -joy_threshold:
            move(-1, jump)
    else:
        pressed_keys = pygame.key.get_pressed()
        if pressed_keys[pygame.K_LEFT]:
            move(-1, jump)
        elif pressed_keys[pygame.K_RIGHT]:
            move(1, jump)
        else:
            move(0, jump)
```

DOWNLOAD THE FULL CODE:
rpimag.co/makegamescode

The joystick `get_axis()` function lets you read the position of an analogue joystick axis. It returns a value of approximately zero when nothing is pressed (in other words, when the joystick is centred), a value between zero and 1 for one direction, and between negative 1 and zero for the other. Axis 0 represents the X axis; you can use `joystick.get_axis(1)` for the Y axis. If you find that your square is moving even when you're not pressing the joystick, try increasing the value of `joy_threshold` to 0.10 or higher.

What you've learned

You've learned that Pygame creates a list of events that occurred every time the frame is updated, and that you can work through them to check for events that you want to use. You also learned that you can get a list of all the currently pressed keys without needing to poll for events. You learned that Pygame receives key codes when buttons are pressed, but has a big list of key code events that you can use so you don't have to remember all of the numbers. You learned that you can get mouse events whenever you like, and that you can get coordinates of where the mouse is and which buttons are pressed. You've also learned how to simulate gravity and jumping, and you've thought about how things move in the real world too. Congratulations! You now have the beginnings of a real game.

Part 03

GPIO Zero: user input with a push button

Make things happen at the press of a button, and create a fun two-player reaction game

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

Raspberry Pi's GPIO pins aren't just for outputs like LEDs – they can connect to inputs, too. Consider the simple push button, which can be used to trigger other components or functions. Let's connect a button and write a program to print a message when it's pushed. After that, we'll use it to light an LED, then add a second button for a fun two-player reaction game.

Connect the button

Shut down your Raspberry Pi and unplug it from power before building your circuit. Connect the LED as shown in **Figure 1**, remembering to add a resistor for its ground connection. Add the push button to the breadboard as shown in the diagram, with its pins straddling the central groove. Connect a pin-to-socket jumper wire from one pin's column to GPIO 21 on the GPIO header. Then connect a pin-to-pin jumper wire from the other column (on either side of the groove) to the '–' (ground) rail. Finally, connect a pin-to-socket jumper wire from the ground rail to a GND pin on the GPIO header.

Button pushed

We'll now test our circuit with a simple Python program to show a message on the screen whenever the button is pushed. Create a new file with the following code, then save it as **button.py** (see Part 02 of this guide for an overview of editing and running code):

YOU'LL NEED:

- 1× solderless breadboard
- 2× push buttons
- 1× LED
- 1× 330Ω resistor
- 4× pin-to-socket jumper wires
- 2× pin-to-pin jumper wires

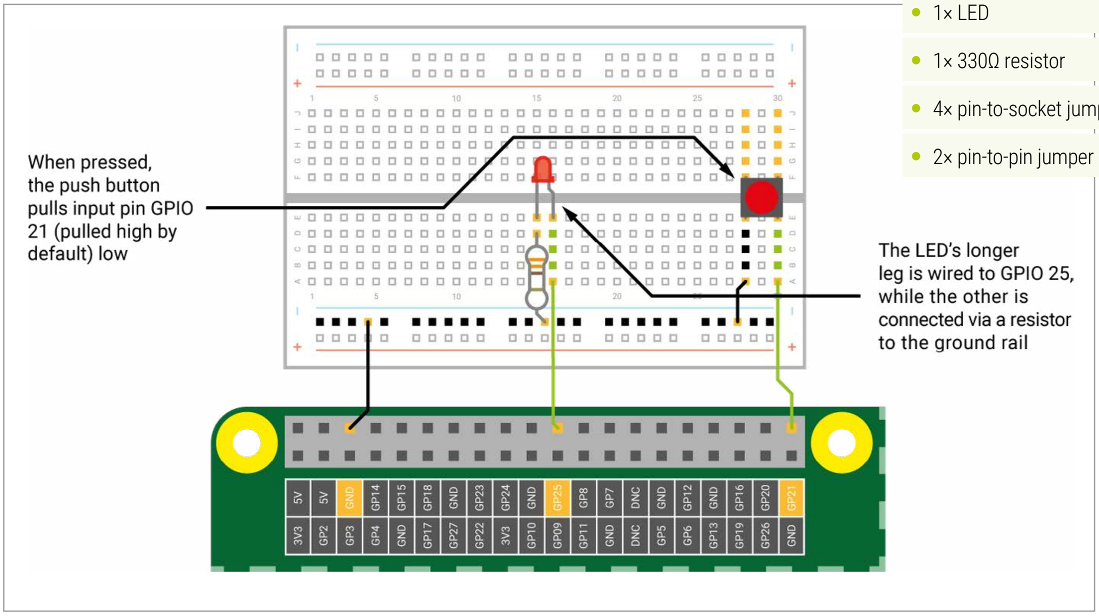


Figure 1: Wiring up a button

```
from gpiozero import Button
button = Button(21)
while True:
    if button.is_pressed:
        print("Button is pressed")
    else:
        print("Button is not pressed")
```

At the start of this short program, we import the `Button` class from GPIO Zero. We then set the `button` variable to the GPIO 21 pin, so we can read its value. Finally, we use `while True:` to create a never-ending loop that checks whether the button has been pressed or not and prints a status message on the screen. When you run the code, you'll get a scrolling list of messages that change when you press the button. To stop the program, press **CTRL+C**. The assembled project on a breadboard is shown in **Figure 2** (overleaf).

You can also trigger a Python function when the button is pressed, as shown in the following example. Note that we don't use parentheses when assigning the function name to the `when_pressed` event. Unlike the previous example, this will only print text when you press the button. Save this as **button_func.py** and run it. You can exit it with **CTRL+C**.

```
from gpiozero import Button
from signal import pause
button = Button(21)

def button_pressed():
    print("Button was pressed")

button.when_pressed = button_pressed
pause()
```

Wait for it

GPIO Zero's `Button` class also includes a `wait_for_press` method which pauses the script until the button is pressed. Create a new file, enter the following code and save it as **button_wait.py**. This will only print the message at the bottom on the screen once the button has been pressed, and the program will exit immediately after you press the button.

```
from gpiozero import Button
button = Button(21)
button.wait_for_press()
print("Button was pressed")
```

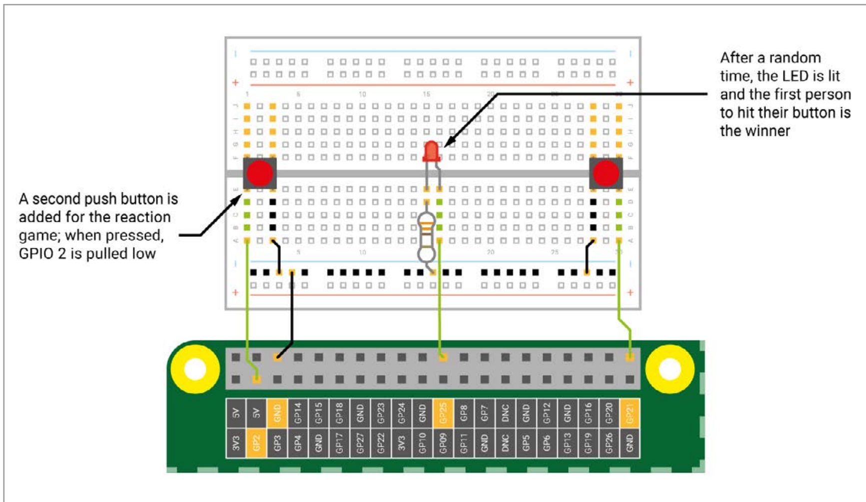
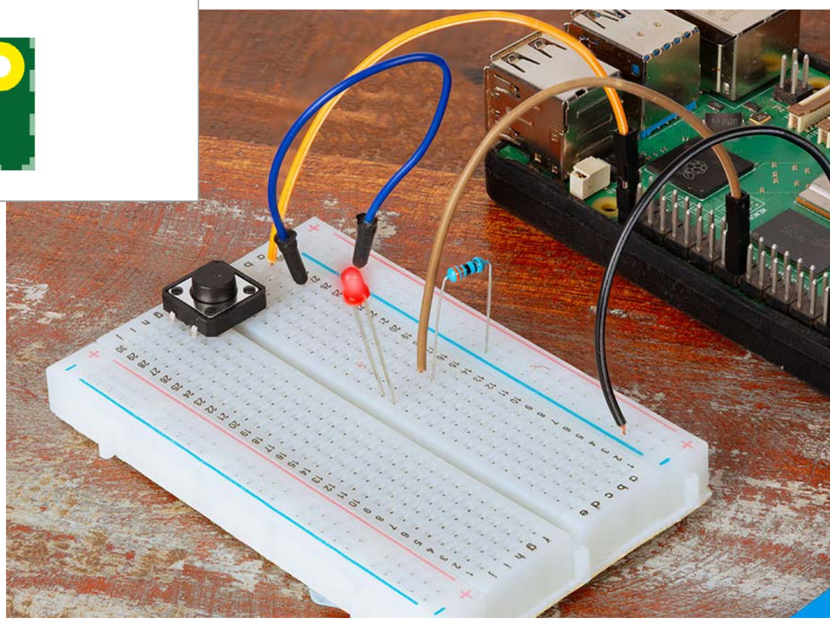


Figure 3: There are now two buttons and an LED for our reaction game

Figure 2: When the button is pressed, GPIO 21 registers the low signal and our program turns the LED on



Light an LED

Let's try using the button to control the LED. Create a new file, enter the following code, and save it as **button_led.py**.

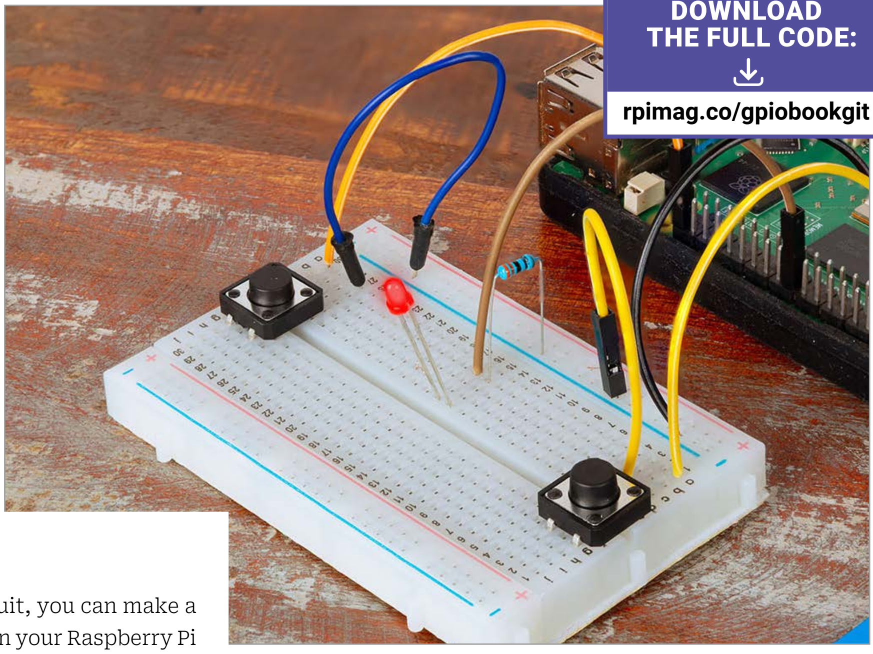
```
from gpiozero import LED, Button
from signal import pause

led = LED(25)
button = Button(21)

button.when_pressed = led.on
button.when_released = led.off
pause()
```

At the top, we import the `LED` and `Button` classes from GPIO Zero, along with the `pause` function from `signal`. We then allocate variables to the LED and button on GPIOs 25 and 21 respectively. When the button is pressed, the LED is turned on; when released, it's turned off.

It's also possible to keep the LED lit for a set period after pressing. Open a new file, enter the following code and save it as **button_off_delay.py**. This time, we trigger a function when the button is pressed as before, but we also set up another function that's triggered when the button is released – it waits three seconds before turning the LED off. As with other programs that use a `pause` or infinite `while` loop, you'll need to stop the program with **CTRL+C** to exit it.

