Recreating DOOM **with Super FX 3 Pico chip**

Build a **Pico RC submarine & marine projects**

Make a **facial recognition camera system**

Raspberry Pi

Official Magazine
#157 | September 2025

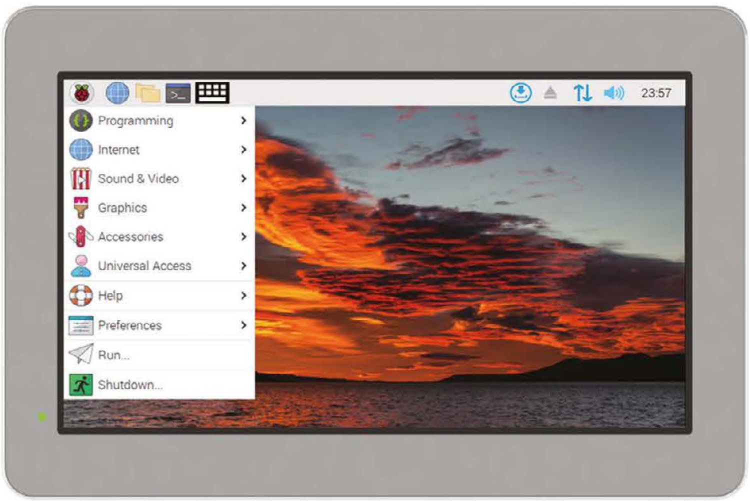
STARTER **ELECTRONICS**

Learn how to **build and program circuits**

£7.99
ISSN 2977-4403 (Print)
09
9 772977 440004

Industrial Raspberry Pi

ComfilePi



LISTED

Powered by Raspberry Pi

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

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

Visit www.comfiletech.com

COMFILE TECHNOLOGY

Welcome to Raspberry Pi Official Magazine

Editor
Lucy Hattersley
This month Lucy grinned in front of a facial recognition project. But drew the line at showing her paws to the Rock, Paper, Scissors build.

rpimag.co

One amazing thing about Raspberry Pi products is how close you get to the metal.

Metal – in this context – isn't music, materials, or military hardware, but the raw, physical hardware that makes a computer tick. Below your programs, underneath any operating system, below the firmware, sits the metal: this is where the magic happens.

Raspberry Pi Pico lets you reach into that world. It's a mighty microcontroller board packed with GPIO (general-purpose input/output) pins. These enable you to hook up electronic components and start experimenting with digital making. Suddenly you're not just coding – you're making!

Attach some buttons, lights, sensors, motors, and you can start to imagine all kinds of physical builds. Pretty much anything used in modern electronics can become part of your next project.

This month, Ben Everard kicks things off with a brilliant beginner-friendly project: building a real-world game with Pico. It's a joyful reminder of why we love Raspberry Pi in the first place. It's not about the software; it's about rolling up your sleeves and creating with hardware.

Lucy Hattersley – Editor

Contents

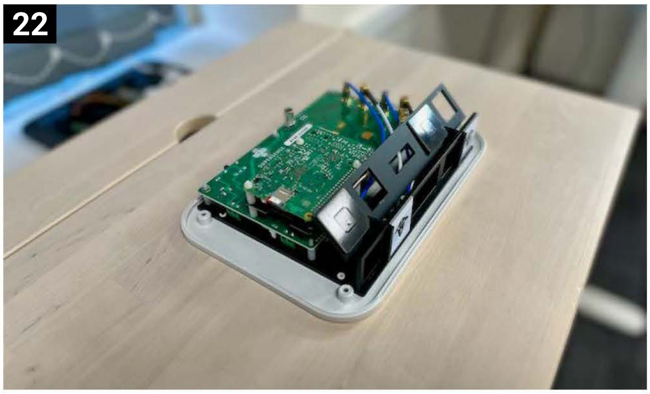
World of Raspberry Pi

Project Showcase

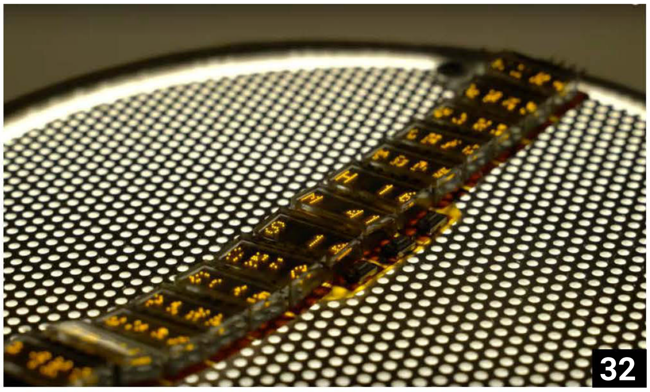
16

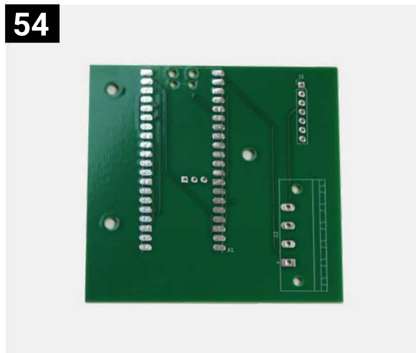
26

22



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





Top Projects

Feature

Tutorials

Feature

Win 1 of 3 Raspberry Pi 500 kits

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

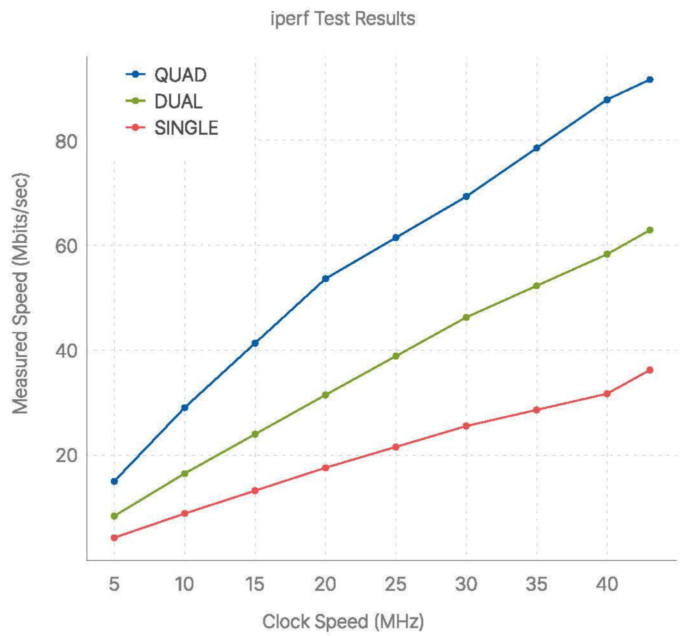
- **High-speed QSPI Interface**
  SINGLE, DUAL, QUAD modes supported
- **8 Independent Sockets**
  Multi-device connectivity made easy
- **Low Power & Wake on LAN**
  Efficient and remote-friendly
- **Seamless IoT & Industrial Integration**
  POS, cameras, automation & more

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



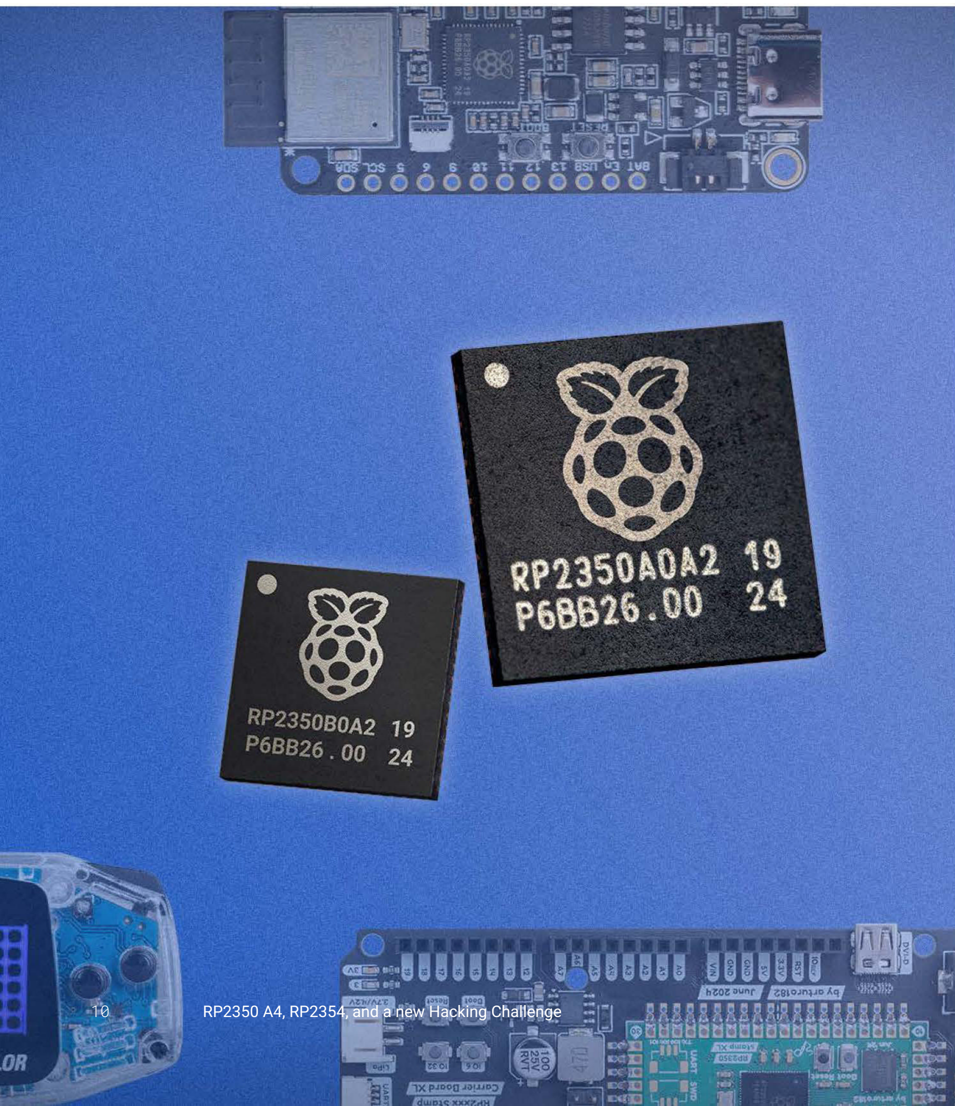
WIZ630io

W6300-EVB-Pico
(W6300 + RP2040)

W6300-EVB-Pico2
(W6300 + RP2350)

RP2350 A4, RP2354, and a new Hacking Challenge

Raspberry Pi introduces two new variants of RP2350 microcontroller. By **Lucy Hattersley**



◀ RP2350A and RP2350B microcontroller chips

Raspberry Pi has introduced new variants of its RP2350 microcontroller chip under the A4 stepping (rpimag.co/rp2350). These replace the launch model, which was designated A2, and feature fixes for issues found during Raspberry Pi's Hacking Challenge.

Alongside RP2350, Raspberry Pi is launching two RP2354 models. These are pin-compatible variants of the 60-pin RP2350A (with 30 GPIO pins) and 80-pin RP2350B (with 48 GPIO pins) and add 16Mbit/2MB of flash memory in-package.

RP2354 contains a Winbond flash die in-package, providing an elegant solution for developers who require a modest amount of storage. Each RP2354 part costs just 20 cents more than the equivalent RP2350 part.

▶ The A4 stepping fixes security issues found during Raspberry Pi's Hacking Challenge

RP2350 has found its way into countless third-party products

Fixes from the Hacking Challenge

The new variants also address issues found during Raspberry Pi's Hacking Challenge (see Raspberry Pi Official Magazine issue 151, **rpimag.co/151**).

"Like any silicon device, RP2350 wasn't perfect on day one," notes Eben Upton, Raspberry Pi co-founder and CEO. "The launch stepping, designated A2, is affected by a number of errata, including an error in the GPIO pad design which prevents pads from properly going into a high-impedance state [known as Erratum 9 in the RP2350 datasheet, **rpimag.co/e9**], and a number of security issues identified by participants in our RP2350 Hacking Challenge."

A4 also addresses boot ROM security vulnerabilities outlined in Errata 20, 21, and 24; a security vulnerability related to one-time programmable (OTP) outlined in Erratum 16; and a vulnerability of the GPIO_NSMASK registers on 60-pin RP235x parts outlined in Erratum 3.

"We're happy to announce the immediate availability of a new A4 stepping, which addresses the vast majority of these issues," says Upton.

One vulnerability has not been fixed, yet. IOActive discovered a hack called "Extracting antifuse secrets from RP2350 by FIB/PVC". In this attack, IOActive (almost) demonstrated that data bits stored in the RP2350 antifuse memory array could be extracted using a well-known semiconductor failure analysis technique: passive voltage contrast (PVC) with a focused ion beam (FIB).

"In principle it may be possible to extend this attack to retrieve the complete contents of the OTP [one-time programmable]," notes Upton. "An upcoming application note will describe how to store secrets in OTP so as to mitigate both the current vulnerability and a hypothetical future attack which can achieve complete readback."

On the success of RP2350 and the Pico 2 microcontroller boards that use it, Upton adds: "Since August last year, we've sold over half a million Pico 2 and Pico 2 W boards. RP2350 itself has found its way into countless third-party products, from secure displays to development boards to synthesizers."

A new Hacking Challenge

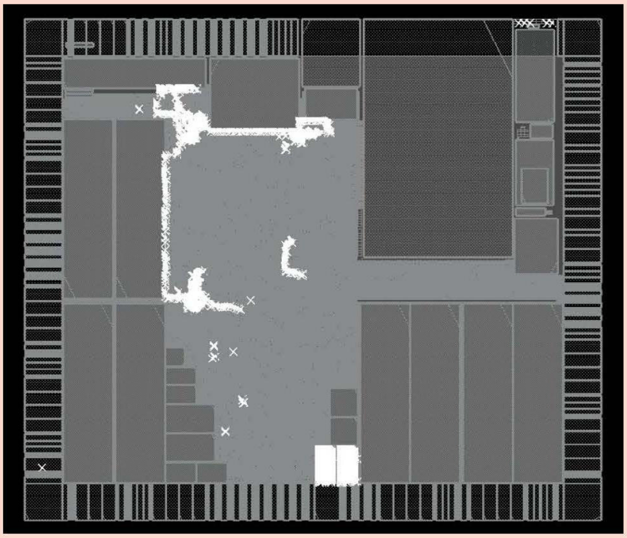
To coincide with the availability of the A4 stepping and RP2354 variants, Raspberry Pi has launched another RP2350 Hacking Challenge.

This time, it is offering a $20,000 prize for a practical side-channel attack on the power-hardened AES library that underpins the RP2350 decrypting bootloader.

rpimag.co/hackingchallenge2

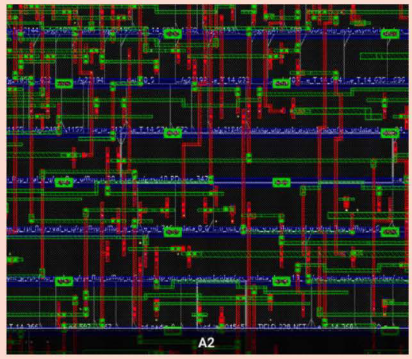
Anatomy of a metal spin

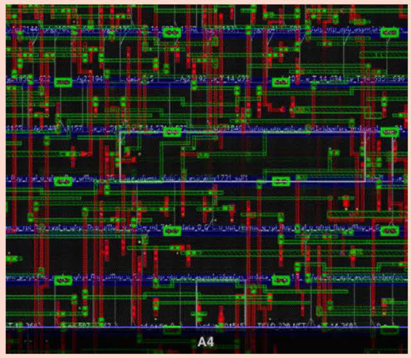
The A4 stepping is what Upton calls "a metal spin". In this upgrade, "the functional changes, including the updated boot ROM, are implemented by modifying some of the layers of wiring that connect the gates in the design."



Zooming in, here is the implementation of an individual fix (in this case, part of the OTP security hardening required to address Erratum 16). A2 (above) and A4 (below)

This image shows all the locations in which A4 differs from A2; the dense block of changes at the bottom of the die implements the updated boot ROM





Price structure

The complete price structure for Raspberry Pi microcontroller products is summarised in the table below. 7-inch reels of each product contain 500 units; 13-inch reels contain either 3400 units (RP2040, RP235xA) or 2500 units (RP235xB).

	Single unit	7-inch reel	13-inch reel
RP2040	$1.00	$0.80	$0.70
RP2350A	$1.10	$0.90	$0.80
RP2350B	$1.20	$1.00	$0.90
RP2354A	$1.30	$1.10	$1.00
RP2354B	$1.40	$1.20	$1.10

Frequently asked questions

How do I know whether I have an A2 or an A4?
The stepping identifier can be found on the top of the package, as shown below.

Is A4 software-compatible with A2?
Yes. We have made a few minor changes to version 2.2.0 of the Pico SDK (**rpimag.co/picosdk**) and to Picotool (**rpimag.co/picotool**) to support the A4 stepping.

Will you EOL the A2 stepping?
We have already ceased production of A2, moved all production to A4, and withdrawn the remaining inventory of A2 from channel. A4 is a drop-in replacement for A2, and customers should not experience any issues migrating to A4.

What happened to A3?
A3 was an intermediate stepping used to qualify a subset of the fixes present in the A4 stepping.

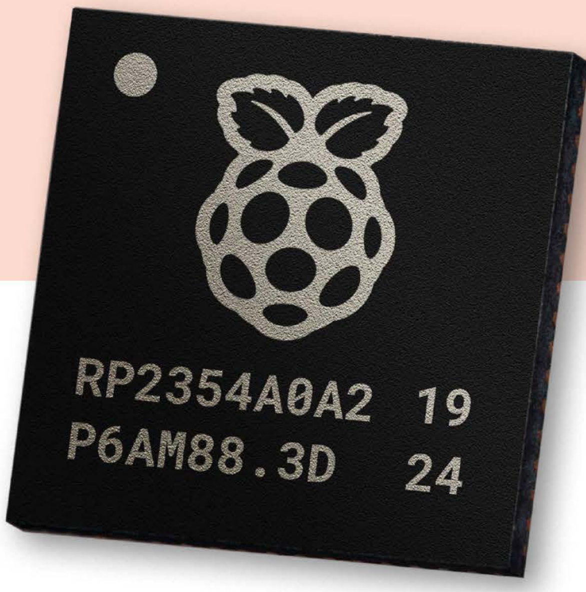
Will there be a PCN in PIP?
You can find the product change notice for the A2 to A4 transition here: **rpimag.co/a4pcn**.

Will Pico 2 products switch to A4?
Eventually, all Pico 2 products will use the A4 stepping. While the A3 stepping will not be made available to silicon customers, approximately 30,000 units of A3 inventory will be used to build Pico 2 and Pico 2 W products.

Did you change the pinout or package design with A4?
No: the A4 stepping is only an update to the metal layers of the RP2350 die.

Do I still need to use the Abracon polarised inductor and Abracon crystal with A4?
Yes: please keep following our hardware design guide and reference BOM/layout.

Will there be an A5?
We have no current plans for an A5 stepping. Over time, it is possible that more security vulnerabilities or functional issues will be found, at which point we will react appropriately.



▶ RP2354 adds 16Mbit/2MB flash memory to the chipset. A small amount of storage for persistent data without developers needing to add a separate storage solution

Connect for Organisations

Raspberry Pi Connect for Organisations features enhanced audit logging. By **Lucy Hattersley**

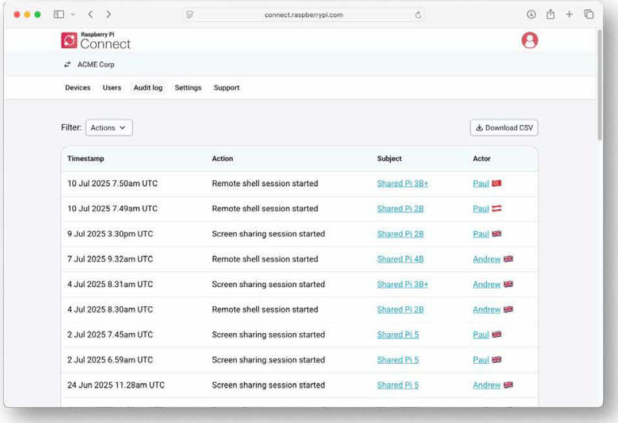
Raspberry Pi has announced an upgrade to Raspberry Pi Connect – its software designed to enable you to remotely access a Raspberry Pi with screen sharing and remote shell via a web browser.

Connect for Organisations features an improved audit log designed to improve operational oversight. Administrators can review activity from the past 90 days. "With the most recent events first, you can see every screen sharing and remote shell session, the creation of access tokens, the addition and removal of devices, and any changes to users and their roles," says Paul Mucur, senior principal software engineer at Raspberry Pi.

What's new?

New features also include:

- Country tracking: see where events took place and spot suspicious activity
- Download CSV: download detailed logs for auditing and security reviews
- Filtering: focus on the specific actions important to you

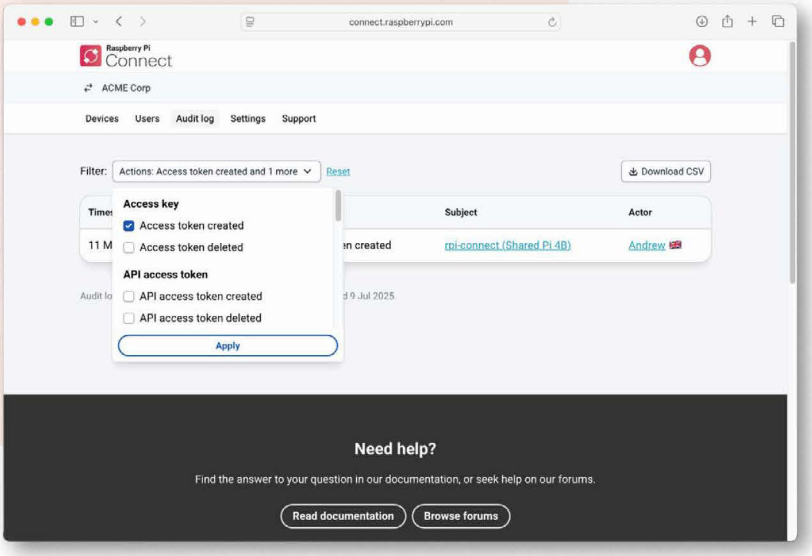


▶ The audit log lists remote shell and screen sharing sessions with Timestamp, Action, Subject, and Actor

"We now geolocate every event by IP address, storing only the country code in order to preserve privacy," explains Mucur. "This information is included in both the web interface and the newly added CSV export option to help you easily spot access from an unexpected country."

Connect for Organisations offers a Download CSV function that enables administrators to export a complete audit trail for offline review in spreadsheet software. It also offers the ability to filter the audit log by action, such as 'Screen sharing session started" or 'Invite accepted'.

"Together, these new features can aid administrators in a security investigation if a team member reports unusual activity," says Mucur.



▲ Administrators can filter by specific events, such as access token creation, to focus on relevant security activity

These new features can aid administrators in a security investigation

These improved audit log features are available immediately for existing Connect for Organisations users. Connect for Organisations is available with a four-week free trial and then from $0.50 per device per month. Discover more at Connect for Organisations: **rpimag.co/connectfororgs.**

Build HAT goes open source

Raspberry Pi opens Build HAT firmware to developers under BSD licence. By **Lucy Hattersley**

Raspberry Pi has made the firmware for its superb Build HAT (rpimag.co/buildhat) add-on board fully open source. Build HAT is Lego Education-compatible hardware that enables Raspberry Pi to control Technic motors and sensors (like those found in the Lego Education SPIKE Prime sets).

"We're delighted to announce that the Build HAT firmware, together with its signing keys, is now open source and available under the permissive BSD 3-Clause licence," says Gordon Hollingworth, CTO (Software) at Raspberry Pi.

According to Hollingworth, the decision to release the firmware source reflects a broader commitment to enabling users to "understand and take control of the technology they use where we can."

"When we launched the Build HAT, the firmware that runs on its on-board microcontroller was released in binary form only," Hollingworth wrote in the announcement. "While it was fully functional and documented via the Raspberry Pi Foundation's Python Build HAT library (**rpimag.co/pythonbuildhat**), the firmware itself wasn't something you could inspect, modify, or rebuild."

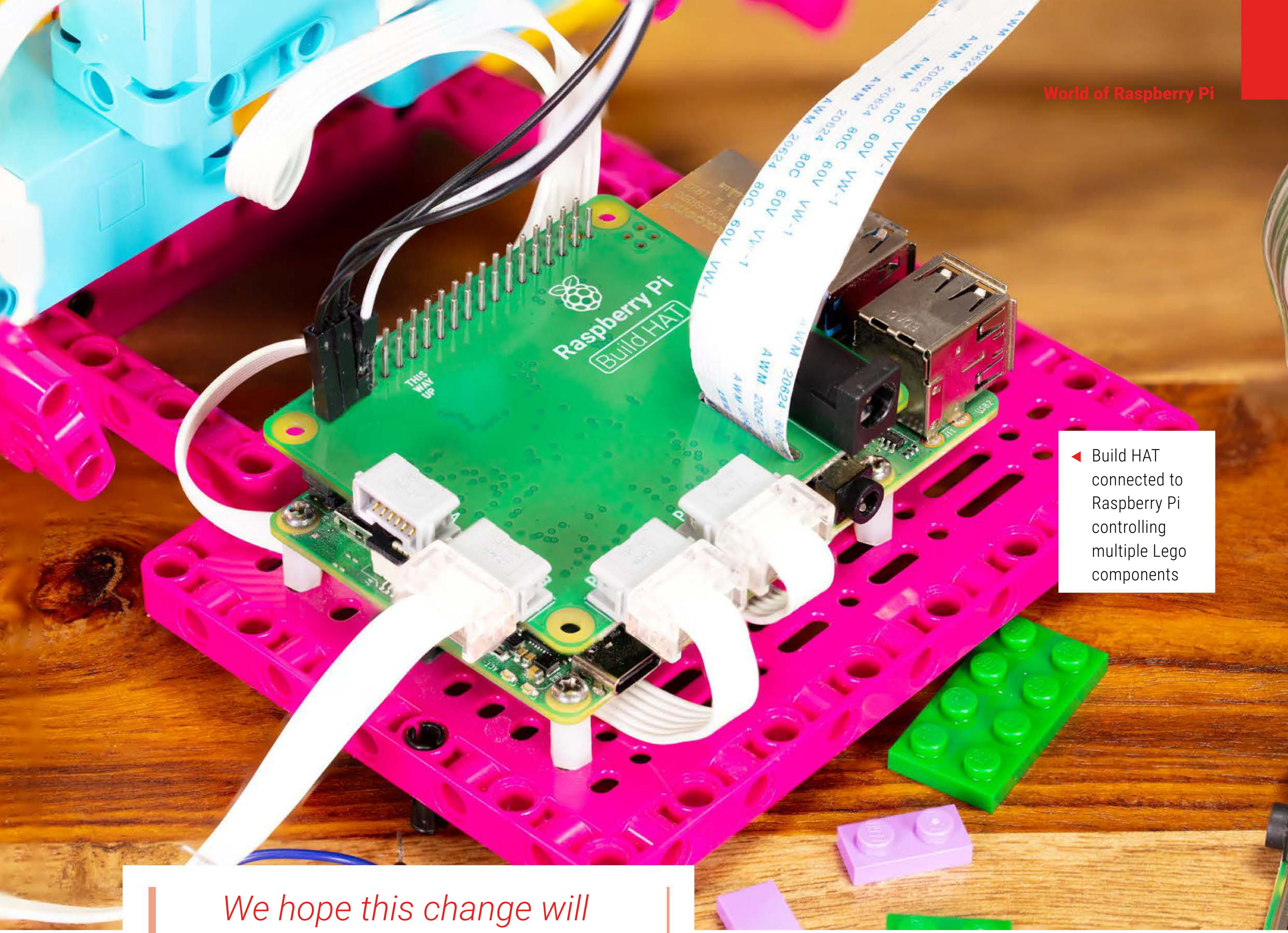
▼ Build HAT enables direct control of Lego Technic motors and sensors

Check out the repo

The official Build HAT GitHub repository (**rpimag.co/buildhatgit**) contains the complete source code, build instructions, and protocol documentation for host communication, so you can start exploring and modifying it for your own projects.

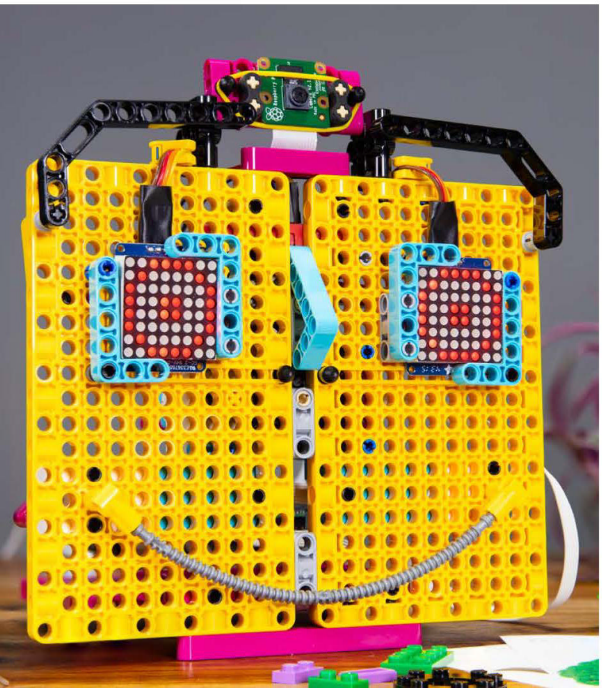
The Raspberry Pi Foundation's Python Build HAT library has also been updated with new documentation to include guidance on using the board with home-built firmware, paving the way for deeper experimentation and integration.

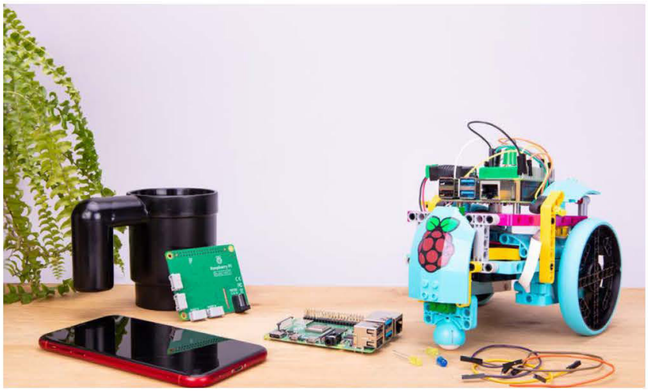
"We hope this change will inspire even more amazing projects and brilliant builds," Hollingworth noted, extending thanks to contributors Chris Richardson, Patrick Cherry, and Lego Education for their support in making the open-source release possible.



Build HAT connected to Raspberry Pi controlling multiple Lego components

We hope this change will inspire even more amazing projects and brilliant builds





This Lego face features a controllable mouth and eyebrows

A Lego Technic robotic car driven by Raspberry Pi and Build HAT

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

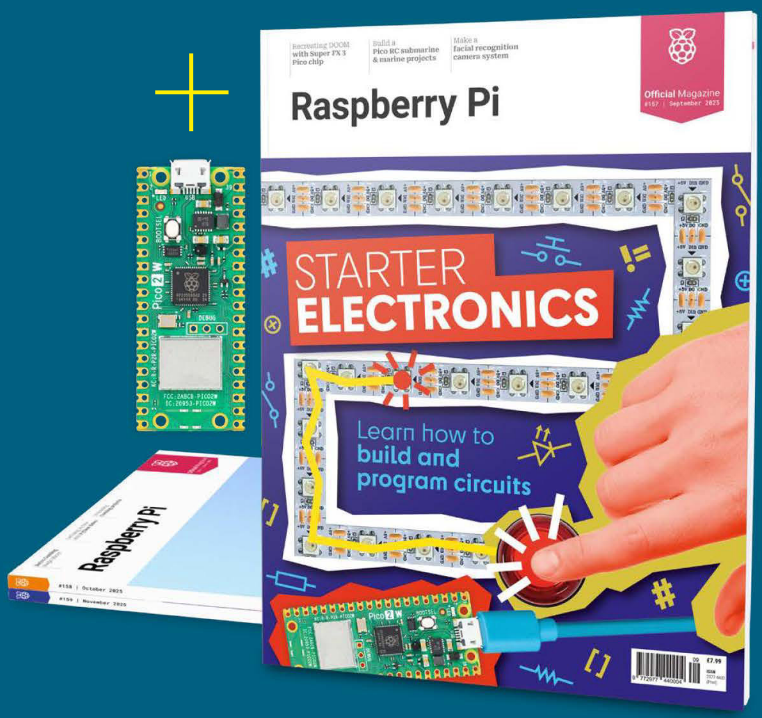
£10 (UK only)

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**

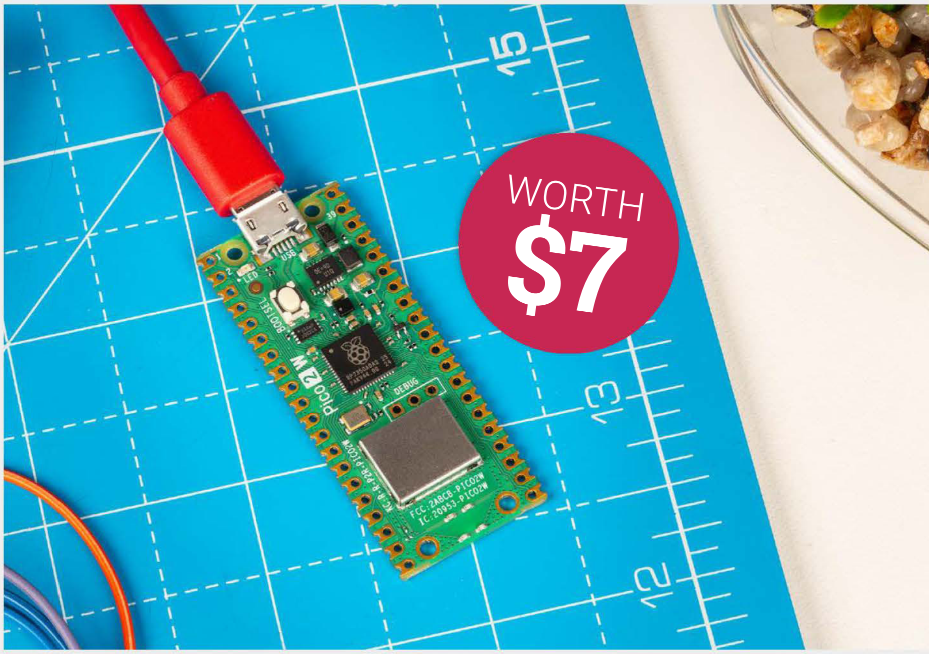
Subscribe for £10 is a UK-only offer. The subscription will renew at £15 every three months unless cancelled. A choice of free Raspberry Pi Pico 2 W or Pico 2 is included with all subscriptions.

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 RP2040-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

Edgeberry

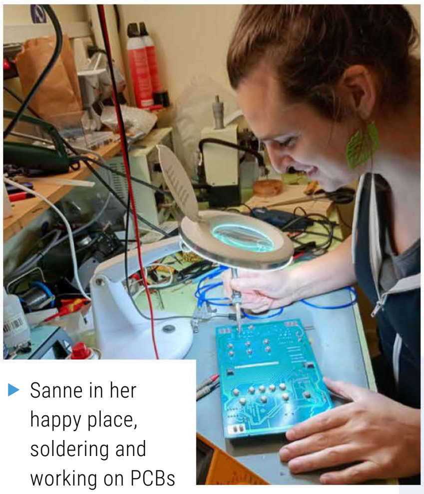
Turning home projects into a development IoT platform. **Rob Zwetsloot** connects with the story

Maker

Sanne Santens

An embedded systems and Linux creator who has had a lifelong fascination with intelligent machines.

rpimag.co/edgeberry



▶ Sanne in her happy place, soldering and working on PCBs

A lot of Raspberry Pi projects come to us after someone decided they couldn't get the right parts for their own personal project as... well, they didn't exist. Sanne Santens' impetus for her project was not much different.

"Edgeberry is the result of many years of throwing Raspberry Pi [computers] all over the place, literally, with all kinds of sensors, actuators, etc. glued to them, with custom-designed enclosures for protection or integration in its environment," Sanne says. "I found myself basically redoing substantial parts of the system over and over – but just with small [variations]: making robust power solutions, mounting Raspberry Pi [boards] in boxes, wiring up interface boards, debugging flaky connections, writing device management features in an online dashboard."

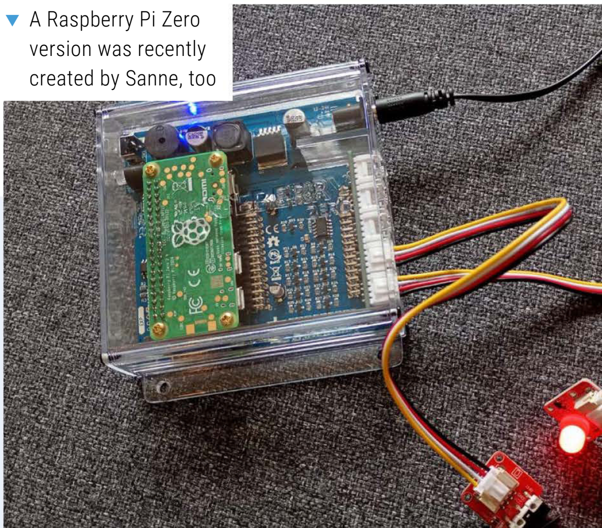
After noticing the commonalities of the devices she had been building, she decided to create her own custom expansion board that combined them all together. "Edgeberry, to me, is the solid building plate / foundation on which I build my 'IoT' ideas," Sanne explains.

IoT development

This foundation became something Sanne believes can be helpful to many kinds of Raspberry Pi users.

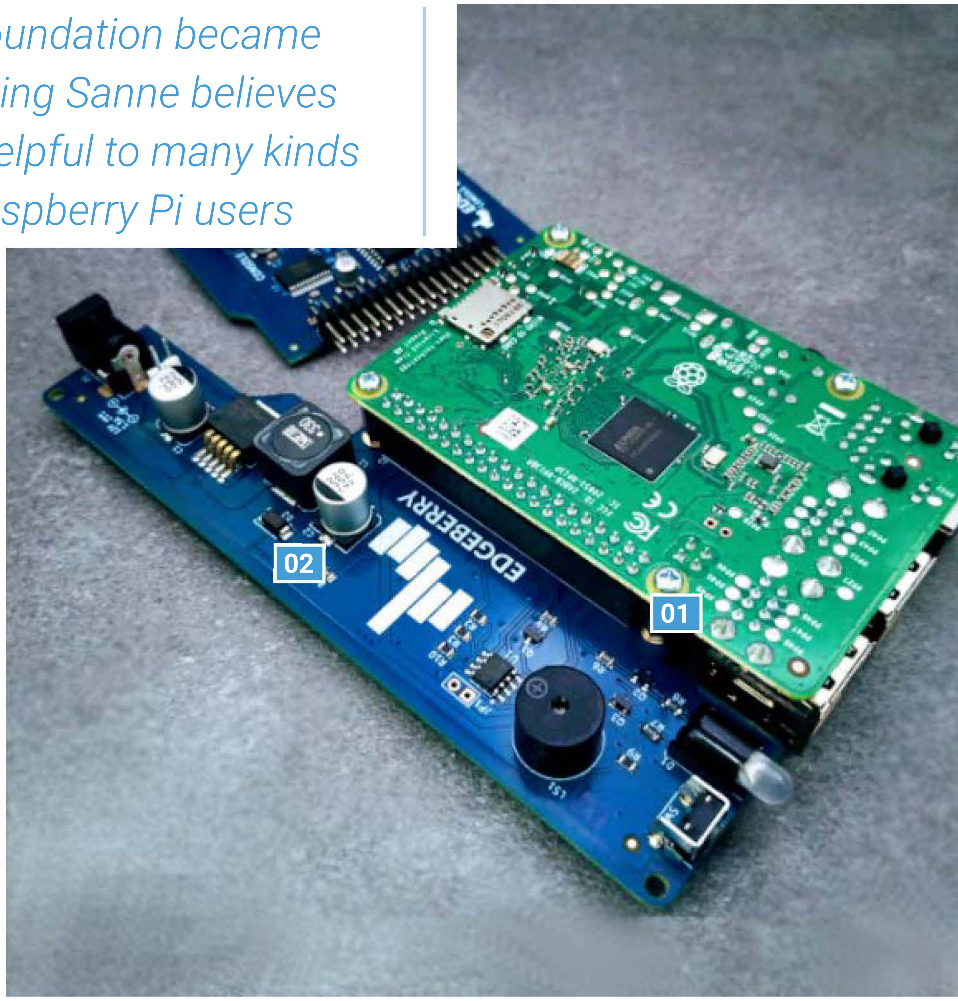
"It's a modular expansion system that turns Raspberry Pi into a robust, adaptable IoT edge device," Sanne tells us. "It combines a custom HAT-style board, a robust 3D printable enclosure, built-in user interface elements (like a status LED and buzzer), and a flexible hardware cartridge slot for application-specific electronics (leaving most of Raspberry Pi's beloved I/O pins intact). Edgeberry simplifies the messy part of getting from prototype to real-world deployment for trying out (and iterating on) an idea providing you with a stable, repeatable base you can trust... It's meant for anyone who wants to build connected devices that work reliably outside the lab environment."

▼ A Raspberry Pi Zero version was recently created by Sanne, too



▼ Edgeberry is quite compact, even when the expansion cards are added

This foundation became something Sanne believes can be helpful to many kinds of Raspberry Pi users

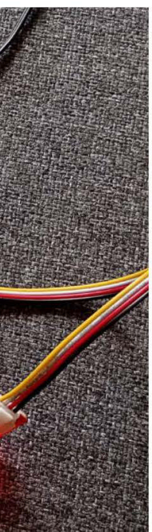


It's all open source too, available on GitHub (**rpimag.co/edgeberry**), so if you need to add more hardware yourself, you can use the KiCad template provided.

Real-world uses

"Edgeberry is already powering real deployments," Sanne says. "It is, for example, the foundation of the Freya Vivarium Control System. It's also been adopted in R&D environments where engineers need to quickly test ideas without worrying about power supplies, mounting, or reliability. It also has been tested in the classroom environment to give students hands-on experience with IoT concepts, without the fragility of breadboards and jumper wires. It fits wherever you need Raspberry Pi to cross into the physical world — cleanly and confidently. And I expect lots more applications to build on it in the future (even if just because I'm using it myself now, for everything)."

If you don't want to make an Edgeberry yourself, various kits are available from Elecrow, with some extra cartridges too.



Quick FACTS

- Edgeberry is part of the Powered by Raspberry Pi program
- Sanne has been using Raspberry Pi like this since it was released
- It was designed in KiCad and FreeCAD
- … Which Sanne claims have matured enough to make it easy
- As it's fully open source, you can make one yourself

01. A main board is attached to Raspberry Pi to add a simple set of extra features

02. The expansion cards are then slotted in as and when you need them

Supersense SenS2 dementia aid

Benignly tracking someone's activity levels using radar and Raspberry Pi could give families peace of mind, discovers **Rosie Hattersley**

Makers

James Brown and Matt Ash

Supersense founders James and Matt used their engineering skills to develop a means of helping ageing family members to live safely at home.

supersense.com

Many readers will recognise the dilemma: an ageing parent or family member who is determined to stay in their own home, maintain their independence, and not be a burden to others. Cost, convenience and quality of life for them and those providing care is a difficult balance all round. Cambridge startup Supersense's Matt Ash and James Brown have direct experience of this precise scenario and used this knowledge to create a Raspberry Pi-controlled hub to support families in which someone lives with dementia. SenS2 uses radar to check unobtrusively for routine activity. It features several different sensors and uses machine learning to plot the person's usual routines and flag up deviations that may signal something's up. Crucially, it also sends daily messages confirming all seems well, so contact from the service becomes second nature.

Sensitive subject

Medical experts and families recognise that patients with dementia should ideally continue to live in their own homes and be in a familiar environment as long as possible. Three quarters of GPs report that those who do lead longer, more enriched lives, prompting calls for technology to identify issues that might crop up.

The Supersense team's own experiences (**rpimag.co/supersenseblog**) showed family members living some distance away often understand the problems their loved ones face, but need to be forewarned of emerging issues. "With dementia, it's difficult for people to explain the problems they are having day to day, because they either forget or they just can't find the words," suggests Matt, recalling his and his siblings' experience of caring for their mum. "It wasn't quite clear what was wrong if she was sounding anxious, because she

01. The Supersense SenS2 hub uses radar to detect a dementia patient's movements at home

02. Sensors on Raspberry Pi 4 monitor light, heating, and humidity.

> *This Raspberry Pi 4-based radar hub can recognise movement through two or three walls of a home*

just couldn't explain herself." Visiting could often involve a kind of detective work to discern the underlying problem, since their mother could not articulate what was wrong. Supersense set about using technology to help families in that situation, "giving family carers a bit more information about how their loved ones are doing when they can't be there, but also guidance on how their loved one is progressing in terms of their symptoms and what they can do about it".

Matt observes that "people with dementia struggle with wearable technology" and also to adapt to new technology. This Raspberry Pi 4-based radar hub can recognise movement through two or three walls of a home and monitors temperature levels to ascertain whether the heating has come on. "It just sits in the living room and quietly collects information about their loved ones and their pattern of life. Are they up and about in the morning, have they had a bad night's sleep?"

▼ Families on the pilot scheme found the SenS2 hub a reassuring presence

Less intrusion

Using radar is considered to be less intrusive than having cameras or motion sensors placed strategically around the home. “The person moving just looks like a block moving around,” says Matt. “You can’t see what they’re wearing or what they’re watching on TV. It really respects the privacy of the individual.” SenS2 does not directly act as a fall detection device, but will send out a WhatsApp to the family if movement is not detected for a while and is out of character.

James says feedback from families affected by dementia has shaped both what the system does and how it works, “from avoiding the need for wearables or cameras, to testing ideas quickly in real homes”. Building SenS2 around Raspberry Pi 4 has been “a big help in making that possible, [letting] us move fast, learn from real situations, and build something that can flex as we go” and that provides “simple, helpful updates for families”.

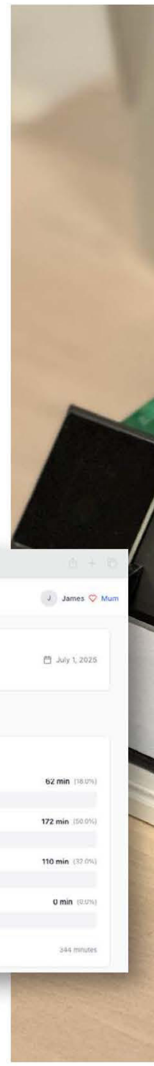
Quick FACTS

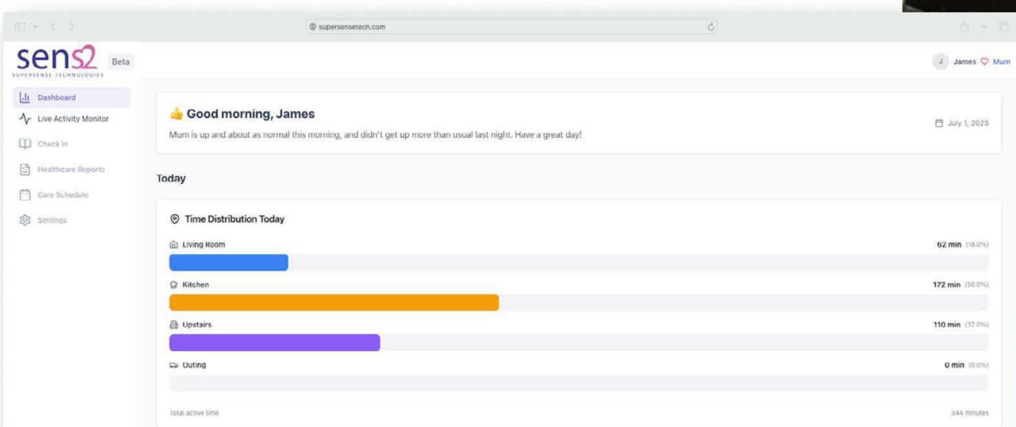
- WhatsApp messages and emails provide daily updates
- Most reassure that the dementia patient is up and about as usual
- Trends over time identify longer sleep patterns or reduced activity
- Supersense intends to add extra sensors based on trial feedback
- Changing patterns may suggest the disease is progressing or more help is needed

Reassuring presence

Supersense builds a model of how the individual’s pattern of life is and how it’s changing over time. For those who like graphs and data, there’s a dashboard showing longer-term trends that reveal changing daily patterns such as spending less time in the kitchen or sleeping more this month than last. Increased restlessness at night could prompt the family to take the parent out more during the day to help regulate diurnal patterns.

The simplicity of using a family and carer WhatsApp chat for updates, rather than yet another app, has proven popular with early testers who were given demos in carer cafes. “You don’t need the latest smartphone. You don’t need to download anything. You can just use WhatsApp and use natural language.” Most people just want a thumbs-up emoji to say everything’s normal.



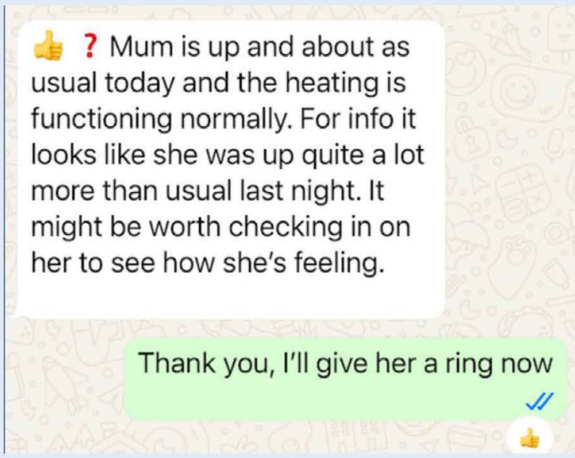


Logging into the Supersense dashboard shows useful activity trends over weeks and months

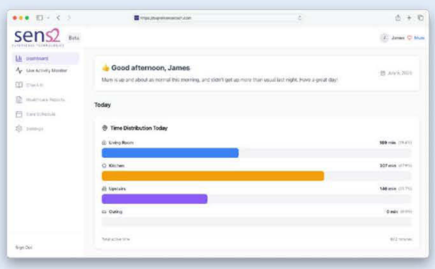
▼ Opening the case reveals the SenS2's internal electronics, including Raspberry Pi

Benign observation

1. Raspberry Pi 4 or 5, a custom circuit board, and sensors including radar and an AI HAT form the basis of the Supersense SenS2 hub.



2. WhatsApp messages in the form of an emoji or a natural language text reassure the family that all is well and activity levels are normal for the person being monitored.



3. The Supersense dashboard provides an overview of the patient's activity levels over time, helping identify issues or the possible need for extra visits or assistance.

DOOM SNES 2025

Emulating a special co-processor for a bygone console is no problem for RP2350. **Rob Zwetsloot** plays with power while knee-deep in the dead

Maker

Randal Linden

Randal has a long history porting and emulating games in the video game industry, and worked with Matt Cope and Gavin Bell of Bitmap Bureau on this project.

limitedrungames.com

bitmapbureau.com

Did you know that you can still technically make games for old consoles? It's been a popular thing to do for special editions and anniversaries (promo material for the film *Sonic the Hedgehog 3* was even distributed on working Mega Drive / Genesis cartridges). Now is the turn of DOOM on SNES, celebrating its 30th anniversary with an updated re-release.

"We've updated and expanded the game with tons of new features including 14 new levels, improved frame rate and performance, added circle-strafing, and even vibration effects with our new rumble game controller." Randal Linden of Limited Run Games (LRG) tells us. "The final project uses custom software, tools, firmware, and hardware to create a unique cartridge that works in any Super Nintendo!"

That custom hardware? A Raspberry Pi RP2350 microcontroller chip, of course.

> *The RP2350 offered advantages that were impossible to resist*

Super effects

DOOM on SNES was one of the few games to make use of the Super FX chip created for Star Fox (named Starwing in the UK, for weird legal reasons). This resulted in an extra hurdle in creating new, compatible carts for 30+-year-old console.

"[The Super FX chip] is a high-speed RISC co-processor that can be used to accelerate and optimise game logic, math, and graphics rendering," Randal tells us. He would know: he was basically a one-man team behind the original port to SNES. "When we started working on DOOM SNES 2025, our initial focus was on FPGAs to simulate the Super FX chip, but the RP2350 offered advantages that were impossible to resist: first and foremost was the lower cost of the device, both in terms of production but also for development. Next, the numerous GPIOs and efficient PIO programs enabled us to interface the various memory devices

01. The RP2350 chip replaces – and upgrades – a Super FX chip

02. It slots in to original SNES hardware and runs without any modding to the console used by the cartridge hardware with the Super NES easily and effectively. Finally, the Raspberry Pi SDK is open source and fully documented, which made our bring-up process nice and easy."

To prove the concept would work, LRG put together a mini console they referred to as Imp with off-the-shelf parts and a Raspberry Pi Pico 2.

"We were able to prove our new 'FX 3' system was rendering and playing the game properly, even though at this point we weren't connected to a SNES. We could also run lots of metrics and debugging quickly to get the FX 3 core stable," Randal mentions. The original SNES port used a slightly upgraded version of the Super FX chip created after Star Fox, colloquially referred to as Super FX 2, hence the FX 3 moniker. "We then designed our initial SNES PCB designs around socketing a Pimoroni PGA2350; it was a great way to prove the concept was then able to run on a SNES itself."

Rip and tear

In the final product, RP2350 performs three main functions according to Randal: a high-speed interface to multiple memory devices, simulation of Super FX, and 'graphics format conversion operations'.

"We have a pre-production tool which takes the Super NES program executable code, processes it, and generates C source code which is then compiled using the Raspberry Pi Pico SDK," Randal explains. "The resulting program includes logic which sets up PIO and DMA chains to

Quick FACTS

- Randal started the original port in his spare time
- id Software didn't know about it until it was ready
- The Super FX used in DOOM was clocked at 21MHz…
- … and had 512kbits (64kB) of RAM
- The Imp prototype used a tiny Qw/ST Pad

◀ The proof of concept was a little system nicknamed Imp, one of the enemies in DOOM

▼ The packaging apes the Super Nintendo game boxes of the time

respond to memory accesses in parallel with the Super FX simulation. After the frame buffer is generated by the Super FX simulation, the graphics conversion logic processes the image from its chunky graphics format into the required Super NES planar format."

Creating the game like this also allows them to add more content that wouldn't have fit on the original cart – supposedly only 16 bytes were free when it was released.

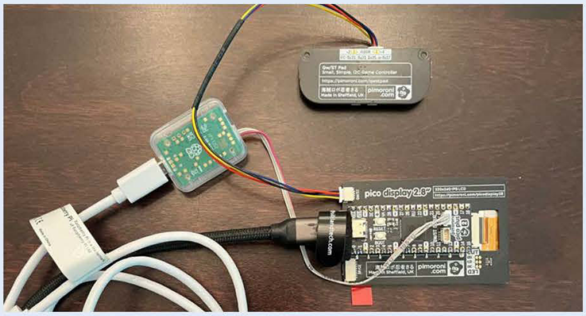
Playing with super power

"DOOM is widely recognised as one of the greatest games of all time, but when it first came out, the best (and only) way to play the game was using an expensive PC," Randal says. "At the time, many people didn't have a high-end computer, but they did have a Super Nintendo, so it was for them that I developed DOOM for SNES."

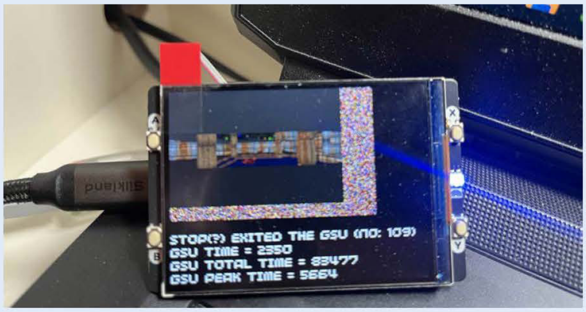
The game itself will be out and ready to slam into your slightly yellowed Super Nintendo later this year. The moons of Mars are waiting.

▼ It is much more powerful than any game released during the original run of the SNES

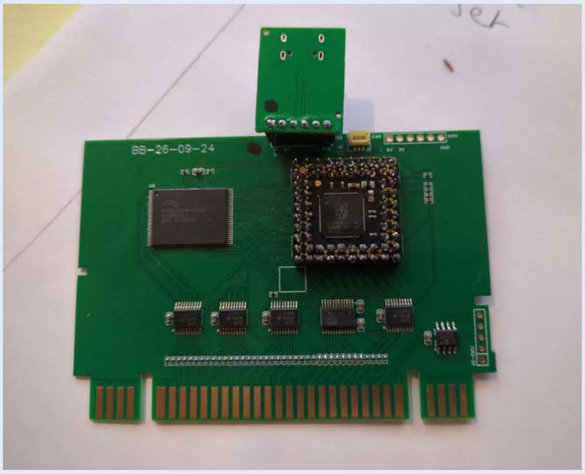
Enhancing a port



1. Like a lot of electronics projects, you can create a prototype of your final circuit with parts you have lying around.



2. Debugging is important, so once the circuit is created, it's time to test, test, test.



3. Finally, you can start creating custom PCBs and add an RP2350 into it, even if the SNES didn't originally use one.

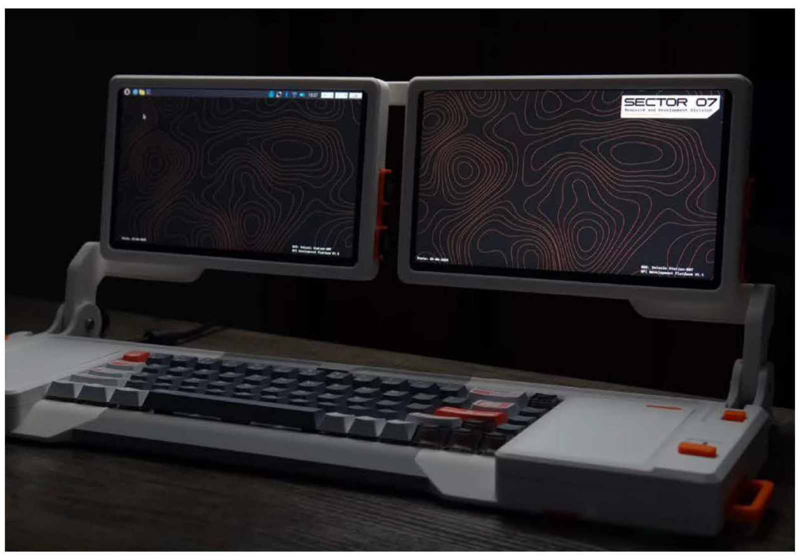
Dual-screen cyberdeck

By Sector 07

rpimag.co/DualScreenCyberdeck

We love a nice cyberdeck build, and this one by Sector 07 has everything we want from one. It's built on a Raspberry Pi 5, it's fully 3D-printed, and all the design files are on GitHub for anyone to come along and have a go themselves. Three custom circuit boards, designed in KiCad, break out the GPIO pins from the Raspberry Pi. These boards were made by PCBWay – if you need a tutorial or three on how to design and get boards manufactured, you're in luck, as we've featured getting designs fabricated by an external service in this and the previous two issues.

The maker has added a few unique touches to make this cyberdeck more useful than just a laptop. There's a linear slider, which can be used to control volume. There are four programmable buttons, for whatever hotkeys the user so wishes to program. And there's a rotary encoder with a push button, which is also programmable. There's a Qwiic connector at the back of the device for attaching I2C sensors. And there's also a full complement of GPIO pins exposed, so that you can connect any HAT. Both screens can rotate 90 degrees – there's a mechanical stop inside each mechanism so that the user doesn't accidentally over-rotate and damage the cables. The Raspberry Pi's SD card is accessible too, so you don't have to take the whole Raspberry Pi out to change OS images (though you can, as the maker has designed it to be easy to remove).



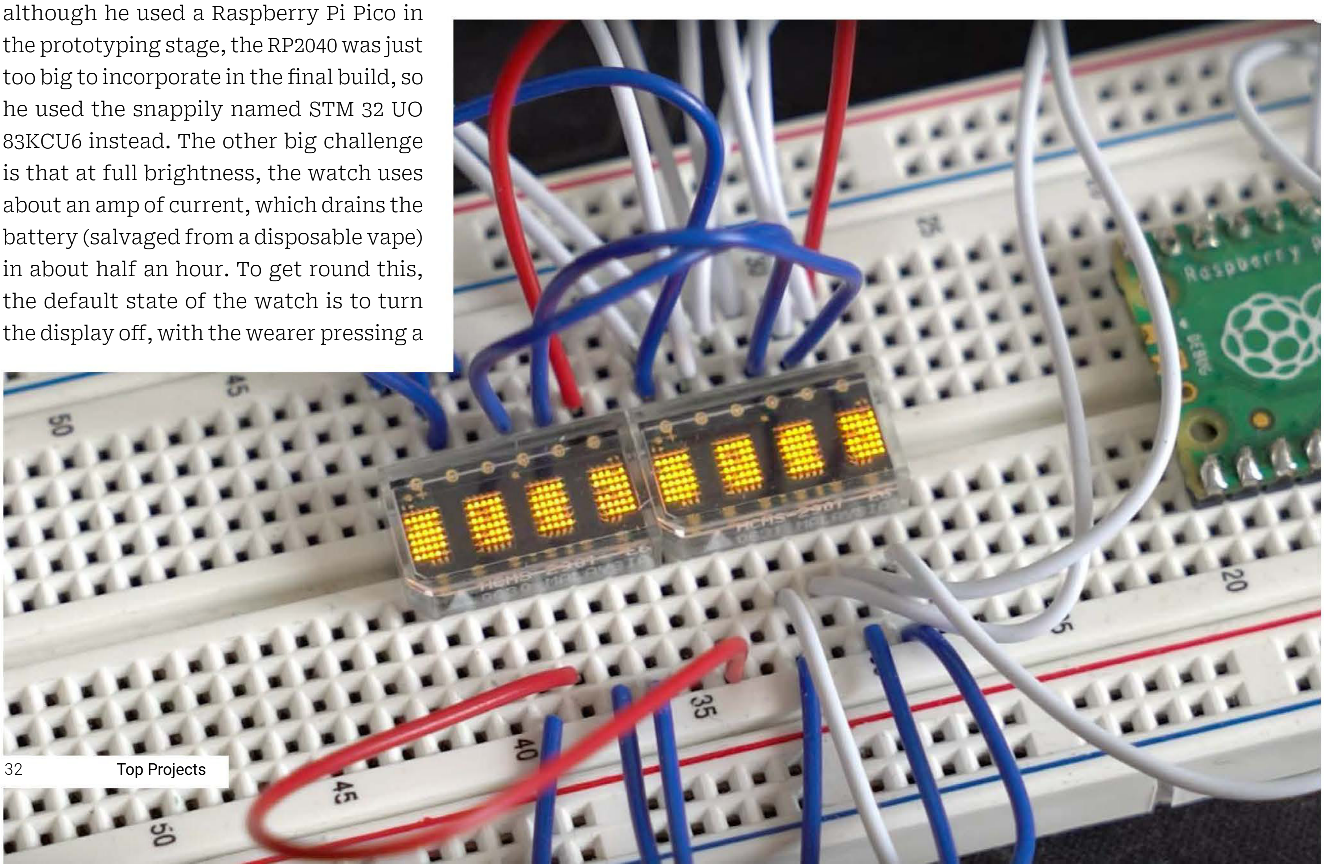
Twin screens mean you can code on one screen while watching David Bowie's 1978 *Musikladen* show on the other screen

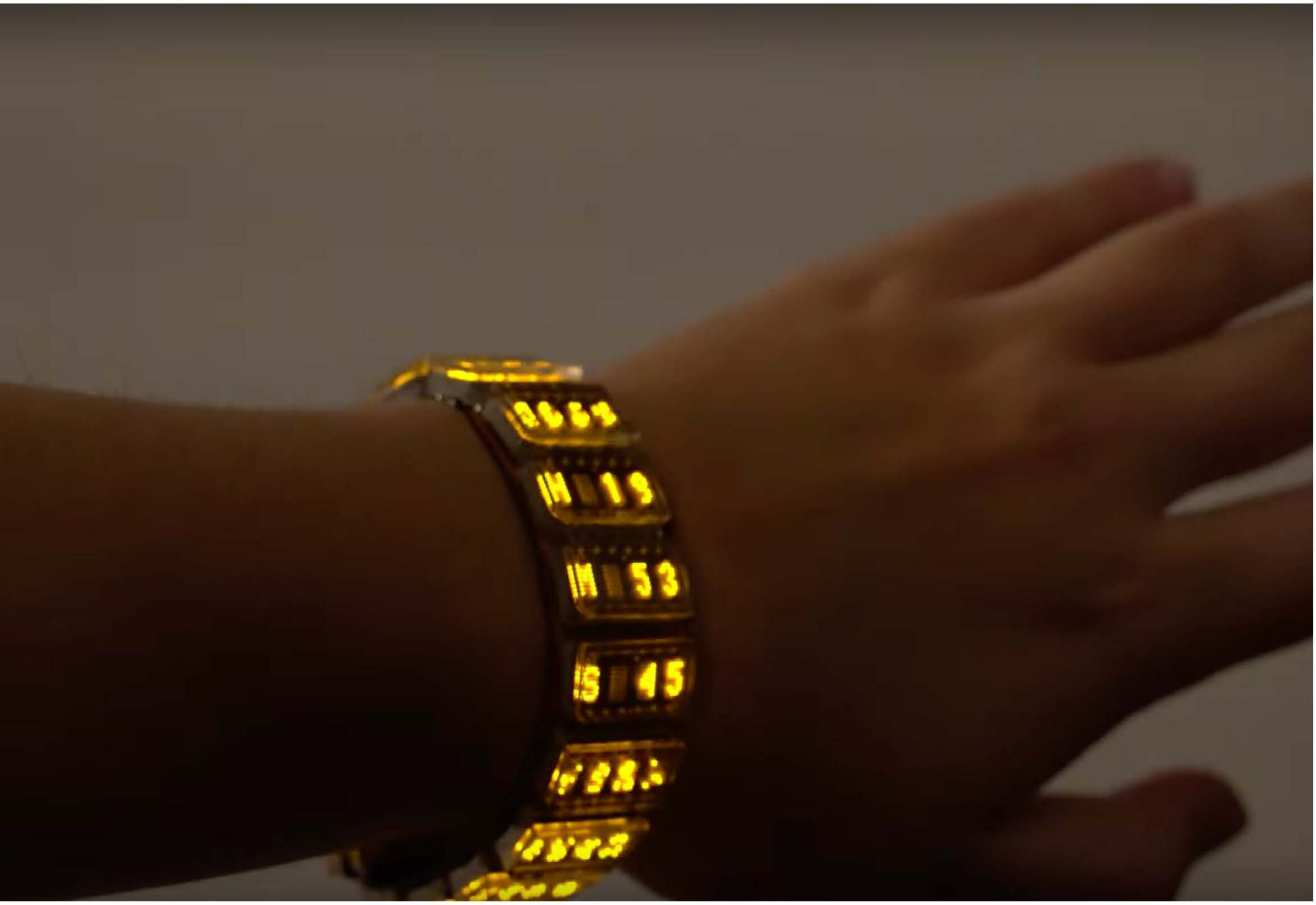
Flexible PCB watch

By Sahko

rpimag.co/FlexiblePCBWatch

Like too many of us, YouTuber Sahko has tons of components sitting around waiting to be used. To this end, he found a load of tiny HCMS2901 displays and thought he'd turn them into a watch, mounted on a flexible PCB so you can wear it around your wrist. It's a very cool idea: rather than a strap that holds a display, the whole thing is a continuous display, and it threw up some unusual engineering challenges. The first of these was the choice of microcontroller; although he used a Raspberry Pi Pico in the prototyping stage, the RP2040 was just too big to incorporate in the final build, so he used the snappily named STM 32 UO 83KCU6 instead. The other big challenge is that at full brightness, the watch uses about an amp of current, which drains the battery (salvaged from a disposable vape) in about half an hour. To get round this, the default state of the watch is to turn the display off, with the wearer pressing a button to get the time and the cool *Matrix*-style rainfall animation. Through-hole-soldering components on to a flexible PCB worn next to soft human skin brings its own problems. Last but not least, the cost of the displays – never mind the microcontroller and charging circuitry – came to $420, though Sahko recommends the similar HCMS3901 display as an alternative, as it's more easily available and plays more nicely with 3.3V.





▶ A three-pin header going into a three-socket header forms the watch clasp

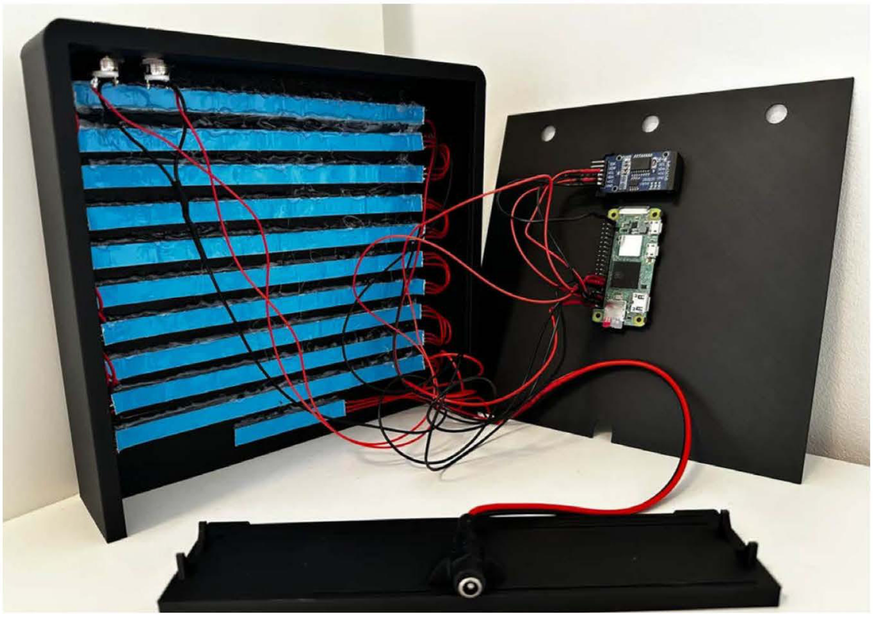
German word clock

By Reputila

rpimag.co/WordClock

We've seen word clocks previously in several configurations, including laser-cut, 3D-printed, and one intriguing version that used servos to bring the projected words in and out of focus, rather than revealing them through a grid like usual. However – and this is strange, given that we live in a big, varied world – we've never seen a word clock in a language other than English. At least, not until now. Reddit user Reputila built this example with their first 3D printer, a Bambu Lab P1S. The electronics are based on a Raspberry Pi Zero W that they had lying around, which enables the time to sync over Wi-Fi while the clock is in range of the internet, and there's a DS3231 real-time clock to make sure that the device doesn't lose the time when it's either not connected to the internet or power. As far as controls go, there are a couple of buttons to control the brightness and colour of the LEDS.

Somehow we feel we'd be more punctual if we told the time in German



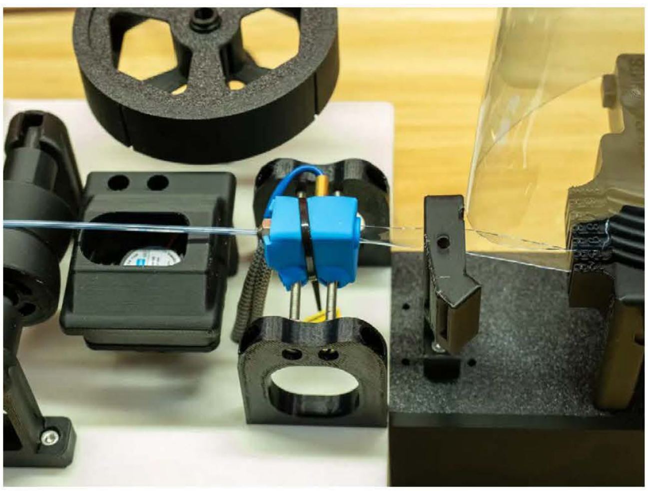

ESKISTAFÜNF
ZEHNBYGVORG
NACHVIERTEL
HALBVORNACH
EINSLMEZWEI
DREIAUJVIER
FÜNFTOSECHS
SIEBENLACHT
NEUNZEHNELF
ZWÖLFUNKUHR

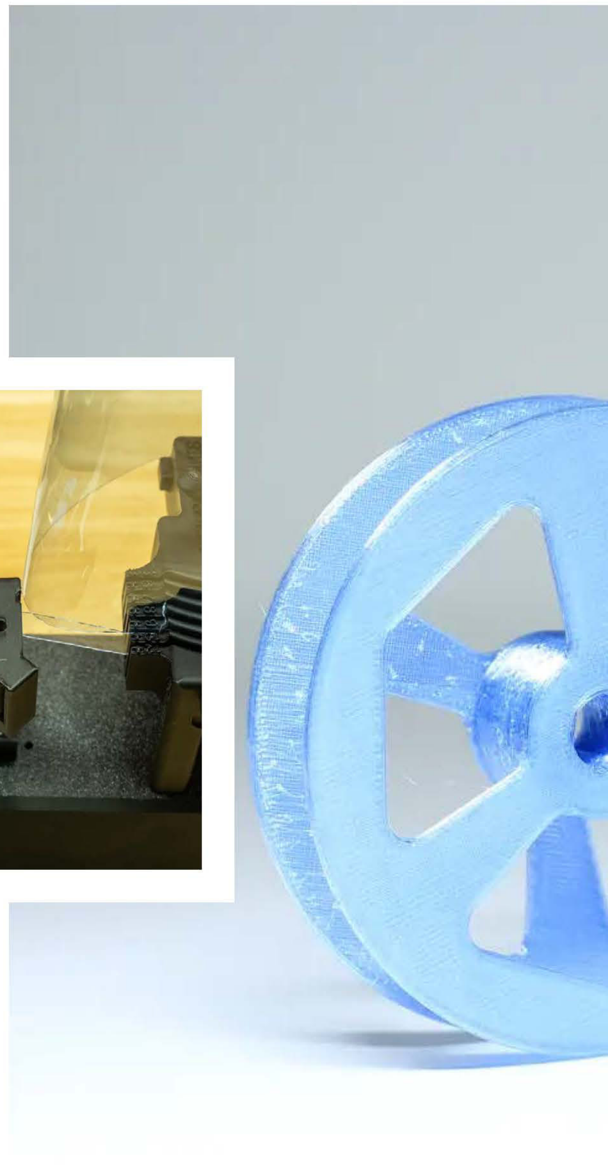
PETmachine

By Igor Tylman

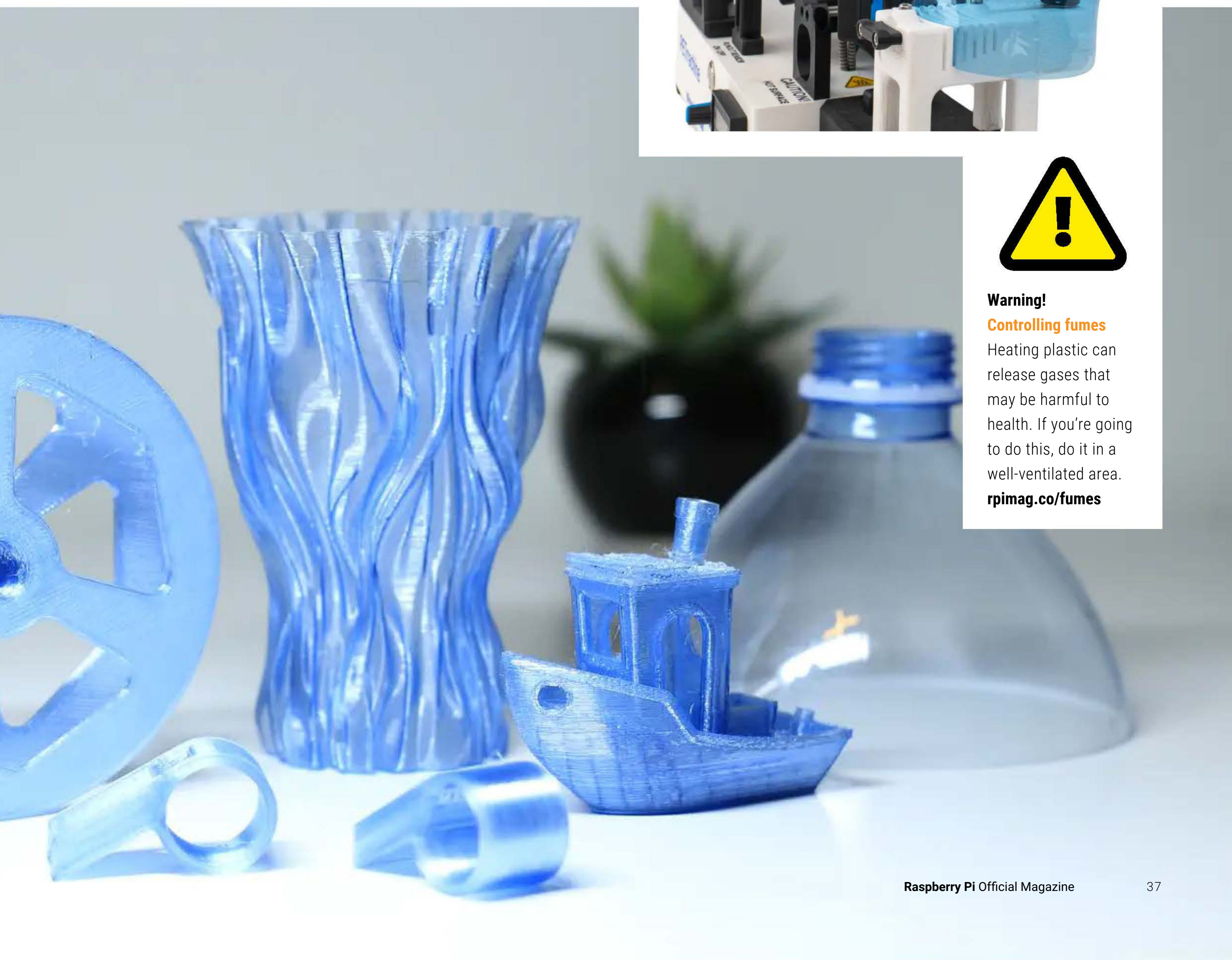
rpimag.co/PETmachine

The doom and gloom over climate change and ecological collapse does have one silver lining: the high numbers of plastic bottles that we're currently throwing away into landfill, or rivers and seas, are a brilliant potential source of 3D printing filament. Polyethylene terephthalate, or PET as it's abbreviated, is strong, light, non-reactive, and is a barrier to oxygen, which is the molecule that among other things enables bacteria to turn beer into vinegar. It's an incredibly useful material in food packaging for all these reasons, and in its modified form of PETG, it's also widely used in 3D printing. Igor Tylman has built a machine that can reliably turn empty PET bottles into 3D printing filament, potentially turning rubbish into gold. There's no place on Earth that doesn't have an excess of plastic bottles, so this technology really could lower the entry barriers to 3D printing for many people. It's pretty simple: the rotating bottle is continuously sliced into a single thin strip which runs through a heater to shape it into a consistent 1.75mm wide filament. The machine itself is 3D-printed, and the electronics and build instructions are available from Igor's website. The G at the end of PETG stands for 'glycol-modified', which is a process that makes the PET easier to work with; without this modification, any printer using this recycled filament will need an all-metal hotend capable of reaching 270°C, on top of which the makers recommend using a dual-drive extruder.





▶ The machine cuts a PET bottle into a continuous strip, feeds it into a heated extruder, and out comes a 1.75mm filament strand neatly wound onto a spool



Warning!

Controlling fumes

Heating plastic can release gases that may be harmful to health. If you're going to do this, do it in a well-ventilated area.

rpimag.co/fumes

Cute Chemistry

The hardest science just became the softest

rpimag.co/3DPrintCrochet

Crochet is not only a mindful way of active relaxation and a way of making nice woolly garments to keep yourself and your loved ones warm (or giant woolly resistors if you so wish – see HackSpace Magazine issue 40 for more) – it's also intensely mathematical. As Nicola King discovered in HackSpace Magazine issue 81, there's a whole world of equations wrapped up within every piece of crochet – it's just numbers and yarn, after all.

While it's a bit of a stretch of the imagination, we think that there's something reassuringly mathematical about the way a crochet pattern lends itself to being 3D-printed. Think of it like G-code for your fingers, if you will. If you squint, we hope you can see the influences in this 3D-printed crochet-style chemistry set, printed by Reddit user 59 Bassman and designed by NaraaPrints. The designer recommends printing at a tiny layer height of 0.16mm, with two walls and 15% infill; the set shown here was printed in PLA, with wood-fibre PLA used for the test tube stands and the corks.

STARTER ELECTRONICS

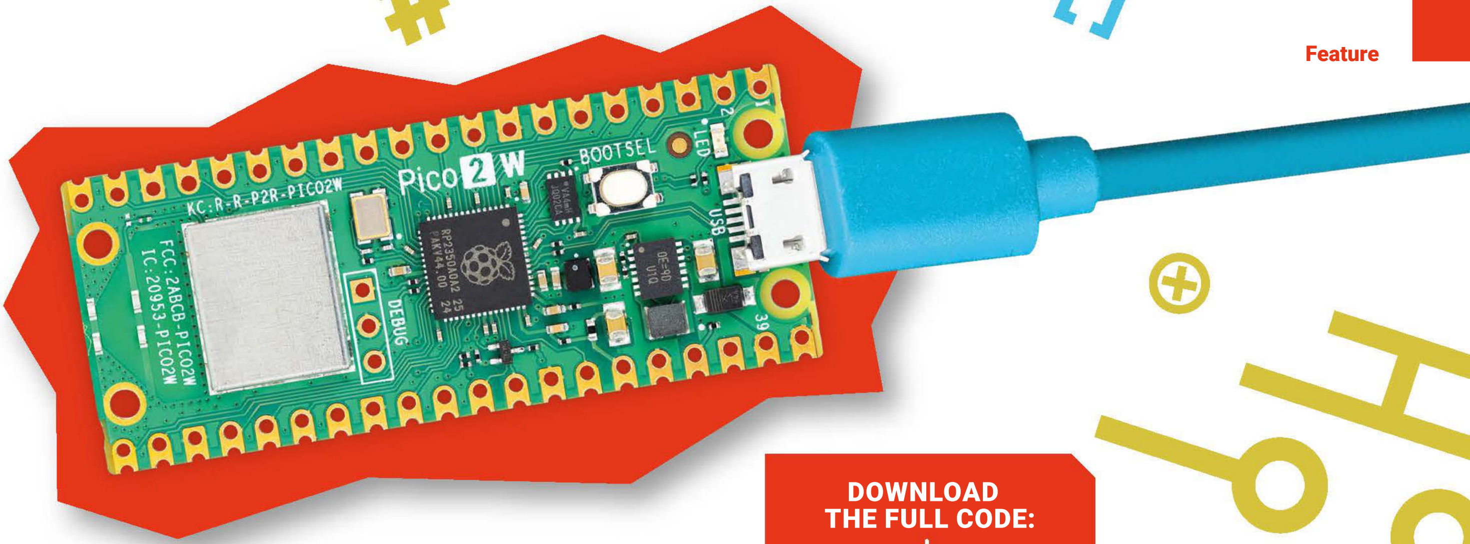
Get started with electronics with this simple one-dimensional game for Raspberry Pi or Raspberry Pi Pico

Maker

Ben Everard

Ben's main mission in life is to add more LEDs to the world. He is yet to find a project that isn't improved by adding more LEDs.

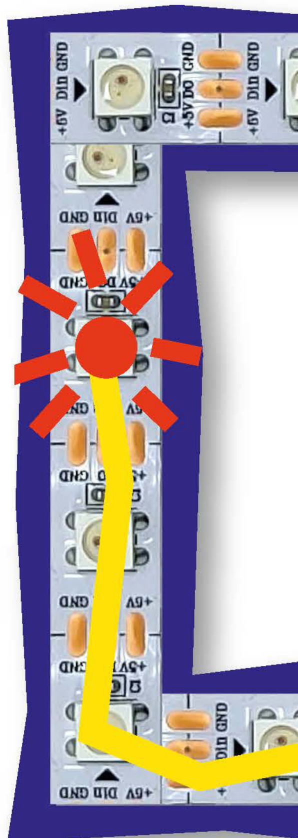
glowingart.co.uk



DOWNLOAD THE FULL CODE:

rpimag.co/github

Language: CircuitPython



Raspberry Pi computers and microcontrollers are great tools for interacting with things in the real world. We can add our own hardware or interface them with other electronics using their GPIO (General-Purpose Input/Output) pins. These pins can detect voltage (input), send out voltage (output), and even transfer more complex data using low-level protocols. In this article, we'll use these three different abilities to build a game.

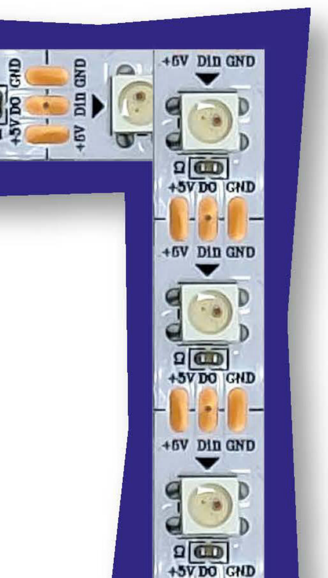
We're going to use CircuitPython, because this lets us use the same code on both Pico and Raspberry Pi

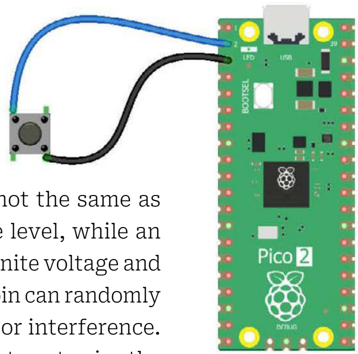
We're going to use CircuitPython, because this lets us use the same code on both Pico and Raspberry Pi. This allows you to use whatever hardware you already have, and it also helps you understand a little more about the capabilities of each device. If you've not used CircuitPython before, take a look at the boxes throughout this guide to get started.

Button basics

Let's start with one of the simplest uses for a GPIO pin: reading voltages. Most GPIO pins are digital, which means they don't measure the exact voltage level. Instead, they only detect whether the voltage is high (3.3V on Pico and Raspberry Pi) or low (0V). (Some pins on Pico can measure varying voltage levels – these are called analogue pins – but we won't be using those in this project.)

To read data using a GPIO pin, we need to build a simple circuit that gives us 3.3V when something is 'on' and 0V when it's 'off'.





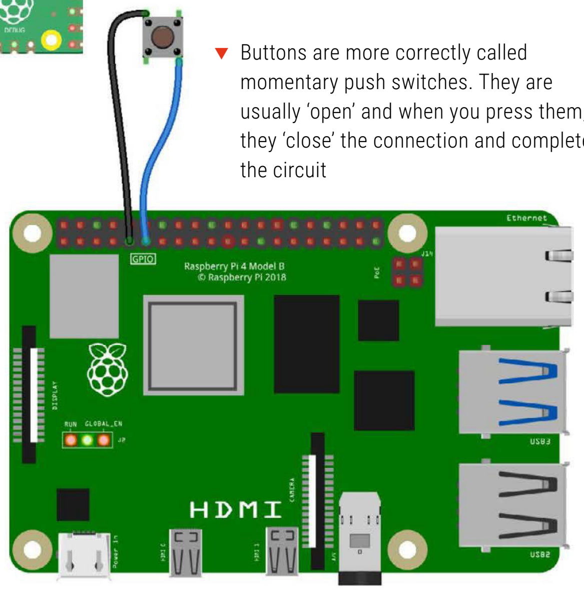
◀ **Figure 1:** If your button has four connections, use diagonally opposite ones

It's important to understand that 0V is not the same as an unconnected pin. 0V is a defined voltage level, while an unconnected pin (called 'floating') has no definite voltage and may behave unpredictably. In this state, the pin can randomly read as HIGH or LOW due to electrical noise or interference. This is why we use a pull-up or pull-down resistor – to give the pin a default, reliable state when nothing else is connected.

Figure 1 shows a simple circuit for connecting a button to a GPIO pin on Pico. When the button is not pressed, the GPIO pin is connected to 3.3V via the pull-up resistor so reads HIGH. When the button is pressed, it connects the GPIO pin directly to ground (the GND pin). This creates a clear path for current to flow to ground, and the GPIO reads a LOW signal (0V).

Since pull-up and pull-down resistors are very often needed with GPIO pins, both Pico and Raspberry Pi have internal resistors that you can use. You just have to initialise them in your program.

Now, with that theory behind us, let's connect a button to Pico. There are lots of different buttons you could use for this. If you want, you could even just use a wire to connect GND and a GPIO pin and attach and detach the wire to simulate pressing the button.



▼ Buttons are more correctly called momentary push switches. They are usually 'open' and when you press them, they 'close' the connection and complete the circuit

Using Raspberry Pi

You can use CircuitPython code on a Raspberry Pi by installing Blinka. Blinka acts as a compatibility layer, allowing CircuitPython features to work with regular Python. To install this, we need to first create a virtual environment.

```
$ python3 -m venv venv --system-site-packages
$ source venv/bin/activate
```

Next, set up your Raspberry Pi and install Blinka with these commands:

```
$ pip3 install --upgrade adafruit-python-shell
$ wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/raspi-blinka.py
$ sudo -E env PATH=$PATH python3 raspi-blinka.py
```

The first command installs a required package, the second downloads the Blinka installer script, and the third runs the installer.

You may be asked to reboot the machine when this runs.

Once the installation is complete, you can use Blinka to run CircuitPython code with the regular Python interpreter. Just make sure your virtual environment is activated. You can activate it at any time by running the following command in the terminal:

```
$ source venv/bin/activate
```

To run any of the code in this article, save it to a file and run:

```
$ python <filename.py>
```

The code to read this in Circuit Python is shown below.