```
from gpiozero import LED, Button
from time import sleep
from signal import pause

led = LED(25)
button = Button(21)

def button_released():
    sleep(3)
    led.off()

button.when_pressed = led.on
button.when_released = button_released
pause()
```

DOWNLOAD THE FULL CODE:
rpimag.co/gpiobookgit



▲ **Figure 4:** The LED is lit! In this reaction game, the first person to now press their button will win

Reaction game

If you add a second push button to the circuit, you can make a simple two-player reaction game. Shut down your Raspberry Pi and disconnect it from power. Add a second button as shown in **Figure 3**, connecting it to the ground rail and GPIO 2; move the LED and its connections to the middle, if not there already.

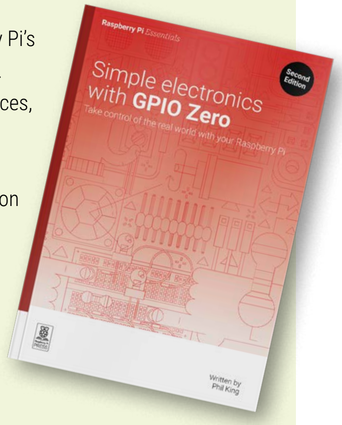
Now you're ready to plug your Raspberry Pi in and boot it back up. Create a new file, enter the following code, and save it as **reaction_game.py**:

```
from gpiozero import Button, LED
from time import sleep
import random
led = LED(25)
player_1 = Button(21)
player_2 = Button(2)
time = random.uniform(5, 10)
sleep(time)
led.on()
while True:
    if player_1.is_pressed:
        print("Player 1 wins!")
        break
    if player_2.is_pressed:
        print("Player 2 wins!")
        break
led.off()
```

As before, we import the classes required (along with the `random` module) at the top. We assign variables to the LED and two buttons, then create a `time` variable equal to a random number between 5 and 10; after sleeping for this number of seconds, the LED is turned on, as shown in **Figure 4**. The `while True:` loop is terminated with `break` when someone presses their button, after printing the appropriate victory message.

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

The Big Quiz of the Year 2025

Think you know your Raspberry Pi and computer technology? We're here to put that to the test.

By
Raspberry Pi Official Magazine team

Issue 149, January

It was a busy month for us with new products left, right, and centre. But how well do you remember the start of the year?

1

Raspberry Pi monitor was revealed (after a long internal development time). How big is its screen?

- ❑ **A.** 13.3 inches
- ❑ **B.** 14 inches
- ❑ **C.** 15.6 inches
- ❑ **D.** 17 inches

2

A Raspberry Pi board gained wireless functionality. But which one was it?

- ❑ **A.** Raspberry Pi Zero 2 W
- ❑ **B.** Pico 2 W
- ❑ **C.** Pico Wi-Fi Connect
- ❑ **D.** Pico Wow, no wires!

3

The team at Raspberry Pi got fed up with third-party hubs and decided to make their own. How many USB 3.0 ports does it feature?

- ❑ **A.** Just one for laughs.
- ❑ **B.** Three's a magic number.
- ❑ **C.** Four, of course.
- ❑ **D.** Lucky eight.

4

Raspberry Pi released a brand-new remote access service. What's it called again?

- ❑ **A.** Connect
- ❑ **B.** Attach
- ❑ **C.** Affix
- ❑ **D.** Conjoin

5

Raspberry Pi 500 was released as well. But how much RAM does it come with?

- ❑ **A.** 2GB is plenty, as we often say
- ❑ **B.** 4GB because Raspberry Pi OS is efficient
- ❑ **C.** 8GB because we love you
- ❑ **D.** 16GB, steady on now!

Issue 150, February

We sent out the magazine in style with 150 projects. The old name was much loved and still with us in spirit as we got ready to go official.

6

Somebody got 16GB RAM for the new year. But which Raspberry Pi got its upgrade first?

- ❑ **A.** Raspberry Pi 4
- ❑ **B.** Raspberry Pi 500
- ❑ **C.** Raspberry Pi 5
- ❑ **D.** Raspberry Pi Pico

7

A new project supports carbon removal. But how much does it cost you to mitigate the carbon footprint of a Raspberry Pi board?

- ❑ **A.** I'd buy that for a dollar
- ❑ **B.** A two-dollar word
- ❑ **C.** Three bucks a pop
- ❑ **D.** $4 and worth every cent

8

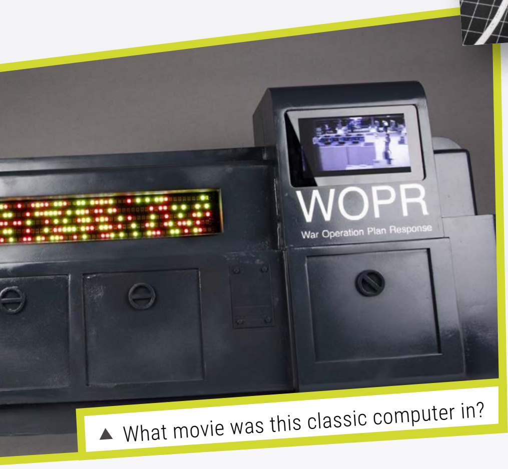
Toby recreated WOPR at Raspberry Pi towers. But what movie did the famous WOPR computer star in?

- ❑ **A.** 2001: A Space Odyssey
- ❑ **B.** WarGames
- ❑ **C.** The Terminator
- ❑ **D.** The Hitchhiker's Guide to the Galaxy

9

We covered a floppy disk archiver made by Graham Hooley to back up Amiga games. But who remembers what size floppy the Amiga used?

- ❑ **A.** 8-inch (go big or go home)
- ❑ **B.** 5.25-inch (as God intended)
- ❑ **C.** 3-inch (because Amstrad was right all along)
- ❑ **D.** 3.5-inch (and not in any way floppy)

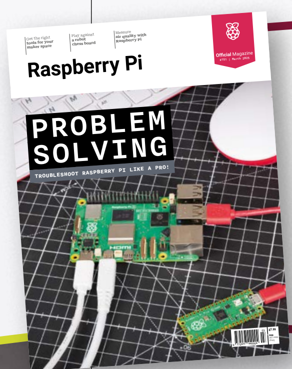


▲ What movie was this classic computer in?

10

It's usually used to detect earthquakes, but this month we saw a clever project use it to map caves! But what is the seismology technology called?

- ❑ **A.** Raspberry Pi Shake
- ❑ **B.** Raspberry Pi Rattle
- ❑ **C.** Raspberry Pi Roll
- ❑ **D.** Raspberry Pi Bash



Issue 151, March

A new look and a new name as we officially became *Raspberry Pi Official Magazine.*

11

Raspberry Pi revealed the results of its RP2350 hacking challenge. How much did each person get who found a security flaw?

- ❑ **A.** $500
- ❑ **B.** $1000
- ❑ **C.** $10,000
- ❑ **D.** $20,000

12

Oooh, Valve updated its video game service so you could stream games on Raspberry Pi this month. You're lucky to even get a magazine with that distraction. What's Valve's streaming software called?

- ❑ **A.** Stream Link
- ❑ **B.** Steam Link
- ❑ **C.** Scheme Link
- ❑ **D.** Spleen Link

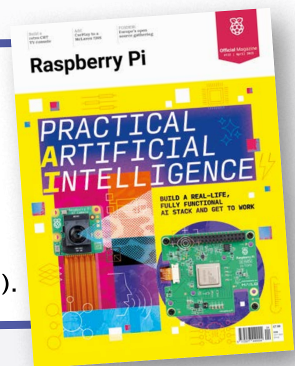
13

We started a new tutorial series on v2 of Raspberry Pi's most famous HAT. The one that went up to the ISS, you know. What's it called again?

- ❑ **A.** Sense HAT
- ❑ **B.** Sensibility HAT
- ❑ **C.** Space HAT
- ❑ **D.** Touch and play HAT

Issue 152, April

We got all into AI this month in a big way (well Lucy did, at least).



14

What is Raspberry Pi's smart camera called?

- ❑ **A.** Raspberry Pi AI Camera
- ❑ **B.** Raspberry Pi Smart Cam
- ❑ **C.** Raspberry Pi ANPR Cam
- ❑ **D.** Raspberry Pi Spy Cam

15

Raspberry Pi has an AI HAT+ as well. And how many TOPS does it provide?

- ❑ **A.** 2 or 4
- ❑ **B.** 13 or 26
- ❑ **C.** 69 dude
- ❑ **D.** 180

16

KG built a CRT emulation console this month, and we all went a bit misty-eyed. What's CRT stand for again?

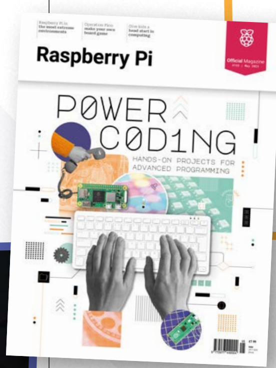
- ❑ **A.** Compact Raster Terminal
- ❑ **B.** Cathode Ray Tube
- ❑ **C.** Cosmic Radiation Technology
- ❑ **D.** Classic Round Television

▼ We miss round displays

17

Toby made us a 3D printed recreation of an extremely dangerous nuclear sphere from the Manhattan Project in 1946 and told us all about it (a bit too maniacally, if you don't mind us saying). What is the death-ball better known as?

- ❑ **A.** Satan's Core
- ❑ **B.** The Metal Core
- ❑ **C.** The Demon Core
- ❑ **D.** Rufus



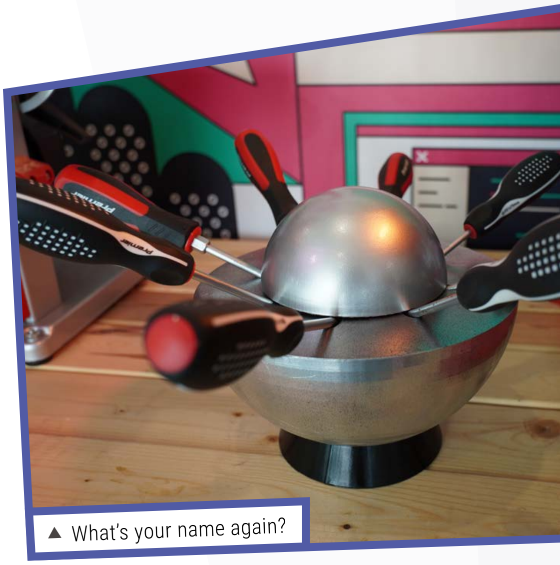
Issue 153, May

We got a spring in our step and ran a feature on power coding.

18

Raspberry Pi released a new power supply that's more powerful than any Raspberry Pi needs. Why? Because they make better power supplies than anybody else. What's its wattage?

- ❑ **A.** 45W
- ❑ **B.** 33W
- ❑ **C.** 27W
- ❑ **D.** 55W



▲ What's your name again?

19

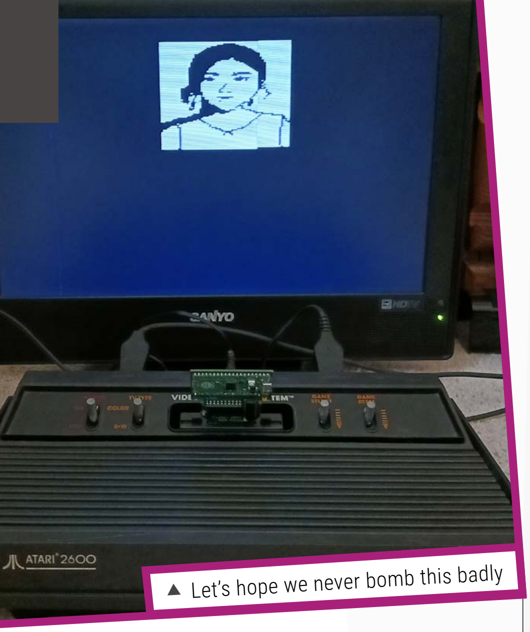
Raspberry Pi won the coveted TSMC Trophy for embedded computing. A real achievement. TSMC is probably the most important building on earth: so you know what it stands for, right?