The biggest difference between Pico and a Raspberry Pi computer is the pin numbering is a bit different. Pico pins are called GP0 to G22 while Raspberry Pi pins are D0 to D26. Not all the pins can be used for everything as some are reserved for particular functions. In general, the pins on Pico are more flexible than the pins on a Raspberry Pi computer.

We've provided both pin versions in the code below and you'll need to comment out (add a # at the start of the line) the one you're not using, and uncomment (remove the # at the start of the line) the one that you are using.

Buttons usually have either two or four connections. If your button has two connections, then you can connect it either way around. If it has four connections, then there are really two sets of two. The easiest way to do it is to use diagonally opposite connections – either diagonal works.

We'll show the direct wire connections for the circuit, but you can do this using wires or a breadboard.

```
 import board
 import digitalio
 import time

 #Pico
 BUTTON_PIN = board.GP1

 #Raspberry Pi
 #BUTTON_PIN = board.D17

 button = digitalio.DigitalInOut(board.GP14)
# Use appropriate GPIO pin
 button.direction = digitalio.Direction.INPUT
 button.pull = digitalio.Pull.UP  # Enables
internal pull-up resistor

 while True:
     if not button.value:  # Button pressed
(pin reads LOW)
         print("Button is pressed")
     else:
         print("Button is not pressed")
     time.sleep(0.1)
```

One slightly odd thing about this code is the line:

```
if not button.value
```

You might expect that to read `if button.value`, but we're using a pull-up which reads HIGH (which evaluates as `True`) when the button is pressed and LOW (i.e. `False`) when it is pressed. This might seem a bit backwards at first, but it helps if you think about what's going on at an electrical level.

Blink and you'll miss it

We've created a simple input device, and now let's look at the simplest output device: a light-emitting diode, or LED. LEDs light up when you put a voltage across them. However, LEDs are sensitive to too much current, which can damage them. You can calculate the exact resistor value needed using formulas based on voltages and current limits, but to keep things simple, just use a 330Ω resistor in series with the LED. **Figure 2** (overleaf) shows you how to connect up a resistor and an LED to a GPIO pin. Unlike pull-up or pull-down resistors, Pico doesn't provide internal current-limiting resistors for LEDs, so we need to add an external one.

CircuitPython on Pico

This tutorial will work on any version of Pico. To install CircuitPython on Pico, you first need to download the appropriate UF2 file from **rpimag.co/circuitpython**. Then, press and hold the BOOTSEL button on the Pico while connecting it to your computer via USB. Keep holding the button until the Pico is fully connected.

Once it's connected, you should see a new drive called 'RP2' appear. Drag and drop the UF2 file onto the RP2 drive. This installs CircuitPython onto Pico. After a few seconds, the RP2 drive will disappear, and a new drive called 'CIRCUITPY' will appear.

Now download the Mu code editor from **codewith.mu** and install it on your computer. You can use this to enter, edit, and save the code we're creating in this article. Click the Serial button in Mu to view the output from CircuitPython running on Pico. This can be particularly useful in finding any errors.

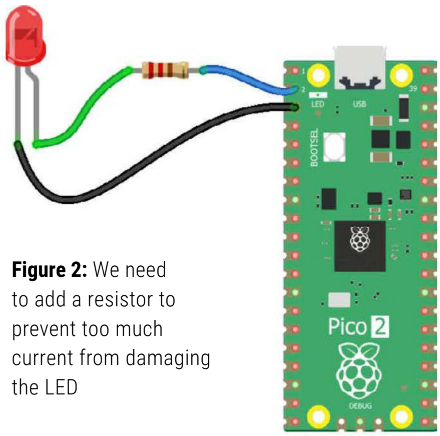


Figure 2: We need to add a resistor to prevent too much current from damaging the LED

Unlike buttons, LEDs have an orientation. That means that they have to be connected the right way around. Usually, there is a flat side and this marks the side that should be connected to ground (GND). You won't damage the LED by connecting it the wrong way around, so you can just pop it in and if it doesn't work, turn it around.

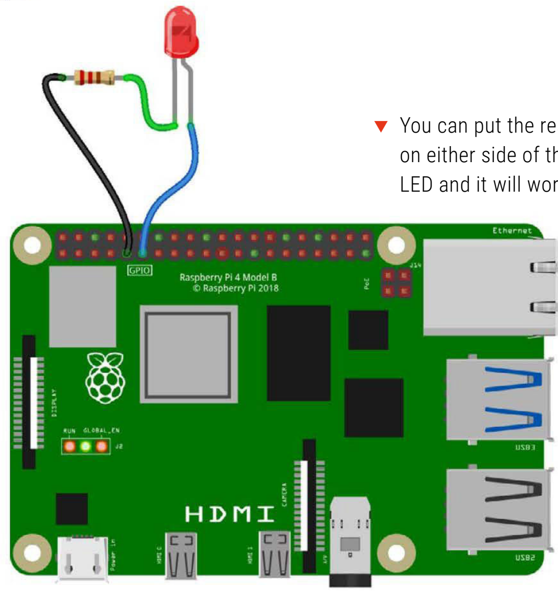
Here's some CircuitPython code to blink an LED:

```
import time
import board
import digitalio
#Pico
#LED_PIN = board.GP1

#Raspberry Pi
LED_PIN = board.D17

led = digitalio.DigitalInOut(LED_PIN)
led.direction = digitalio.Direction.OUTPUT

while True:
    led.value = True    # Turn LED on
    time.sleep(0.5)     # Wait for 0.5 seconds
    led.value = False   # Turn LED off
    time.sleep(0.5)     # Wait for 0.5 seconds
```

LED there be (more) light

Buttons and LEDs both work with a single bit of data. That means they can only be in one of two states. A button is either pressed or not pressed, and an LED is either on or off. Many other devices are more complex and need to send and receive more detailed information – for example, to control colours, positions, or movement.

We're going to use WS2812B LEDs, which are also known as NeoPixels. These allow you to control many RGB LEDs from a single GPIO pin using a special communication protocol. Each LED's colour can be changed by adjusting how much red, green, and blue light it emits.



You can put the resistor on either side of the LED and it will work

Wiring them up is simple. You just need to connect the 5V pin on the LED strip to the 5V output on Pico, GND to GND, and the Data In pin on the strip to a GPIO pin. Be aware that these LED strips are directional. This means you must connect the GPIO pin to the correct end, which is usually marked with 'DIN' or an arrow. If you connect to the wrong end, the LEDs won't respond.

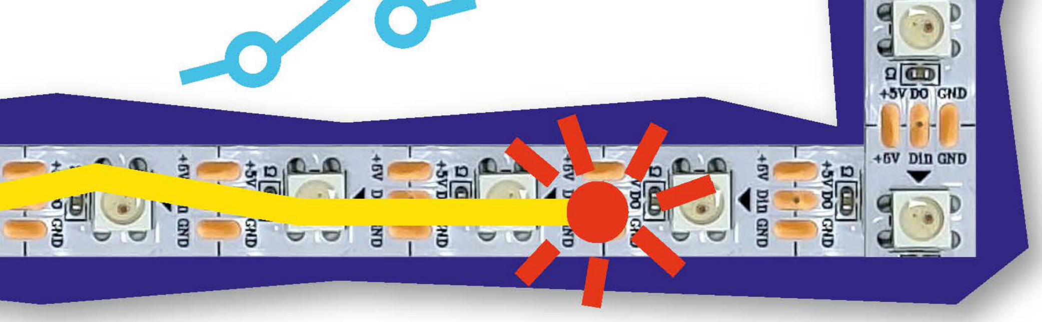
WS2812B LEDs come in a lot of different forms, but when you get them, there is typically a wire with a three-pin connector on the end. Usually, the socket end of the connector is on the Data In side, but this isn't completely standard. For testing things out, we like to use pin-to-socket headers and we can poke the pin into the socket on the LED strip, then connect the socket to a GPIO on either Pico or Raspberry Pi.

Since these have a slightly more complex communications protocol, we need to add a module that implements this protocol. The process for adding modules is a little different on each platform.

On Pico, head to **rpimag.co/circpylibraries** and download the bundle for the version of CircuitPython you have. This will be a zip file. Extract it and copy the **neopixel.mpy** file from within the **lib** folder to the **lib** folder on the CIRCUITPY drive.

On Raspberry Pi, make sure you're in the virtual environment you created when you set up Blinka, then run:

```
pip install adafruit-circuitpython-neopixel-spi
```



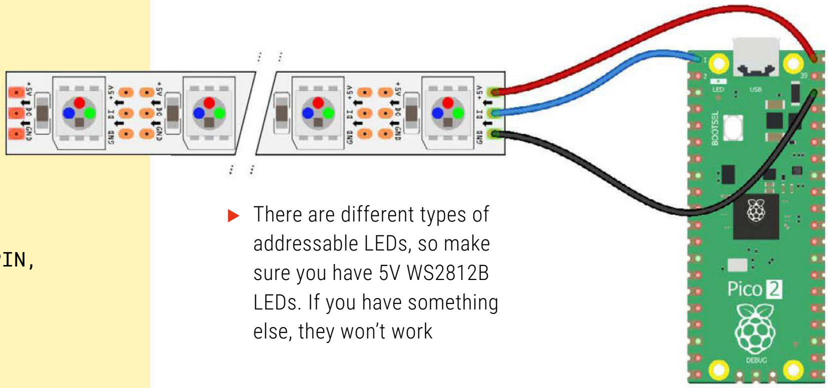
Here is some code to test everything out by flashing up some colours.

Initialising WS2812B LEDs is a bit different on each platform, so there are two blocks of code: one for Raspberry Pi and one for Pico.

```
import board
import busio
import time

NUM_PIXELS = 10
BRIGHTNESS = 0.5

#For Pico
#import neopixel
#PIXEL_PIN = board.GP0
#BUTTON_PIN = board.GP1
#pixels = neopixel.NeoPixel(PIXEL_PIN,
NUM_PIXELS, brightness=BRIGHTNESS,
auto_write=True)

#For Raspberry Pi
#import neopixel_spi
#PIXEL_PIN = board.D10
#CLOCK_PIN = board.D11
#pixels = neopixel_spi.NeoPixel_SPI(busio.
SPI(clock=CLOCK_PIN, MOSI=PIXEL_PIN), 10)

colors = [
    (255, 0, 0),     # Red
    (0, 255, 0),     # Green
    (0, 0, 255),     # Blue
    (100, 100, 100) # White
]

while True:
    for color in colors:
        pixels.fill(color)
        time.sleep(0.5)
```

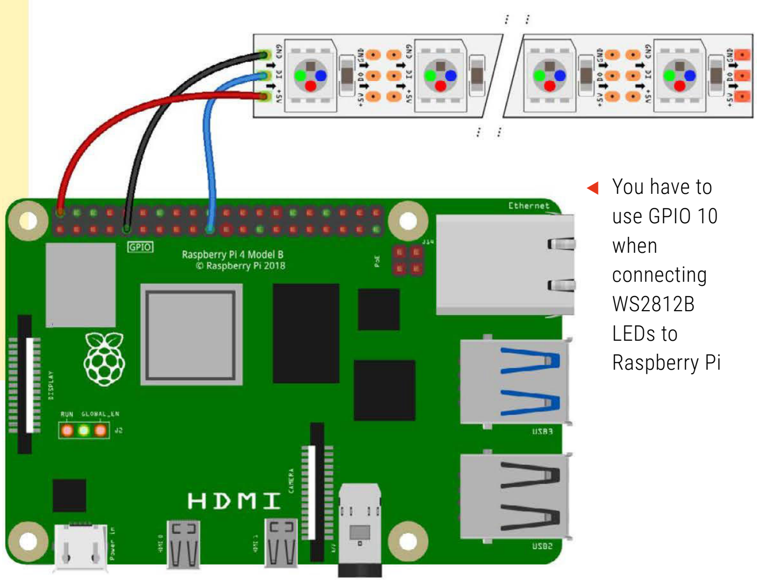
On Pico, we can use any pins for the LEDs; however, on Raspberry Pi we are more limited. We have to use GPIO 10 for the pixel pin because this one is attached to the SPI peripheral, and it's this that we're using to send the data out to the LEDs.

On Raspberry Pi we've used two pins for the LEDs, but we'll leave one of them unconnected. It's because we're using a bit of hardware designed to do something different to control the NeoPixels. We do have more flexibility with the button pin, but can't just use any as some are reserved for specific functions.

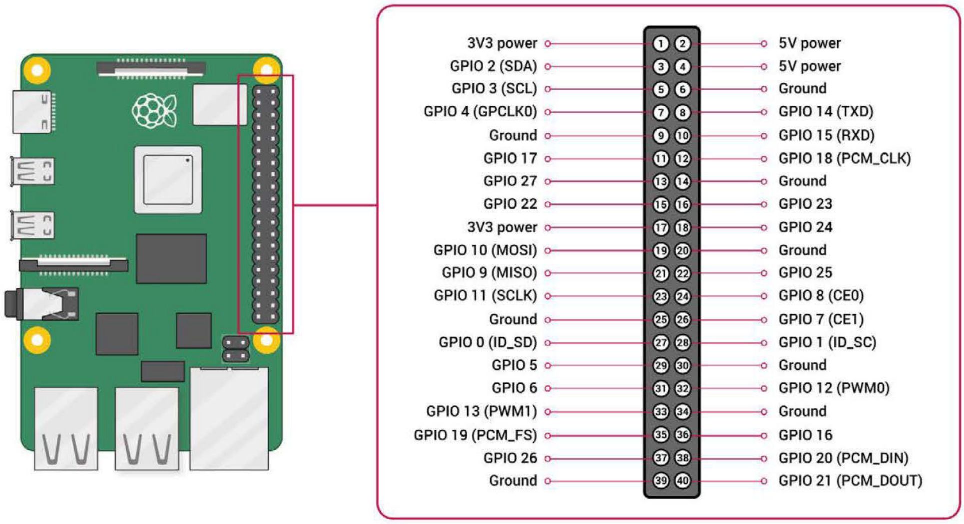


There are different types of addressable LEDs, so make sure you have 5V WS2812B LEDs. If you have something else, they won't work

> WS2812B LEDs come in a lot of different forms



You have to use GPIO 10 when connecting WS2812B LEDs to Raspberry Pi



Pico's pins are labelled on the underside, but for Raspberry Pi, you'll need to refer to a pinout diagram

Ready, set, glow!

Now let's add a button and make a game. This is a cross between Chrome's dinosaur game and a set of Christmas lights. The idea is simple. The lights will light up green, and a red pixel will randomly appear and move towards the end. You have to press the button in time to 'jump' over the red pixel when it reaches the end of the strip. If you're successful, the strip will light up white.

The code for this is:

```
 import board
 import busio
 import time
 import random
 import digitalio

 NUM_PIXELS = 10
 BRIGHTNESS = 0.5

 #For Pico
 #import neopixel
 #PIXEL_PIN = board.GP0
 #BUTTON_PIN = board.GP1
 #pixels = neopixel.NeoPixel(PIXEL_PIN,
NUM_PIXELS, brightness=BRIGHTNESS,
auto_write=False)

 #For Raspberry Pi
 #import neopixel_spi
 #PIXEL_PIN = board.D10
 #CLOCK_PIN = board.D11
 #BUTTON_PIN = board.D17
 #pixels = neopixel_spi.NeoPixel_SPI(busio.
SPI(clock=CLOCK_PIN, MOSI=PIXEL_PIN), 10)

 DELAY = 0.3

 GREEN = (0, 255, 0)
 RED = (255, 0, 0)
 WHITE = (100, 100, 100)
 FLASH_DURATION = 0.6

 # Button setup
 button = digitalio.DigitalInOut(BUTTON_PIN)
 button.direction = digitalio.Direction.INPUT
 button.pull = digitalio.Pull.UP

 def flash_color(color, duration=
FLASH_DURATION):
     pixels.fill(color)
     pixels.show()
     time.sleep(duration)

 while True:
     red_pos = 0

     # Move the red pixel forward
     while red_pos < NUM_PIXELS:
         # Draw strip
         for i in range(NUM_PIXELS):
             if i == red_pos:
                 pixels[i] = RED
```

```
            else:
                pixels[i] = GREEN
        pixels.show()

        # Wait for DELAY, check for button
press
        t0 = time.monotonic()
        pressed = False
        while time.monotonic() - t0 < DELAY:
            if not button.value:
                pressed = True
                break

        if pressed:
            if red_pos == NUM_PIXELS - 1:
                # SUCCESS!
                flash_color(WHITE)
            else:
                # FAIL!
                flash_color(RED)
            break

        red_pos += 1
    else:
        # If it reaches the end and you
didn't press, flash red for "miss"
        flash_color(RED)

    # Small pause before restarting
    time.sleep(0.7)
```

Here ends our whirlwind tour of electronics on Pico and Raspberry Pi. We've skipped over some of the detail to help you get up and running quickly, but hopefully you now know the basics of connecting other hardware to these two devices.

You've also probably got a bit more of an idea about the differences between Pico and Raspberry Pi. The former has more flexible I/O options and a far simpler software setup, and the latter has several orders of magnitude more processing power. While there is some overlap between which projects work on which device, they are really very different platforms.

If you take the skills you've learned in this article further, we'd love to hear from you. Every Monday we take to social media to find interesting projects, so if you're posting pictures at the start of the week, tag #MakerMonday.

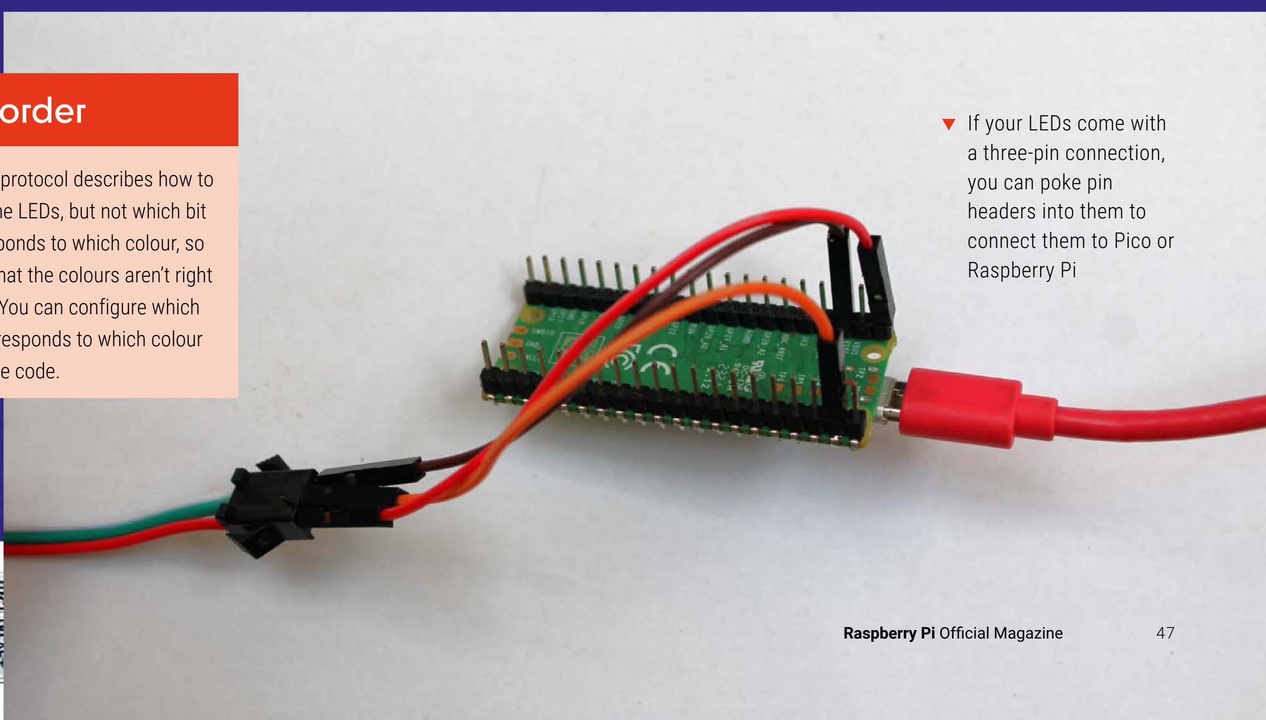
Voltage problems

WS2812B LEDs require the digital signal voltage to be at least 70% of the supply voltage. This means that it should be at least 3.5V. Pico and Raspberry Pi operate at 3.3V. This is so close that almost all the time you'll be fine connecting the data-in line to the GPIO pin. Just occasionally, you'll get a strip that behaves strangely.

You can do one of two things. You can cross your fingers and hope for the best (and this is what most people do) or you can correct the problem, which typically involves using level shifters. We almost always just use a 3.3V data signal and hope for the best, unless we're creating a project that has to work reliably, especially in adverse conditions (such as outside in the rain).

Colour order

The WS2812B protocol describes how to send data to the LEDs, but not which bit of data corresponds to which colour, so you may find that the colours aren't right on your LEDs. You can configure which bit of data corresponds to which colour by changing the code.



▼ If your LEDs come with a three-pin connection, you can poke pin headers into them to connect them to Pico or Raspberry Pi

Part 01

Simple Electronics with GPIO Zero: In the beginning

Connect electronic components to your Raspberry Pi and write code to interact with the real world

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

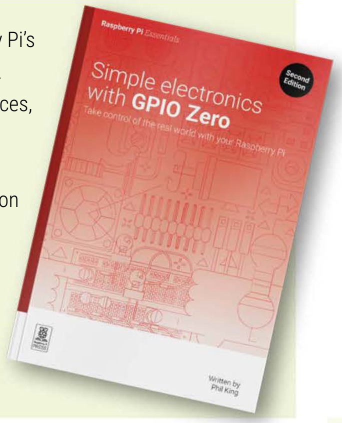
Raspberry Pi is a great platform for learning computing. Whether that's writing your own programs or building a media server, Raspberry Pi has the tools, resources, and community support to help you learn how to build what you want. Raspberry Pi is also very good at *physical computing* – programming and interacting with the real world through electronics. As the name suggests, physical computing is all about controlling things in the physical world with your programs: using hardware alongside software. When you set the program on your washing machine, change the temperature on your programmable thermostat, or press a button at traffic lights to cross the road safely, you're using physical computing.

Electronic circuits are the physical part of a physical computing project. You'll connect these circuits to your Raspberry Pi, which, together with the code you'll write, is the computing part of the project. These circuits can be simple or very complex and are made up of electronic components such as LEDs, buzzers, buttons, resistors, capacitors, and even integrated circuit (IC) chips.

At its simplest, an electronic circuit lets you route electricity to certain components in a specific order, from the positive end of a circuit to the ground (or zero volts) end. Think of a light in your house: the electricity passes through it, so it lights up. You can add a switch that breaks the circuit, so it only lights up when you press the switch. Now it's an interactive electronic circuit.

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.



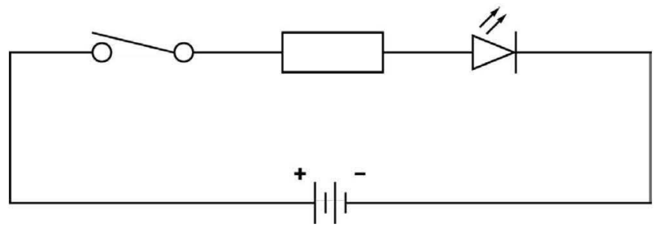
rpimag.co/gpiozerobook

Reading circuit diagrams

Building a circuit can be easy if you know what you're doing, but if you're making a new circuit or are new to electronics in general, you'll most likely have to refer to a circuit diagram. This is a common way you'll see a circuit represented, and these diagrams are much easier to read and understand than a photo of a circuit. However, components are represented with symbols which you'll need to learn or look up.

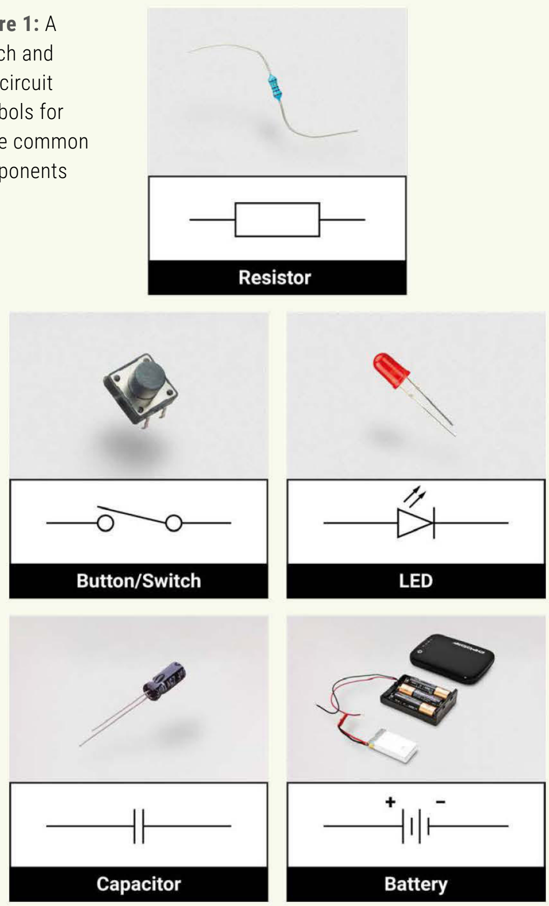
> *If you're making a new circuit, you'll most likely have to refer to a circuit diagram*

Figure 1 is an example of a light circuit. Here we have a power source (a battery in this circuit), a switch, a resistor, and an LED. The lines represent how the circuits are connected, either via wire or other means. Some components can be inserted any way round, such as the resistor or switch. However, others have a specific orientation, such as the LED. LED stands for *Light-Emitting Diode*, and diodes only let electricity flow freely in one direction; luckily, real-life LEDs have markers such as a longer leg or a flat edge to indicate which side is positive, making them easier to wire up.



▲ **Figure 1:** A switch and LED circuit

▶ Symbols for some common components



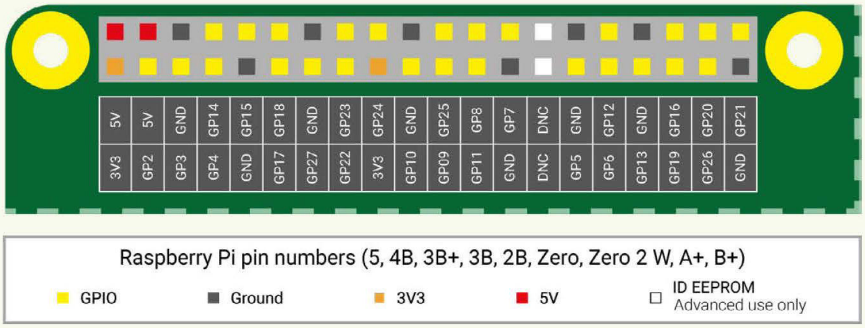


Figure 2: Raspberry Pi GPIO pinout

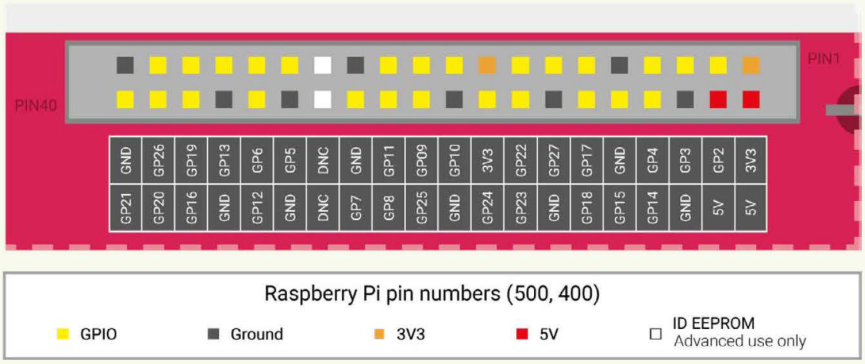


Figure 3: Raspberry Pi 400 and 500 GPIO pinout

Introducing the GPIO header

At the top of Raspberry Pi's circuit board, or at the back of a Raspberry Pi 400 or 500, you'll find two rows of metal pins. This is the GPIO (general-purpose input/output) header and it's there so you can connect electronic components to the Raspberry Pi. As the name suggests, these pins can be used for both input and output.

Raspberry Pi's GPIO header is made up of 40 pins, as shown in **Figure 2**. Some pins are available for you to use in your physical computing projects, some pins provide power, and other pins are used to communicate with add-on hardware.

The Raspberry Pi 400 and 500 compact keyboard computers have the same GPIO header with all the same pins, but it's turned upside-down compared to other Raspberry Pi models. **Figure 3** assumes you're looking at the GPIO header from the back of Raspberry Pi 400 or 500. Always double-check your wiring when connecting anything to the GPIO header on one of the compact computer models – it's easy to forget, despite the Pin 40 and Pin 1 labels on the case!

Raspberry Pi Zero 2 W has a GPIO header too, but doesn't necessarily have header pins attached. If you want to do physical computing with Raspberry Pi Zero 2 W, or another model in the Raspberry Pi Zero family, you'll need to solder the pins into place using a soldering iron. If that sounds a little adventurous for now, check with an approved Raspberry Pi reseller for a Raspberry Pi Zero 2 WH with the header pins already soldered into place for you.

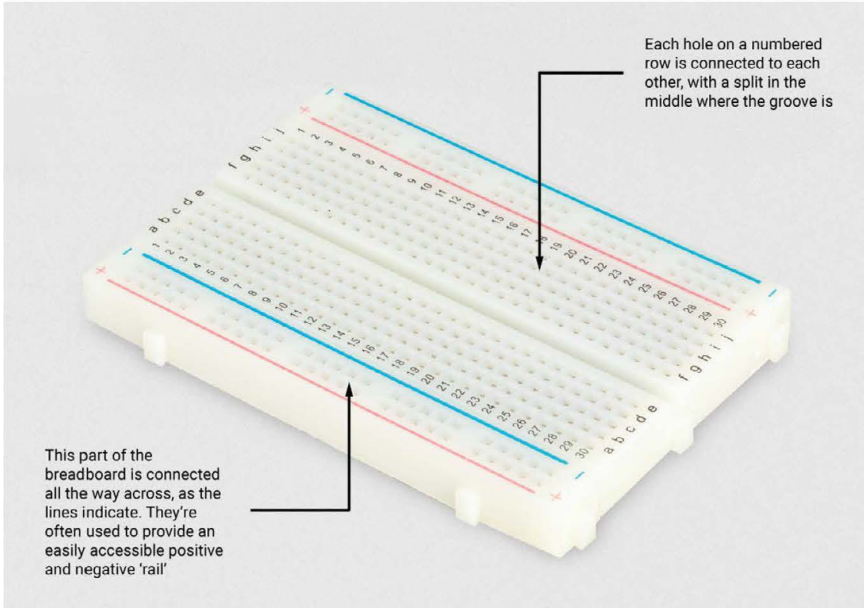
Raspberry Pi and electronic circuits

Involving a Raspberry Pi computer in a circuit is quite easy. At its most basic, it can provide power to a circuit, as well as a ground (abbreviated as GND) end through the GPIO pins. Some pins are always powered, mostly at 3.3V, and several pins offer a ground connection. Most pins can be programmed to create or recognise a HIGH or LOW signal, though; in the case of the Raspberry Pi, a HIGH signal is 3.3V and a LOW signal is ground or 0V.

In an LED circuit, you can wire up an LED to a 3.3V pin and a ground pin and it will turn on, but you will need a low-value resistor (around 330Ω is good) in there somewhere to keep from burning out the LED. If you instead put the positive end of the LED onto a programmable GPIO pin, you can only turn it on by running some code that makes that pin go to HIGH. We'll look at controlling LEDs in the next part of this series.

Staying grounded

You'll sometimes see ground referred to as negative, particularly in descriptions of a battery's positive and negative terminal, but also as the minus symbol (–) on a breadboard and some components. In the kind of circuits you'll see in this tutorial series, you'll be working with 5 volts, 3.3 volts, and a 0 volts (ground).



▲ **Figure 4:** A prototyping breadboard

Wiring up a circuit to a Raspberry Pi computer is simple. To create the physical circuits in the guides throughout this series, we're using solderless prototyping breadboards, as shown in **Figure 4**. These allow you to insert components and wires to connect them all together, without having to fix them permanently. You can modify your circuits and completely reuse your components because of this.

Using GPIO Zero

Once the components are all hooked up to your Raspberry Pi, you need to be able to control them. Raspberry Pi is set up to allow you to program it with the Python language. GPIO Zero makes it easy to program components in Python. It comes pre-installed in the latest Raspberry Pi OS desktop image. If you don't have it, however, you can install GPIO Zero manually: after performing a package list update by entering `sudo apt update` in a terminal, run `sudo apt install python3-gpiozero`.

GPIO Zero was created to simplify the process of physical computing, helping new coders to learn. It's a Python library which builds upon the existing GPIO libraries RPi.GPIO and pigpio. However, while those libraries provide an interface to the GPIO pins themselves, GPIO Zero sits above them and provides a way to interface to the devices that you connect to those pins.

This simplifies thinking about physical computing. Consider wiring a simple push button to GPIO 4 and ground pins. In order to react to this button, we need to know that the pin should be configured with a pull-up resistor, and that the pin state when the button is pushed will be LOW.

Here's what this would look like in the classic RPi.GPIO library:

```
from RPi import GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.IN, GPIO.PUD_UP)
GPIO.wait_for_edge(4, GPIO.FALLING)
print("Button pressed")
```

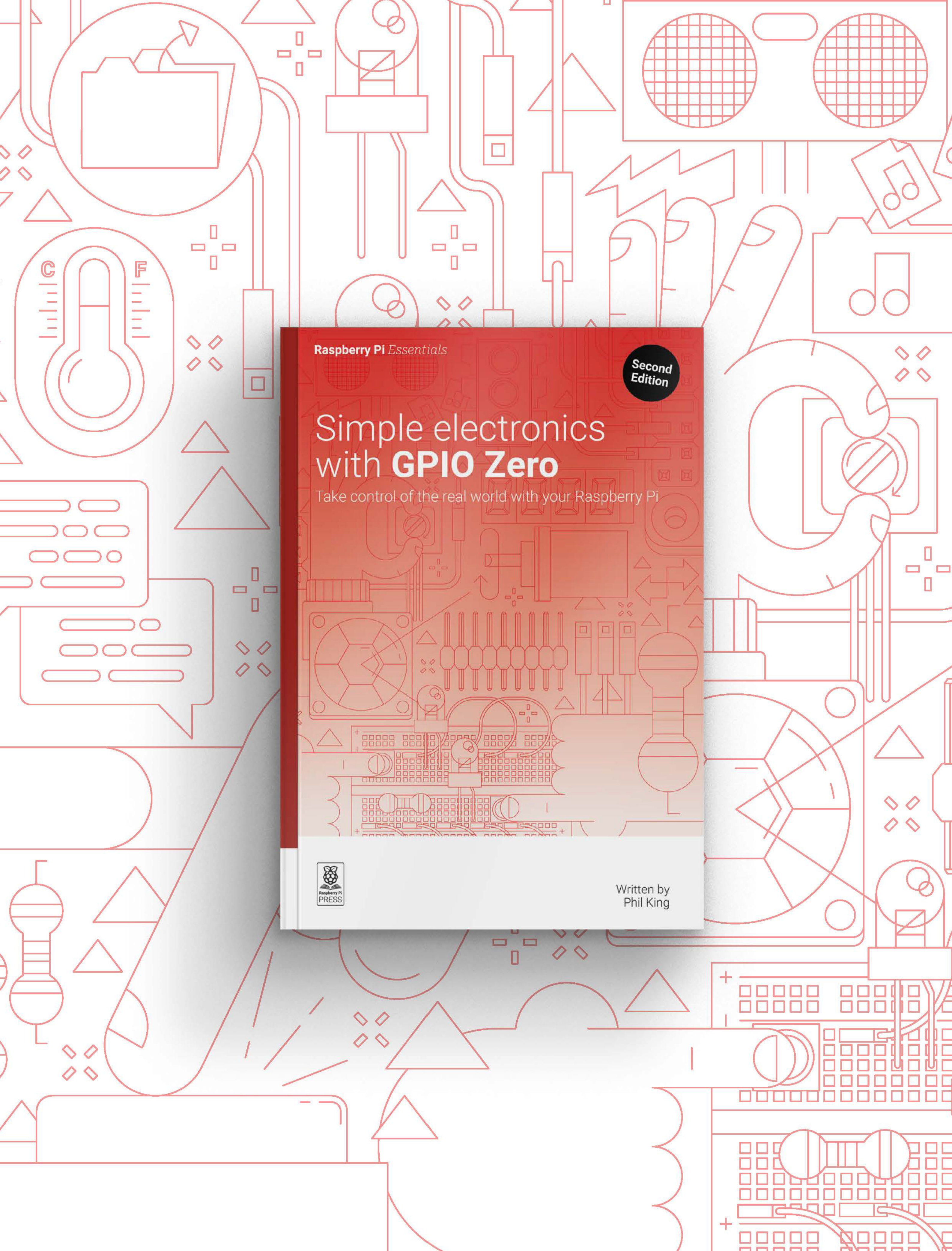
To complete beginners, there's quite a lot going on there, which gets in the way of experimenting with it and even learning the simple logic required. Here's the equivalent code in GPIO Zero:

```
from gpiozero import Button

btn = Button(4)
btn.wait_for_press()
print("Button pressed")
```

The *boilerplate*, the setup code you must write without necessarily understanding its inner workings, is reduced to the bare minimum that we need. The name 'GPIO Zero' derives from this 'zero boilerplate' philosophy, which was first espoused by Daniel Pope's Pygame Zero library.

The logic is straightforward, with no curious inversion of the input value.

So, now you've learnt about GPIO Zero and how it makes coding much simpler, it's time to get started doing some physical computing with it. Next time, we'll show you how to wire up some LEDs on a breadboard and control them using GPIO Zero's LED class.


Raspberry Pi Essentials
Second Edition
Simple electronics with GPIO Zero
Take control of the real world with your Raspberry Pi
Raspberry Pi PRESS
Written by Phil King

Part 03

Manufacture a complete build

Use on-demand fabrication to create projects for you

Maker

Ben Everard

Ben is melting in the summer heat of the workshop, so is outsourcing his builds until the weather becomes more suitable.

glowingart.co.uk

One of the most challenging parts of a project like this is ensuring that everything fits together

In the last couple of articles in this series, we've looked at getting a PCB made, and getting 3D printing done commercially. In this issue, we're going to bring them together and get our project made commercially. We're going to build a weather station where the PCB and enclosure are made for us. All we have to do is solder the electronic components onto the PCB and program it.

Before we dive into the project, let's think a bit about why we might want to do this. We, as makers, have many tools and techniques available to us – including factories that can make things. What you need to do is decide what you want to do. There's a huge range of considerations that might be important to you in thinking about whether you want to use a factory or a more hands-on route.

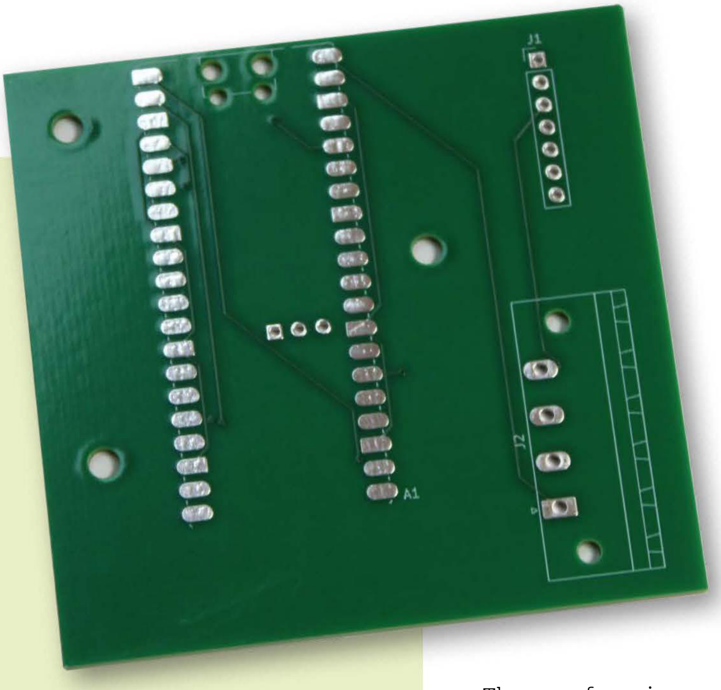
For example, here are a few reasons why you might want to outsource your project to a factory:

- If a professional final appearance is important to you
- If you lack space, time, or tools to build things
- You might get better longevity
- You enjoy the process of design more than the process of building things

However, you might want to use a more DIY route if:

- You enjoy the hands-on process of making things
- You don't enjoy doing CAD
- You like to hack things together as you go rather than finalising the design up front
- Keeping costs low is a priority

There isn't a universal right-or-wrong here; it's about finding the process that works for you.



Our PCB as it came back from the factory

Warning!

Hot solder

Soldering irons get very hot, and stay hot for a long time after they're unplugged. Make sure that you put the iron in the stand when you're not using it and don't touch the metal parts – even after it's unplugged.

rpimag.co/soldering

Feeling board

Our weather station needs a PCB to hold the electronics and an enclosure to keep it safe outside. Let's start by looking at the PCB.

The electronics we're using are a Bosch BME280 temperature, humidity and pressure sensor, and a PMS5003 particulate matter (air pollution) sensor. We'll use a Raspberry Pi Pico W to control these and upload the results to an online repository.

We could design a PCB to hold the bare BME280 sensor, but it's a tiny device and quite hard to solder. If you like a challenge, then you might want to do this, but we took the easy option and used a module that holds the BME280 and a few other components to help it work: an Adafruit BME280 module. You should be able to use one from another vendor, but the pinout might be different, so you'll need to check this and make any changes if necessary.

The PMS5003 connects via a cable that requires a particular socket that's a bit hard to source, so we've opted to cut the ribbon cable and instead put a set of screw terminals on the PCB. We don't need all the connections, just 5V, Ground, Rx, and Tx.

We won't go through the full design process, but it's a fairly simple PCB using the same method that we looked at in part 1 of this series. You can download the full design at **rpimag.co/github**.

Once the PCB has arrived, you'll need to solder everything together and wire it up. We soldered socket headers for the BME280 and then pushed it into place. The PMS5003 is wired up as follows:

There are four pins we need:

- Power
- Ground
- Rx
- Tx

These are the first, second, fifth, and fourth pins respectively on the PMS5003 (where pin 1 is closest to the centre of the device).

These connect to the PCB in the following order:

- Ground
- Tx
- Rx
- Power

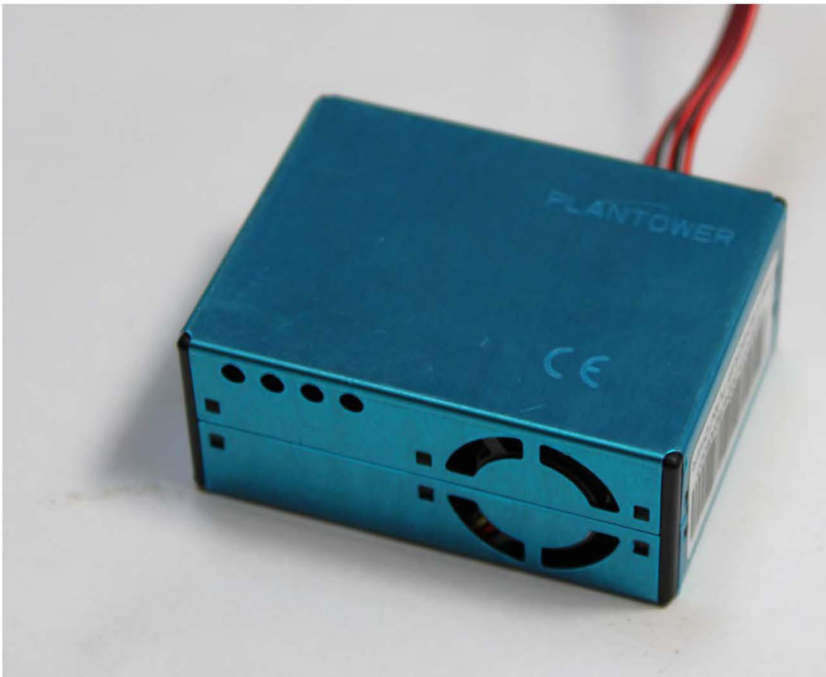
...starting with the end closest to the USB port.

Boxed in

Now that we have our PCB, we need an enclosure. We've got the full range of industrial 3D printer materials to choose from, and we selected sintered nylon. This produces parts that are significantly tougher than those made on most desktop printers, so our weather station should survive even if it takes a few bumps along the way.

Our design is a box that's open at the bottom, has a sloped lid to shed rain, and includes two screw holes for mounting.

You could design something like this in any CAD package. We used FreeCAD because it's the tool we're most familiar with. You can get the design files from **rpimag.co/github**.



▶ The PMS5003 is a low-cost particulate matter sensor that's great for DIY projects

Making it fit

One of the most challenging parts of a project like this is ensuring that everything fits together. In this case, that meant making sure the mounting holes for the bolts holding the PCB in place aligned properly, and that the enclosure was the right size to hold all the components.

FreeCAD does have the ability to import KiCad models; however, that would only work if all the parts we used in our PCB had 3D models associated with them, which ours don't.

In lieu of tools that make co-designing these parts easy, we've kept the interface as simple as possible: just three M2.5 holes. With fewer elements overall, there's less to keep track of, and less that can go wrong. That said, there are still a few traps that could derail the project if we're not careful.

We nearly sent the project off with screws that were impossible to tighten. It's all well and good making sure all the holes are aligned, but if there isn't enough clearance to fit a screwdriver in, it's all a bit pointless. Fitting components in three dimensions is significantly more complex than fitting components in two dimensions, so it's best to err on the side of caution and give yourself as much room as you can.

We had both parts made by PCBWay, which offers PCB fabrication and 3D printing services, and good maker support.

Cloud bound

Now that we've got our hardware sorted, let's move on to the software. We're using Pico W because we want to upload the sensor readings to the internet.

Of course, 'the internet' is a big place. We want somewhere to store the raw values and visualise them as charts – maybe even set alerts if a reading is particularly high (for example, if outdoor pollution is unusually bad).

There are a few options we could use for this. For example, Google Sheets has an API that allows data to be sent from a device like the Pico W — but we're going to use Adafruit IO. It's a system built for makers to store sensor readings (among other things). The free tier is perfectly capable for our purposes.

You'll need an Adafruit account. If you've used any of the firm's services before, you'll already have one; if not, you can create one for free now. Head to **adafruit.io** and click on either Get Started For Free or Sign in. Once you're set up and logged in, you'll see a key icon in the top right. Click on this to see the API keys which you'll need shortly.

We're going to use CircuitPython for our code because it supports everything we're trying to use. You can download the firmware from **circuitpython.org**.

Install the following libraries from the CircuitPython Library bundle.

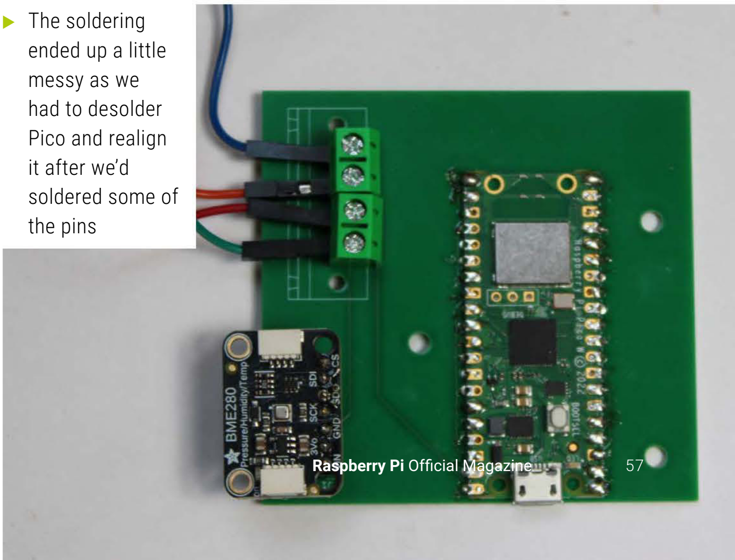
- Adafruit_bme280
- Adafruit_io
- Adafruit_minimqtt
- Adafruit_pm25
- Adafruit_connection_manager.py
- Adafruit_requests.py
- Adafruit_ticks.py

Once these are installed, you can use the following weather station code to read the data and upload everything to Adafruit IO. You'll need to update the username, SSID, and password, then save the file as **code.py** on the CircuitPython device. It should then just work automatically, uploading data every minute – if there are any errors, they'll be reported on the serial console.

```
import adafruit_connection_manager
import socketpool
import adafruit_requests
import board
import wifi
import ssl
import busio
from adafruit_io.adafruit_io import IO_HTTP,
AdafruitIO_RequestError
from adafruit_pm25.uart import PM25_UART
from adafruit_bme280 import basic
import time


#included here for brevity, but more secure to
#include them in environmental variables
aio_username = "XX"
aio_key = "XX"
ssid = "XX"
password = "XX"


#set up hardware
uart = busio.UART(board.GP0, board.GP1,
baudrate=9600)
pm25 = PM25_UART(uart, None)
i2c = busio.I2C(board.GP3, board.GP2,
frequency=100000)
bme280 = basic.Adafruit_BME280_I2C(i2c)


#connect to wifi
wifi.radio.connect(ssid, password)
pool = socketpool.SocketPool(wifi.radio)
requests = adafruit_requests.Session(pool,
           ssl.create_default_context())


#connect to Adafruit IO
io = IO_HTTP(aio_username, aio_key, requests)
print("connected to io")


try:
# get feed
    temp_feed = io.get_feed("temp")
    humid_feed = io.get_feed("humid")
    pm25_feed = io.get_feed("pm25")
except AdafruitIO_RequestError:
# if no feed exists, create one
    temp_feed = io.create_new_feed("temp")
    humid_feed = io.create_new_feed("humid")
    pm25_feed = io.create_new_feed("pm25")


while True:
    print("\nTemperature: %0.1f C" %
bme280.temperature)
    print("Humidity: %0.1f %%" %
bme280.relative_humidity)
    io.send_data(temp_feed["key"],
str(bme280.temperature))
    io.send_data(humid_feed["key"],
str(bme280.relative_humidity))


    aqdata = pm25.read()
    print("PM2.5: %0.1f" % aqdata["pm25
standard"])
    io.send_data(pm25_feed["key"],
str(aqdata["pm25 standard"]))

    time.sleep(60)
```

This code sends data to Adafruit IO, where it can be viewed by logging into the **adafruit.io** website and navigating to the appropriate feeds. From there, Adafruit IO provides tools to visualise the data using various formats such as charts and gauges.

In Adafruit IO, the script creates three separate feeds. To visualise this data with graphs, you can create a dashboard. We'll leave this as an exercise for the reader. It should be straightforward because Adafruit IO offers a user-friendly graphical interface for creating dashboards.



▶ The soldering ended up a little messy as we had to desolder Pico and realign it after we'd soldered some of the pins

The 3D-printed nylon enclosure, ready to be used

In this project, we've designed our parts and sent them to be made at a factory. This approach has become affordable for one-off projects in recent years, and if you prefer the digital side (design and programming) to the physical side of making, then it's a great option. Once your designs are set up, you just send them off and then wait for everything to be delivered to your door.

We've designed our parts and sent them to be made at a factory

New to CircuitPython?

If you haven't used CircuitPython before, here's a quick primer. It's a version of Python 3 built for microcontrollers, though it also works on computers.

Perhaps the most useful feature is the huge array of libraries for maker hardware. You can download both CircuitPython and the Library Bundle from **circuitpython.org**.

CircuitPython programs are just text files interpreted on the device itself. However, there are some minor issues with how it reads files, so we recommend using the Mu editor (**codewith.mu**).

CircuitPython has excellent documentation, and you can find more information at **rpimag.co/WhatIsCircuitPython**.

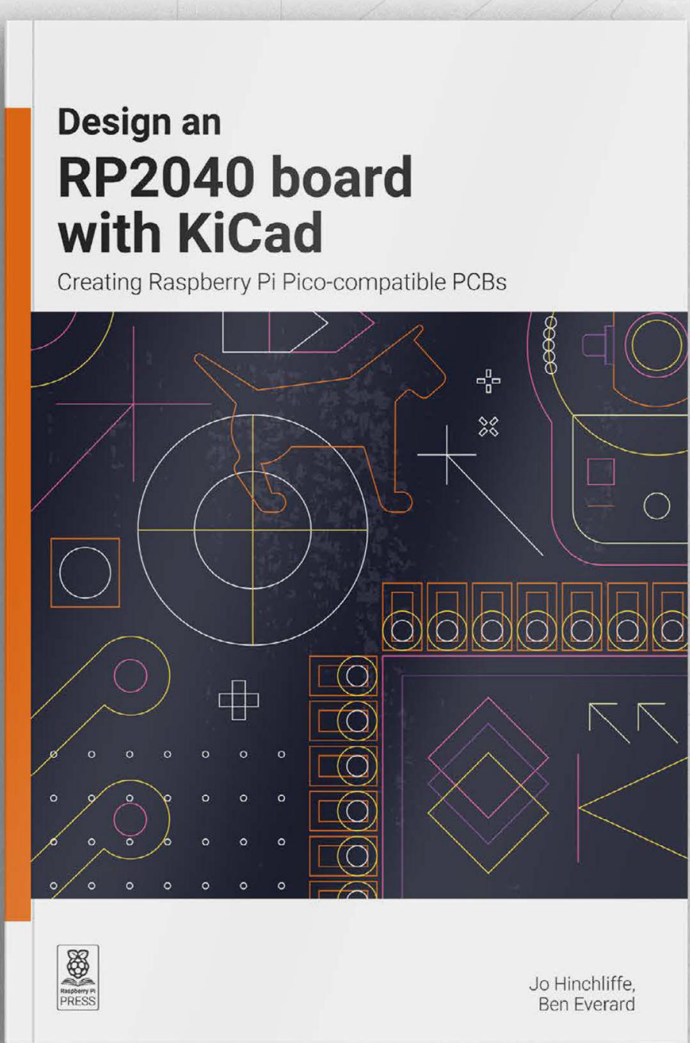
Productionising

The code we've got here is enough to get started, but it lacks a couple of features that you may want for a more robust setup.

Firstly, storing passwords and keys directly in the main code increases the risk of accidentally committing sensitive information to version control systems like Git.

Secondly, the code lacks error handling for runtime issues such as Wi-Fi disconnections. While rebooting the microcontroller can often resolve issues, implementing automatic recovery logic – such as reconnecting to Wi-Fi – would improve robustness.

It's a bit of a squeeze, but everything fits into the enclosure



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

Part 05

Retro gaming in RISC OS

What delights can RISC OS offer the keen video gamer? Let's take a look! By **Ian Osborne**

Maker

Ian Osborne

Ian's been working in tech and video games magazines for far longer than is healthy. As well as Raspberry Pi, he also writes about other computers, retro gaming, and anything else that pays.

ijosborne.bsky.social

Go on, admit it. You like computing diversions, especially video games. Sure, you're interested in the Raspberry Pi as a development tool, and you love tinkering with RISC OS on a practical level. But when you have a spare half hour to kill, your thoughts turn to gaming. Or maybe you make time for your games, and enjoy them as a hobby? And why not? Video games are awesome.

There's a wealth of video games available for RISC OS, some of which come bundled with the installation, and others that can be downloaded from the PlingStore app. So boot up RISC OS on your Raspberry Pi, and let's have some fun!

You may be surprised to find you already have a decent handful of games and other distractions on your RISC OS-powered Raspberry Pi. Boot it up and open your SD card. From the window that appears, open the Diversions folder. Here, depending on the install you used, you should see a decent collection of gaming software and app toys. Mine has 14 of them, all waiting to be played.

The first game in the Diversions collection is !Blocks. As you have probably guessed, this is a Tetris-type game, where you make lines with falling pieces. Use the **1** and **3** keys to move a falling block left and right, the **2** key to rotate it, and the **SPACE** bar to make it fall faster. Click on the game window with your middle button and you can redefine these keys if you wish.

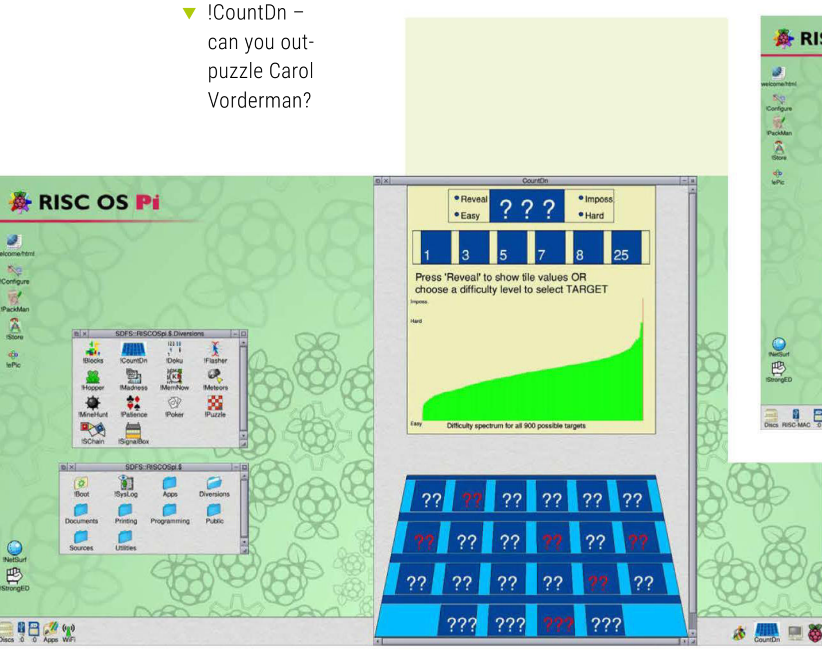
▲ You can redefine the keys in the RISC OS !Blocks game

A game of PackMan?

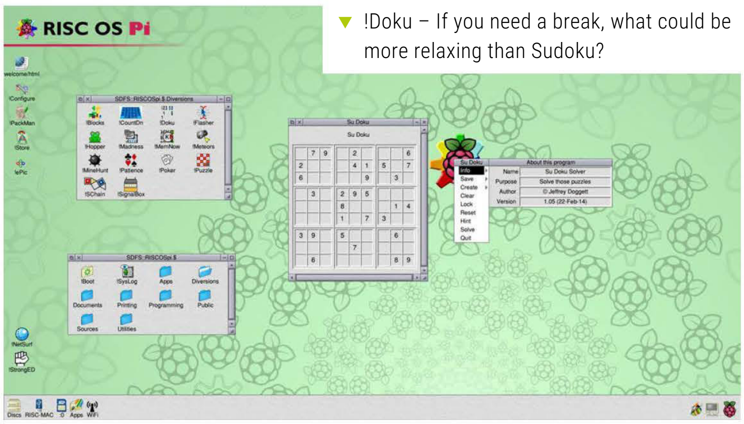
Gamers and tinkerers who are new to RISC OS might make a beeline for !PackMan, an app found in the Apps folder. Alas, they'd be disappointed. The !PackMan app isn't a RISC OS interpretation of the famous Namco arcade game Pac-Man, but a packaging system used to distribute updates to the RISC OS Pi disc image. A useful tool, but don't expect to munch power pills while dodging ghosts.

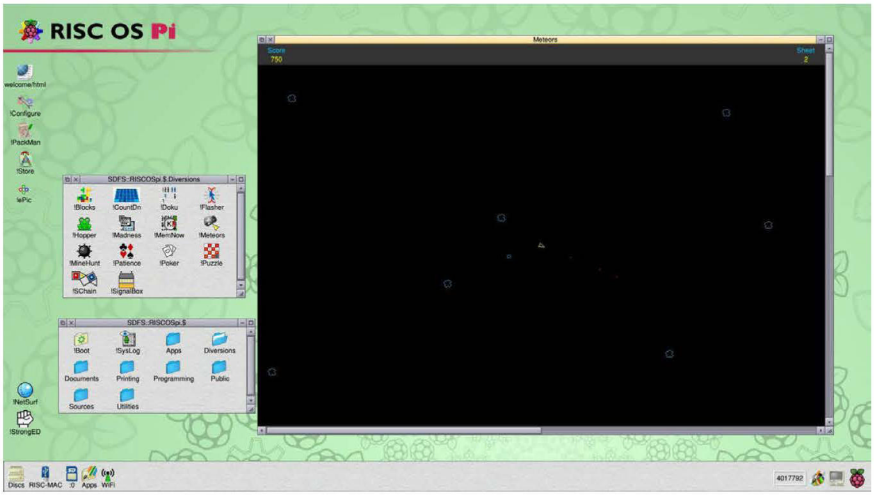
There's a wealth of video games available for RISC OS, some of which come bundled with the installation

▼ !CountDn – can you out-puzzle Carol Vorderman?



▼ !Doku – If you need a break, what could be more relaxing than Sudoku?





▲ !Meteors is a great interpretation of the Asteroids coin-op

Asteroids action

!MemNow displays the currently available RAM in the icon bar. A useful tool, but why it's in the Diversions folder and not Utilities is beyond me. Far more interesting is !Meteors. No prizes for guessing what it is – it's Asteroids action all the way, with this great RISC OS interpretation of the late 1970s Atari coin-op. Controls are simple. Rotate your ship with **Z** and **X**, accelerate with **SHIFT**, and fire with **RETURN**. The **SPACE** bar is hyperspace, and **N** brings forth a new onslaught of floating space rocks. You can also use **ESC** to toggle between full-screen and windowed play, and **P** to pause and unpause. It's a great way to kill a little time, and if you're of a certain age (like me), it will bring back a few fond memories too.

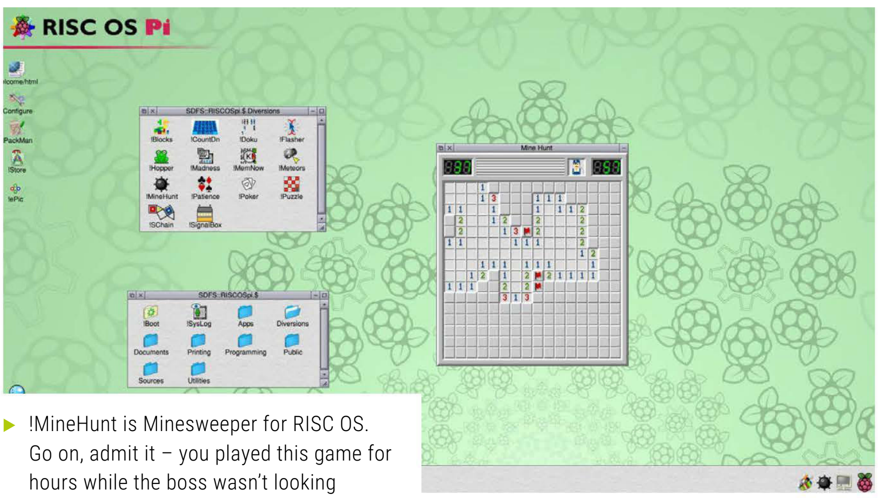
Then there's a game that should be familiar to anyone who grew up using Windows. It's called !MineHunt, and yes, it's Minesweeper – nothing to do with the Minecraft server of the same name. As usual, you're presented with a grid of squares. Left-click on one to open it. The number of adjacent squares (vertical, horizontal, or diagonal) containing a mine is shown; right-click to place a flag on a square you think contains a mine. Find all the mines and you win. Click on a mine and you lose. You can change the skill level by middle-clicking the grid, and even create a custom game.

Gaming by numbers

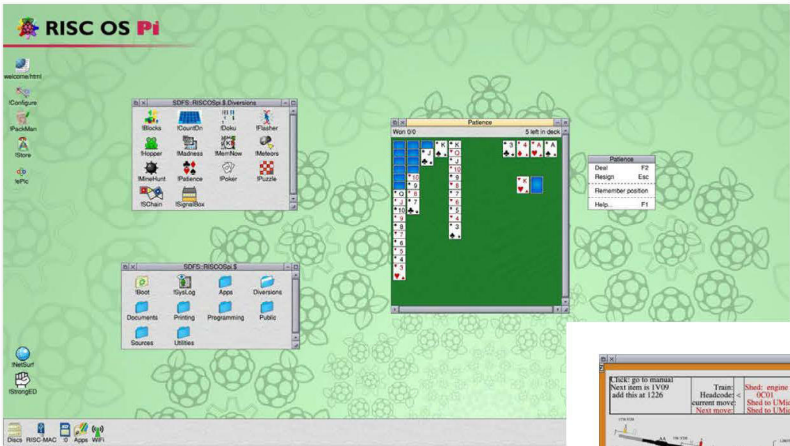
!CountDn reproduces the 'numbers' game from the popular TV show *Countdown*. You choose six tiles from those on offer, and the game then calculates a number you must get to using those on your tiles with basic maths operations (+, −, ×, ÷). Choose between Easy, Hard, or Impossible difficulty, then work it out with a pen and paper. It's not exactly feature-packed, but it's a decent way to flex your mathematical muscles while waiting for a program to compile.

Next in the Distractions window is !Doku. No, it's nothing to do with Star Wars; instead, it's a Sudoku puzzle. Open the app for an empty grid. Middle-click and choose Create, then a difficulty level, and the grid is populated by a puzzle. Click on a square and type a number to fill it in. You can save your progress, or get a hint from the middle-button menu. A quick hint from me, too: the app only allows correct entries, so if you're stuck, click on a square and tap each number in turn and see which one is allowed. Not that our readers need to resort to such tactics, right?

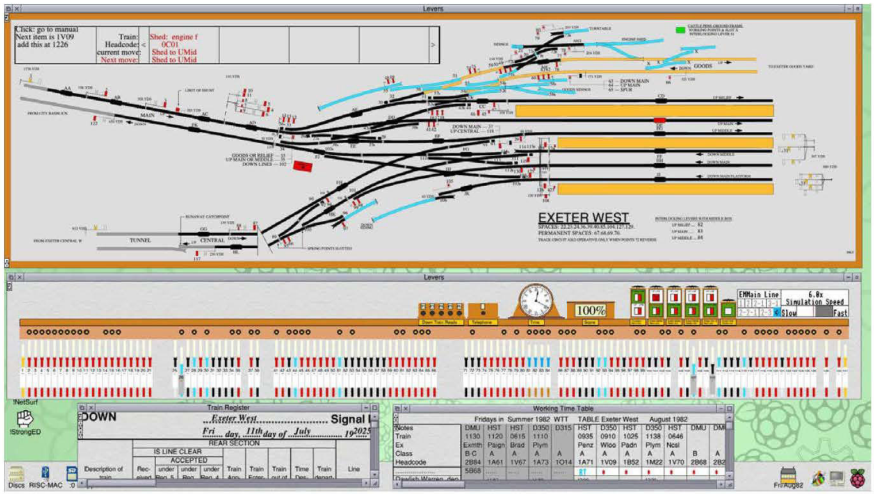
Click on the !Flasher icon in the icon bar and the caret flashes, helping you find it if you've lost it. Nothing to see here. Nothing to see with !Hopper either, a Frogger derivative that only runs under emulation. So it's on to !Madness, another distraction that's not actually a game. Click it and a Madness window opens; sort of. It's actually just the title bar, found in the bottom-left corner of the screen. The app makes every window except the Madness window move, slowly drifting across your desktop. To stop it, close the Madness window. It's great for pranking your friends and colleagues.



▶ !MineHunt is Minesweeper for RISC OS. Go on, admit it – you played this game for hours while the boss wasn't looking



◀ !Patience, another favourite time-waster in 1980s and 1990s offices



▲ !SignalBox – rather you than me

▼ !Puzzle – tackle one this size if you dare

On the cards…

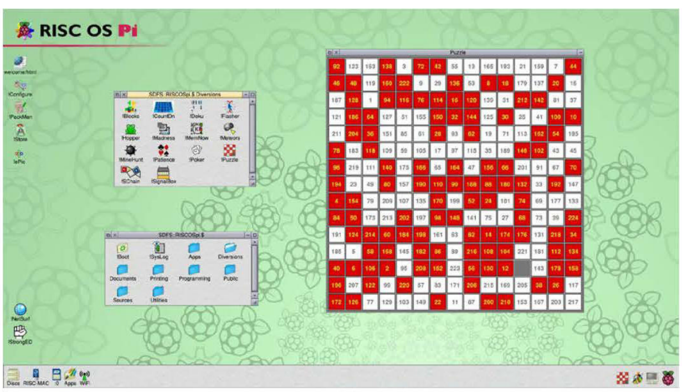
Next there's a couple of card games. !Patience is another Windows favourite played in offices all over the world to waste a few minutes before it was time to go home. And !Poker is, well, poker. I can't comment on whether it plays a good hand, as I never did get to grips with whether a full flush beats a royal house or however it goes, but it's poker.

On to !Puzzle. The aim of the game here is to get the numbers in order by sliding the tiles. It's like one of those puzzles you get in Christmas crackers, but here you can set the board to whatever size you like with the middle button.

My copy of !SChain failed to open correctly, so it's on to !SignalBox, a computerised simulation of several mechanical railway signal boxes. If you want to learn what it was like to be a signal box operator half a century or so, this app is for you, though it takes a lot of work to get to grips with it. Read about it at **rpimag.co/sigboxrisc.**

It don't mean a thing if it ain't got that Pling…

If you want more games and distractions for your RISC OS Raspberry Pi, try the PlingStore. Open the !Store app, and click on Categories. In the list that appears is an entry for Games. Click it and it's ticked. Now click on the 'A' icon in the top-left of the window for a list of all the games available on the PlingStore. You can reorder them by clicking the column headers atop the list – click Price, for example, and you can have them arranged by expense, high-to-low or low-to-high. The latter is a great way of finding everything on the store that's free.



Build a face recogniser

Use AI HAT+ and a Raspberry Pi Camera Module to build a facial recognition system. Words by **Lucy Hattersley**

Maker
David Plowman
David is an engineer at Raspberry Pi with a special interest in camera software and algorithms, and image processing hardware.

raspberrypi.com

Raspberry Pi AI HAT+ is a powerful image recognition tool capable of producing up to 26 TOPS (trillion operations per second). In this tutorial, you'll build a real-time face recognition app for Raspberry Pi.

When you run the application, a live camera preview runs inside a PyQt5 window on the desktop; you type a name, press **ENTER**, and the app collects a few frames to learn that face. After that, it draws a box around any detected face and, if there's a match, prints the person's name on the video. It also shows a thumbnail to help you see what the recogniser is using.

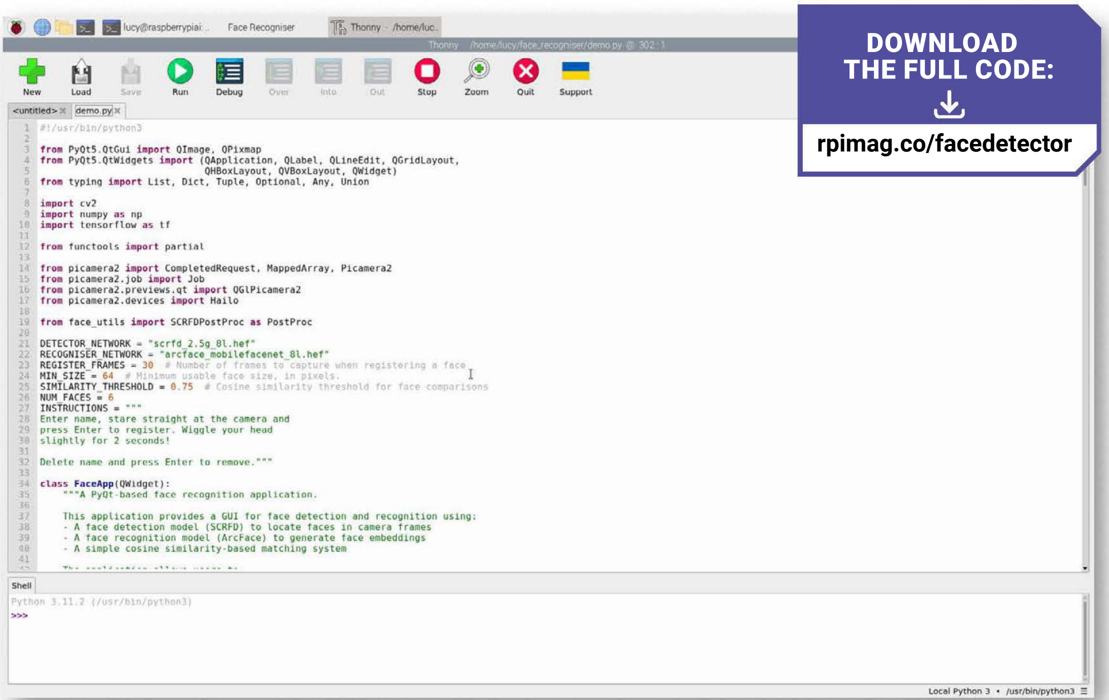
This program, written by David Plowman – an engineer at Raspberry Pi – is a great introduction to the power of the AI HAT+ hardware. The Face Recogniser utilises two different Hailo models: one to detect faces, and then another more powerful model to analyse the face against registered identities.

Setup

You will need a Raspberry Pi 5 equipped with an AI HAT+ (**rpimag.co/aihatplus**), along with a Raspberry Pi Camera Module 3. Note that this project will work with a Raspberry Camera Module 2 and does not require a Raspberry Pi AI Camera (it uses the Hailo models used in AI HAT+, not the IMX500 ones used by Raspberry Pi AI Camera.

Start by cloning the repository from GitHub. This will download the **demo.py** code that we will run, and be looking at, along with the two Hailo models.

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



DOWNLOAD THE FULL CODE:

rpimag.co/facedetector

Warning!

Biometric data

This code is for demonstration purposes only. Under GDPR requirements in the UK and EU, biometric data is a special category and processing requires a lawful basis. You should gain explicit consent when working with biometric data. Rules may vary in different locations.

rpimag.co/biometric

▲ The **demo.py** code in Thonny IDE

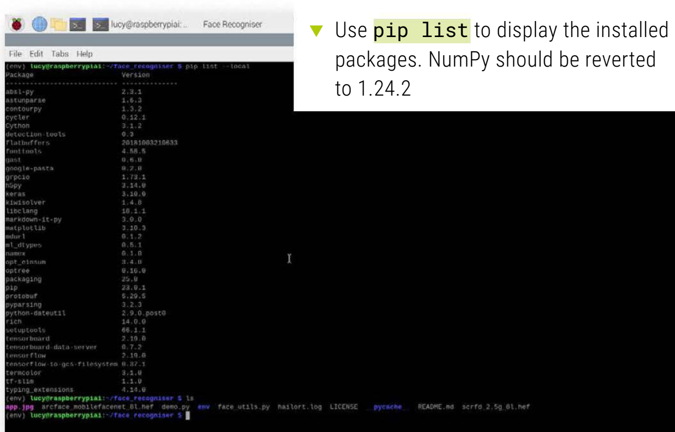
Navigate into the source directory and take a look at the files:

```
$ cd face_recogniser
$ ls -l
```

Here you will see the following files:

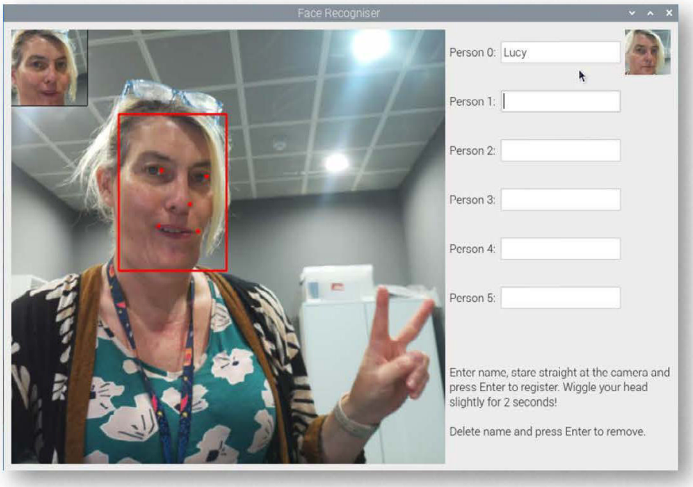
- **app.jpg.** An example image of facial detection.
- **arcface_mobilefacenet_8l.hef.** The ArcFace model is used to perform detailed facial recognition.
- **demo.py.** The main program code used to create the GUI and run the two models.
- **face_utils.py.** A support file that provides an SCRFD method for turning the raw images detected into usable results.
- **LICENSE.** Information about using the code.
- **README.md.** Instructions on how to run the model.
- **scrfd_2.5g_8l.hef.** The Sample and Computation Redistribution for Face Detection model provides the real-time detection of faces.

Face Recogniser utilises two different Hailo models: one to detect faces, and then another more powerful model to analyse the face against registered identities



▼ Use `pip list` to display the installed packages. NumPy should be reverted to 1.24.2

▼ Wiggle your head horizontally when performing the initial detection



Before you can run the **demo.py** file and start the app, you will need to set up a virtual environment and install TensorFlow and detection tools. Make sure to include the `--system-site-packages` option so that you include libcamera and Picamera2.

```
$ python -m venv env --system-site-packages
```

Before you work on a project, run the following command from the root of the project to start using the virtual environment:

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

You should then see a prompt similar to the following: `(env) $`. Now install the required Python packages:

```
$ pip install tensorflow
$ pip install detection_tools
$ pip uninstall numpy
```

The NumPy uninstall is necessary because TensorFlow and detection_tools will install NumPy 2.x, which doesn't work with Bookworm apt packages. When you uninstall NumPy, it will revert to NumPy 1.24.2. You can check this with `pip list`.

Run demo.py

With everything installed, we can run the code:

```
$ python demo.py
```

The Face Recogniser window will appear, displaying a preview from the camera. Enter a name in the Person 0 field and press **ENTER**. Look straight at the camera and wiggle your face left and right. A snapshot of your face will appear next to Person 0 and when you look at the camera, it will display the name you entered in the top-right of the preview window.

Add another name in the Person 1 field and get somebody else to look at the camera when you press **ENTER**. Now, when you and the other person look at the camera, it will detect – and name – the person who is looking. You can add up to six different people in the demonstration.

Close the preview window by clicking the X icon when you are finished

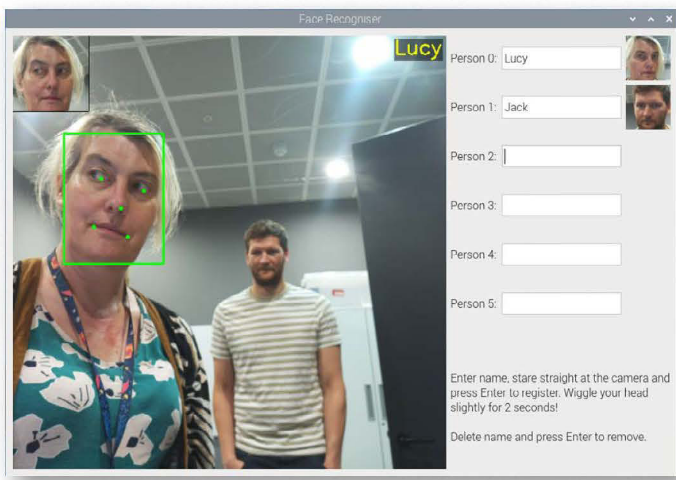
Investigate the code

Open the **demo.py** code in Thonny to investigate it:

```
$ thonny demo.py
```

The program starts with imports for the various functions. These convert the NumPy arrays captured by the camera to Qt (pronounced 'cute') and provide structure for the GUI elements. It also imports the Picamera2 stack to provide support for the camera, including the Hailo module.

▼ The person closest to the camera should be the one detected by the `DETECTOR_NETWORK`



The Face Recogniser window will appear displaying a preview

Next, we come to the constants used by the application. Following the PEP-8 style guide (**rpimag.co/pep8**), these are written in upper-case:

```
 DETECTOR_NETWORK = "scrfd_2.5g_8l.hef"
 RECOGNISER_NETWORK =
"arcface_mobilefacenet_8l.hef"
 REGISTER_FRAMES = 30  # Number of frames to
capture when registering a face
 MIN_SIZE = 64  # Minimum usable face size, in
pixels
 SIMILARITY_THRESHOLD = 0.75  # Cosine
similarity threshold for face comparisons
 NUM_FACES = 6
```

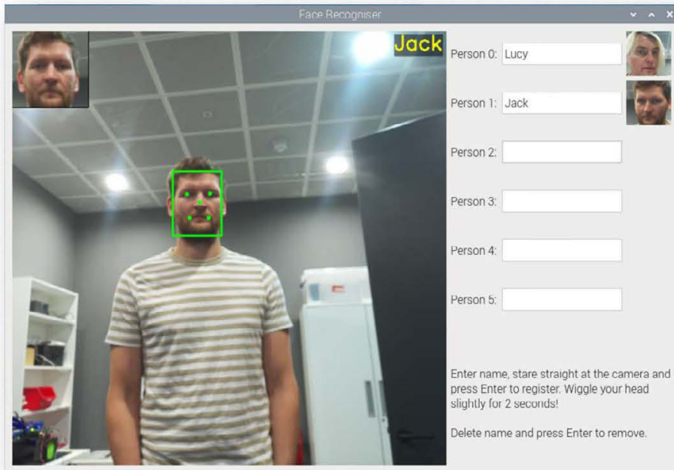
The first two lines import our two models. The other constants affect the detecting variables. You can experiment with `REGISTER_FRAMES`, `MIN_SIZE`, and `SIMILARITY_THRESHOLD`. The `NUM_FACES` constant adjusts how many faces can be stored by the program, and the GUI will adjust accordingly.

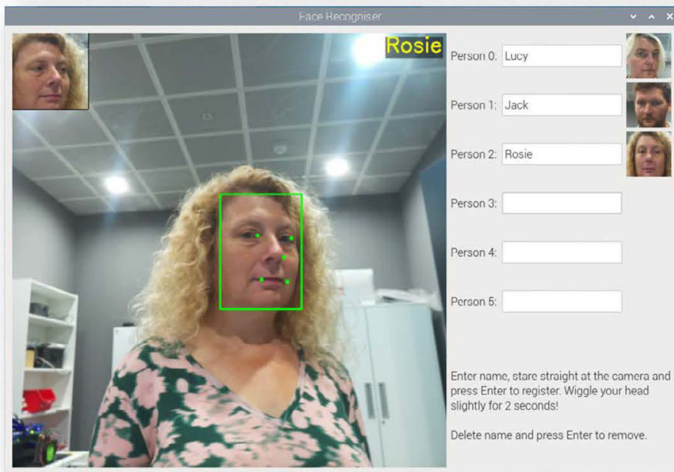
The following methods from the `FaceApp` class set up the application:

- `__init__`. This loads the two models, configures the camera, and creates the GUI.
- `enter_pressed`. This handles the **ENTER** key press in the text box.
- `draw_callback`. This draws the annotations on the camera preview.
- `capture_done`. This processes the captured camera frames.
- `create_crop`. Creates an aligned and cropped face image.

Thanks to the work of David Plowman, the code is well documented with comments explaining how each part works. We hope you enjoy this powerful demonstration of how to create a facial recognition system with Raspberry Pi. This project does not store video and everything is forgotten when it is closed. Even so, please read the warning at the front regarding GDPR and biometric data if you wish to work further with this kind of project.

▼ The `RECOGNISER_NETWORK` performs the task of identifying the detected faces



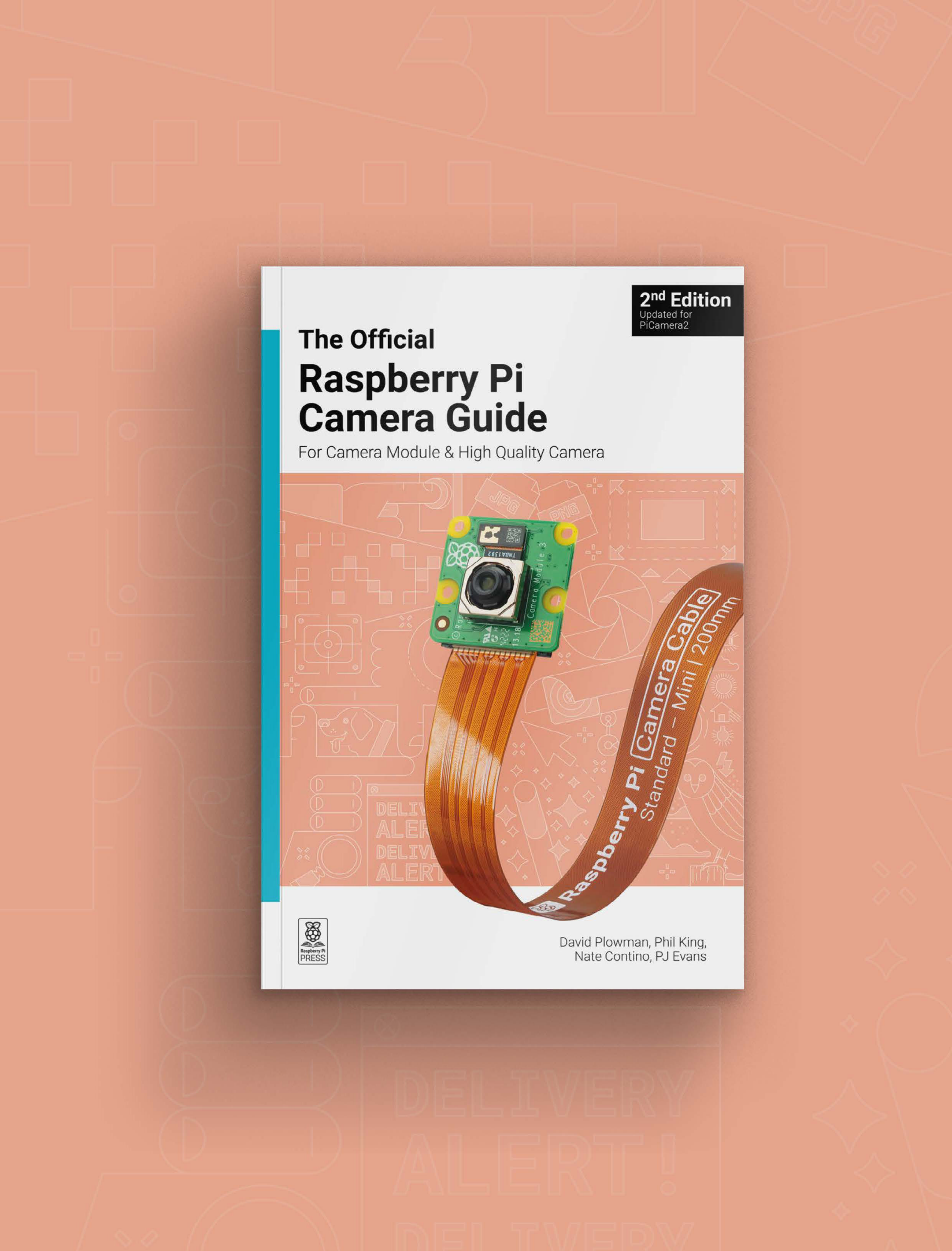


▲ Multiple people can be identified

Why ArcFace?

ArcFace is also known as 'Additive Angular Margin Loss'. This process introduces an angular margin (hence 'arc') to the traditional softmax loss function, which provides greater separation between different face identities.

insightface.ai/arcface


2nd Edition
Updated for PiCamera2
The Official
Raspberry Pi Camera Guide
For Camera Module & High Quality Camera
Raspberry Pi Camera Cable
Standard – Mini | 200mm
Raspberry Pi PRESS
David Plowman, Phil King,
Nate Contino, PJ Evans

Unusual tools: corner-cutting shear

The tool that makes cutting corners at work an acceptable practice

Maker

Dr Andrew Lewis

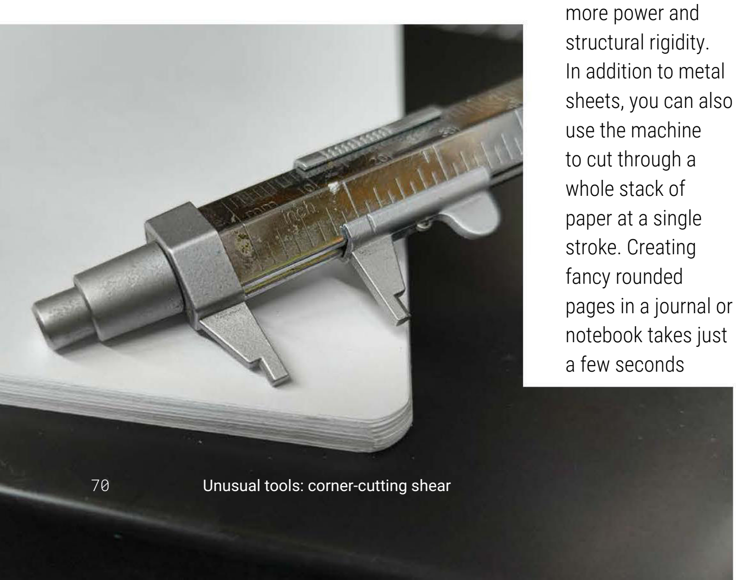
Andrew is a specialist maker and fabricator, and is the owner of the Andrew Lewis Workshop.

lewiswork.etsy.com

Warning!

Sharp blades

As well as plastic and paper, a corner-cutting shear will carve a neat radius into human flesh if you use it without due care. Watch your fingers!