- ❑ **A.** Tanzania Silicon & Manufacturing Cooperative
- ❑ **B.** Transistor Supply & Making Company
- ❑ **C.** Taiwan Semiconductor Manufacturing Company
- ❑ **D.** The Sprocket and MacGuffin Company

20

Let's start a fight. What's the best programming language?

- ❑ **A.** Python
- ❑ **B.** C
- ❑ **C.** Java
- ❑ **D.** Rust

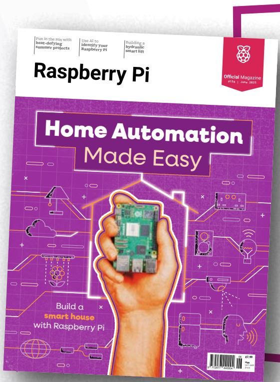


▲ Let's hope we never bomb this badly

21

More nostalgia as Ian fires up RISC OS on a Raspberry Pi 400. Which computer did RISC OS first appear on?

- ❑ **A.** BBC Micro Model B
- ❑ **B.** Acorn Electron
- ❑ **C.** Acorn Master System
- ❑ **D.** Acorn Archimedes



Issue 154, June

Home automation was made easy this month. We're still talking to our lights.

22

Raspberry Pi introduced a new soldering process that improved efficiency and reduced our carbon footprint. Clever stuff. Remember what it was called?

- ❑ **A.** Intrusive reflow
- ❑ **B.** Backslide 360
- ❑ **C.** Double tap
- ❑ **D.** Solder two-times

23

Nick Bild shared his Atari 2600-based digital photo frame this month. What Atari game was so fantastically, legendarily bad, that hundreds of thousands of copies ended up in landfill?

- ❑ **A.** The Goonies
- ❑ **B.** Back to the Future
- ❑ **C.** The Karate Kid
- ❑ **D.** E.T. the Extra-Terrestrial

24

Idar Rahkmutulin shared his PiEEG brain scanning project with us. What does EEG stand for?

- ❑ **A.** Enhancedencephalicgrid
- ❑ **B.** Electroencephalogram
- ❑ **C.** Electroelasticgradient
- ❑ **D.** Externalencodinggoose

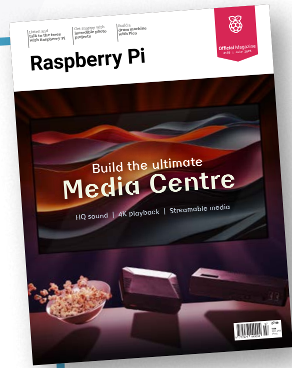
25

Rob Miles built a MIDI instrument based on Raspberry Pi Pico. What does MIDI stand for?

- ❑ **A.** Modular Interactive Data Interface
- ❑ **B.** Micro Instrument Device Integration
- ❑ **C.** Music Into Dramatic Input
- ❑ **D.** Musical Instrument Digital Interface

Issue 155, July

We showed you how to build the ultimate media centre this month. Sold well too. We should do that again sometime.



26

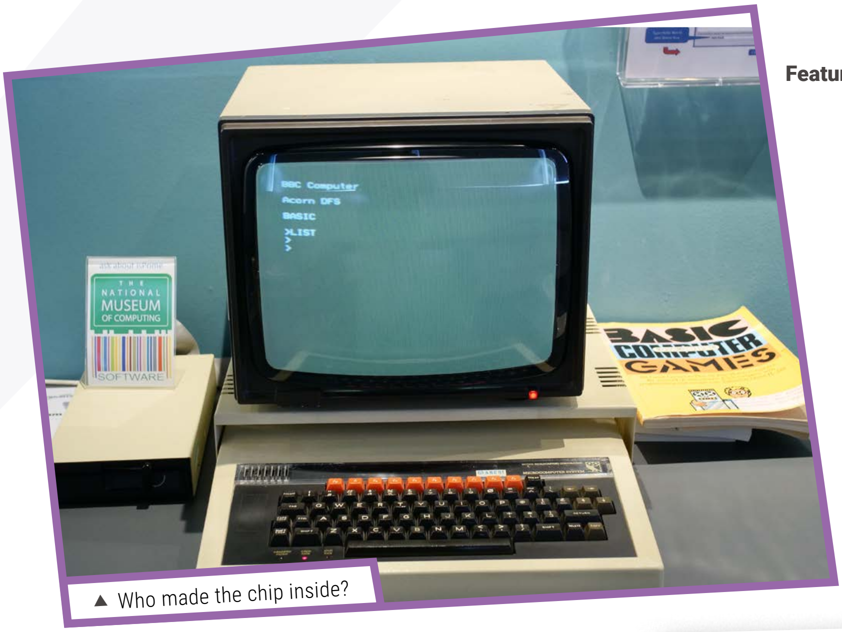
Richard Smedley's brilliant book, *Conquer the Command Line* entered its third-edition and we started running tutorials. The command line is based upon the GNU Project. A gnu is a large, hairy African wildebeest. What's GNU stand for, again?

- ❑ **A.** GNU's Not Unix!
- ❑ **B.** Good Neat Unix?
- ❑ **C.** Grand Network of Users~
- ❑ **D.** Generally Necessary Utils.

27

And let's not forget the importance of the man who built the Linux kernel alongside GNU, giving us a decent operating system at last. What's his name?

- ❑ **A.** Lars Ulrich
- ❑ **B.** Linus Sebastian
- ❑ **C.** Linus Torvalds
- ❑ **D.** Linus van Pelt



▲ Who made the chip inside?

28

And what distribution (or flavour) of Linux is Raspberry Pi OS based upon?

- ❏ **A.** openSUSE
- ❏ **B.** Fedora
- ❏ **C.** Arch Linux
- ❏ **D.** Debian

29

The versions of Raspberry Pi OS are named after characters from a famous animated movie series. Which one?

- ❏ **A.** Shrek
- ❏ **B.** Toy Story
- ❏ **C.** Despicable Me
- ❏ **D.** Kung Fu Panda

30

Last Linux question, we promise. What is the Linux mascot, a penguin, called?

- ❏ **A.** Tux
- ❏ **B.** Tax
- ❏ **C.** Tix
- ❏ **D.** Tex

31

Creative Mindstorms showed us how to build a 3D printer out of Lego. What's Raspberry Pi's HAT (now open source) that you can use to control Lego?

- ❏ **A.** Brick HAT
- ❏ **B.** Build HAT
- ❏ **C.** Brock HAT
- ❏ **D.** Best HAT

32

Our friend and former HackSpace guy, Ben Everard, returned to show us some advanced maker projects. He started by designing a PCB, as you do. What does PCB mean?

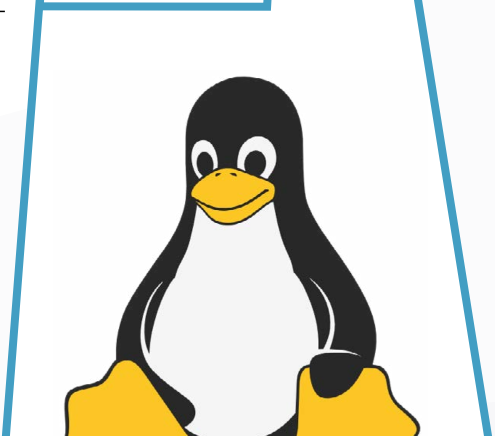
- ❏ **A.** Printed Chip Board
- ❏ **B.** Personal Computer Board
- ❏ **C.** Printed Circuit Board
- ❏ **D.** Professional Circuit Biscuit

33

Lucy reviewed the DreamHAT+ mm-wave radar. Which Scottish physician and engineer demonstrated the first working radar system in 1935?

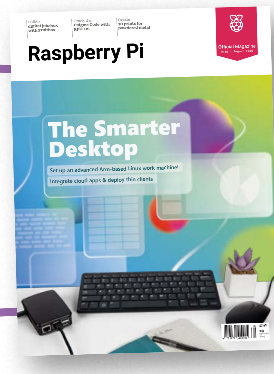
- ❏ **A.** Rupert Wilson-White
- ❏ **B.** Robert Watson-Watt
- ❏ **C.** Walter White-Watt
- ❏ **D.** Ronald Witterson-Witt

▼ Name that penguin



Issue 156, August

We set up Raspberry Pi as a smarter desktop this month.



34

A maker going by the nom de plume Rumbledethumps showed off his Picocomputer 6502. The 6502 was a massively important chip, used in the BBC Micro amongst countless others, but which company made it?

- ❏ **A.** MOS Technology
- ❏ **B.** Zilog
- ❏ **C.** Motorola
- ❏ **D.** Texas Instruments

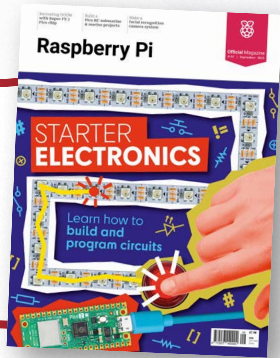
35

Phil King made a jukebox using Raspberry Pi and a custom piece of software. What was the software he used called?

- ❏ **A.** Meatbox
- ❏ **B.** Fishbox
- ❏ **C.** Vegbox
- ❏ **D.** Fruitbox

Issue 157, September

Starter electronics this month as we built and programmed circuits. What can you remember from this issue?



36

Andrew Lewis shares an unusual tool with us every month. This month was a little more unusual than most. But which of these is a real tool?

- ❑ **A.** Flange reassurance mallet
- ❑ **B.** Nylon bush installer
- ❑ **C.** Hole punching flange joggler
- ❑ **D.** Rubber gasket press

37

Elecrow makes an electronics playground and starter kit with Raspberry Pi and components. We reviewed the third version this month. What's it called?

- ❑ **A.** RookPi
- ❑ **B.** CrowPi
- ❑ **C.** StarlingPi
- ❑ **D.** RobinPi

38

What is the name of the Python library used to control Raspberry Pi's GPIO pins?

- ❑ **A.** GPIO Zero
- ❑ **B.** GPIO Hero
- ❑ **C.** GPIO Nero
- ❑ **D.** GPIO Giro

39

Raspberry Pi released a new stepping for RP2350 to address the issues found during our hacking challenge. The stepping shares its name with a paper size (no relation though). What is it called?

- ❑ **A.** RP2350 A5
- ❑ **B.** RP2350 D2
- ❑ **C.** RP2350 C5
- ❑ **D.** RP2350 A4

40

The Supersense SenS2 dementia aid impressed Rosie Hattersley with its unique privacy-protecting solution to monitoring elderly relatives. What technology does it use?

- ❑ **A.** Electrochemical
- ❑ **B.** Radar
- ❑ **C.** Vibration
- ❑ **D.** Magnetic field

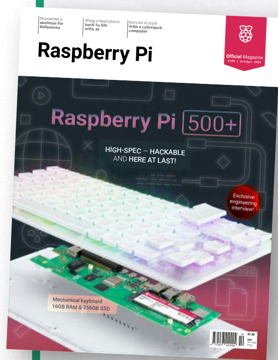
41

Rob looked at a DOOM 2025 project that uses RP2350 to recreate the Super FX chip found for games in which console?

- ❑ **A.** SNES
- ❑ **B.** Sega Mega Drive/Genesis
- ❑ **C.** Sony PlayStation
- ❑ **D.** Apple Pippin

Issue 158, October

After a long internal wait, the wonderful Raspberry Pi 500+ was finally able to show its face. Don't you just love that keyboard?



42

Our smaller Touch Display 2 was also revealed this month. How small is its display?