▼ Unlike the flimsy hobby-store version of this tool, the industrial version has quite a lot more power and structural rigidity. In addition to metal sheets, you can also use the machine to cut through a whole stack of paper at a single stroke. Creating fancy rounded pages in a journal or notebook takes just a few seconds

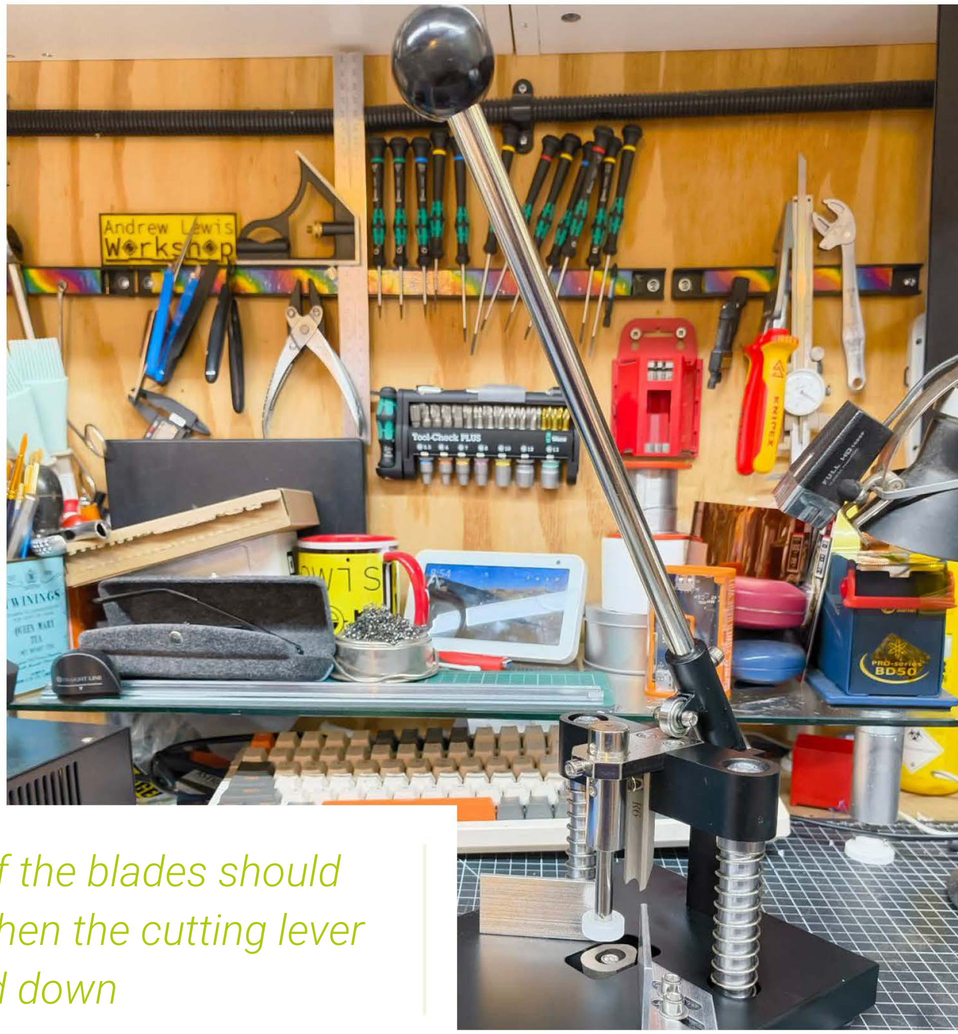
Cutting a radius onto a sheet of metal, paper, or card is surprisingly difficult to do well. You mark the corners as accurately as you can with a scribe, cut with a sharp tool, or even grind away slowly with an abrasive sheet. When you're finished, the corner still looks slightly crooked, and the more corners you cut, the worse the effect seems to get. For professional results, you need a machine that cuts the perfect corner every time.

The aptly, though not particularly imaginatively named, industrial corner-cutting machine (or corner radius cutter) lets you make repeated cuts in paper and thin metal sheet. While it's not as powerful as a hydraulic press with a corner-cutting die fitted, it can still do a surprising amount of work for a small machine. The machine is essentially a modified arbor press and jig, with a moving shaped blade in the top and a matching fixed blade surface at the bottom. Pulling the lever on the machine lowers the top blade down so that the moving blade cuts into the sheet and removes the corner. The offcut drops out of the bottom of the machine, leaving your piece of material neatly radiused.

Follow the curve

Typically, a corner-cutting machine will come with more than one blade so that you can cut different-sized corner radii. Changing the blade isn't complicated, but it can be a little bit fiddly to get right if you're not used to working with this type of machine. There are three important parts of the machine to consider when you are changing the blades: the upper (moving) blade, the lower blade (sometimes called a button or anvil), and the jig. The upper and lower blades are a matched pair, so if you're going to fit a 10mm radius upper blade, you'll also be fitting a 10mm radius lower blade. The cutting edges of the blades should match up

- ▶ While many people have seen small plastic cornering tools in craft stores, they might not realise that an industrial version is available and it can cut more than just a few sheets
- ▼ Some corner-cutting machines have a fixed position for the blades, while others are adjustable – it just depends on the brand



The cutting edges of the blades should match up perfectly when the cutting lever is pulled down

perfectly when the cutting lever is pulled down. Fitting the upper blade is usually just a matter of removing a machine bolt and swapping the blade over. The lower blades sometimes perform multiple duties, with a single button being radiused to match two different blades. You'll have to make sure these are the right way around and positioned correctly by fitting the lower blade in place loosely, slowly bringing the upper blade down, and then adjusting the fit to match the curve of the upper blade. Once the upper and lower blades are properly aligned, tighten the bolts and double-check nothing has moved.

With the upper and lower blades aligned, you need to adjust the fit of the jig. The purpose of the jig is to hold the piece that you want to cut in the right position as the blade comes down. This position will change whenever you change the radius of the cutter. Normally, for right-angled pieces, you'll want the angle between the two sides of the jig to be 90 degrees, with a 45-degree angle to the direction of the blade. That means the point of the corner should hit the midline of the blade. You'll also need to make sure that the jig is set so that the edge of the blade contacts at the edge of the material. The easiest way to do this is to use a square to set a 45-degree angle on your jig, and then set one side by eye so that the point of the material touches the midpoint of the blade. Slide the material forward and nibble away at the corner using the blade, until the blade touches the outer edge of the material. Then without moving the material, lock both sides of the jig into place flush with the edges of the material. It can be a bit fiddly and some fine adjustment might be needed, but the end results are well worth the effort.

QUICK TIP

Loosening the jig bolt slightly and tapping it with a plastic hammer is usually all that is needed to make fine adjustments.

Part 03

Conquer the command line: Permission to install

We look at the efficient system for installing and updating software in Debian systems

Maker

Richard Smedley

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

about.me/RichardSmedley

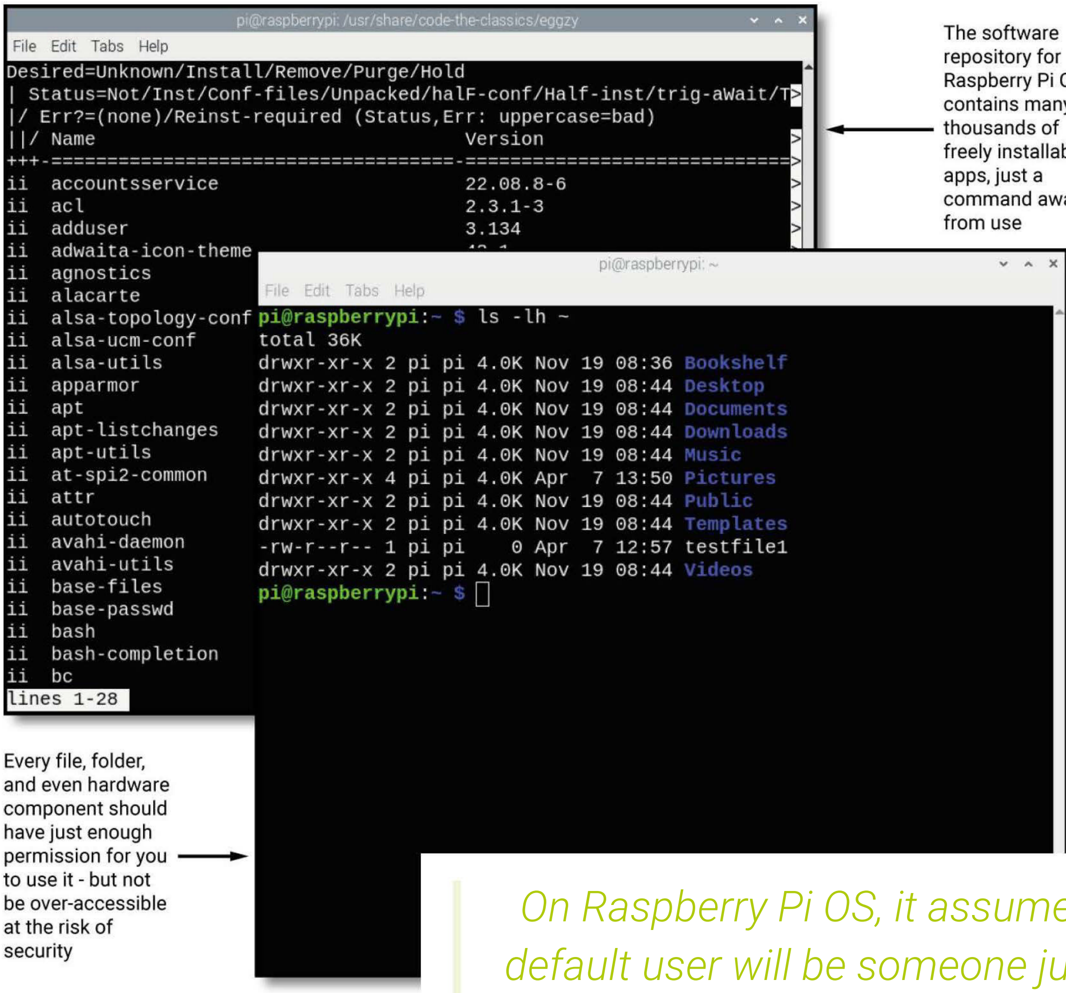
Installing software should be easy, but behind every piece of software is an evolving set of dependencies that also need installing and updating. Keeping software separate from its dependencies reduces unnecessary bloat and duplication, but adds the potential for bugs, missing files, and even totally unresolvable clashes.

Fortunately, Debian GNU/Linux cracked the problem back in the 1990s with the Debian Package Management system and the Advanced Package Tool (APT) – and Debian-based systems, like Ubuntu and Raspberry Pi OS, inherit all of the benefits. Here we'll show you the basics you need to know to install new software and keep your system up to date from the command line, and then look at the not entirely unrelated field of file ownership and permissions.

Using the `apt` command to update your system's list of installable software should be as simple as issuing the command like so: `apt update`. Try this logged in as user pi, though, and you'll just get error messages. The reason for this is that changing system software on a Unix-like system is a task restricted to those with administrative permissions: the godlike superuser, or admin, also known as root.

Pseudo root, su do

We'll get onto permissions properly a bit later, but for now you'll be pleased to know that you can fake it, using the `sudo` command. `sudo` potentially offers a fine-grained choice of permissions



The software repository for Raspberry Pi OS contains many thousands of freely installable apps, just a command away from use

Every file, folder, and even hardware component should have just enough permission for you to use it - but not be over-accessible at the risk of security

The command line gives you power without peril

On Raspberry Pi OS, it assumes that the default user will be someone just wanting to get on with things

for users and groups to access portions of the admin user's powers. However, on Raspberry Pi OS it assumes, quite rightly, that the default user will be someone just wanting to get on with things, and `sudo` in front of a command will pretty much let you do anything (without prompting you for a password, which would be the norm on any other Unix-like system). You have been warned!

The following commands will update the installed software on Raspberry Pi OS or any apt-based system like Debian or Ubuntu:

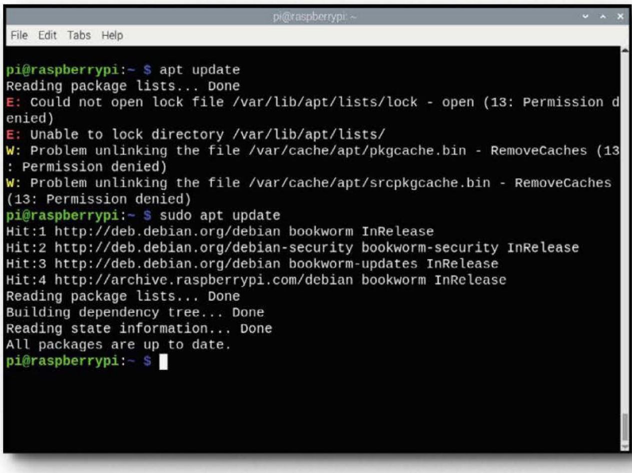
```
$ sudo apt update
$ sudo apt upgrade
```

You can wait for one to finish, check everything is OK, then issue the other command, or you can save waiting and enter both together with:

```
$ sudo apt update && sudo apt upgrade
```

SHARED RESPONSIBILITY

Share your Raspberry Pi? Read up on `sudo` and the `visudo` command to find how to give limited but useful admin privileges to other users.



▲ Raspberry Pi OS updates its listing of thousands of available apps, providing you give it admin permissions

The `&&` is a Boolean (logical) AND, so if the first command doesn't run properly, the second one will not run at all. This is because for a logical AND to be true, both of its conditions must be true.

It's always worth running the `update` command before installing new software, too – minor updates are made even in stable distributions such as Raspberry Pi, to address any issues. We've just run an update, so no need to repeat that for now. Sticking with a command-line theme, we're going to install an old suite of terminal games:

```
$ sudo apt install bsdgames
```

When prompted, type **Y** and press **ENTER**.

Searchable list

It is possible to find particular apps with `apt search` (for example, `apt search games`). You can also examine individual packages with `apt show` (for example, `apt show bsdgames`).

APT is actually a front end to the lower-level dpkg, which you can call to see what you have installed on the system: `dpkg -l`. Even on a fresh system, that's a large listing: we'll show you how to get useful information from such listings another time.

You can see the list of games you installed earlier with this command (scroll back to the top of the output and look for files in **/usr/games**):

```
$ dpkg -L bsdgames
```

Downloaded packages hang around in **/var/cache/apt** and if you find yourself short on disk space, issuing `sudo apt clean` will clear out the archive, without affecting the installed software.

Now, remember the extra details that `ls -lh` showed us before? Try `ls -lh /etc/apt`. That `-rw-r--r--` at the beginning of the listing for **sources.list** comprises file attributes, telling you who may access the file and how (more on that in a moment). Other entries in the listing have a `d` at the beginning, indicating they are directories. You'll also see hardware devices have a `c` here, for character device – `ls -l` on **/dev/input**, for example. On GNU/Linux, everything is a file, even your mouse! A dash (`-`) at the start tells us this is just a regular file; it's the remaining nine characters that cover permissions.

Every file has an owner and a group membership. Files in your home directory belong to you. If you're logged in as user pi and `ls ~ -l`, you'll see `pi pi` in each listing, telling you the owner and the group. Note that we put the switch at the end this time: that's a bad habit under certain circumstances, but we're just showing you what is possible. Owner and group aren't always the same, as `ls -l /dev` will show you.

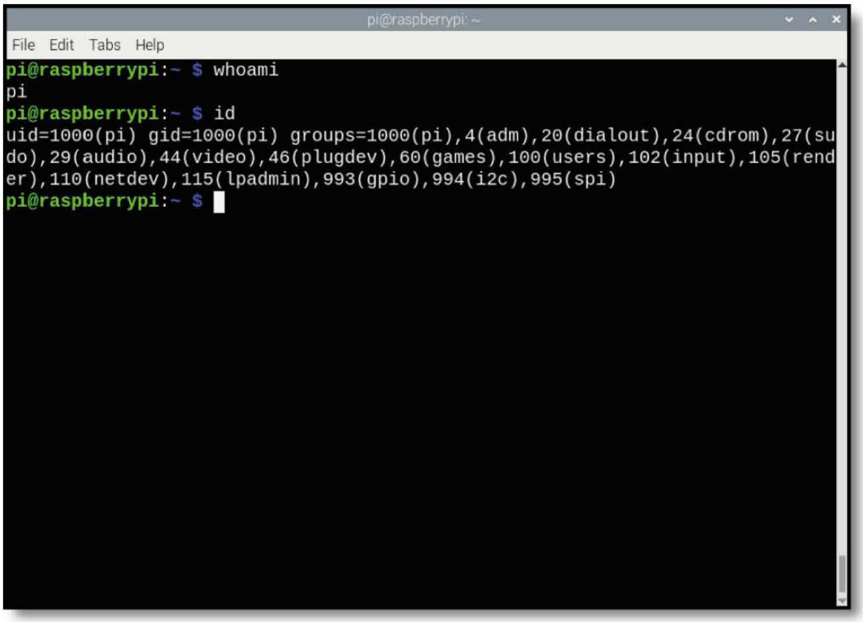
SUDO WITH CAUTION

Fine-grained permissions make for greater security, but can trip you up. If you run a command and it gives you an error, you may be tempted to re-run it with `sudo` in the hope that your superuser powers will solve it. But before you do that, take a breath, research the error message you received, and proceed with caution. These restrictive permissions are there for a reason!

File attributes and permissions

The file attributes, after the file type, are three groups of three characters (`rwx`) telling you which users may read, write, or execute the file or directory for, respectively, the user who owns the file, the group owner, and everyone else ('others'). Execute permissions are needed to run a file if it's a program, and for directories, so that you may `cd` into them. In the case of **/etc/apt/sources.list**:

- The first `rw-` tells you that the owner (root) can read or write it, but the `-` in the third location means it doesn't have the execute permission set (if it did, it would be `rwx`). Because this is neither a script nor a program, but a data file, the execute permission is not needed.
- The following `r--` tells you that members of the group that owns the file (also named root) have only read access to the file. But the owner's permissions take precedence, so the root user (which includes you, acting under the power of `sudo`) will always have access.
- The final `r--` is for all other users, who have only read access.



FREE TO USE

Software in the Raspberry Pi OS repository is not just free to use, but freely modifiable and redistributable. Free software, like Raspberry Pi OS's Debian base, is built on sharing: for education and for building community.

`cd` into **/usr/lib/firefox**, then enter the command `sudo chmod a-x launcher.sh` – the `a` stands for all (user, group, and others), use `u`, `g`, or `o` to just change one. In this case, the `-` means to take away a permission. After you do this, try opening Firefox from the main menu and it won't work. You can return things to normal with `sudo chmod a+x launcher.sh`.

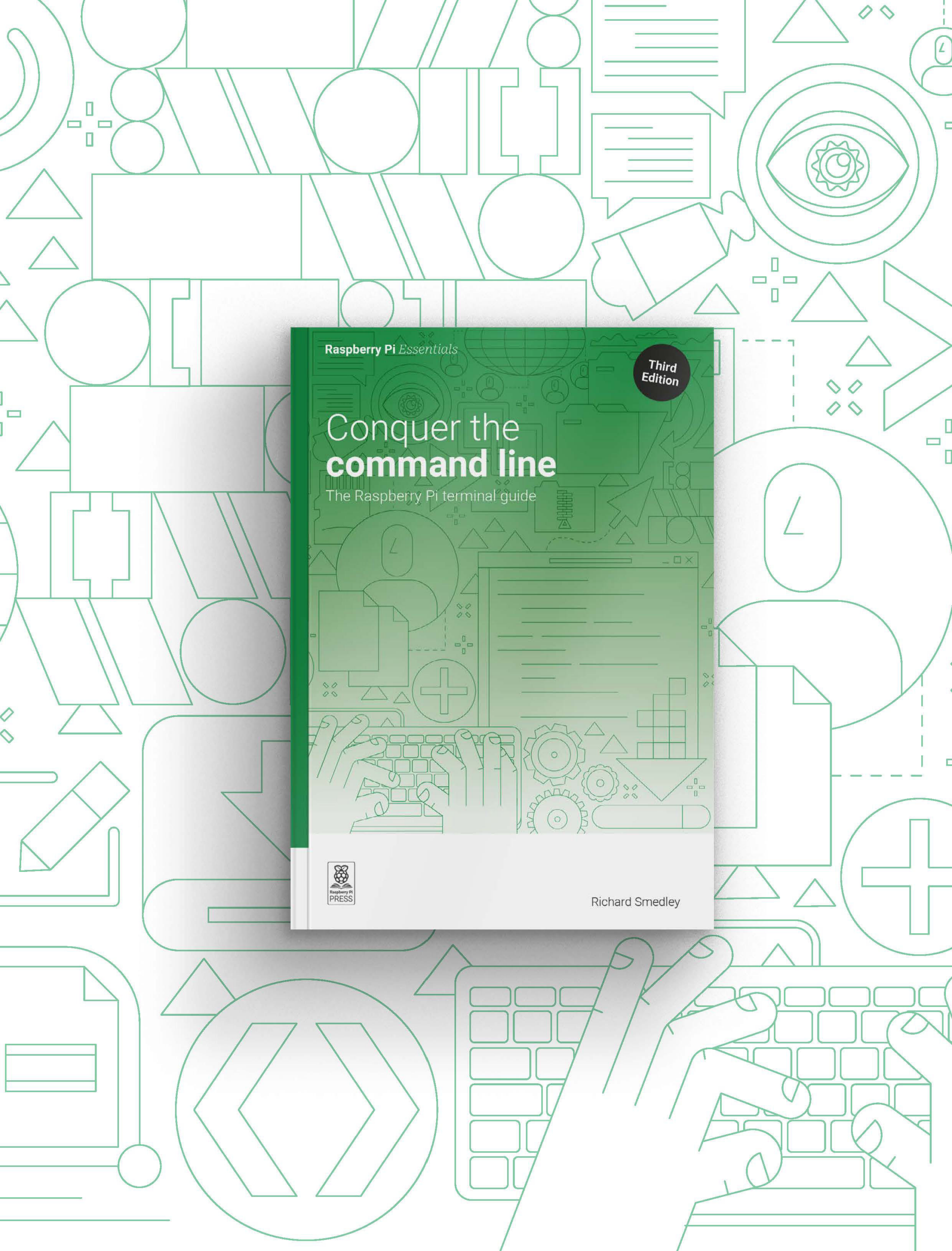
Octal version

As an alternative to using the `a+x` notation, we could have typed `sudo chmod 755 launcher.sh`. Those numbers are an octal representation of user, group, and others' permissions: in each case, read is represented by 4, write by 2, and execute by 1, all added together. So here we have 7 for read+write+execute for user, and 5 for read+execute for group and all other users. `ls -l` and you'll see we're back to `-rwxr-xr-x`.

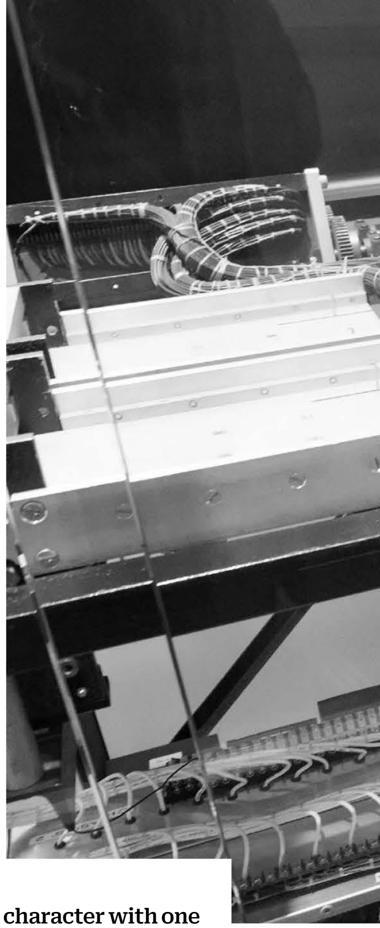
You can use `chown` to change who owns a file and `chgrp` to change which group it belongs to. Make a new text file and then run `sudo chown root myfile.txt` — now try editing it and you'll find that while you can read the file, you can no longer write to it. You can also make a file that you can write to and run, but not read!

In the next issue, we'll be doing useful things with the output of our commands; before moving on, though, why not try some of the games from the bsdgames package we installed? If you'd like to try an old-school role-playing game, try running hack, but make sure you have a few hours free first! If you enjoy interactive fiction, bsdgames-adventure is an implementation of the classic Colossal Cave Adventure.

▲ The `id` command shows what group access you have, for permission to use and alter files and devices


Raspberry Pi Essentials
Third Edition
Conquer the command line
The Raspberry Pi terminal guide
Raspberry Pi PRESS
Richard Smedley

ABC (Atanasoff-Berry Computer)



As difficult as ABC: designing the first electronic digital computer. By **Tim Danton**

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.

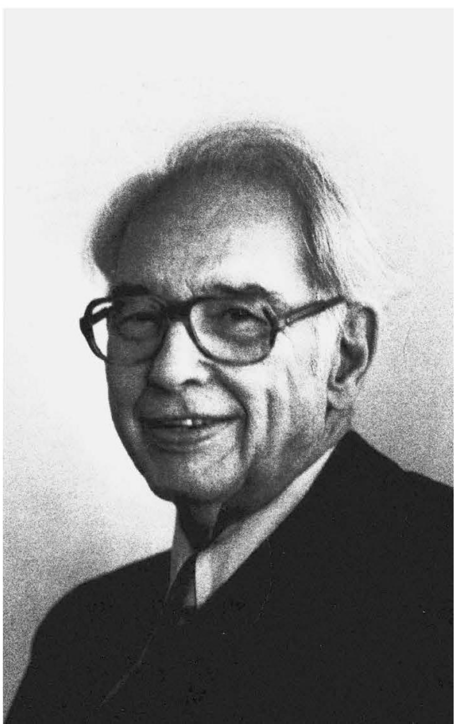
dantonmedia.com

John Atanasoff reveals much of his character with one simple anecdote in his autobiographical essay, 'The Beginning'.[1] It all started in 1913, he explains, when his father bought a new slide rule that he ended up giving to his son. "In two weeks or thereabouts, I could solve most problems with it. Could you imagine how a boy of nine-plus, with baseball on his mind, could be transformed by this knowledge?"

We think it's safe to say that an ordinary child of nine would not be transformed. But Atanasoff was no ordinary child, and he would grow up to be quite an extraordinary adult.

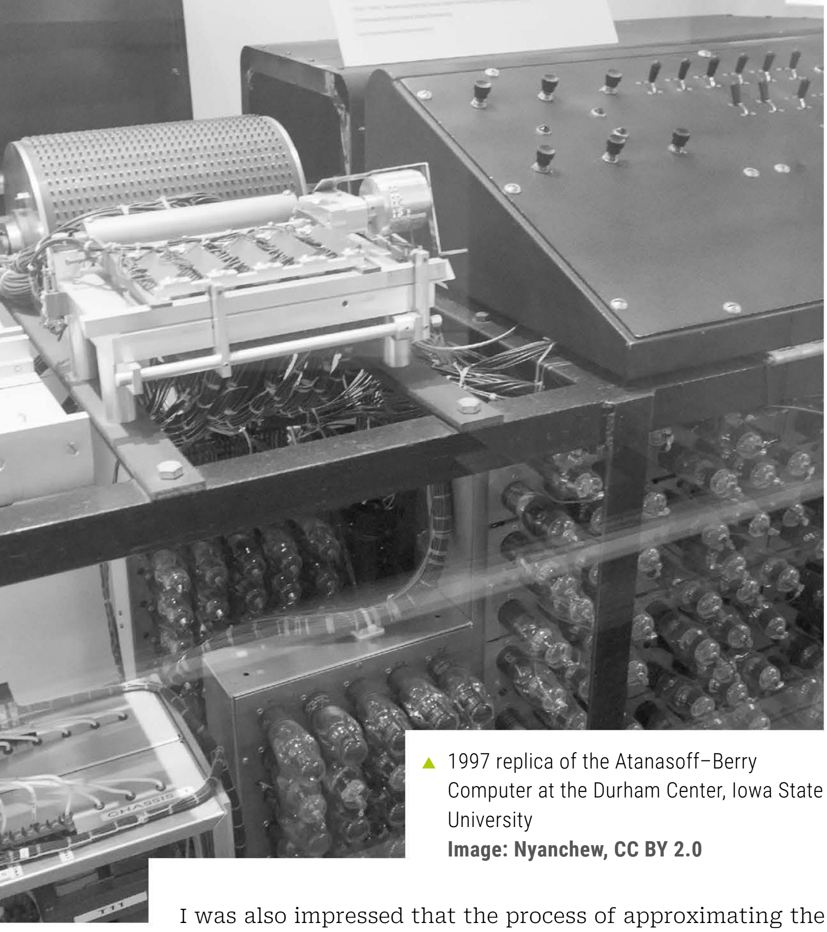
The journey from slide rule to the world's first electronic computer took a shade under three decades, with stops along the way to study electrical engineering at the University of Florida and earn a master's degree in mathematics at Iowa State College (now Iowa State University).

It was whilst working on his PhD thesis, at the University of Wisconsin, that Atanasoff first grew frustrated with existing methods of calculation. "This work demanded long calculations with a table computing machine/tabulator to solve the equation of Shreudinger [Schrödinger]," he wrote. "Such calculations required many weeks of hard work on a desk calculator such as the Monroe, which was all that was available at the time.



▶ John Atanasoff **Image: Eye Steel Film, CC BY 2.0**

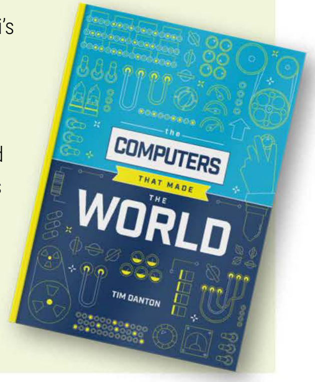
1 'The Beginning, John Atanasoff', in Dimitar Shishkov (ed.), *John Atanasoff: The Father of the Computer* (TANGRA TanNakRa, 2001, ISBN 978-9549942248), pp69-101



▲ 1997 replica of the Atanasoff–Berry Computer at the Durham Center, Iowa State University
Image: Nyanchew, CC BY 2.0

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 12 influential computers built between 1939 and 1950. You can pick up a copy on the Raspberry Pi Press store.



rpimag.co/computersworldbook

I was also impressed that the process of approximating the solution of partial differential equations required a great many calculations, a fact that ultimately motivated my work in automatic computing."

This was in 1930. After returning to Iowa State College, where he would soon be made assistant professor of mathematics and physics, he and a colleague ingeniously adapted a college-owned IBM tabulator so that it could solve a set of linear equations.[2] As Atanasoff described it, the tabulator was "the largest 'computer' of the day". These were huge machines that performed operations based on data held on punch cards; a technology with roots trailing back to Jacquard's loom in the early 1800s.

Although the tabulator slashed calculation times compared to the Monroe, John Atanasoff realised that the days of such clunky, mechanical machines were coming to an end. He even wrote to IBM to share his ideas, but the company was not interested in the thoughts of an assistant professor from a provincial college. If anything, it appears to have had disdain. One internal IBM memo, which came to light many years later, stated simply and decisively: "Atanasoff should not be allowed near the tabulator".[3]

Unaware of IBM's attitude, Atanasoff continued to correspond with the company to share his ideas of how the tabulator could be improved by embracing electronic computing. He only stopped when he received a letter from IBM categorically stating that it was not interested in building electronic computers. A letter that caused him to remark at the time that he should frame it; he could see the future far more clearly than IBM's executives.

Living with a man who is constantly infatuated with strange ideas is quite exciting. It is never boring

Even if Atanasoff could have persuaded IBM to make this jump into the unknown, it seems unlikely that he would have waited. In 1985, 35 years into their marriage, his second wife Alice wrote: "Living with a man who is constantly infatuated with strange ideas is quite exciting. It is never boring. Sometimes I tell him that other people go and buy what they need but he wants to invent his own machine, to build some device."[4]

This character trait continued throughout Atanasoff's life. When he retired, he not only designed his own house for a newly acquired 200-acre farm, but also a mechanical tree planter.

2 Anyone curious as to how Atanasoff achieved this can read the eleven-page paper he wrote that detailed the process, which is digitised at **digitalcollections.lib.iastate.edu**. Search for 'Solution of Systems of Linear Equations by the Use of Punched Card Equipment'.

3 Nikolay Bonchev, 'The Route to the Invention', in Dimitar Shishkov (ed.), *John Atanasoff: The Father of the Computer*, p44

4 Alice Atanasoff, 'Living with a man who is constantly infatuated with strange ideas is quite exciting', reprinted in Dimitar Shishkov (ed.), *John Atanasoff: The Father of the Computer*, p171

During this evening in the tavern, I generated within my mind the possibility of regenerative memory. I called it 'jogging' at the time

Using this, he planted an astonishing 36,000 trees.[5]

▲ A monument to John Atanasoff in Sofia, Bulgaria
Image: Nickolay Angelow, CC BY-SA 2.5

A vision of the future

With hindsight, it seems obvious that Atanasoff's problem-solving nature, background in electrical engineering, and frustrating experiences with mechanical computing devices would ultimately lead to his invention of an electrical computer. But this oversimplifies things: one aspect of Atanasoff's genius was his ability to not only recognise the tabulators' inherent limitations, but to analyse the problem and then break it down into component parts. If anything, his biggest problem – ironically mirroring the equation sets that he was trying to solve – was the number of unknowns. Put simply, he had too many options to choose from.

"I thought I knew how a computer should work," he wrote in the mid-1980s.[6] "First it would have to add and subtract, and later one could compound these operations into multiplication and division… From the start, I was interested in carry-over;[7] it is the crux of the digital method."

In contrast to the ENIAC and Harvard Mark I, completed almost a decade later, Atanasoff embraced the use of binary rather than decimal numbers. "A digital computer also requires some entity to represent numbers," he wrote in the same article. "While historically this system used numbers to the base 10, we intend no such restriction, for in theory any greater than unity can be used as the base. My own device… and most modern computers use the base 2."

He added: "In looking over the 1936 art in computing, I had become convinced that a new computer should provide for a much larger retention of data. Almost at the start, I called this 'memory'. The word seemed natural to me, as I suppose it did to others since it is still in use in a wide field including computers."

Atanasoff had solved, in theory, a couple of fundamental problems that stood in the way of automatic computers. What he didn't have was a blueprint for a working computer itself. Now we fast-forward to a courtroom in 1971. By this time, computers were big business, and the Sperry Rand company, the owner of electronic computing patents based on the ENIAC, was demanding huge royalties. To be precise, 1.5% of the cost of any new computer, payable by every manufacturer.

Every manufacturer, that is, except IBM. In 1956, it had struck a secret deal with Sperry Rand to share their intellectual property across the newly emerging field of electronic data processing. Sperry Rand had the lion's share of the patents, including the so-called ENIAC patent, which is why IBM paid $10 million as part of the deal.

Independent observers, had there been any, may well have commented that this was a small amount to pay for such a pivotal piece of IP. Why did Sperry Rand accept such a low figure? The answer almost certainly stems from a visit to Atanasoff by an IBM patent attorney, AJ Etienne, in June 1954. According to Atanasoff's granddaughter, Etienne said: "If you will help us, we will break the Mauchly-Eckert computer patent; it was derived from you." But that was the last Atanasoff heard from the lawyer.

It was only years later, when the matter came to trial, that Atanasoff realised the discussions went no further because IBM had secretly settled with Sperry Rand. The obvious but unproven interpretation being that IBM had used its knowledge of 'prior art', particularly relating to the ABC's memory system, as a negotiation tool. Rather than test the argument in court, how much better for both parties to settle the dispute behind closed doors?

5 Tammara Burton, *World Changer: Atanasoff and the Computer* (TANGRA TanNakRa, 2006, ISBN 978-9549942941), p173

6 John Atanasoff, 'Advent of Electronic Digital Computing', in IEEE Annals of the History of Computing, Vol 6, Issue 3, July–Sept 1984, p237

7 Atanasoff is referring to the challenge of 'carrying' values across multiple digits during addition and subtraction, a fundamental aspect of performing complex calculations.

When Honeywell bought General Electric's computer division in 1970, it had no knowledge of Atanasoff's computer or the secret cross-licensing deal between two of its arch-rivals. What it did know is that Sperry Rand wanted $250 million in royalties, equivalent to $2.1 billion in 2025. A figure it later reduced to $20 million, compared to the $150 million it was asking from six other companies.[8]

Pub inspiration

Honeywell decided the time was ripe to challenge this patent. Part of its argument was that John Mauchly – one of the co-creators of the ENIAC – had based a significant amount of that computer's design on the ABC. As Atanasoff's son put it in a 2010 article:[9] "John Mauchly visited Iowa State in 1940–41 and illegally borrowed some of the ideas in the ENIAC development and patent."

Honeywell enlisted Atanasoff as its key witness in the trial. "Well, I remember that the winter of 1937 was a desperate one for me," he recalled under oath in June 1971, "because I had this problem and I had outlined my objectives [for the computer] but nothing was happening, and as the winter deepened my despair grew… we come to a day in the middle of winter when I went to the office intending to spend the evening trying to resolve some of these questions and I was in such a mental state that no resolution was possible.

"I went out to my automobile, got in and started driving over the good highways of Iowa at a high rate of speed." At that point, Iowa was a dry state, and it's unlikely to be a coincidence that he was heading into Illinois for a drop of liquid inspiration. The long drive also allowed his mind to mull over complicated problems, a method he had used before. After "several hours" he arrived at a bar in Rock Island, which he describes more colourfully as a "speakeasy" elsewhere. Some of the nature of the place is reflected in the fact that it featured a secret tunnel for escaping cop raids.

After sitting down and ordering a drink, Atanasoff said that "my thoughts turned again to computing machines… During this evening in the tavern, I generated within my mind the possibility of regenerative memory. I called it 'jogging' at the time."

Three problems down, one to go. So, with much poetic licence, let us imagine Atanasoff silently shouting "Eureka!" as the final element fell into place. "During the same evening, I gained an initial concept of what is today called the 'logic circuits'. That is, a non-ratcheting approach[10] to the interaction between two memory units, or, as I called them in those days, 'abaci'. I visualised a black box which would have the following action: suppose the state of abacus one and the state of abacus two would pass into the box; then the black box would yield the correct results."

He added: "During that evening in the Illinois roadhouse, I made four decisions for my computer project:

- I would use electricity and electronics as the media for the computer.
- In spite of custom, I would use base-2 numbers (binary) for my computer.
- I would use condensers for memory, but 'regenerate' to avoid lapse.
- I would compute by direct logical action, not by enumeration."

After a year of experimenting with jogging and logic circuits (thank goodness for those years studying electrical engineering), Atanasoff was confident enough in his idea to build it. Now all he needed was money and an assistant. A grant from the dean of the graduate school provided the first, an accidental meeting with a professor of electrical engineering his second: "I have your man," the professor replied, after he and Atanasoff bumped into one another on campus. "Clifford Berry."

A lazy biographer might argue that Berry was Watson to Atanasoff's Holmes, but that would ignore the younger man's intellectual prowess. "Atanasoff's propensity for coming up with a succession of new ideas while working on projects had overwhelmed a number of his graduate students," wrote Tammara Burton, his granddaughter, in her biography *World Changer: Atanasoff and the Computer*,[11] "but Berry had no problem keeping pace with Atanasoff. This ability earned Atanasoff's unqualified respect and admiration and provided the basis for an ideal working relationship and a lifelong friendship."

Atanasoff always heaped praise upon Berry. To the extent that when he realised that the computer needed a name for the sake of posterity, he decided upon the Atanasoff-Berry Computer. Although the beauty of the ABC initials may, just perhaps, have been another reason.

8 Alice R Burks and Arthur W Burks, *The First Electronic Computer: The Atanasoff Story* (The University of Michigan Press, 1988, ISBN 978-0472100903), p197

9 'John Atanasoff and Clifford Berry: Inventing the ABC, A Benchmark Digital Computer', in Electronic Design magazine, 12.09.10, p59

10 Unlike a ratcheting device, where motion can only happen in one direction, here Atanasoff envisaged a device where movement could happen in either direction.

11 Tammara Burton, *World Changer: Atanasoff and the Computer*, p100

The team formed, the money in place, Atanasoff and Berry set to work on the prototype in early autumn 1939. Atanasoff had a theoretical blueprint in mind, but putting theory into practice meant building and testing every component.

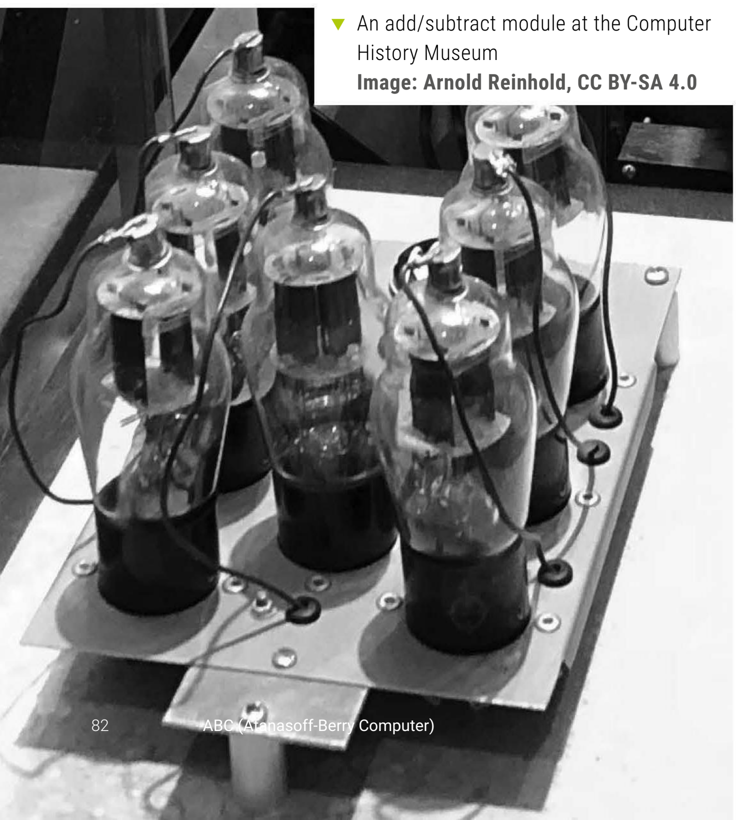
One of the most crucial was the ABC's memory. "I chose condensers (or capacitors) as the element for memory, because a condenser can give a good voltage to actuate a vacuum tube," he wrote, "and because the vacuum tube will give enough voltage to charge the condenser."

This was the concept of 'jogging' in electrical form, and a concept that we still use to this day in computers in the form of memory chips (dynamic random access memory, or DRAM). Just like the ABC's condensers, this needs a constant top-up supply of electricity to stay alive. Or, to keep jogging. "Jogging is reminiscent of the little boy going to the grocery store and reciting, 'a dozen eggs, a pound of butter etc.'," to quote Atanasoff once more. "Over and over, hoping to arrive at the store before his memory has failed."

The prototype

Atanasoff remembers the prototype being finished in November 1939, but according to materials submitted in the court case, the first demonstration happened a month earlier. Either way, Atanasoff's 'crude device' proved a success.

The prototype counts so many firsts that it seems almost unbelievable that it was created by two men, neither experts in this field, in a matter of months. It used vacuum tubes (valves) rather than electro-mechanical switches. It used binary rather than decimal. It used logic systems. It used capacitors to create memory. It took the idea of jogging from theory into practice. It was, quite simply, brilliant.



▼ An add/subtract module at the Computer History Museum
Image: Arnold Reinhold, CC BY-SA 4.0

But the prototype was also extremely limited. It could only add or subtract numbers, and not even big ones. The wheel mechanism contained two rings of 25 condensers each, meaning that it could represent numbers up to 25 base-2 places, equivalent to about eight decimal places.

Atanasoff and Berry knew they had made a breakthrough by creating this prototype, but to turn the proof-of-concept into a computer would take money. More money than they could raise directly from Iowa State College, despite it generously awarding a further $810 grant on completion of the prototype.

This was enough for Atanasoff and Berry to build the frame of the computer. It was roughly the size of an office desk, six feet (1.8m) across, and at roughly three feet (0.9m) wide it was a couple of inches wider than a typical door frame; a decision that would cause a fatal logistical problem further down the line. The two men worked fervently on the project whenever they could in the early months of 1940.

By this point, Atanasoff realised this was a significant invention that required patenting. So, over the course of weeks, he authored a paper called 'Computing Machine for the Solution of Large Systems of Linear Algebraic Equations'.[12] It was both the basis of a patent submission – a copy would eventually be sent to a patent attorney – and a begging letter for $5330. This, he estimated, would cover the cost of construction, materials, revisions, and two research assistants to help.

It is always hard to convert such sums into something meaningful today, but $5330 inflates to around $120,000 in 2025 terms. Bearing in mind the ambition of what Atanasoff wanted to achieve, that's a modest sum. As he signed off the report himself: "…it would have seemed absolutely impossible to the writer two years ago to have designed and constructed a computing machine of so large a capacity on so small a budget."

By capacity, Atanasoff literally refers to what the machine would be capable of. At its heart, this was "a computing machine which has been designed principally for the solution of linear algebraic equations," to quote Atanasoff's document, and that meant it could tackle problems that might otherwise take weeks to solve using mechanical aids. He set out nine potential fields of application, ranging from "multiple correlation" to "perturbation theories of mechanics, astronomy and the quantum theory".

He explained: "Since an expert [human] computer requires about eight hours to solve a full set of eight equations in eight

12 Reprinted in Brian Randell (ed.), *The Origins of Digital Computers: Selected Papers* (Springer-Verlag, 1973, ISBN 978-3540113195), pp305-325.

unknowns… to solve 20 equations in 20 unknowns should thus require 125 hours. But this calculation does not take into effect the increased labour due to greater chances of error in the larger systems and hence the situation is much worse than this."

This difficulty, wrote Atanasoff, was why the "solution of general systems of linear equations with a number of unknowns greater than ten is not often attempted." His machine would open up a new field of research and possibility. And all for $5330? Who could possibly say no?