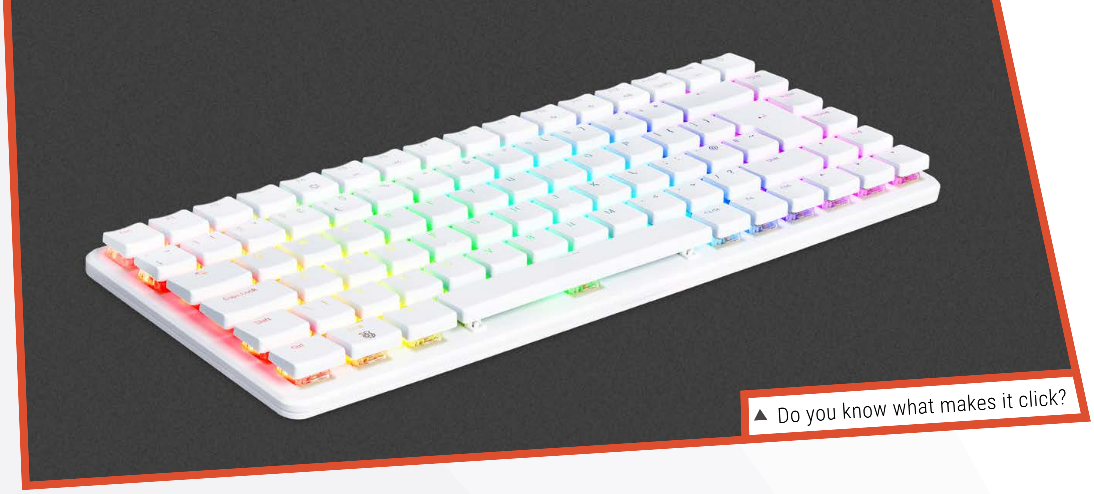
- ❑ **A.** I-inch
- ❑ **B.** 3-inch
- ❑ **C.** 5-inch
- ❑ **D.** 6-inch

43

Elvis Andrés Ayala talked to us about Project Trinidad, a project to send small satellites into space. What are these satellites known as?

- ❑ **A.** BoxSats
- ❑ **B.** CubeSats
- ❑ **C.** SpaceCubes
- ❑ **D.** SubeCats

▼ Get in touch with our display

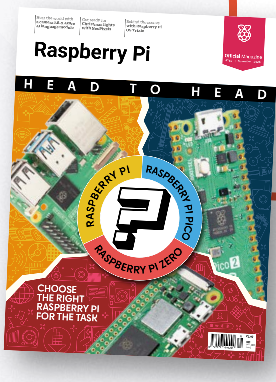


▲ Do you know what makes it click?

44

Spacerower showed us his Instant Camera that uses a printer to print out stickers. Which company invented the first instant camera?

- ❑ **A.** Polaroid
- ❑ **B.** Kodak
- ❑ **C.** Canon
- ❑ **D.** Eastman



Issue 159, November

We put all the Raspberry Pi computers in a head-to-head competition to figure out which is best. Only kidding, they're all great.

45

Mehrdad Mazjoobi showed us his Ubo Pod AI, a voice-controlled assistant. In 1961, Bell Labs demonstrated the first ever voice synthesizer. What song did it sing?

- ❑ **A.** Daisy Bell
- ❑ **B.** Jingle Bells
- ❑ **C.** Silver Bells
- ❑ **D.** Hell's Bells

46

Speaking of Raspberry Pi 500+, what is the name of the keyboard switches that give it that clicky sound?

- ❑ **A.** Kailh Box White
- ❑ **B.** Holy Panda
- ❑ **C.** Oil Kings
- ❑ **D.** Gateron Blue

47

We introduced a new flavour of Raspberry Pi OS. What is this one called?

- ❑ **A.** Chuckles
- ❑ **B.** Trixie
- ❑ **C.** Forky
- ❑ **D.** Zurg

48

We interviewed Grant Sinclair. His uncle, Sir Clive Sinclair, remains in the hearts of many UK computer enthusiasts. His first breakthrough, way before the Spectrum, was a slimline calculator. What was it called?

- ❑ **A.** Sinclair Executive
- ❑ **B.** Sinclair Businessman
- ❑ **C.** Sinclair Bossman
- ❑ **D.** Sinclair Supervisor

49

We ran a chapter of Tim Danton's incredible *The Computers That Made The World* book, this time on the Complex Number Calculator. Which lesser-known genius built this one?

- ❑ **A.** Howard Aiken and Grace Hopper at Harvard University
- ❑ **B.** John Atanasoff and Clifford Berry at Iowa State College
- ❑ **C.** George Robert Stibitz and Samuel Williams at Bell Labs
- ❑ **D.** Reginald Babblesworth at Stanford University

50

Last, but certainly not least. What did this fine magazine used to be called?

- ❑ **A.** MagBerry
- ❑ **B.** RaspiMag
- ❑ **C.** PiMagz
- ❑ **D.** The MagPi

Answers: 1C | 2B | 3C | **4**A | 5C | 6C | 7D | 8B | 9D | **10**A | **11**D | 12B | 13A | **14**A | 15B | 16B | **17**C (and D, its original nickname) | **18**A | 19C | **20**A, B, C, D (trick question, all coders get a point) | **21**D | **22**A | **23**D | **24**B | 25D | **26**A | **27**C | **28**D | 29B | 30A | 31B | 32C | 33B | 34A | 35D | 36C | 37B | 38A | 39D | 40B | 41A | 42C | 43B | **44**A | **45**A | **46**D | **47**B (Forky's next btw) | **48**A | **49**C | **50**D (never forgotten)

Powered by Raspberry Pi

There are now hundreds of products with Raspberry Pi in one form or another at their centre. This includes consumer kit that promises exciting new project features, HATs and accessories for both hobbyist and industrial use, and specialist hardware versions with Compute Module at the heart of their DNA. The Powered by Raspberry Pi badge of approval helps assure you that a product has been thoroughly tested and is guaranteed to work flawlessly with Raspberry Pi computers and microcontrollers. This issue, we feature half a dozen products from around the world that are helping improve driver and passenger safety, drone pilots' chances of a successful landing, and marine pilots' navigation accuracy. We've also got treats for fans of vintage computers and gaming and AI photography.

rpimag.co/poweredbyraspberrypi

APPLY TO POWERED BY RASPBERRY PI

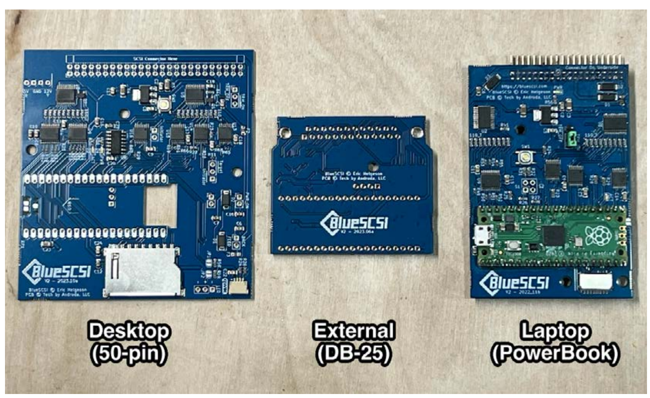
Our Powered by Raspberry Pi logo shows your customers that your product is powered by our high-quality Raspberry Pi computers and microcontrollers. All Powered by Raspberry Pi products are eligible to appear in our online gallery.

rpimag.co/poweredbypiapply

BlueSCSI

USA | bluescsi.com

Many of us love embracing older technology to enjoy games and programming experiences from a decade or three ago. Inevitably, the storage formats of the 1980s and 1990s have long been superseded along with the drivers written to work with them. That doesn't mean you can't run older programs, of course. Emulators for well known home computers are incredibly popular. BlueSCSI offers a neat way to access games, applications, and files hidden away on otherwise obsolete external drives so you can enjoy them all over again. This modern, open-source solution replaces your old SCSI drives – including CD-ROM and magneto-optical – with a simple and reliable SD card, offering a fantastic upgrade for your classic Mac, Amiga, Atari and more!



Candera CGI Studio Professional

Austria | cgistudio.at

Any full-size Raspberry Pi computer can be used to run Candera's CGI Studio Professional HMI (human-machine interface). Its rapid design tools are custom-made for small-to-medium-sized businesses and include an invaluable Scene Composer and pre-built players for Linux-based devices. Certified for Raspberry Pi, CGI Studio Pro offers Python-scripting with data model access, making it ideal for designing user interfaces and customer menus for any number of applications. The impressively capable design tool is a popular choice in the automotive field. Version 3.15, launched in spring 2025, extends the IntuitiveHMI design suite with simplified workflows, improved graphics, and added AI options including SoundHound voice recognition for designers creating interfaces across automotive, medical, and other industries.

Clickdrive

Singapore | clickdrive.io

Keeping with the vehicular theme, Clickdrive is a driving training system aimed at both driving schools and public transport companies who have found it invaluable in improving staff retention rates. A self-install kit with wired and wireless options, GPS, and HD video camera, Clickdrive provides a form of real world training via a combination of videos. This includes the option for bespoke instructor clips, GPS and motion sensors for location accuracy, object detection, and performance analysis. While driving games and simulators focus on overcoming obstacles and taking turns at high speed, Clickdrive records routes driven for self-improvement rather than fun, using customisable training programmes. The Singapore-based company has a roster of satisfied clients including the city's own SBS Transit authority and other public transport companies. The Clickdrive Pro system provides 360-degree video feedback alongside objective driving telemetry analysis so drivers can receive individual post-drive reviews and tailored improvement advice.

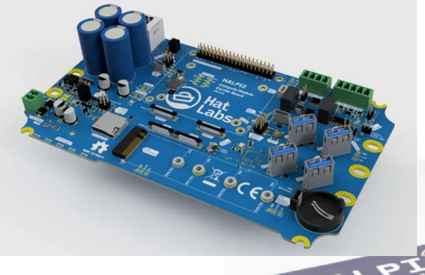
Landmark Precision Landing System

USA | landmarklanding.com

Flying machines have long caught the imagination of amateur pilots, so when drones arrived on the scene, their success was little surprise. If you're anything like us, though, the joy of seeing your craft aloft is tinged by anxiety about the seemingly inevitable sudden descent back and potential curtailment of a new hobby. Landmark specialises in helping PX4 and ArduPilot drone and model aircraft pilots achieve precision landings time after time. (OK, the clue's in the company name.) Promising centimetre-level landing accuracy, the system works in various lighting conditions, including direct sunlight and at night (with target illumination). The landing module attaches to your Raspberry Pi via a single cable, while a ground station such as Mission Planner or QGround Control is used for all configuration.

Hat Labs HALPI2

Finland | hatlabs.fi



Raspberry Pi's Compute Module with its industrial-grade specification is becoming an increasingly popular choice for marine applications. Finland's Hat Labs is a long-established open source and open hardware marine specialist. As well as being a keen sailor, founder Matti is an IoT veteran with many years' experience with CAN bus and NMEA 2000 products. The Helsinki based firm's HALPI2 is a marine plotting platform based around CM5 and an ITX motherboard in a custom-designed, prebuilt, fully functional Raspberry Pi boat computer protected within a waterproof and ruggedised case. HALPI2 plots and tracks routes and acts as a data acquisition and visualisation device, providing a large degree of boat automation and control.

EDATEC CM5 AI Camera Series

China | edatec.cn

EDATEC makes robust hardware based on open-source principles using equipment including Raspberry Pi. Emerging from the management team at industrial supplier Farnell in 2017, EDATEC was among the very first to recognise Raspberry Pi's potential as a modular industrial platform – and one of the first to gain Powered by Raspberry Pi accreditation. The 12MP ED-AIC3100 uses Compute Module 4 with its 64-bit SoC platform to power and control a quad-core AI camera with a 12mm autofocus liquid lens and a C-Mount lens. The 3100 series camera is protected by a bright blue IP65 shockproof metal case that can withstand temperature variations of 0~45°C and has a mounting bracket to absorb vibrations. Running 64-bit Raspberry Pi OS, the AI camera weighs just 400g and can be triggered with a single button press or remotely, acquiring and processing images at 70 frames per second before efficiently making sense of their contents.