Not the Research Corporation, a body founded in 1912 to support scientists and inventors to create patented products that could then be licensed. Its confidence in Atanasoff's work, and the amount of money involved, sparked a sudden interest in the computer from Iowa State College's president, Charles E Friley. After largely ignoring the project, which had been relegated to a basement within one of the Physics buildings, Friley told Atanasoff to sign an agreement that would give the college 90% of the proceeds of future income generated by the computer (and its patents). If Atanasoff didn't agree, Friley threatened, he would withhold the $5330 of funding.

> *Eckert and Mauchly did not themselves invent the automatic electronic digital computer, but instead derived that subject matter from one Dr John Vincent Atanasoff*

Atanasoff, never afraid to speak his mind, did not bow under this pressure. "I made what I thought was a small noise, but it was regarded by some as a loud one," he said later.[13] After much negotiation, he agreed to hand over the patent to the college but, in return for paying half the cost of the patent, he would benefit from half of any proceeds. It is a reflection of Atanasoff's sense of fairness, and his appreciation of Clifford Berry's work, that he agreed to pay 10% of his income to his assistant.

The ENIAC connection

All this happened in July 1941. A month earlier, Atanasoff had played host to John Mauchly, who would go on to co-create ENIAC. The pair first met at the annual meeting for the American Association for the Advancement of Science (AAAS) in late December 1940, after Atanasoff attended a lecture by Mauchly on his use of a harmonic analyser to help forecast weather (as we will discover in the story of the ENIAC, a keen interest of Mauchly's).

Atanasoff approached Mauchly at the end of the lecture and explained his work on the recently finished prototype. They started writing to one another, and it wasn't long before Mauchly asked if he could see the computer in person.

On that fateful visit – Mauchly arrived on the evening of Friday 13 June, leaving early on Wednesday 18 June – the three men (including Berry) would spend hours dissecting the workings of the machine. Exactly what happened would be much discussed and much disagreed about in court three decades later, but from all we know about Atanasoff's personality, and his desire to share information, it would only be natural for him to go into detail with this most eager of students.

His first wife, Lura, felt her husband was being far too open. "Several times during the visit, Lura warned her husband that he was 'talking too much' to this man about whom she had serious misgivings," wrote Atanasoff's granddaughter.[14] "She understood the necessity of protecting intellectual property and sensed in Mauchly a sinister motive."

According to Atanasoff, there was only one request that he denied his guest: to take away a copy of the 35-page report that had been sent off to the Research Corporation. At this point, he was nearing the end of his negotiations with Iowa State College about the patent application, and was wary of sharing any written documentation. He did allow Mauchly to borrow a copy and take notes, however, and Lura believed that their guest stayed up long into the night to copy them verbatim from the report.

Over the next few months, during which Mauchly would continue to write and ask for updates, Atanasoff and Berry dedicated whatever time they could to their computer. It worked, except for a problem with the output. As with so much of the machine and the theory that lay behind it – Atanasoff even invented his own version of Boolean algebra, having been unaware of George Boole's breakthrough in this area almost a century earlier – the two men had few off-the-shelf solutions to their problems.

One was how to output results onto cards that could then be fed back into the machine for the next cycle of operations. They came up with a solution where a 5000V electronic spark scorched a tiny hole in the card to signify the digit one (if it was blank,

13 Tammara Burton, *World Changer: Atanasoff and the Computer*, p117

14 Tammara Burton, *World Changer: Atanasoff and the Computer*, p122

that meant zero). The cards could then be read because another electronic spark – this time 3000V – could arc through the hole to create a connection. It was yet another ingenious solution to a problem, but not perfect: Atanasoff estimated the failure rate to be between one in 10,000 and one in 100,000. That may sound small, but it's enough to cause inaccuracies over the course of a long calculation.[15]

By spring 1942, this problem aside, the computer was finished. Iowa State College even started a PR campaign, arranging newspaper and radio interviews with the two inventors. 'Machine to Solve Algebraic Problems Replaces 100 Computers', read the headline in numerous papers in June[16] (with 'computers' here referring to humans performing mathematical calculations). It concluded: "The machine is probably the largest ever built. It will replace 100 expert computers with calculating machines when in action. The completion and trial has been delayed by Iowa State College activities in connection with national defence, Atanasoff said. He expects to give the machine an initial trial before the summer."

How frustrating for both men, then, that war intervened before they could find a permanent solution to the output problem. Berry knew he could be drafted at any time: he had now completed his master's degree and the computer was not directly related to the war effort. He chose to take matters into his own hands, accepting a position related to national defence in mid-1942 – and whisking away Atanasoff's secretary, Martha Jean Reed, as his newly wedded wife.

In the scant time Atanasoff wasn't dedicating to additional wartime duties and academic demands, he detailed the documentation necessary for the patent application. An application that Iowa State College never submitted, in part, perhaps, due to the distraction of the war.

Atanasoff was likewise distracted. For the rest of World War II, he would act as a consultant to the Naval Ordnance Laboratory, moving away from Iowa (and his family, one of the contributing factors to the eventual divorce from his first wife, Lura) to Washington DC. He was rapidly promoted to Chief of the Acoustics Division, earning the Navy Civilian Service Award in 1945.[17]

15 Alice R Burks and Arthur W Burks, *The First Electronic Computer: The Atanasoff Story*, p63

16 'Machine to Solve Algebraic Problems Replaces 100 Computers', Telegraph Press news service, Sunday 14 June 1942

17 Tammara Burton, *World Changer: Atanasoff and the Computer*, p134

Still, though, he could not escape the pull of computers. In 1945, he was invited to head up the US Navy's project to "design and build a new high-speed digital electronic computer for general research applications".[18] He accepted, and in February 1946 he first saw a demonstration of the ENIAC.

Neither of the ENIAC's inventors was present at the demonstration, so Atanasoff could ask no detailed questions. Indeed, the computer was then considered to be an object of national security classification, so he could only take any claims at face value. He certainly couldn't peek into the ENIAC's inner workings, as Mauchly had into the ABC so many years previously. What's more, this huge machine was surely light years ahead of the ABC in terms of complexity.

This is why, when Atanasoff found out that the Iowa State College had never paid for his patent to be filed, he was annoyed but not distraught. Times had moved on, and quickly. Plus he was busy with two high-ranking Navy roles, including having to monitor the atomic blasts happening at the Bikini Atoll. Soon after, he was informed that the Navy's computer project was being abandoned.

With Atanasoff still being kept busy with US Navy duties, life continued at Iowa State College. That included a new chair of the Physics Department, who was probably unaware of the now abandoned ABC computer. It's believed that when a grad student asked to use the room, the chair gave his permission to dismantle the machine – it couldn't simply be moved because the chassis was wider than the door. All that is now left of the original machine are the two memory drums and one of the 30 add-subtract modules with its seven dual-triode vacuum tubes.[19]

18 Tammara Burton, *World Changer: Atanasoff and the Computer*, p134

19 One drum can be found at the university museum, the other in the Smithsonian, while John Gustafson (who we will hear from shortly) donated the add-subtract module to the Computer History Museum in Mountain View, California, when the ABC replica moved there.

The machine is probably the largest ever built. It will replace 100 expert computers with calculating machines when in action

The trial

But for the trial, this would have been the end of the ABC (which at this point, remember, did not even have a name). Its place in computing history was only sealed when US District Judge Earl R Larson recorded his verdict on 19 October 1973 that the ENIAC patents were invalid. His third finding was the most controversial, stating: "Eckert and Mauchly did not themselves invent the automatic electronic digital computer, but instead derived that subject matter from one Dr John Vincent Atanasoff."[20]

Effectively, and this remains hugely controversial, the US justice system had named Atanasoff as the creator of the first electronic computer. This would have been huge news, but one day later, President Nixon fired the special prosecutor looking into Watergate, in what would become called the Saturday Night Massacre. There was no space in newspapers to cover the story of a patent dispute.

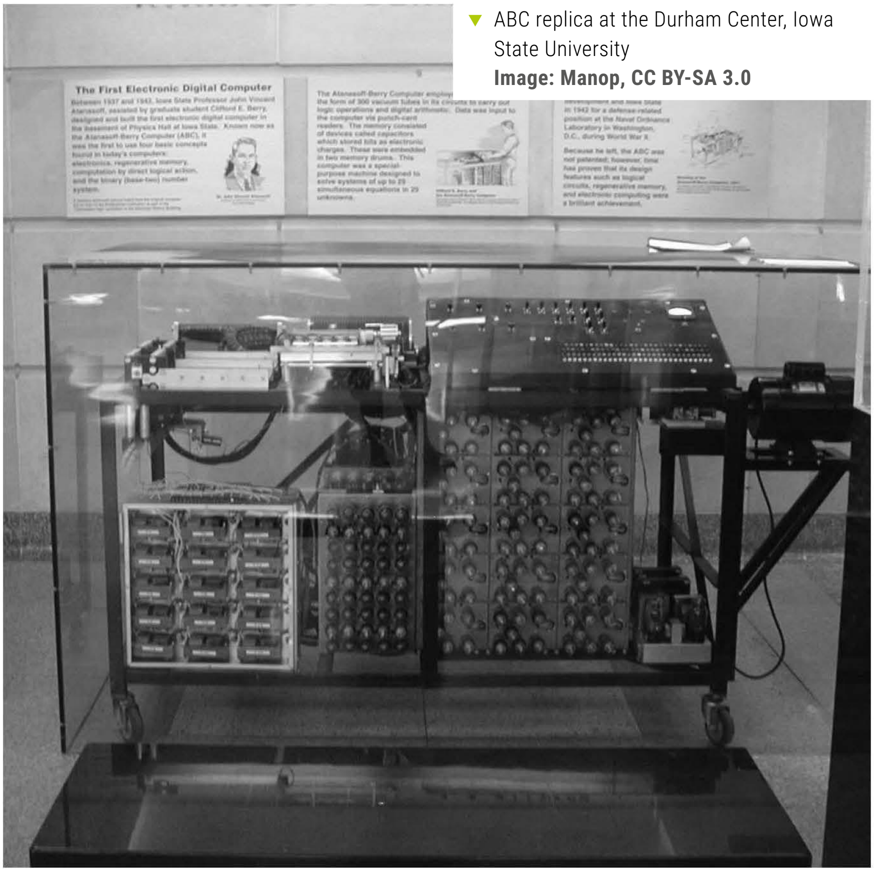
It took another three months, and the determination of Atanasoff's second wife, to bring it to the attention of an Iowa-based reporter. But it was worth the wait. On 27 January 1974, the Des Moines Sunday Register stamped 'COURT: COMPUTER IOWAN'S IDEA' across its front page. This time, the story soon spread across the country, and belatedly the Iowa State College – now named Iowa State University (ISU) – took note. From then onwards, the establishment made much of its part in the history of computing.

This came to a head in 1994, when the ISU decided to build a working replica of the computer. "George Strawn, who had been a recent Department Chair of Computer Science (not the one who disassembled the original ABC) and Delwyn (Del) Bluhm spearheaded the idea and worked to get it funded," said John Gustafson,[21] founder of the ISU's Scalable Computing Laboratory and by that point author of several papers on how the ABC operated. On hearing of the project, Gustafson quickly lent his enthusiastic support to it.

20 Judge Earl R Larson's official notes, 19 October 1973. A copy can be read at **jva.cs.iastate.edu/Legal%20Decision_32-2.pdf**

21 In email to author

"This ABC Replica Project was conceptually important," wrote Del Bluhm, the project's first director, in 2013,[22] "since a working replica of the ABC had to be tested to prove that it did in fact work properly and therefore was the world's first electronic digital computer."



▼ ABC replica at the Durham Center, Iowa State University
Image: Manop, CC BY-SA 3.0

The only problem was that this project would cost hundreds of thousands of dollars. A sum the university was not willing to cover. Fortunately, Strawn and Bluhm secured a large chunk of funding from Charles W Durham, a former student of Atanasoff's at the ISU, and started putting the plan together.

John Atanasoff was still alive at this point, and gave the team access to all his documentation and diagrams. Sadly, he would not live to see the finished replica, dying in June 1995 at the age of 91.

By this time, Bluhm knew the full scale of the challenge. "Our effort involved not only reproducing parts from limited original

22 Del Bluhm, 'Documenting the ABC Replica Project and its Contributors', in Atanasoff Today, Spring 2013, p14

designs, but also re-engineering designs after interviewing those people who either worked on the original computer or saw the prototype in operation," he wrote.

Finally, by the summer of 1996, an enthusiastic team – including engineers who had worked on early electronics so were familiar with vintage vacuum tubes – were ready to start the building process. The replica was finished in April 1997, but with one caveat: it didn't actually work. While all the components and wiring were present, they weren't necessarily in the right order.

Bluhm's team had done their job of faithfully creating the replica. The challenge was that they were engineers rather than computer scientists, so while all the components and wiring were present, they were still some distance from having the working replica required to prove to naysayers that the ABC computer worked as designed. At this point, John Gustafson was asked to take control of the replica project.

As part of the project, the team had gathered as many people as possible who had seen, heard or worked with the ABC during its short active life, and one of those people was Clara Smith. Previously a secretary in the ISU's Statistics department, she revealed that the ABC had been put to real work by head of the department, Professor George Snedecor.

"She sent [Atanasoff and Berry] systems of equations for least squares fit to be solved," said Gustafson.[23] "They sent back the answers, and then she'd use the desktop calculator, like a Marchant calculator, to check the answer. And that's not much work: that's just about six multiply-adds, and you've got to see whether AX does equal B. But it's much harder to do the inverse – to solve a system with two equations, two unknowns – than to check that it's correct."

Clara Smith explained that Snedecor then used those results in his research papers, which were published at that time. "So I would say the ABC was a production computer," said Gustafson. "It was put into practical use by Snedecor exactly as Atanasoff envisioned."

Due to the nature of human memory, not all of the people Gustafson and his team spoke to gave such detailed answers as Clara Smith, but over time he was able to build up an idea of what the ABC looked like, what it sounded like and tiny details such as where the odometer, which counted upwards to indicate number outputs, was placed. They also discovered that the team building the ABC "had a deck of five cards that was their standard five-by-five test to see whether the machine was operating," said Gustafson.

With all possible information now gathered, it was time to get their hands dirty and start the wiring. Or at least, for Charles Shorb to get his hands dirty. "He was a grad student in my research group who could really do everything," said Gustafson. "He was scared to death at being given the project. 'Frankly,' he said, 'I don't know what to do.' I said, you will."

From the moment he saw the replica, Shorb was in awe of both its original inventors and everyone who had worked on the project so far. "It was the most beautiful wiring I'd ever seen in my life," said Shorb.[24] "You think of old computers taking up a whole room, but [Atanasoff] created basically an automatic Marchant calculator that's the size of a desk. That's phenomenal."

The ABC was so compact because no space was wasted. "The control circuits were sat right behind the add/subtract modules, then you have the circuits that sat behind that control the thyratron firer for the base two input/output," said Shorb. "And then right behind that were the control circuits where the wiring went up to the commutators that actually was the brains of the whole thing." Looking at this, following the trail of wires, was reminiscent of modern circuit design, Shorb added.

He then almost ruined the whole project by switching it on, a move greeted with wisps of smoke. Fortunately, that problem was easy to debug – the power bus was wired incorrectly – and he set to work. Painful work, as it turns out, because the extreme voltage, going from −120V to +120V, resulted in "way too many" electric shocks for the young grad student.

Despite the pain, what followed was three months of working "day and night" on the wiring and the circuitry, to ensure that the replica worked exactly as Atanasoff and Berry intended. Fortunately, they had left behind detailed schematics. Despite his work ethic and evident motivation, Shorb could not finish everything before he left in September to join Intel. Due credit should go to Guy Helmer, now a professor at ISU, who later completed the wiring.

By the time Shorb left for Intel, he had wired five of the fifty blocks, and that was enough for the team to achieve its goal: to create a working replica. One that would prove wrong any doubters – the ABC was more than just a concept.[25]

So what was it like to use? While in operation it was "no louder than a sewing machine, said Gustafson, "from an electrical signal viewpoint it was horrifically noisy". "There was so much jitter. Brushes on metal produce a terrible oscilloscope signal, let me tell you, and I thought this machine was never going to work.

23 Interview with the author

24 Interview with the author

25 While the public can't try the ABC out for themselves, visitors to the Iowa State University computer lab can still admire the replica. You can also see it in action on YouTube thanks to a 1999 recording featuring Gustafson and Shorb at **youtu.be/YyxGIbtMS9E**

But the extreme voltage was enough that even a noisy signal was able to compute and make things click along. And so it was a thrill when we gave it a problem and it actually solved it."

It was time to go public, which the team did at an event at the National Press Club in Washington DC on Wednesday 8 October 1997. Gustafson and Shorb devised a simple problem where the answer was three: 9 - (3×2) = x. They explained to the watching journalists – which included staff from the National Geographic and Time Magazine – that when they heard three clicks on the odometer they knew the computer had worked its magic.

"Switches were flipped, a punch card was inserted and an electric motor turned a drum studded with copper contacts," wrote Kenneth Pins in the write-up for The Des Moines Register.[26] "It was 2½ minutes later when the numbers on a device resembling a card's odometer advanced three clicks, revealing that the answer was 3."

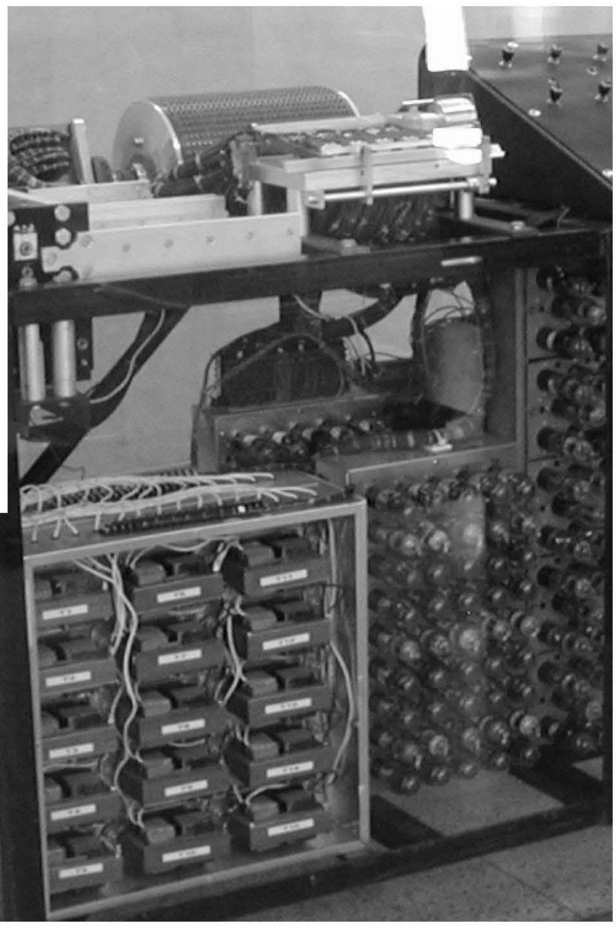
The extreme voltage was enough that even a noisy signal was able to compute and make things click along

After heading out on a nationwide tour, the ABC replica now resides at the Computer History Museum in California, next to the ENIAC. As one of the few people to operate the ABC replica, we asked Shorb if he thought the ABC's impact would have been different if Atanasoff and Berry had the time to solve the card output problem. "Absolutely," he said, adding: "I think that history would have been more kind to the ABC if they would have gotten the patent."

But he also makes an important point: "When you really take a look at the history of computing as a whole, you realise that no single person can claim anything… but you at least should give credit where credit is due for the advances that people make."

Dr John Atanasoff and Clifford Berry certainly deserve credit, as do the team who built the ABC replica almost 60 years later.

26 Kenneth Pins, 'Computer age dawns again thanks to ISU', in The Des Moines Register, Thursday 9 October 1997, p1



ABC replica at the Durham Center, Iowa State University **Image: Manop, CC BY-SA 3.0**

DIVE INTO MARINE TECHNOLOGY

Discover Raspberry Pi sea-based projects and build an underwater robot.

By **Lucy Hattersley**

The ocean is a vast subject: mysterious to humans, both physically and technically. It's striking how much more we know about space than our own ocean floor. We have mapped the Moon, Mars, and Venus more than our own oceans.

The ocean is physically challenging, with high pressure, total darkness, and extreme cold adding mechanical considerations on top of the usual project build requirements. Salt water is tough on equipment, and opaque to light. Radio waves do not travel well underwater, especially those at higher frequencies like Bluetooth, Wi-Fi, or standard radio communication. The oceans are huge!

> It's no surprise to find Raspberry Pi equipment working under the ocean

Well-engineered Raspberry Pi hardware is at the forefront of this challenge. It's no surprise to find Raspberry Pi equipment working under the ocean, measuring everything from marine populations to mapping the ocean floor.

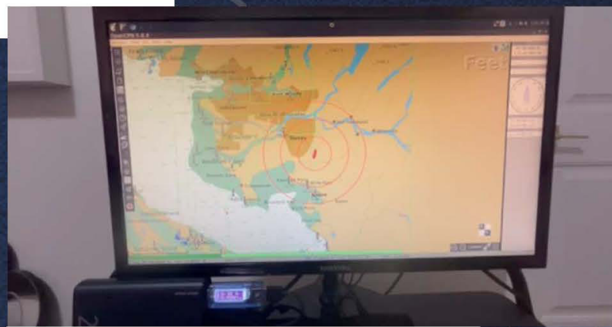
In this feature, we'll look at the professionals. Then Jo Hinchliffe is going to take us over TOUV (a Tiny Open-source Underwater Vehicle). From offering inspiration to your practical starter guide to underwater robotics, this feature will submerge you in detail.

Let's dive on in...

Nemo-Pi

Save Nemo creates underwater weather stations for coral reef monitoring. These solar-powered buoys measure water conditions including temperature, pH, visibility, current strength, and chemical concentrations. The systems upload real-time data to public servers, helping divers and researchers investigate reef conditions.

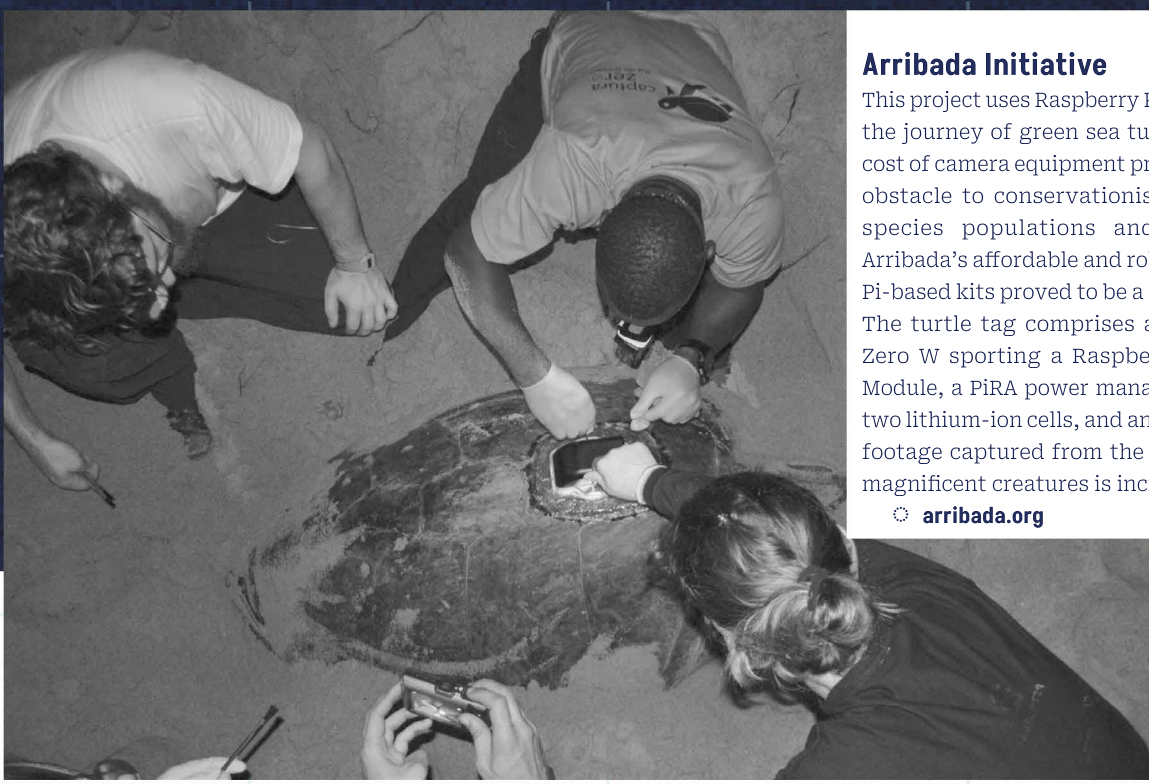
nemopi.com



Dave the MMP

Dave is retrofitting a Cal 2-30 sailboat using a marine navigation system built on Raspberry Pi 5. This provides a powerful alternative to expensive commercial chart plotters, running OpenCPN (**opencpn.org**) software with full NMEA (National Marine Electronics Association) integration. His YouTube channel shows a full integration of Raspberry Pi 5 with his boat.

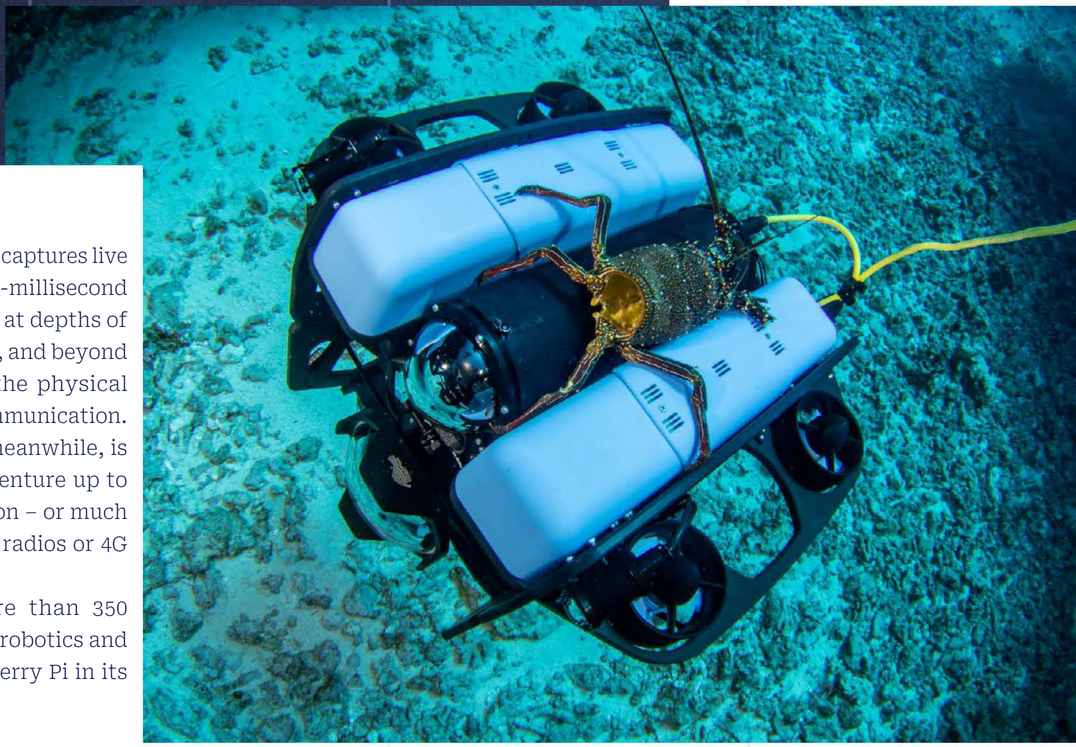
rpimag.co/davethemmp



Arribada Initiative

This project uses Raspberry Pi Zero to follow the journey of green sea turtles. The high cost of camera equipment presented a huge obstacle to conservationists monitoring species populations and behaviours. Arribada's affordable and robust Raspberry Pi-based kits proved to be a game-changer. The turtle tag comprises a Raspberry Pi Zero W sporting a Raspberry Pi Camera Module, a PiRA power management board, two lithium-ion cells, and an enclosure. The footage captured from the backs of these magnificent creatures is incredible.

arribada.org



Blue Robotics

The six-thruster craft BlueROV2 captures live 1080p video footage with a 200-millisecond latency. BlueROV2 can operate at depths of up to an impressive 300 metres, and beyond this is primarily limited by the physical tether cable that it uses for communication. The BlueBoat surface vessel, meanwhile, is connected via radio and can venture up to a kilometre from its base station – or much further if higher-performance radios or 4G are used.

Blue Robotics offers more than 350 products in the field of marine robotics and plans to continue using Raspberry Pi in its professional productions.

rpimag.co/bluerobotics

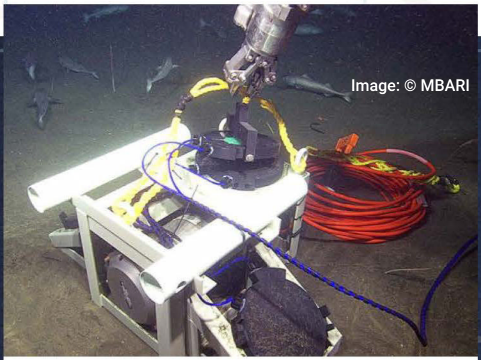


Image: © MBARI

Raspberry Shake

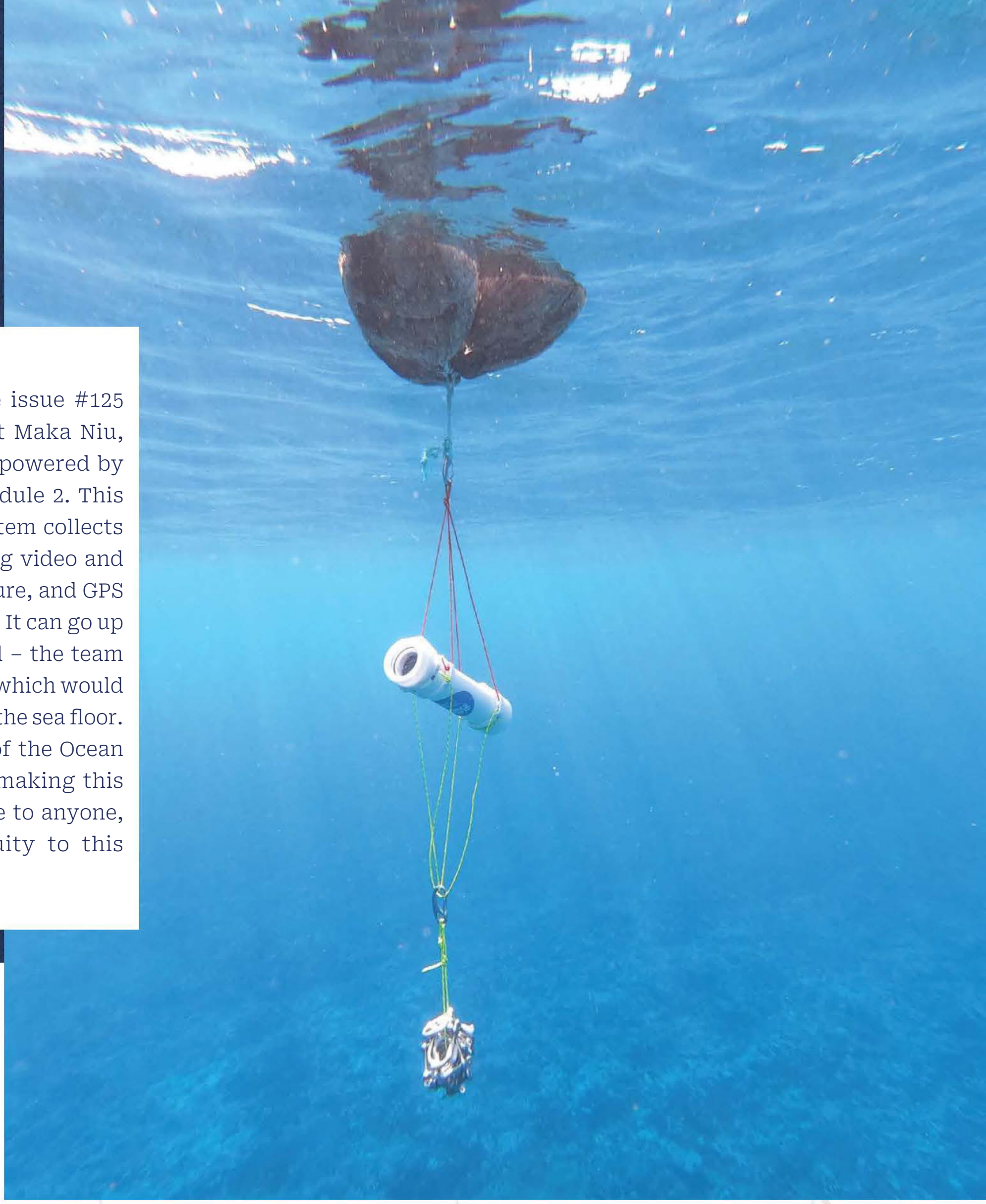
We're big fans of Raspberry Shake. Normally this equipment is used to measure seismic activity on the surface of the globe, but in 2020 the Monterey Bay Aquarium Research Institute placed one 900 metres below sea level. The final unit consists of a small fibreglass frame with a titanium pressure housing that holds Raspberry Shake. On the sides and top of the frame are black plastic 'paddles' that use radio waves to transmit power and data wirelessly underwater.

rpimag.co/shakeocean

OpenPlotter

On the ocean surface, Raspberry Pi has become the foundation for marine computing systems. Once dominated by expensive, closed-source systems, the world of marine electronics has been upended by OpenPlotter. Tailored for Raspberry Pi, this Linux-based operating system transforms the single-board computer into a complete marine navigation hub. It brings together weather information, chart plotting, GPS coordinates, and nearby AIS (automatic identification system) targets. It even provides an onboard wireless hotspot.

openmarine.net/openplotter



Maka Niu

Back in *The MagPi* magazine issue #125 (**rpimag.co/125**) we looked at Maka Niu, a modular deep-sea explorer powered by Raspberry Pi and Camera Module 2. This robotic deep-sea imaging system collects several types of data including video and still imagery, depth, temperature, and GPS location when out of the water. It can go up to 1500 metres below sea level – the team has designs to go 6000 metres, which would enable it to reach 99 percent of the sea floor.

Katy Croff Bell, president of the Ocean Discovery League, says: "By making this technology open and available to anyone, we hope to bring more equity to this important field."

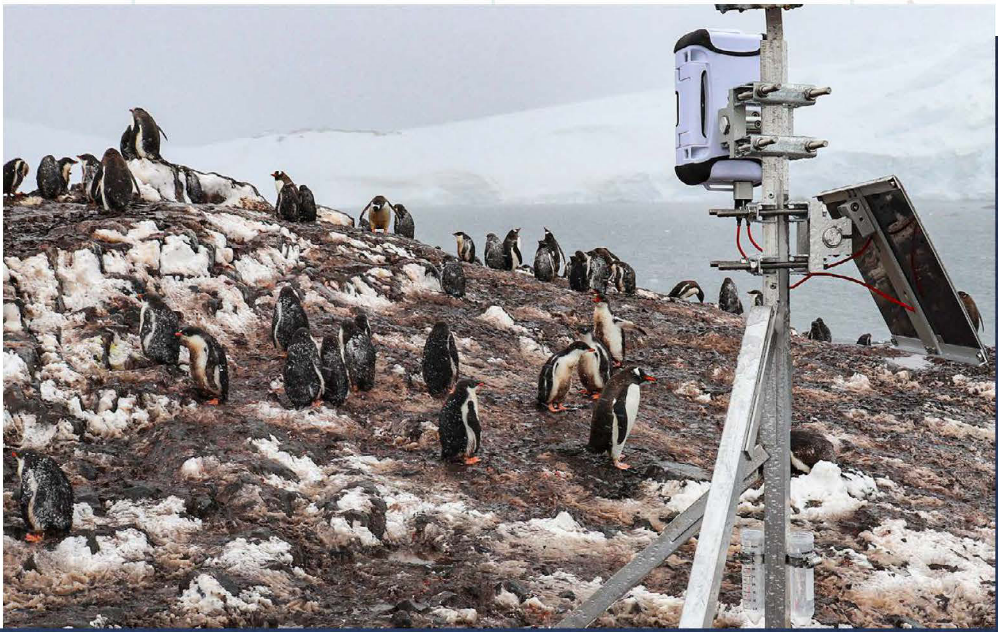
rpimag.co/makaniu

It can go up to 1500 metres below sea level

Penguin population monitoring

The Arribada Initiative also uses Raspberry Pi as part of Penguin Watch – an ongoing project to monitor penguins using Raspberry Pi hardware surviving multiple hard Antarctic winters. This project was covered alongside turtle monitoring in *The MagPi* magazine issue 123 (**rpimag.co/123**). Arribada initiative director Alasdair Davies explains that Raspberry Pi has always been one of the tools he uses "because it's so accessible and affordable for anyone to get involved. You're always partnering with a researcher or a local community member who works with an NGO and has a specific challenge. We get called in to solve it with technology. They'll always say it has to be affordable, it has to be repairable and accessible."

rpimag.co/arribada



Underwater ROV

This homemade remotely controlled submarine combines with a Unity app for an Android device that is used to record video and drive the submarine.

It uses a Raspberry Pi as its central control unit, a Camera Module 2 for video recording, sensors, and drivers, as described in the schematic (linked below). A test vehicle was constructed out of PVC-U sewer pipe with custom 3D-printed PLA parts and an acrylic dome, all sealed together using epoxy glue.

It's a great example of homemade hacking and what can be done to get started with putting a Raspberry Pi underwater.

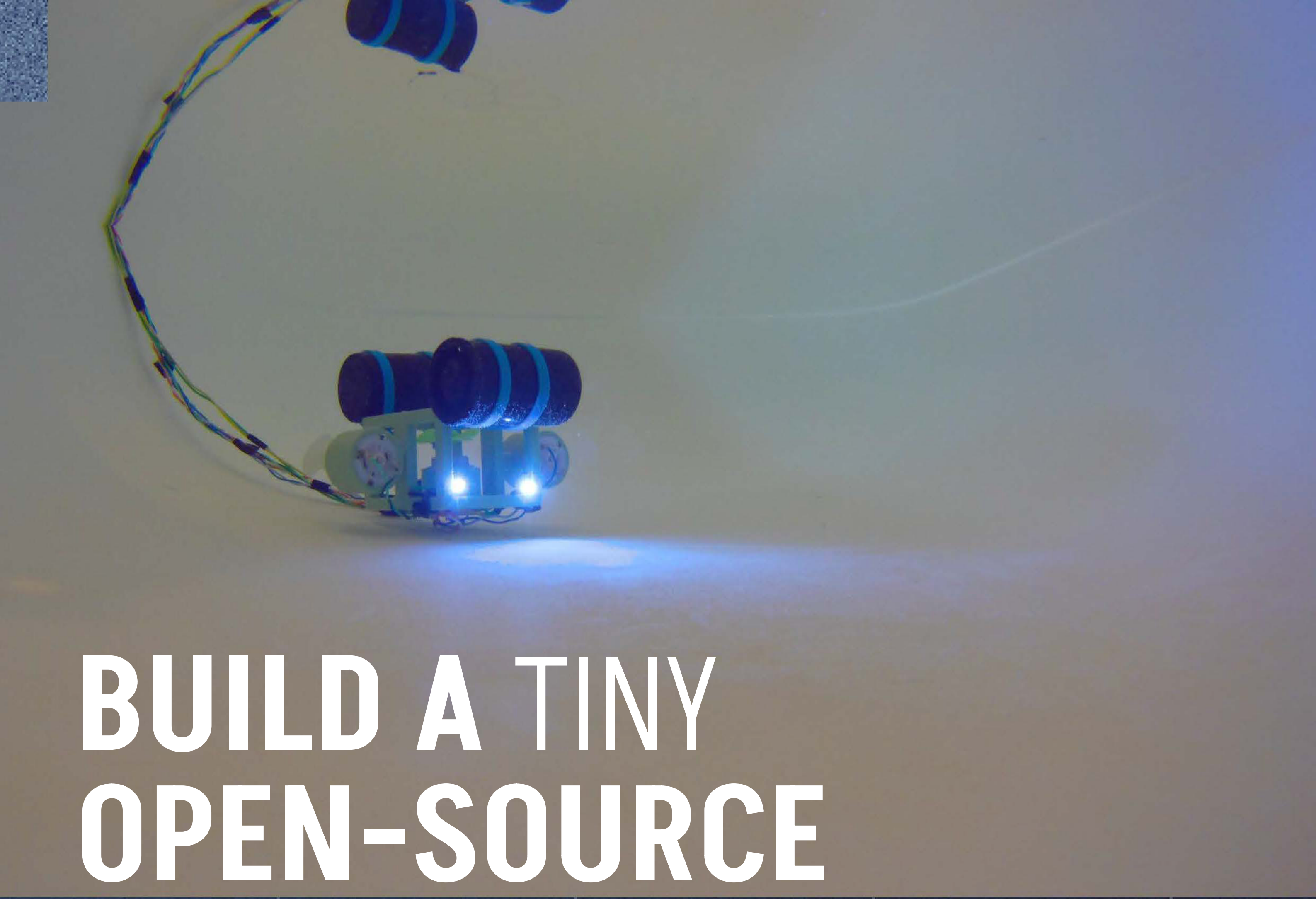
rpimag.co/underwaterrov

Autonomous recording for marine ecology

Cetacean species, including whales, dolphins and porpoises, are considered indicators of the health of marine ecosystems around the world. While a number are known to be endangered, a lack of data means that the population size and conservation status of many species are impossible to estimate. These animals are vulnerable to the effects of human activities and the noise they cause.

A team from the University of São Paulo in Brazil published a paper about the flexible, low-cost autonomous recorder they have built, based on a Raspberry Pi. The hydrophone – an underwater microphone – is protected by a cage made of stainless steel. This enclosure successfully withstands pressures of up to 10 bar, equivalent to those experienced almost 100m underwater, in pressure-chamber testing.

rpimag.co/underwaterrecording



BUILD A TINY OPEN-SOURCE UNDERWATER VEHICLE

With some 3D printing, motors, and a Raspberry Pi Pico, you can be up and running with an underwater explorer!

By **Jo Hinchliffe**

▲ The Tiny Open-source Underwater Vehicle (TOUV) being put through its paces in fresh water (salt water will corrode components)

▲ **Figure 1:** the main structural components for a TOUV – a chassis frame, 3 × 30mm motors, 2 × 35mm film canisters, some propellers, and some cable ties

About a decade ago, we were inspired by a workshop at Liverpool Makefest where an organisation called the Dark Water Foundation made all manner of water monitoring devices.

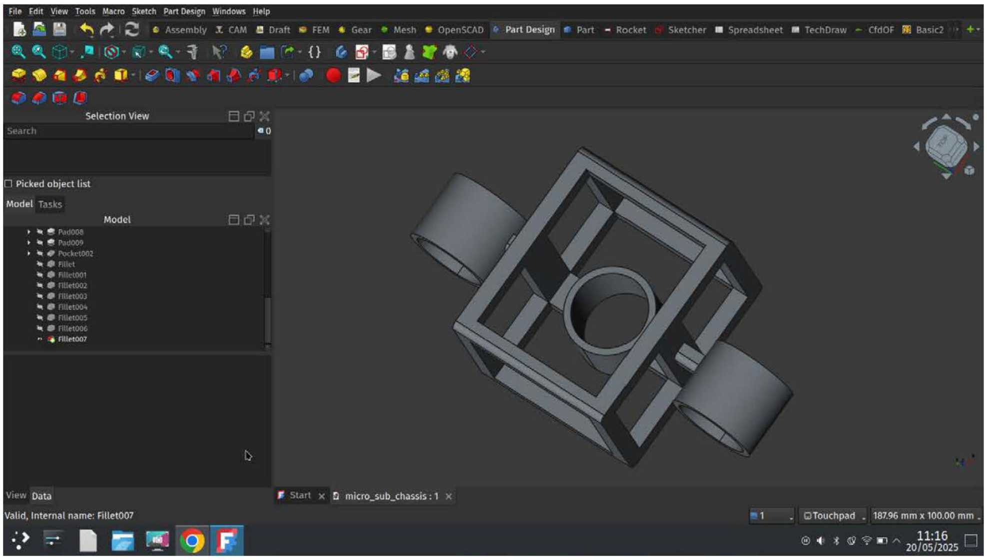
Dark Water Foundation got attendees to make miniature palm-sized underwater rovers using Lego, 35mm film canisters, and some small brushed DC motors. "DC motors?" I hear you cry. Well yes, the revelation on seeing this workshop was that small 3–6V DC motors run really well in fresh water without any attempts to waterproof them. The caveat being you are probably shortening their life somewhat, and you need to do a little work to dry them out after each mission.

Many of the original links to this workshop are now missing; so, with absolute respect for the original project, we decided to make a more modern attempt influenced by that original design. There still is an Instructables page with some images and component lists, but sadly the links to components from the page aren't currently working, so we've made some educated guesses and experimented to come up with a working solution.

> For our more serious testing, we moved to a second-hand inflatable hot tub

The original design was around a 60–75mm cube for the frame made from Lego. We set out to 3D-print a similar-sized frame using the free and open-source FreeCAD package to realise our ideas. The original design used 30mm brushed DC motors and we discovered that these at the time were rated to 5400rpm. We couldn't find that exact specification, but found some close enough listed as 3–6V and 6500rpm. These are perfect because we intend to make a moderately more advanced controller for our design – we can use motor drivers to control our speed.

Once the motors arrived (**Figure 1**), we could then grab a set of callipers and take some motor measurements and begin our CAD work. In FreeCAD, we used the Part Design workbench to create the chassis, building up the frame by



◀ **Figure 2:** Designing the chassis in FreeCAD was a pretty simple task

adding sketches to faces and extruding them (**Figure 2**). Our main chassis area is a cuboid frame and the three thrusters: two set up for forward and turning (and potentially reverse if needed) on the sides of the frame and one inside the frame, set in the vertical axis to control vertical position. The two horizontal thruster mounts are offset slightly from the centre in an attempt to distribute the weight a little.

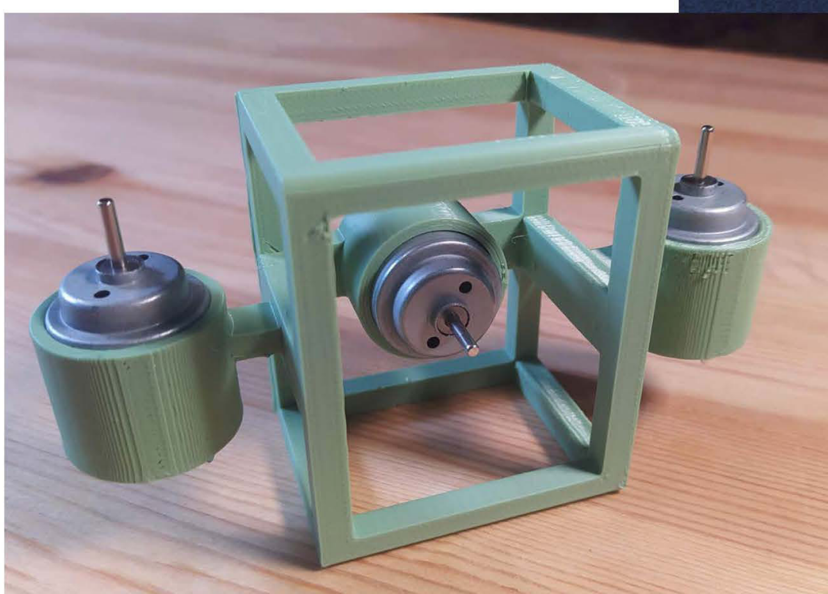
Part of the design led us to consider what our buoyancy approach might be. There are two or possibly three options. We can either aim for absolute neutral buoyancy, where the vehicle neither descends or ascends in the water until a motor turns, positive buoyancy where the vehicle always wants to float and the Z axis thruster pushes it down, or negative buoyancy where the vehicle always sinks until the Z axis thruster supplies lift. Of course, the optimal is neutral buoyancy, but this is hard to achieve, particularly when you have tether wires which change the mass of the setup as it moves; also, for neutral buoyancy, we'd need to be able to switch the rotation of the Z axis thruster.

Negative buoyancy is reasonably easy to achieve. We are going to add two 35mm film canisters (still widely available online) and we can simply fill these with water to make the vehicle sink. Neutral buoyancy has some advantages and a disadvantage. An advantage is that if you completely fill the film canisters with water then they are equal and balanced – if they are part full, it can be difficult to not have the water move around and change the orientation of the vehicle. The disadvantage of negative buoyancy is that it uses a little more power as you'll use the thruster a lot more. And if for some reason the power or motors fail, your vehicle will sink – perhaps not too much of an issue at this scale as you can pull it out by the wire tether.

After printing a test frame in PETG filament, the assembly of the vehicle is pretty straightforward. With a reasonably well-calibrated 3D printer, the thruster mounts are tight enough for the motors to be a push fit (**Figure 3**). We used some 26mm two-blade nylon propellers for the horizontal thrusters and a larger three-blade 30mm propeller for the Z axis, making sure to order ones with the correct 2mm holes to mount to our 30mm 3–6V brushed DC motors.

It's important to try and find a balance of long wires that are thin and flexible to create the power lines for the motors. We found an old three-metre network patch cable that

▶ **Figure 3:** the motors are a friction fit in the chassis, meaning you can adjust their position a little, and are easy to replace if ever needed

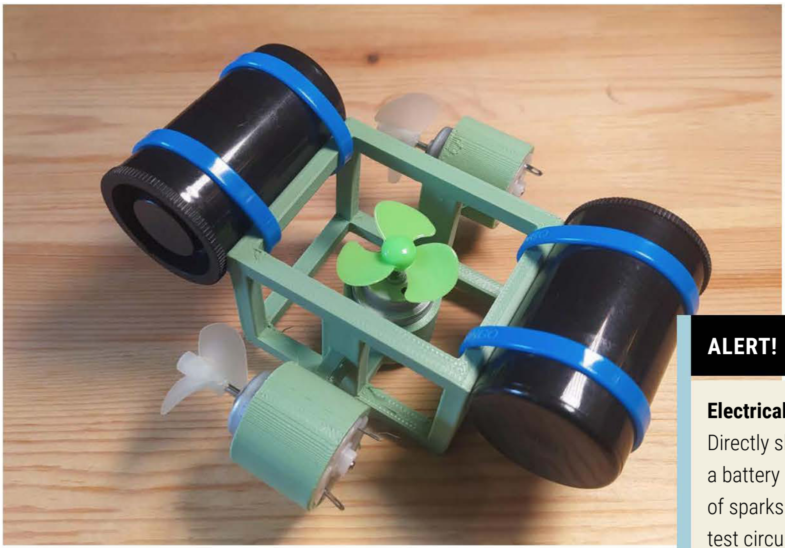


we pulled apart to get long lengths of suitable wire. If you wanted to spend rather than find something from your junk cable pile, you could probably find thinner and more flexible solutions. As the chassis sits quite nicely on a worktop, we inserted the motors into the chassis and then soldered the wires on, using the chassis as a helping hand.

Adding the buoyancy tanks is as simple as two cable ties wrapped around the chassis and the 35mm film canisters (**Figure 4**). You can experiment with the position of the tanks slightly after they are fitted. If you place them more out in the front and the rear, if the vehicle is floating you might find your Z axis thruster sits slightly out of the water. Raising the tanks above the vehicle means the Z axis propeller is always covered in water.

Once you have your motors in place and wired and you have fitted the buoyancy canisters, it's time to test. We ran our first tests by simply holding the motor wires across an 18650 battery momentarily to work out the polarity. With all that identified, we then couldn't resist playing with it in a washing-up bowl full of water. This proves quite worthwhile as although you can't really do too much, you can get a sense for if you want to have positive or negative buoyancy and play with adding different amounts of water to the floatation canisters.

At this point, you could certainly make up a straightforward controller box with just some buttons or switches for each motor and some batteries. However, we went with the addition of a Raspberry Pi Pico as a controller, as this allowed us to drive the motors using pulse-width modulation (PWM), which allows us to control their speed and direction. To do this, we needed to use some motor drivers and we actually used a StoRPer board, which is a PCB we made as part of our *Design an RP2040 board*



▲ **Figure 4:** Adding the 35mm film canisters to act as buoyancy tanks

ALERT!

Electrical issues

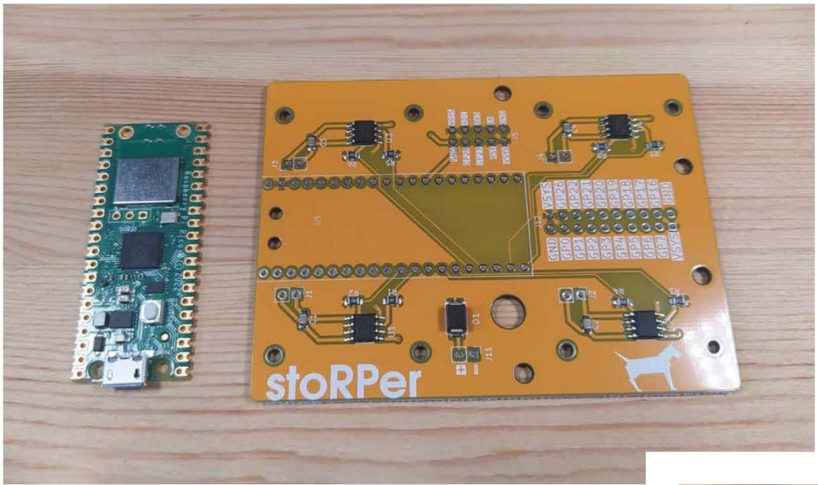
Directly shorting wires across a battery carries a slight risk of sparks. Consider building a test circuit. Li-ion cells can also cause burns or fire if shorted. The TOUV project uses 3-6V DC motors. Please do not use higher voltages or larger voltage motors.

Motors and water

One of the main reasons this project has been in our minds ever since seeing the Dark Water Foundation Lego robots at Liverpool Makefest a decade ago is that we were shocked to see that DC motors ran underwater with no attempt at waterproofing. This is not a common practice, but it does work.

We can't say that it is the healthiest environment for small brushed motors, but it certainly isn't as bad an idea as it sounds. The motors run fine and we've tended to take some care when we finish an underwater session. We've spun the motors out of the water for a good two minutes of constant running to push out remaining moisture. We've also given them a good blast of either canned air, or a decent blow from the mouth when our canned air ran out.

We are sure that running DC motors in fresh water is not good for them and we are definitely shortening their lifetime; however, ours have survived a lot of underwater time now and we are still on the original pair. We also found our motors online at less than 60 pence each when buying a ten pack, so it isn't the most costly thing to replace them if and when they fail.



▲ **Figure 5:** Using a StoRPer board is functionally the same as adding some DRV8833 modules to a Pico

Warning!
Chemicals
Please handle chemicals such as chlorine responsibly.
rpimag.co/chemicals

with KiCad book (**rpimag.co/kicadbook**) and articles that featured in *HackSpace Magazine*. The StoRPer board (**Figure 5**) is essentially a breakout board that connects some of Pico's GPIO pins to four DRV8833 motor drivers. You could totally emulate this board on a breadboard using the StoRPer schematic and components; however, DRV8833 motor drivers are commonly sold as small modules on breakout boards and that would make for a simple option.

For our first controller system, we mounted three momentary press-to-make buttons in a panel (**Figure 6**) and connected them to Pico's GPIO. In the MicroPython code (**Figure 7**), we have identified the GPIO pins for the buttons and defined them and set up a PWM system for each of the motors. Swapping the pair of pin number values for each motor can then reverse that motor and we can also change the speed values for each motor.

For our more serious testing, we moved to a second-hand inflatable hot tub. These can often be found second-hand, neglected and filthy, for very little money and make for an excellent large test environment for water-based projects. We gave ours a pretty good scrub and when we filled it, we gave it a dose of chlorine to kill off any bugs a couple of days before using it (to allow the chlorine to dissipate). As an aside, it's great fun to place an action camera in a waterproof case in the tub/tank and capture some footage/images of your underwater test missions! Here is a link to an edited minute of underwater testing: **rpimag.co/touvvid**.

One thing we will change is that our wires are hard-wired to the controller and it would be beneficial, especially at the testing stage, to be able to quickly remove the wires so the

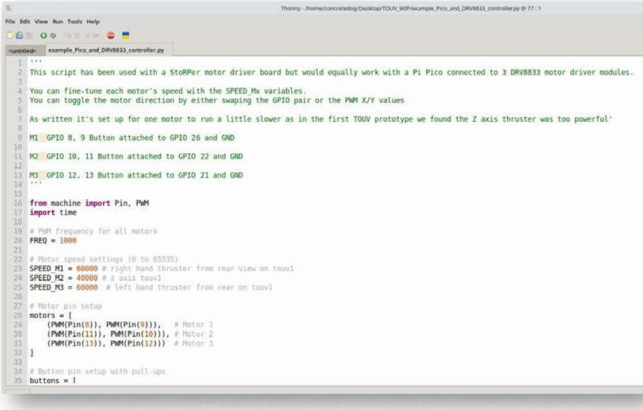
◄ **Figure 6:** Adding a small panel into which we mounted the buttons as a first attempt at a controller

We mounted three momentary press-to-make buttons in a panel

LEDs for looks

Sometimes you add features just because. This was certainly the case with the pair of LEDs we mounted to our build. Obviously if we had some kind of camera system on board then lighting might have a purpose, but we just added these for the cute factor. It's as straightforward as soldering a current-limiting resistor in series to the anode of an LED – we went with 100 ohms, which offered protection across the range of voltages we were intending to use on our build (3.7–5V).

Once soldered to some wires, we added a little heat-shrink to stop the LED anode and cathode potentially touching, and for a little structural reinforcement. We then used some small cable ties to attach the LEDs to the chassis.



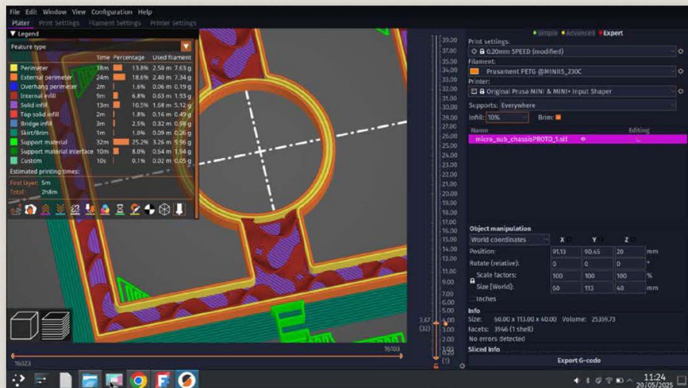
◀ **Figure 7:** Using MicroPython on the Pico meant we could quickly make changes to motor speeds and directions

controller alone can be taken back to the laptop inside (not near the hot tub) for code tweaks. We did a lot of drying of the vehicle and wires every time we needed to make a change. We powered our system a couple of different ways: either with an 18650 cell or with a USB power bank. With the 5V of the power bank, the motors spin quite quickly and the vehicle has too much thrust and is difficult to control, but this is easily rectified with changes to the speed values in the MicroPython script. We're currently set up with a slight positive buoyancy with the Z axis thruster set to turn a lot slower than the horizontal thrusters as it's nice to be able to descend smoothly and gradually. We've also ended up drilling the small holes in the chassis to flood the chassis when submerged.

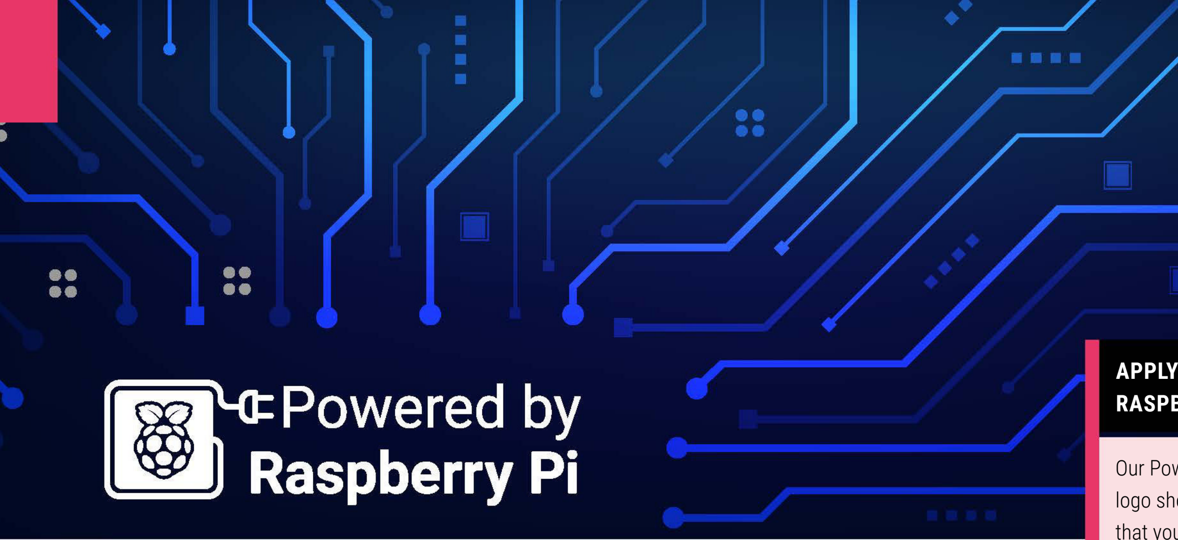
It's great fun to tinker with and we have a few ideas for improvements. It would be great to make the wiring a little more flexible with some thinner and more flexible wires. We'd also like to add more functions to the controller. A switch to reverse the direction of the horizontal thrusters would be nice, as well as some latching switches that can set motors to be permanently running at a slow rate, perhaps even with potentiometers to vary the speeds. For example, we could set a slow speed for the Z axis motor that, when activated, counters the positive buoyancy, allowing the TOUV to have a kind of 'altitude hold'. It could be fun to add attachments like a hook or a magnet for fun 'rescue the object' games and challenges. We would of course love to get a camera on board, and sensors and more functionality, but we think that might be best on a larger DIY submersible project. Keep an eye out for that in the future. In the meantime, if you want to build a TOUV, the files are in this GitHub repository: **rpimag.co/touvgit.**

Larger vehicles

Researching DIY underwater vehicles, there is a whole scene of larger vehicles with their chassis built from PVC or ABS plumbing pipes. Most of these builds have small holes drilled into the pipes to allow the chassis to fill with water when submerged. This has some great benefits; the added mass stabilises the vehicle in the water and it means that you are only dealing with your actively added floatation tanks or tubes when tuning the setup.



After some pondering about designing holes into the small chassis, we realised this could be achieved a different way. Inside filament-type 3D-printed structures, you tend to get your slicer software (that prepares 3D objects for printing) to automatically add a percentage of infill. This infill is generated in patterns to create some kind of lattice inside your object, adding to the strength of the structure. Most slicers offer a collection of different patterns that can be used to create infill. A lot of the time, a square or hexagonal mesh is fine, but there are some interesting options. One option is a curvy gyroid infill. One interesting aspect of gyroid infill is that although it creates strong bonds between the parts it fills, it doesn't ever close off sections within the object. This means that if you print with a gyroid infill and then drill some small holes in the corners of the chassis, the chassis can completely fill with water and also completely drain water after an underwater session.



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

It's no secret that Raspberry Pi products are a popular component in commercial products, as well as in our favourite hobbyist projects. Whether it's the full-board version, such as Raspberry Pi 5, or its smaller embedded partner the Compute Module 5; or the even smaller form-factor Raspberry Pi Zero, Pico, or the tiny-but-mighty RP2040 and RP2350 range of microcontroller chips; you'll find them all sitting inside some of the best products around.

Raspberry Pi technology sits behind over 350 professional products: from underwater rovers to washing machines. We're incredibly proud that our computer hardware helps developers bring their industrial ideas to life.

The Powered by Raspberry Pi programme is a distinguished mark of quality for products built on Raspberry Pi technology. It's our way of recognising the professionalism of others. Such products carry a distinctive Powered by Raspberry Pi logo and have gained approval from Raspberry Pi's in-house team for compliance with a stringent set of standards. Whether you're building a sophisticated industrial machine, or elegant consumer gadget, or looking for what people are developing in the Raspberry Pi ecosystem, our Powered by Raspberry Pi programme is your guide to what is on the market.

rpimag.co/poweredbyraspberrypi

PT Ecological Services

India | ptecologicalservices.com

PT Ecological Services was founded in 2011 with a mission to monitor and clean up pollution at sewage plants in India. The nitty-gritty of sewage industries involves surveying the sites in question and providing real-time monitoring so spills and spikes in smog and effluent releases can be identified and causes traced.

PT Ecological Services' OPM200/OPM300 Pollution Monitoring System is based around Raspberry Pi 3 and bristles with advanced sensors that measure pH levels, dissolved oxygen, turbidity, and pollutants in water. Turbidity refers to the amount of sediment or suspended particulates in water, so could include sewage. At industrial sites, the PTES system might be used to measure instances of heavy metals, pesticides, microbial contaminants, and organic compounds.

Dreamboards DreamHAT+

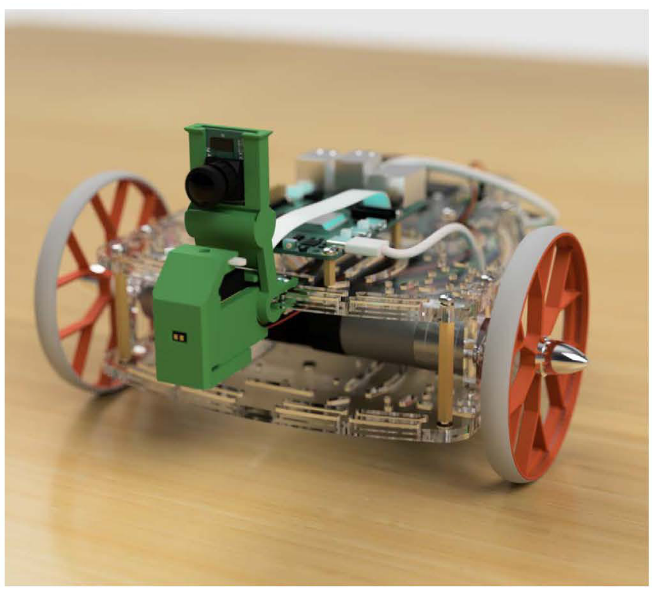
UK | dreamboards.co.uk

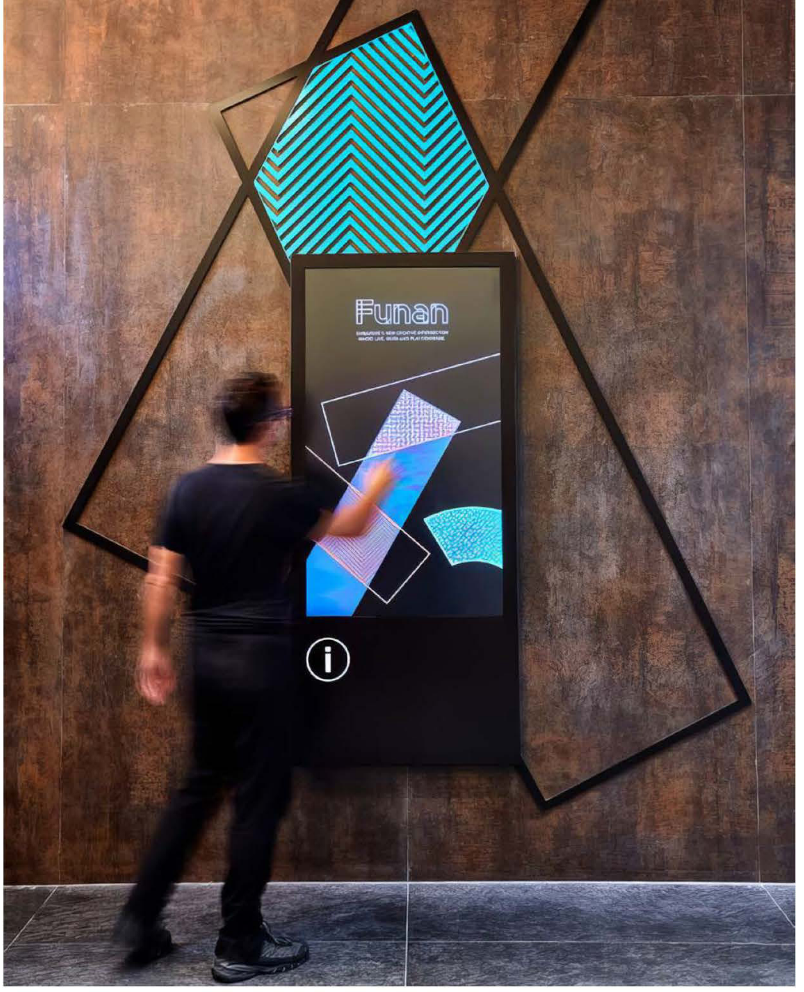
DreamHAT+ is a 60GHz millimetre-wave-sensing board for Raspberry Pi 4 or 5 that can be used for motion tracking, proximity detection, and gesture recognition. Reviewed in *Raspberry Pi Official Magazine* issue 154 (**rpimag.co/dreamhatreview**), the DreamHAT+ sports an Infineon BGT60TR13C chip, three receivers, and a single antenna. The short 60GHz wavelength provides high resolution and marks out the DreamHAT+ as a Raspberry Pi add-on most suited to personal radar and movement alert uses, such as detecting humans and perhaps pets.

Manchester Robotics Puzzlebot

UK | rpimag.co/puzzlebot

Foundational robotics and real-world applications come together in this Raspberry Pi edition of Manchester Robotics' Puzzlebot. Beginning as a project at The University of Manchester, MCR2 is on a mission to address the growing demand for skilled robots and aerospace engineers while also democratising the industry by widening access to professional roles and apprenticeships. In the past six years, MCR2 has provided more than 3000 Puzzlebots to enthusiastic individuals keen to get a foot in the door of the robotics profession. With a proud academic heritage and many educational papers to its name, MCR2 aims to encourage wannabe roboteers from beginner to engineer, fostering a love of making, empowering learners to "create, innovate, and solve real-world problems [thereby shaping] the workforce of tomorrow". The collective describes the Raspberry Pi 5 version of Puzzlebot as the ultimate platform for robotics education and experimentation, offering unmatched versatility and computational power in an affordable and accessible package.





Wink digital signage

Latin America | winkdigital.io

Digital signage brings your messaging to life, whether it's promoting a community event, displaying the menu at your coffee shop or greeting visitors as they arrive at your offices. Dynamic advertising is that bit more eye-catching with vibrant images. Wink is an innovative platform for efficient content and device management, designed to easily integrate with any type of digital display. It's flexible, user-friendly, and optimised to simplify businesses' technology management. Wink Digital primarily focuses on the Latin American market, offering touchscreen displays, devising video walls, and developing digital content controlled by Raspberry Pi.

Sprosskraft Grow Manager Kit

Germany | sprosskraft.de

Raspberry Pi environmental sensor kits are ideal for keeping tabs on whatever you wish to grow, enabling you to check in on plants' health remotely as well as in person and learn whether they need more or less watering. Germany's Sprosskraft Grow Manager Kit features lighting as well as moisture sensors, so you can create artificial daytimes or use lights to extend the length of each day's growing time. The indoor growing kit uses the MQTT (Message Queuing Telemetry Transport) protocol for messaging in a modular IoT setup to which additional sensors can be added. Everything is stored on a local server, so checking in on your plant's growth spurts does not incur potentially costly cloud storage fees. Sprosskraft's smart grow-box has moisture and light settings that can be easily customised to the crop you're growing, and it provides reassuring updates and alerts via an online portal or custom app. The Sprosskraft Grow Manager Kit works with Raspberry Pi 5 and Zero 2 W.

Hellbender

USA | hellbender.com

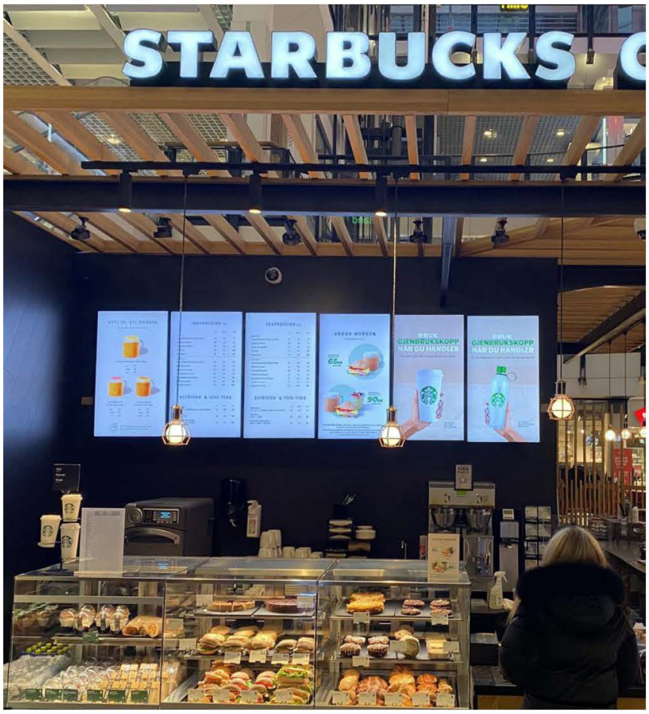
Hellbender boasts several Raspberry Pi and RP-based products, including its $49 landmark RP2350A Development Board. Designed to be a general-purpose microcontroller and sensor board ready to be integrated into any project, this board is supported through the Raspberry Pi Pico SDK.

They also recently launched the Robotic Navigation System (RNS) Dev Kit that utilises Raspberry Pi and NVIDIA NX hardware. Hellbender has a specific focus on developing computer vision products. Its AITRIOS Workmate Camera is a forward-looking camera based around a 1.5GHz Raspberry Pi Compute Module 4 and the Sony AITRIOS IMX500 (the same inference chip found in Raspberry Pi's AI Camera).

SignageOS

France | signageos.io

France's SignageOS seeks to be the ultimate in digital displays, helping businesses "create mission-critical networks at scale". Rather than offering a one-size-fits-all approach to digital signage, the company prefers to help suggest bespoke setups that enhance the business they promote and its aims. SignageOS supports more than 100,000 different connected Internet of Things devices, across 50-plus hardware platforms with both cloud-based control consoles and DevSpace HTML5 app environments. This all but guarantees compatibility and a fast route to market to create user-friendly apps to control your suite of digital displays in real-time. There is no more hassle with hardware compatibility of your code and a guarantee that screens and displays of all types can be centrally controlled from "the single pane of glass you've been waiting for".



ONLY THE **BEST**

Raspberry Pi Pico add-ons

By **Phil King**

Raspberry Pi Pico is a range of microcontroller development boards with a small footprint and low power drain, making them ideal for embedding in digital projects that don't require the superior processing power of a single-board computer. It's hardly surprising that Pico boards are so popular with makers and have been used in countless projects – we highlighted some of our favourites in issue 154's 'Ten amazing…' section (**rpimag.co/154**).

Since the original RP2040-powered Pico model launched in January 2021, it has been joined by Pico W, adding Wi-Fi and Bluetooth connectivity, and then the Pico 2 and 2 W, based on a new RP2350 chip with extra processing power. There are also Pico versions with GPIO headers pre-soldered.

Even better, there's a large range of add-ons and accessories available for Raspberry Pi Pico. We've picked out some of the best here. Although Pico is typically used in a headless setup, some add-ons enable you to output audio and view a display, while others make it easier to break out the GPIO pins for electronics prototyping and robotics. Check out our Pico picks…

Pico Display Pack

Pimoroni | £19 / $21 | pimoroni.com

There are adapters that enable Pico to output video to a monitor, but for most projects you'll want a much smaller display. While it's relatively straightforward to connect the relevant GPIO pins to an I2C or SPI mini LCD, Pimoroni's Display Pack is a much neater solution with no wiring required: just plug your Pico into the female headers on its rear.

Three screen sizes are available: the original slimline 1.14-inch (240×135 pixels), along with the later 2.0-inch and 2.8-inch models (both 320×240). All feature four tactile buttons, an RGB LED, and an IPS LCD that looks very sharp with vibrant colour.

Another major plus is that Pimoroni has created C/C++ and MicroPython libraries, along with a few example programs such as bouncing balls, rainbows, scrolling menus, star field, and thermometer. The simplest way to get started is to flash the custom MicroPython build to Pico and start playing.

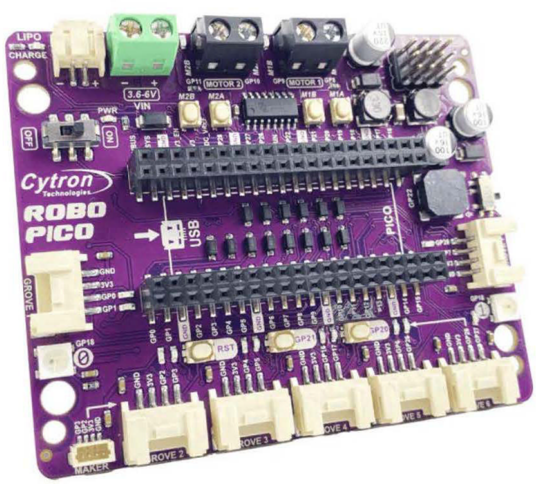
▲ The Display Pack comes in three sizes; this is the 2.8-inch version

Verdict

Add a vibrant mini colour display to your Pico projects.

Robo Pico

Cytron / The Pi Hut | £15 / $20 | cytron.io / thepihut.com



This feature-packed robotics board includes screw terminals to connect two DC motors (or single stepper motor), and a twelve-pin header for up to four servos. Cytron has crammed in no fewer than seven Grove connectors (for sensors, mini OLED, etc.), along with a 'maker' JST-SH port. Add to that two on-board RGB LEDs, a reset button, two user buttons, and a piezo buzzer (with mute switch).

A nice touch is the presence of four test buttons for the motors, plus status LEDs for these and 13 GPIO pins – easily accessible via extra female headers next to Pico. There are three ways to supply power: via Pico's USB port, a terminal block for voltage in (3.6–6V), and a two-pin port for a single-cell LiPo battery pack. There's even a handy power switch.

While you can code in MicroPython or C/C++, Cytron has provided an excellent multi-part guide using CircuitPython.

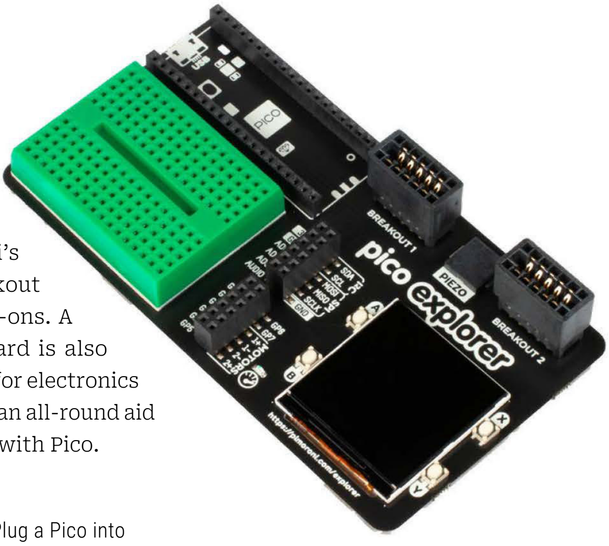
◄ Connections, buttons, and LEDs galore on this impressive add-on

Verdict

Crammed with useful features, it's a versatile robotics board.

Explorer Base

Pimoroni | £25 / $28 | pimoroni.com



▲ Plug a Pico into the female headers and start exploring

This versatile Pico breakout board even features a mini colour LCD screen (240×240 pixels), ideal for viewing/charting data from sensors, surrounded by four tactile buttons. Alongside the display are clearly labelled female headers (24 pins in all) that break out a selection of Pico's GPIO pins, including I2C, SPI, and ADC. An audio pin is connected to an on-board piezo speaker.

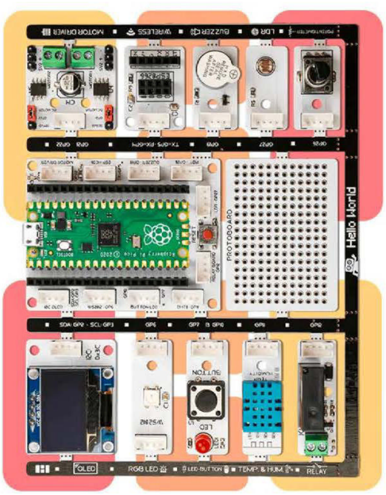
A major bonus is the inclusion of four motor pins linked to the built-in DRV8833 dual H-bridge motor driver chip, which can deliver 1.5A RMS current output to control two DC motors. In addition, there are two slots for Pimoroni's extensive range of Breakout Garden sensors and add-ons. A mini 170-point breadboard is also crammed onto the board for electronics prototyping, making this an all-round aid for exploring electronics with Pico.

Verdict

A user-friendly way to experiment with electronics prototyping.

PicoBricks Base Kit

PicoBricks | £52 / $70 | picobricks.com



Aimed mainly at STEM educators, but also great for newbies to physical computing, this kit eliminates hurdles such as jumper cable clutter and soldering. It comes with a Pico W slotted into the headers and surrounded by an array of ten electronics modules, including sensors, relay switch, motor driver, potentiometer, RGB LED, and mini OLED screen. Interestingly, these modules are already connected to Pico, but can be snapped off and wired to it with Grove cables if you prefer. In addition, there's a protoboard area and separate mini breadboard and jumpers.

Perhaps the best part is the collection of e-books which guide learners through using the modules in electronics projects, using either MicroBlocks or BricksIDE block-based coding. A stripped-down Main Board version of the kit (excluding Pico W and some extras) is also available, as well as a more extensive IoT Expert kit.

Verdict

A neat module-based electronics learning platform, ideal for youngsters.

◀ The modules can be snapped off if you prefer

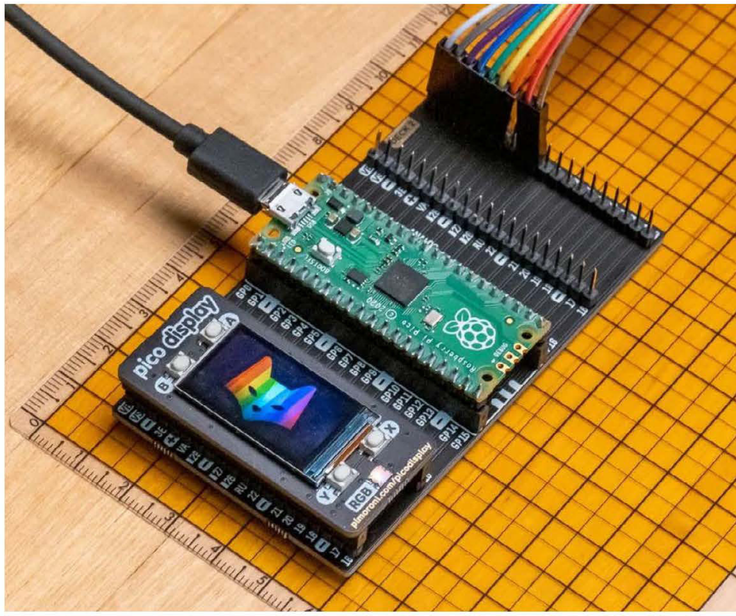
Omnibus

Pimoroni | £8 / $9 | pimoroni.com

Why use just one Pico add-on when you can use two? That's where the Omnibus comes in, expanding Pico's GPIO headers into two 'decks' with 40 male pins each, all clearly labelled. So, with Pico inserted in the middle, you could have a Display Pack on one side and either another add-on board on the other side or use jumper wires to connect other electronics.

One minor issue is that it's designed with Pimoroni's range of Pico Packs in mind, which have Pico mounted on their rear. For other boards, to avoid them being mounted upside down, you may need to use male-female jumper wires to connect them.

Another thing to consider is possible data pin conflicts between two boards, so you need to check which pins they use – even more so if using Pimoroni's expanded Pico Decker (£13 / $15) with its four decks.



▲ Expanding Pico's GPIO headers into two 'decks'

Verdict

Use more than one add-on with Pico, but watch out for pin conflicts.

2.13-inch E-Ink E-Paper Display Module

Waveshare / The Pi Hut | £16 / $22 | waveshare.com / thepihut.com

▲ Show data and simple graphics on this monochrome e-ink display

Verdict

A simple black-and-white display with a very low power drain.

If you just need a small screen to show data periodically, an e-ink display is ideal. The main advantage is that it doesn't use any power to keep showing the same info, only when changing what's shown on the screen. So it's perfect for battery-powered IoT projects.

There's a range of Waveshare e-ink displays for Pico, varying in size – all the way up to 7.5 inches. Some have a limited colour palette. This 2.13-inch model is purely black and white, but that does mean its screen refresh is faster (about two seconds) than on colour e-ink displays. It also supports partial refresh, which is useful for updating certain digits/text on screen.

Pico plugs into the female headers on the rear of the display and communicates via SPI. To get started, you can download Waveshare's C or MicroPython demo code to see how to show text and draw lines and rectangles.

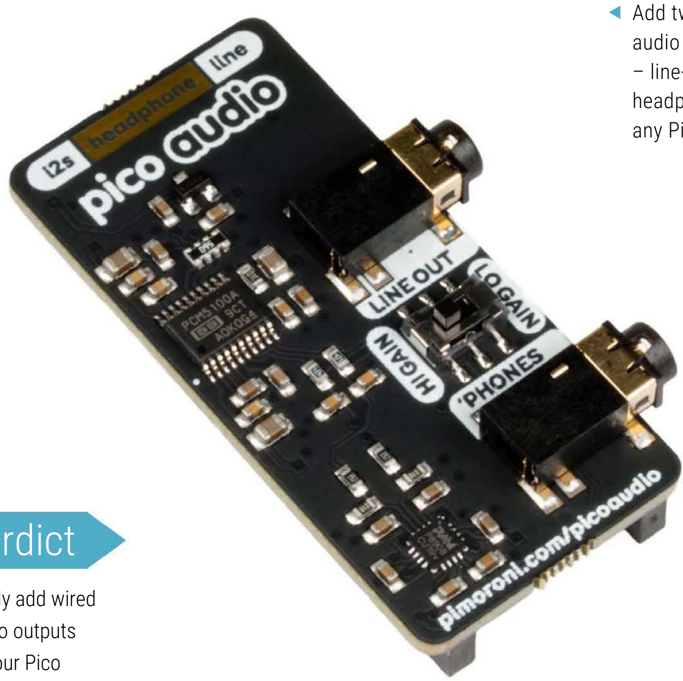
Pico Audio Pack

Pimoroni | £15 / $17 | pimoroni.com

While it is now possible to get Raspberry Pi Pico W to output audio wirelessly via Bluetooth, Pico boards lack a wired audio port. One of the first Pico accessories to appear on the market, the Audio Pack solves the problem. It provides not one but two standard 3.5mm jack outputs: line-out (for external amp or powered speakers) and headphones.

Crucially, it delivers high-quality audio: up to 32-bit, 384kHz stereo sound through the line-out connector, thanks to the on-board PCM5100A DAC. It also packs a PAM8908JER mini amplifier to output a louder signal from the headphone jack – switchable between high or low gain. A MicroPython audio player example is now available (**rpimag.co/audiopackmp**) if you're not au fait with C/C++.

The I2S audio protocol only uses a handful of Pico's GPIO pins, although you will need an Omnibus or similar expander to break out the rest.



Add two wired audio outputs – line-out and headphones – to any Pico

Verdict

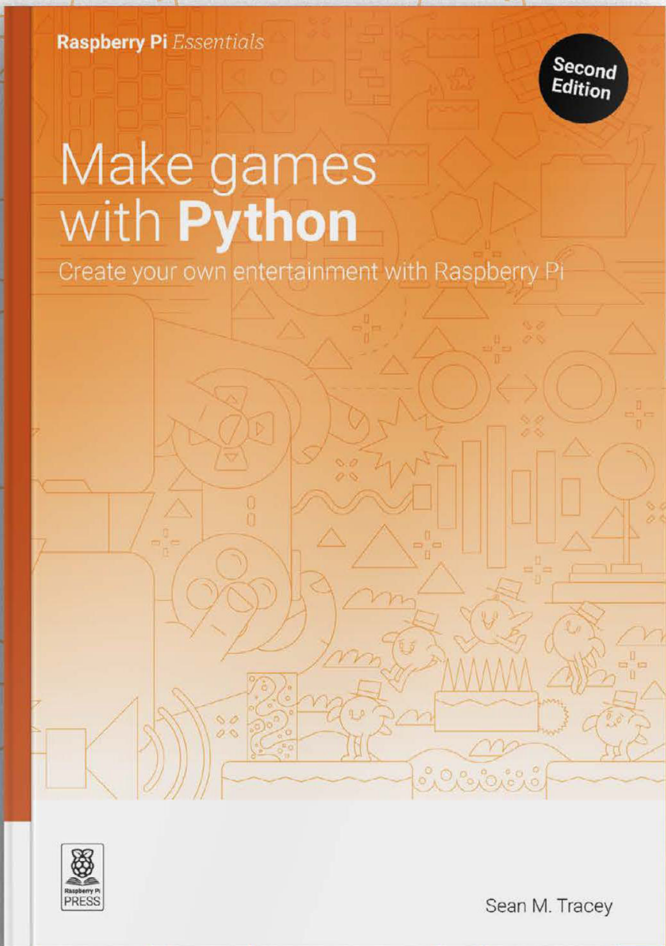
Easily add wired audio outputs to your Pico projects with this board.

DEBUG PROBE

Raspberry Pi | £12 / $12 | raspberrypi.com

Prefer to do C/C++ or bare-metal coding on Pico? Debugging can be a pain. Step forward the Debug Probe. Just connect it to Pico's SWD (Serial Wire Debug) header – or JST port on Pico H – and UART pins, then both devices via USB to a computer running OpenOCD and GDB debugging software. See the documentation at **rpimag.co/debugprobedoc**.

▶ Helping to debug your Pico projects



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

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

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

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

Argon ONE UP

Compute Module 5-based laptop offers open-source, modular computing. By **Lucy Hattersley**

Argon   **rpimag.co/argonupks** $400 (TBC)

SPECS

INPUT/OUTPUT
14-inch IPS display (1920×1200, 60Hz, 250 nits), an integrated 1080p front camera, and stereo speakers

STORAGE
M.2 SSD (PCIe 2.0 ×1), microSD

CONNECTIVITY
Wi-Fi 5, Bluetooth 5.0, 2 × USB 3.1, 2 × USB Type-C, HDMI 2.0 (4K@60Hz output), 3.5mm audio jack

DIMENSIONS
Aluminium alloy chassis, 265 × 233 × 26.3mm, 1300g

NOTE!
PREVIEW

This article is a preview based on pre-release hardware. We will review the Argon ONE Up in a later edition.