ONLY THE **BEST**

Control devices

By **Phil King**

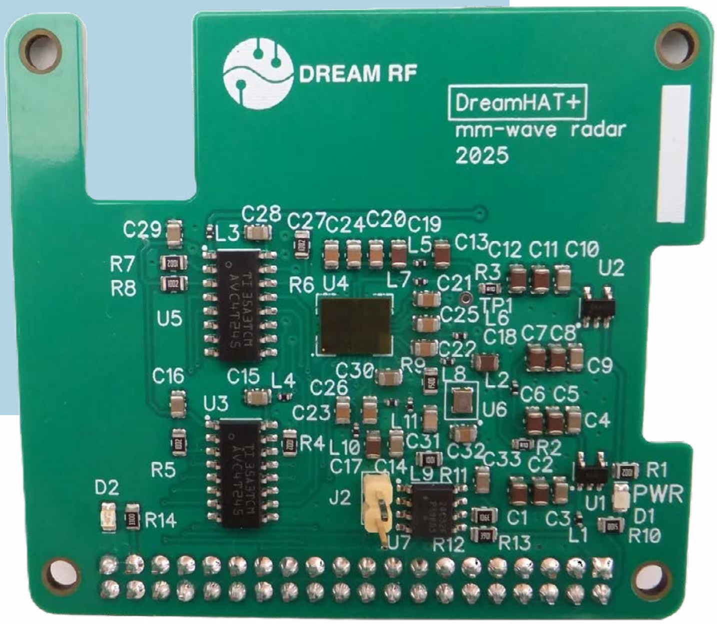
While Raspberry Pi is often used with a standard keyboard and mouse as input devices, there are many other possible control methods out there. Electronics projects typically make use of push-buttons and rotary potentiometers, but there are alternatives that might well be more suitable for your project, such as capacitive touch sensing.

USB or Bluetooth joysticks and gamepads are ideal for playing games, but may also come in useful for controlling projects such as robotic arms. While many only have digital direction sensing, we take a look at an analogue joystick offering far more nuanced 360° control.

Gesture detection is another option. One way is to use a camera with AI analysing the view to detect gestures or poses. Another is offered by the DreamHAT+ with its mm-wave radar. We also explore tilt control with the IMU on the Sense HAT, along with a programmable Pico-powered keypad with RGB lighting.

DreamHAT+

DreamRF | £100 / $135 | dreamboards.co.uk



▲ Equip Raspberry Pi with 60GHz millimetre-wave radar

The DreamHAT+ equips your Raspberry Pi 4 or 5 with millimetre-wave radar with 15m range. Operating at the 60GHz frequency gives it a high resolution (3cm) that can be used for human detection and tracking, as well as gesture recognition. You can see both of these in action in the videos on the DreamRF YouTube channel (**rpimag.co/dreamrfvid**). In one demo, a walking person's movements are tracked and turned into a Cartesian plot to provide a real-time 2D visualisation; in another, the HAT tracks and visualises the movements of someone's body and arms with a range-Doppler plot.

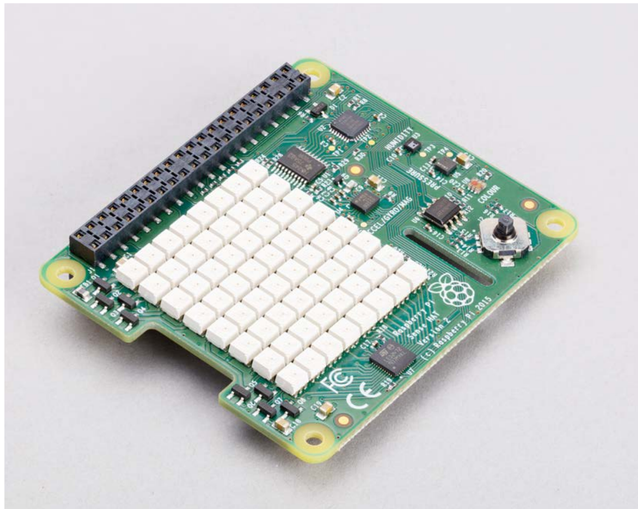
Python code for these visualisations is provided with the special OS image for the HAT. Other examples enable you to read and store data in real time and process it. While not documented, there's the potential to adapt the code to trigger or control something with body movements or gestures. It makes for an interesting alternative to using a camera and AI.

Verdict

High resolution radar that can be used for gesture detection.

Sense HAT

Raspberry Pi | £29 / $32 | raspberrypi.com



▲ Sense HAT includes an IMU that enables tilt control

As used to perform experiments on the International Space Station (as part of the Astro Pi educational programme, astro-pi.org), Raspberry Pi Sense HAT packs a raft of sensors alongside an RGB LED matrix. As well as a mini joystick, it features an IMU (inertial measurement unit) comprising a gyroscope, accelerometer, and magnetometer (compass). This means it can detect rotation, movement, and orientation. So you can use it as a 3D control device, measuring pitch, roll, and yaw as you move and tilt the board.

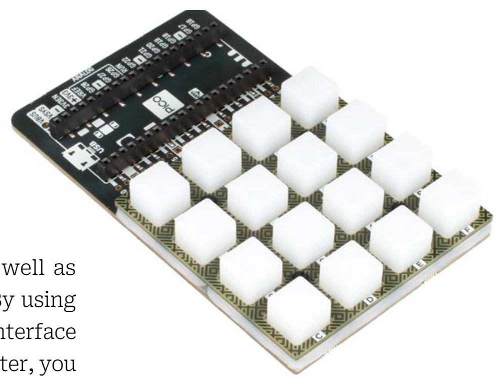
An officially supported Python library makes it easy to program projects by taking readings from the IMU in several ways, as well as the onboard pressure and humidity sensors – which also measure temperature. There's a light and colour sensor, too. Function details and a few code examples are provided in the official documentation for the Python library: **sense-hat.readthedocs.io**. Sense HAT will also work with C/C++ and Scratch.

Verdict

Add tilt/rotation control to your project with Sense HAT.

Pico RGB Keypad Base

Pimoroni | £22 / $24 | pimoroni.com



Need an auxiliary keypad for your Raspberry Pi (or other) desktop PC setup? Mount a Raspberry Pi Pico on the Keypad Base and you're all set. With an RGB LED under each silicone key, its 4×4 grid can light up in various shades. If you'd prefer less squishy keys, the similar Keybow 2040 is a good alternative.

While Raspberry Pi Pico is programmable in MicroPython or C/C++, the PMK CircuitPython library offers more scope for keypad customisation. You can attach decorator functions to keys, as well as detecting combinations of keys. By using the keypad as a USB HID (Human Interface Device) for your connected computer, you can send key presses and text strings; with USB MIDI, you can use it as a music controller, sending MIDI notes.

It's a versatile input device, suitable for all sorts of projects. Check out our Mole Bop game tutorial in issue 131 (**rpimag.co/131**), for one fun example. Another is using it as an OBS Studio controller for livestreaming: **rpimag.co/obskeypad**.

▲ This keypad is a versatile, programmable input device

Verdict

Pico-powered macro keypad with lots of possibilities.

Mini Analogue Joystick

The Pi Hut | £20 / $27 | thepihut.com

While there are lots of console/arcade-style controllers that you can plug into a USB port on Raspberry Pi, most of them feature a joystick or joypad capable of only reading a limited number of directions (typically eight) using clicking digital switches. If you want more nuanced control, an analogue stick is needed, as it's able to read movement in any direction with variable pressure. This one from The Pi Hut features two 10K potentiometers to measure movement in the x and y axes, with a stick tilt range of ±25°.

To use it, you need to connect one side of each potentiometer to ground, and the other side to a higher voltage, then read the analogue voltage from the middle connector – using an analogue input pin on Raspberry Pi Pico, for example. To use a Raspberry Pi computer, you'll need an ADC chip (e.g. MCP3008) to read the analogue signals. Another thing to note is that the stick is designed to be panel-mounted on a joystick box, which you'll need to build/supply.

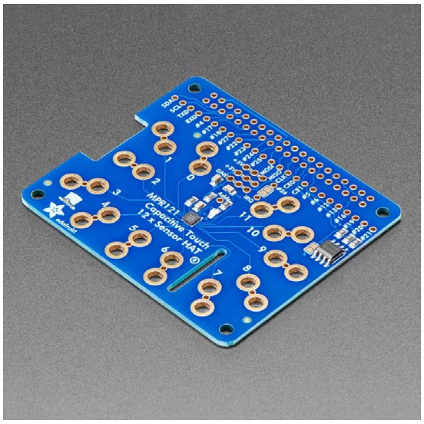
▲ This joystick is designed to be panel-mounted

Verdict

Enjoy more nuanced analogue control with this stick.

Capacitive Touch HAT

Adafruit / The Pi Hut | £14 / $19 | adafruit.com / thepihut.com



► You'll need to solder a header to the board

While simple push-button switches are often used as inputs for Raspberry Pi projects, capacitive touch sensors are a more versatile alternative. They enable you to turn any conductive item into a touch switch by wiring it up, such as cutlery, conductive fabric/thread, copper tape, and even pieces of fruit.

Featuring twelve touch sensors, the Capacitive Touch HAT is based on the MPR121 chip (also available on STEMMA QT / Qwiic breakout boards) that makes it a lot easier to do capacitive sensing than with analogue inputs.

You do need to solder the supplied 20×2 female header to the board before mounting it on Raspberry Pi, but it only uses the two I2C GPIO pins. You can then start connecting conductive items to its figure-8 holes using alligator clips (not supplied). A CircuitPython library – usable on Raspberry Pi after installing Adafruit Blinka – makes for easy programming to detect presses in Python. How about a fruit piano?

Verdict

Twelve-channel touch sensor board in handy HAT form.

AI Camera

Raspberry Pi | £63 / $75 | raspberrypi.com

A more instinctive control method for a device is to use physical gestures. With the advent of AI and machine learning, the view from a camera can be analysed to interpret hand, head, or body movements as commands. Raspberry Pi AI Camera is ideal for this kind of thing. Just connect it to the camera port of your Raspberry Pi computer – it will work with any model.

It's built around a 12-megapixel Sony IMX500 Intelligent Vision Sensor with an integral AI accelerator. That means it can run a variety of neural networking models, such as TensorFlow and PyTorch, leaving Raspberry Pi's processor free to perform other tasks. Another plus point is its seamless integration with Raspberry Pi's camera software stack, including rpicam-apps and Picamera2. Check out issue 147 (**rpimag.co/147**) for a starter tutorial.

As an alternative, you could use a standard Camera Module with a Raspberry Pi AI HAT+, which has higher theoretical performance, but that only works with Raspberry Pi 5.

▲ With an integrated AI accelerator, this is one smart camera

Verdict

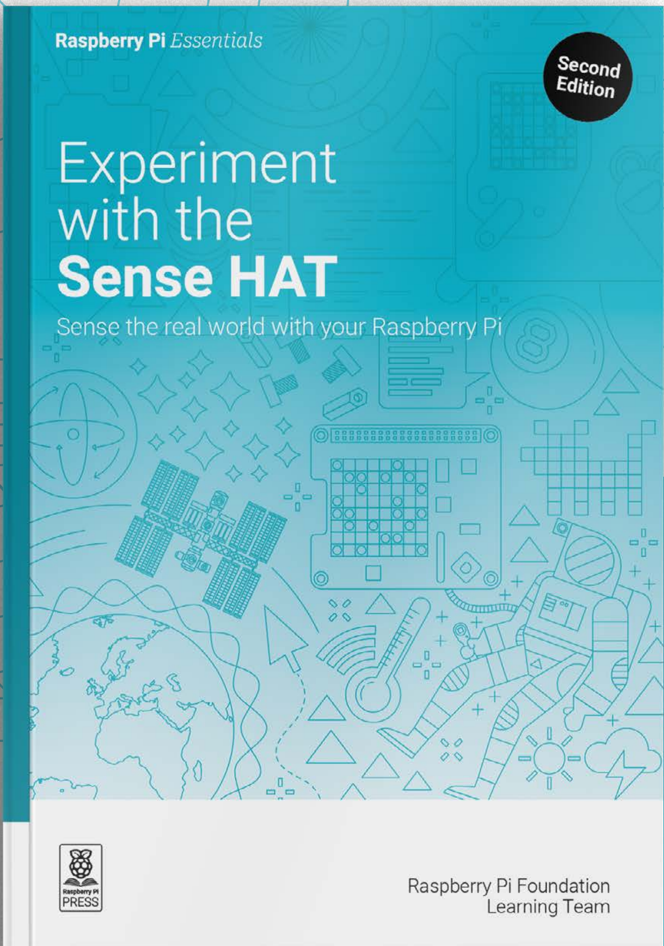
Smart camera that can be used for pose detection and gesture control.

8BITDO PRO 2

8BitDo | £45 / $45 | 8bitdo.com

Many game controllers, such as this one, can be used with Raspberry Pi. Some plug into a USB port, while others connect wirelessly via Bluetooth. As well as for gaming, they can be used in a whole range of projects, such as for controlling a robotic arm.

▲ This wireless controller works well with Raspberry Pi



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.

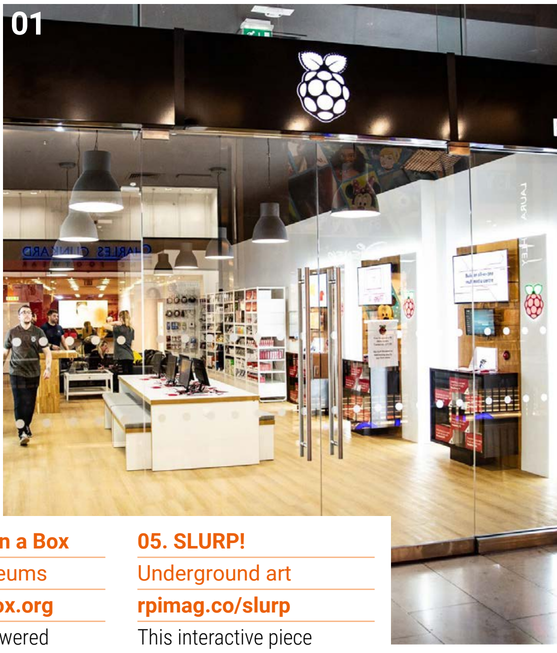
- ***Getting started with Sense HAT***
- ***Learn by building:***
  - *A digital twist on the Magic 8 Ball*
  - *Your own interactive pixel pet*
  - *A sparkly light show*
  - *An environmental data logger*
  - *Flappy Astronaut, a low-res, high-fun video game*

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

10 amazing: museum projects

Preservation and displays with a bit of help from Raspberry Pi

Raspberry Pi secretly powers a lot of things around us. From flight details to cocktail dispensers, to security tagging and space simulations. Many humble information kiosks are also powered by Raspberry Pi, something you'll see in museums, but that's not the only place you'll find a Raspberry Pi when you head to one. Here are just some ways Raspberry Pi boards have been used for museums...


01

01. Raspberry Pi Store

Hidden exhibit

rpimag.co/camstore

At the back of the official Raspberry Pi Store in Cambridge is a bench with a glass top – peering down, you can see various important items from Raspberry Pi history.

02. Centre for Computing History

Old and new

rpimag.co/comhistory

As well as display kiosks, Raspberry Pi is used in emulation of old hardware that can't – or is too risky – to run any more.

03. Digital Rail

Wide kiosk

rpimag.co/digirail

With a 1920×710 touchscreen display, this museum piece for accordions is interactive and powered by a Raspberry Pi Model 3B+.

04. Museum in a Box

Portable museums

museuminabox.org

Raspberry Pi-powered boxes of learning that use NFC tags, 3D printed artefacts, and video and sound so that folks unable to get to a museum can still experience one.

05. SLURP!

Underground art

rpimag.co/slurp

This interactive piece created for the London Canal Museum makes members of the public part of the (fictional) history of the ice wells.

06. Clay Interactive Installations

Learning through play

rpimag.co/claymuseum

The V&A Museum of Childhood transformed many exhibits with a Raspberry Pi to be interactive 'imagination playgrounds' using lights, sound, two-way mirrors, and more.

07. MP4Museum

Synchronised playback

mp4museum.org

More on the kiosk end of things, this is a specific build that will automatically play video files from connected storage, and can be synced with other players for multi-screen setups.

08. Astro Pi

Conserved Raspberry Pi

rpimag.co/astropism

This Raspberry Pi *is* the museum piece – an actual Astro Pi used on the International Space Station to run experiments written by school students.

09. Visual Theremin

See the music

rpimag.co/vistheremin

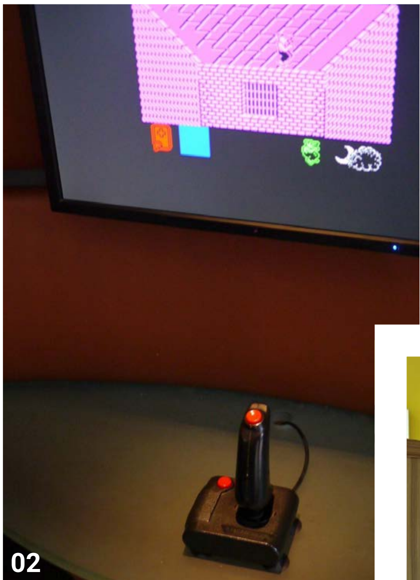
Theremins are classically used for spooky, eerie sounds and often linked to space as well. This museum exhibit teaches more about the theremin and allows you to see how the sound waves work as it plays.

10. Smart museum exhibit

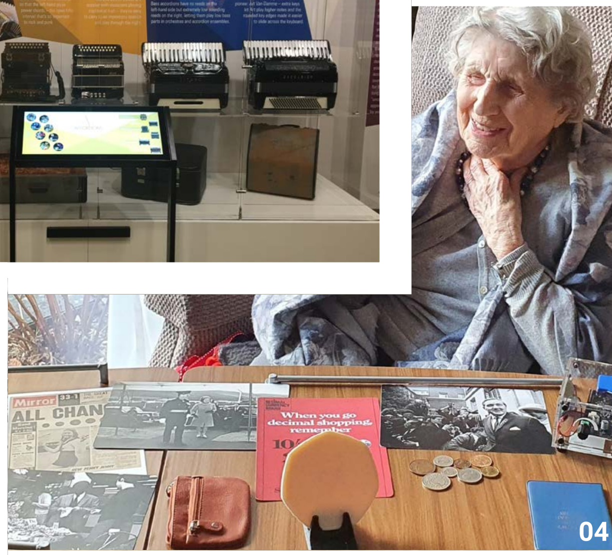
Age-appropriate info

rpimag.co/smartmuse

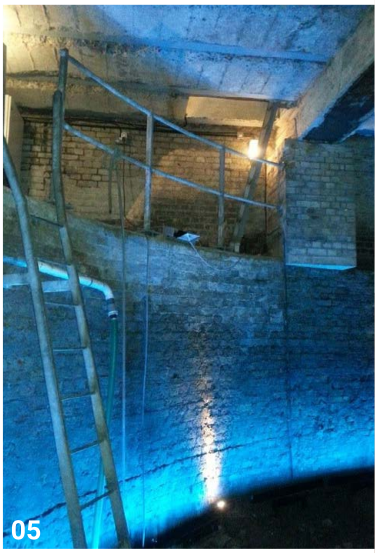
This clever Raspberry Pi 3 uses a webcam and Windows IoT Core with Azure Face Recognition to figure out roughly who is looking at the exhibit, and displays info curated for that demographic.


02

03



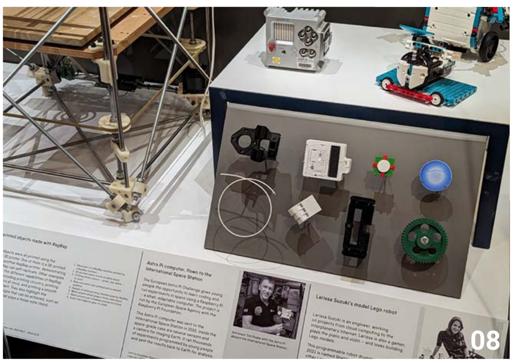
04

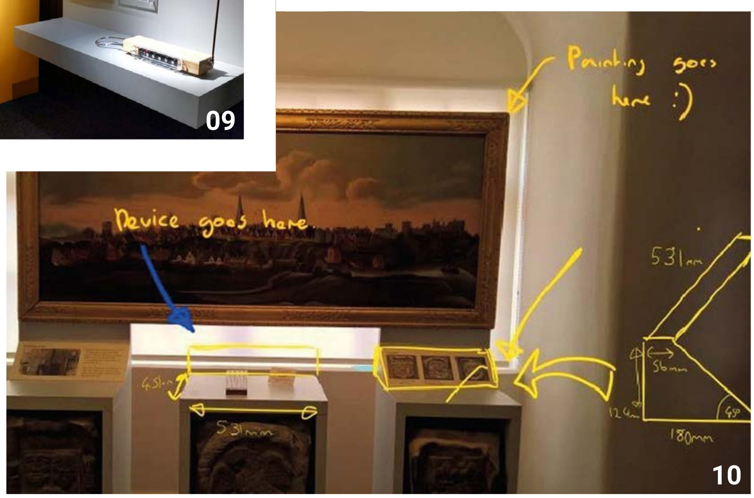

05


06

07


09


08



10

Phil Hutchinson

Working behind the scenes at element14, Phil helps bring YouTube, and hackathons, to life

- **Name** Phil Hutchinson
- **Occupation** Head of element14 Presents & Programs
- **Community role** Community organiser
- **URL** rpimag.co/element14

Among the Raspberry Pi and maker communities at large are the organised communities within them – the sum of their parts making the whole that much greater. Premier Farnell, a leading partner with Raspberry Pi, has its own community called element14, which Phil Hutchinson helps manage.

"I come from a background of always being the 'technical guy' in every role," Phil says. "I was the tech for a community radio station, the technician on a paintball site, technical manager of a chain of live venues. Lots of scrappy startups. Every role needed something fixing, and I am that guy."

When did you learn about Raspberry Pi?

Leeds Hackspace was giving coding lessons on the [original] Raspberry Pi Model B and I was very excited to see a full computer running on something the size of a credit card. My mind was blown at the possibilities.

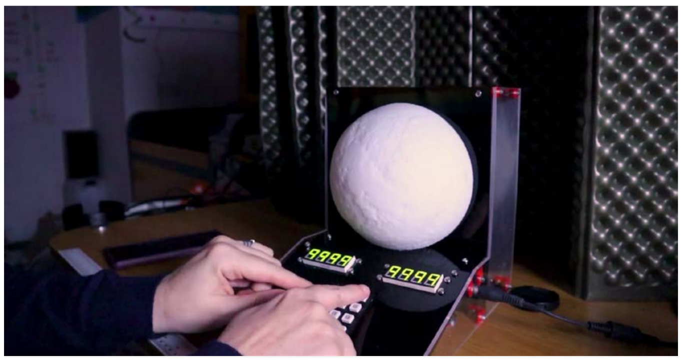
How did you start with element14?

Farnell and element14 community are based out of Leeds in the UK and I am a member of Leeds Hackspace; when they were looking for a maker with events and project experience, I was suggested. The first two questions were: "Star Wars or Star Trek, and why?" and secondly, "Have you heard of Raspberry Pi?"

What are some of your favourite community events you've held?

We have hosted hackathons across the world and many local to us out of the UK. There is something beautiful about assembling people with a wide range of skills and watching the cogs in their heads turn! We ran a Raspberry Pi-powered Hackathon where we gave the entrants $1m^3$ of space and asked them to produce enough food for a journey to Mars. It was great fun, until one contestant went away for the summer and left the hydroponic algae system running in his dorm room.

Phil hunting for satellites with a PVC pipe and some coat hangers



RPOM contributor Lorraine Underwood makes a Moon Phase Display with a Pico: **rpimag.co/moonphasedisp**

What projects have you made with Raspberry Pi?

I have been getting into a little bit of SDR [software-defined radio] to have some fun hunting satellites, but I have to say the engineers over on the element14 Presents YouTube channel have created everything from Clem's custom GPUs, Katie's timelapse camera, all the way to Milos's DIY hotplates using Raspberry Pi and Raspberry Pi Pico.

> *We have hosted hackathons across the world*

What events are you currently planning?