▲ The Argon ONE Up looks like any other laptop, but is modular and contains Compute Module 5

Raspberry Pi-based laptops are a cause for interest and excitement for the editorial team. The dream of using a laptop built around our favourite computer is hard to resist.

Argon's ONE Up is the latest, and most convincing, attempt to build a working laptop around Raspberry Pi. Previous efforts included pi-top [4], a modular solution that clips Raspberry Pi in a chunky case to the back of the display that also clips to a well-rounded robot for educational environments; and CrowPi2 and CrowPi L, which are both chunky, but great for learning electronics.

What we have here is the real deal. Argon ONE Up is powered by a Raspberry Pi Compute Module 5 (CM5) board. CM5 packs all the functionality and power of

▲ A GPIO module connects to two USB-C ports on the side of Argon ONE Up to provide breakout access for electronics projects and HAT-based hardware

◄ A flap on the rear of the device provides access to the Compute Module 5 'brain' and the upgradable M.2 NVMe storage

> *A dream machine for makers. It offers a durable, high-quality, upgradable and repairable open-source laptop built around Arm hardware and running Linux*

Raspberry Pi 5 into a smaller board for industrial and embedded applications.

Flip Argon ONE Up over and underneath you'll find a flap that can be unscrewed. This offers access to the CM5 port and an M.2 NVMe (Non-Volatile Memory Express) for storage.

This is a dream machine for makers. It offers a durable, high-quality, upgradable and repairable open-source laptop built around Arm hardware and running Linux. Excited? We certainly are.

More than a case?

In some respects, Argon ONE Up sits in the same family as other Argon products we have tested, such as Argon ONE V3, or Argon NEO 5. These cases have consistently impressed the team with good build quality, high-quality materials, excellent cooling solutions, and expanded features (such as full-size HDMI ports, IR sensors, and SSD support).

It's clear that this product is much more ambitious. While the Argon ONE Up is technically a case insofar as it houses a Raspberry Pi, it goes above and beyond to provide a "true laptop experience".

Argon ONE Up packs a 14-inch 1920×1200, 60Hz, 250 nits display with an integrated 1080p front camera and stereo speakers, along with a keyboard, trackpad, and array of input/output ports.

The case is made from an aluminium alloy chassis measuring 265×233×26.3mm and it weighs in at a scant 1300g. The end result is, visually at least, the same as a MacBook or Microsoft Surface laptop.

It has a scissor-switch, chiclet-style keyboard in a 75% layout (that is, with

▼ The Argon ONE Up in development. It looks just like any other laptop, but is packed with Raspberry Pi power

▼ Inside Argon ONE Up, we see Compute Module 5 and an internal cooling fan

function keys and an extra column on the right). There is also a dedicated Raspberry Pi button and backlit functionality. It's better than you think. It's much better than it needed to be: a pleasure to type on and stylish to use. Up there with a major brand offering. Well done, Argon.

Pre-production

What we are looking at is a pre-production model here (it is a second test run from the production line). It is not a finished version, and our laptop had limitations, which is why this is a preview instead of a full review.

On our test model, the backlit keyboard works, but the function key modifiers do not. The volume control does not work, although the speakers sound fine at the full level. We couldn't get the keyboard or trackpad to work with Ubuntu. The web camera was not working.

We will perform a full review when we get to it, but expect these elements to be fixed in a final version. The things that fail on our test unit strike us as production run errors rather than fundamental flaws.

▲ Argon ONE UP with the rear of the case removed

▲ An M.2 NVMe drive providing storage

Open and upgradable

Obviously, we like this laptop because it is built around Raspberry Pi hardware, and especially because it is an Arm-based CPU. This makes it more energy-efficient and it runs cooler and faster than Intel-based laptop hardware.

There are other Arm-based laptops around, including Apple's M-series and Microsoft's recent Surface models. While these have charms, they also have locked-down operating systems that are vastly more expensive than Argon's solution. They lock you into macOS and Windows, whereas Argon ONE Up enables you to run a variety of Linux-based operating systems.

More importantly, Argon ONE Up is a modular computer that is both upgradable and repairable. It's designed for a Compute Module 5 acting as a mainboard. This can be swapped out if needed and future Compute Module hardware can be introduced if it matches the same footprint as Compute Module 4 and 5.

The Compute Module 5 is replaceable, so you could upgrade from a 4GB, to 8GB or 16GB RAM model. The M.2 NVMe storage drive can be replaced over time. You can replace the 54Wh battery. There is a fan inside that can be cleaned (although we found it rarely turned on thanks to the effective heatsink properties of the case).

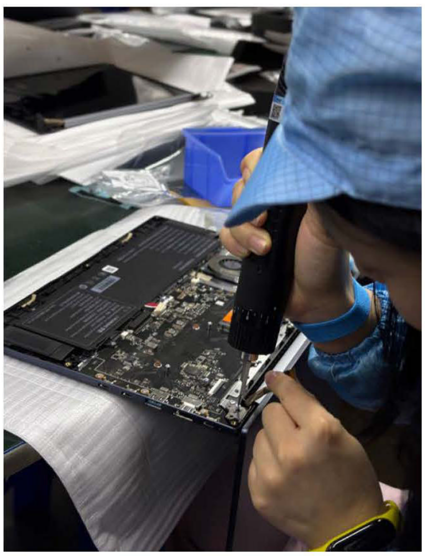
The only part that is sealed and non-replaceable is the keyboard and trackpad. Argon tells us that it is ultrasonic-welded to the case "in order to preserve the slim profile."

GPIO and expansion

This wouldn't be a Raspberry Pi product if you couldn't access the GPIO (General Purpose Input/Output) pins. Being able to program and control electronic components is to us an inherent part of the digital making process.

Argon has a clever solution that uses two USB-C ports on the left side of the Argon Up laptop and a dongle to provide GPIO access. The dongle has an added power button to disconnect power to the 5V and 3.3V connections on the GPIO.

As you'd expect, when the GPIO module is not connected, the two ports function as standard USB-C ports.



▲ Argon ONE Up pre-production models being tested on the factory floor

Pricing it up

The target price for the base unit will be $400 without the CM5 module, power supply, or M.2 NVMe storage. An early Kickstarter is available with a pre-release price of $350.

The laptop can be charged with any 45W power supply, but Argon recommends the official Raspberry Pi 45W Power Supply (**rpimag.co/45w**). Add another $88/£50 for a Compute Module 5 with no eMMC, and approximately $35 for an NVMe drive, and you are looking at around $500 for a working laptop.

We can't wait to test out the finished working product

This puts it in the Chromebook OS territory, but it's a far more capable laptop than any Chromebook we've seen. And frankly, this is much better than a $400 laptop needed to be. We're keeping a close eye on this one.

It's rare that we preview a product like this, preferring instead to stick with a review. And we will be getting a final production version of Argon ONE Up in a following edition. The equipment we are testing is very much a work in progress, but it's good work so far and on a great track. We can't wait to test out the finished working product.

▲ Compute Module 5 provides the heart of this laptop

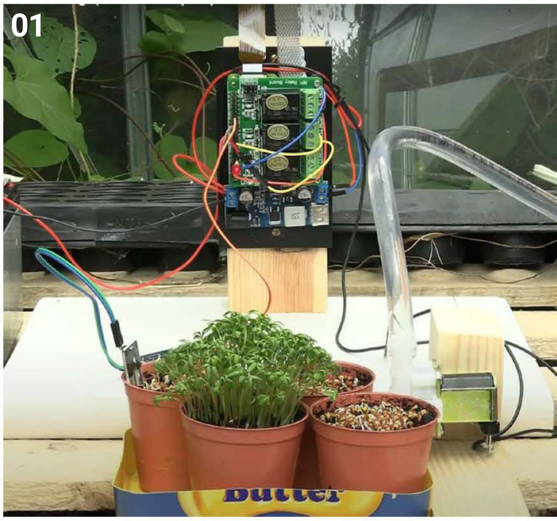
◀ The Argon ONE up is upgradable and repairable thanks to its modular components

10 amazing:

Raspberry Pi projects in the garden

Raspberry Pi is green, and so is grass. Coincidence? Yes

We're British, so we do love a nice well-kept garden, or back yard as you might call it in other English dialects. We're also fans of automation and data, so using a Raspberry Pi to help improve our gardening is an exciting prospect. Here's just a few excellent ways to do so...



01. Watering and time-lapse

See it grow

rpimag.co/waterlapse

This very simple project uses a moisture sensor to make sure the plant has enough water. And a camera for the time-lapse video.

02. Szerafin mushroom farm

Fungi optimisation

rpimag.co/szerafin

Temperature and humidity are measured by this system to keep the environment perfect for the mushrooms to grow.

03. Freight farms

Professional hydroponics

rpimag.co/freightfarms

There have been a few Raspberry Pi farm ideas in the past, but none have been created to work in an actual shipping container at such high volume.

04. Pollination monitor

Track the bees

rpimag.co/pollination

How do you know your strawberries have pollinated? Using Raspberry Pi and computer vision to track bees visiting your plants, of course.

05. Remote agricultural monitoring

Tracking fruit flies

rpimag.co/remoteagri

Invasive fruit flies can destroy crops if left unchecked, This Raspberry Pi trap sees what insects are showing up in case action is needed.

06. Open Weed Locator

Save on spray

rpimag.co/openweed

A large method of checking weeds in large areas of greenage using computer vision and many Raspberry Pi boards attached to a weed-spraying robot.

07. PiMowBot

Robotic lawn trimming

rpimag.co/pimowbot

Roombas are well-established robots that hoover your house. What if there was an equivalent for your grass? Meet PiMowBot. But be careful of its spinning blades.

08. Hydroponic garden

Grow at home

rpimag.co/hydroponics

Hydroponics is a fancy word for just using nutrient-filled water to grow plants, and is great for growing your own veggies using Raspberry Pi automation.

09. MudPi

Smart watering system

rpimag.co/mudpi

Using various sensors and automated irrigation systems, you too can keep your garden perfectly watered all year round with MudPi.

10. Garden Lighting

Flower bed illumination

rpimag.co/dmxlight

Using a Pico and a garden bollard, you can easily create custom lights to make your garden look very pretty during the moonlit hours.

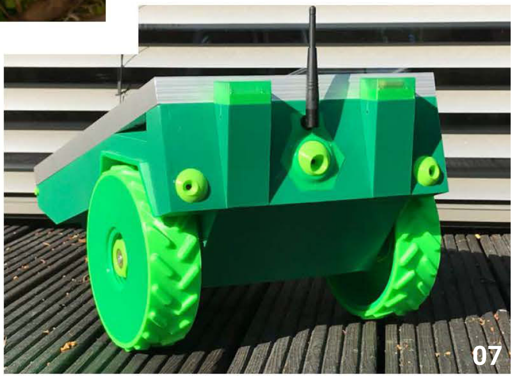

02

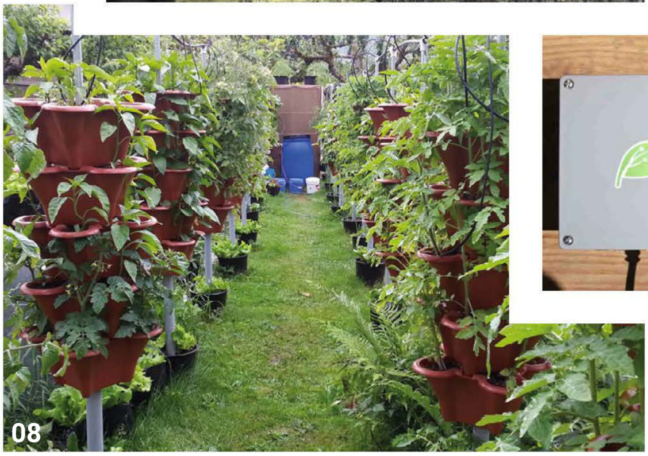

03
©Freight Farms

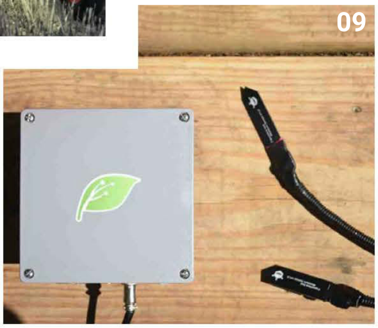

04


05


06


07


08


09


10

Jo Hinchliffe

The concretedog is a long-time maker who loves soldering, rocketry, and much more

- **Name** Jo Hinchliffe
- **Occupation** Maker and writer
- **Community role** Community development and relations
- **URL** mastodon.social/@concretedog

You may have seen his name in the magazine, but Jo Hinchliffe is also a prolific maker outside these pages under his alter ego concretedog, and making is a lifelong hobby for the brickcanine.

"I've always made stuff," Jo recalls. "In a way, being shown how to solder as a kid was a gateway. I realised at first I could fix cables and things, and then over time I realised I could build electronics kits. Then I got into the idea that if I understood a bit about how components work, I could probably build my own projects. I've kind of done similar in a lot of subject areas now… rockets, robots, PCB design, boatbuilding, and more."

When did you learn about Raspberry Pi?

I remember posting about that original Eben Upton video about Raspberry Pi before they were released in 2011. Then I recall pressing **F5** a lot one morning when the first 10,000 units were released. I got one and I believe I actually created the first ever blog post posted on a Raspberry Pi: **rpimag.co/blogpi**.

How did you start writing in *HackSpace*?

So here's a story that shows that sometimes it is absolutely about being in the right place at the right time! I'd seen *HackSpace Magazine*, and in fact I'd met Ben Everard and Andrew Gregory in passing at a Guild of Makers event. A short while later, I was attending a Midland Rocketry club event in the middle of nowhere, north of Birmingham. I was there to fly some rockets, but I spotted Ben Everard looking a little bit lost with a small *HackSpace Magazine*-branded rocket in his hands. I went up and reintroduced myself and sort of looked after him that weekend, introducing him to people and the systems of a launch event so he could launch the *HackSpace Magazine* rocket.

▼ An ingenious way to wind up your carbon filament with a Raspberry Pi

He explained that *HackSpace Magazine* wanted to do a rocket-themed issue (**rpimag.co/hs12**). I told him I'd written a tutorial for the United Kingdom Rocketry Association about designing and simulating a rocket with an open-source bit of software called OpenRocket and that, as it was introducing people to a community I love, the magazine could republish it for free. I then also slipped in that I could do other stuff and write about it… and I've been pitching for the magazine(s) ever since!

What's your favourite thing you've made for the magazine?

Favourite things I've made is a hard one! I loved making my small rowing boat for *HackSpace Magazine* (**rpimag.co/hs36**), that was awesome and a build I'd wanted to do for a long time. Likewise, I enjoyed flexing my design muscles and building the Carbon-Fibre Filament Winder project (**rpimag.co/150**). If I can be annoying and slightly change the question, though, to "what are the favourite things you've written for the magazines", I loved writing the FreeCAD series which then became the *FreeCAD for Makers* book download (**rpimag.co/freecadbook**). There's also a really small one-off article I wrote that I absolutely adored researching and writing. It's nowhere near as spectacular as most of the stuff I've built or written about, but I loved it as a super-geeky subject… It's a 'Knots In Spaaace' piece which looked at the largely overlooked art of cable lacing and why you can still see it on high-resolution photos of rovers on Mars! [See it in *HackSpace* issue 43, **rpimag.co/hs43**.]

Knots in space – the art of cable lacing

FEATURE

The added bonus is that every time we lace a cable, we think of space exploration

Figure 4 Using some scrap cables, we spot-tied them using the NASA-compliant spot tie knot in the E-Glass/PTFE lacing tape

Figure 6

aficionados, which of these is best, is a debate that runs deep! The square knot is formed by taking the running end of our cord coming from the clove hitch, and bringing it over to form a loop, and then our tie-off end goes under the running end, and then over and through the loop we formed (**Figure 7**).

Cinch that half of the square knot tight, and then

SPACE SPACING

Spacing between spot ties, running lock stitches, or cable ties is specified in the NASA standards, and is linked to the diameter of the cable bundle. For example, for a cable bundle of 6.4 mm or less, the stitches, however formed, should be no more than 19.1 mm apart. Moving to larger diameters and the spacing increases. For example, a 25.4 mm bundle spacing will be a maximum of 50.8 mm. For any cable bundle diameter over 25.4 mm, the maximum spacing has an upper limit of 76.2 mm.

▲ We're not sure we covered these knots in cub scouts

What's your favourite thing you've made with Raspberry Pi?

Again, hard to choose just one, but I am going to go with a classic: building a SatNOGS ground station. We did a huge feature on the Libre Space Foundation (LSF) in issue 18 of *HackSpace Magazine* (**rpimag.co/hs18**) and at the time I was a core contributor to LSF. Within the article collection in that issue, I built the simplest version of a SatNOGS ground station and connected it to the global network and then recorded some satellite passes, scheduled via the website. I even caught a conversation from the International Space Station.

How I Made

'CYSGOD' – A DIY PLYWOOD BOAT

Starting my boatbuilding journey

▲ A plywood boat is more stable than you'd think

> *I believe I actually created the first ever blog post posted on a Raspberry Pi*

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. Hmm, we reckon an automated food dispenser

02. It's been a while since we made a Pico drone ourselves…

03. It's always amazing to see what people are doing with RP2350

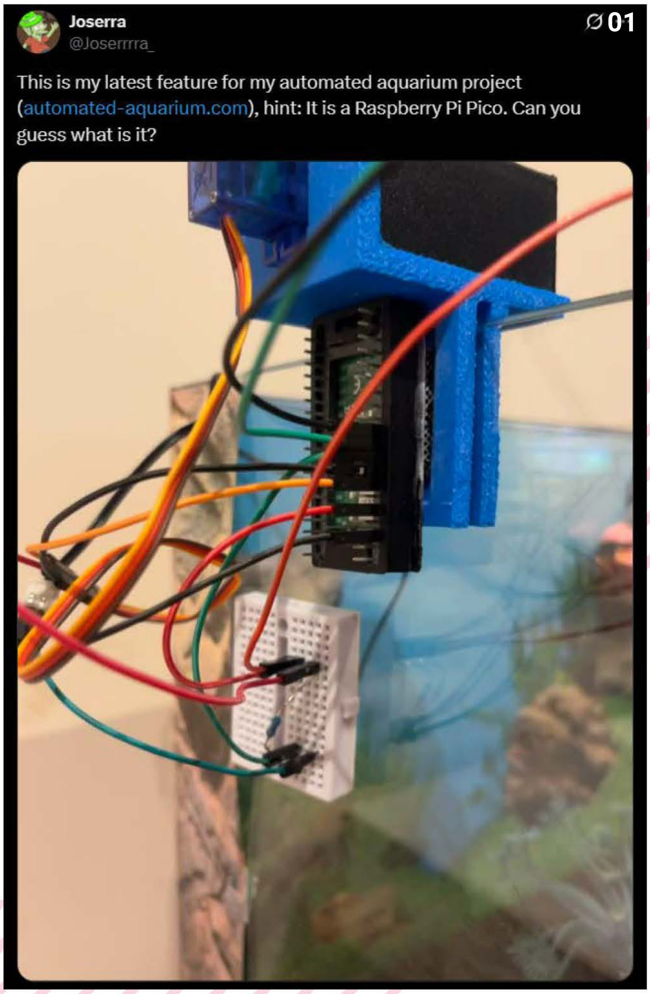
04. A very cool medical project that could truly help people!

05. This is a great way to learn Pico coding

06. We wonder if that vessel is seaworthy

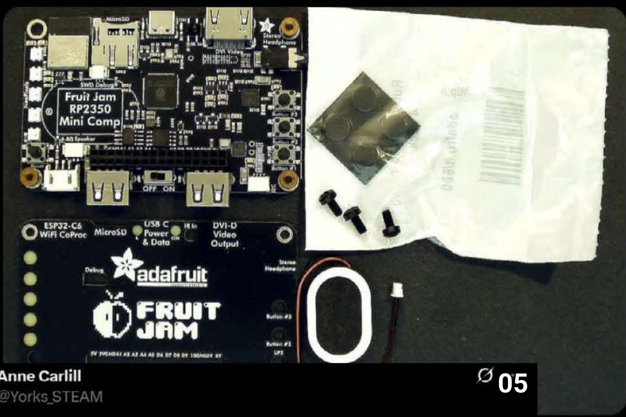
07. It's great to see yet another model railway powered up with a Raspberry Pi

08. Seeing a replica WOPR from the classic movie *WarGames* always makes us happy



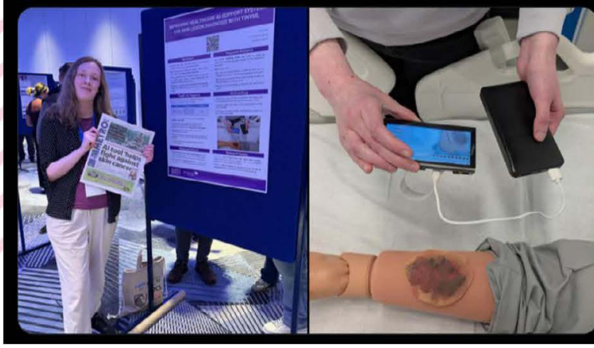
Anne @anne engineer — 03

With the release of the @adafruit Fruit Jam, I and others are making games, emulators and a whole "operating system" for this versatile RP2350B-based board. Much fun!



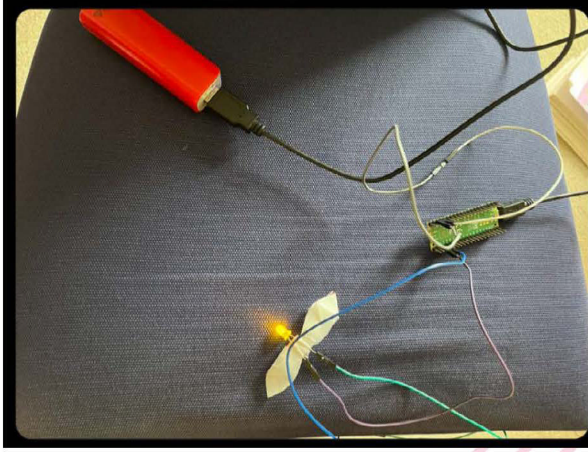
Tess Watt @TessWOfficial — 04

My PhD project uses Raspberry Pis to scan skin lesions in remote areas



Anne Carlill @Yorks_STEAM — 05

Just doing one of the #CodeClub activities, Pico Firefly, but it took me a long time to get the PicoZero code onto my Pico so I'm pleased I've completed it at last. #MakerMonday



Simon Prickett | simonprickett.dev @simon_prickett — 06

I will be playing with this ultrasonic distance sensor, a Pico WH and MicroPython.

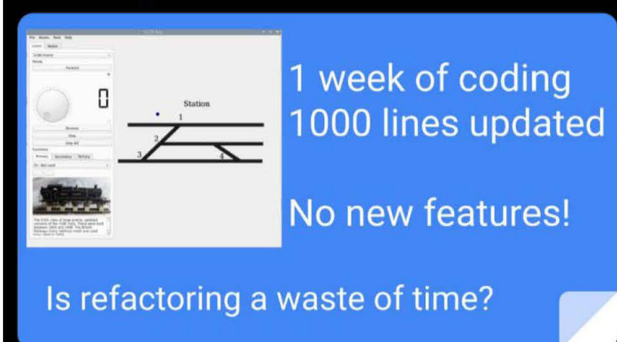
Stewart Watkiss @stewartwatkiss — 07

Sometimes you need to take a break from adding new features and fix bad code.

I've been refactoring my Raspberry Pi model railway GUI application.

penguintutor.com/projects/pyvlcb

#MakerMonday



Nick Bild @NickBild79 — 08

It's happened. WOPR is back:

Simulate Talking With WOPR From the Movie WarGames

From hackster.io

‘Friends’ – a Raspberry Pi-powered self-balancing sculpture

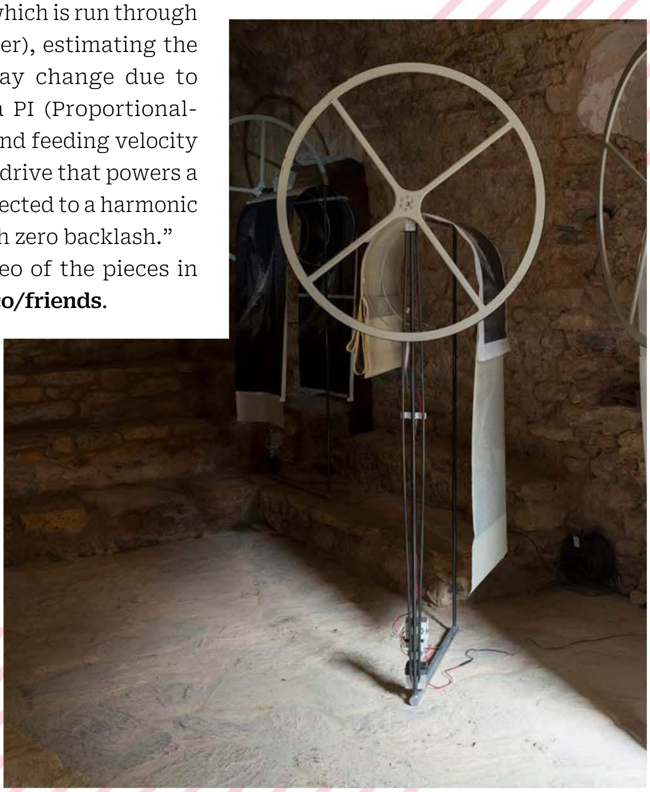
Dr Kevin Rathbone emailed us about a contemporary art project he’s created using Raspberry Pi. Called Friends, this trio of robots was part of an exhibition in Faugères, France called ‘Que Rien Ne Bouge’, which we are reliably told means ‘Let Nothing Move’.

“The artwork consists of three carbon-fibre frameworks that each balance on a single edge using the principle of an inverted pendulum with a reaction wheel,” explains Kevin. “To contrast the dynamic nature of the piece, the frameworks support rice-paper-bearing screen prints of Rodin’s *Étude de robe de chambre pour Balzac*, representing the material stability typically used in traditional stone or bronze sculptures.”

The project was created with Antoine Espinasseau, a French artist based in Brussels. We’ve seen balancing robots before, but not balancing art.

“A Raspberry Pi 3B at the heart of the system runs a C program, interfacing with a custom add-on board,” Kevin says. “The board integrates an IMU, current sensing, a piezo sounder, and a CAN HAT to communicate with a servo drive. The control loop runs at 200Hz, continuously adjusting the balance by reading gyro and accelerometer data (which is run through a complementary filter), estimating the tilt angle (which may change due to a breeze), running a PI (Proportional-Integral) controller, and feeding velocity commands to a servo drive that powers a brushless motor connected to a harmonic drive – a gearbox with zero backlash.”

You can see a video of the pieces in action here: **rpimag.co/friends**.



▶ True to the theme of the exhibition, the frames don’t move

Crowdfund this

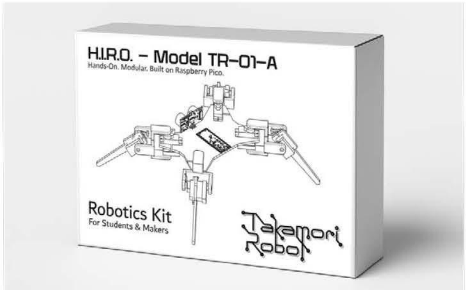
Raspberry Pi projects you can support this month

Argon ONE UP

The Argon ONE UP Kickstarter is in the final stretch of its campaign for this Raspberry Pi (Compute Module 5) powered laptop that we've previewed in this issue. It's probably the most complete Raspberry Pi laptop we've ever seen, designed for both regular computer use and hardware development. Lovely.

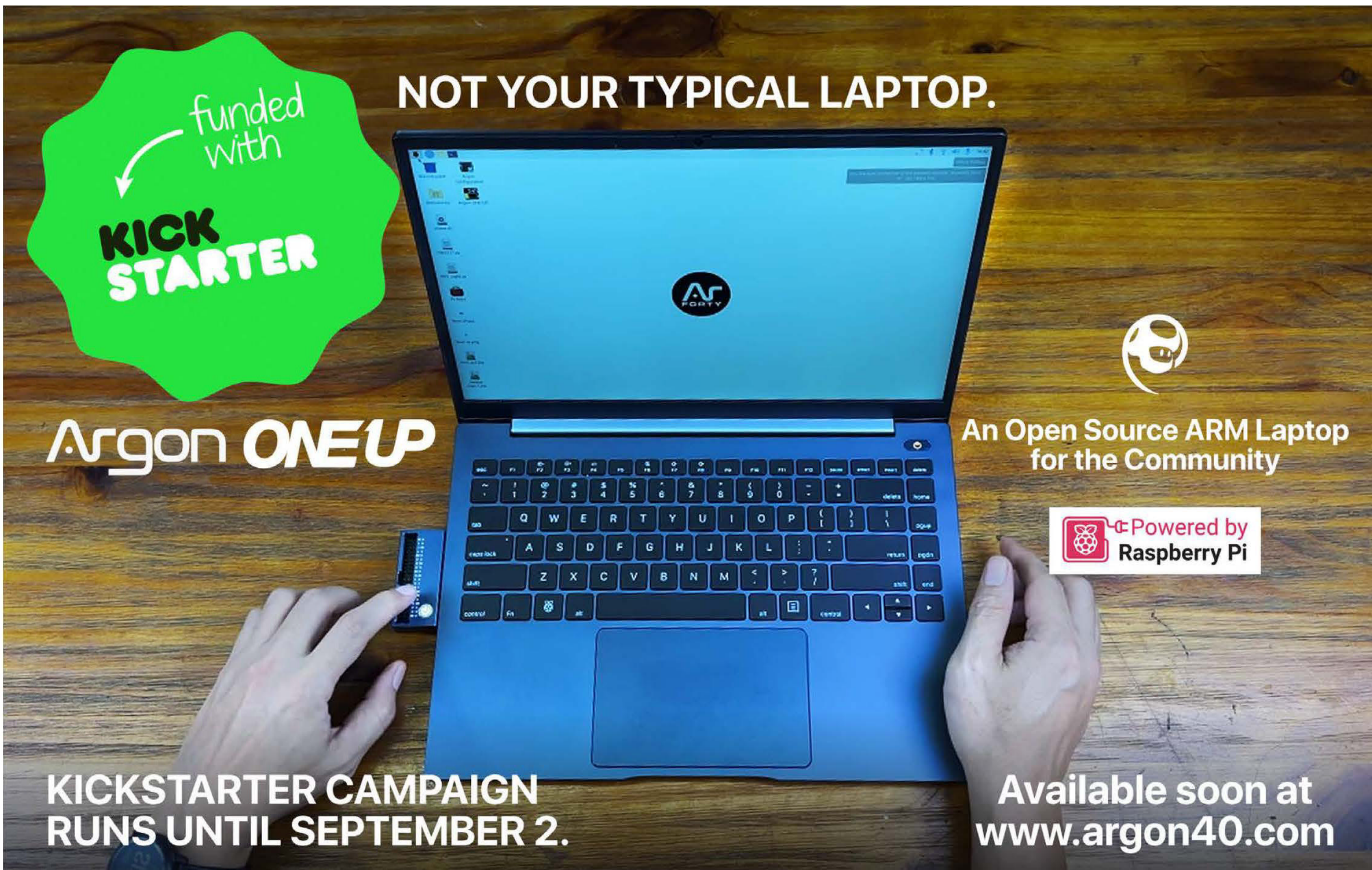
▶ **rpimag.co/oneup**

H.I.R.O.



A modular robot kit for students and makers powered by Raspberry Pi Pico, H.I.R.O. stands for 'Highly Interactive Robotic Organism'. We're pretty sure that's a backronym, but it does come with motors, various sensors, and can be easily built upon.

▶ **rpimag.co/hiro**



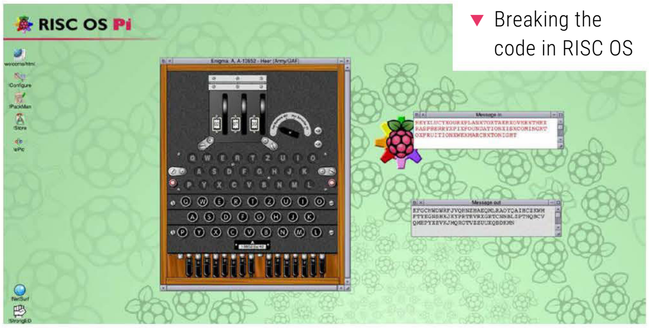
Your Letters

Build an Enigma machine

Your tutorial on using an ancient artefact of computing (RISC OS) to explore an even more ancient artefact (the Enigma machine) blew my mind [in issue 156]. I've always been roughly aware of Britain and Poland's contribution to code-breaking during the war – and Jon Bon Jovi's, if you believe Hollywood. But reading that issue, it clicked with me that the first version of RISC OS was released only barely closer to the present day than it was to the cracking of the Enigma code. They're both relics, but somehow you can run RISC OS on a computer to this day.

Ian, via email

For anyone new to cryptography and the role it played in launching the computer age, we strongly recommend a visit to Bletchley Park. There's a conceptual thread that runs through rotors and keys being pressed, to valves, relays, transistors, stored programs, multi-purpose computing, and today's chips with their billions of transistors, and that thread starts at Bletchley Park. It's a mind-blowing place. And all that history has happened so, so recently – it's all happened just about within living memory, which is something you can't say for any other field of human endeavour. We're still in the foothills, and it's up to all of us to ensure that we take the right path.



Breaking the code in RISC OS

Industrial 3D printing

I learned about the existence of 3D printing in metal a while ago – the person I was talking to had been studying it as part of a PhD program, and they were telling me about using it in shipyards to make propellers for giant container ships. The logic being that for every day these ships are out of action, they lose millions of pounds, so while 3D printing a propeller might not be the best way, if it's the fastest way it quickly becomes the best way.

That was a few years ago, talking to someone about high-level, big business stuff. How amazing that today I can send off a design on the internet and get a 3D-printed metal part in the post a few days later.

Robert, via email

We agree 100%. As long as you have the design right, you can abstract the difficult, dirty and dangerous elements of manufacturing away to give you a complete factory in your laptop – or your Raspberry Pi 5, if you sensibly followed our guide to setting up a desktop machine with Raspberry Pi, which was also in last issue.

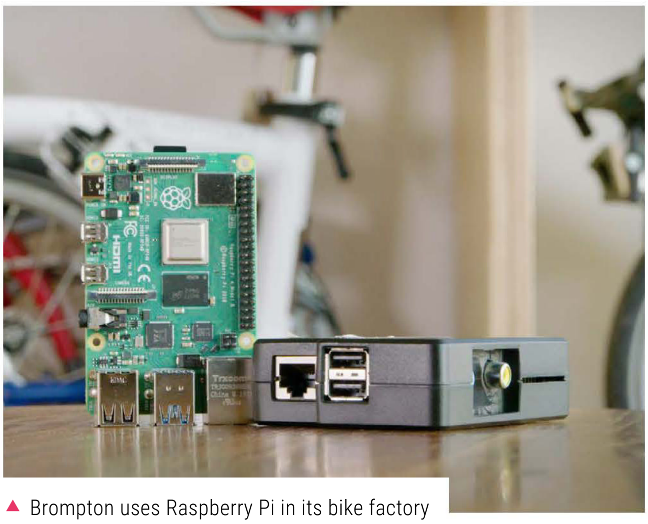
A design 3D-printed in metal

Raspberry Pi in industry

I had to wipe away a tear of patriotic pride after reading that Brompton has been using Raspberry Pi in its factory for the last twelve years. One great British product helping to make another great British product. Wrap me up in a Union Jack and play *God Save the Queen*. Twelve years! Brompton must have been the earliest of early adopters given that the original Model B only came out 13 years ago!

David, via email

We've still not got used to singing *God Save the King* either; years of muscle memory can't simply be turned off like that, so it comes out like 'God Save the Kwing'. But yes, it does give us a hearty glow to see Raspberry Pi out in the wild wherever that might be, whether that's making a lift go up and down, automating a production line, managing inventory, or anything else. If you need a computer that won't break the bank and won't let you down, it's ideal.



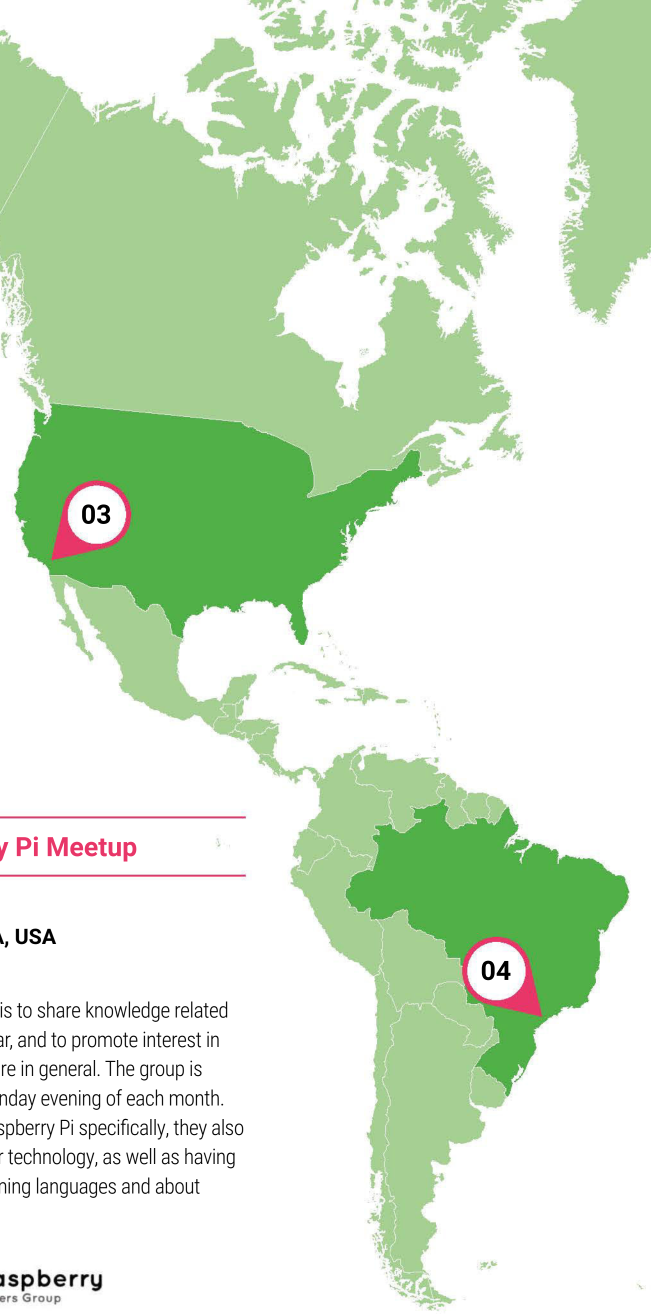
▲ Brompton uses Raspberry Pi in its bike factory

Contact us!

- @rpimagazine
- @rpimag
- rpimag.co/facebook
- magazine@raspberrypi.com
- forums.raspberrypi.com

Community Events Calendar

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



01. Gitex Nigeria 2025

Monday 1 September to Thursday 4 September
Eko Hotel Convention Centre, Lagos, Nigeria
rpimag.co/gitex2025

Raspberry Pi is delighted to be visiting Gitex in Abuja and Lagos, Nigeria, this September. There, you'll be able to meet a member of the team and discover how Raspberry Pi can help you with your own solutions. You'll also be able to discuss how local partners can help with your projects, and hear about the technical, product, and compliance support services Raspberry Pi can offer to industrial customers.

02. Melbourne Raspberry Pi Makers Group Meeting

Sunday 7 September
Docklands Makerspace and Library, Melbourne, Australia
rpimag.co/mrpmgm157

This meetup is open to everyone with an interest in electronics, robotics, home automation, 3D printing, laser cutting, amateur radio, high-altitude balloons, space tech, etc. Makers are invited to bring along their projects and project ideas, and come connect with other makers. Get your questions answered, show off the work you are doing, and get support to resolve nagging issues.

03. Riverside Raspberry Pi Meetup

Monday 8 September
3600 Lime St, Riverside, CA, USA
rpimag.co/rrpm157

The purpose of Riverside Raspberry is to share knowledge related to Raspberry Pi hardware in particular, and to promote interest in tech development in the Inland Empire in general. The group is currently meeting on the second Monday evening of each month.

While the group is focused on Raspberry Pi specifically, they also cover topics about all kinds of maker technology, as well as having discussions about various programming languages and about electronics in general.

Riverside Raspberry
Raspberry Pi Users Group
Let your computer play in the real world!
Meetup
www.meetup.com/Riverside-Raspberry

04. FIEE 2025

Tuesday 9 September to Friday 12 September
São Paulo Expo, São Paulo, Brazil
rpimag.co/fiee2025

Newark, a Raspberry Pi Approved Reseller with local support in Brazil, invites you to discover the power of Raspberry Pi solutions designed for automation, control, and industrial connectivity.

FULL CALENDAR
Get a full list of upcoming community events here:
rpimag.co/events

05. Electronica India 2025

- **Wednesday 17 September to Friday 19 September**
- **Bangalore International Exhibition Centre, Bangalore, India**
- **rpimag.co/elecindia25**

The Raspberry Pi team is delighted to be exhibiting alongside Silverline Electronics at Electronica in Bangalore, India, this September. There, you'll be able to meet the team and experience the full range of technology, including Raspberry Pi single-board computers (up to Raspberry Pi 500), the Raspberry Pi Pico family, and RP2350-based solutions, as well as AI products, cameras, and the latest industrial device: Compute Module 5.

Official Raspberry Pi Event

electronica India 2025
September 17-18-19, 2025
Bangalore International Exhibition Centre (BIEC), Bengaluru

Win 1 of 3 Raspberry Pi 500 Desktop Kits

Raspberry Pi 500 is a fast, powerful computer built into a high-quality keyboard. The Raspberry Pi 500 Desktop Kit comes with a mouse, power supply, and HDMI cable, plus the Raspberry Pi Beginner's Guide, to help you get the most out of your new computer.

Head here to enter:
rpimag.co/win
Learn more:
rpimag.co/500kit

Terms & Conditions

Competition opens on **27 August 2025** and closes on **25 September 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.

Official Magazine
#158

DIY HALLOWEEN

Build a frighteningly good skeleton

On sale 25 September

PLUS!

- Raspberry Pi vs Pico vs Zero
- **x64 and x86 emulation**
- Explore FPGA with Pico

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

Tim Danton, Ben Everard, Rosemary Hattersley, Jo Hinchliffe, Phil King, Andrew Lewis, Ian Osborne, David Plowman, Richard Smedley

Publishing

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Director of Communications
Helen Lynn

CEO
Eben Upton

Distribution

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

Subscriptions

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

ISSN: 2977-4403 (Print)
ISSN: 2977-4411 (Digital)

Ten years at Raspberry Pi

It turns out Rob has spent a decade at the Raspberryiest Place on Earth

Due to the nature of print lag, I am writing this column nearly three weeks before you'll read it. However, if you're reading this on release day of this magazine – or maybe got one a few days early through a print subscription, **rpimag.co/subscribe** – it will have passed 24 August 2025, and that marks ten years of officially being an employee of Raspberry Pi. Pop the champagne!

I've spent that time writing on *The MagPi* and now *Raspberry Pi Official Magazine* – 120 issues' worth, to be exact!

> *I've forged strong bonds and seen how Raspberry Pi has truly helped change the world*

That's roughly 1.5 million words (and probably one million typos – thank you, Phil and Nicola for sub-editing my work into something readable over the years!) over magazines, books, blogs, tweets, etc. I expect my gold watch from Eben shortly.

How things change

I've been writing about Raspberry Pi since I got into the magazine business in 2012, so it's always been a part of my (professional) life, and I've been able to track it from launch to now. Despite that consistency, a lot has happened since then. In the time I've been working here, I've gone through three cars, four offices at Raspberry Pi (RIP 30 Station Road, we hardly knew ye), several laptops, innumerable new Raspberry Pi products since the release of Raspberry Pi 2, a worrying number of Monster energy drinks, I became a hat guy and then just decided to go bald, visited a Disney theme park nearly a dozen times, lived through two Nintendo generations, a pandemic, the fall of the Marvel Cinematic Universe, much political upheaval, and much more – you get the idea.

Through that, I've forged strong bonds with people in and out of the community, and seen how Raspberry Pi has truly helped change the world – both in education and in the hobby circles I was already a part of. And I'm proud I get to be a part of – and document – it.

Changes to come

I feel like I often wax lyrical about the nature of the future when I write these kinds of columns. I do very much like to look forward after all. Right now, that future looks bright – y'all really seem to like the rebrand, which is always nice to see. The things I would love to see stay the same going forward are all the incredible projects I get to see every #MakerMonday on our social media sites, along with maintaining the various relationships I've made, keeping Raspberry Pi computers around the house managing various media and automation, slipping Picos into random projects, and much more. See you next issue!

Rob Zwetsloot – Author
This is the longest job Rob has ever had, although he is just a wee baby at just 38½ years old.

rpimag.co