We will be attending Embedded World in Nuremberg, Germany, where we will be hosting our own talk stage at the Farnell booth in March 2026. Come and see myself and Mayer Makes!

Where can folks find you?

At element14 Presents and the element14 community (**rpimag.co/element14**), we are always looking for makers and engineers to get involved in making and building projects with Raspberry Pi. We also give out loads of free hardware and have a community of engineers ready to chat. Come join us!

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. Simple and effective, unlike all those Wii wheels

02. We hope no surviving French royalty knocks on John's door

03. Oh, this looks very satisfying to use

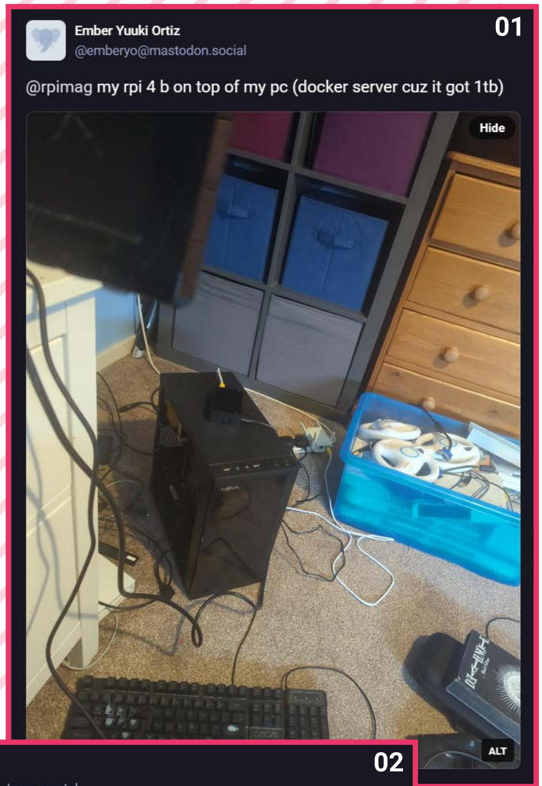
04. Ah! Who knew a Pico could be so spooky and scary?

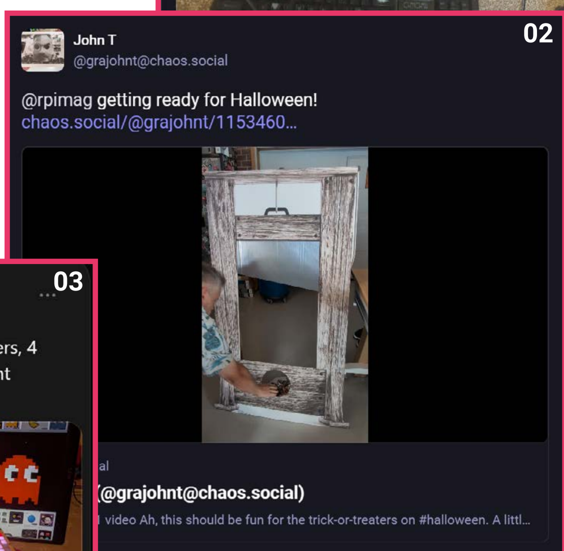
05. You should definitely not touch the plate to make sure the warning is correct

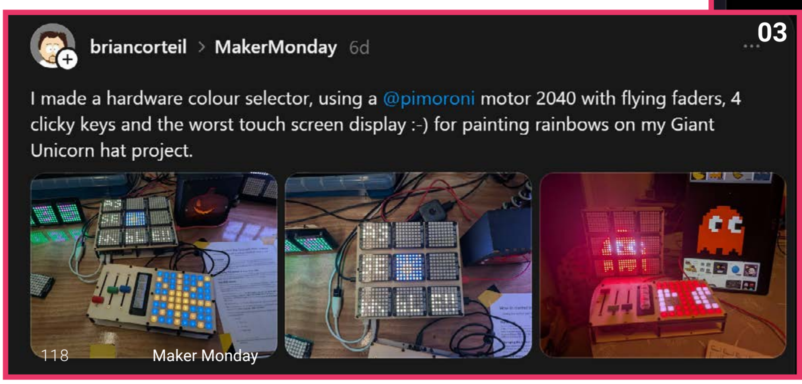
06. At the time of writing, Halloween has just finished. It's time for a Christmas tree now

07. An interesting use of AI image editors, as seen in Top Projects

08. Two great tastes together at last







04

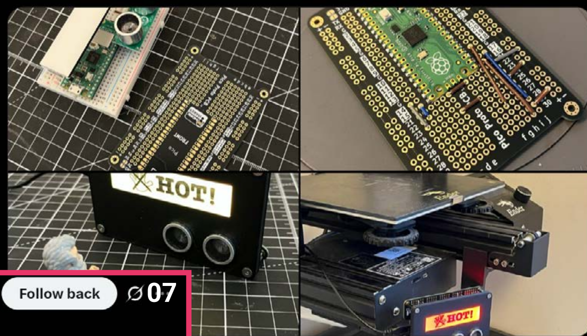
GurgleApps
@GurgleApps@mastodon.social

The spooky eyes & flame are completed for Clive & his family, our annual #Halloween #Pumpkins. They compliment our flickering flame kit. All his "brains" use #RaspberryPi Picos. Find out more, tune into our show to be released 6pm BST
youtu.be/DnkPIKpWXn4?si=dcwJNG...
#stemeducation #electronics #maker @RaspberryPi @rpimag

05

Pater Practicus
@PaterPracticus

Just finished my colour-changing proximity alert sign to warn folks of the hazards of my 3D printer. Full development from prototype to soldered PCB (using @monkmakes handy Pico Breadboard and Proto PCB) can be seen in my new video: youtu.be/uaNNTNKOxVg?si... #MakerMonday



07

Nick Bild
@NickBild79

Follow back

Banamera is an AI-powered digital camera that edits your photos using voice commands:

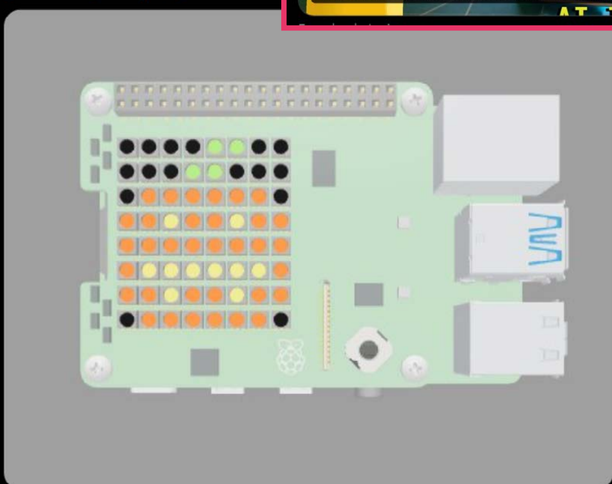
06

Igor De Souza
@Igfasouza

It the time for the year already.

Sense hat with #java @pi4j

github.com/Pi4J/pi4j-driv...



08

Roland Schulz
@r_schulz_maker

I turned the world a little upside down. Have you ever tried to program a @Raspberry_Pi Pico with an @arduino Uno? It works pretty well, if the Uno has the lastname Q. A simple led blinking code with ThonnyIDE and the world as we know it is no longer the same.#makermonday

Raspberry Pidols

Japanese TV is famous for its broad range of 'variety shows', magazine programmes covering a variety of topics with the help of celebrities, often including challenges and silly pre-recorded prank videos. Tokyo PC Club is a little different in that it focuses on idols (reductively, a sort of pop star personality; we need more space than the mag provides to go into it in full detail) from the group Nogizaka46 learning how to code their own game, while playing some smaller indie games together.

We recently caught up with Masafumi Ohta, head of the Japanese Raspberry Pi User Group (**raspi.jp**), who appeared on the show recently to give the girls a look at Raspberry Pi – showing off the AI Camera and helping them with some programming challenges. The episode aired on TV Tokyo and we don't think it's available on YouTube, so you'll just have to marvel at the screenshots for now.

The AI camera was shown off identifying people

The idols got to see Raspberry Pi 5 and Zero in action

The idols were impressed that Raspberry Pi was an entire PC in such a small formfactor

Masafumi Ohta showed the idols how to use Raspberry Pi

Crowdfund this

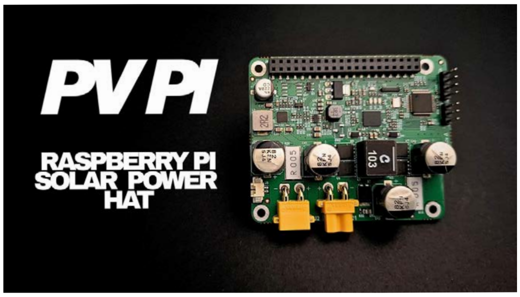
Great crowdfunding projects this month

I2C Link Relay

This IoT board allows you to connect up to eight relay boards, with optocouplers to keep stuff isolated and safe from high voltages. It even has some extended GPIO pins and a set of status LEDs.

▶ **kck.st/4oITz2X**

PV PI



This special attachment for your Raspberry Pi allows you to power it from solar batteries, specifically 12V LiFePO4 batteries, and provides "10A of true MPPT solar battery charging" – which probably makes sense to folks more knowledgeable about solar power.

▶ **kck.st/4otz5v8**

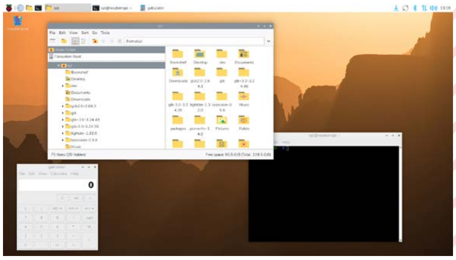
Your Letters

Trixie

I don't understand a lot of the stuff that goes on underneath the surface when it comes to computers, but I do know that newer is better. So a new operating system [the latest version of Raspberry Pi OS, which we wrote about last issue], has got to be a good thing. By coincidence, last month saw the end of support for Windows 10 – if you want to upgrade to Windows 11, you need a whopping 9GB of space just to install the bare OS.

David, via email

We're so used to using Raspberry Pi as a platform for other things that we can sometimes forget that it's a great computing platform in its own right. The fact that you can get new versions of the operating system for free is both baffling and wonderful for someone who has grown accustomed to paying money for new versions of Windows. And you could argue that the bigger gaps there are between upgrades, the bigger the upheaval when it does become time to change, and the reluctance there is to do it, resulting in out-of-date and unsupported software being used in all sorts of important systems. With Raspberry Pi OS being free, and constantly updated, you've no reason not to try the latest version.



▲ The new Trixie version of Raspberry Pi OS

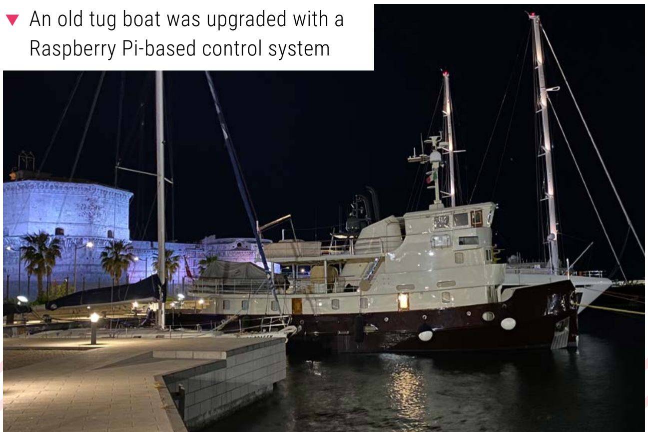
Sfera Baltic boat

I've been thinking about using a Raspberry Pi to turn a remote-controlled boat into something more autonomous. Partly because I have the boat already, and it would be cheaper than building something like a drone from scratch, and partly because I'm close to a small lake that's shallow enough to wade into and pick up a boat if it runs out of battery power. I've got a plan, and I just need to wait for a weekend to myself so I can start to put it all together… and then I saw your write-up of an actual ocean-going boat that uses Raspberry Pi as a navigation, power management, and who knows whatever else computer. It makes my months of procrastination look ridiculous – if you can control a full-sized ship from a computer, you can definitely control a little toy boat.

Sue, via email

Go for it! There are millions of brilliant ideas that never get made – don't let yours go the same way! Our best advice would be to forget about controlling a full-size ship (for now) and focus on one feature at a time. Maybe add a speedometer, or add an IMU module so that you can see which direction your boat is facing – that way, with a little maths, you should be able to work out where it is. After that, the world's your oyster – just make sure to take a spare battery with you if you're out on the lake!

▼ An old tug boat was upgraded with a Raspberry Pi-based control system



Retro gaming

Christmas is coming, and I have a couple of nephews who are labouring under the delusion that they want a Nintendo Switch. That's their mum's problem, not mine: I intend to make sure that they grow up knowing games from the Good Old Days, which in my case is roughly 1990–95. It was cutting edge at the time, but easily emulatable by Raspberry Pi nowadays, so I get to be the cool uncle and download some games onto a spare Raspberry Pi 4 to blow their tiny minds. And as luck would have it, I see a guide to retro gaming hardware in issue 159 – Santa might even bring an extra little something, just for me.

Brian, via email

Christmas really is all about gathering round the obsolete gaming setup and humiliating children at games that were already old before they were born. We'll put in a good word for you with Father Christmas the next time we see him – hopefully you'll get something nice like the 8BitDo Arcade Stick, which we love for the way it looks so much like unreleased Nintendo hardware from the late eighties. And if anyone else is stuck for gift ideas, there's always our lead feature this issue on page 32!

▼ A superior controller for your retro gaming

Contact us!

@rpimagazine

@rpimag

rpimag.co/facebook

magazine@raspberrypi.com

forums.raspberrypi.com

USA SPECIAL!

6 ISSUES FOR $43

Subscribe online:

rpimag.co/subscribe

Continuous credit card orders will auto-renew at the same price unless cancelled.
A choice of free Pico 2 W or Pico 2 is included with all subscriptions. This is a limited offer.
Not included with renewals. Offer subject to change or withdrawal at any time.

Community Events Calendar

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

01. rAge Expo

- **Friday 5 December to Sunday 7 December**
- **Fourways Mall Rooftop, Sandton, South Africa**
- **rpimag.co/re160**

Make sure you stop by the PiShop display at the NAG stand while you're exploring rAge 2025! They're offering the latest and greatest in Raspberry Pi and all things maker tech, perfect for anyone looking to level up their coding, electronics, or gaming-adjacent projects. Come chat with their representative, discover exclusive show specials, and get any tips and tricks that you need to build your next awesome creation, all located within the vibrant heart of the NAG stand.

02. The Hobby Show

- **Friday 5 December to Sunday 7 December**
- **Johannesburg Expo Centre, Johannesburg, South Africa**
- **rpimag.co/ths160**

PiShop.co.za, a Raspberry Pi Approved Reseller, will be exhibiting at The Hobby Show. Visitors can look forward to exploring a wide range of Raspberry Pi products and accessories, seeing live demos, and getting expert advice on DIY electronics, coding projects, and more.

03. Cornwall Tech Jam

- **Saturday 13 December**
- **Fibrehub, Pool, UK**
- **rpimag.co/ctj160**

The monthly Cornwall Tech Jam brings together passionate volunteers from the local industry who are ready to guide and inspire the next generation of tech creators. There will be loads of Raspberry Pi technology to learn to code with.

04. Fruitful Raspberry Jam

- **Saturday 20 December**
- **Fruitful Hall Secondary School, Ota, Nigeria**
- **rpimag.co/frj160**

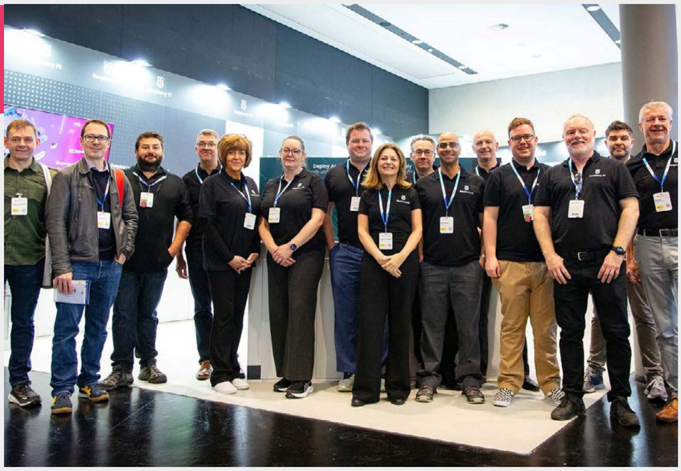
Join the organisers at Fruitful Hall for the Fruitful Raspberry Pi Jam in Ogun State — a hands-on event featuring coding activities using Scratch and Python. Explore Raspberry Pi 500, Raspberry Pi 5, and Raspberry Pi Pico, versatile devices that promote practical learning in coding, robotics, and AI. Plus, get a first look at the new 5-inch variant of Raspberry Pi Touch Display 2.

FULL CALENDAR
Get a full list of upcoming community events here:
rpimag.co/events

05. Embedded World 2026

- **Tuesday 10 March 2026 to Thursday 12 March 2026**
- **Messezentrum 1, Nürnberg, Germany**
- **rpimag.co/ew26**

The Raspberry Pi team is looking forward to returning to Embedded World in 2026. There, you'll be able to meet them and experience demos from across the full spectrum of Raspberry Pi products, including Raspberry Pi Pico 2, the AI product range, RP2350-based solutions, and Raspberry Pi's latest industrial device: Compute Module 5.



Official Raspberry Pi Event

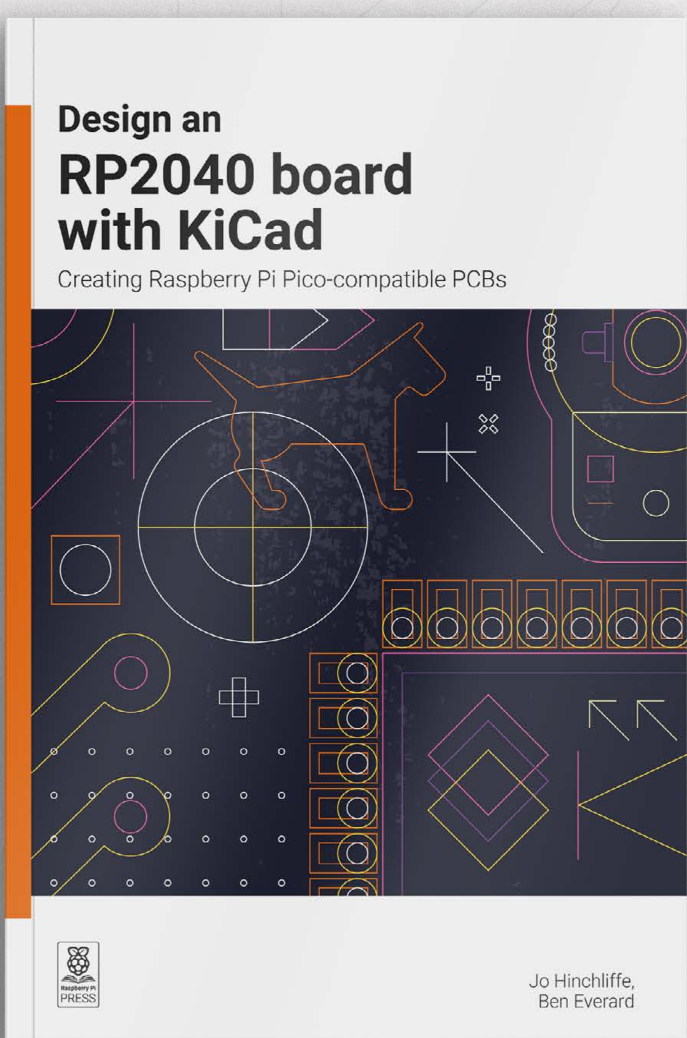
Win
A Raspberry Pi 500+ & Raspberry Pi Monitor

Raspberry Pi 500+ needs no introduction. This full-power Raspberry Pi comes with 16GB RAM, a 256GB NVMe SSD drive, and a wonderful clickable mechanical keyboard. The Raspberry Pi Monitor is a 15.6-inch delight to look at with an IPS display and two front-facing speakers. The whole setup is the best computer around.

Head here to enter:
rpimag.co/win
Learn more:
rpimag.co/products

Terms & Conditions

Competition opens on **19 November 2025** and closes on **18 December 2025**. 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.



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

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 #161

Brilliant First Builds!

Inspirational makes with Raspberry Pi

On sale 18 Dec

PLUS!

- First steps for newcomers
- **Build your own AI dataset**
- Best environmental sensors

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, Ben Everard, Rosemary Hattersley, Jo Hinchliffe, Phil King, Andrew Lewis, Simon Long, KG Orphanides, Sean M Tracey

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

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001. Raspberry Pi Official Magazine is published by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products, or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).

ISSN: 2977-4403 (Print)
ISSN: 2977-4411 (Digital)

Hope for the future

The future isn't something we inherit; it's something we build.

By **Lucy Hattersley**

The team was chatting the other day about terrible computers in days of yore. Of which there were many. Some of which remain decidedly rose-tinted-glasses-proof despite the passage of time.

I remember the Dragon 32, which had a bug in the CPU that early models shipped with. This meant that you had to slowly load a tape with a software fix every time you started up the machine. Then there was the Philips G7000 which had a keyboard so bad that Spectrum fans laughed at it.

Rob recently one-upped all of this with a tale of the ill-fated Coleco ADAM (way before his time). The ADAM was an expansion system for the ColecoVision video games console. It featured a hardware flaw that caused a strong electromagnetic pulse (EMP) when you switched it on. This pulse had enough intensity to wipe any magnetic tapes left inside, or nearby the computer. Which meant that it could wipe out your tape collection every time you turned it on.

> *I still believe it's the best idea in the world to learn to code*

My point is that things get better. I'm reminded of this as we move into a new generation of computing. One that is far removed from where we started out.

Looking to the future

The stack is fundamentally much higher now than it was when computers were young. Even if you aren't using a sealed piece of glue and glass like a tablet, it's hard to get anywhere near the metal in any meaningful sense. I always want to know not just that something works, but how it works.

There's a thought that AI may not only do the coding for us but get rid of coding as a concept entirely. After all, why ask an AI to write a program for drawing art, or making music, or understanding maths, when you can just ask it directly for the end result?

Our Raspberry Pi colleague Jonathan Bell recently shared this Nokode project (**github.com/samrolken/nokode**). It's an early exploration of a web server with no application logic.

Then there is the impressive NeuralOS (**neuralos.lambda.ai**). This project uses an LLM to recreate an operating system (any operating system or program) that you can describe. Ask it to recreate Debian Linux, or a Matrix-inspired OS, or a religious fever dream OS and it will do its best. NeuralOS doesn't code an operating system. It just responds to keyboard and mouse input and provides visual output. Results are patchy, but it's potentially an interesting view of the future.

Love and grace

Maybe our future is like an Iain M. Banks Culture series. We live unimaginably comfortable lives; all watched over by machines of love and grace.

I suspect that the people who are getting to grips with code today are the ones who will own tomorrow. Behind every beatific influencer is a more powerful nerd who built the algorithm that made them famous.

I still believe it's the best idea in the world to learn to code. It teaches you how to think. This is what will get you on in life. And, if not, at least we will be able to understand how our robot overlords tick (and tock).

Lucy Hattersley – Author

Lucy has been reading *Computing with Quantum Cats* this month. It's a great book if a little mind-bending as it dives into the Many-Worlds Interpretation and computing in the multiverse. She's not going to pretend to understand it (yet).

rpimag.co

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

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

Available at these great Pi stores:

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

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

PiKVM

Remote control redefined

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

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

PiKVM V4 Plus

The most feature-rich edition

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

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

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI
Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBii · Cytron Marketplace
ROBU.IN Your ideas, Our Parts · NETTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

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

List of official resellers by country: