Listen and talk to the trees with Raspberry Pi

Get snappy with incredible photo projects

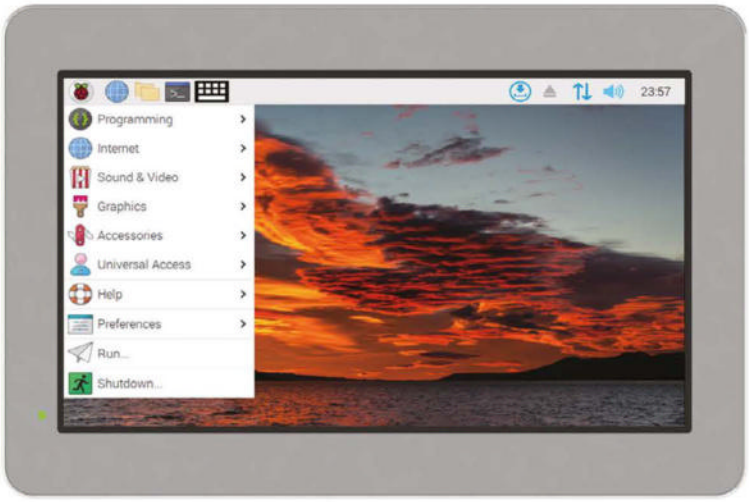
Build a drum machine with Pico

Raspberry Pi

Official Magazine
#155 | July 2025

07
£7.99
ISSN
2977-4403
(Print)

Industrial Raspberry Pi
ComfilePi



UL LISTED

Powered by Raspberry Pi

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

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

Visit www.comfiletech.com

COMFILE TECHNOLOGY

Welcome to Raspberry Pi Official Magazine

Editor
Lucy Hattersley
Lucy is reminiscing about 'Arms' the poorly named robot arm she once built and stuck googly eyes on.

rpimag.co

My kind of fun isn't settling in to a TV binge, though many of my friends adore it. My happy place is here with my keyboard, diving into nerdy books, comics, or leaping into a video game!

So, for someone like me, the thrill of crafting an Ultimate Home Media Centre, like the sensational one PJ has whipped up for us this month, is the pure joy of 'building' even more than the 'playing'.

Who wouldn't be utterly captivated by what it offers? Off-the-shelf systems often feel like they're playing catch-up. Some boxes might shy away from BBC Sounds with its HTML5 heart, YouTube comes and goes, and ITVX can feel like a quest.

Then there's the bigger adventure of building and enjoying your very own media library – an incredibly liberating feeling in the modern TV streaming world!

PJ has been tweaking his Ultimate Home Media Centre system for years, and this latest creation is the best yet.

Lucy Hattersley – Editor

We have over 400,000 new, name-brand products in stock and ready to ship–with more added daily.

If you can envision it, we can help you build it.

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

DigiKey

we get technical

Contents

World of Raspberry Pi

010 Argon ONE ⊥UP laptop now on Kickstarter

012 Raspberry Pi Connect 2.5 has been released

014 SECO Clea now native on Raspberry Pi

016 Subscribe to *Raspberry Pi Official Magazine*

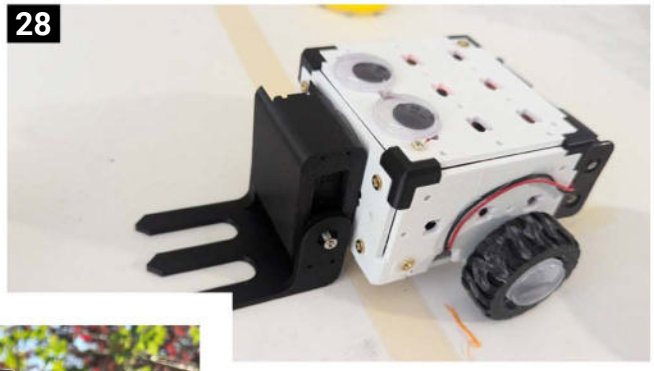
Project Showcase

018 Avanade Intelligent Garden

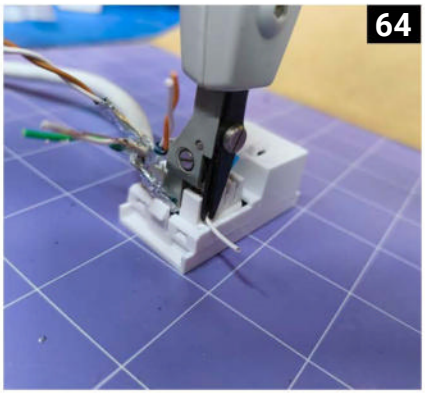
022 BaBot

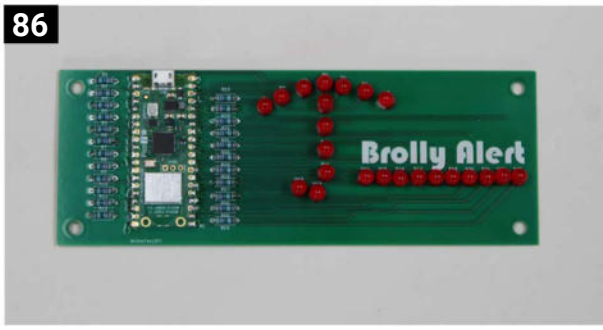
024 3D-printed RC car

028 Bentobot



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

032 CNC water cooling

034 Ultimate Lego 3D printer

036 PiPod

038 Simpsons and Futurama portable TV

040 3D print showcase

Feature

042 Build the Ultimate Media Centre: Set up a Jellyfin server for your own home

Tutorials

052 Conquer the Command Line – part 1

056 Pico drum machine

060 RISC OS for Raspberry Pi – part 3

064 Unusual tools: Krone tool

066 Hack a robot arm – part 1

072 Build a Sense HAT data logger

080 Origami paper circuits

086 Design your first PCB

Feature

094 Raspberry Pi photo projects: ideas to make the most out of your Camera Module

PCBWay

FULL-SERVICE ELECTRONICS MANUFACTURER

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

CUSTOMIZED TECHNICAL PARTS & PCB MANUFACTURED

WHY CHOOSE PCBWAY?

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

Hassle-free ordering

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

On-time shipping

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

Quality assurance

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

Customer support

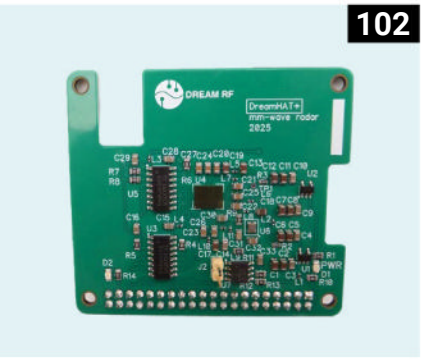
The customer service teams work inshifts to provide 24-hour support. You can always contact a live customer service person.

Scan the QR code for more details!

CONTACT US:

www.pcbway.com

service@pcbway.com



Reviews

102 Dream HAT+

104 ALPON X4

106 Only the best: Motor controllers

112 Ten amazing: art programs

Raspberry Pi Community

114 Open Hardware Summit 2025

116 Interview: Pater Practicus

118 This Month in Raspberry Pi

122 Your Letters

124 Community Events Calendar

Competition

Win 1 of 5 Sense HAT V2 bundles

WIZnet

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

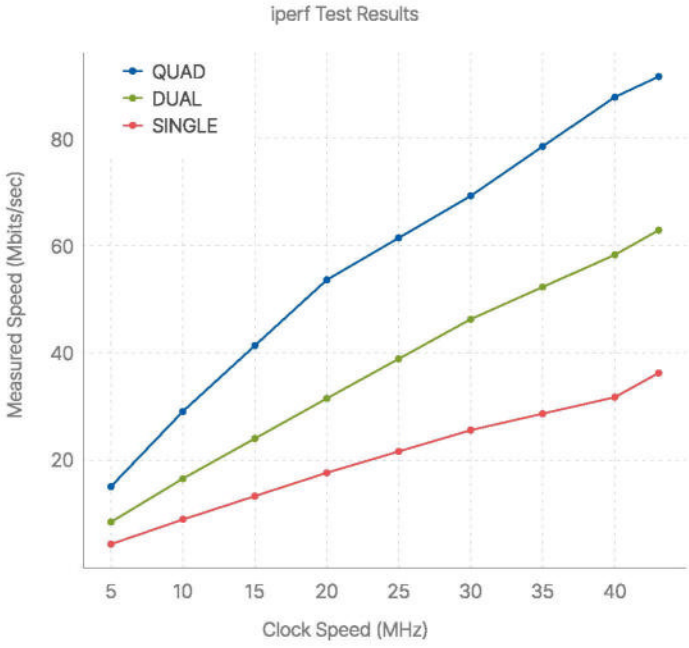
- **High-speed QSPI Interface**
  SINGLE, DUAL, QUAD modes supported
- **8 Independent Sockets**
  Multi-device connectivity made easy
- **Low Power & Wake on LAN**
  Efficient and remote-friendly
- **Seamless IoT & Industrial Integration**
  POS, cameras, automation & more

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) 

Argon ONE 上 UP laptop

Compute Module 5-based laptop seeking funding. By **Lucy Hattersley**

▲ The Argon ONE 上 UP laptop with CM5 inside

Argon 40 has just placed its upcoming Argon ONE 上 UP laptop on Kickstarter (rpimag.co/argonupks). Built around Compute Module 5, the laptop promises to be "powerful, modular, and built for makers, developers, and tinkerers."

"For years, Raspberry Pi users have wanted a real laptop – not just a DIY workaround," writes Joseph Zapanta, chief operating officer at Argon 40.

> *We've waited for this moment*

Powering the portable

The Argon ONE 上 UP laptop (上 also means 'up') is powered by Raspberry Pi's Compute Module 5. This provides all the powerful hardware features of Raspberry Pi 5 in an embedded board. CM5 is available with Wi-Fi and Bluetooth, up to 16GB SDRAM, and up to 64GB of eMMC storage (rpimag.co/cm5).

Compute Module 5 features the Broadcom BCM2712 quad-core 64-bit Arm Cortex-A76 (Armv8) system-on-chip (SoC) running at 2.4GHz. The target price of the Argon ONE 上 UP Laptop without the CM5 module is $299 (excluding shipping and taxes). For the unit with a CM5 8GB SDRAM and 256GB M.2 NVMe it will be $399.

Raspberry Pi's GPIO functionality is provided by means of a custom USB-C dongle with GPIO pins.

We have great respect for Argon's hardware design and manufacturing skills, with cases such as the V3 (rpimag.co/argonv3) and V5 (rpimag.co/argonv5) earning high-scoring reviews

Argon ONE 上 UP is housed in an aluminium chassis with a 14-inch IPS display (1920×1200, 250 nits). It will feature upgradable storage via M.2 SSD (PCIe 2.0) or a microSD card. "We've waited for this moment," writes Zapanta. "The hardware is here. The experience is in place. And the Raspberry Pi community deserves a laptop that stays true to the Raspberry Pi spirit while levelling up with them."

▲ A custom GPIO dongle provides Raspberry Pi I/O for electronics projects

Technical Specification

Dimensions
265 × 233 × 26.3mm, 1300g

Audio / video
14-inch IPS display (1920×1200, 60Hz, 250 nits)

Integrated 1080p front camera

Stereo speakers

Storage
M.2 SSD (PCIe 2.0 ×1) or microSD

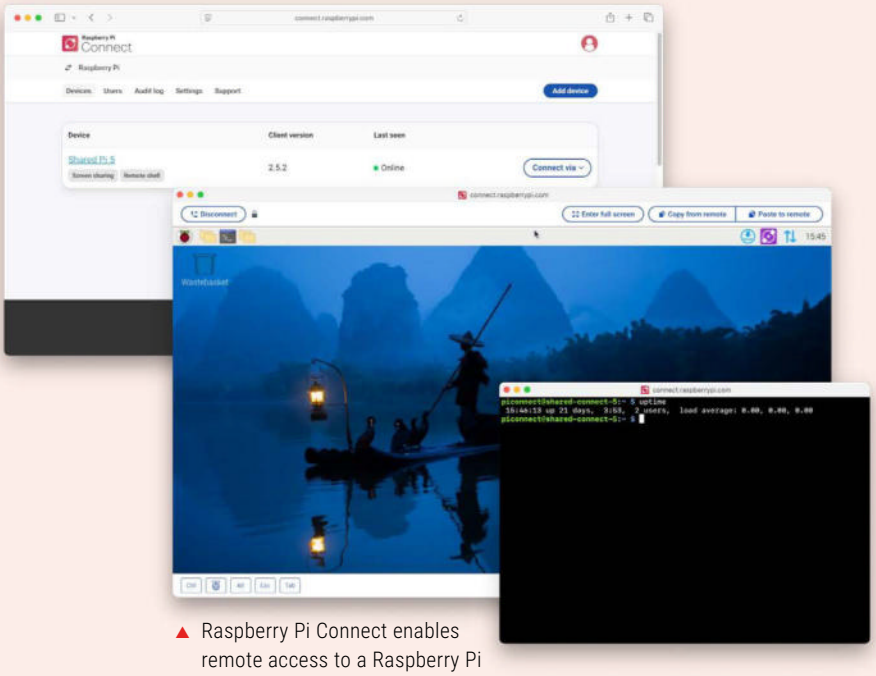
Connectivity
Wi-Fi 5, Bluetooth 5.0

2 × USB 3.1 Gen1
2 × USB Type-C (PD + Data + OTG)
HDMI 2.0 (4K@60Hz output)
3.5mm audio jack

Connect 2.5

Raspberry Pi Connect is out of beta: simple remote access, now even better.

By Chris Lowder



▲ Raspberry Pi Connect enables remote access to a Raspberry Pi from any web browser

An event is broadcast to the device to wake it up

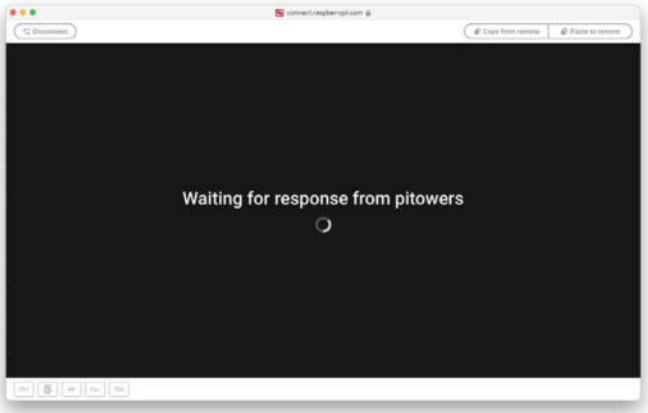
It's been just over a year since Raspberry Pi launched the Connect beta, giving you simple, remote access to your Raspberry Pi straight out of the box, from anywhere in the world (rpimag.co/connect). The response from users has been fantastic, and Connect has rapidly reached an install base of over 100,000 devices.

Raspberry Pi is excited to announce that following the recent release of version 2.5, it's dropping the 'beta'.

Smarter wake-ups

Prior to version 2.5, the Connect client software running on a Raspberry Pi device connected to the service would continually poll Raspberry Pi's servers for requests to connect. This worked well for us because it was easy to scale – traffic was a predictable shape; there was just a lot of it. But it wasn't ideal for users: their devices were regularly waking up to make HTTP requests, and data usage was higher than it needed to be.

Starting with version 2.5, the Connect client now holds a single long-lived HTTP connection to a Raspberry Pi server. Now when you click the Connect button



◀ Raspberry Pi Connect waiting for a response

on connect.raspberrypi.com, an event is broadcast to the device to wake it up and start the process of establishing a connection.

Optimised heartbeat

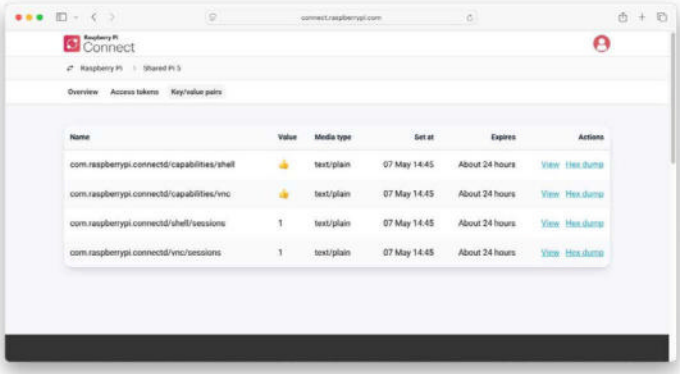
Separately from connection negotiation, the Connect client sends heartbeats to Raspberry Pi servers, periodically and also on startup and shutdown of a device and in response to changes to its internal state. For example, the user disallowing screen sharing via the CLI (command-line interface) would trigger a heartbeat. This information is then used to keep your dashboard on connect.raspberrypi.com up to date.

Prior to version 2.5, the Connect client would send four heartbeats in rapid succession; this wasn't a conscious design decision, but a side effect of how the client evolved over time. Starting with 2.5, these heartbeats are now debounced, and users should see many fewer requests to the Connect API outside of connection negotiation.

Also starting in 2.5, each individual heartbeat is now compressed before it is sent to the server, making it about 50% smaller.

How to update

To update to the latest version of Raspberry Pi Connect only, run the following commands (if you have installed Connect Lite, replace `rpi-connect` with `rpi-connect-lite`):



▲ Connections can be secured using SSH key/value pairs

```
$ sudo apt update
$ sudo apt install --only-upgrade rpi-connect
```

This week's other Raspberry Pi software news is that we've released a new version of Raspberry Pi OS; this has the latest version of Connect installed, so you might want to consider updating your OS. Read our post about the new release for instructions on how to do that: rpimag.co/osmay2025.

If you haven't tried Connect yet, check out our official guide to get it up and running on your devices: rpimag.co/connectdoc.

Raspberry Pi and SECO

Clea software suite available natively in Raspberry Pi OS. By **Lucy Hattersley**

▲ The Clea software suite is now natively available in Raspberry Pi OS

Raspberry Pi and SECO have announced that Clea is now natively available for Raspberry Pi OS.

Roger Thornton, director of applications at Raspberry Pi, calls it a "partnership that brings together Raspberry Pi's powerful, flexible computing platforms with SECO's expertise in edge computing and HMI (Human-Machine Interface) solutions."

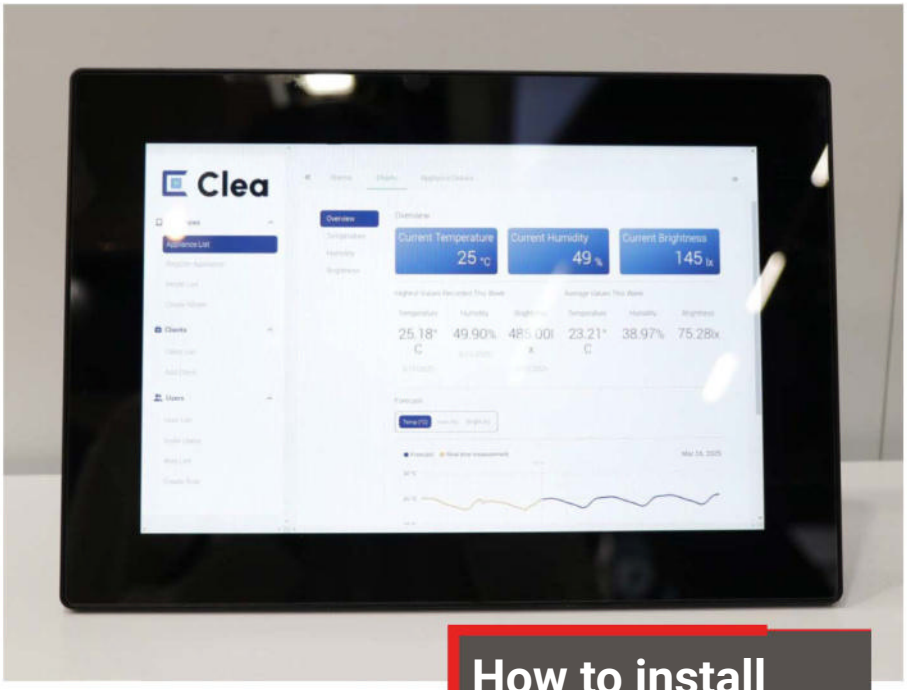
SECO's Clea (clea.ai) is a software suite designed to build and manage Industrial Internet of Things (IIoT) solutions. It integrates edge computing, cloud services, artificial intelligence, and device management into a unified platform.

"This streamlined integration makes it easier than ever to develop and deploy industrial applications, pairing the performance and reliability of Raspberry Pi with the advanced management capabilities of Clea," writes Thornton on the Raspberry Pi blog (rpimag.co/cleanews).

"Industrial developers want to move fast without compromising reliability or security," says Fausto Di Segni, SECO's European Head of IoT and AI. "By making Clea available natively in Raspberry Pi OS, we're removing friction from the development process and enabling a faster path to scalable, production-grade IoT solutions."

Clea provides robust features including remote device management, over-the-air (OTA) updates, and secure data handling, "all capabilities critical to modern IIoT deployments," says Thornton.

SECO's Clea documentation will guide you through the features (docs.clea.ai).



▲ Clea brings together edge computing, cloud services, artificial intelligence, and device management

This streamlined integration makes it easier than ever to develop and deploy industrial applications

Compute Module 5-based HMI

In late 2024, SECO announced its Pi Vision 10.1 CM5 (rpimag.co/pivision), a Human-Machine Interface (HMI) built on Raspberry Pi's Compute Module 5.

Thornton says, "Designed for industrial integration in demanding environments, this robust and feature-rich new device takes advantage of the power of Compute Module 5 to meet the needs of industrial customers for performance and longevity at scale."

SECO's Pi Vision 10.1 CM5 provides a "seamless path from prototype to deployment, helping customers to accelerate development and simplify the transition to volume manufacturing."

▲ SECO's Pi Vision 10.1 CM5

How to install Clea

Getting started with Clea on Raspberry Pi OS is as simple as running a few commands in the terminal:

Enable Clea's apt repo:

```
$ sudo apt update
```

```
$ sudo apt install -y apt-repo-clea
```

Install:

```
$ sudo apt update
```

```
$ sudo apt install -y astarte-message-hub edgehog-device-runtime
```

```
$ sudo apt install -y edgehog-device-runtime-forwarder
```

Full setup instructions are available in the Clea OS Get Started guide for Raspberry Pi.

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.

Avanade Intelligent Garden

Inviting visitors to interact with the plants at the Chelsea Flower Show proved a positive showcase for AI and Raspberry Pi 5, reports **Rosie Hattersley**

Maker

Pete Gallagher

Electronic and electrical engineer Pete works for Avanade, and has been an enthusiastic user of Raspberry Pi since its 2012 launch.

rpimag.co/avanade

Keen gardener King Charles III famously promoted the idea of talking to plants in a 1986 episode of children's TV show ***Blue Peter***. This same idea could be found at this year's Chelsea Flower Show, where technology company Avanade embedded sensors in trees and used an app to prompt conversations about each tree's health.

Alongside this, Avanade's Pete Gallagher created a bespoke Pollinator Camera, based on Raspberry Pi 5 and the Hailo AI accelerator, to photograph and detect the number of insects visiting the show garden each day. The Intelligent Garden won a Gold award for Best Construction at Chelsea Flower Show.

Trees, please

School children and conservation charities often use a form of AI to record wild visitors using hidden cameras and Raspberry Pi to identify them, but such uses could be better known. Avanade decided to bring a similar form of citizen science to people who may be unacquainted with how AI – or, more properly, machine learning – could be used to enhance their very own gardens.

To this end, Avanade approached renowned gardeners Tom Massey and Je Ahn to design an appealing garden with plenty of different species of trees, as well as plants that would attract insects and butterflies. The Times journalist Lucy Bannerman described the Intelligent Garden as a "garden that tells people how it is feeling". Given the blazing heat in London in late May, it mostly reported (via an auto-generated text message) being pretty thirsty!

Avanade's Helen Woodfield explains that plenty of trees are being planted in the UK. Leaf cover helps to cool streets,

> *All the stats say that street trees in cities are really suffering; 50% die within the first ten years*

remove pollutants, and provide much-needed shade. However, all the stats say that street trees in cities are really suffering; 50% die within ten years, wasting money and the goodwill of those who fundraised or paid for each tree. Having people adopt individual trees, regularly watering them with the necessary 30 litres per week, completely changes this narrative.

Nurturing green shoots

Alongside the chatty trees, the Intelligent Garden featured a Raspberry Pi 5-based AI Pollinator Camera Trap on the roof of the main Chelsea Flower Show pavilion which was covered in a meadow of wild grass. The bespoke bird box by Sebastian Cox is a thing of beauty in its own right.

"This camera trap aims to replicate a Royal Horticultural Society initiative called FIT (Flower Insect Timed) counting, which aims to monitor the number and type of pollinating insects visiting a quadrant of garden space and landing on flowers so the garden owner can assess whether they are attracting enough butterflies and bees," says Intelligent Garden designer Pete.

"By supplementing good horticulture and science with the power of AI, we hope to improve the chances of urban tree survival and help to maintain the existence of our much-needed urban green spaces."

The Pollinator Camera recognises pollinators and keeps a running total of the number of insect detections in the past

01. A Pollinator Camera installed on the pavilion roof at Chelsea Flower Show recorded insect visitors

02. Raspberry Pi 5 and Hailo AI accelerator automatically recognised and recorded photos of bees and butterflies

◀ A custom-built bird box houses the Pollinator Camera
▼ Pete in the Intelligent Garden with the Pollinator Camera hardware

30 minutes over a 24-hour period along with thumbnail images of each insect or butterfly identified. "It can detect flowers, bees, and butterflies currently and, when insects appear in frame, we take still images as well as videos, both with bounding boxes to show the objects for use in the application," Pete explains. The YOLO object detection setup currently records each instance, with tags added using Label Studio. He tried using AI to label them, but it was hit and miss so he ended up doing this part manually.

The next version of the Pollinator Camera software will identify individual bee species and insect types and record how many of each visits the garden. The live data means the thumbnails already being logged can be added to the dataset (or zoo) to train the next iteration of the software to improve its accuracy. "Of course, if you start training on an image that's got a box around it, it's going to be very quick to identify boxes, not bees. It's the same for the video as well." To get round this, Pete created "a little piping script" that captures one-second interval screenshots from the video. Avanade could use each of those "because of course, the position of the bee would change, and it would rotate, and it seemed important to train it on all of those". There are detailed build instructions at rpimag.co/intelligentgarden. Meanwhile, Avanade urges makers and nature fans to try and build their own versions and share their findings. There's even talk of a competition for the best young citizen science projects based on the one at Chelsea Flower Show!

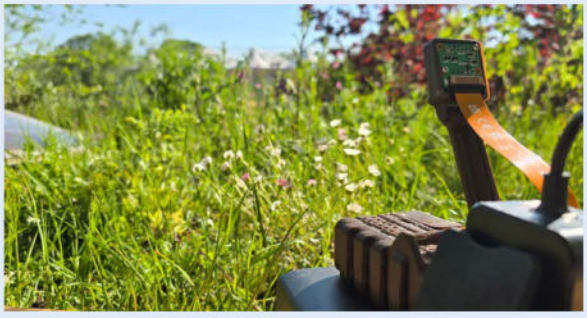
▼ The Pollinator Camera detects and logs numbers of bees arriving in the flower garden transect

Quick FACTS

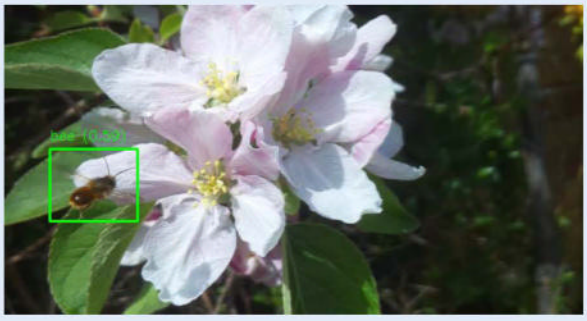
- Pete previously built stamp vending machines with modems for the Royal Mail
- He also worked on a project to save elephants near the Zambezi river
- Avanade's aim was to introduce AI to people who have only heard about its downsides
- Citizen science and engagement with nature caught the imagination of visitors
- The Intelligent Garden won the Chelsea Flower Show's Best Construction award

Set up a smart garden

1. A pre-built or 3D-printed bird box can be used to hide the camera-based setup. Using an 8GB Raspberry Pi 5 with an HQ Camera, 40W power supply, and the Raspberry Pi AI HAT+ 26 TOPS made integration very simple, says Pete.



2. Installed on the pavilion roof, the camera automatically logs insect visitors and sends data back to Avanade's server via a fixed Power over Ethernet connection.



3. Insects identified are shown in a bounding box. The thumbnail imager can be dragged into your dataset for ML training purposes. Build instructions for the Pollinator Camera can be found at **rpimag.co/intelligentgarden**.

BaBot

Andrew Gregory discovers a robot that's better at balancing than he is

Maker

Johan Link

Johan is a student and passionate maker from Switzerland. He's always loved building things, and started working on the first prototype of BaBot as a high school project six years ago.

ba-bot.com

Have you ever seen a robot that can play catch? Well now you have. This is BaBot – a fully open-source robot built by architecture student Johan Link. And it's absolutely ingenious – there's no camera here to keep track of the ball, as you might think at first glance. Instead it uses a matrix of infrared LEDs and phototransistors to detect the position of the ball. The LEDs emit light, and the phototransistors measure what's being reflected – in a way, Johan has created a very simple version of an eye, with the photoresistors standing in for the cells in the retina.

Quick FACTS

- Johan built the first prototype in 2018
- Since then he's refined it, and now it's ready for the world
- The BaBot uses an ATmega32U4 microcontroller
- All the code is open source, and available on GitHub
- The two halves of the robot are connected by magnets

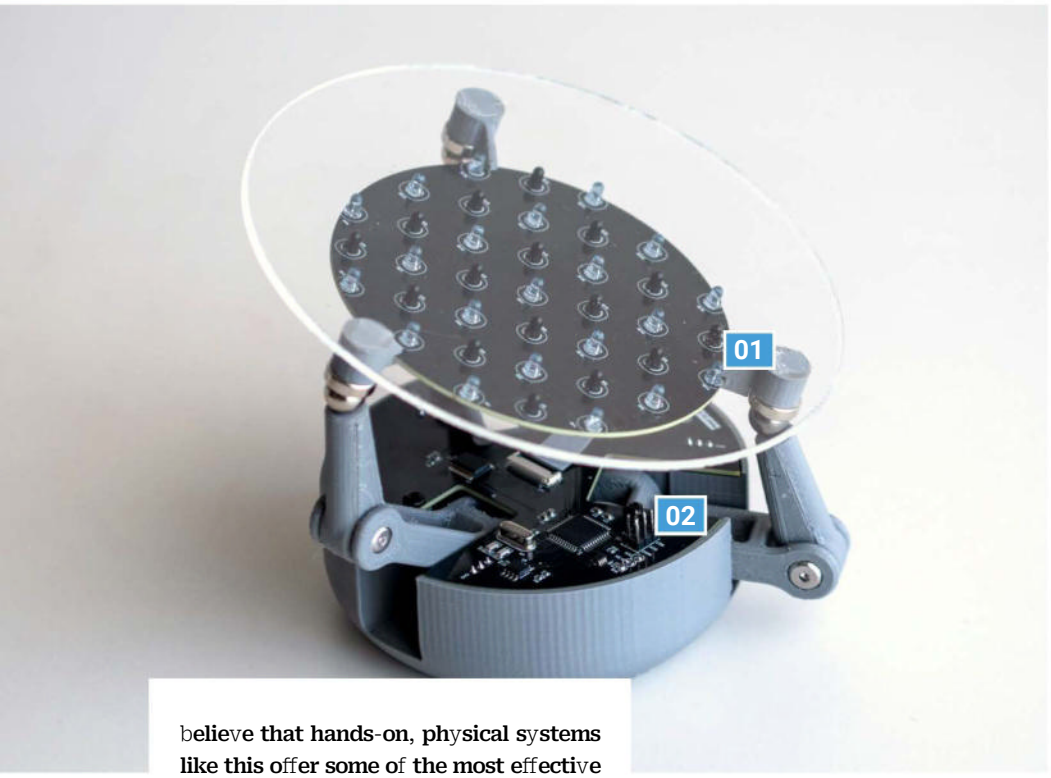
The ball location data is sent to an ATmega32U4 microcontroller (the kind that's also used in the Arduino Leonardo), which controls three servos attached to simple mechanical linkages that connect via magnets to the top plate; this adjusts the position of the top part of the robot, balancing the ball on the acrylic plate on top, and creating a continual feedback loop between the ball, plate, sensors, and servos.

You can just about make it out in the photos, but the joints between the top half of the robot and top of the mechanical arms are metal balls and magnets; this gives the joint smoothness and ensures no loss of contact to mess with the algorithm's calculations.

BaBot isn't a new idea: Johan made the first version as a high school project in 2018, when videos of it went viral on Instagram. The version you see here represents years of passion, ingenuity, and hard work. "I wanted to create something that brings abstract engineering concepts to life," Johan tells us. "BaBot is a great way to teach and explore robotics, mechatronics, and control systems, but beyond that, it's simply fun to watch and interact with. I

▼ The robot was programmed in the Arduino IDE

BaBot



01. BaBot uses two custom PCBs: one to hold the IR LEDs and sensors…

02. …and another to house the motors and processing

believe that hands-on, physical systems like this offer some of the most effective learning experiences."

The first version of BaBot used an overhead camera to monitor the ball, which required much more processing power; using LEDs and photoresistors has enabled the use of a much smaller controller, but it did give Johan a headache when it came to dealing with ambient light.

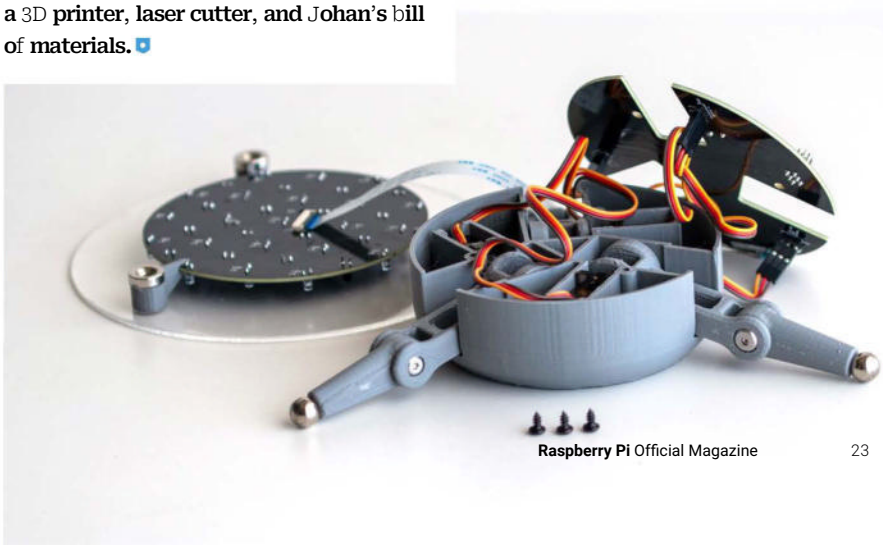
> *There's no camera here to keep track of the ball*

The robotic balancing act is powered by a 5V, 10A DC power supply, as well as what Johan describes as a "classic PID algorithm".

The microcontroller processes the data using the PID (Proportional-Integral-Derivative) algorithm, which works out the difference between the ball's current position and the desired target position (typically the centre of the plate). It then calculates the correct angles for each of the servos; the plate's orientation changes, and the process repeats approximately 30 times per second to keep the ball perfectly balanced.

If you fancy building your own ball balancing robot you can buy the kit from Johan at ba-bot.com; alternatively, you can make your own with a 3D printer, laser cutter, and Johan's bill of materials.

▼ Johan got the custom PCBs made at PCBWay



3D-printed remote control car

Printing your own car no longer seems the stuff of futuristic fantasy, as this Raspberry Pi aficionado demonstrates. **Rosie Hattersley** reports

Maker
Eugene Tkachenko
Eugene is an optimistic and open-minded tech enthusiast who produces great videos about his many Raspberry Pi builds.

rpimag.co/itkacher

Eugene Tkachenko is a confident YouTube presenter who demonstrates his many projects with enthusiasm and aplomb. He spotted Raspberry Pi's potential from the get-go, noting "boards for various needs and budgets, from affordable options for simple projects to boards capable of handling AI and heavy computations".

His first Raspberry Pi project, an underwater drone, appeared back in issue 80. With around 15 years experience as a developer and multiple Raspberry Pi builds to his name, Eugene set himself a new challenge: to design and assemble a fully 3D-printable remote control car using no aluminium, steel gears or fancy parts (rpimag.co/3dprintedcar). He chose Raspberry Pi Pico 2 W for his 3D-printed car "because of its robust processing capabilities and Wi-Fi support" and anticipated the size of his 3D printer's bed being the project's main hurdle. "On paper it sounded simple. Reality had other plans."

Eugene relishes the Raspberry Pi community in which engineers provide excellent documentation, and enthusiasts make using Raspberry Pi even easier with third-party libraries and articles. "It's easy to start and become proficient with such a versatile product," he proclaims. His experience with other aspects of the RC car project were not so smooth.

> *Unlike metal, plastic struggles under high torque, leading to wear and eventual failure over time*

A bum steer

Eugene based his design on real-world, full-scale car components. "The goal wasn't to reinvent the wheel, but to adapt proven concepts into a 3D-printable format" that would work with components such as brushed motors, wires, and connectors. Reimagining standard automotive parts using various types

01. Eugene Tkachenko's 3D-printed remote-control model car even has stacked TPU discs instead of metal shock absorbers

02. Raspberry Pi Pico 2 W sends PWM signals to the H-bridge and servo to drive and steer the car

Warning
Fire risk!
During the build process, the H-Bridge exploded. Be careful when testing and follow the insulation instructions.
rpimag.co/firesafety

of plastic meant experimenting to find combinations that could hold up under real RC car conditions. He "burned out several H-bridges, shredded countless 3D-printed gears, and cracked more than a few axles", but learned plenty about material limitations and mechanical stress along the way.

Durability proved a big issue when it came to the TPU (thermoplastic polyurethane) gears and drive train. "Unlike metal, plastic struggles under high torque, leading to wear and eventual failure over time." He had anticipated such issues, but was shocked when his 3D-printed pinion shaft for the differential gear system quickly melted due to the heat caused by friction. This problem was eventually fixed by adding bearings (tiny rolling cylinders) around the shaft, along with ventilation holes in the gear case.

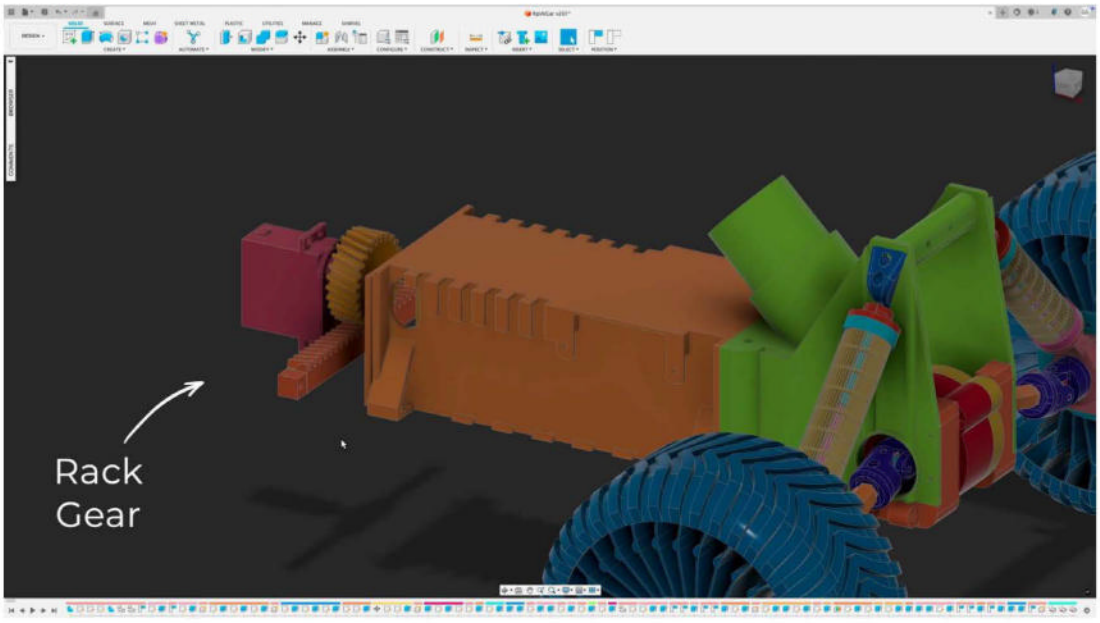
Many revisions later, he added huge Michelin-style TPU airless tyres capable of driving in all weather and terrains.

Smoke and (wing) mirrors

Eugene used the open-source FreeRTOS (freertos.org) for microcontrollers alongside Raspberry Pi Pico SDK. Pico acts as a Wi-Fi access point, allowing a mobile phone to connect directly and send control signals. It then processes those inputs

Quick FACTS

- The project took around four months to complete
- The parts cost a total of $105
- The current version of the car uses a Bluetooth controller
- Eugene is considering adding an AI camera and 5G network
- This would allow the car to be driven from a remote location



◀ After many revisions, planetary gear reduction sorted out torque and force distribution issues and reduced failures

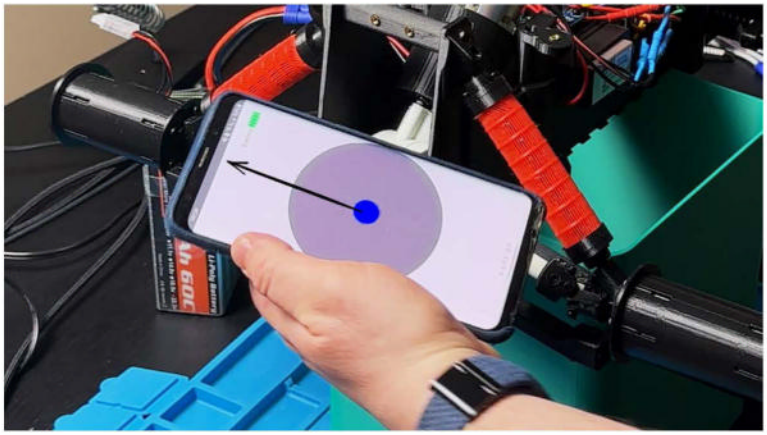
▲ A self-balancing joint reduces rough impacts, allowing the car to tackle all terrains

and uses GPIO pins and PWM to control all the car's electronic components, such as motors and servos.

He cites a major advantage of Raspberry Pi as the ease of swapping boards. Although he settled on Pico 2 W, it wasn't the only option. "During development I was able to replace a more expensive board with a less powerful one in under an hour, with no loss in functionality. This makes prototyping cost-effective and accessible for anyone." Eugene also finds 3D printing a boon, allowing for "rapid prototyping and easy customisation, making it possible to test and refine designs quickly and cost-effectively".

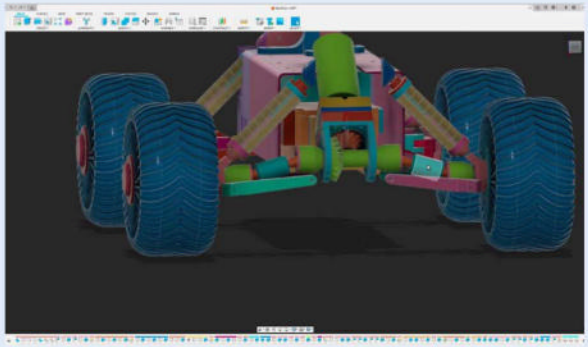
Once he got everything built, the car refused to move, the wheels barely turned. Instead, Eugene noted a distinct smell... then the H-bridge exploded! A complete rebuild along with a massive motor controller with proper thermal insulation did the trick and the drive train "finally came alive".

Eugene is realistic about the creation process. "None of my projects work on the first try. Even not on the second. Eventually they work, not as intended, but still, I am having a lot of fun!" For the behind-the-scenes story of what ought to have worked and didn't, Eugene's post on Hackster is well worth a read: rpimag.co/3dcarhackster.

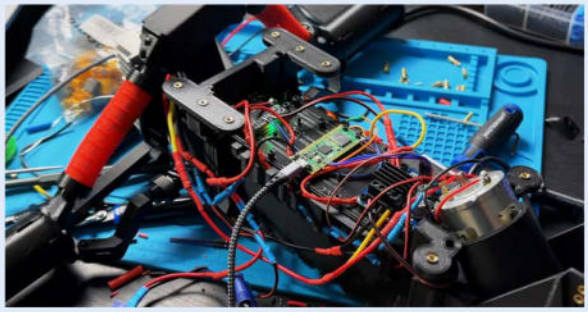


▼ Eugene created an Android app that communicates with Pico 2 W to control the RC car

Kit car



1. Almost everything is 3D-printed, including the TPU wheels and chassis – apart from Raspberry Pi Pico 2 W and H-bridges for connectivity, brushed 777 motors, and electronic connectors. The RC car has a servo-controlled gear and rack system for steering.



2. The motor is huge, but Pico 2 W fits snugly inside the chassis. Eugene programmed it in FreeRTOS to act as a wireless access point so he can control the car using the Android app he wrote.

3. Eugene says the 6500mAh Youme Power batteries were "absolutely worth the extra outlay" for the full hour of RC powered enjoyment they provide. For more on the build, see his excellent YouTube video: **rpimag.co/3dprintedcar**.

Bentobot

Want to battle real robots without building one? Bentobot has you covered, as **Rob Zwetsloot** finds out

Maker
Thomas Nguyen
A Vietnamese engineer and hobby roboticist living in Florida, he began his Raspberry Pi journey in 2020.

bentobot.tech

If Pi Wars is anything to go by, the appeal of robot battling will always be strong. However, Pi Wars is, for many good reasons, non-destructive – maker Thomas Nguyen ignored all these reasons when he created the online sensation Bentobot.

"[It's] the most accessible way to experience combat robotics," Thomas says. "I built three combat robots and developed an application that lets people control them from anywhere in the world. I streamed twice weekly, hosting fights between players from India, Canada, the Netherlands, the US, and beyond. Players simply grab their phone or computer, sit on their couch, and pilot real robots in combat from wherever they are."

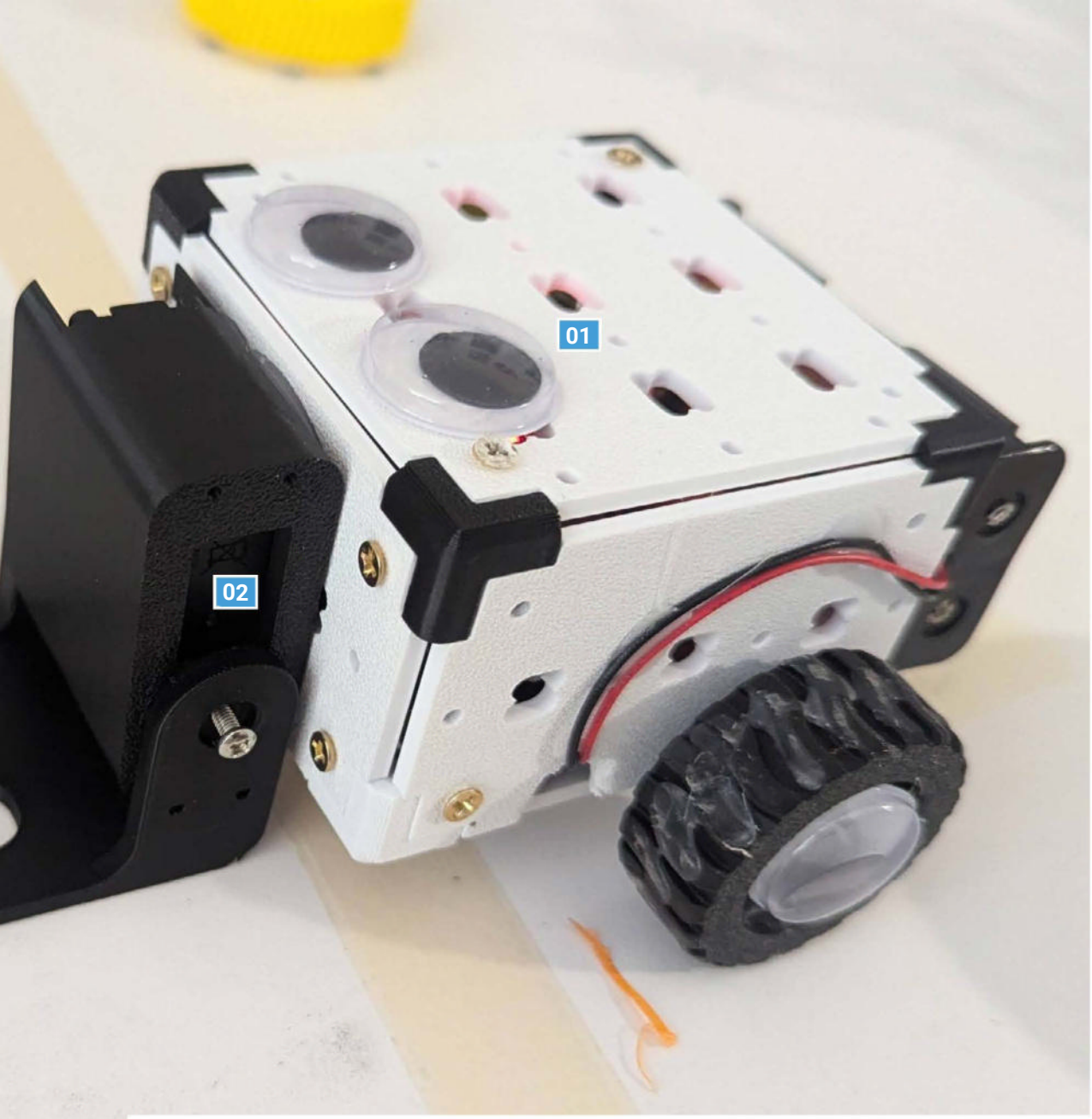
While making fun robots with his online friends, Thomas wanted them to try them out. As is common these days, his friends are scattered around the globe, so his solution was to create a "simple app" so they could control the robots remotely.

"Word spread quickly, and soon people from around the world wanted to play," Thomas explains. "What started as a solution for friends became a global community experience."

Crumple zones

The remote robots are powered by a Raspberry Pi Zero 2 W, Thomas's go-to microcomputer, and a popular choice for robots everywhere.

"I designed the robots for easy repairability – essential given the combative nature of the game," Thomas says. "Each of the three robots is controlled by a Raspberry Pi Zero 2 W and features three DC motors: two for movement and one for the signature weapon. I use either TT motors or modified N20 motors, depending on the application."



01. Googly eyes are very important to make sure the robot gets extra power

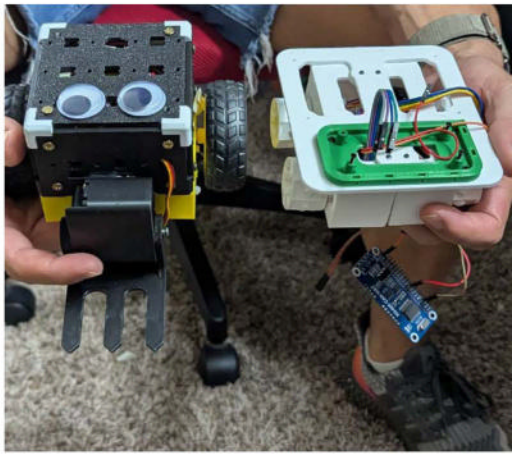
02. The weapons on the three robots are all slightly different

I designed the robots for easy repairability – essential given the combative nature of the game

Each robot can operate for about four hours on battery power. "Everything fits compactly within a 4 × 4 × 4-inch chassis," Thomas says. "Each robot sports googly eyes and a unique weapon – flippers, spinners, or blades – inspired by NHRL (National Havoc Robot League) combat robots. The control application was built in Flutter for true cross-platform compatibility (iOS, Android, Windows, macOS, and web). It connects to a real-time Firestore database – when players update commands in the app, each robot instantly receives the changes and responds accordingly."

Thomas even created a custom programming language for this called comfyScript (rpimag.co/comfyscript), which specifically allowed for ultra-low-latency network communications on Raspberry Pi robots.

▼ Cube robots are easy to build and repair



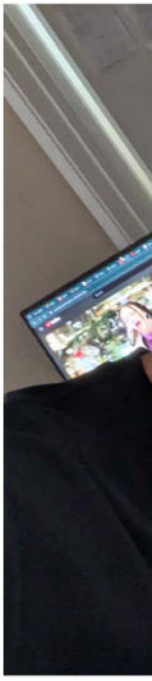
Live entertainment

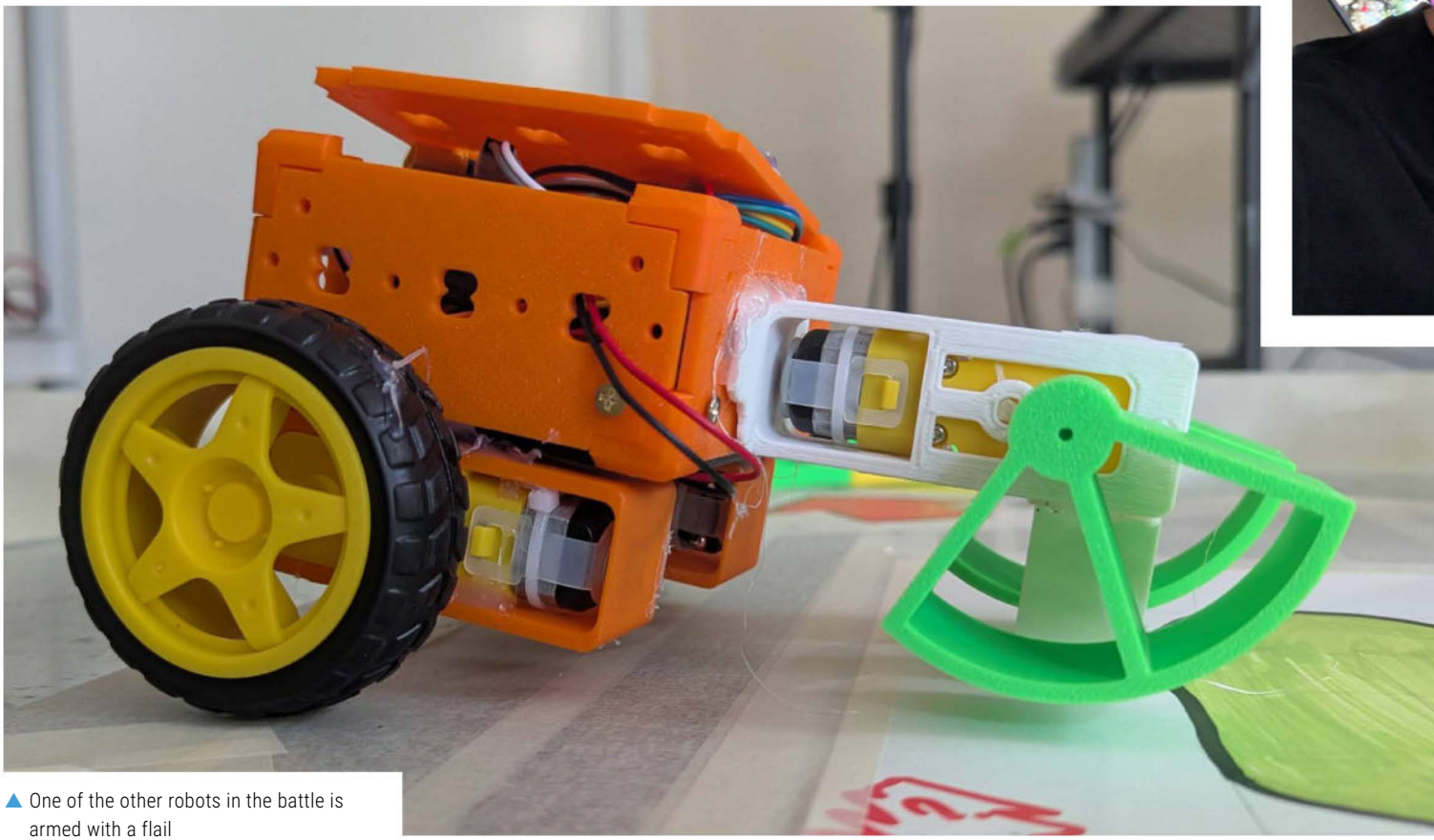
The fights are broadcast live on YouTube, with members of the chat being given access to a robot.

"They log into the application and start battling immediately," Thomas explains. "The interface is elegantly simple: one joystick controls robot movement, one slider operates the weapon, and a live camera feed provides low-latency visual feedback. Players compete to score five points by immobilising opponents or destroying significant portions of the arena."

Bentobot ran for two "intensive months", and Thomas says that many incredible friendships were forged in an active and welcoming community.

"Due to time and budget constraints, I've retired the active project, but I've open-sourced all code and mechanical designs with comprehensive instructions [at rpimag.co/bentobotgit]," Thomas tells us. "By popular demand, I'm now helping former players organise their own Bentobot battles in their local communities. I'm excited to pursue more community-centric projects because the most meaningful building happens when we create with, and for, others."





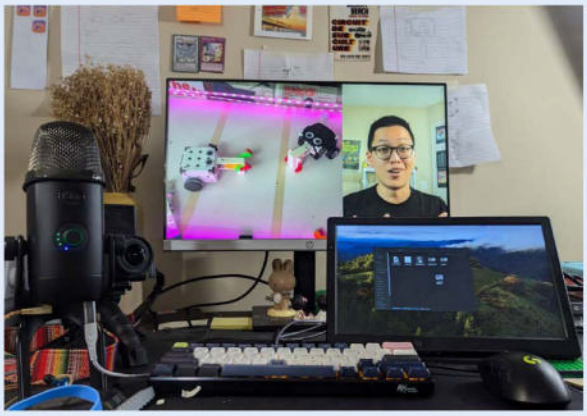
▲ One of the other robots in the battle is armed with a flail

Quick FACTS:

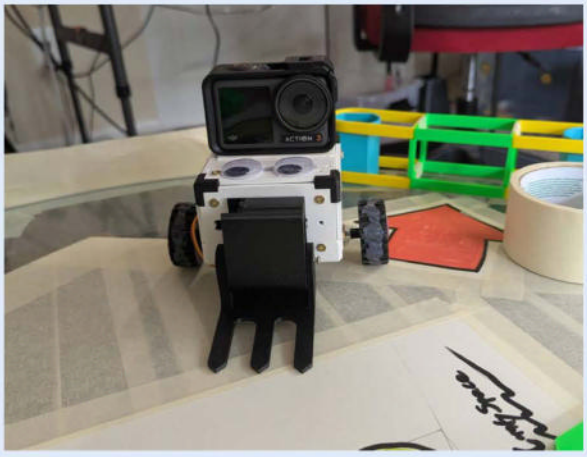
- Parts were designed in Onshape
- The chassis is 3D-printed
- Thomas's friend Ankitt helped with the application
- An entire church congregation in India took part in a battle
- Thomas is now working on LLMs for Raspberry Pi

▲ The idea came from Thomas building robots with his online friends

Go head-to-head



1. Watch the stream on YouTube. You get a live view of the robots, and Thomas will choose the players for each match.



2. Log into the custom interface to control your robot. It includes ultra-low latency so the lag is very low.

3. Fight, fight, and fight with your robot! Hopefully your tactics will succeed and you will be crowned victor.

CNC water cooling

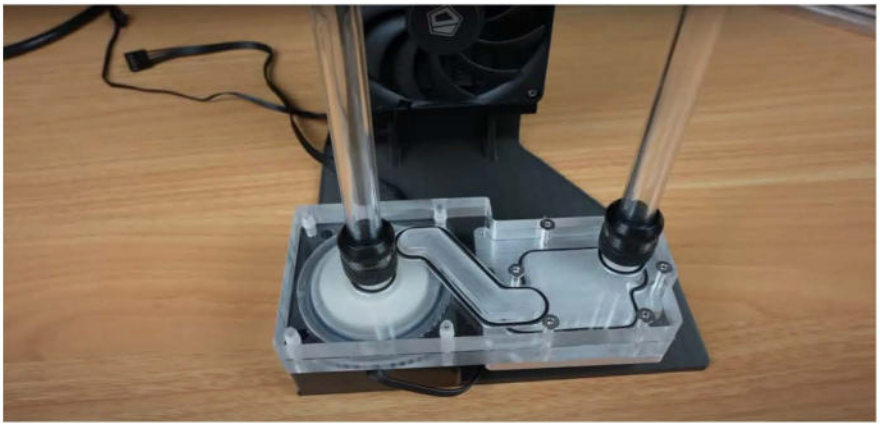
By Michael Clements

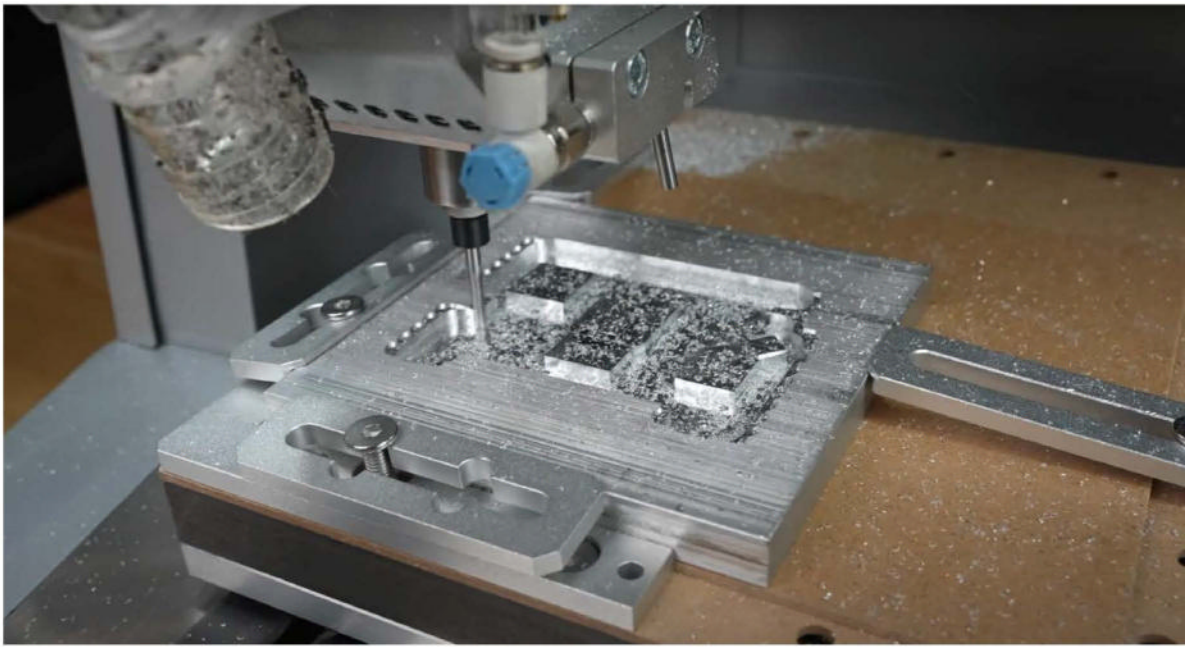
rpimag.co/CNCWaterBlock

If your Raspberry Pi 5 is running hot and you want to bring the temperature down a bit, you could try closing a couple of browser tabs. Or open the window. Or maybe buy the official Active Cooler from Raspberry Pi, which clips on and comprises a blower and an aluminium heatsink. In the unlikely event that this isn't enough, maybe you could take a leaf out of Michael Clements's book and mill a custom aluminium block, with a milled acrylic reservoir to hold bright green cooling fluid, pumped through a radiator and cooled by a fan running on a separate 12V power supply. As Michael says, this approach is "obviously a totally over-the-top project – but that's what made it fun".

Michael milled the aluminium out of a pice of 10mm stock on a Carvera Air home CNC machine; this block makes direct contact with the bits of the Raspberry Pi that produce heat: the CPU, the RAM, the USB and Ethernet controllers, and the power circuitry. On top of this there's the water block, which also houses the pump. The device is mounted on a 3D-printed stand, with the Raspberry Pi 5 attached to the aluminium block via four M2.5 screws, the holes for which Michael tapped himself.

The sheer ridiculousness of this build (that's Michael's own word) is what caught our eye; then when we learned what has gone into it, we were blown away by the variety of different skills he had to use. You'd never know it, but this is the first time he's had to bend hard-line pipes. It's incredibly clean, and it works too – so hats off!





▲ Michael's homemade cooler keeps the Raspberry Pi 36 K (Kelvin) cooler than the equivalent setup using a Raspberry Pi Active Cooler

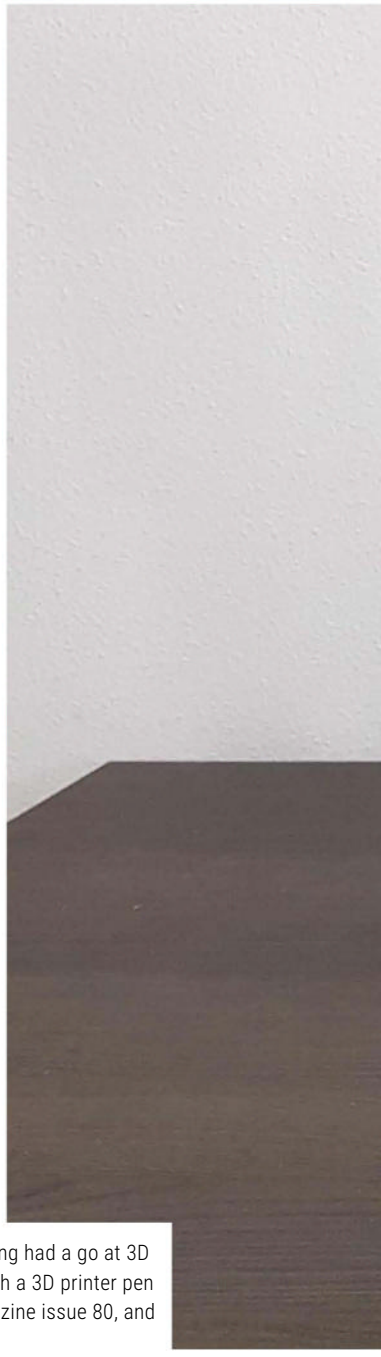
Ultimate Lego 3D printer

By Creative Mindstorms

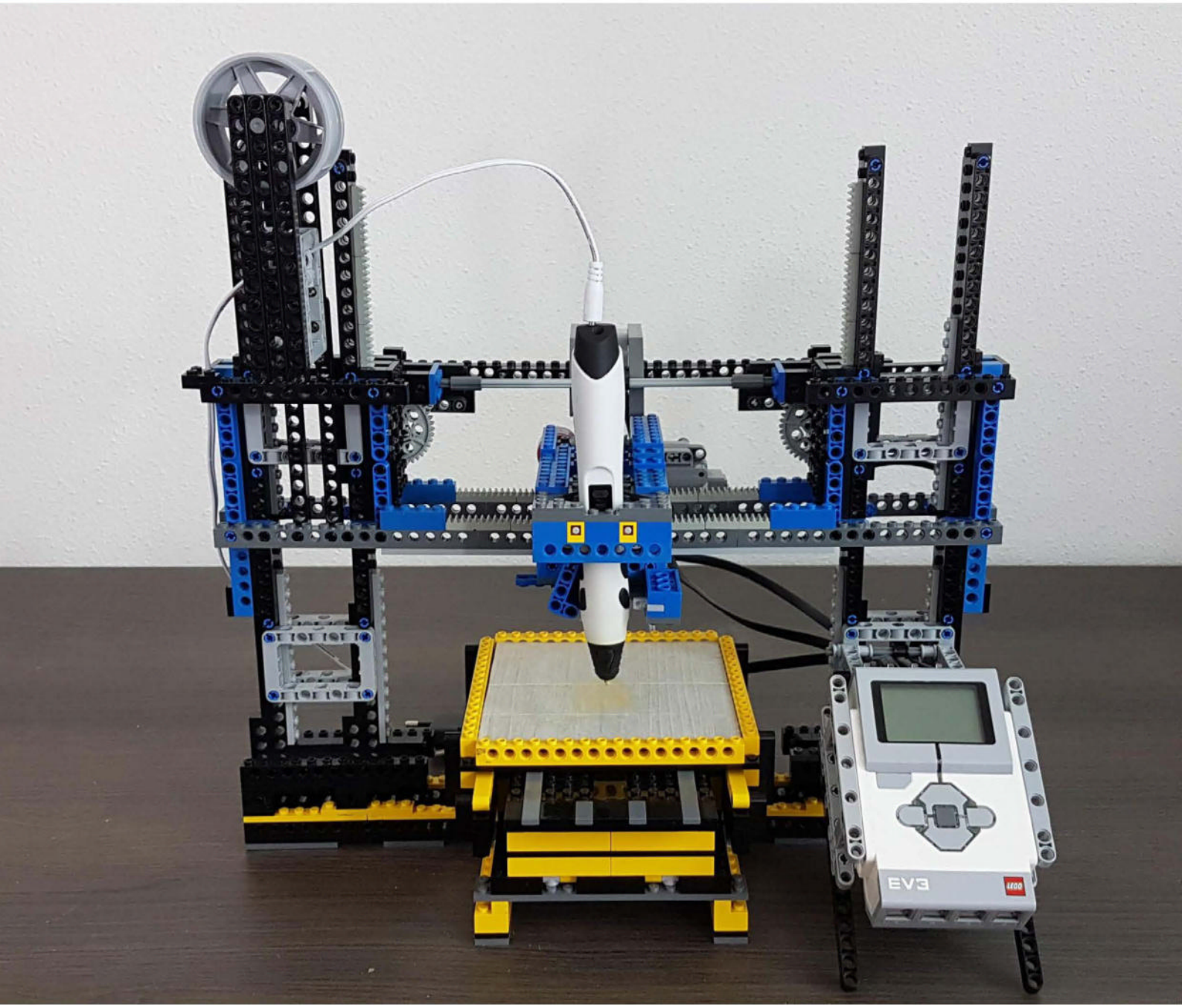
rpimag.co/LegoPrinter

Before Raspberry Pi, before Arduino, there was Lego Mindstorms. Mindstorms was an early attempt at putting physical computing into the hands of Lego aficionados, and while it was relatively expensive and relatively fiddly to use, it retains a core of dedicated fans who love that you can build computerised robots out of original Lego parts that were first released in the late 1990s.

This 3D printer from the Creative Mindstorms YouTube channel is the latest Mindstorms build we've seen that's really impressed us. It uses Lego Mindstorms components, including four motors, and a load of Lego bricks to form an actual, working 3D printer. There's a 3D printer pen to produce the hot plastic rather than an extruder, but that could change in a later iteration. What's important here is that the Lego setup can print 3D models just like any other printer. It's not as neat as a Prusa XL, but after you've built one of these you'll know a lot more about how a 3D printer works.



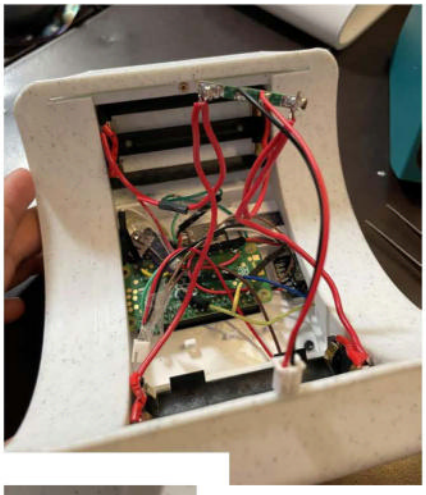
▶ Our writer Nicola King had a go at 3D printing by hand with a 3D printer pen in HackSpace magazine issue 80, and found it… tricky

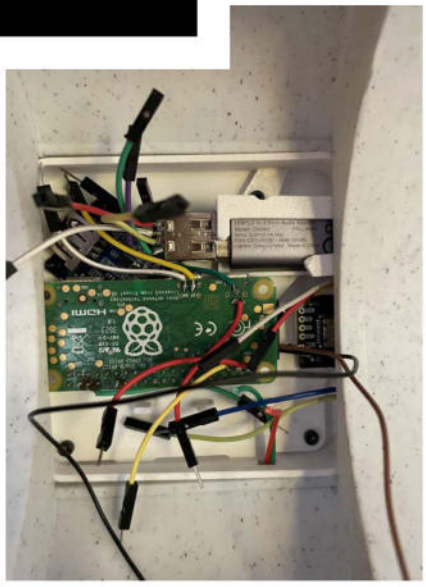

EV3

PiPod

By kinpro1024

rpimag.co/PiPod





The PiPod is a throwback to a simpler time, before streaming, before automatic shuffle and having Beethoven's 9th interrupted by ads became the new norm. A time when you'd save up your money, go into town, buy a whole album on compact disc or vinyl, and study every aspect of the cover artwork on the long bus ride back before you got back to listen to it. The maker of this device is searching for a time when we listened to music properly, but also had the ability to carry thousands of songs with us on a mobile device. And so he created the PiPod: a portable music device that not only plays tunes, but also displays the artwork that album designers used to take such a lot of effort to get right.

The PiPod uses a USB DAC, components that enable the battery to work (including a battery charge module, battery protection circuit, and a battery indicator module), a Waveshare 4-inch DPI Square Capacitive Screen 720 × 720, and a Raspberry Pi Zero 2 W. Oh, and a 3D printer and about 500g of filament.

The 3D-printed case is shaped to fit around the wearer's arm, like the phones worn by all those people we see out running. What you see here is not the first iteration; the maker has documented a raft of mistakes, iterations, and happy little accidents to get to the finished version.

▲ We're simple creatures: we see David Bowie's *Blackstar* album, and we like

Simpsons and Futurama portable TV

By oculiaeternam

rpimag.co/SimpsonsFuturamaTV

Reddit user oculiaeternam took the (non-functioning) guts out of a 1984 Bentley portable TV, and replaced them with a Raspberry Pi Zero 2W, a 5-inch LCD screen, a new speaker, and a single-channel amplifier. The result is a portable TV with a load of spare room inside the case, with a Raspberry Pi Zero 2 W inside.

There are a few things to note here. The first is that the original volume control is now attached to the amplifier, so the form of the original device follows the function of the new device. The second is that the LCD screen is ever so slightly a different aspect ratio than the original broadcast episodes of *The Simpsons*, so the visual joke about Duff Lite and regular Duff beer coming from the same tanks in the brewery won't work.

The other thing that caught our eye with this build is that it uses ChatGPT for something more useful than rubbish AI art: there's a script running which decides which episode to play next, which depending on the prompt (*The Simpsons* and *Futurama* can be streamed from Disney+), may be as good a way as any to get randomness on a computer.

ChatGPT has its uses – one of which is deciding which episode of *The Simpsons* / *Futurama* to play next

3D print

Why spend two minutes cleaning up when you could spend a couple of hours printing a dust catcher?

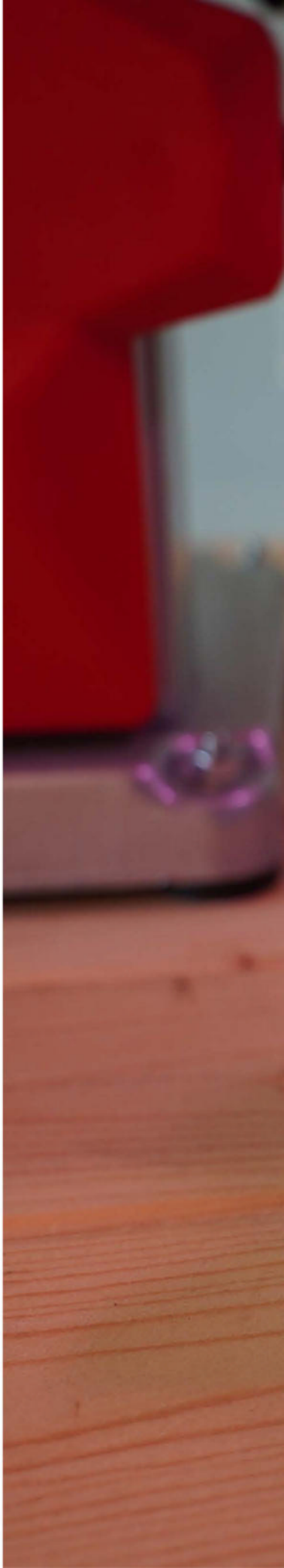
By **Toby Roberts**

rpimag.co/DustCollector

Not all power tools are created equal. Take the hand drill: it's brilliant. Cutting through wood, metal and masonry with ease, it recharges quickly, it makes a cool noise, and it gives you the perfect excuse to acquire a collection of drill bits that you absolutely need as if your life depends on it even though you might only use them once in a blue moon.

In contrast, consider the vacuum cleaner: used to clean surfaces that will inevitably only get dusty again. Heavy, bulky, whiny, vacuum cleaners are not our favourite things. But Italian designer AK3D has come up with a way to make vacuum cleaners more appealing: by printing a range of attachments to connect them to a drill, creating a vacuum-drill hybrid. That's more like it.

This attachment was printed on a Bambu Lab A1 using PLA and the automatic material system, which lets you print in up to four colours in a single print. The main body was printed with red filament for the initial layers, transitioning to black for the remainder. This colour change serves a functional purpose, improving the visibility of alignment holes and allowing for more accurate drilling.



Build the ultimate Media Centre

Whether it's rugged server cases or beautiful set-top boxes, we've got you covered. By PJ Evans

Building a media centre is a wonderful home project. You can take full control of your music, video, and television experience.

Like all computer projects, there are two elements: hardware and software. Building your perfect media centre is no different. There's lots of choice and what's best will depend on your use case. So, we've gathered up the very best in hardware and software to help inspire your media centre project.

Argon ONE cases

Unless you're a bit of a minimalist and laugh in the face of static electricity, you're going to need a case with some active cooling included. Media centres can be resource-hungry things and will use techniques like hardware graphics acceleration to process video. It also matters whether you want a box tucked away that can be accessible over the web or via dedicated client, or if you're after a case on display for use in your home theatre. Either way, Argon has you covered with the excellent ONE range (argon40.com). Choose from ONE V3 for a good-looking unit with IR (Infrared) control built-in or the more industrial ONE V5 for dual NVMe storage and built-in DAC (but no IR). See also the excellent KKSB Case which can contain a DAC Pro HAT (rpimag.co/kksb).

▲ The Argon ONE V5 is the case of choice for a display-less media server

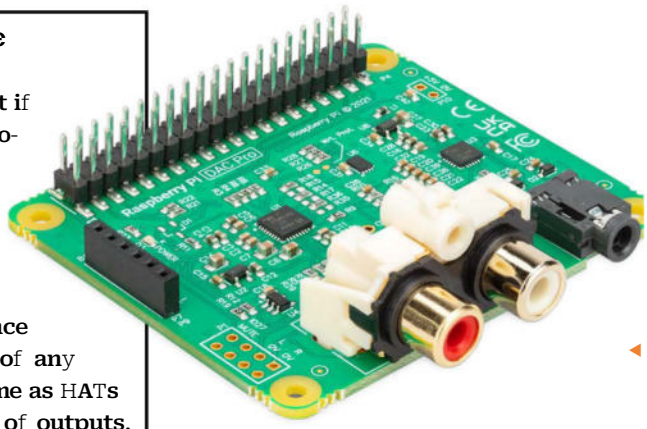
Argon IR Remote

If you're keen on the ONE V3 case with its sleek lines and small footprint, you might want to take advantage of its onboard infrared receiver. Argon sells an optional remote control (rpimag.co/argonremote) with the drivers to help integrate with your choice of media centre software. It's inexpensive and very Apple-like in design. Pair it with the ONE V3 and you can design and build your own version of Apple TV or Fire TV Stick and enjoy it from the comfort of your sofa.

▲ Add this sleek remote to the IR-capable Argon ONE V3

Digital-to-analogue converters

An essential investment if you're looking at an audio-first project and are not using HDMI for audio, a digital-to-analogue converter (DAC) provides a much better audio experience than the capabilities of any Raspberry Pi. They come as HATs and provide a variety of outputs, from a headphone jack to RCA line-out connectors, and some even have amplification onboard to connect directly to speakers. Combine one of them with a touchscreen and you have a standalone media centre. Costs vary by capability, but the official DACs are a great place to start (rpimag.co/dacpro).



The official DAC Pro HAT adds superb audio to your project

Media centres can be resource-hungry things and will use techniques like hardware graphics acceleration to process video

SmartiPi Touch 2 and Touch Display

If you miss jukeboxes or fancy a more interactive experience, how about creating a touchscreen-controlled media centre? We love the SmartiPi Touch 2 enclosure for the official Raspberry Pi Touch Display (v1). It can be wall-mounted or stands freely and the larger rear enclosure means you can fit a decent DAC inside for improved sound. When paired with a dedicated music OS such as moOde, you can control everything without a dedicated keyboard. Perfect for communal spaces (rpimag.co/smartipi2).

Add this SmartiPi case to a Raspberry Pi and touchscreen for your own jukebox

Raspberry Pi Zero

What's that? A Raspberry Pi Zero as a media centre? Well, not exactly. If whole-house audio is your thing, the humble Zero is a great solution for creating casting points around the house. Simply add a DAC HAT, a case (dedicated DAC cases are available), and then install Shairport Sync (rpimag.co/shairport). Connect to any active speaker system and you will then see that system advertised as an AirPlay destination. Small, cheap, and energy-efficient.

A Raspberry Pi Zero W or 2 W with a DAC makes whole-house audio cheap and easy

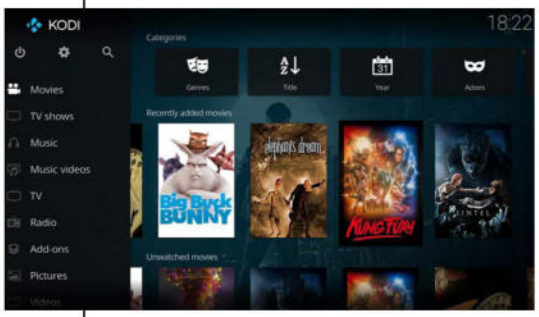
Essential media centre software

You've got all the physical goodies, now what about the software? PJ Evans covers your options

It's all very well having a fancy case, remote control, and AirPlay setups all around the house, but how are you going to make it all work together? There are many options for accessing and playing back your media, so it's worth taking time to look at the capabilities of the main contenders. There are three main use cases: local media playback (like an Apple TV), client playback (like Spotify), and audiophile-grade options. We'll look at our favourites from these categories.

LibreELEC & Kodi

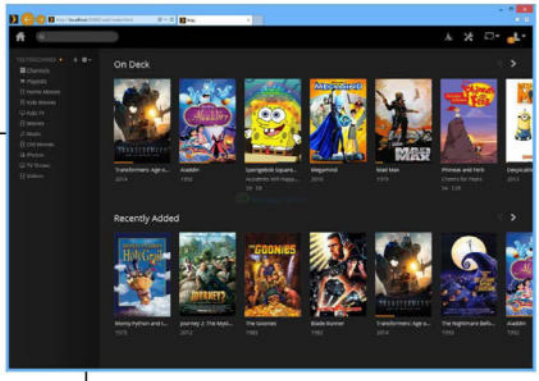
Kodi has been a long-term player in our first category, for local media. The software is mature and its plug-in architecture means that literally hundreds of add-ons can be installed to increase its functionality. It's a perfect solution for building your own Fire TV Stick or Apple TV rival, with a highly customisable interface and support for HD or UHD (4K) films, lossless music, and photo libraries. For Raspberry Pi, LibreELEC (libreelec.tv) is a dedicated operating system for Kodi which ensures smooth installation and no wasted resources. You can create a LibreELEC image directly from Raspberry Pi Imager (rpimag.co/software).



▲ Kodi with LibreELEC is your fastest route to a full-featured living room setup

Plex

Want to get to your media anywhere in the world with offline access? How about a dedicated music app or on-the-fly video transcoding? Best of all, would you like to do this without having to fiddle about with routers and DNS (Domain Name System)? Plex is a fantastic choice for a media server (plex.tv). It's not designed to be accessed directly, but supports web browsers, mobile apps, and there are clients for Apple TV and similar. If you want access outside the home and to download media, you'll need a paid-for subscription, but for hassle-free media management it's hard to beat. Don't mind routers and DNS? Have a look at our Jellyfin tutorial overleaf.



▲ Plex makes accessing your content from anywhere simple

▶ Build cheap AirPlay receivers with Shairport Sync

Shairport Sync

This 'headerless' server (it has no user interface) is squarely aimed at whole-house audio projects. It's a small service that, once installed, converts the Raspberry Pi into an AirPlay 2 target, supporting a wide range of casting devices, including iPhones. Better yet, it supports the multiroom sync protocol. Perfect for creating cheap audio 'nodes' using a Raspberry Pi Zero, DAC HAT, and speakers. Best of all, it's open source so completely free to use (rpimag.co/shairport).

Project Recipes

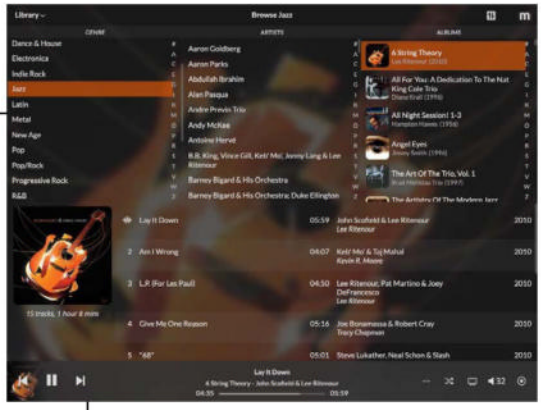
Set-top Box
- Raspberry Pi 5
- Argon ONE V3
- Recalbox & Kodi

Audiophile's Dream
- Raspberry Pi 5
- KKSB Case
- Raspberry Pi DAC Pro HAT
- moOde

Whole-House Audio
- Raspberry Pi Zero 2 W
- HiFiBerry DAC+ Zero
- Active speakers
- Shairport Sync

moOde

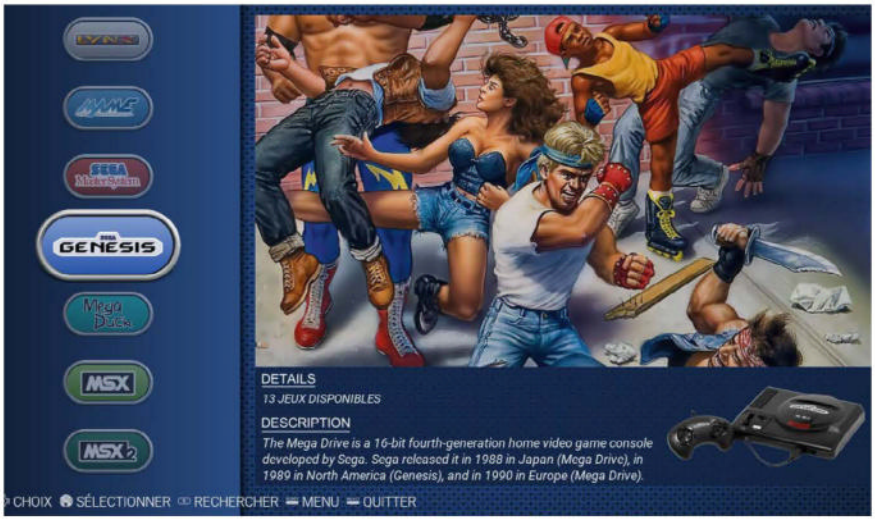
If you want to take your music listening seriously, Raspberry Pi has become a serious contender in the audiophile world. Companies such as Bang & Olufsen have produced DAC (digital-to-analogue converter) HATs using the finest components, turning the Raspberry Pi into a music-lover's delight. Even the lower end of the market produces some very good audio output. To get the best control, there are a few dedicated operating systems available such as moOde (moodeaudio.org), aimed at producing the best playback experience. It works great with a touchscreen or you can control it from a browser. See also the popular Volumio (volumio.com).



▶ moOde is one of the best choices for serious audio folk

> Raspberry Pi has become a serious contender in the audiophile world

▼ Want to add gaming? You can have Kodi and Recalbox on the same machine



Recalbox

Plot twist. When is a media centre not a media centre? When it's a retro gaming station! Recalbox (recalbox.com) is our favourite platform for playing retro games and among its many plug-ins is Kodi, the play-all media centre. So why not have your cake and eat it? Get a case with forward-facing USB (such as the Argon ONE V5) and you can have the nostalgic wired controller experience and have a full-blown media centre at your disposal as well.

Build a home media centre

Access your music, movies, and photos anywhere in your home or anywhere in the world, subscription free

Maker

PJ Evans

PJ is a writer, software engineer, and one of those people who drones on about a director's 'vision', constantly pausing films to discuss the scene composition.

mastodon.social/ @mrpjevans

▲ A small box that frees your media

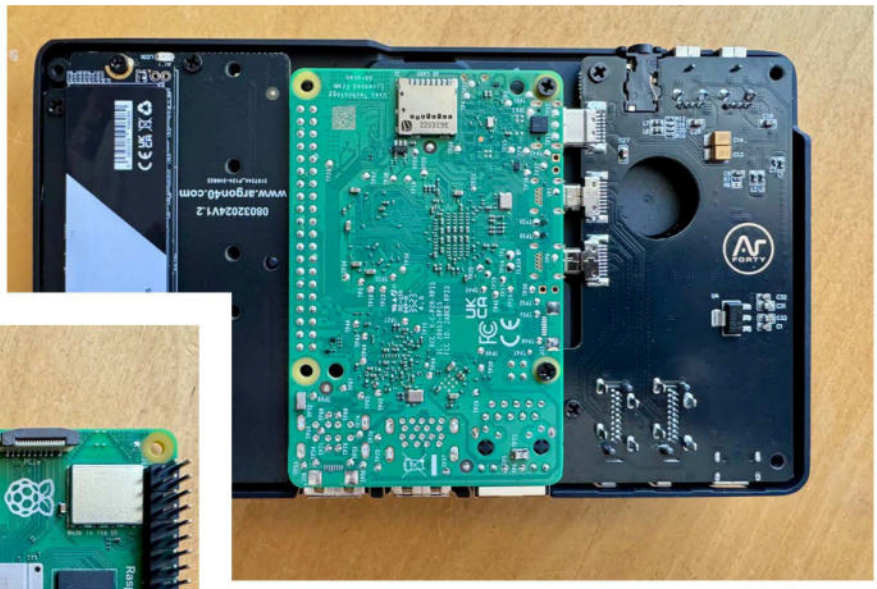
Raspberry Pi as a media centre has been a classic project since day one. As new models have been released, the capabilities of this minuscule device have increased exponentially. From a humble audio player to a fully capable video streamer, we are now in the era of the Raspberry Pi as a serious 4K media device. The options for setting up such a system are many, and things can get complicated. So here's a reference guide for setting up a great media system that is subscription-free and, with a bit of configuration, can be accessed anywhere in the world.

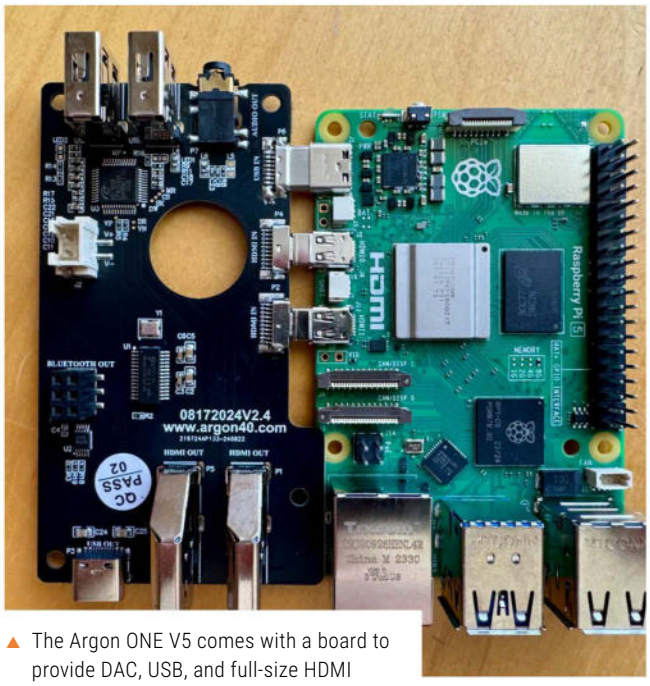
A perfect pairing

We're going to base our media centre around two great products: Argon's latest case, the Argon ONE V5, and for our software, Jellyfin. Argon has created a bit of a masterpiece with its new product; in fact, in our review we gave it a straight 10 out of 10. This is the perfect case for our project as it not only has built-in support for M.2 NVMe solid-state drives, but also comes with a DAC as standard. So if you want to use the device as not only a server, but a client as well, you're all set. Our media centre will be equally at home as both a server of images, audio, and video, as well as a 'set-top box' that sits in your living room.

Using an M.2 SSD means speed; and that can be important, especially with video. They are also more reliable than SD cards. We'll start by imaging the standard version of Raspberry Pi OS (64-bit) to our M.2 drive. We're using a 128GB NVMe drive from Argon, but you can use any size you like. To image the M.2 SSD, we recommend getting an inexpensive M.2 to USB adapter. You can then connect the drive just like a USB stick and use Raspberry Pi Imager. Select the recommended operating system and don't forget to configure the hostname, username, and password. You can configure Wi-Fi as well if you wish, but we strongly recommend a wired connection for judder-free viewing.

▼ The Argon ONE V5 case takes longer to assemble than most, but it's worth it





▲ The Argon ONE V5 comes with a board to provide DAC, USB, and full-size HDMI

Grab the screwdriver

Once your SSD is ready, proceed to build your case. The Argon ONE V5 is a solid piece of engineering and will take you a little time to assemble everything. Luckily, there are good instructions included. Be particularly careful with the wiring for the NVMe connector and the built-in fan. Assembly is a matter of connecting the case's board to the Raspberry Pi, screwing them both to the chassis, adding the M.2 SSD, and then connecting the wiring. Once done, the bottom and top covers can be added. Optionally you can also install the small OLED screen accessory, which can provide rolling status information such as CPU usage, temperature, and IP addresses.

Typing time

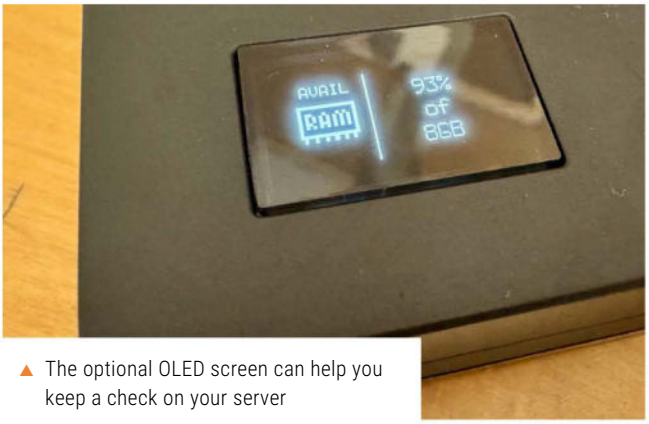
You should now be able to hook up your media centre to a monitor or TV using the full-size HDMI connectors. You also need to decide at this point how to connect to the Raspberry Pi for further configuration. Either connect a keyboard and mouse, or use Secure Shell (SSH) to access Raspberry Pi, as everything can be done from a terminal screen. The system should reboot twice as it sets up and then drop you into the familiar Raspberry Pi desktop. As ever, the first job is to make sure everything is up to date. Open a terminal window (or SSH in) and run:

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

This will ensure you have the latest software and security patches installed.

Before we install Jellyfin, Argon provides software to help manage the various features of the case, such as the power button and the display. Open a terminal window and run the following command:

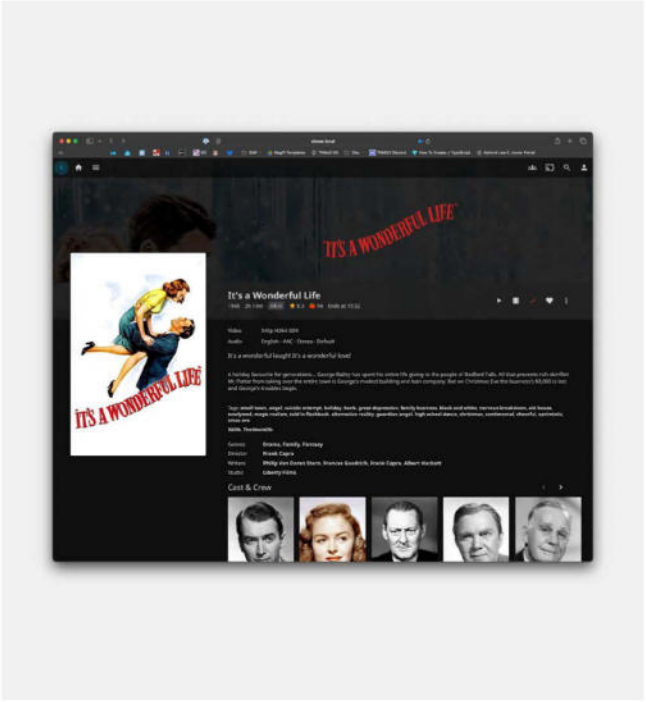
```
$ curl https://download.argon40.com/argon1v5.sh
| bash
```



▲ The optional OLED screen can help you keep a check on your server

This will set up everything for you. If you have the screen installed, it should immediately start showing information. Now you can use the front button to safely switch your computer on and off. If all is well, we can go ahead and install Jellyfin itself. Run the following:

```
$ curl https://repo.jellyfin.org/install-
debuntu.sh | sudo bash
```



▲ Jellyfin automatically populates metadata

This runs a remote script that makes installing Jellyfin a breeze. It'll take care of everything for you. At the end, you'll see confirmation that the Jellyfin service is running and you're ready to configure everything.

The next thing to do is think about is how you want to organise your media. To make things as easy as possible, we recommend creating a root directory as follows:

```
$ sudo mkdir /Jellyfin
```

...and creating directories underneath it such as:

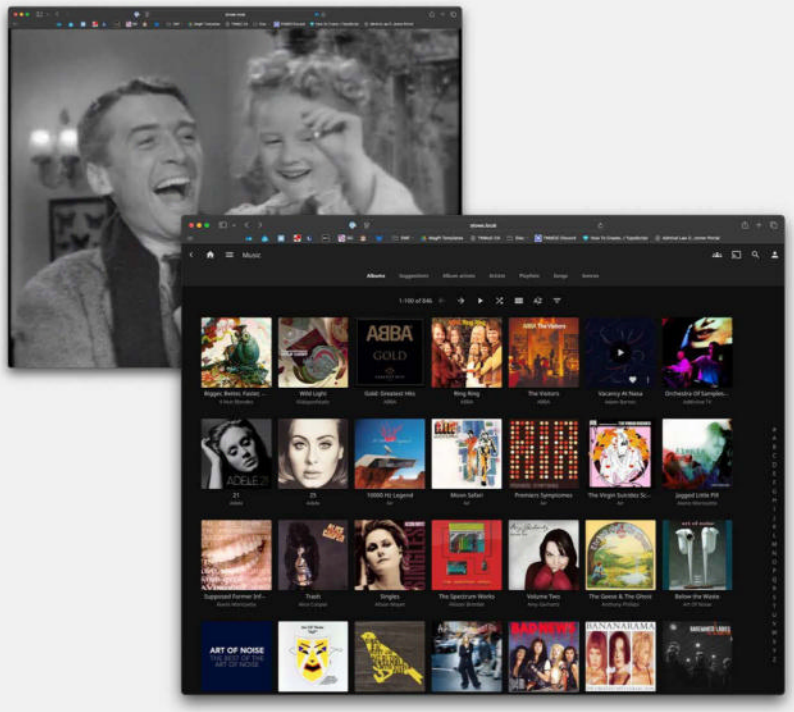
```
$ sudo mkdir /Jellyfin/Music
$ sudo mkdir /Jellyfin/Movies
$ sudo mkdir /Jellyfin/TV
```

You can choose whatever you want. To get files onto the server, copy them into your home directory and then move them into the Jellyfin directory like this:

```
$ sudo mv /home/pi/Music /Jellyfin/Music
```

Start with a small amount of media as you can add more at any time.

▼ Watch classic movies anywhere, including public domain material from archive.org



▲ Jellyfin provides a beautiful interface for your favourite media

Adding media

Nearly all audio purchased online is DRM-free and can be added to your media library. Jellyfin can handle all the common formats you will come across.

Nearly all movies are DRM-protected and will not be playable by Jellyfin. However, there is a wealth of public domain content available at sites such as The Internet Archive (**archive.org**) which you are free to download and enjoy.

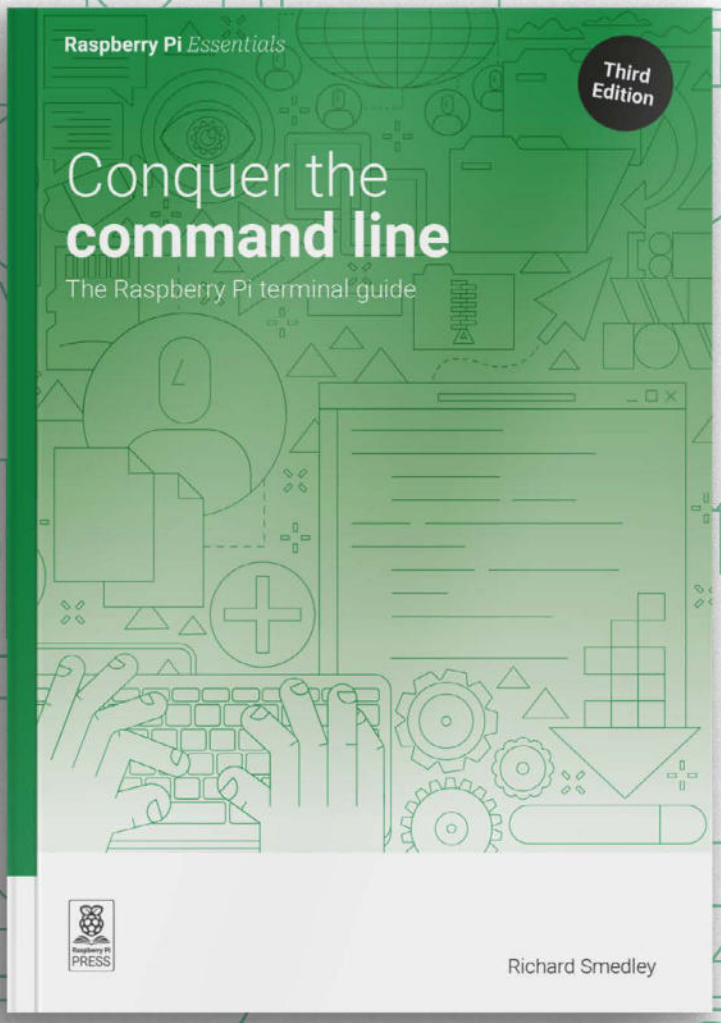
Scan and go

Jellyfin has installed a local web server which you use to access your media. To find it, open any browser on the same network and go to http://<your server name>:8096/ (where <your server name> is replaced by the hostname you gave your server. You now step through the creation of the admin account and the adding of your first media library. Choose a category (such as 'Movies') and select the directory you created and populated in the previous step. Jellyfin will help you categorise and populate your media with metadata from various internet sources. You can always make corrections if it gets it wrong.

Once setup is complete, Jellyfin will scan your media in the background, so expect things to look not-quite-right at first, but then you'll see album covers and film posters as it catches up. Now you have a media server! Any computer with a browser can access your media centre and enjoy the sights and sounds it provides. Better yet, you can install either the official or one of many third-party Jellyfin clients for both iOS and Android. Just point them at your server and you have all of your media in the palm of your hand.

Go remote

The final step is to get your media available anywhere. There are different ways of doing this. The simplest is to get either a static IP address (or use a dynamic IP service such as DuckDNS) and then map a port route from your router to your media server. You can then use your new address (as provided by DuckDNS) as your server access point. Your server is now available anywhere. This isn't the most secure approach, though. A better method is to set up Wireguard VPN as a secure way of accessing your home network from afar and then connecting over its encrypted connection. Find out how to do this in issue 145 (rpimag.co/145).


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

Part 01

Conquer the command line

We take a look around and discover that things aren't as strange as they might appear

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

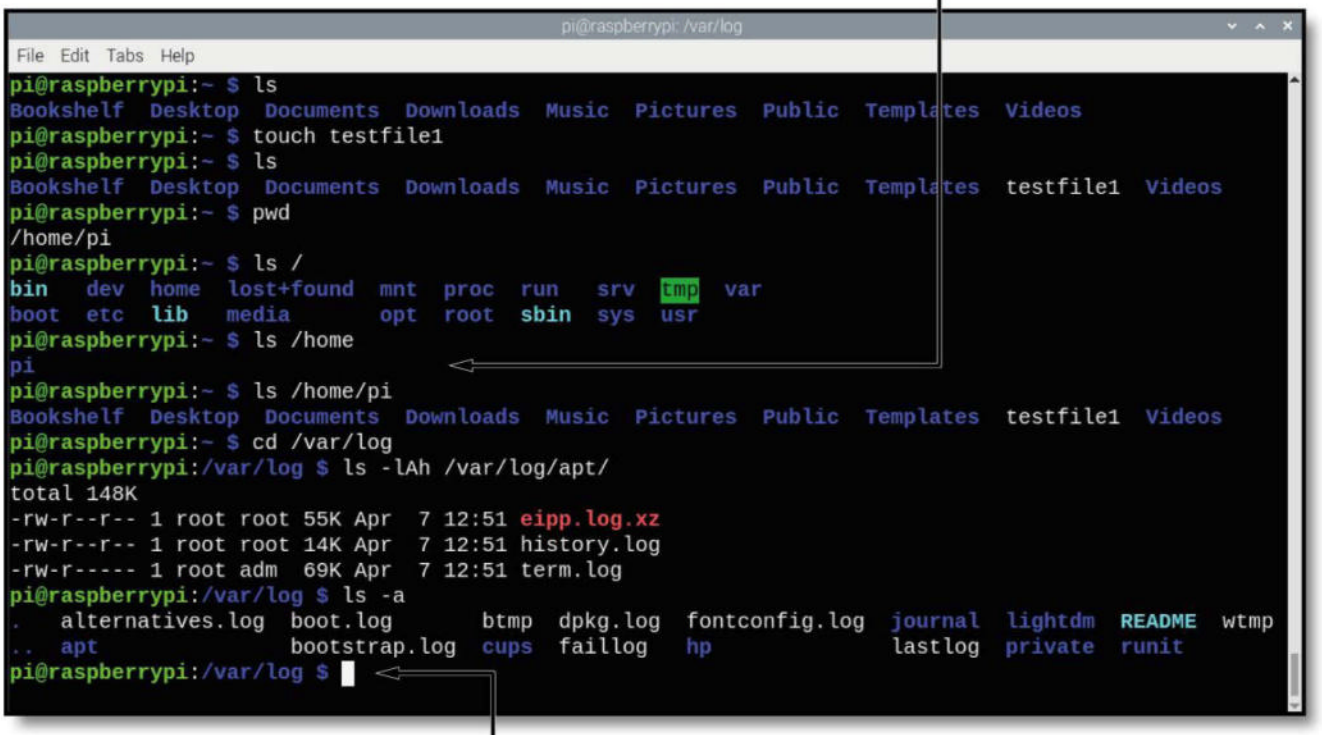
It's not a throwback to the past, but a quick and powerful way of getting your Raspberry Pi to do what you want, without all that repetitive strain injury-inducing menu chasing and icon clicking. The command-line interface was a great step up from manually toggling in your instructions in octal (base-8), using switches on the front of the machine! Graphical user interfaces (GUIs) brought friendly visual metaphor to the computer, losing some power and expressiveness. With Raspberry Pi, you can get the best of both worlds by knowing both: after reading through this guide, you'll soon be as comfortable at the command prompt as you are at your desktop.

By default, Raspberry Pi OS boots you straight to a GUI, though you can change this behaviour in the settings. The command-line environment is still there: hold down the CTRL+ALT keys and press F1 (the first function key on the keyboard), and you'll arrive at a virtual console. Press CTRL+ALT+F2 through to F6 and you'll find five further consoles waiting for you to log in.

You can drop into these any time you like, but for now press CTRL+ALT+F7 and you'll be back in mouse and menu land. The command line is also available through a program called a terminal emulator (sometimes referred to as a term). You'll also find people referring to the shell, which is the program that runs in the terminal and interprets your commands. On Raspberry Pi and most GNU/Linux systems, the shell is a program named Bash (the Bourne-again shell). Don't worry about the distinction between the terminal and shell for now; just click on the icon at the top of the screen that looks like a black television screen, or go to Accessories > Terminal in the Raspberry Pi menu: the Terminal now awaits your commands.

You'll soon be comfortable at the command prompt

The command line is only a click away: it is called Terminal and you can find it under Accessories in the menu



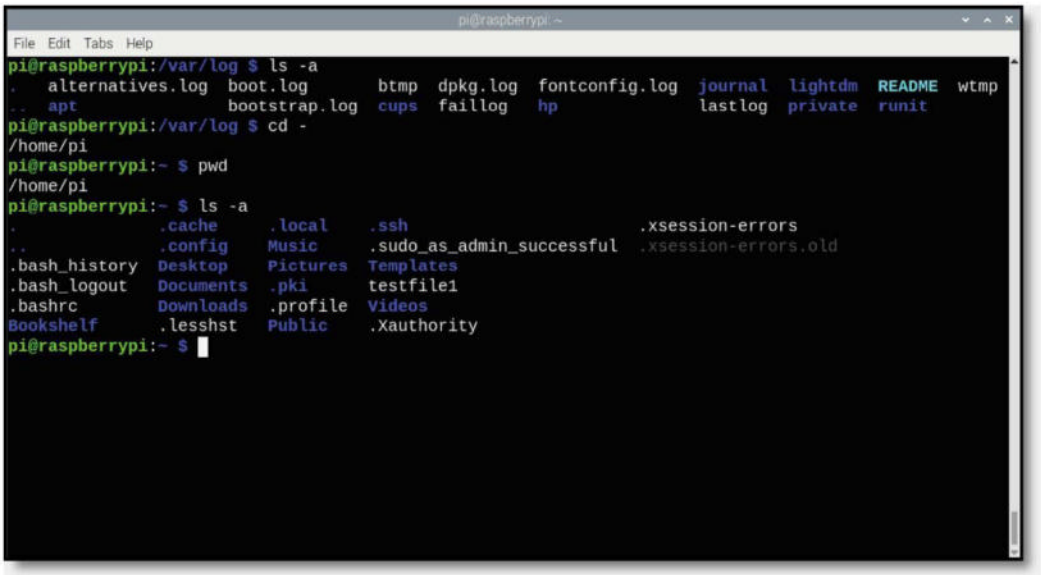
Commands are terse, but, once learned, they're a quick way of navigating and reading your files and folders

Look around

If you're used to looking at files and folders in a file manager, try to clear your mind of the icons and concentrate on the names. Type `ls` and press RETURN. On a fresh Raspberry Pi OS installation, you'll just see a few directories, including Bookshelf. Type `ls Bookshelf` and you'll see a listing of what's in it. Commands like `ls` are not cryptic (at least not intentionally) but they are terse, dating back to a time when the connection to the computer was over a 110 baud serial line, from an ASR 33 teletype terminal. If you think it's strange to be defined by 50-year-old technology, just remember that your QWERTY keyboard layout was reputedly designed both to stop mechanical typewriter keys jamming, and to enable salespeople to quickly type 'typewriter' using the top row!

TAB COMPLETION

You don't need to type all of `ls Bookshelf` (for example) – after typing `ls B`, hit the **TAB** key and it will auto-complete. If you've more than one file beginning with B, they'll all be listed and you can type more letters and hit **TAB** again.



▶ **Figure 1:** `ls -a` shows (dot) files in your listing that are usually hidden from view

PRESS RETURN

To save repeating it in the text, we'll confirm here that each time you type in a command, you need to hit the **RETURN** or **ENTER** key at the end, to tell the shell that you've issued a command.

File path

You can list files and folders anywhere in your system (or other connected systems) by providing the path as an argument to your command. The path is the folder hierarchy: on a Windows computer, in a graphical file browser, it starts with 'My Computer'; on your Raspberry Pi it starts at /, known as root, the root of your file system. Try entering `ls /` – again we get terseness, with names like bin (short for binary) which is the directory where many programs are kept (try `ls /bin` to see its contents). `ls /dev` shows hardware devices in Raspberry Pi. Try `ls /home` – see your username there? That's you! In this guide, we'll show this as 'pi', but you should have chosen your own username when you set up your Raspberry Pi. If you have created extra users, they'll all be listed there too: every user gets their own home directory; ours is the /home/pi folder, the working directory where you begin each terminal session. Earlier, with Bookshelf, we used the relative path (the absolute path would be /home/pi/Bookshelf) because we were already home. If you need to check your current working directory, type `pwd` (print working directory).

There's no place like ~

For any logged-in user, their home directory is abbreviated as `~` (the tilde character). Type `ls ~` and you'll see. There's apparently not much in your home directory yet, but Raspberry Pi OS keeps a lot hidden from the casual glance: files and folders beginning with a dot, known as dot files, contain configuration information for your system and its programs. You don't need to see these files normally, but when you do, just ask ls to show you all files with a command switch (see Figure 1). You can do this with either the full switch `--all`, or the abbreviation `-a` like so: `ls -a ~`.

Traversing the pathways of the directory hierarchy can be easier from the command line than clicking up and down a directory tree, particularly with all the shortcuts given. Your `ls -a` showed you . and .. as the first two directories; these shortcuts represent the current and the parent directory respectively. Try listing the parent directory from /home/pi with `ls ..` – entering `ls ../../` will show you two layers up. If you want to list the hidden files without the . and .. appearing (after all, they're present in every directory, so you don't need to be told), then the switch to use is `-A`.

Before we move on to other commands, try chaining switches together: `ls -lh ~`. Here's what is happening:

```
-l
```

...gives you more information about the files and folders.

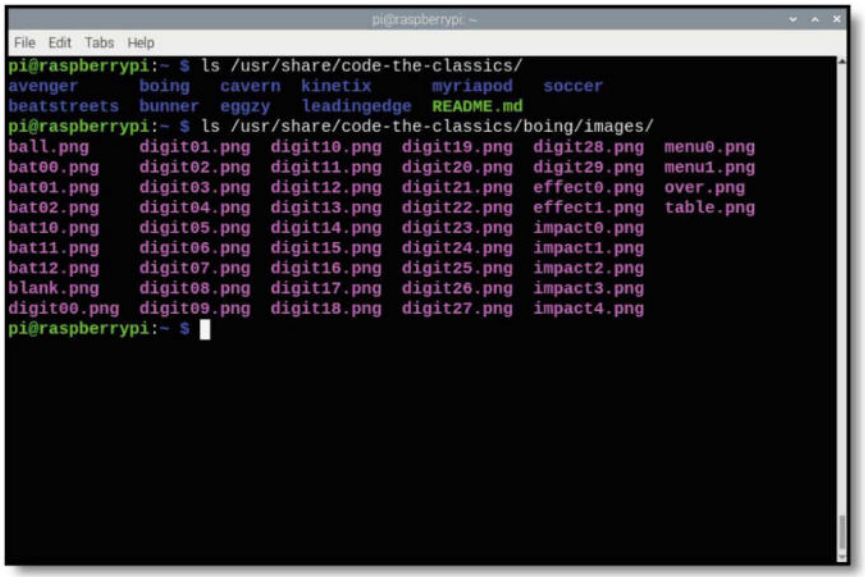
```
-h
```

...changes the units from bytes to kB, MB, or GB as appropriate.

We'll look at some of the extras the `-l` listing shows you in more detail in a future part of this guide.

READ THE MANUAL

Help is included, with man(ual) pages, but they can be a little overwhelming. Use them to check out some extra options beyond the switches like `-a` we're using here. To read the `ls` man page, type `man ls`.



▶ Who needs icons when you can fit a listing of files into a small window? Coloured fonts indicate file types

Time for change

That's enough looking: let's start moving. `cd` is short for change directory, and moves you to anywhere you want in the file system: try `cd /var/log` and have a look (`ls`, remember). Files here are logs, or messages on the state of your system that are saved for analysis later. It's not something you'll often need to think about: Raspberry Pi OS is a version of an operating system (Debian) that also runs across data centres and supercomputers, where problem monitoring is very important. It is, however, useful to know, particularly if you have a problem and someone on a forum advises you to check your logs.

`cd ~` will take you where you expect it. Try it, then `pwd` to check. Now try `cd -` (that's a hyphen), the '-' is a shortcut for 'wherever I was before I came here'. Now we've looked around, we can move on to beginning to do things to our files, as we'll explore next issue.

cd is short for change directory, and moves you to anywhere you want in the file system

Pico drum machine

Play drum samples at the press of a button

Maker

Phil King

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

philkingeditor.com

YOU'LL NEED

- Raspberry Pi Pico (any model) with GPIO headers soldered on
- DF Mini Player
- microSD card
- Mini speaker (up to 3W)
- 2 × half-size breadboards (or one full-size)
- 6 × momentary push buttons
- 19 × pin-to-pin jumper wires
- Drum samples

While the first ever drum machine is considered to be the Rhythmicon, developed by Leon Theremin (yes, he invented that instrument too) in the early 1930s, digital drum machines came to the fore in the 1980s with the likes of the Linn LM-1. The latter cost the equivalent of over $19,000 today, and yet we can now create a DIY drum machine using a $5 Raspberry Pi Pico.

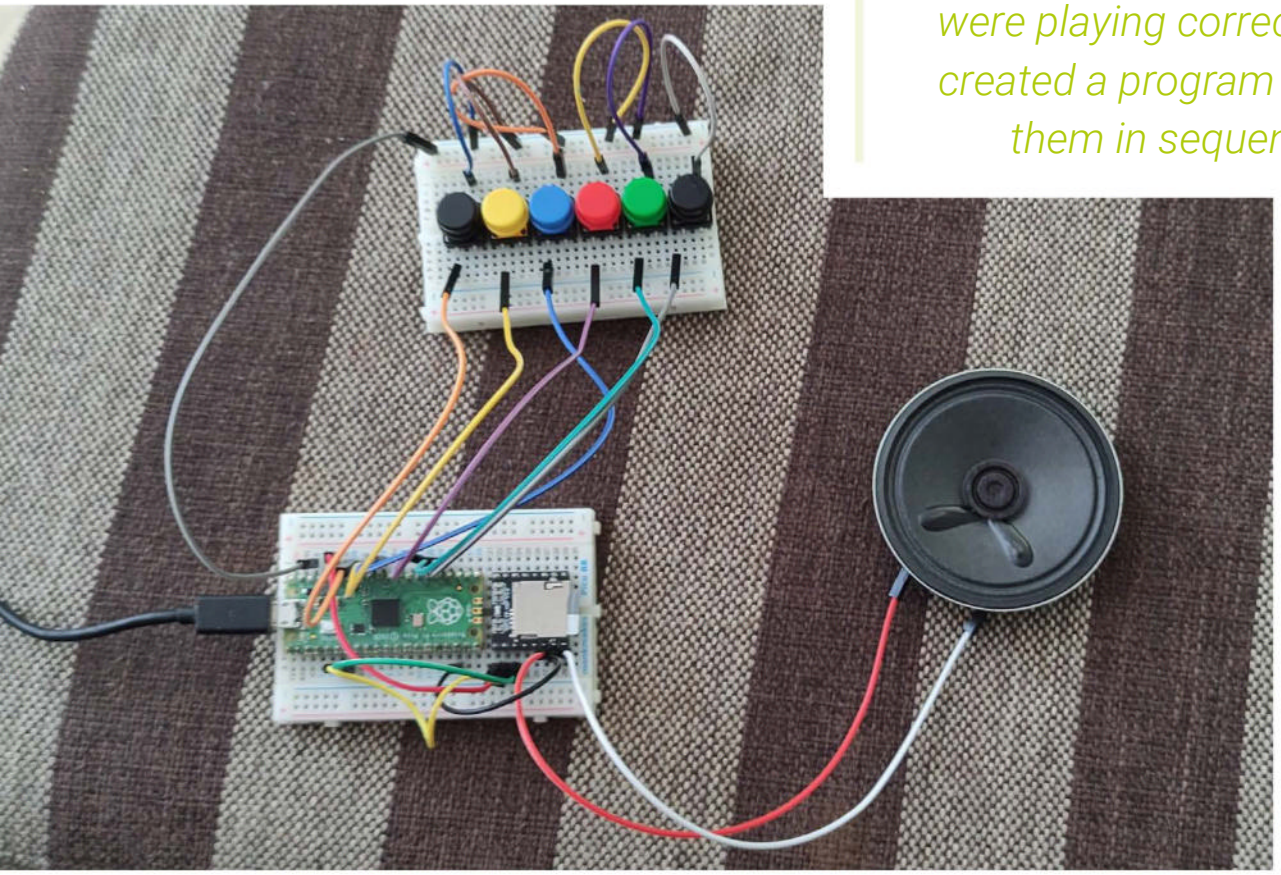
Inspired by Arnov Sharma's project on Hackster (rpimag.co/picodrummachine), your author decided to try to create a Pico-powered drum machine using a DF Mini Player to play the drum samples. While Arnov's version uses a custom-created PCB for the buttons, we decided to keep it simple with a breadboard-based design, the downside being a spaghetti of jumper wires to connect everything up. Still, the principle is the same: you press different push buttons to trigger different drum samples played by the DF Mini Player, outputting audio to a mini speaker.

Building the circuit

To make it easier to wire the buttons to Raspberry Pi Pico, we opted to put the buttons on a second half-size breadboard, as you can see in the Figure 1 wiring diagram. Each four-legged button spans the division in the middle of the breadboard, with the pins on one side connected to a ground rail. The other side of each button is connected to a GPIO pin on Pico – we used GPIO 28, 27, 26, 21, 20, and 19 – pulled up (in the code). So when you press the button, the GPIO pin is pulled low, and the code senses this and triggers the DF Mini Player to play the relevant sample – more on that later.

The DF Mini Player is placed on the same breadboard as Pico, connected via UART RX and TX pins, along with 3.3V power and ground. We use its speaker output pins to connect to a mini speaker (ours was 2W, 8Ω).

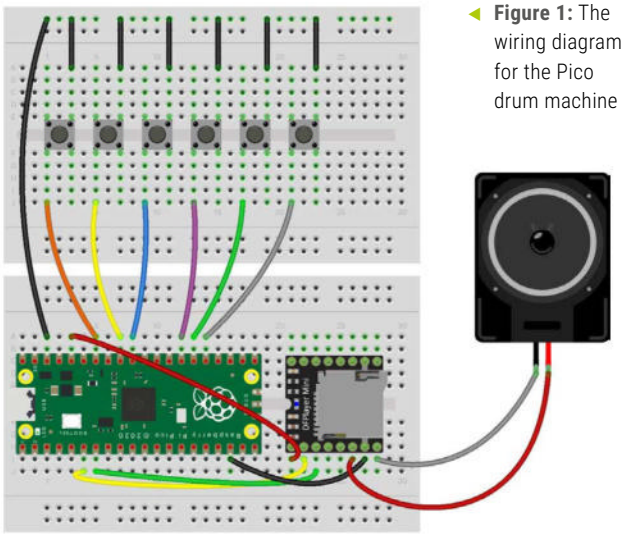
To test our drum samples were playing correctly, we created a program to play them in sequence



▲ We used two breadboards for our Pico drum machine, although the wiring is still a bit spaghetti-like

To play some sounds, we need some drum samples. There are lots of free, open-source ones available online. We got ours from here: rpimag.co/drumsamplesgh. We found that some of the samples were a little long – and the DF Mini Player can only play one file at a time – so we opted to edit them in Audacity (available from audacityteam.org). If, after trimming the end of a sample, you find it ends too abruptly, you can always apply a fade-out effect to the end.

The microSD card must be formatted as FAT32, so we erased it in Raspberry Pi Imager: select Choose OS > Erase, then Choose Storage and select the card. In addition, the files must be named 0001, 0002, 0003, etc., with the relevant suffix – you can use MP3 or WAV files. As we used the latter, ours were named 0001.wav, 0002.wav... 0006.wav. We found that it doesn't matter whether you put them in a folder or not.



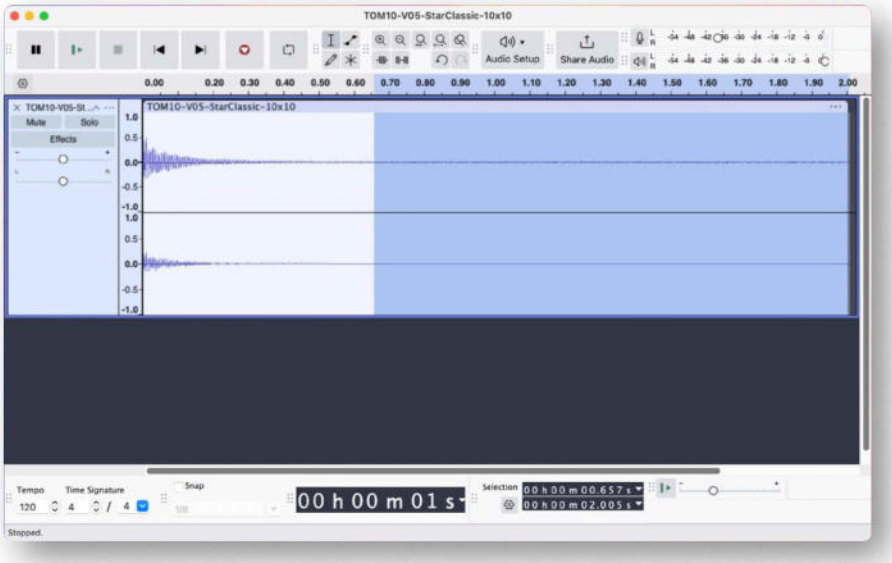
◀ **Figure 1:** The wiring diagram for the Pico drum machine

Warning!

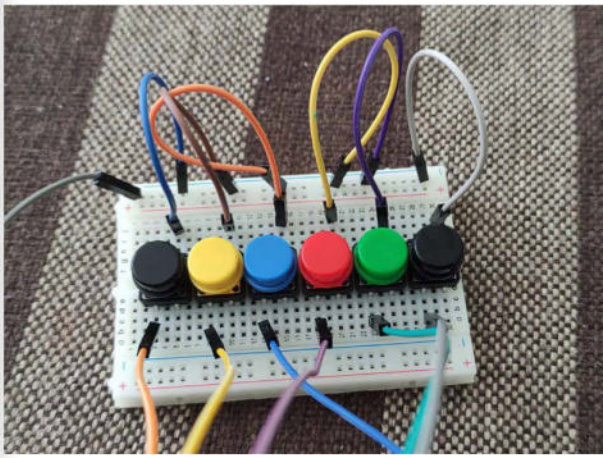
Hot solder

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



We edited our drum samples in Audacity to reduce the length of some of the longer ones



A close-up of the buttons. Each has one pin wired to the ground rail, and a pin on the other side wired to a GPIO pin on Pico

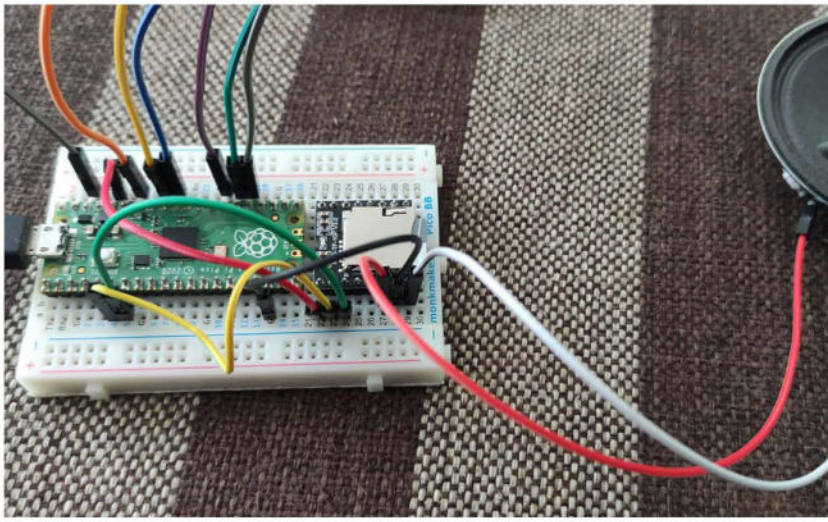
Coding it

Unlike Arnov, who programmed his drum machine in C, we opted to use MicroPython. For this, we made use of Stewart Watkiss's DF Mini Player library at rpimag.co/dfminilibrarymp. Just download the dfplayermini.py script from there and then, using the Files tab in Thonny IDE, right-click and upload the file to your connected Pico (which already has MicroPython installed). You can then call the library in your programs.

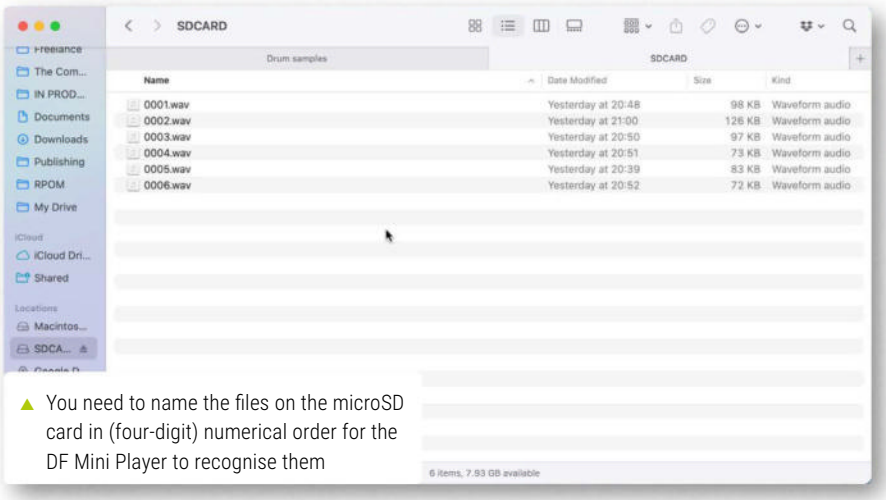
To test our drum samples were playing correctly, we created a program (play_drums_seq.py) to play them in sequence:

```
from dfplayermini import DFPlayerMini
import time

player1 = DFPlayerMini(1, 4, 5)
player1.reset()

print ("Set SD Card")
read_value = player1.select_source('sdcard')

print ("Set Volume 30")
read_value = player1.set_volume(30)

read_value = player1.query_num_files()
print (f"Num files {read_value}")

for i in range(6):
    print("Play",i+1)
    read_value = player1.play(i+1)
    time.sleep(1)
```

After importing the libraries, we set up a `player1` object to work with UART 1 on GPIO pins 4 and 5. We then reset the DF Mini Player so it's ready to start communicating. We select the SD card as the audio source, then set the volume – we maxed ours up to 30. We then query the number of files on the card (it should be six), before playing each one in turn.



Along with Pico, we placed the DF Mini Player on another breadboard and connected it to a mini speaker



▲ You need to name the files on the microSD card in (four-digit) numerical order for the DF Mini Player to recognise them

Speaker soldering

The SPK_1 and SPK_2 pins on the DF Mini Player should be connected to the positive and negative terminals (or wires), respectively, of your mini speaker. While you can get away with wrapping the stripped ends of the jumper wires around the terminals, it's best to solder them in place for a more secure and reliable connection. The sound from our mini speaker was a little tinny, so you might get more bass out of it by mounting it in some kind of enclosure.

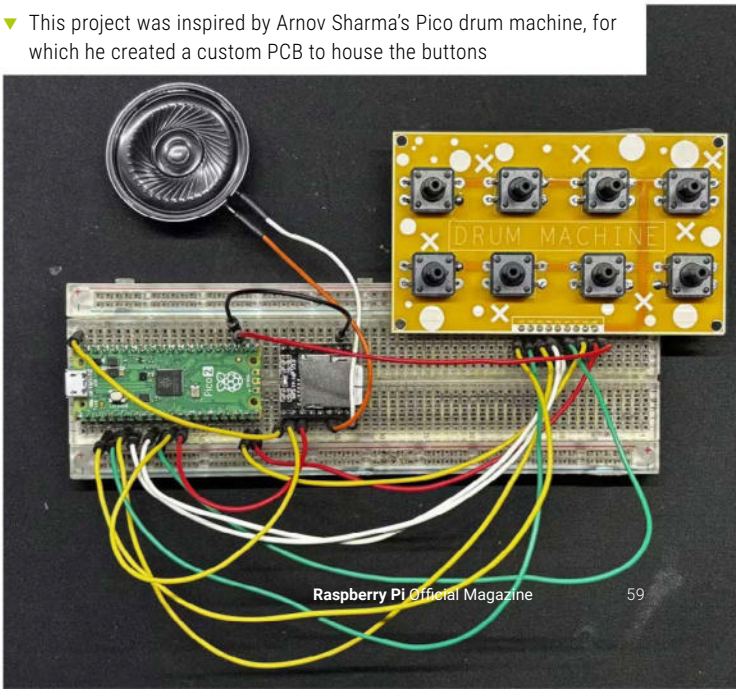
Next, we adapted this to create our main program to read the button presses and trigger the sounds accordingly:

```
from dfplayermini import DFPlayerMini
from machine import Pin
import time

player1 = DFPlayerMini(1, 4, 5)
player1.reset()

print ("Set SD Card")
read_value = player1.select_source('sdcard')

print ("Set Volume 30")
read_value = player1.set_volume(30)

read_value = player1.query_num_files()
print (f"Num files {read_value}")

# Define GPIO pins for buttons
button_pins = [28, 27, 26, 21, 20, 19]

# Initialize input pins
buttons = [Pin(pin, Pin.IN, Pin.PULL_UP) for
pin in button_pins]

# Main loop
while True:
    for i in range(len(buttons)):
        # Read button state
        if buttons[i].value() == 0:
            player1.stop() # stop current sound
            read_value = player1.play(i+1)
        time.sleep(0.01) # Debounce delay
```

Here, we add a line at the top to import the `Pin` method from the `machine` library. We create a list to set the GPIO pins for the buttons, then initialise them as inputs with the pin pulled up. In the main loop, we read the button state and then, after stopping any currently playing sound, play the relevant drum sample. We added a very short debounce delay to prevent a button press causing multiple triggers.

Hands up: the performance wasn't as good as we'd hoped, with a noticeable lag between pressing a button and the sound being played. As already mentioned, the DF Mini Player can only play one file at a time, which is a major limitation for a drum machine. An alternative would be to use an I2S-based audio board such as the Waveshare Pico-Audio to play the drum sounds. Still, our little Pico drum machine does work and you could use it to trigger other samples, such as spoken phrases or funny noises. Alternatively, using four of the buttons, you could try out Stewart Watkiss's MP3 player project (rpimag.co/dfminimp3player) to play songs stored on the microSD card.

▼ This project was inspired by Arnov Sharma's Pico drum machine, for which he created a custom PCB to house the buttons



Part 03

Setting up email in RISC OS

What email clients does RISC OS offer? Let's take a look and find out

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 Macs, retro gaming/computing, and anything else that pays.

@ijosborne

Now we've been over the basics of RISC OS on Raspberry Pi, let's take a look at how to set up email. The easiest way to use email on your RISC OS Raspberry Pi is to open the !NetSurf web browser and point it at your email service's web client. This is the case for web-based email services such as Google Mail or Apple iCloud, but also for your ISP's webmail service. After all, if you only occasionally need to use email from your RISC OS Raspberry Pi, setting up a dedicated email client is hardly worth the effort.

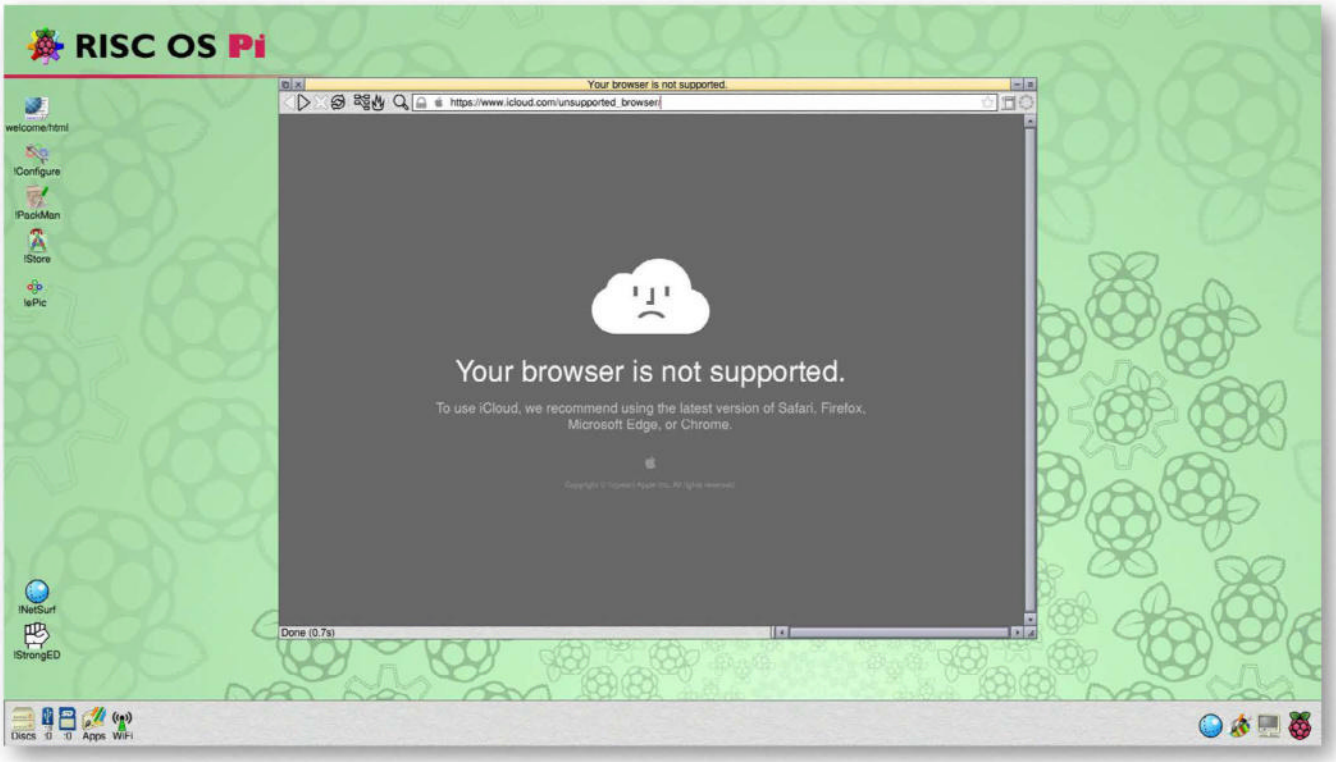
First of all, go to the web address for the email service in question. For the one offered by your ISP, go to the service provider's home page and navigate to the webmail page from there. Input your username and password, and you can use the email services just like you do from any other computer browser – almost. Unfortunately, !NetSurf doesn't fully support JavaScript, so web email services which use it extensively might not work correctly on the RISC OS browser, even if you've enabled it. If this is the case with your email provider of choice, read on – we'll take a look at setting up a client next.

Email clients

There are a few free clients available. Let's check one out. Open the !Store app, and in the PlingStore window, click Categories. Tick the Network/Internet option, and click the Products List icon, the one showing the 'A' at the top of the PlingStore Categories window. Another window opens, showing a list of available apps in the Network/Internet category. Find Pluto by Martin Avison – it's free. Double-click on it to open the product window, and then click the Get icon to download it. When it arrives (as a zip file), a window opens. Drag the !Pluto icon – and

Unfortunately, !NetSurf doesn't fully support JavaScript

▼ Even with JavaScript enabled, !NetSurf might give you problems when using webmail



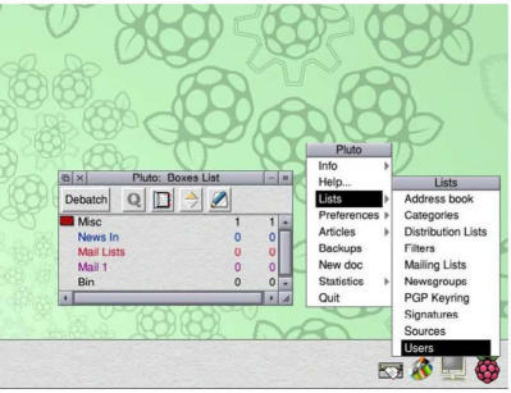
anything else you want to keep – onto a directory on your SD card. The obvious thing to do is drag the app into the Apps folder and the ReadMe (if you want it) into the Documents folder. If you regularly save ReadMe files, better rename it – after moving it, middle-click > File > Rename.

Double-click the !Pluto icon (the one you just moved, not the one in the downloaded zip file) to open the app, and if you get a message about directories being missing, click the Create button. The directories are created, and the app launches. As you can see, you've one email to read. Unsurprisingly, it's from the developers welcoming you to the app. There's nothing else in there, and an email client isn't much use until you've set it up to work with your email accounts, so let's get started on doing just that.

Enabling JavaScript on !NetSurf

!NetSurf is currently a very primitive browser with very limited JavaScript support. Full support is planned, but is a long way off.

JavaScript is disabled by default, but if you want to use it, middle-click !NetSurf in the icon bar at the foot of the screen, and left-click Choices. In the configuration window that appears, click Content to open another window. In this new window, uncheck the Disable JavaScript box and click the Set button.



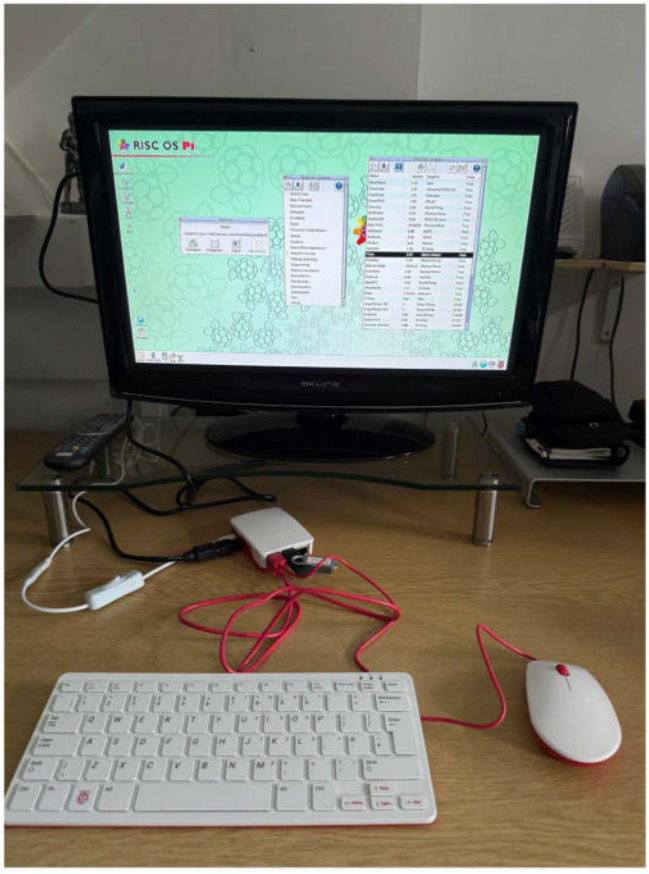
▲ Adding a user in the !Pluto email client

Creating a new user

First we must create a new user. Middle-click the !Pluto icon bar icon, then go to Menu > Lists > Users. The Users List appears, and there's only one name in it – the ubiquitous A.N. Other. Middle-click that and choose Edit User. In the window that pops up, add your details. They're fairly straightforward – the User ID is the part of your email address before the @ symbol, and @Domain is the part after it. After that, just customise it to your own requirements.

Yet as seasoned email tinkerers will know, this isn't enough. What about all those other stats? Well, they're not added in the !Pluto app; you need another application to handle the actual fetching and sending of emails (and News, if you want to set it up to receive newsgroups too). These are called Transport apps, and we need to download one before we can proceed. To read more about !Pluto, and Martin Avison's other applications, go to **avisoft.force9.co.uk**.

There are a few Transport apps on offer, but for the purpose of this guide, we'll use !AntiSpam. First of all, download it from **aconet.nl/antispam**. It's not in the !Store app, but you can get it using !NetSurf. Open the website, and near the top is a link displaying the current version number, its release date, and how large a download it is (let's avoid using the beta of the next version for now). Click it and a zip folder arrives, which you can move to wherever you want, before opening it and moving !AntiSpam and its ReadMe file to wherever you want, then deleting the zip. If this is your first rodeo with RISC OS, doing these straightforward tasks might seem awkward to you, but you soon get used to its idiosyncrasies.



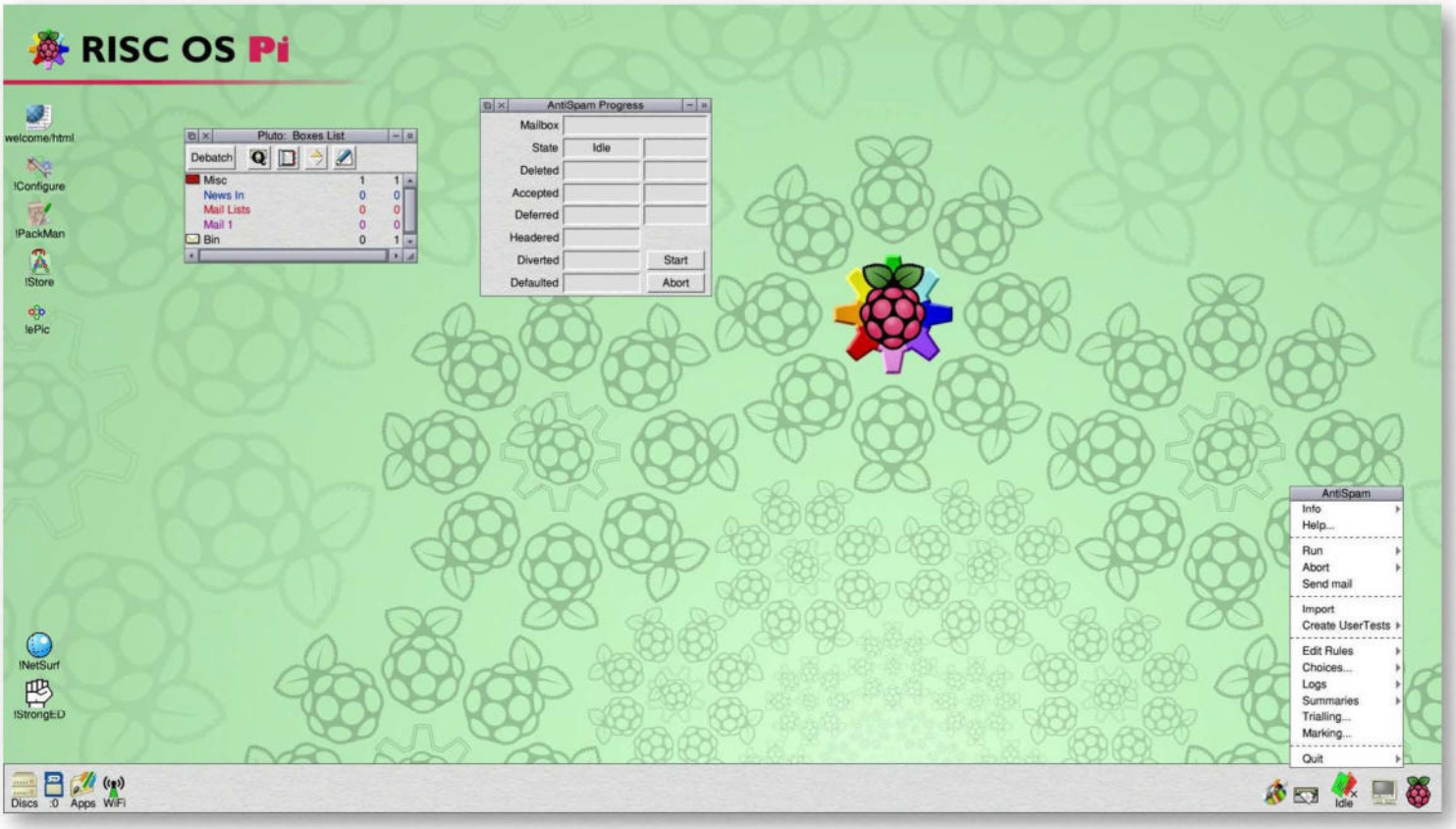
▲ Download an email client from the RISC OS !Store app

Setting up !AntiSpam

Before you can run !AntiSpam, you need to install another app called !SysLog. If you use RISC OS Select, Adjust, or Six, you already have !SysLog, but if you launch !AntiSpam and get a message

Assistance from !StrongHelp

The !StrongHelp app is 'a simple hypertext program for reading and designing on-screen manuals'. If you get stuck while configuring your RISC OS apps, download !StrongHelp from **rpimag.co/fjg**. Most app help files are stored inside the app's directory so are readily available, but if you need something else, there's a large collection of manuals at **rpimag.co/riscosmans**.



Which RISC OS?

If you ever need to check which version of RISC OS you have running on your Raspberry Pi, middle-click the raspberry icon in the toolbar (the Task Manager icon), drag your pointer to Info and then to the triangle after Info. A window pops up, telling you which RISC OS you're running, and when it was released.

saying, '!SysLog application not seen by filer', you can download !SysLog from rpimag.co/syslogrisc. To install !SysLog in the recommended location, double-click on your SD card icon, then drag !SysLog onto the window that pops up. The app will now always be there for you when you boot up. According to its developers, 'SysLog is a freeware program for RISC OS computers. It provides a module which offers a central logging facility that can be used by programs to log information at various levels of importance.' But all we need to concern ourselves with at the moment is it lets !AntiSpam work.

Now it's time to do some configuring. If it's not already running, launch !Pluto and middle-click its icon in the icon bar. Go to Preferences > News/Mail, and a window appears. Click the Mail Transport window, and in the drop-down, choose AntiSpam. Launch !AntiSpam, and you're invited to configure it. We have to play it by ear a little here, as the settings you configure depend on your ISP or web-based email provider. There's one thing we'd like to bring to your attention, though. If you get something wrong and need to go back to the !AntiSpam configuration window, middle-click the icon bar !AntiSpam icon, and in the menu, select Choices without going into one of its submenus. The configuration window appears, and you can review your settings once again. When you're done, you should have email up and running on your RISC OS Raspberry Pi.

▲ If you need to reconfigure !AntiSpam, middle-click its icon and click on Choices

Newsgroups on !Pluto

It's also possible to use the !Pluto app for newsgroups, but you'll need a different transport app. Go to **heenan.me.uk/acorn** and download NewsHound. With !Pluto running, middle-click the icon bar icon and go to Preferences > News/Mail. Click the News Transport window, and in the pull-down, select NewsHound.

To switch between emails and newsgroups in the !Pluto app, click the top-left icon in the !Pluto window. The postbox icon is Send for both. Maybe we'll do a full tutorial on this in a future issue.

Unusual tools: Attack of the Krone

The **ni**che tool that every **n**etwork e**ngin**eer k**n**ows about

Maker

Dr Andrew Lewis

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

lewiswork.etsy.com

QUICK TIP:

Normally, the wrong punch-down tool won't fit in the wrong type of connector. A Krone tool won't fit into a 110 terminal, and vice versa.

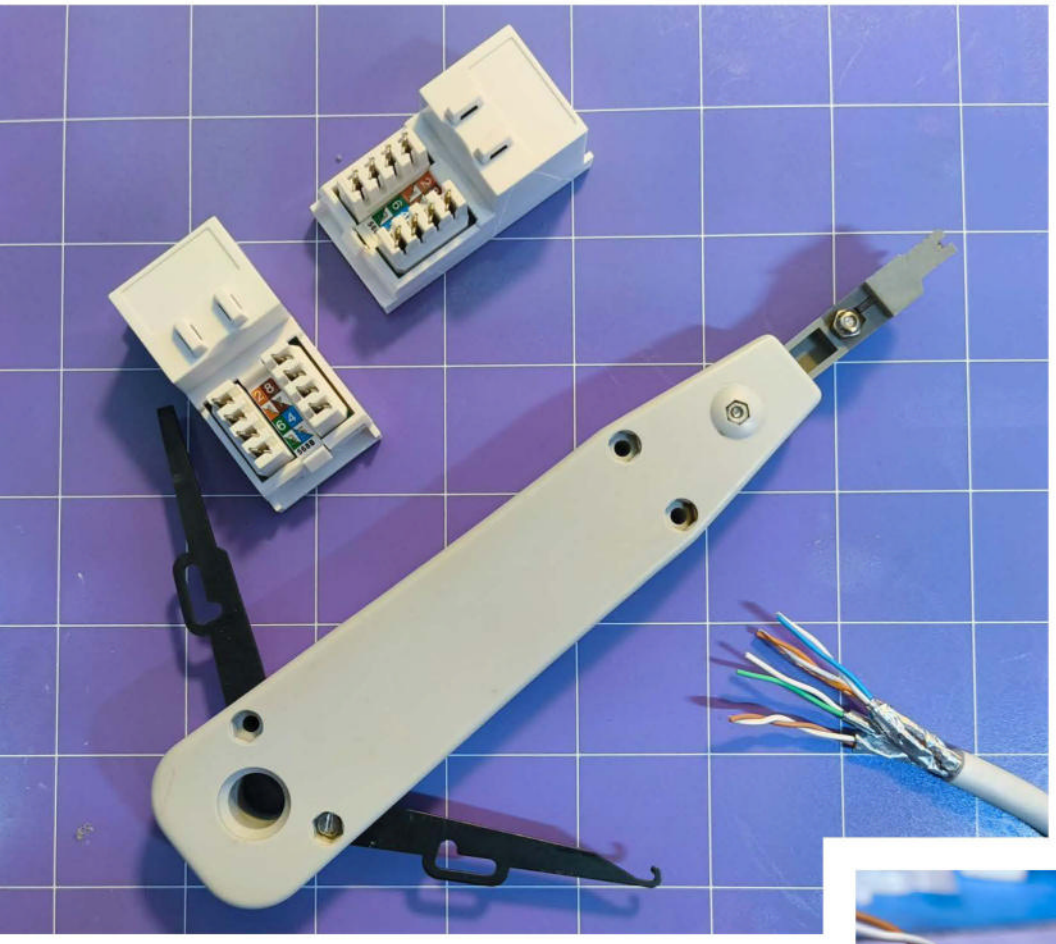
Some tools are instantly recognisable to everybody. There are very few people in the world who wouldn't recognise a hammer or a screwdriver if they saw one. Other tools have a more limited field of recognition, and if someone recognises the tool, you can make a good guess about what sort of job or hobby they have. If someone recognises a punch-down tool, you can assume they've had experience with computer networking or telephony.

The punch-down or Krone tool is used by engineers to connect network cables in sockets and patch panels. Although the tool is extremely common, it's virtually unheard of by the majority of the population. The tool is designed to insert the wires into an IDC (Insulation Displacement Connector), which is a fancy way of saying that the connector itself removes (cuts through) the insulation on the wire you insert. Imagine pushing an insulated wire onto a razor blade. Eventually, the blade will cut through and touch the inner wire. This is exactly what the IDC connector is designed to do.

Used by engineers to connect network cables in sockets and patch panels

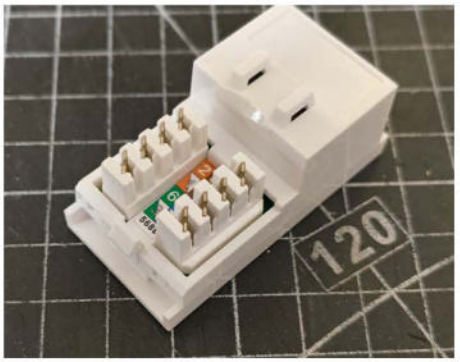
Highly technical pokey stick

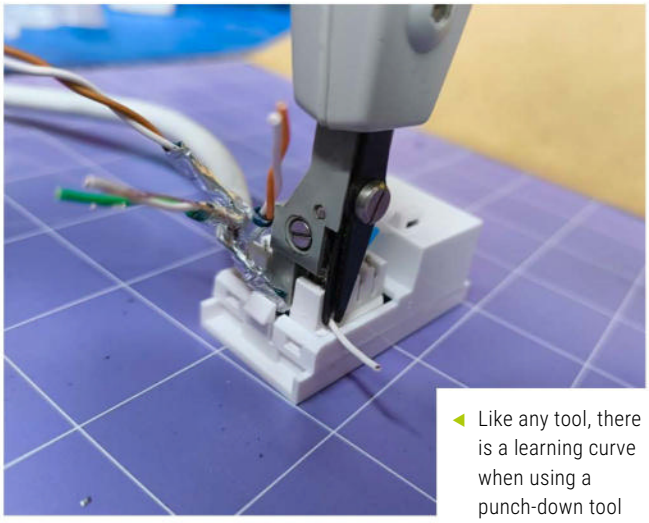
There are several different standards in use when it comes to IDCs. Although some people use the term 'Krone tool' when they're talking about punch-down tools, this is often a misnomer. Krone is a European company that developed a standard for IDCs that arranges the contact slots at 45 degrees to the wire. This allows Krone connectors to be used with either stranded or solid core wire. The more basic 66 and 110 standard IDCs common in the USA use perpendicular contact slots and require solid core wire.



◀ Although it looks like a tool used by a vintage tobacconist to clean the dottle out of pipes, the punch-down tool is at the heart of every cable engineer's toolkit

▼ There are several different standards in use when it comes to IDCs





◀ Like any tool, there is a learning curve when using a punch-down tool

Let's ignore any standards-based technobabble, and take a look at the basics of a Krone tool. Why is the Krone tool different from a simple metal stick, for example? IDC connectors work by having sharp blades positioned in them where you push the wires. Why not just push the wire into place with a screwdriver tip or a thin bit of metal? The answer is that you can. It's not ideal, but in a difficult situation, a thin bit of metal will do the trick. In fact, most Krone tools come with a pop-out metal shim in addition to the regular punch-down blade for doing exactly this. Pushing the wire between the blades is the principal goal of the tool, but making sure that enough pressure has been applied to pierce the insulation and stop the wire falling out again is also important. Some punch-down tools emit a mechanical clicking noise once the tool has been fully pushed down, and many of them have a built-in blade to trim off any excess wire that's sticking out of the edge of the connector. How the blade operates might vary a bit depending on the model of tool that you're using. Some use a simple sharp edge that contacts with the edge of the terminal and slices off the end of the wire when the tool is pushed right down. Other tools have a more complex miniature scissoring wire cutter on the edge of the tool, closing on the wire and snipping like a tiny pair of wire cutters.

A common feature of punch-down tools is that the wire cutting mechanism can be disabled. On some tools, you change or reverse the end of the tool to position a blunt or sharp edge. On tools with a scissoring cutter, the blades can be locked into place by pushing a pin into a slot or turning a screw on the body of the cutter. Disabling the cutter can be useful in situations where you want to loop a wire through several sockets to connect them to a common source for patching, such as a telephone wire coming into a building.

It's particularly important to keep the tool straight when you're pushing wires down onto the connectors. If you vary the angle too much, you can damage the contact slots and the wire won't stay in place. If that happens, you'll just have to throw the connector away and start again. If you make a mistake while wiring, some tools also have a little hook on them that can be used to snag incorrectly positioned cables and pull them free.

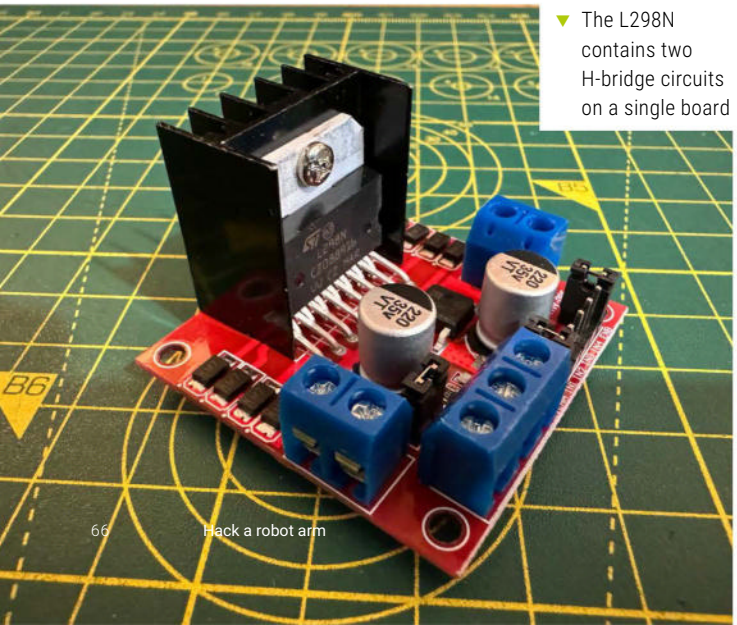
Part 01

Hack a robot arm

Upcycle a toy robot arm to add Raspberry Pi smarts and internet remote control

Maker
PJ Evans
PJ is a writer, software engineer, and very average roboticist. His robot arm enjoys teasing the cat.

mastodon.social/@mrpjevans



▼ The L298N contains two H-bridge circuits on a single board

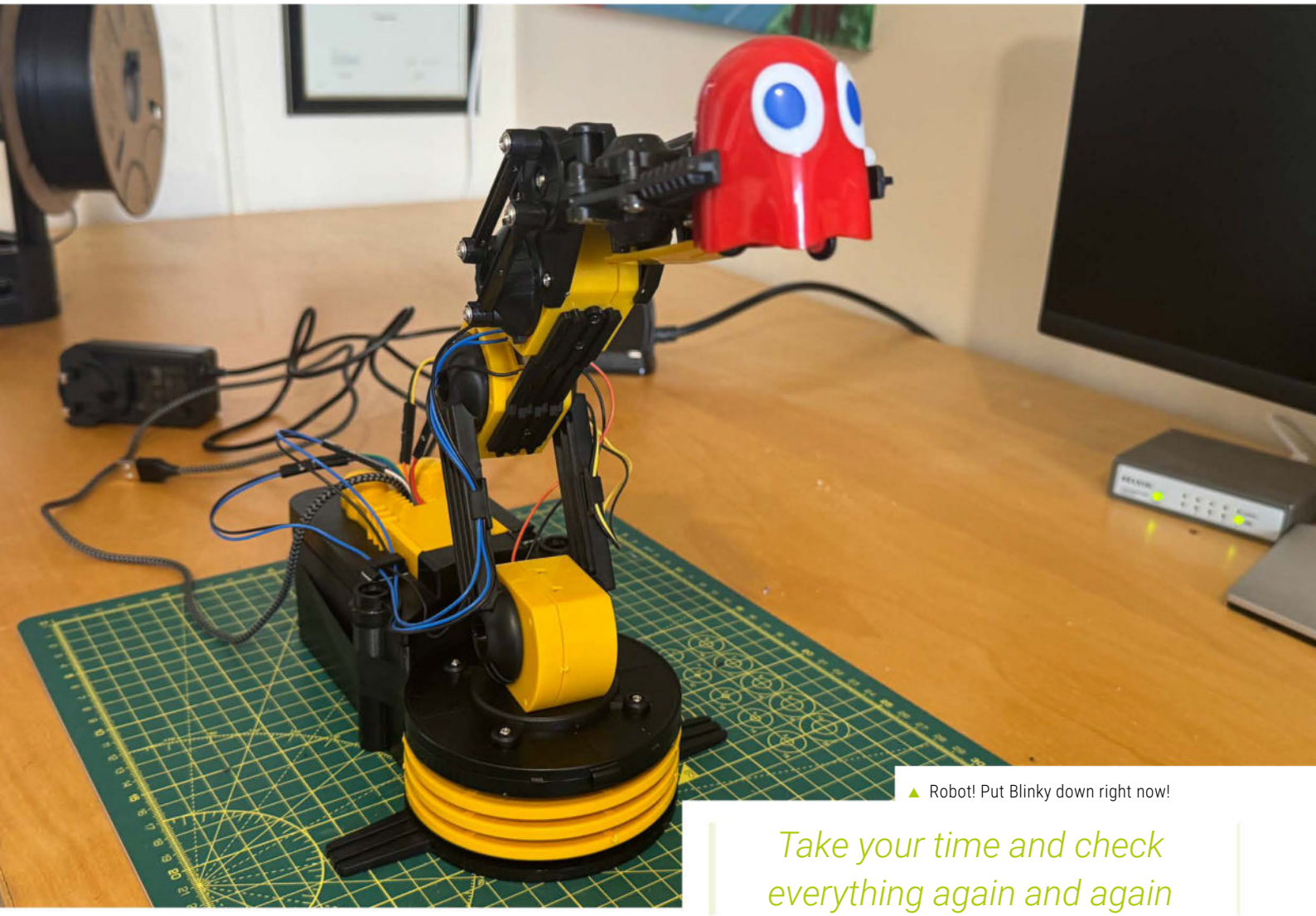
Ever buy a gadget? Does it get played with for a while then consigned to the shelf or cupboard? Upcycling old gizmos and sprinkling a little Raspberry Pi magic is one of our favourite things. Take the OWI-535 (rpimag.co/owi535) robot arm. This fun toy, with several points of articulation and a gripping claw, could be found in electronics shops under various brands. It's still widely available. Although a USB version is now on sale, we wanted to see if we could convert the original remote control version and make it Raspberry Pi powered. Even if you don't have this particular device, this should be a useful guide to motor control.

How the arm works

The OWI-535 robot arm is made up of five DC motors, each controlling a different point of articulation. The base rotates, there is a shoulder, elbow, and wrist, then finally the claw. Each point is bidirectional: the base can rotate either way, the joints can move back and forth, and the claw can adjust its grip. Each motor is connected to a control board by standard jumper cables (useful!) and has two wires. To control the arm, we need to be able to power the motors and also be able to reverse the DC current in order to change direction.

Raspberry Pi challenges

The use of two-wire DC motors presents us with a problem. If it were three wires, a ground and a positive per-direction, it would be straightforward, but a motor that will spin based on polarity of the current applied is not something Raspberry Pi's GPIO can handle out of the box. Also, the current required by the motors to operate is well out of the reach of a Raspberry Pi and we would start damaging things. We need an interface between the two that can help us switch the motors. Enter the humble H-bridge.



▲ Robot! Put Blinky down right now!

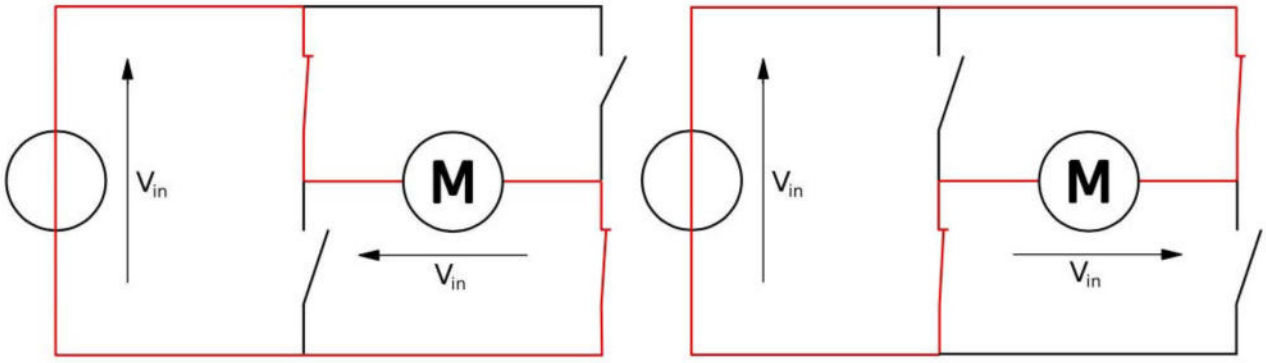
Take your time and check everything again and again

H-bridge over troubled motors

The H-bridge is a very common circuit in robotics that solves our problem. It's a wiring method that uses four switches to reverse the current to a motor. See how the diagram (Figure 1, overleaf) shows both states, causing the motor to spin one way or another. Using five H-bridge circuits and a separate power supply, we can safely control all the motors. We'll do this using three L298N (rpimag.co/l298n) circuit boards. These are cheap and widely available. Each one contains two H-bridge circuits onboard and they are easy to wire up.

Understanding the L298N

The L298N board can control two motors. Each connects its two wires to the board. Power is provided by an external source connected to the power terminals. We need 12V at about 1A. Such supplies are widely available. Finally, control is provided by providing a signal to each 'Enable' line (this allows us to control speed) and by sending one of two lines to 'high', controlling direction. We now have the three-wire setup we originally wanted. The control lines can be safely connected to your Raspberry Pi without risk of damage.



▲ **Figure 1:** This shows how the H-bridge circuit switches polarity (Credit: Cyril Buttay)

Prototyping

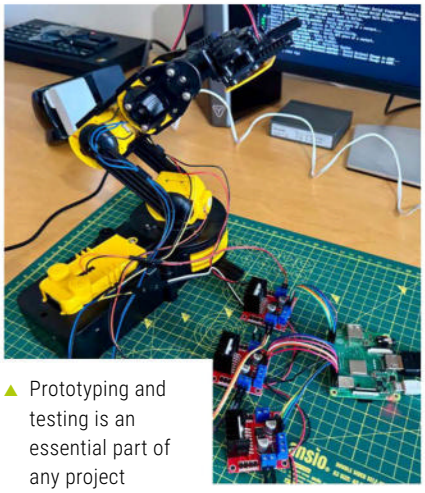
Before doing any irreversible work on your arm, we recommend you do some experimenting first as this will help you understand how everything works. We're using a Raspberry Pi Pico W for our project. Choose one of the motors – the base is a good start. Using some jumper leads, connect its two wires to one of the motor drivers on an L298N. Hook up a 12V supply to the power terminals (but don't switch on yet) and then connect EN1 to GPIO 2, IN1 to GPIO 3, and IN2 to GPIO 4. Finally, connect a wire between any GND (ground) pin on Raspberry Pi Pico to the GND terminal where you connected the power supply.

First moves

Using USB cable, hook your Raspberry Pi Pico W up to another computer that can run Thonny IDE. Ensure the Pico has the MicroPython firmware loaded (rpimag.co/micropython). Thonny should be able to see the Pico and its file system. To help us drive the motors, we need the DCMotor library that you can download here: rpimag.co/dcmotorpy. Copy it to the Pico and then enter the code in the listing (or download it from rpimag.co/owi535testpy). Turn on your 12V supply and run the code. If all is well, the base should rotate one way for half a second and then the other. If it doesn't, check all your wiring carefully.

Preparing the arm

We'll come back to the code in the next part of this tutorial. For now, we're going to concentrate on making our project look great, so make sure everything is powered down. The OWI-535 comes with an elongated base designed to take four D-cell batteries. That gives us some potential room to house our electronics, but not quite enough. To make everything fit, we decided to 3D-print an extension to the housing, making it tall enough to safely contain the three L298Ns and Raspberry Pi Pico. If you have access to a 3D printer, you can download the STL file here: rpimag.co/owi535stl. We then removed the circuit board and carefully removed the stands that held the batteries in place to create plenty of room. With the extra space provided by the 3D print, we are now able to wire up and install the electronics.



▲ Prototyping and testing is an essential part of any project

Full wire-up

The final part for this month requires a little patience. Following the example for one motor in Figure 2, wire everything up as shown in the mapping diagram (Figure 3). We recommend a Pico with pre-soldered headers to make connections easier. Every L298N needs to be wired up to the 12V supply and ground by chaining them together, then we need to connect all the control lines to the GPIO pins of the Pico as shown. Finally, the motors need to be connected to the terminals. Use additional jumper cables between the terminals and the motors so they fit and can easily be changed around. Take your time and check everything again and again. Don't apply any power until you're sure it's OK.

That's all for now

Next month we'll test everything out, install the circuitry, and then add software to take control of the robot arm. Even if you don't have this particular model, the same principle of driving motors using H-bridge circuits can be applied to many different projects.

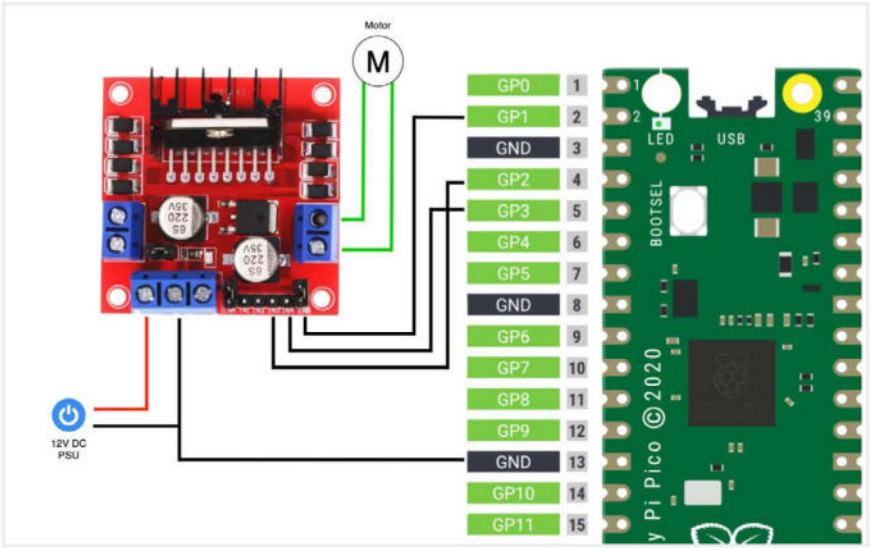


Figure 2: It's easier to show a single motor wire-up – repeat this for all five motors

owi535_test.py

DOWNLOAD THE FULL CODE:

rpimag.co/owi535testpy

> Language: **MicroPython**

```
001. from machine import Pin, PWM
002. from time import sleep
003. from dcmotor import DCMotor
004.
005. # Pins (Motor 1 , Motor 2, Enable)
006. motor_pins = [2,3,1]
007. frequency = 1000
008. speed = 10
009.
010. motor = DCMotor(Pin(motor_pins[0], Pin.OUT),
     Pin(motor_pins[1], Pin.OUT),
     PWM(motor_pins[2], frequency))
011.
012. print("Forward")
013. motor.forward(speed)
014. sleep(0.5)
015.
016. print("Backward")
017. motor.backward(speed)
018. sleep(0.5)
019.
020. print("Stop")
021. motor.stop()
```

Take your time and check everything again and again

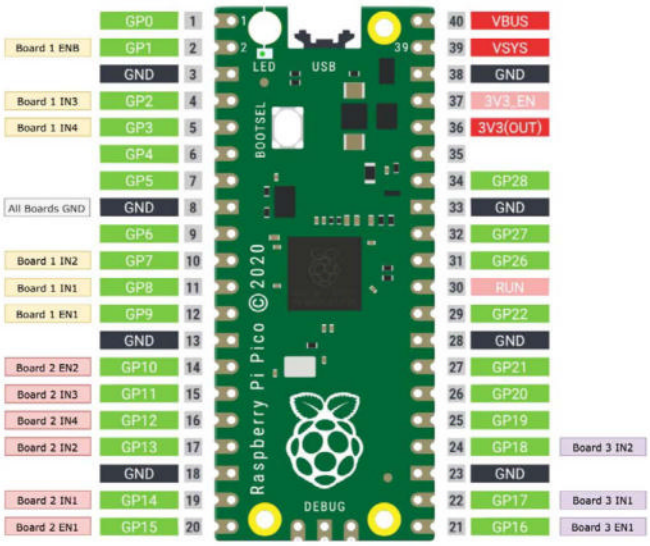
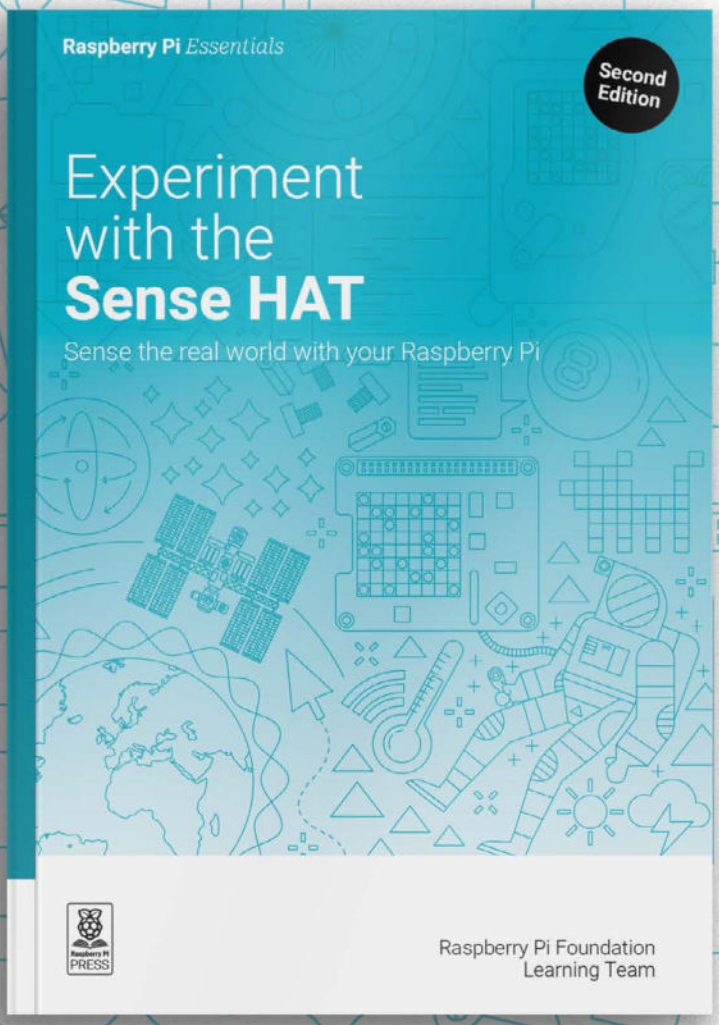


Figure 3: This maps the L298N control lines to the Raspberry Pi Pico W – start with one motor and test it


Raspberry Pi Essentials
Second Edition
Experiment with the
Sense HAT
Sense the real world with your Raspberry Pi
Raspberry Pi PRESS
Raspberry Pi Foundation
Learning Team

Build a Sense HAT data logger

Use your Sense HAT to take environmental readings, so you can walk around the house saying "sensors indicate...".

By **Lucy Hattersley**

Maker

Lucy Hattersley

Lucy is editor of *Raspberry Pi Official Magazine* and just wants a tricorder: sensing, recording, and computing.

rpimag.co

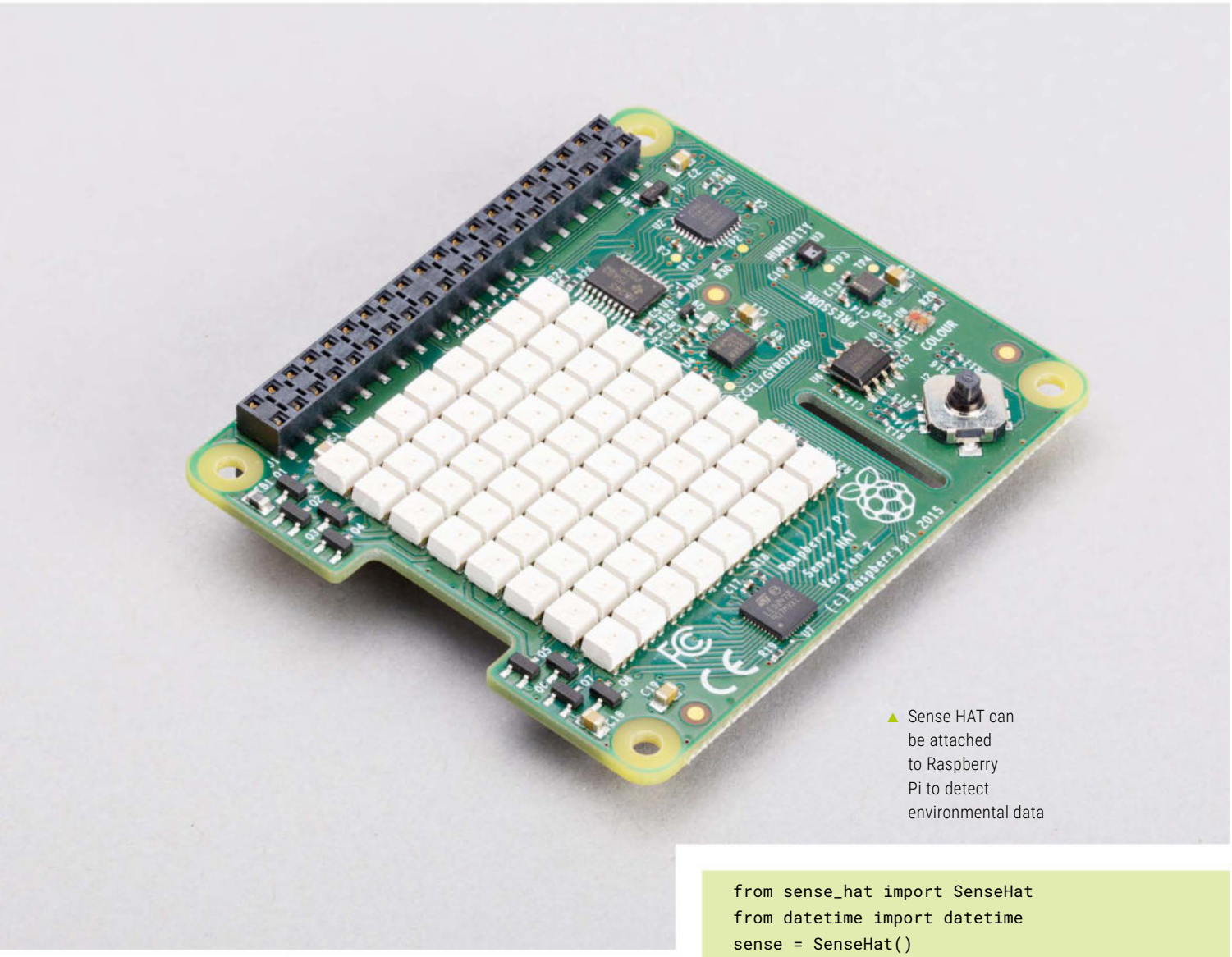
During the Astro Pi mission, a pair of Raspberry Pi computers with Sense HATs attached captured and logged a range of data about life on board the International Space Station.

In this activity, you will use a Raspberry Pi, a Sense HAT, and some Python code to create your own data-logging tool, which you can use to capture interesting data and perform experiments at home.

Using the sensors, we can measure the following conditions:

1. Temperature
2. Humidity
3. Pressure
4. Orientation
5. Acceleration
6. Magnetic field
7. Colour and brightness (V2 Sense HAT only)

First, we'll write a short script to get data from the Sense HAT and output it to the screen. Open the Thonny editor (click the Raspberry Pi menu and select Programming > Thonny); if Thonny is already open, press the New button to start a new program. Save the new file as datalogger.py. To begin this script, you will need to import some Python modules to control your Sense HAT and fetch the date and time from the Raspberry Pi. Start by adding these three lines of code:



▲ Sense HAT can be attached to Raspberry Pi to detect environmental data

In this activity, you will use a Raspberry Pi, a Sense HAT, and some Python code to create your own data-logging tool

```
from sense_hat import SenseHat
from datetime import datetime
sense = SenseHat()
```

If you have a V2 Sense HAT, then you need to set up the colour sensor with the following extra lines of code.

```
sense.color.gain = 60
sense.color.integration_cycles = 64
```

Next, we'll create a function that will fetch all the sensor data and return it as a list. Start by defining your function (`def get_sense_data():`) and creating an empty list (`sense_data = []`). The remaining function calls collect the data from the different sensors. In each case, you are appending the results to the `sense_datalist`.

```
 def get_sense_data():
     sense_data = []
     # Get environmental data
     sense_data.append(sense.get_temperature())
     sense_data.append(sense.get_pressure())
     sense_data.append(sense.get_humidity())
     # Get colour sensor data (version 2 Sense
HAT only)
     red, green, blue, clear =
sense.colour.colour
     sense_data.append(red)
     sense_data.append(green)
     sense_data.append(blue)
     sense_data.append(clear)
     # Get orientation data
     orientation = sense.get_orientation()
     sense_data.append(orientation["yaw"])
     sense_data.append(orientation["pitch"])
     sense_data.append(orientation["roll"])
     # Get compass data
     mag = sense.get_compass_raw()
     sense_data.append(mag["x"])
     sense_data.append(mag["y"])
     sense_data.append(mag["z"])
     # Get accelerometer data
     acc = sense.get_accelerometer_raw()
     sense_data.append(acc["x"])
     sense_data.append(acc["y"])
     sense_data.append(acc["z"])
     # Get gyroscope data
     gyro = sense.get_gyroscope_raw()
     sense_data.append(gyro["x"])
     sense_data.append(gyro["y"])
     sense_data.append(gyro["z"])
```

Complete your data capture by adding the current date and time to the list.

```
     # Get the date and time
     sense_data.append(datetime.now())
```

The function should return the `sense_data` list at the end.

```
     return sense_data
```

To finish off, you can look at the data by printing out the list within an infinite loop. Add the following to the end of your script, then save and run the code.

```
 while True:
     print(get_sense_data())
```

You should see a continuous stream of data in the shell, with each line looking something like this:

```
 [38.94975280761719, 1012.20166015625,
32.560699462890625, 17,
 10, 11, 20, 144.1204202422324,
12.432924190508864,
 355.4826729655122, -33.784427642822266,
-23.137866973876953,
 3.40659761428833, -0.19746851921081543,
-0.054034601897001266,
 0.9491991996765137, -0.014246762730181217,
 0.009623102843761444, -0.010858062654733658,
 datetime.datetime(2022, 7, 25, 14, 26, 51,
341250)]
```

If you have an original Sense HAT without the colour sensor, you'll get an error unless you remove these lines:

```
     # Get colour sensor data (version 2 Sense
HAT only)
     red, green, blue, clear =
sense.colour.colour
     sense_data.append(red)
     sense_data.append(green)
     sense_data.append(blue)
     sense_data.append(clear)
```

Writing the data to a file

The program you have produced so far is able to continually check the Sense HAT sensors and write this data to the screen. We can make this more practical by altering the program to write the data to a *comma-separated value* (*CSV*) file instead, which you can examine once your logging program has finished. To create this file, you will need to do the following:

1. Create the file
2. Add a header row for each sensor reading
3. Write the data to the file

Import the `csv` module's `writer` class by adding the line `from csv import writer` to the top of the program:

```
from sense_hat import SenseHat
from datetime import datetime
from csv import writer
sense = SenseHat()
```

Remove your current `while True` loop. Replace it with these lines to open and write to a new data.csv file:

```
with open('data.csv', 'w', buffering=1,
newline='') as f:
    data_writer = writer(f)
```

Create a variable to hold the data from the function call within a new `while True` loop, then write that data to the new file. Make sure to set the file buffering mode to line mode (expressed as a `1`) so that every row is saved immediately and data loss is avoided.

```
    while True:
        data = get_sense_data()
        data_writer.writerow(data)
```

Run your program for a few minutes. When you stop the program, you should be able to find the data.csv file in your file manager.

The first line should look something like this:

```
36.324222564697266,1024.387939453125,32.6043815
612793,0,0,0,1,
138.03485829553443,12.164214303276808,353.073809
95988177,
-10.638025283813477,-
9.077208518981934,1.978834867477417,
-0.20660144090652466,-
0.11602965742349625,0.9455438256263733,
-0.005754658952355385,-
0.00629773736000061,0.00323345884680748,
2022-07-26 11:12:45.316169
```

Adding a header to the CSV file

Your program is collecting so many different types of data in the CSV file that it can be hard to identify which type of data each column contains. To solve this problem, you can write a row to the top of the CSV file before you start the infinite loop.

Add a call to `data_writer.writerow` after you've created your `writer` object and before the `while True` loop starts. Pass it an array with the names of the columns you'd like to use. Your code (everything after `return sense_data`, that is) should look like this:

```
with open('data.csv', 'w', buffering=1,
newline='') as f:
    data_writer = writer(f)
    data_writer.writerow([
        'temp', 'pres', 'hum',
        'r', 'g', 'b', 'clear', # V2 only
        'yaw', 'pitch', 'roll',
        'mag_x', 'mag_y', 'mag_z',
        'acc_x', 'acc_y', 'acc_z',
        'gyro_x', 'gyro_y', 'gyro_z',
        'datetime'])
    while True:
        data = get_sense_data()
        data_writer.writerow(data)
```

Recording at specific time intervals

At the moment, your script records data as quickly as it possibly can. This is very useful for some experiments, but you may prefer to record data once per second, or even less frequently.

Normally, you would use a `sleep()` function to pause the script, but this can cause inaccurate readings from some of the IMU sensors. Instead, you can use `timedelta` to check the time difference between two readings.

At the top of your program, set the length of delay between readings in seconds, and fetch the current date and time.

```
timestamp = datetime.now()
delay = 1
```

Within your `while` loop, you can calculate the difference between the current time and the time stored in the `data` list (the last element, which we can refer to with the array index `-1`):

```
    while True:
        data = get_sense_data()
        time_difference = data[-1] - timestamp
```

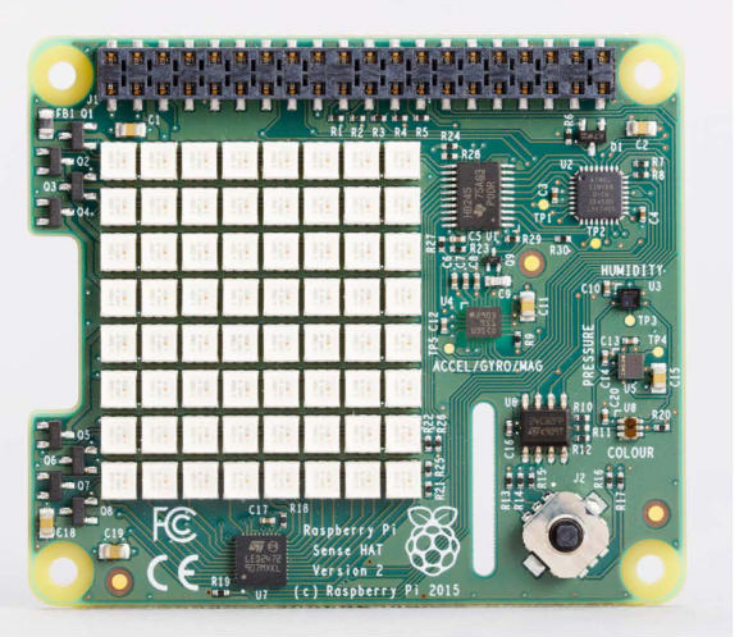
If the `time_difference` is greater than the delay you have set, the `data` can be written to the file. Change your `while` loop to look like this:

```
        while True:
            data = get_sense_data()
            time_difference = data[-1] - timestamp
            if time_difference.seconds > delay:
                data_writer.writerow(data)
                timestamp = datetime.now()
```

See the end for the full code listing and download link.

While your program is running, open the terminal, use `cd` to change directory to the directory that contains the data.csv file, and run the command `tail -f data.csv` to watch as each line is added to the file:

If you have LibreOffice installed, you can use LibreCalc to open it as a spreadsheet. If you don't have LibreOffice installed, click the Raspberry Pi menu, choose Preferences > Recommended Software, and use the Recommended Software tool to install it. You should see that the file is only being written to periodically. You can adjust the delay time you have set to write more frequently or less frequently.



▼ Sense HAT can be used to detect temperature, humidity, pressure, orientation, acceleration and magnetic field. Version 2 also adds a colour and brightness sensor

Starting your data logger on boot

This step is completely optional, but you might want to have your script run as soon as the Raspberry Pi boots up. To do this, you can use a *cron job*. Have a look at the section below to learn how to edit your crontab file to start scripts on boot.

Sometimes you don't want to manually start a script that you have written. You may need the script to run once every hour, or maybe once every 30 seconds, or even every time your computer starts. On Linux-based systems like Raspberry Pi OS, this is a fairly easy task, as you can use a program called *cron*. Cron will run any command you tell it to run, whenever you have scheduled for it to do so. It will reference what is known as the *cron table*, which is normally abbreviated to *crontab*.

TIP

Make sure the headers are in the same order as the data produced by your `get_sense_data()` function. If you have an original Sense HAT without a colour sensor, be sure to remove the line that starts with `'r', 'g', 'b', 'clear'`.

Editing the crontab

To open the crontab, you first need to open a terminal window. Then you can type:

```
crontab -e
```

The `-e` in this command is short for edit. If this is your first time opening your crontab, you'll be asked which text editor you would like to use:

```
rpi@raspberrypi:~ $ crontab -e
no crontab for rpi - using an empty one

Select an editor.  To change later, run
'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /bin/ed

Choose 1-3 [1]:
```

Unless you have plenty of experience using `ed` or `vim`, the simplest editor to use is nano; type `1` to select it and press ENTER.

The nano editor is a simple command-line text editor. If you want to learn more about using nano, see rpimag.co/nano.

Syntax for cron

The crontab contains all the basic information you need to get started. Each line that starts with a `#` is a comment and is therefore ignored by the computer. At the bottom of the crontab, you should see a line that looks like this:

```
# m h  dom mon dow   command
```

1. `m` is short for minute
2. `h` is short for hour
3. `dom` is short for day of the month
4. `mon` is short for month
5. `dow` is short for day of the week
6. `command` is the shell command that you want to run

Creating a new cron job

To create a cron job, you need to decide under which circumstances you would like it to run. For instance, if you wanted to run a Python script on the 30th minute of every hour, you would write the following, making sure to change the `rpi` username to your own username; if you saved datalogger.py in a different location, change the path as needed:

```
30 * * * * python3 /home/rpi/my_script.py
```

If you wanted it to run every 30 minutes, you would use:

```
*/30 * * * * python3 /home/rpi/my_script.py
```

The `30` is telling the script to run every 30 minutes. The asterisks indicate that the script needs to run for all legal values for the other fields.

Here are a few more examples.

Run a script at 11:59 every Tuesday:

```
59 11 * * 2 python3 /home/rpi/my_script.py
```

Run a script once a week on Monday:

```
0 0 * * 1 python3 /home/rpi/my_script.py
```

Run a script at 12:00 on the 1st of January and June:

```
0 12 1 1,6 * python3 /home/rpi/my_script.py
```

> *You might want to have your script run as soon as the Raspberry Pi boots up*

Run on boot

One incredibly useful feature of cron is the ability to run a command when the computer boots up. To do this, you use the `@reboot` syntax.

```
@reboot python3 /home/rpi/datalogger.py
```

Because the data logger runs continuously until you stop it, this is probably the best option. If you were to run it, for example, every 30 minutes, you'd have multiple data loggers trying to read the sensors and trying to write to the file, which would quickly become problematic!

If you want to quit your data logger after starting it from your crontab, you can run the following command from the terminal:

```
pkill -f datalogger.py
```

Edit and save the file

You can add your cron job to the bottom of the crontab. Save and exit nano by pressing CTRL+X, then Y when you are prompted to save, and ENTER to accept the file name (don't change it).

Selecting the data to be recorded

You might not always want to record all the sensor data. One solution to this is to simply comment out the lines you don't need in your `get_sense_data()` function by adding hashtags (#) in front of them.

Can you set up your script to pass only the sensors you want to use into your `get_sense_data` function, ensuring that only data from those sensors is recorded? Don't forget to add a method to alter the header row of your CSV file as well!

datalogger_with_delay.py

DOWNLOAD THE FULL CODE:

rpimag.co/dataloggerpy

> Language: **Python**

```
001. from sense_hat import SenseHat
002. from datetime import datetime
003. from csv import writer
004. sense = SenseHat()
005. timestamp = datetime.now()
006. delay = 1
007.
008. sense.color.gain = 60
009. sense.color.integration_cycles = 64
010.
011. def get_sense_data():
012.     sense_data = []
013.     # Get environmental data
014.     sense_data.append(
     sense.get_temperature())
015.     sense_data.append(sense.get_pressure())
016.     sense_data.append(sense.get_humidity())
017.     # Get colour sensor data (version 2
     Sense HAT only)
018.     red, green, blue, clear = sense.colour.
     colour
019.     sense_data.append(red)
020.     sense_data.append(green)
021.     sense_data.append(blue)
022.     sense_data.append(clear)
023.     # Get orientation data
024.     orientation = sense.get_orientation()
025.     sense_data.append(orientation["yaw"])
026.     sense_data.append(orientation["pitch"])
027.     sense_data.append(orientation["roll"])
028.     # Get compass data
029.     mag = sense.get_compass_raw()
030.     sense_data.append(mag["x"])
031.     sense_data.append(mag["y"])
032.     sense_data.append(mag["z"])
033.     # Get accelerometer data
034.     acc = sense.get_accelerometer_raw()
035.     sense_data.append(acc["x"])
036.     sense_data.append(acc["y"])
037.     sense_data.append(acc["z"])
038.     # Get gyroscope data
039.     gyro = sense.get_gyroscope_raw()
040.     sense_data.append(gyro["x"])
041.     sense_data.append(gyro["y"])
042.     sense_data.append(gyro["z"])
043.
044.     # Get the date and time
045.     sense_data.append(datetime.now())
046.
047.     return sense_data
048.
049. with open('data.csv', 'w', buffering=1,
     newline='') as f:
050.     data_writer = writer(f)
051.     data_writer.writerow([
052.         'temp', 'pres', 'hum',
053.         'r', 'g', 'b', 'clear', # V2 only
054.         'yaw', 'pitch', 'roll',
055.         'mag_x', 'mag_y', 'mag_z',
056.         'acc_x', 'acc_y', 'acc_z',
057.         'gyro_x', 'gyro_y', 'gyro_z',
058.         'datetime'])
059.     while True:
060.         data = get_sense_data()
061.         time_difference = data[-1] -
062. timestamp
063.         if time_difference.seconds > delay:
064.             data_writer.writerow(data)
065.             timestamp = datetime.now()
```

RETRO GAMING WITH RASPBERRY PI

3RD EDITION

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

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

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

Origami paper circuits

From a flat square to a finished, illuminated sculpture... let's return to the fold

Maker

Nicola King

Nicola is a freelance writer and sub-editor. There's a good reason why so many ancient and heritage crafts are still popular today, and she's all in!

@holtonhandmade

QUICK TIP

You can use all sorts of paper for origami – the key characteristics needed though are suppleness, an ability to hold a fold well, and some degree of strength. Wrapping paper works a treat.

Origami is an ancient Japanese art form, and the word itself stems from two words – 'ori' (from the root verb 'oru') means 'to fold' and 'kami' is one of the Japanese words for paper. The aim of the craft is to create something very beautiful from a flat two-dimensional sheet of paper. Equipped with knowledge of just a few basic folds, you can create some fairly intricate designs.

If you are looking for an inexpensive hobby, this one surely hits that mark, as all you really need is some paper and a scoring and folding tool. In this tutorial, we are going to make some simple origami shapes. We are then going to use a Raspberry Pi Pico and LED to shine a light on our creations – blending an ancient craft with the very modern, if you will. Great for hand-eye coordination and mental concentration, this mindful art is an excellent way to boost your creativity. At the end of the origami process, we'll have exactly what we started with, i.e. a piece of paper, but we'll have transformed it into something a little more interesting. We'll then attach it to a mount and add the circuitry.

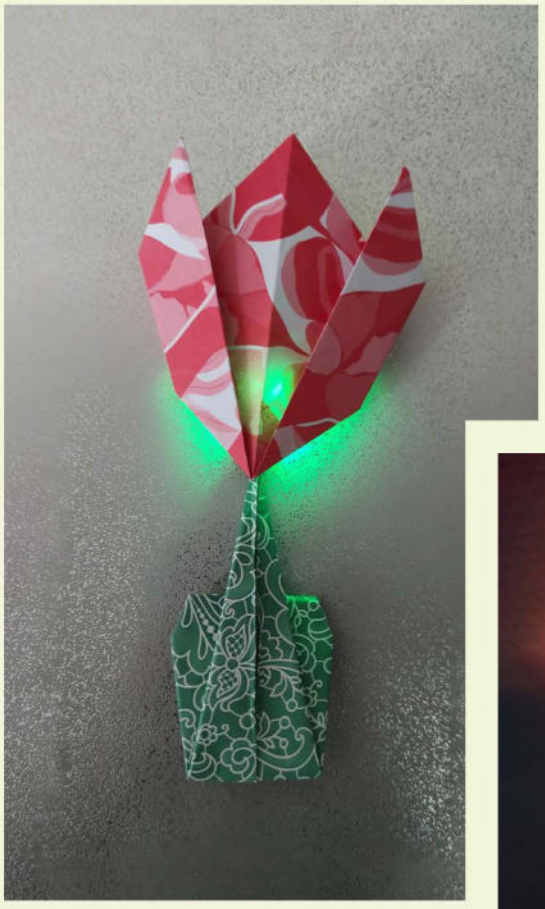
Tiptoe through the tulips

We are keeping this simple – we're new to origami, we want to start with the basics and work our way up, and we only have a few pages in which to cover the entire topic. Our aim is to open the door of the origami rabbit hole that you can delve more deeply into. Plus, we'll be spending some time lighting up our creation too. So, a simple tulip shape is the goal, and it should take you a few seconds to make each one.

Before we get underway, it's worth emphasising a couple of points. Firstly, the key to producing a high-quality piece of origami is in the quality of folding itself, so don't rush it and pay attention to accuracy. Use a scoring tool with a ruler if you want really crisp, precise folds, especially if your paper is a bit thicker. Secondly, be aware which side of the paper should be

YOU'LL NEED:

- Colourful paper to fold (perfect squares in shape)
- Scoring tool and ruler (optional)
- Tweezers
- Folding bone
- Scissors
- A4 piece of cardboard/ mountboard
- Blu tack / other adhesive putty
- RGB LED
- Raspberry Pi Pico
- Resistors
- Jumper wires
- Power source



▲ Our very simple origami make with some Raspberry Pi thrown in for good measure. Once you've learnt the basic folds, you'll find that much more complex designs are within your reach, and challenging yourself is the best way to improve

We're using an RGB LED controlled by a Raspberry Pi Pico to create a glowing effect that cycles through all the colours of the rainbow

Origami origins

Origami's main requirement in terms of tools is, obviously, paper. It's thought that the first paper was made under the Chinese Han dynasty way back in 105 CE. Japan was not far behind in its move into paper-making in around the 6th century, and that country's love of paper since then has been enduring. It's hardly surprising then that the art of folding paper appears to have originated in one of these two countries; no one is completely sure who can claim that title.

Initially, folding paper was used for ceremonial purposes as paper was something of a luxury, but it's believed that from around 1600 onwards, the art of paper folding had become much more recreational as mass-produced paper became more affordable to the everyday person. People have been folding paper for centuries, giving each other paper-folded gifts to cement friendships.

Originally known as orikata (paper folding), the descriptive term that we now recognise, origami, started to be used much more around the late 1800s. It's now popular the world over. As well as being a meditative hobby, origami is used by teachers to teach geometry (e.g. folding used to illustrate symmetry and fractional concepts), spatial relationships, design principles, and problem-solving skills, as origami is rooted in mathematical principles.

facing you as you fold – if you are using origami paper, as we are, one side is coloured and one side is white, so you need to ensure that you have the correct side facing you, per the instructions you're following.

We're using origami papers that are 15cm by 15cm, so completely square in shape. With the white/wrong side facing you, fold the paper diagonally in half to make a triangle – this is called a valley fold (an inward pointing line), and is the most basic origami technique. Then fold that triangle in half, and open the fold just made, ensuring that the folded line is facing upwards. Next, take the points on the base of the triangle upwards, as shown in Figure 1. Turn the flower over, and then fold the sides as in Figure 2. Turn the flower over again, and you have the head of your tulip.

QUICK TIP:

Take care when using any tools; paper is delicate, and a gentle touch will always be the best policy.

Stalk talk

The stem of the flower would be perfect in a green colour, if you have it. Wrong/white side facing you, fold the paper diagonally again as before. Then open it and fold as per Figure 3. Turn it up the other way and then fold both sides per Figure 4, so you have a kind of kite shape. Then fold both sides in again, per Figure 5. Turn the stem over and fold the pointed end up to the top. Then fold in half, per Figure 6. Pinch the pointed end and pull it out slightly, and turn the outer layer outwards so it resembles a stem.

Brighten your bloom

The final step is the circuitry part, which is actually quite straightforward. We estimated where our tulip would sit on an A4 piece of cardboard, and made a small hole with some scissors that an LED (which will illuminate the head of the flower) could be poked through.

QUICK TIP

When you are making very small origami shapes, you can also use a small pointed stick, like a cocktail stick, to help gently push the paper into the shape you want.

In this example, we're using an RGB LED controlled by a Raspberry Pi Pico to create a glowing effect that cycles through all the colours of the rainbow using PWM (pulse-width modulation). First, wire up the circuit on a breadboard as shown in Figure 7. Our RGB LED is a common cathode type with one cathode leg to connect to a GND pin on Pico. The other three legs are for red, green, and blue elements of the LED. The order is typically red, cathode (longer leg), green, blue – if yours differs, adjust the connections accordingly. We connected our red, green,

▼ You don't need much to begin origami

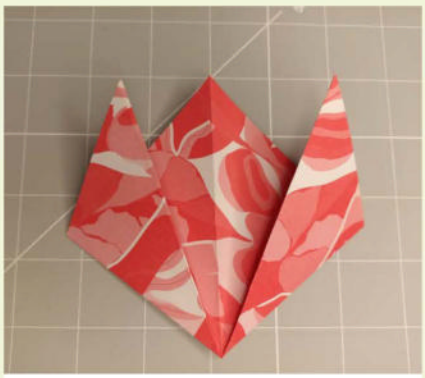
Crazy for cranes

If you are at all familiar with origami shapes, the classic crane shape is probably a design that you will have heard of, as it is possibly the most famous and loved of all origami designs, and is famous throughout the world.

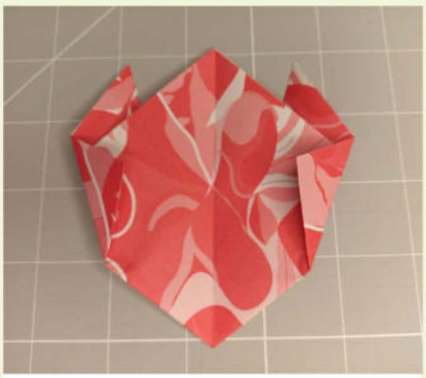
The crane, or 'orizuru' to give it its correct term, is a symbol of peace, hope, and good fortune, so if someone gives you one as a gift, that's a thoughtful gesture that should be long appreciated. The crane itself is a bird that is seen as mystical and is held in high regard in Japan.

There was even a Japanese book published in 1797 called *Hiden Senbazuru Orikata* which described instructions on how to fold 1000 cranes, and is thought to be one of the first books on origami. There's a popular legend that if someone folds that number of cranes, then they will be granted one wish, but over time the making of 1000 cranes is now entwined with the aims of general happiness, good luck in personal pursuits, and so on.

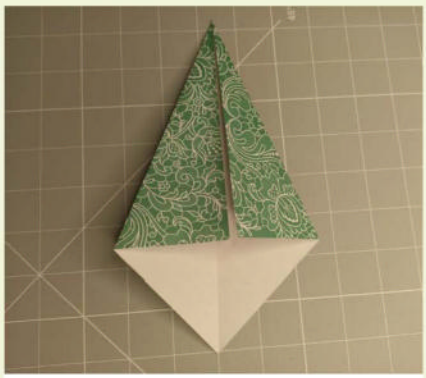
If you'd like to make one yourself, this link provides some excellent instructions, and in around 15 minutes you should have your own crane to admire: **origami.me/crane**.



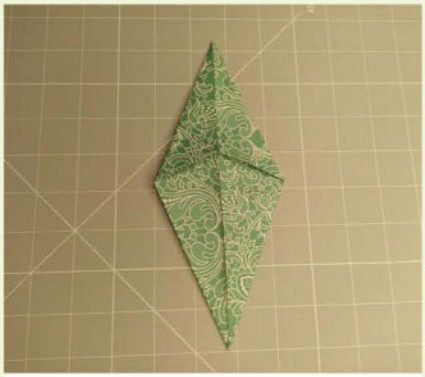
▲ **Figure 1:** Once you've made a few of these flowers, muscle memory will kick in and they will take you seconds to make



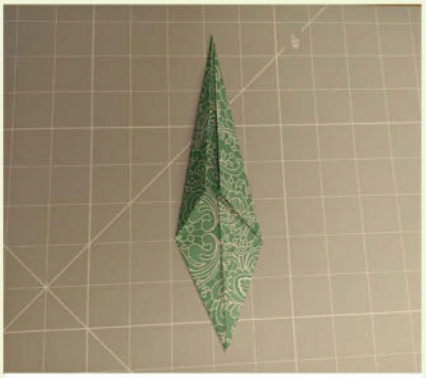
▲ **Figure 2:** Making origami shapes is a great way of using up scrap paper that's lying around, as well as being a relaxing pastime



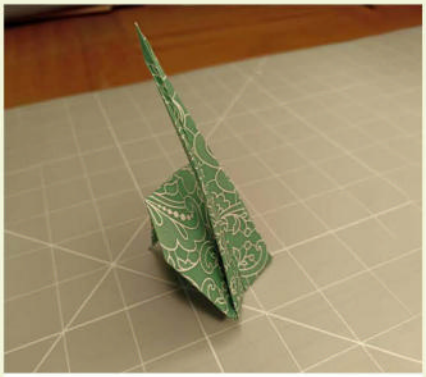
▲ **Figure 3:** Starting the stem. If you have a hole in base of your flower, you can just feed the top of the stem into it



▲ **Figure 4:** It's very important to make folds absolutely precise, otherwise the rest of the process ends up askew and errors multiply



▲ **Figure 5:** Sounds obvious, but make sure your hands are clean before you start so that you don't taint your piece in any way



▲ **Figure 6:** You can make a soft crease before you commit to any origami fold, so if you are slightly off you can correct it more easily

We made a small hole with some scissors that an LED (which will illuminate the head of the flower) could be poked through

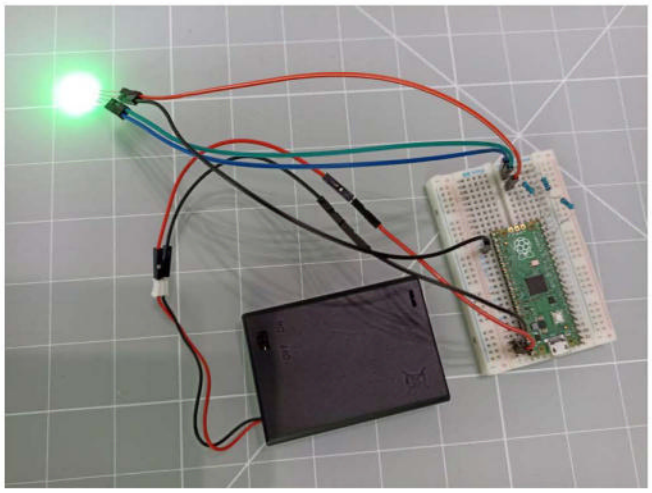
QUICK TIP

Do make sure that you are folding your paper on a completely flat surface, as any ridges or dents will impact the aesthetic of your final piece.

and blue legs via resistors (we used 150Ω ones) to the GPIO 13, 14, and 15 pins respectively.

You can check everything is connected correctly by first cycling through red, green, and blue separately. We're using the picozero MicroPython library to make it simpler. Connect Pico to a computer and write the following code using Thonny (or another IDE):

▼ The electronic circuit, using Pico. As shown here, you can take the RGB LED off the breadboard to use it in the origami flower



```
from picozero import RGBLED
from time import sleep

rgb = RGBLED(red = 13, green = 14, blue = 15)

while True:
    rgb.color = (255, 0, 0)
    sleep(0.5)
    rgb.color = (0, 255, 0)
    sleep(0.5)
    rgb.color = (0, 0, 255)
    sleep(0.5)
```

Run the code and the RGB LED should light red, green, and blue in sequence, with an interval of half a second. The code for gradually cycling through the whole range of colours – between red and green, green and blue, then blue and red – is even simpler, thanks to picozero's built-in `cycle` method:

Technology and origami: a complementary relationship

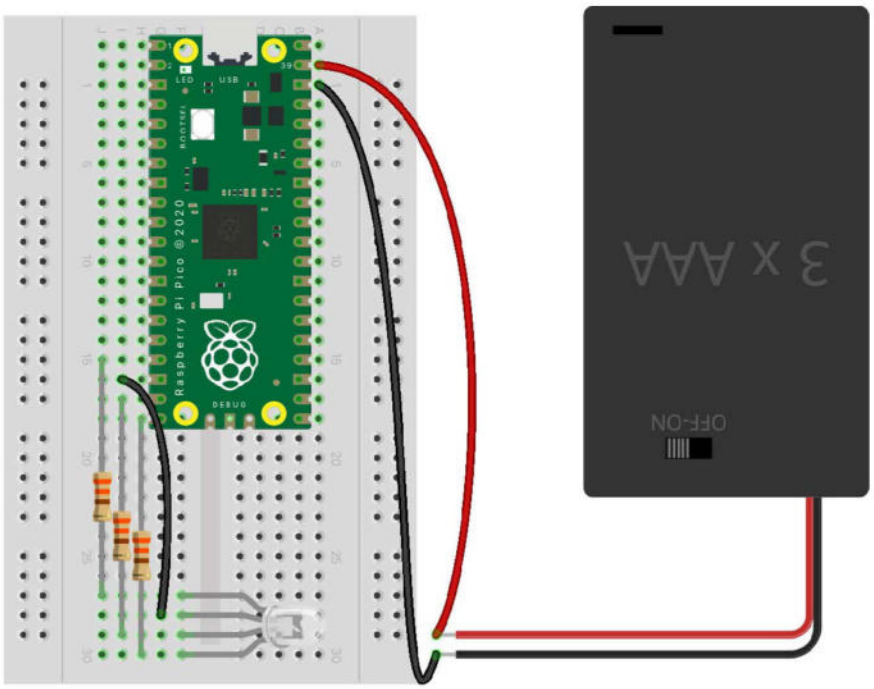
A number of very clever people are folding paper and inserting some tech into their work at the same time, and are even using origami to help engineer other structures, using plenty of mathematics and experimentation. If you are looking for some inspiration, these projects are sure to make you fans of folding:

Gain some inspiration from Dr Matthew Gardiner (**matthewgardiner.net**), who is an artist well known for his work with origami and robotics. In fact, back in 2003, he coined the term 'Oribot', and his website is a fascinating treasure trove of images of his work, videos, kits, and so on.

Ross Symons is the origami artist behind White on Rice (**white-onrice.com/tutorials**), a website where you can find some great tutorials on how to fold origami-style. In particular, we'd draw your attention to his Origami Meditation Mural, a clever piece of interactive wall art made of 64 origami flowers. The really smart bit though is that when you press some of the flowers, a sound is played. He used Raspberry Pi (of course!), a Pi Cap add-on board, and some electric paint and copper tape in its construction. Capacitive sensing is at the heart of this super project: **youtu.be/zUX7ya_oAgw**.

Origami engineering – yes, it's a thing! Engineers are using the principles of origami to help solve engineering challenges. Origami is a useful tool because the goal is it to take a flat sheet of paper and transform it through folding and sculpting. You can apply these conventions to any flat material. This form of engineering allows objects to fold compactly, while the structural integrity is maintained. Examples would include collapsible packaging for the benefits of recycling, implements involved in medical diagnostics, or even origami-inspired robot plants that can monitor environments, to name a few. This interesting video provides some enlightening background to the topic: **youtu.be/ThwuT3_AG6w**.

NASA even has a page on its website which delves into 'Space Origami', and shows you how to make your own Starshade model, a NASA device that will be used to shield telescopes' cameras from the bright light of stars, and will fold out to its full size in space: **rpimag.co/starshade**.



▲ **Figure 7:** The wiring diagram for connecting Pico to the RGB LED on a breadboard

```
from picozero import RGBLED

rgb = RGBLED(red = 13, green = 14, blue = 15)
rgb.cycle()
```

If you want the colours to cycle more slowly or in a different order, you can add parameters inside the brackets of `cycle()`. For instance:

```
rgb.cycle(fade_times=3, colors=((0, 0, 255),
(0, 255, 0), (255, 0, 0)))
```

This example will increase the fade time between main colours from 1 to 3 seconds and go in the order blue, green, red.

Save your colour-cycling code as main.py

Save your colour-cycling code as main.py on Raspberry Pi Pico and it will then run automatically when connected to a power source. We used a 3 × AA battery pack connected to Pico's VSYS and GND pins; alternatively, just use a USB power bank.

Now take your RGB LED off the breadboard and push its legs, toward the rear, through the small hole you made in the cardboard. Connect long jumper wires from its legs back to the holes in the breadboard where they were previously inserted.

Turn on the power to Pico and the RGB LED should create a colour-cycling glow to illuminate your origami flower and the card behind it. For best effect, try it in a dark place for an impressive glowing flower.

Part 01

Design your first PCB

Use Pico W to alert you to rain and never forget your brolly again

Maker

Ben Everard

Ben is a freelance writer and artist. Both occupations involve an abundance of LEDs.

glowingart.co.uk

Many electronics projects begin life on a breadboard. It's an easy way to test things out as you build your circuits. You can adjust, change, develop, and build things. However, what do you do when you want to make your projects permanent? You could leave it on a breadboard, but sooner or later, things will fall out. There is protoboard – PCB that's just made up of through-holes that you can solder components into. This can work, but it's a bit messy and it is awkward to use with surface-mount parts. The solution is to design your own PCB.

This sounds like a complex process, but it can be surprisingly straightforward. PCB design tools are becoming easier to use, and there are loads of PCB manufacturers that are well set up for hobbyists.

In this article, we're going to design and fabricate a simple PCB. It's a 'Brolly Monitor' with a Pico W that grabs a weather forecast. If there's rain forecast, it lights up some LEDs in the shape of an umbrella. You can mount this PCB near your door and hopefully, it'll remind you to grab your brolly if there's rain forecast.

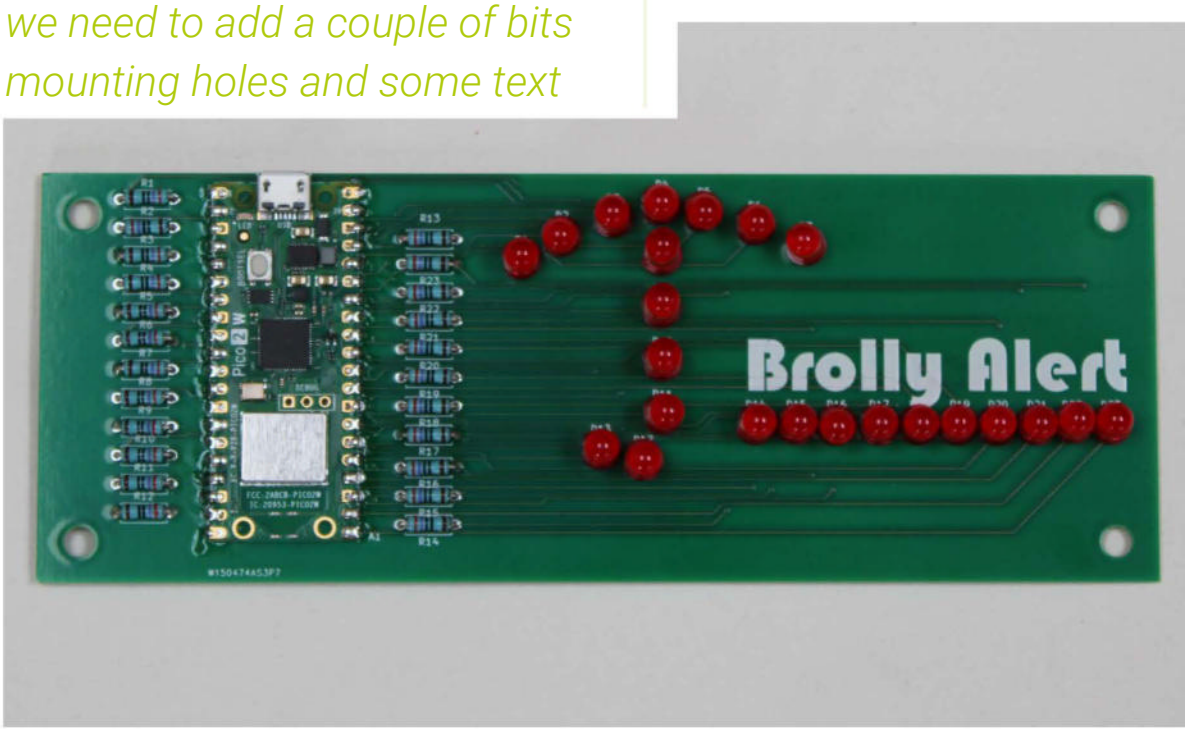
Let's take a look at how to do this with KiCad, an open-source PCB design tool that runs on Windows, Linux, and macOS that you can download from kicad.org. The latest version is 9 and if you have an older version already installed, we'd recommend you upgrade before continuing.

Designing a PCB with KiCad is a four-stage process:

- Design your schematic
- Assign footprints to the parts you've used
- Lay out the PCB
- Export the files for manufacture

Let's look at each one in turn.

This is our board electrically complete now, but we need to add a couple of bits – some mounting holes and some text



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

Schematic

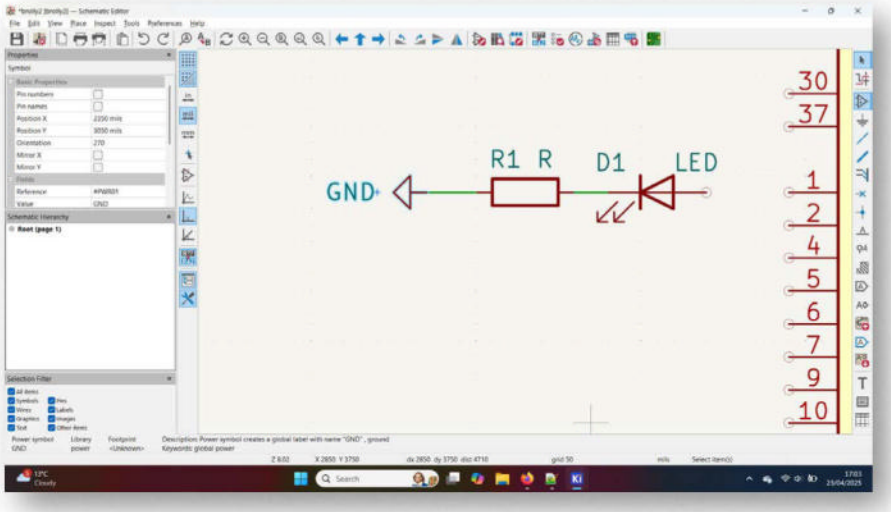
A schematic diagram shows the logical arrangement of electronic parts, and abstracts away a lot of the detail. For example, when you draw an LED in a schematic, you don't usually mark what colour it is, what size it is, whether it's through-hole or surface-mount, or any of that detail.

Similarly, the position of the objects on a schematic isn't the same as their position in the real world. All we're interested in is the type of object and the connections between them.

In KiCad, schematics are created using the aptly named Schematic Editor, so open KiCad, create a new project, and click on Schematic Editor. This will open a new window with a blank schematic in it.

The first thing we need is some components: a Pico W, an LED, and a resistor. To place a component (known as a symbol in the Schematic Editor), click on the op-amp icon in the right-hand toolbar (a triangle with a '+' and '–' in it). This will open up a dialog box where you can choose what symbol you want. Search for Pico and select RaspberryPi_Pico_W. Since they are pin-compatible, this works for either a Pico W or a Pico 2 W. Even though the Pico W is pin-compatible with a non-wireless Pico, we'd recommend using the specific Pico W symbol as the associated footprint has keep-out areas that tell the PCB design tool not to block the Wi-Fi signal.

Once you've highlighted the Pico W, you can click OK to return to the schematic. The symbol will be attached to the mouse pointer, so you can place it where you want to and then left-click to deposit it on the page. Now we do the same for a resistor (search resistor and select device R), and LED (search LED and select device LED).



◀ **Figure 1 :** All budding PCB designers start with an LED and a resistor – it's tradition

We'll need lots of LEDs and resistors, but only add one for now. Each LED will be attached to a GPIO pin on one side and a resistor on the other which will in turn connect to ground.

With these components added, you can switch back to the mouse pointer tool, and we'll organise the things we need. Highlight the LED by clicking in the middle of it (make sure the entire LED is highlighted as it's possible to only select part of a symbol), then click and drag it so it's near GPIO 0. The resistor is by default placed vertically, and we want it horizontal, so highlight it, and click the rotation icon on the top toolbar until it's horizontal. Now place it just to the left of the LED. We want to connect the resistor to ground, so select the ground icon (a line arrow pointing down) on the right-hand toolbar. This will open a dialog box that can be used to create a variety of power connections. Click to expand the power section, then select GND and click OK, then place the ground just to the left of the resistor. Again, rotating this to horizontal will help keep things compact.

If you zoom in so you can see the LED closely, you'll see that it's labelled LED and D1. These labels are a little way away from the symbol, and since we're about to create quite a few copies, it's a bit tidier if we click and drag them close to the symbol. This time, click into the characters rather than the symbol itself and you can drag them in. Do the same with the resistor.

Now we're ready to connect the LED, resistor, and wire together. Select the Draw Wires tool from the right-hand toolbox (a thin blue line), and you can create the connections. Each component has a small circle where a connection can join. Click into one of these and then again on the circle on the next component.

Once all the connections are made, you should have something that looks like Figure 1.

Growing the circuit

You can now select this mini circuit of LED, resistor, and ground by clicking and dragging over the lot and copy-paste it until you have 21 identical circuits. Position them so that they're close to the GPIO pins and connect each LED to a GPIO pin 0–22.

The one final thing we need to do is connect the ground on Pico to ground. We need to do this because the GND we've connected the resistors to isn't really a ground, it's what's called a network (or net for short). This means that everything in the same net gets connected together. It's very common to have one called ground like we have done. However, unless we connect the ground net to an actual ground connection, it's not really ground. We'll be supplying power through our Pico's USB port, so this is where our ground will come from.

The final schematic should look like Figure 2.

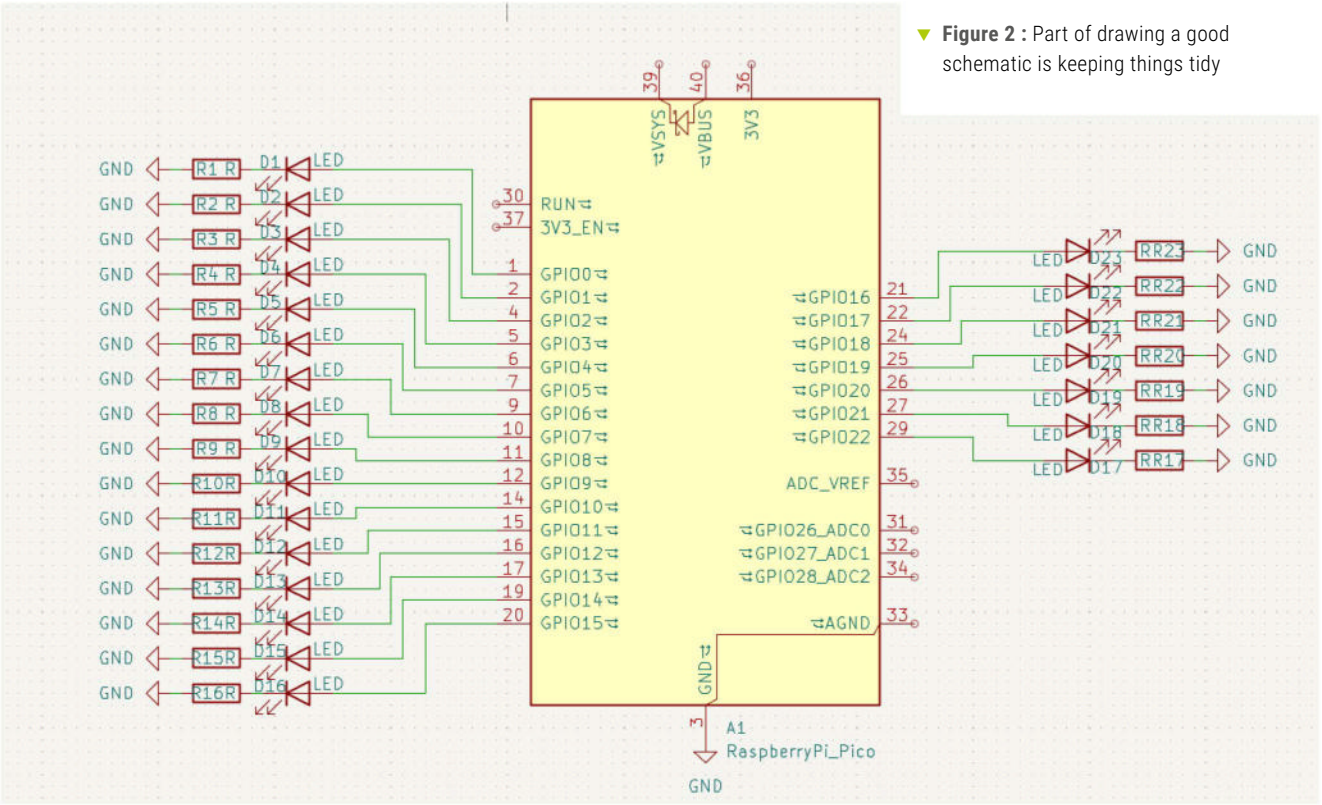
QUICK TIP

The 'Ki' part of KiCad can be pronounced either like 'key' or to rhyme with 'buy'. Both are widely used.

Code

We've written a simple MicroPython script (**brolly.py**, listed on the last page of this tutorial) to run this PCB, which you can download from **rpimag.co/brollypy**. You'll just need to put in your own latitude and longitude, as well as your Wi-Fi credentials. It grabs a forecast from Open Meteo, which is a weather forecasting service that's free for non-commercial use.

Our MicroPython script is quite simple. It just lights up for rain forecast and off for no rain. Each LED is individually controlled, so you can program your own patterns.



▼ **Figure 2 :** Part of drawing a good schematic is keeping things tidy

Leaving footprints

After designing our schematic, we need to assign components to the symbols. For example, we've placed LEDs, but we haven't said what physical package these LEDs will be in. Similarly with the resistors. Click on the Assign Footprints button in the top toolbar and in the new window, you can give a physical 'footprint' (this means the actual PCB shape) to the specific component. You can use whatever type of resistor and LED you want, but we went with through-hole LEDs (footprint 'LED_D3.0mm') and resistors (footprint 'R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal').

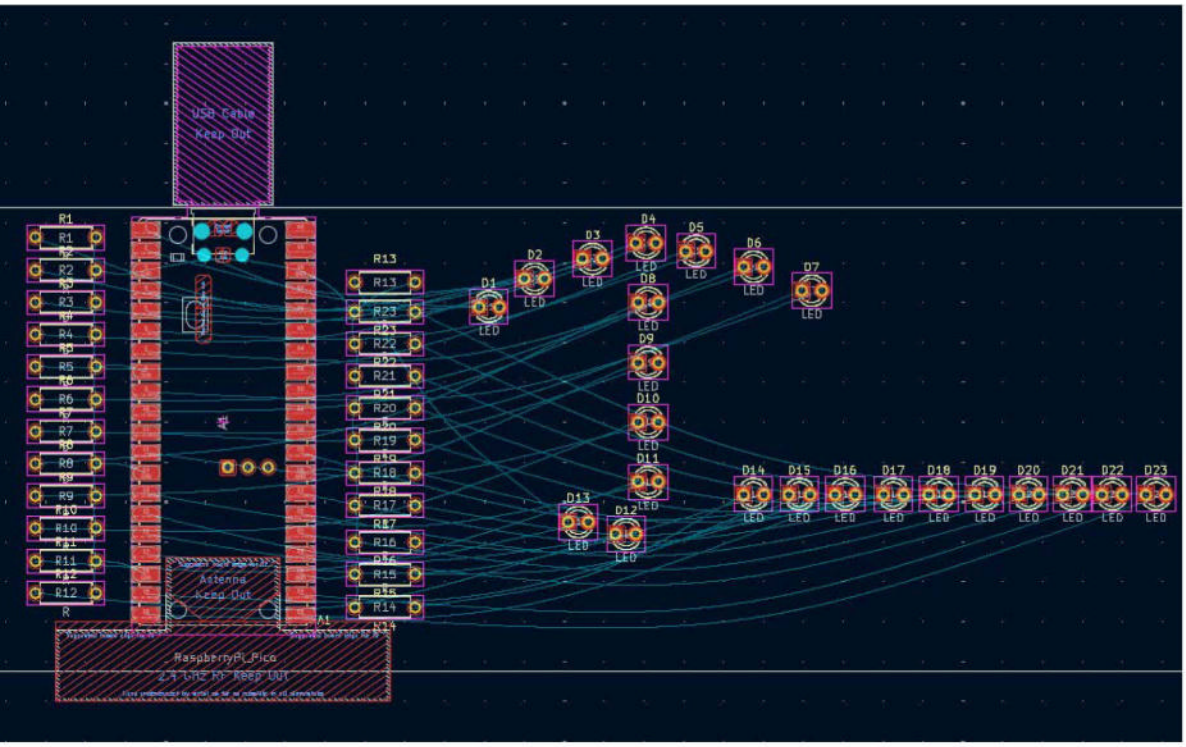
Getting on track

Now we've defined our circuit, let's design the physical PCB layout. Save your project, then click on the PCB icon in the top toolbar (Switch to PCB Editor). This will give you a blank PCB. You can bring in the components from your schematic with Tools > Update PCB from Schematic. Click Update PCB in the dialog to bring things in. You should now have a bunch of components dumped fairly haphazardly on your PCB. You'll also see a set of pale blue lines showing the connections that you need to make.

It's time to arrange things how you'd like them. Usually, at this point in the PCB design process, you want to think about how things will connect and how to organise them to make that easy. However, we're going to arrange things how we'd like them to look and then worry about connections later.

First, drag and drop the components where you'd like them. We're going to place all the components on the top layer of the PCB, so make sure F.Cu (for Front Copper) is selected in the layers box on the left-hand side.

Once you've got them where you want them, you need to define the board outline. Select the edge.cuts layer and then the Draw Rectangles tool and draw a box around where you want the edge of your PCB to be. You want the USB port on your Pico to be on the edge of the board so that there's space for the plug, and you want to make sure that there's a bit of space on the left and right for mounting holes that we'll add later (you can adjust the size of the PCB if you need to later on).



◀ **Figure 3:** You don't have to lay out your components like this – it's up to you

Your PCB should now look something like Figure 3. Everything's where we want it, but there are no traces joining the components together. Before we get to the traces, we'll connect up the ground. This can be done with traces and for a PCB like this, that would work fine. However, another option is to make a 'ground plane'. This is where you dedicate one side of the board to being just a ground connection. Across the whole thing, there's nothing but ground (actually, we'll carve a few bits out for traces in the next step, but it's mostly all ground). In this introductory article, we're not going to delve into the physics of it too much, but it's generally good practice to have a ground plane.

We're going to use the back layer as the ground plane, so select B.Cu in the layers list, then use the Draw Filled Zone tool (the blue polygon in the right-hand toolbar) and click around the outside of the PCB. The exact shape doesn't matter too much as long as it encompasses the whole PCB (going outside it is fine). After the first click, you'll be asked to select the net and you should choose GND. When you're finished, double-click to close the area.

Once you've created the outline, you can switch back to the mouse pointer tool and right-click on the new outline and select Zones > Fill All Zones. Most of the PCB should now turn blue to show that the back layer is mostly copper (if it turns a different colour, you probably have the wrong layer selected). If you zoom in, you'll see that the ground connections are all connected to this plane, but other pins aren't.

Now let's draw the rest of the traces. You can do this manually, but we've got quite a lot of traces and they're not particularly challenging ones (by which we mean they're not high-speed or likely to have problems caused by electrical interference), so it's a bit easier to let the autorouter deal with this for us. KiCad doesn't have a built-in autorouter, so we'll need to use a plug-in. Switch back to the main KiCad window (not the PCB or Schematic Editor) and click on Plugin and Content Manager. Search for Freerouting, click on Install and then on Apply Pending Changes. Freerouting does require Java to be installed, but it can do this automatically on the first run, so don't worry about it now.

Once it's installed, switch back to the PCB Editor and go to Tools, External Plugins > Freerouting. The router needs some guidance for things like how wide to make each trace and how close together traces can be put. It will get this directly from the Design Rules Checker (DRC). We're not going to look at this here because the defaults are fine for us, but as you progress on your PCB making journey, you might find that you have to adjust these values for things like high current traces, or the capabilities of a particular PCB manufacturer.

A new window will flash up and you'll see it gradually filling in the traces. Once it's filled, your PCB should be mostly complete, but the ground plane will now need to be adjusted to take account of some of the traces going across the back. Right-click on the outline of the fill zone and again use the Fill All Zones option to

Going further

This article is a really quick introduction to getting a PCB designed. We've cut a few corners to focus on getting a working product quickly and simply. As you go further in your PCB design journey, you'll need to learn to use more of the features of KiCad. The book *Design an RP2040 Board with KiCad* is designed to help you do just that.

regenerate it. You should now see that any traces on the rear of the board (ones in blue rather than red) have a blank outline where the ground plane is broken.

This is our board electrically complete now, but we need to add a couple of bits – some mounting holes and some text. Mounting holes are footprints, so click on the Place Footprints tool (shaped like a chip in the right-hand toolbar). Search for 'mounting' and you should have a range of options. Most are sized with metric machine screw sizes. We're going to use wood screws to mount our PCB, so we needed a fairly large hole. Mounting holes often have an electrical connection to ground, but we don't want that, so we've selected MountingHole_4.3mm_M4_ISO7380. You can then place the component. We want four of these – one in each corner. You should be able to do this without breaking any traces, but you can expand the PCB outline if you need to (though make sure to refill the ground plane if you do).

The mounting holes have two bits of text with them describing the components. We deleted these to make it a bit tidier. We've not worried about the exact position of the holes and aimed to keep them symmetrical and looking correct. However, if you're planning on matching the PCB with a 3D print or other automatically fabricated part, you'll need to place them precisely.

For the final bit of text, we just wanted to highlight what the PCB actually was. First, select the front silkscreen layer (f.silkscreen) and then the text tool. You'll be able to edit the text, select the size, and then place it on the PCB. You should, finally, have a PCB that's something like Figure 4. You can check everything looks good by going to View > 3D viewer. This gives you a 3D rendering of your PCB and is a great way of checking that you've added everything to the right layer.

Factory files

Now we've got everything set up how we want it, we need to create some files to upload to our PCB manufacturer, and the standard for this is Gerber. A Gerber file is actually a zip file containing separate files for each layer of our PCB and a drill file for the holes.

Most PCB manufacturers use Gerber files, but unfortunately, they're not completely standard and there are a few options you might need to change. We used PCBWay, and the defaults worked fine for us. However, if you're using a different manufacturer, take a look at their documentation for what settings they need.

Project files

We'd strongly recommend that you go through the process of designing the board in KiCad yourself. However, if you want to compare your work to ours, the KiCad files are available at **rpimag.co/brollykicad**.

You should find both the schematic and the PCB design there.

Go to File > Fabrication Outputs > Gerbers. First, select an output directory. This should be an empty directory. Give it a sensible name and maybe add a version number into the name just in case you end up needing to redo it. Click Generate Drill Files > Generate, then Plot. Together, this should create all the files you need. Finally, you just need to compress the directory to a zip file using your operating system's tools.

The process for getting it manufactured is a bit different for each PCB manufacturer. We're using PCBWay, so head to pcbway.com, then click on Instant Quote and Quick Order PCBs. Here you'll be able to upload the Gerber file zip and see a render of the PCB. If this doesn't match what you saw in the KiCad render, then stop and take a look first. If you can't work out why, use PCBWay's chat feature to ask for help – we've found them very helpful in the past.

There are a lot of options, but for the most part, the defaults are fine. The only one we changed is the Surface Finish to HASL Lead Free. Lead in hobbyist electronics isn't always a huge concern, but the price difference is pretty minimal, so we prefer to remove the health risk, especially on an exposed PCB.

That's all there is to it. Select your postage, and complete the purchase. It can take a week or two to get your PCBs (depending on the postage selected). Once they arrive, solder them up and program your Pico (see 'Code' box), then enjoy the smug feeling of always having your brolly when you need it.

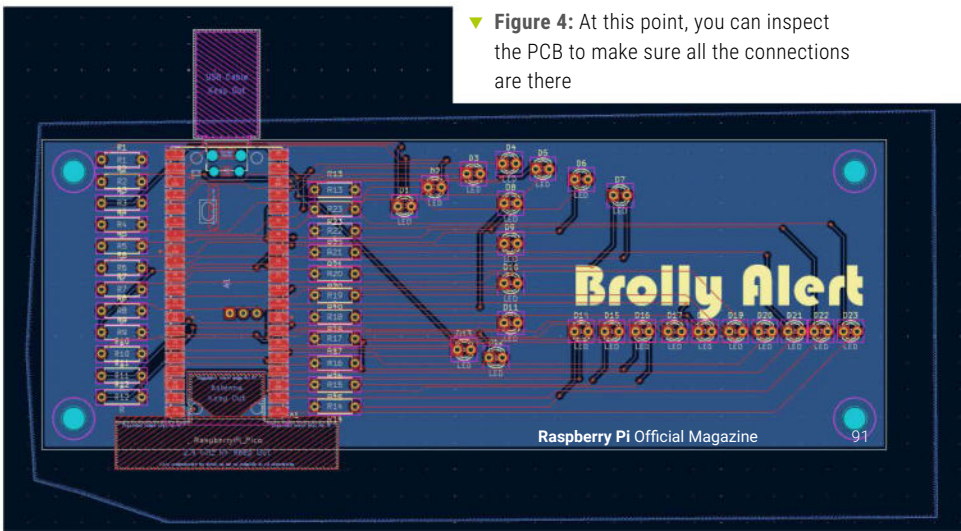


Figure 4: At this point, you can inspect the PCB to make sure all the connections are there

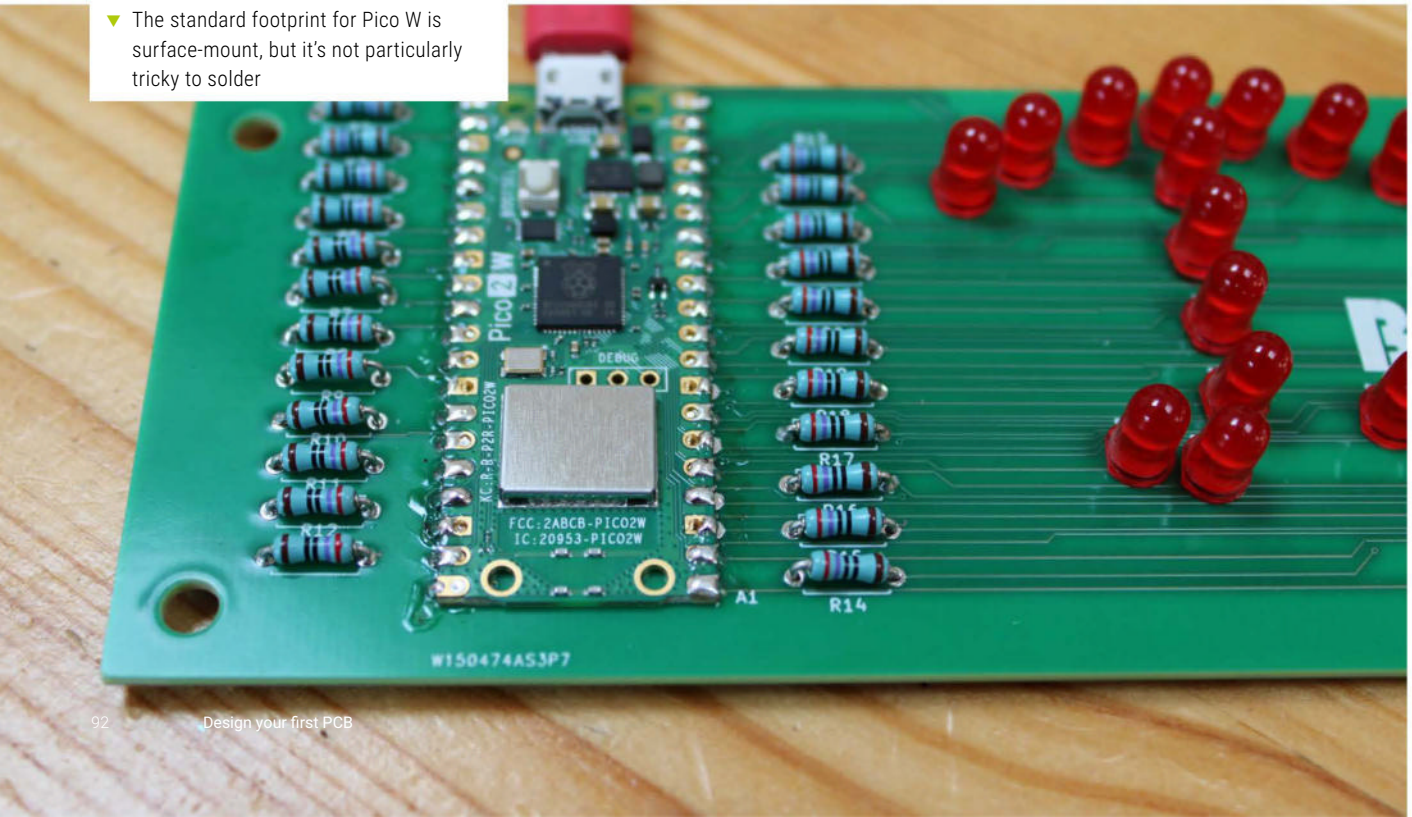
brolly.py

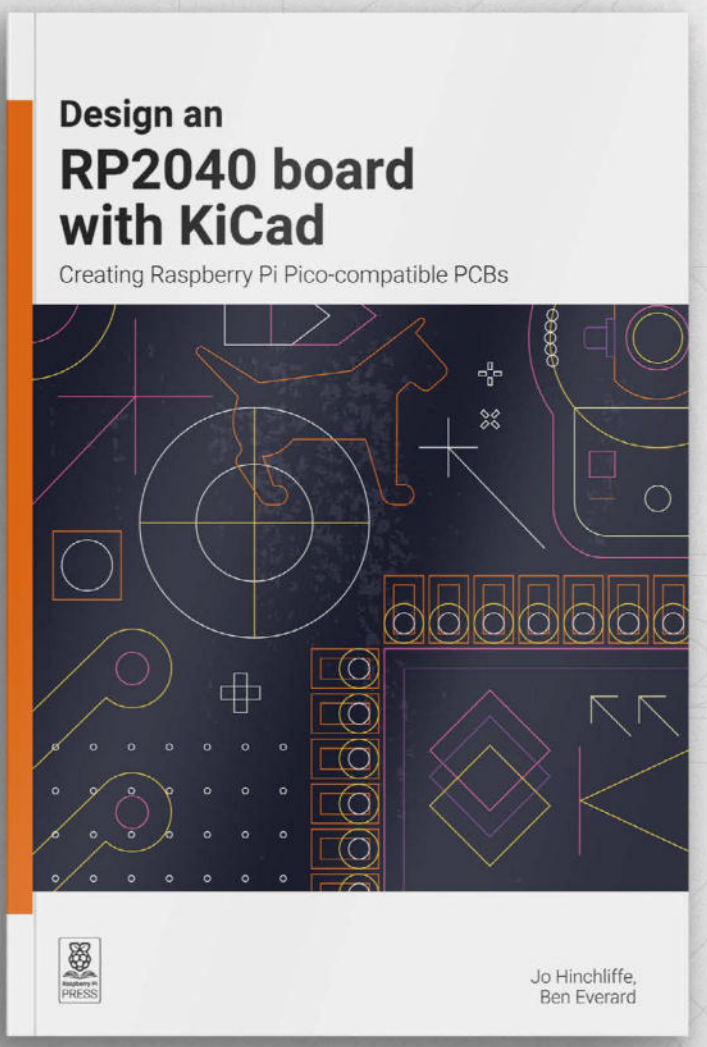
> Language: **MicroPython**

DOWNLOAD THE FULL CODE:
rpimag.co/brollypy

```
001. import machine
002. import time
003. import network
004. import socket
005. import urequests as requests
006. import json
007.
008. ssid = 'YOUR_WIFI_SSID'
009. password = 'YOUR_WIFI_PASSWORD'
010.
011. lat = "52.52"
012. long = "13.41"
013.
014. url = "https://api.open-meteo.com/v1/
     forecast?latitude="+lat+"&longitude="+long+
     "&daily=precipitation_probability_max&time
     zone=Europe%2FLondon&forecast_days=1"
015.
016. wlan = network.WLAN(network.STA_IF)
017. wlan.active(True)
018. wlan.connect(ssid, password)
019. while wlan.isconnected() == False:
020.     print('Waiting for connection...')
021.     time.sleep(1)
022.
023. print(wlan.ifconfig())
024.
025. pins = []
026. for i in range(23):
027.     pins.append(machine.Pin(i, machine.Pin.
     OUT))
028.
029.
030. while True:
031.     res = requests.get(url=url)
032.     if (float(res.json()['daily'][
     'precipitation_probability_max'][0])) > 0.3:
033.     #run for approx one hour
034.         for pin in pins:
035.             pin.on()
036.     else:
037.         for pin in pins:
038.             pin.off()
039.     time.sleep(60*60)
```

▼ The standard footprint for Pico W is surface-mount, but it's not particularly tricky to solder





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

The many and varied uses of photography on Raspberry Pi

Capture the world around you in new and interesting ways.

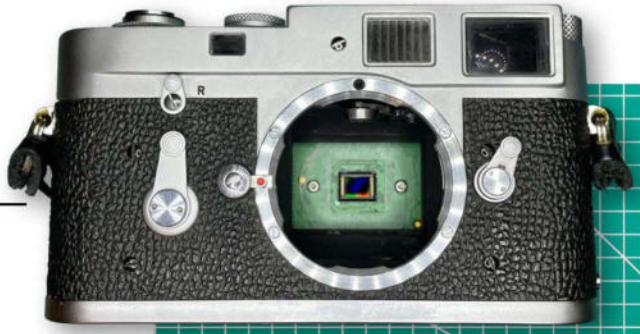
By **Rob Zwetsloot**

Raspberry Pi camera projects have been around for about as long as Raspberry Pi itself – folks were plugging USB webcams into Raspberry Pi long before the first Camera Module ever came out a year later. Now there are all manner of Raspberry Pi Camera Modules and even better webcams to use, along with builds and makes of every variety. Here's just some of the very cool things you can do with a Raspberry Pi and a camera. Say cheese!

MPi point-and-shoot camera

Upcycle an old camera with a Raspberry Pi, film included

rpimag.co/mpi



The HQ Camera Module uses lenses that have a manual focus you can twist, much like cameras of old (and now if you turn autofocus off), and maker Michael Suguitan wanted to capture the original method of camera operation of a Leica M2 using a HQ Camera Module and a Raspberry Pi Zero.

It actually shoots to film and uses the original mechanical shutter. While Michael did add an LCD screen to the rear of the body, he ended up not using it so that he can once again capture the real feeling of old photography – namely, the agonising wait hoping the photo turned out.

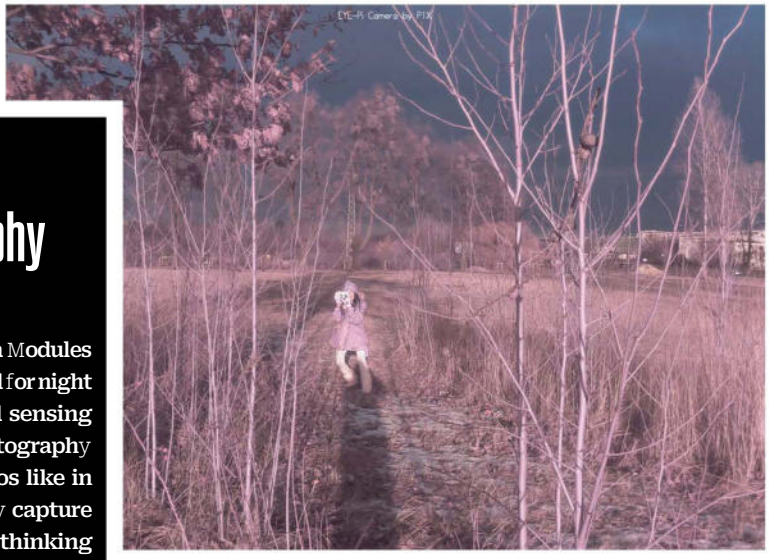
▲ The original range finder is incorporated into the build

▶ For Ultraman, cheese is just SHUWATCH

Eye-Pi IR – infrared photography

rpimag.co/eyepi

The various Raspberry Pi Camera Modules have a NoIR variant – usually used for night photography or other infrared sensing builds. Using it for normal photography creates surreal, beautiful photos like in this Eye-Pi IR project that really capture the imagination – we've been thinking about it for ten years, after all.



◀ It's almost like a Hollywood post-apocalyptic colour grade

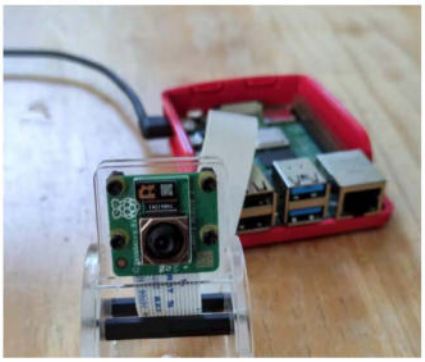
Time-lapse photography

See the world in fast-forward

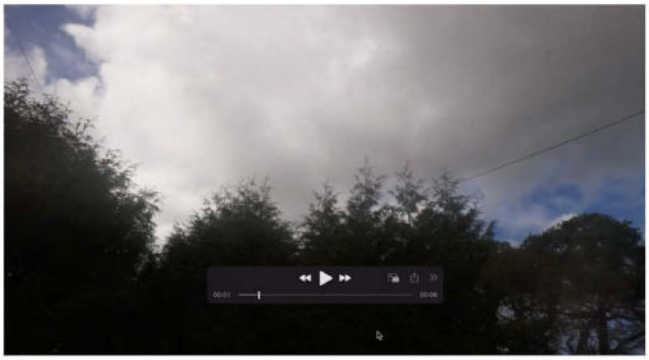
rpimag.co/timelapsevid

As Raspberry Pi Camera Modules are controlled via code or terminal commands, it's very easy to set one up to take photos on specific triggers. A time-lapse video is basically just a series of photos taken at regular intervals that are then strung together – an easy task for Python.

This version is a tutorial from a couple of years ago, so the code will reference libcamera – this has since been renamed to rpicam. It works the same – you'll just need to replace, e.g., `libcamera-still` with `rpicam-still`. You can also find the updated version in *The Official Raspberry Pi Camera Guide 2nd Edition*: rpimag.co/cameraguide.



◀ The setup is very simple – although you may need to weatherproof it if you plan to set up outside



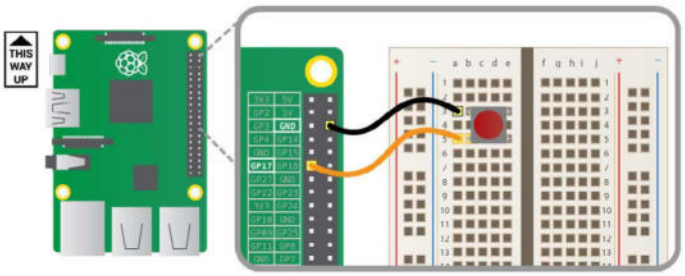
▼ Watch the clouds go by, a favourite British pastime

Astrophotography

rpimag.co/128

Astrophotography works very similarly to time-lapse photography in that it can use multiple photos to create an image. The trick is to have long exposure times, and several hours spare on a dark hill in the countryside. The things you'll see, though...

▶ Star trails are made by compositing multiple photos on top of one another



▲ It's a very simple circuit and Code Club makes it very easy to set up and program
◀ The orange man is what you make when there's no snow

Stop-motion animation

Turn real life into animated shorts

rpimag.co/stopmotion

A classic film-making and animation technique that just about anyone can understand. It's so universal that the Lego movies are entirely computer generated yet replicate the style. It couldn't be easier – set up a scene, snap, set up the next frame.

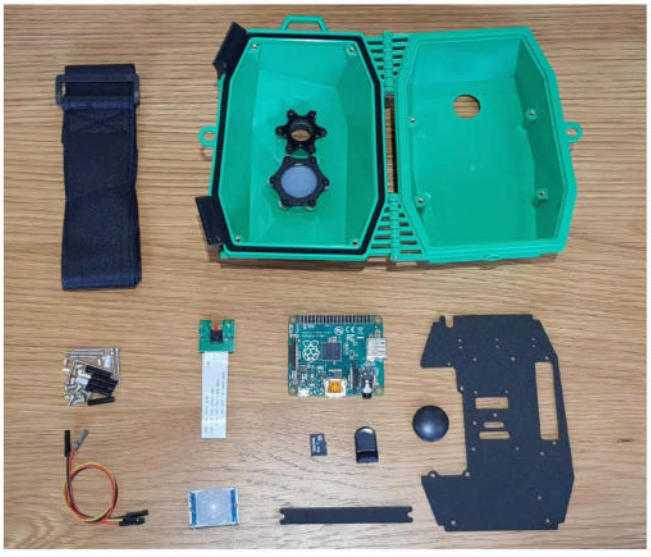
Code Club, part of the Raspberry Pi Foundation, has a great tutorial on how to create your own version with a simple push-button. We've also seen versions that overlay a ghostly image of the previous frame you took so you can better position your next shot.

CinePi – video camera

rpimag.co/cinepixl

The Python library rpicam is fully capable of taking video as well as photos – including high-frame-rate videos for slow-mo shots. CinePi is one of the most impressive applications of this, aiming to create a true cinema camera powered by a Raspberry Pi, with your imagination the only limit.

▶ The beautiful CinePi XL is a fully functional cinema camera



Naturebytes – nature trap

See what beasties lurk in your garden

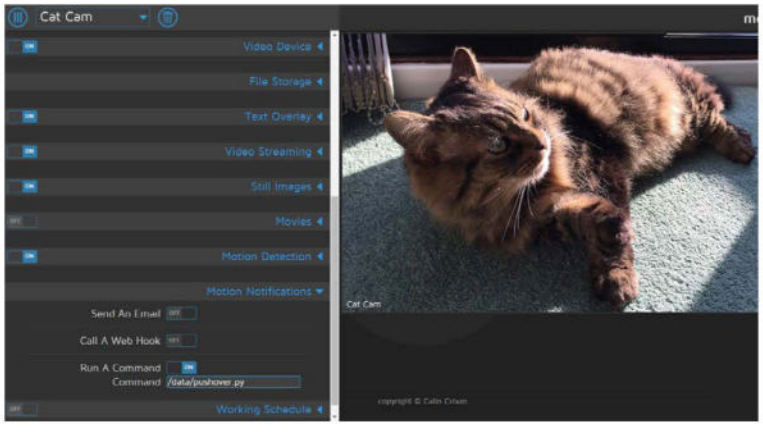
naturebytes.org

This classic Raspberry Pi kit comes with everything you need to get shooting your local wildlife – a weatherproof case, a motion sensor, and ways to affix it to a tree. You can provide your own Raspberry Pi and camera, or get a kit that has one included.

Very simply, once a threshold of motion is detected by a PIR sensor, the camera will take photos and/or record a little video snippet. You can add a NoIR camera and IR LEDs for night-time shots (especially if you want to see some hedgehogs).

▲ A Naturebytes camera in its natural setting. What wonders will it see?

◀ The kit uses a Raspberry Pi A+ and comes with a preloaded SD card

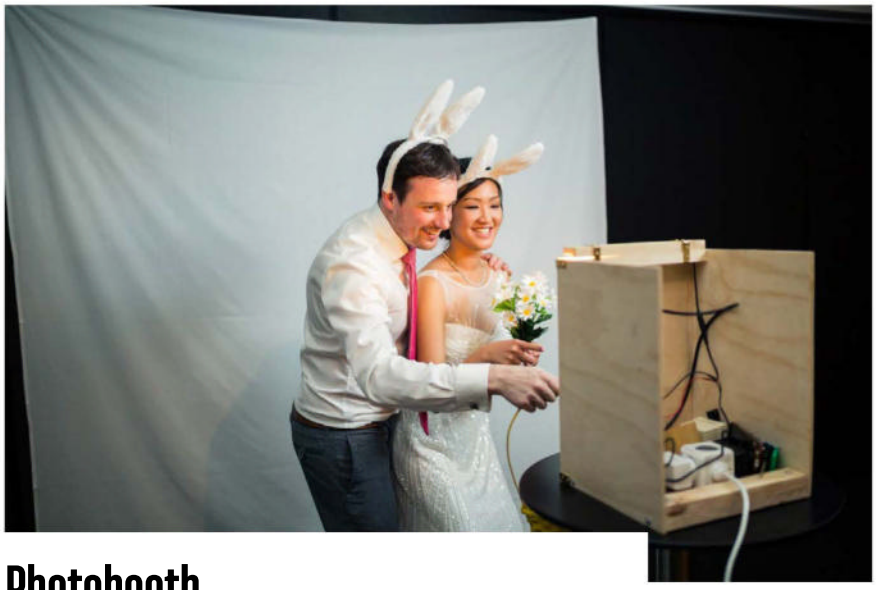


Security camera

rpimag.co/securecam

Home security can be important, and with the custom MotionEyeOS for Raspberry Pi, you can set up your own little network of security cameras in your home. Pi My Life Up has a great little tutorial on how to do this, and includes the use of standard webcams.

◀ You can also use the same tech to keep an eye on adorable friends



If you can't have overlays, you can always use physical props

The box took about 20 hours to build

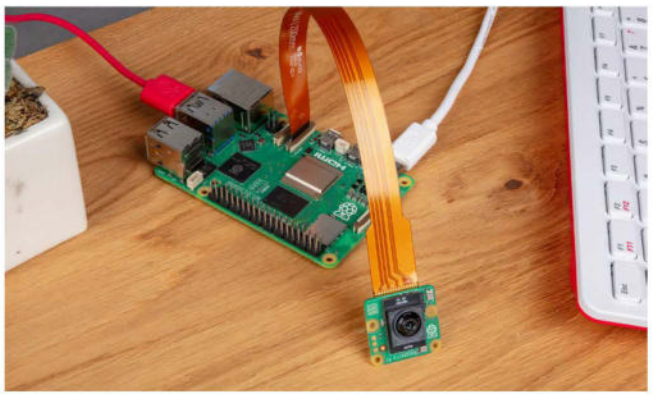
Photobooth

Make weddings and other events something folks will remember

rpimag.co/photobooth

Photobooths can be very simple or quite complex. On the surface, you just need a big button to activate a camera. Perhaps you'll have a countdown, and a screen so people can frame themselves properly. The truly good ones have overlays with stickers that you can move around and even printers.

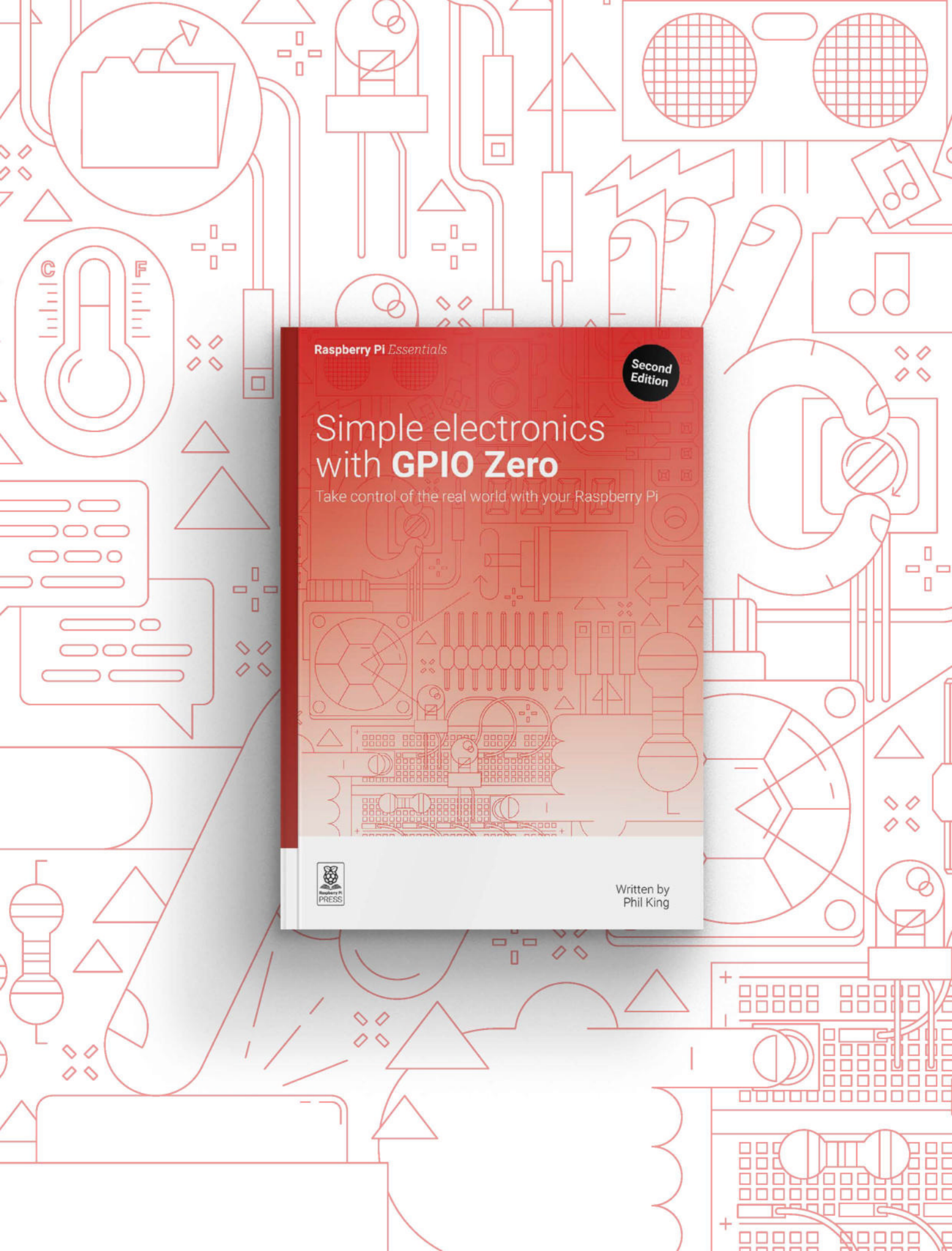
This version was maker Jack Barker's first Raspberry Pi project, and while it didn't have a printer, it was finished in time for his own wedding. It successfully took photos of guests, saved to an SD card to retrieve later, and was apparently a great conversation starter.



AI image recognition

In the previous issue (rpimag.co/154), we had a tutorial on using Raspberry Pi AI Camera to identify Raspberry Pi models. These models can be trained on various things, such as for a face-detection lock or even checking a production line for faulty products.

Raspberry Pi AI Camera is ideal for computer vision projects


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

Simple electronics with **GPIO Zero**

Take control of the real world with your Raspberry Pi

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

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

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

DreamHAT+

Discover high-frequency mmWave radar using the Infineon BGT60TR13C. By **Lucy Hattersley**

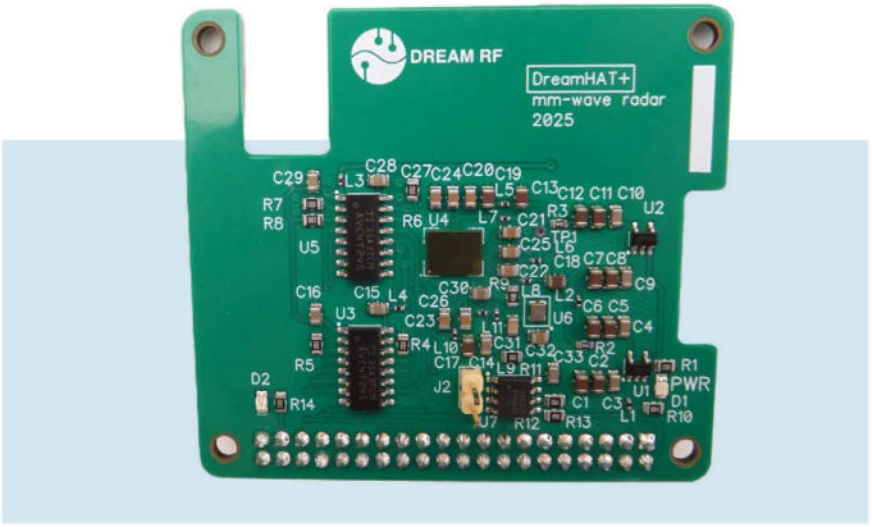
Dream RF **dreamboards.co.uk** £100 / $135

SPECS

OPERATION:
Frequency at 58–63.5GHz; Bandwidth at 5GHz

RANGE:
Distance: 0.1–15m; Resolution: 3cm, Field of view: 40° horizontal, 65° vertical

INTEGRATED ANTENNAS:
1 × TX antenna, 3 × RX antennas, 5dBi (max) gain



▶ The DreamHAT+ board with the Infineon BGT60TR13C

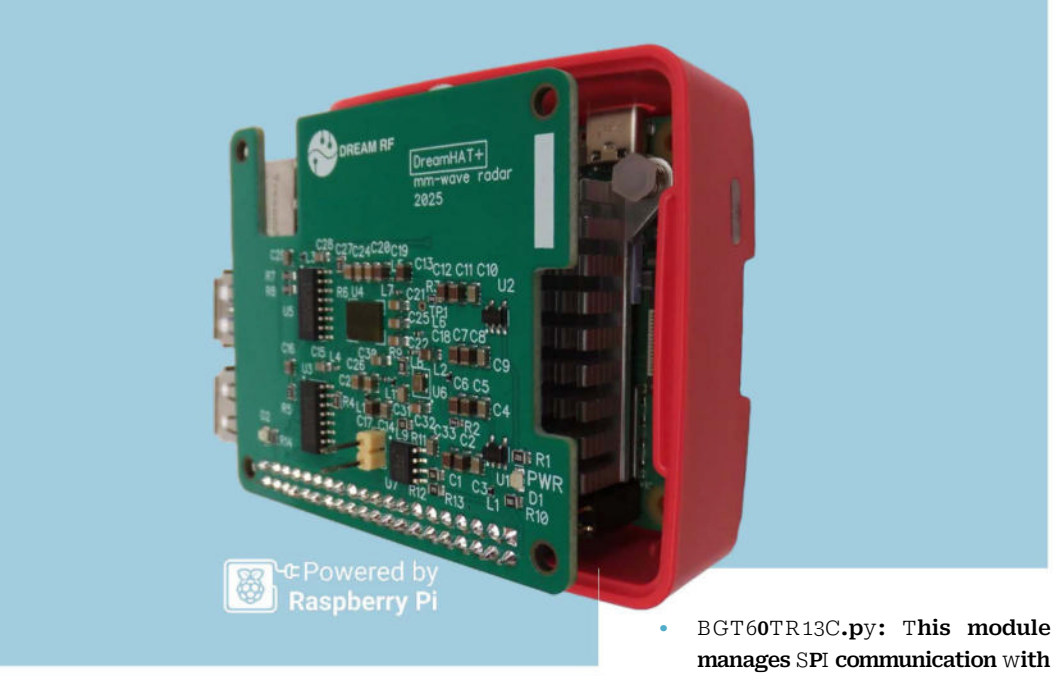
DreamHAT+ is a Hardware Attached on Top (HAT) kit that gives Raspberry Pi a 60GHz millimetre-wave radar for you to play with.

The full-size HAT packs an Infineon BGT60TR13C chip featuring a single transmitter antenna and three receivers (TX/RX), connecting to Raspberry Pi 4 or 5 via GPIO pins and SPI.

Dream RF also manufactures ApRESS (Autonomous Phase-sensitive Radio Echo Sounder), a device that uses Frequency-Modulated Continuous Wave (FMCW), as used to monitor ice loss in Greenland. While that device operated on the 2–8GHz frequency, DreamHAT+ is operating on a much shorter wavelength: 60GHz. This gives it a much higher resolution, making it ideal for human detection and gesture recognition-type projects.

As well as using it to research personal radar (a perfectly fun and relatively low-key technology), it can be used to add situational awareness to robotics, security systems, and wildlife monitors. There are also health and industrial uses (mostly involving monitoring breathing patterns or equipment vibrations).

The latter use case is beyond our scope, but we had a lot of fun experimenting with DreamHAT+.



Get started

Setup – like most HATs - was relatively simple. We connected DreamHAT+ to a test Raspberry Pi 5 and set up the software. Dream RF supplied us with a preconfigured microSD card, which you can flash yourself using the link on Dream RF's GitHub (rpimag.co/dreamhatgit).

You can also start from a Raspberry Pi OS installation and follow the instructions to install the various packages and download the example projects. We recreated both approaches and found flashing the microSD card easier.

On the resources section of the DreamBoards site (dreamboards.co.uk) is a setup guide, product brief, and schematics. You can also look at some real-world examples on Dream RF's YouTube channel (rpimag.co/dreamrfvid). There are currently the 2D Cartesian Plot and Range Doppler examples. It's a good idea to read the documentation and watch the videos first so you get an understanding of the screen output.

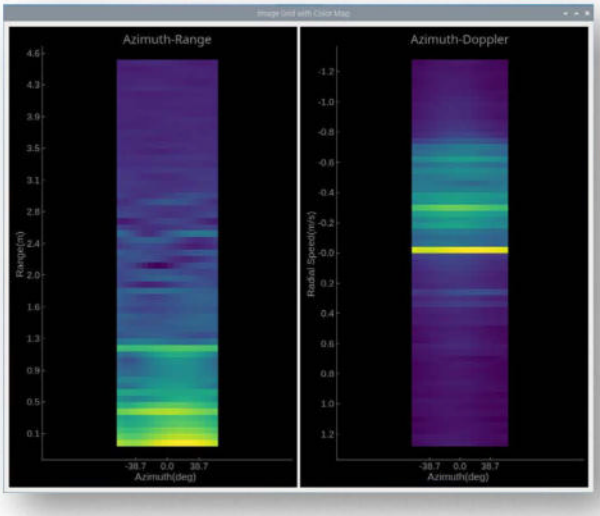
Software stack

There are a range of example projects in the GitHub repository. These include a range-Doppler plot (the classic radar visualisation), an xy plot with tracking and persistence, a Doppler-azimuth plot (that detects objects around the device), and an offline processing example.

Beyond that, you can look at the Python modules to start your own implementations:

- BGT60TR13C.py: This module manages SPI communication with the Infineon BGT60TR13C chip.
- udp_streaming.py: This streams radar data over User Datagram Protocol (UDP).
- gui.py: Provides a graphical user interface for visualising radar data, including range-Doppler plots and heatmaps, facilitating intuitive analysis.
- offline_processing.py: For post-processing of recorded radar data.

While these modules provide a solid Python-based application programming interface API, the documentation around the API isn't as solid as the setup. "This is more than enough to get you started and on the way to creating your own unique 60GHz radar project!" states the GitHub page, somewhat optimistically. Your mileage may vary.



▲ DreamHAT+ assembled inside a case. We found it best positioned horizontally for testing

▼ The DreamHAT+ Doppler-azimuth script

We had a lot of fun experimenting with DreamHAT+

Verdict

We thoroughly enjoyed our time experimenting with DreamHAT+ and the Infineon BGT60TR13C. The hardware and example projects are first-rate. The documentation is in development and a more detailed API reference guide would be helpful for moving beyond the initial projects.

8/10

ALPON X4

This ruggedised Compute Module 4-based computer is ideal for industrial use. By Phil King

Sixfab **rpimag.co/alponx4** £462 / $550 / €550

SPECS

FEATURES:
Compute Module 4, aluminium enclosure with built-in heatsink, 4 × status LEDs, 2 × push buttons

RAM / STORAGE:
8GB LPDDR4, 32GB eMMC

PORTS:
100Mbps Ethernet port, Gigabit Ethernet port, 2 × USB-A, HDMI, GPIO (RJ45, requires adapter), USB-C PD (15V DC 1.8A, 27W), screw terminal block (9–30V DC), PoE+ (optional)

CONNECTIVITY:
Wi-Fi (2.4GHz and 5.0GHz), 4G LTE (via built-in eSIM), GPS/GNSS

▼ Power is provided via USB-C PD with the PSU, or 9V–30V DC to the screw terminals

◀ The ALPON X4 has a range of ports and four antennas for wireless connectivity

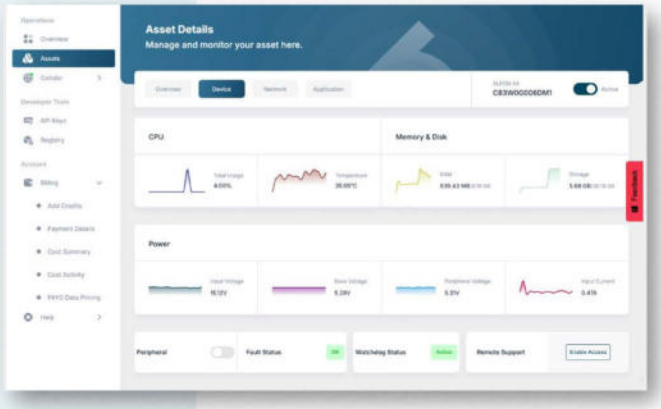
Can an industrial edge computer really be plug and play? The ALPON X4 offers the simplest setup process we've seen. Just register and activate it in your online Sixfab Connect account – the unit even has a QR code to scan to automatically enter its serial number – and then plug it into the supplied 27W USB-C PD power supply. Using its built-in eSIM, the X4 will then connect to the cloud, enabling you to monitor and control it remotely from a web interface. Twin 4G LTE antennas enable it to connect in even the most patchy cellular coverage areas (such as your reviewer's home).

You get three months of ALPON Cloud (plus a 120MB total LTE data allowance per device) free for the first five devices – after that, it's $6 monthly per device, plus mobile data costs (pay-as-you-go or data pool). You may well opt to use Wi-Fi, easily set up via the cloud portal, or a wired Ethernet connection.

Cloud control

While you can connect the ALPON X4 to an HDMI display and USB keyboard to directly access the command line of its custom Raspberry Pi OS Lite 64-bit system, it's just as easy to open a remote terminal from the Sixfab Connect cloud portal.

Two 4mm screw holes can be used to mount the device to a DIN rail or onto a wall



The Sixfab Connect cloud portal enables you to monitor numerous aspects of the device

Applications are all run in Docker containers and can be deployed from the web interface. The simplest and quickest method is to use pre-built ARM64-based images from Docker Hub. Alternatively, you can create and upload your own custom app images to the Sixfab Registry to deploy them manually. Installed apps can be monitored, paused, and restarted from the web interface, along with the option to open a remote terminal into the app container.

In addition, the Sixfab Connect web interface features Device and Network tabs to show you all manner of useful system info, such as CPU activity, temperature, RAM, power, faults status, network status, uptime, and eSIM data usage.

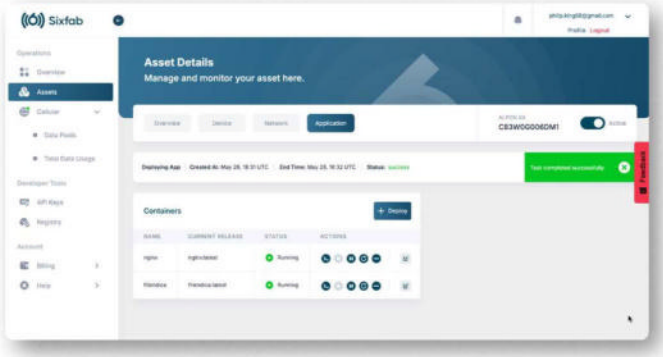
Powerful system

Based around a Raspberry Pi Compute Module 4 with 8GB of RAM and 32GB of eMMC storage, the device itself offers plenty of power in a compact form factor of 111.16 × 99.9 × 33mm, plus the four antennas (including one for GNSS), which can be folded into a 90° position for desktop use or laid flat if you're wall-mounting the unit on a DIN rail. It's quiet too, with a fanless design and a built-in heatsink inside the full aluminium enclosure with an IP40 rating for dust protection in industrial environments.

The extensive online documentation includes tutorials for numerous possible configurations for industrial Internet of Things (IoT) setups using common protocols such as Modbus, Supervisory Control and Data Acquisition (SCADA), and Message Queuing Telemetry Transport (MQTT). As well as connecting local devices via wireless, USB, or Ethernet, there's a GPIO port whose RJ45 connection can be expanded into a terminal block using an optional adapter. This breaks out six GPIO pins plus ground and 5V power, routed through a voltage-level converter to ensure stability.

Other use cases covered in the online documentation include a Pi-hole ad blocker, Openfire private chat server, nginx web server, PiSignage digital signage, and Edge Impulse machine learning. It's a very versatile piece of kit.

The ALPON X4 offers the simplest setup process we've seen



Applications are run in Docker containers and can be deployed and managed in the web interface

Verdict

A powerful edge computer with excellent wireless connectivity, a robust enclosure, convenient cloud-based management, and ultra-easy initial setup.

9/10

ONLY THE **BEST**

Motor and servo controllers

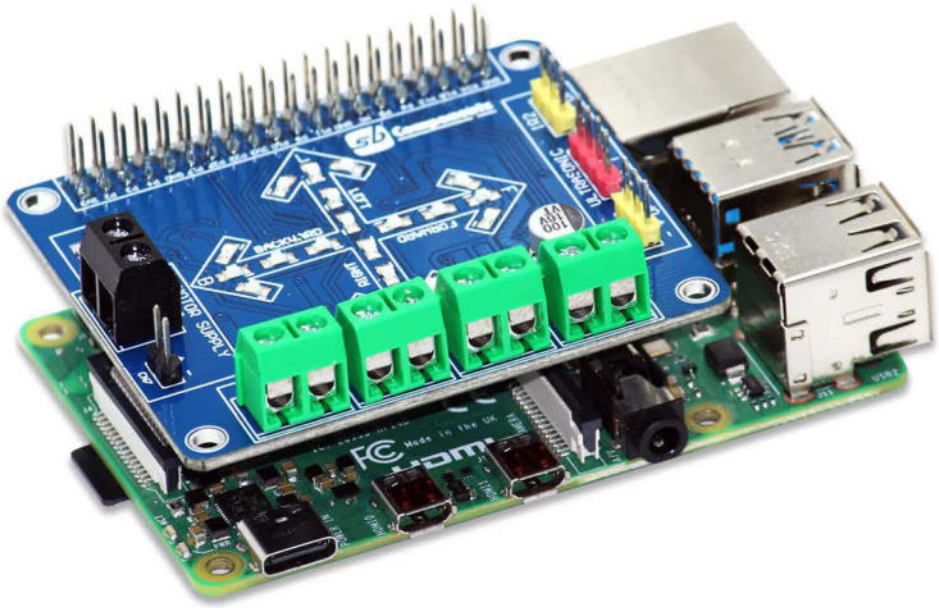
By **Phil King**

There's nothing quite like a project that moves. As well as putting them in robots, you can use motors to create physical movement in all sorts of projects, ranging from a CNC drawing machine to a guitar tuner or a plant rotator.

To control motors with a Raspberry Pi computer or Pico microcontroller, you'll need a driver board equipped with an 'H-bridge' that enables the direction of rotation to be reversed. We take a look at some of the best ones around, from the basic to the more elaborate, and with a range of maximum power outputs.

Drivers can also be used to control stepper motors, which rotate in small steps for very precise movement – which is why they're used in machines such as 3D printers. You need two motor outputs from your board to drive a single stepper.

Finally, servo motors rotate between two positions, their current angle being set by the PWM (pulse-width modulation) signal being sent by the controller. We look at a board to make it easier to control multiple servos.



Motor Shield for Raspberry Pi

SBC | £15 / $21 | sb-components.co.uk

There are numerous motor controller boards available, many of which are of the dual H-bridge type that will drive two DC motors, forward or in reverse, or two sets of motors for the front and rear of a four-wheeled robot. This Motor Shield, however, has four channels – thanks to L293DD chips – so it can drive four motors individually, enabling true four-wheel drive. Alternatively, it can control a couple of stepper motors.

There are eight screw terminal blocks to connect the wires of your motors, plus another two terminals (or male pins) for a power supply, which needs to be in the range 6V–24V – note that Raspberry Pi has to be powered separately. Up to 600mA of current (1A peak) is supplied per channel, which should be enough for most small wheeled robots.

In addition, there are breakout headers for two IR and one ultrasonic sensor, all with 3.3V output protection. Other notable features include programmable LED-lit direction arrows and a GPIO stacking header. Programming is made easier by a dedicated Python library.

▲ Control up to four DC motors, or two steppers, individually

Verdict

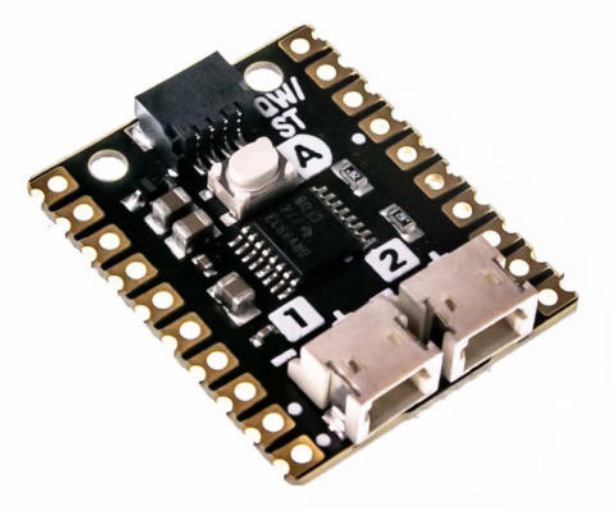
Four output channels are better than two, plus bonus features.

Motor SHIM for Pico

Pimoroni | £10 / $11 | pimoroni.com

Due to its smaller form factor, Raspberry Pi Pico is a popular choice for controlling smaller robots. And if you want to make a truly tiny robot, this miniature motor driver is ideal. You'll need a Pico with male pins soldered to its GPIO headers, then you can slot the Motor SHIM onto the 20 pins nearest its USB port, leaving the other 20 free. Solder it to the pins for a more permanent fit, or use a couple of female headers soldered to the SHIM if you want to remove it from Pico later.

The Motor SHIM features a DRV8833 dual H-bridge motor driver, enabling it to drive two DC motors. Rather than screw terminals, the SHIM features two small two-pin JST-ZH connectors. For each, you just plug in one end of a cable, and the other end into a JST-ZH connector soldered to a DC motor – cables and connectors are available from Pimoroni. Full MicroPython and C++ libraries are provided.



Verdict

At half the size of Raspberry Pi Pico, it is truly tiny.

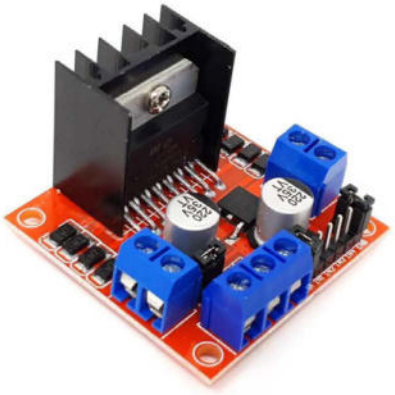
▲ Bonus features include a user button and Qw/ST port

L298N Motor / Stepper Driver

The Pi Hut £4 / $5 | thepihut.com

If you just need a standard dual H-bridge motor driver to control a couple of brushed DC motors, or a single bipolar stepper motor, this one offers exceptional value for money. It's based around the classic L298N chip, one of the most widely used components in electronics and robotics. Just connect some GPIO PWM outputs from Raspberry Pi to its four input pins to control the direction and speed of your motors.

Peak output per channel is 2A, or 1A continuous, which is enough for most small robots and motorised projects. Power supply is via screw terminals, in the range 7V–30V. Note that at a 20V or higher input, and with the onboard 5V regulator enabled, the board can get a little warm, even with its built-in heatsink. Nothing to worry about, and the flyback EMF diodes offer added protection.



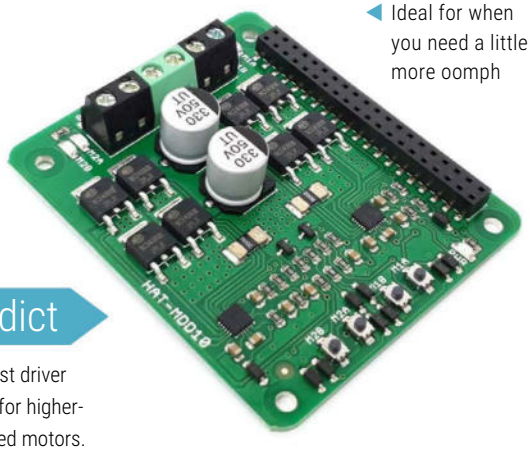
Verdict

A more basic dual motor driver, but great value for money.

◀ An excellent budget option for controlling motors

10A 6V–24V DC Raspberry Pi Motor Driver HAT

Cytron | £22 / $28 | cytron.io



Ideal for when you need a little more oomph

Verdict

A robust driver board for higher-powered motors.

If you need to control more powerful motors for your project, you'll need a driver that can supply the required current. This one from Cytron can supply 10A of continuous current (30A peak for up to ten seconds) to two DC motors or a single stepper motor. As the name suggests, power input is in the range 6V–24V. It comes in the convenient form of a HAT with a GPIO pass-through header retaining full access to Raspberry Pi's pins.

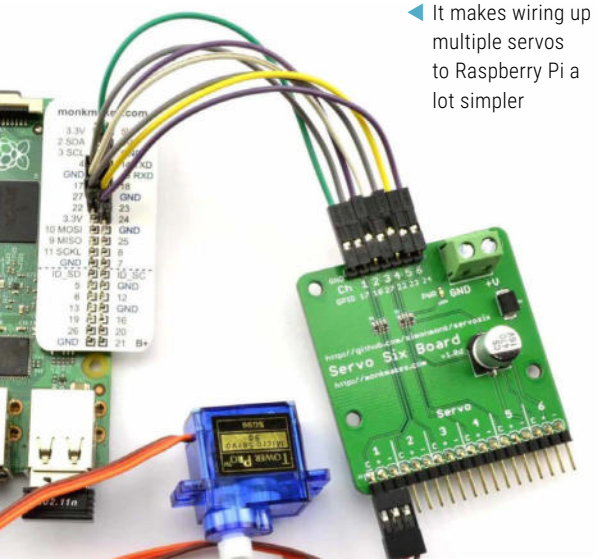
The board is built with solid-state components for quicker response times and lower wear and tear. The dual H-bridge is made using only N MOSFETs for improved efficiency without the need for a heatsink. Interestingly, the driver can accept two types of PWM signal: locked-antiphase and sign-magnitude (with separate PWM and direction signals). Nice touches include motor test buttons (forward and reverse for each) and LED indicators.

Servo Six

Monk Makes | £6 / $8 | monkmakes.com

Verdict

A low-cost board for controlling multiple servos.



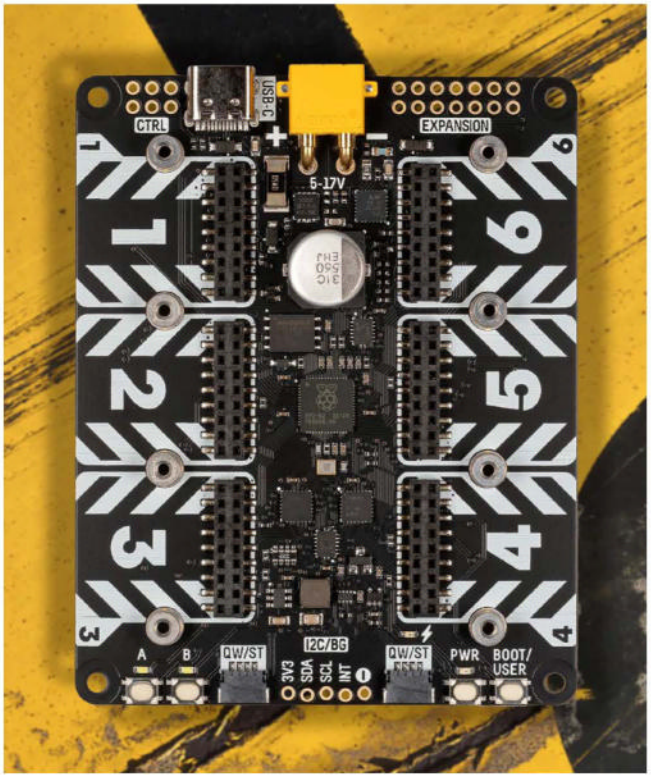
It makes wiring up multiple servos to Raspberry Pi a lot simpler

Unlike a standard DC motor that spins continuously, a servo motor rotates back and forth between minimum and maximum angles. They're used for precision control in things like robot arms. While you can control a lower-power servo directly from Raspberry Pi's GPIO pins, using PWM to set the servo's angle, a HAT like this simplifies the process, especially when using multiple servos.

As its name suggests, the Servo Six can control up to six servos. It's not a HAT, so you need to use jumper wires to connect its inputs to seven pins on Raspberry Pi and then connect your servos using the 18-pin header on the Servo Six's opposite edge. You can easily control the servo angle in Python using GPIO Zero's AngularServo class. The board's power input (via screw terminals) is 3V–12V and it has reverse polarity protection.

Yukon

Pimoroni | From £24 / $27 | pimoroni.com



Based around the same RP2040 microcontroller chip as Raspberry Pi Pico, Pimoroni's Yukon modular ecosystem aims to make advanced robotics easier. The main host board can be equipped with up to six modules which can be swapped in and out as you need for projects – no soldering needed. You can buy a starter kit with some modules, and other extras, included.

Modules available from Pimoroni include various motor and stepper controllers, along with those for LED strips, audio, and wireless communication (which Yukon doesn't have built in). There's even a 'Proto' module that you can customise for your own needs.

When powered with 5V–17V via an XT30 input, such as with a LiPo battery, the Yukon can deliver a whole lot of power – up to 15A of continuous current – with e-Fuse protection. Combined with the swappable modules, this makes it a highly versatile system for experimenting with custom robotics and other projects. A MicroPython library makes it easy to program.

Verdict

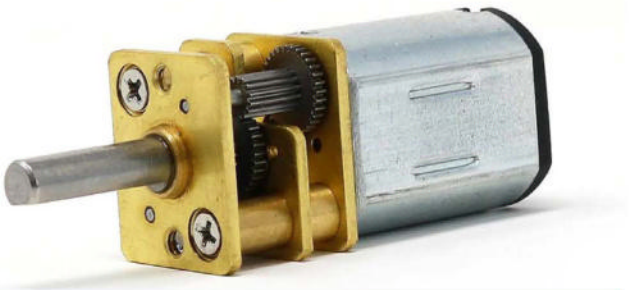
Control motors and a whole lot more with ease.

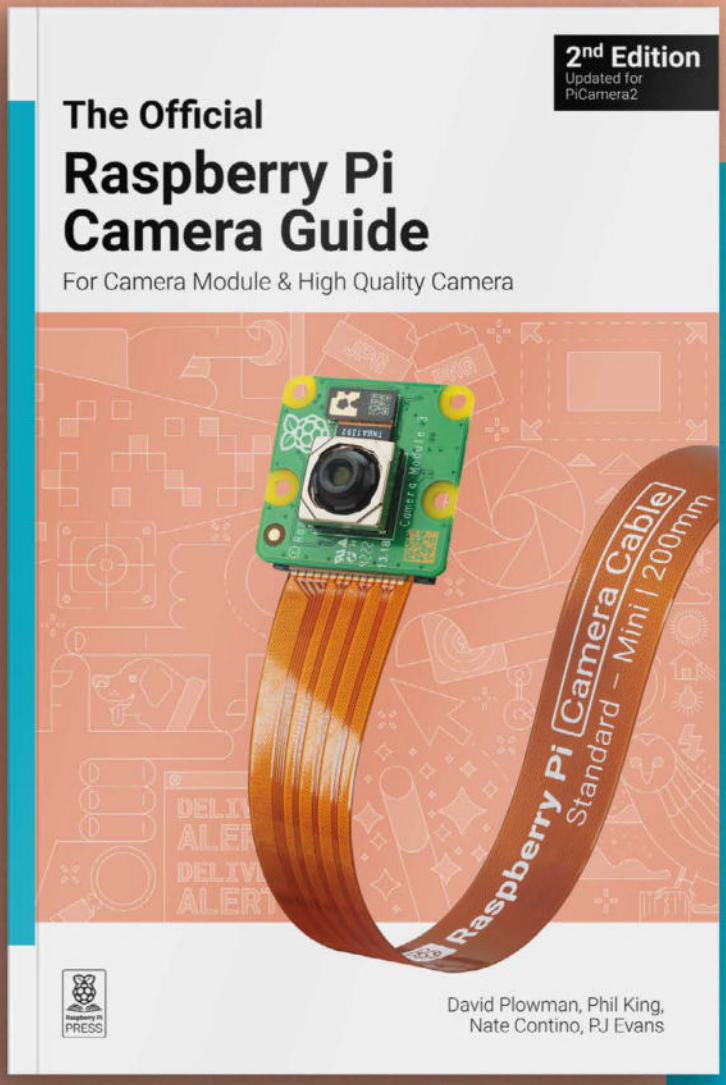
◀ The main board has six 'bays' for adding all sorts of modules

MOTOR GEARING

Various | From £3 / $4 | various

Motors come in various sizes and types, but one thing you need to pay attention to is the gear ratio – equal to the number of 'teeth' on the output gear divided by the number on the input gear (i.e. motor pinion). A higher gear ratio (such as 100:1) means the motor spins much faster than the output shaft, resulting in a lower output speed and higher torque. On the other hand, a lower gear ratio (such as 5:1) results in a higher output speed and lower torque.





Add the power of HDR photography, Full HD video, and AI image recognition to your Raspberry Pi projects with Camera Modules.

- ***Getting started***
- *Capturing photos and videos*
- ***Control the camera with precision***
- *Add artifical intelligence with the AI Kit*
- ***Time-lapse photography***
- *Selfies and stop-motion video*
- ***Build a bird box camera***
- *Live-stream video and stills*
- ***...and much more!***

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

10 amazing:

Art programs

Express yourself on Raspberry Pi with images, audio, and more

Art is incredibly important in the world, and so is self-expression. Whether you're the next Da Vinci or just like to doodle idly, you can easily create your next masterpiece on Raspberry Pi. There's a whole suite of software ready for you to get creative with – here's just some.

01. LibreOffice Draw

Vector graphic illustrator

libreoffice.org

Part of the LibreOffice software suite that is readily available on Raspberry Pi OS, Draw is great for professional illustrations. From simple flow-charts to cool graphic design, it's a powerful tool for any artist.

02. GNU IMP

Photoshop who?

gimp.org

One of the most popular alternatives to Photoshop, this image processor can be used for drawing as well. It's chock-full of features and has plenty of third-party extensions too. This one you may need to install via Pi-Apps.

03. Audacity

Podcast perfection

audacityteam.org

This audio-editing software is a very powerful tool – not just for podcasting but also for music editing, sound effect creation, etc. Phil King used it to edit the samples for his Pico drum machine tutorial this issue.

04. Sonic Pi

Code live music

sonic-pi.net

Using easy-to-understand code, this very special piece of software lets you build music using its programming IDE. You can create music to record as you go, or if you're like creator Sam Aaron, perform live DJ sets with it.

05. RawTherapee

Raw image processing

rawtherapee.com

Photography is art. Advanced photography is very technical art, especially when you start playing with light levels in a raw photo. RawTherapee lets you do this on Raspberry Pi, though.

06. FreeCAD

3D modelling

freecad.org

Want to design up a model for 3D printing? There are some simple online tools we like to use, but FreeCAD is a fantastic bit of software with loads of advanced features.

07. Inkscape

Vector image creation

inkscape.org

A popular piece of software for vector graphics, a little bit similar to Draw, but with more of an artistic flavour than an office one. It's still used by many professionals, although you may need to install it via Pi-Apps.

08. Pixelorama

Retro art

rpimag.co/pixelorama

Pixel art is retro chic, and no longer do you have to paint stuff pixel by pixel, thanks to modern software that lets you emulate the techniques of yesteryear easier than you can say 'PC-98'.

09. Krita

Raspberry Pi canvas

krita.org

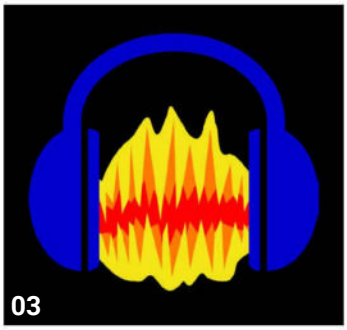
One of the most popular painting programs on any system, this free and open-source piece of software is available on Raspberry Pi via Snap or Pi-Apps. It's very powerful, and can even do 2D animation.

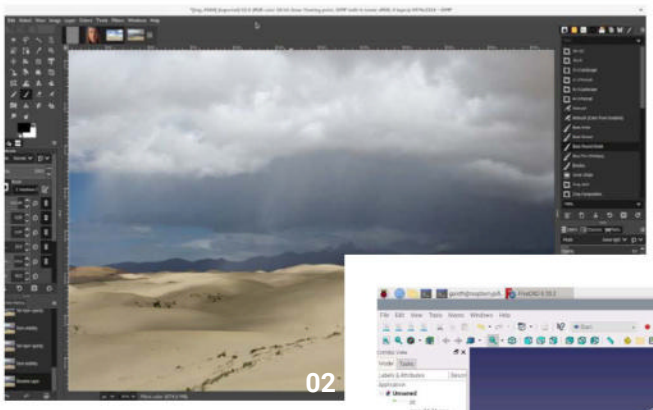
10. ImageMagick

Classic photo manipulation

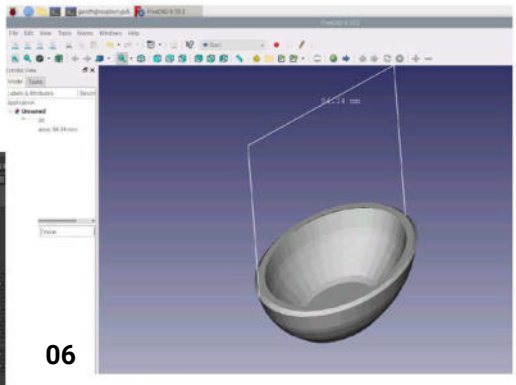
imagemagick.org

This image manipulation suite is a long-running piece of open-source software that is popular among Linux people. It's fantastic for batch processing and automation, keeping it popular to this day.

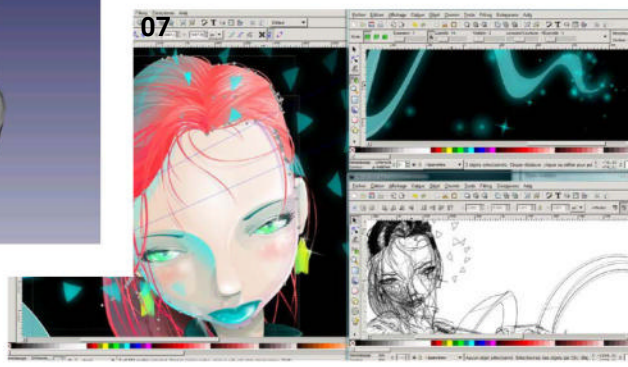

03

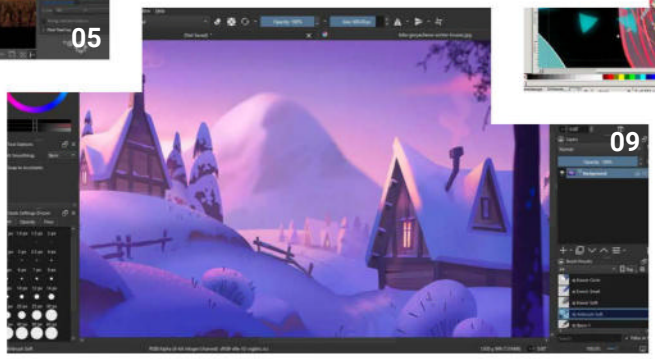

02

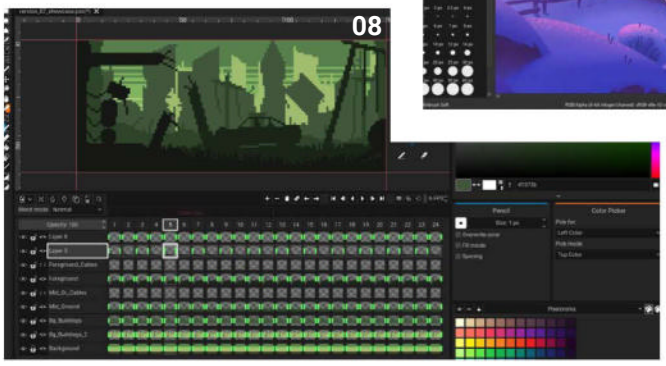
04

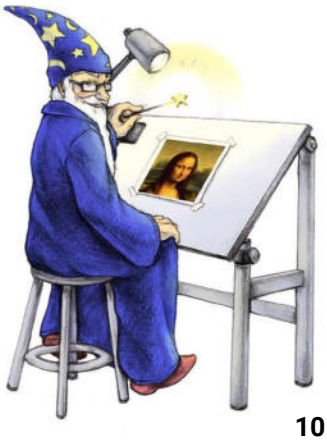

06


05


07

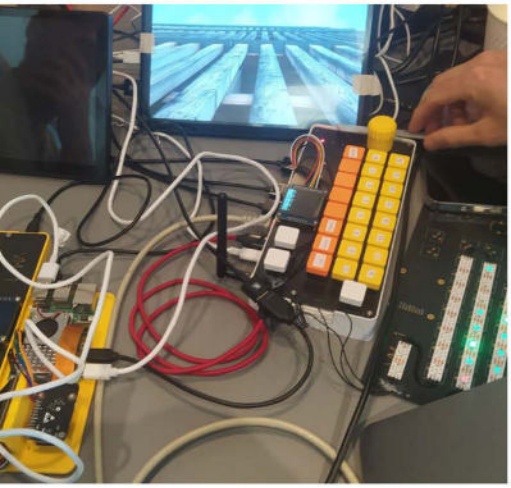

09


08


10

Open Hardware Summit 2025

All the cool new toys from the cutting edge of open source



▲ The main lecture theatre as Open Hardware Summit 2025 opened

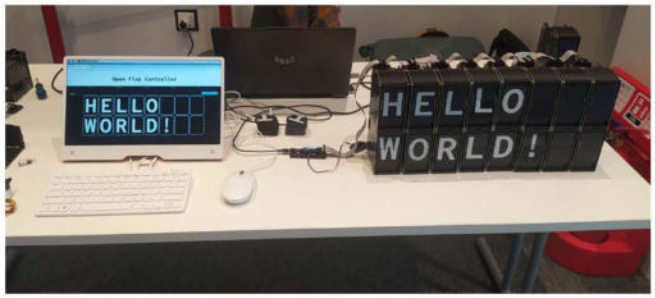
◀ VM-1 is a soon-to-be-released video mixing/playing hardware project built on Raspberry Pi

The Open Hardware Summit (OHS) is a yearly event run by OSHWA, the Open Source Hardware Association. This year, for the first time, it was held in the UK, in Edinburgh. While we love a hi-tech PCB-based conference badge, we also have an ever increasing collection of them in a drawer, so it was a delight to see the conference badge was a machine-embroidered patch with various components you could sew onto it with conductive thread.

OHS was a two-day event, with day one having a fascinating slew of talks and day two being a dual-track affair of themed panels or workshops. Kicking off day one, after opening announcements, was Carson Holgate, who spoke about the fabulous AYAB (rpimag.co/ayab) project which retrofits interfaces to old knitting machines, allowing them to be operated and receive pattern information from a modern software interface. It's a fabulous project, and Carson also took us on a fascinating and insightful potted history of fabric production, knitting machines, and underlying technologies.

Lepidoptera

Almost every talk is worthy of numerous pages – we particularly enjoyed the Mothbox presentation, which is a project using a Raspberry Pi, cameras, and a machine vision system to attract, image, and identify moths! Andy Quitmeyer took us through a fabulous high-energy talk not just on the Mothbox technology but also why bugs and moths are an incredibly important indicator of ecological health and are crucial to monitor.



OpenFlap is an amazing Raspberry Pi-powered project that creates a fabulous high-quality flap-turning display

The OHS badge was a machine-embroidered patch onto which you could sew electronic components!

It's absolutely worth going through all the talks on the OHS2025 YouTube channel (we can't mention every one here), but one other worthy of merit is Gerrit Niezen's talk on 'Growing Food With Electricity', which took us on a whistle-stop tour of the creation of foods/proteins using bioreactors. Gerrit's talk introduced us to Pioreactor, which is a small-form-factor Raspberry Pi-based bioreactor suitable for growing all kinds of cultures.

Aside from the talks, there was a general area where people could showcase their projects, organisations, and or products – either in a formal conference booth, or just placing your project on a table and talking. It was great to see people like Solder Party with their fantastic upcoming range of keyboard and portable systems as well as their giant PCB RP2350 stamp! Another fabulous little section that caught our eye was the wonderful 'Museum of Maker Clichés'. This was a small collection of bell jars each containing a scene depicting a maker cliché, such as 'letting out the magic smoke'. Really fun and thought-provoking, as well as wonderfully made.

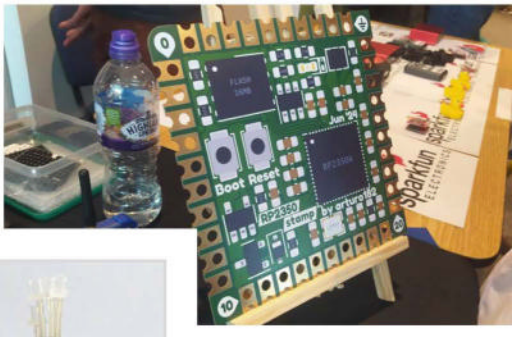
The second day of the Open Hardware Summit had two tracks: the first was a panel track where a few people would speak on a theme and then there was around an hour's discussion and further exploration. Topics included open-source environmental monitoring, open-source hardware for people with disabilities, open-source rocketry, and more.

The second track for day two comprised a number of hands-on workshops. These included a really diverse range of projects, from building a cyclonic vacuum cleaner, to making DIY dye sensitised solar cells, through to 'Touchy Feely Algae' where participants explored plant-based biomaterials.

All the talks and panels from the Open Hardware Summit are already up on YouTube and you can also find many links to information about the projects that were made in the workshops. Of course, as this is all focused around open source, you should be able to replicate the workshop builds at home.

It was a great event that ran smoothly – huge kudos to Lee and Sid and the entire OSHWA crew for getting it together. We look forward to seeing where it is next time, and wherever it is, we urge you to go!

The conference badge was a machine-embroidered patch with various components you could sew onto it with conductive thread



Solder Party had on display lots of their amazing hardware, including this giant version of their RP2350 stamp!

There were many excellent talks and so many fascinating projects; the Pioreactor project certainly caught our attention

Brendan McGrath

As seen on social media as Pater Practicus building wonderful model trains

Name Brendan McGrath
Occupation Graphic designer
Community role YouTuber
URL rpimag.co/paterp

If you follow us on social media, or keep an eye on the #MakerMonday pages in the mag, Pater Practicus – real name Brendan McGrath – will be a familiar sight thanks to his various model train upgrades. He's also a rare maker that didn't grow up coding.

"I've always been quite a practical person, but as a grown-up, inevitably a lot of that energy has been channelled into DIY and doing up our various flats and houses," Brendan explains. "I'm not afraid to tackle most jobs – even if I've never done them before – and over the years I've taught myself skills such as plumbing and electrics, and even how to lay a parquet floor.

"It is only quite recently that I've been 'making' as such. I don't have a background in STEM, coding, or computing (other than programming a ZX81 to randomly generate D&D character stats back in the day), but when I decided to build the boys a gaming computer during lockdown, I realised I could do more than I'd previously thought. Since then, it has been commonplace to have at least one Raspberry Pi or 3D print project on the go at any one time."

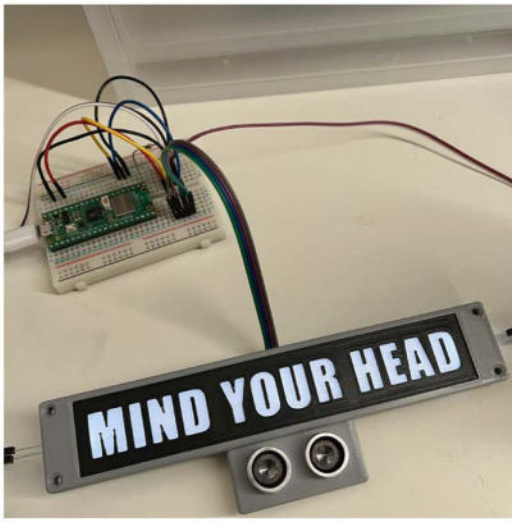
When did you learn about Raspberry Pi?

Our first Raspberry Pi was a 3B, an eleventh birthday present for the boys. They also got a CamJam robotics kit, and together we made 'KV' the robot guinea pig, complete with cardboard-engineered body, and programmed it to do some basic moves.

We didn't really take the Raspberry Pi much further for a while, but I started subscribing to HackSpace and got back into it. I was inspired to revisit KV, improving the chassis build and replacing the Raspberry Pi 3B with a Zero and adding an on/off shim – all of which I filmed for YouTube.

▼ A custom inspection train can squeeze a Raspberry Pi Zero into it

◀ The inspection train in action
▶ This is an essential project for anyone over six feet tall



Wig-Wag lights that flash automatically when a train approaches my level crossing, and have recently completed a scale replica of a Network Rail inspection train – with Raspberry Pi Zero-powered Wi-Fi camera streaming footage of the track ahead, just like the real thing.

I then became intrigued by the possibilities of the Pico for standalone projects... I really liked the fact that I could see my programming do something, even with very simple code – the kind I can get my head around.

> *I really liked the fact that I could see my programming do something, even with very simple code*

Where did your interest in models start?

As part of the Airfix generation, plastic kits and modelling were a big part of my childhood, and I have continued to dabble on and off since. As kids, my brother and I had only had the most basic Tri-ang train set, but when our boys were little, they were given a small set for Christmas, and it rekindled my interest. In the intervening years, technology had really moved on with Digital Command Control (DCC) and I got excited by the possibilities. However, with the cost of new ready-to-run trains now prohibitive I was keen to explore what can be done on a budget with older, second-hand models, adding home-made lights for both locos and coaches – all without spending an arm and a leg.

The crossover of making and modelling is particularly fun. I've made Pico powered

What's your favourite own Raspberry Pi project?

It's almost always the most recent one I'm working on. I'm currently making an ultrasonically triggered 'mind your head' sign for the stairs up to our loft, so that's very much front of mind. Of course, we don't actually need one, but any excuse for a make and to learn something new!

But perhaps the ones I like best of all are those that solve a specific problem, like my Video-call Interruptibility Status Indicator, inspired by the proliferation of Zoom calls during and after lockdown, and that all too common experience of being interrupted by someone who can't see your screen, and doesn't realise you are mid-call.

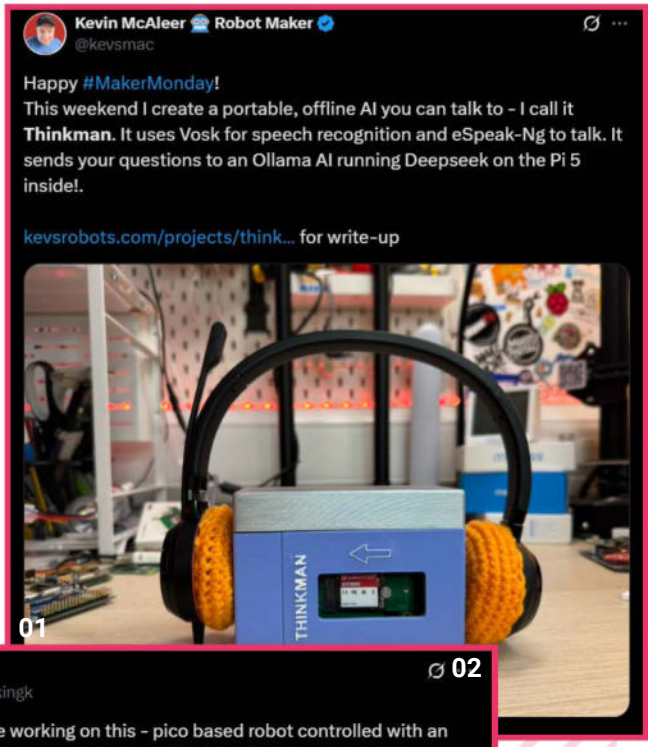
▼ Brendan's kids have helped spark his imagination

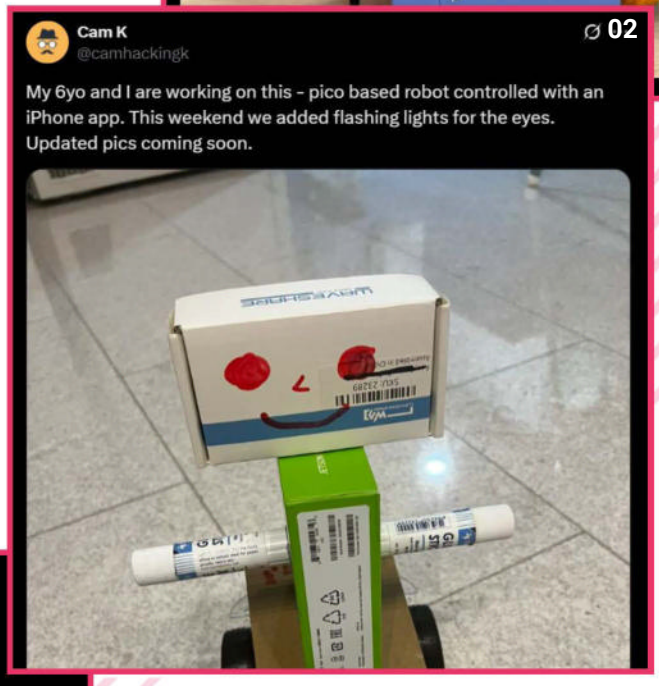
Maker Monday

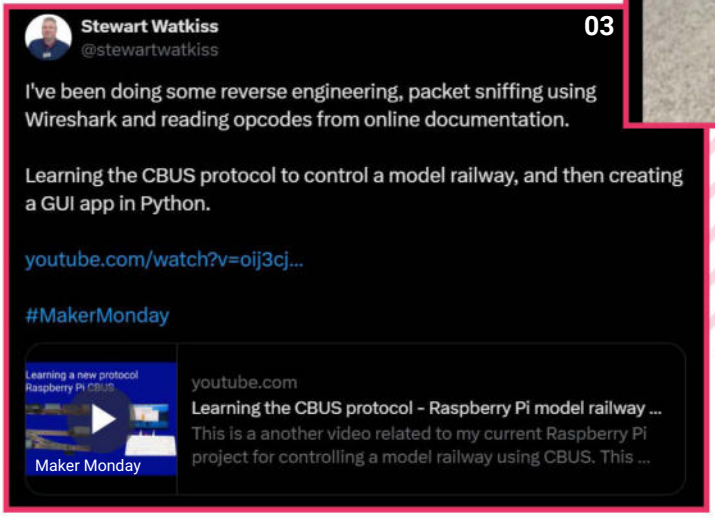
Amazing projects direct from social media!

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!





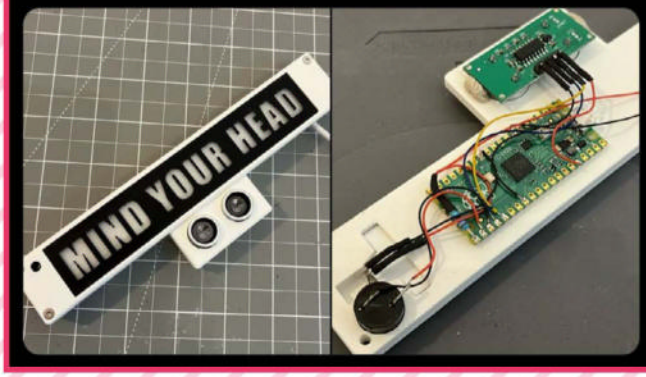


01. This Thinkman is a bit like a Walkman but it will talk to you
02. We love this cardboard robot, fittingly made with tech boxes
03. More deep dives into advanced tech topics with Stewart Watkiss
04. You may recognise this from our interview – Pater (Brendan) is still working diligently on it
05. The Game of Life has rarely looked this good!
06. We now want to make a working giant GBA
07. Yep, Raspberry Pi 4 had the Ethernet port on the other side to the usual B/B+ configuration. It was one of the first things we asked about when we saw it
08. AI -powered robotics are getting more sophisticated

04

Pater Practicus
@PaterPracticus

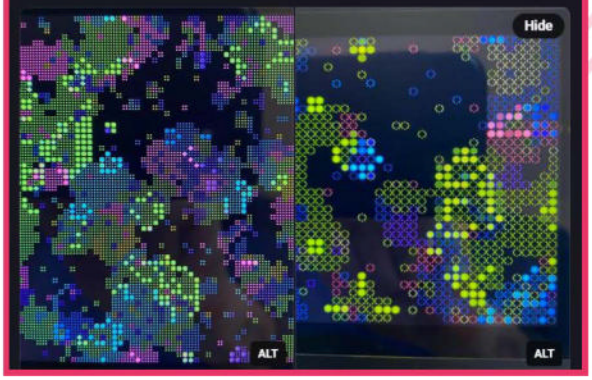
Illuminated 'Mind your head' warning sign all wired up to the Pico inside my 3D printed case. Ultrasonic sensor triggers the sign's LED backlights when someone comes within range, with the buzzer sounding if they get much too close. #MakerMonday



05

Dr Footleg (he/him)
@drfootleg@fosstodon.org

@rpimag After completing my graphics code to draw anti-aliased circles I ported my animations library to the Pimoroni Presto device to run them on the lovely touchscreen, powered by a #RaspberryPi #RT2350 chip as found on the #Pico2. My optimised code only draws anti-aliased pixels when they are against a black background, so my code to clear cells in Game of Life by drawing a black circle is leaving the anti-aliased pixels behind, making a really interesting background effect.
#MakerMonday

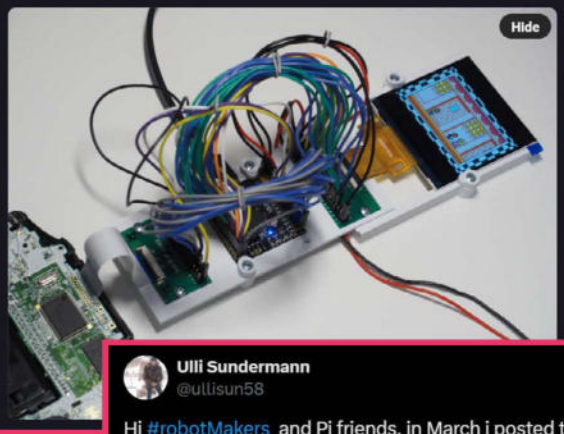


06

DerTobi
@DerTobi@mastodon.social

@rpimag The RP2350 can handle Gameboy Advance display signals ;)

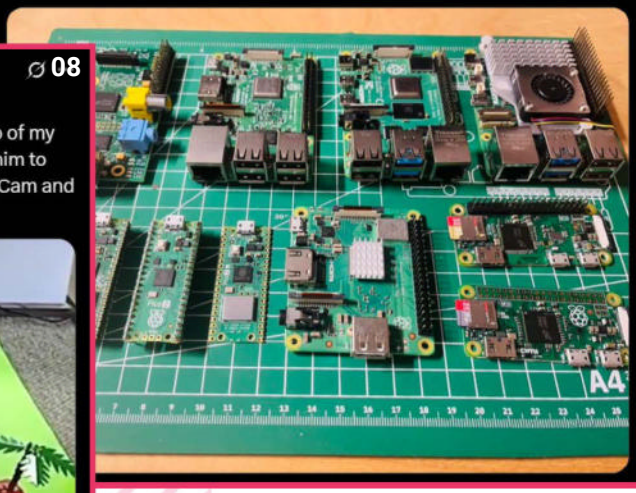
Translate



07

Roland Schulz
@r_schulz_maker

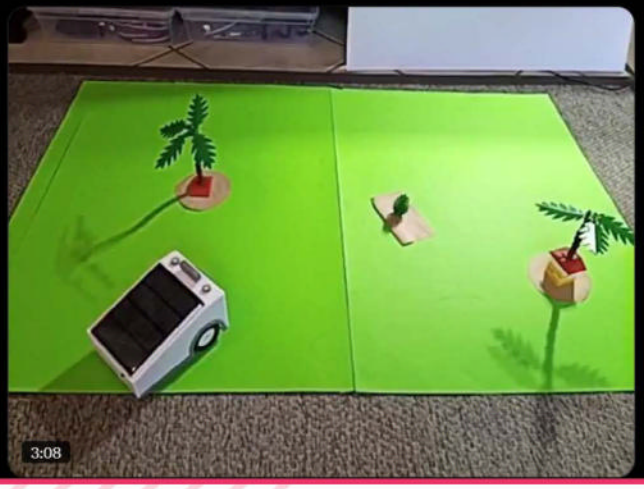
Happy #MakerMonday, I was totally surprised that there is a @Raspberry_Pi family member with an RJ-45 connector on the other side. How can you explain this Pi4? Perhaps, the black sheep of the family 😉!



08

Ulli Sundermann
@ullisun58

Hi #robotMakers and Pi friends. in March i posted the first video of my "oneye" evaluation robot. During the last weeks I "explained" to him to decide where he is not allowed to drive inside is a PI-Zero 2W, PiCam and 1 ToF



Dashcam-pi – a Raspberry Pi car safety camera

A compact camera with some clever code

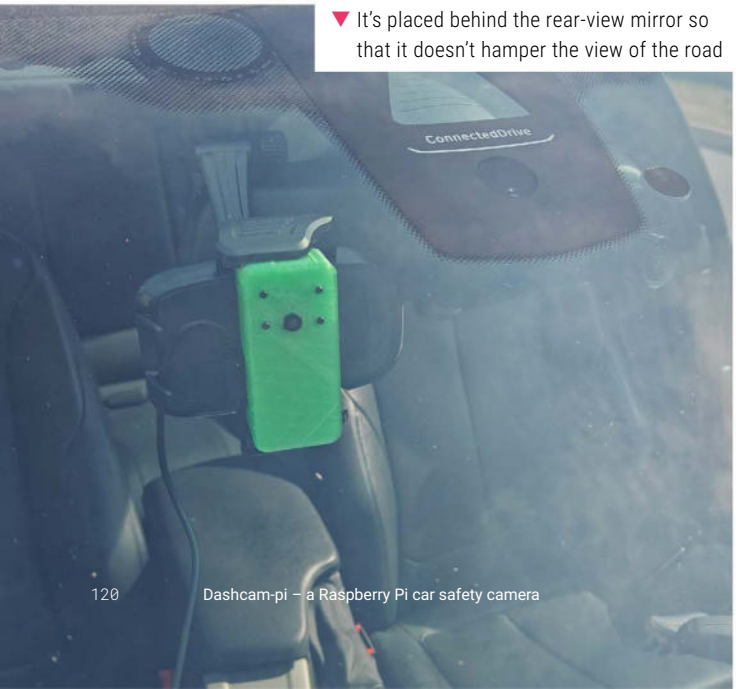
Darko Golner is someone readers may remember from back in issue 143 with his simple GPS tracker made with a Raspberry Pi. He recently emailed us about his new project, dashcam-pi: rpimag.co/dashcampi**.**

"It is designed in a minimalist way using a Raspberry Pi Zero 2 W and a Raspberry Pi Camera Module 3 Wide," Darko says. "The reason for using this combination of a compact Raspberry Pi is to achieve a smaller form factor so the project can fit into a case I designed myself using SketchUp. The choice of the camera is also interesting as I use the Wide version of the Camera Module, which provides a wide-angle video capture similar to commercial car dash products.

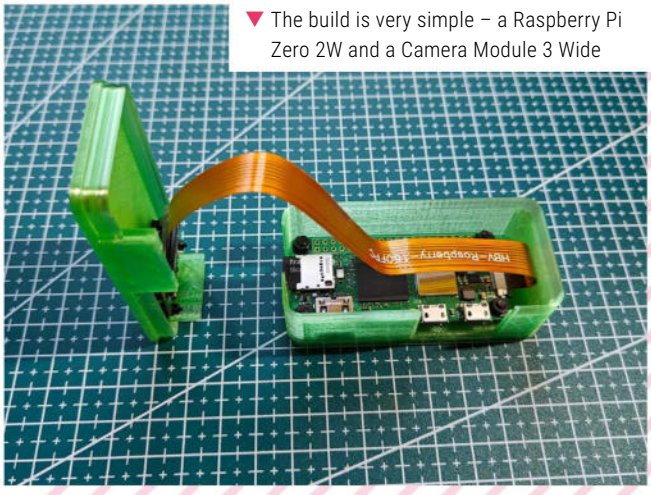
Darko did some fancy code under the hood too, which he was very excited to share.

"Upon startup, Raspberry Pi runs my Python program, which is a web server with an HTML + CSS interface for control," Darko explains. "The software covers functions such as video recording, photo capture, and live camera preview. These functions were quite challenging to implement programmatically for simultaneous use, even with official documentation, but as you can see in the tutorial, I managed to simplify them. The performance of the recordings is comparable to commercial products; this dashcam-pi can record in 1080p at 30fps, though higher performance would require a more powerful Raspberry Pi. The video quality is similar to other Raspberry Pi solutions that use the Camera Module."

Apparently it does require decent lighting to work – but it's a cheap and simple step in the right direction.



▼ It's placed behind the rear-view mirror so that it doesn't hamper the view of the road



▼ The build is very simple – a Raspberry Pi Zero 2W and a Camera Module 3 Wide

Crowdfund this

Great crowdfunding projects this month

VisionForge AR Headset

An AR headset powered by a Raspberry Pi 5 and angled towards education – "making hands-on learning accessible to anyone anywhere". The blurb reckons it will help with electrical repair, construction, pilot training, cooking, and even robotics – although you'll probably need a licensed professional for some of those.

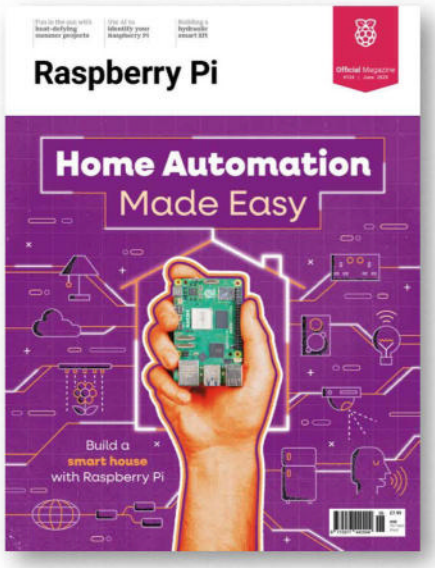
▶ **rpimag.co/visionforge**

CrowPi 3

A new CrowPi from Elecrow is out now on Kickstarter. Designed around Raspberry Pi 5 and AI, it helps folks learn about electronics, programming, and AI from one neat little activity box. There will be over 100 courses available for it, covering over 30 different components.

▶ **rpimag.co/crowpi3**

Your Letters



▲ We explored home automation in issue 154

Home automation made easy

With more time on my hands than I know what to do with (and arthritis that's going to make simple tasks difficult in a few years), I thought I'd try my hand at automating a few bits around the house. The first things I thought I'd do is a device for opening and closing the curtains automatically. I'm working on a system that will close the curtains at a certain time each night, based on the expected sunset and sunrise times for where I live. But I reckon with a bit of thought I could just as easily use a sensor and do it when it's actually dark. After reading the last issue, I know I'm on to something. Oh, and after seeing the word written several times over the last year or so, I now know what Zigbee is!

David, via email

People often mock labour-saving devices, saying that shows laziness, but the truth is that for lots of folks, simple actions such as turning keys in locks and pressing light switches are painful or impossible. Home automation is as much about accessibility as it is about pretending to be Tony Stark. It's something that we should all consider; we're just lucky that as Raspberry Pi enthusiasts we've got an idea of what's possible. But also, we do kind of want to be like Tony Stark just a little bit.

A mixed bag

Thanks for the look at PiEEG last issue. As a big *Ghostbusters* fan as a kid, I was obsessed with the research Dr Peter Venkman was doing at the start of the film, trying to read people's thoughts with the power of the mind. I know it's probably not at that stage yet, but I do like the idea of being able to interface either a machine entirely through brain waves. I just worry that if I can still get things wrong on a command line, a direct brain/computer link is just going to allow me to screw things up even more.

Sarah, via email

There was a lot of variety in last issue: brain waves, the Raspberry Pi classifier that used the Raspberry Pi AI Camera, an open-source insulin pump, and more. The only thing that links everything in the magazine is the ingenuity of the people who made the projects. We all have access to Raspberry Pi computers, Raspberry Pi Pico, and cheap electronics from online auction sites, but the difference it what we do with them. I'm staggered all the time at how creative ordinary people can be, and every day I'm glad that so many people can get their hands on such useful hardware to bring their ideas to life.

▼ Reading brain waves with the PiEEG

Circuits in fabric

Adding a circuit to fabric just looks silly. What are you supposed to do when it rains? And where does the battery go?

Brian, via email

Looking around at our enormous walk-in wardrobe, there are loads of things we don't wear if it looks like it's going to rain. Silk, cashmere, suede boots... and yes, anything with a circuit embedded in it. That doesn't make them useless. A bag with an LED and Raspberry Pi Pico sewn into it might be the most useful thing in the world, and even if it isn't useful, it's pretty damn cool. And as if to prove the point that sewable circuits are here to stay, the badge for this year's Open Hardware Summit (visited by Jo Hinchliffe on page 114) was a conductive fabric patch just waiting for attendees to add their own components. It's the future!

▲ Adding circuits to fabric items is *not* silly!

Contact us!

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

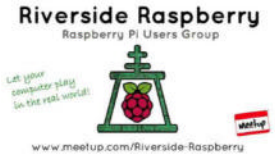
Community Events Calendar

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

01. Melbourne Raspberry Pi Makers Group Meeting

- **Sunday 6 July**
- **Docklands Makerspace and Library, Melbourne, Australia**
- **rpimag.co/mrpmgm155**

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



02. Riverside Raspberry Pi Meetup

- **Monday 14 July**
- **3600 Lime St, Riverside, CA, USA**
- **rpimag.co/rrpm155**

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.

03. Make and Hack : The Medway Makers Meetup

- **Sunday 20 July**
- **Cruden Road, Gravesend, UK**
- **rpimag.co/medway155**

Are you passionate about technology, crafting, and innovation? Do you get a thrill from the whir of servos, the glow of LEDs, and the satisfaction of code running flawlessly? Then Medway Makers have just the thing for you! They are thrilled to announce their next meetup, and you're cordially invited to join a community of like-minded enthusiasts who share your passion for creating and exploring the limitless possibilities of technology.

04. Worcester Linux Users Group

- **Thursday 12 July**
- **Technocopia, Worcester, MA, USA**
- **rpimag.co/wlug155**

One of the longest-running Linux User Groups in Massachusetts. It doesn't matter if you're an old-timer or just new to Linux: if you're interested in all things Linux, from embedded servers to laptops to desktops to supercomputers, you'll have a great time!

Meetings last around two hours and past topics have included everything from issues specific to installing Linux on laptops, to using a Raspberry Pi as a home server, to setting up a ZFS on Linux appliances, to using CAD on Linux and powering your 3D printer using Linux!

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

05. Internet of Things with Raspberry Pi Pico workshop

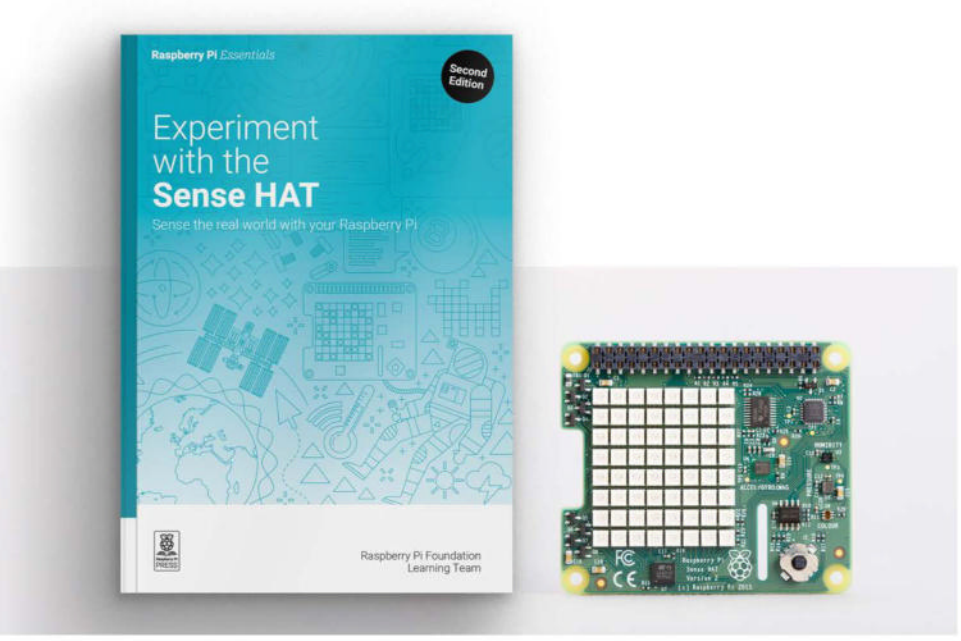
- **Monday 14 July to Thursday 17 July**
- **Kigali, Rwanda**
- **rpimag.co/iotpicokigali**

The Internet of Things with Raspberry Pi Pico is a free workshop taking place in Kigali, Rwanda. Over the course of two days, participants will learn how to use a Raspberry Pi Pico microcontroller to solve real-world challenges using sensors, networking, and cloud technology. This workshop is intended for adult enthusiasts and university students in the Rwanda technology community and it's hosted by GIZ – Digital Transformation Center, Nyereka Tech, and Raspberry Pi.

Official Raspberry Pi Event

Win 1 of 5 Sense HAT V2 bundles

To ce**l**ebrate the pub**l**ication of *Exp**er**im**e**n**t** wi**t**h **t**h**e** S**e**ns**e** HAT (2nd Edi**t**ion)*, we're giving away five copies of the book plus five Raspberry Pi Sense HAT V2s.



Head here to enter:

rpimag.co/win

Learn more:

rpimag.co/SenseHat

Terms & Conditions

Competition opens on **25 June 2025** and closes on **31 July 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.

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

Reach a global community!

Advertise in
Raspberry Pi Official Magazine

Our readers are passionate about technology and the Raspberry Pi ecosystem. From DIY enthusiasts to professional engineers.

Your advertisement can sit alongside our cutting-edge tutorials, features and reviews. And we have flexible advertising options for a range of different businesses.

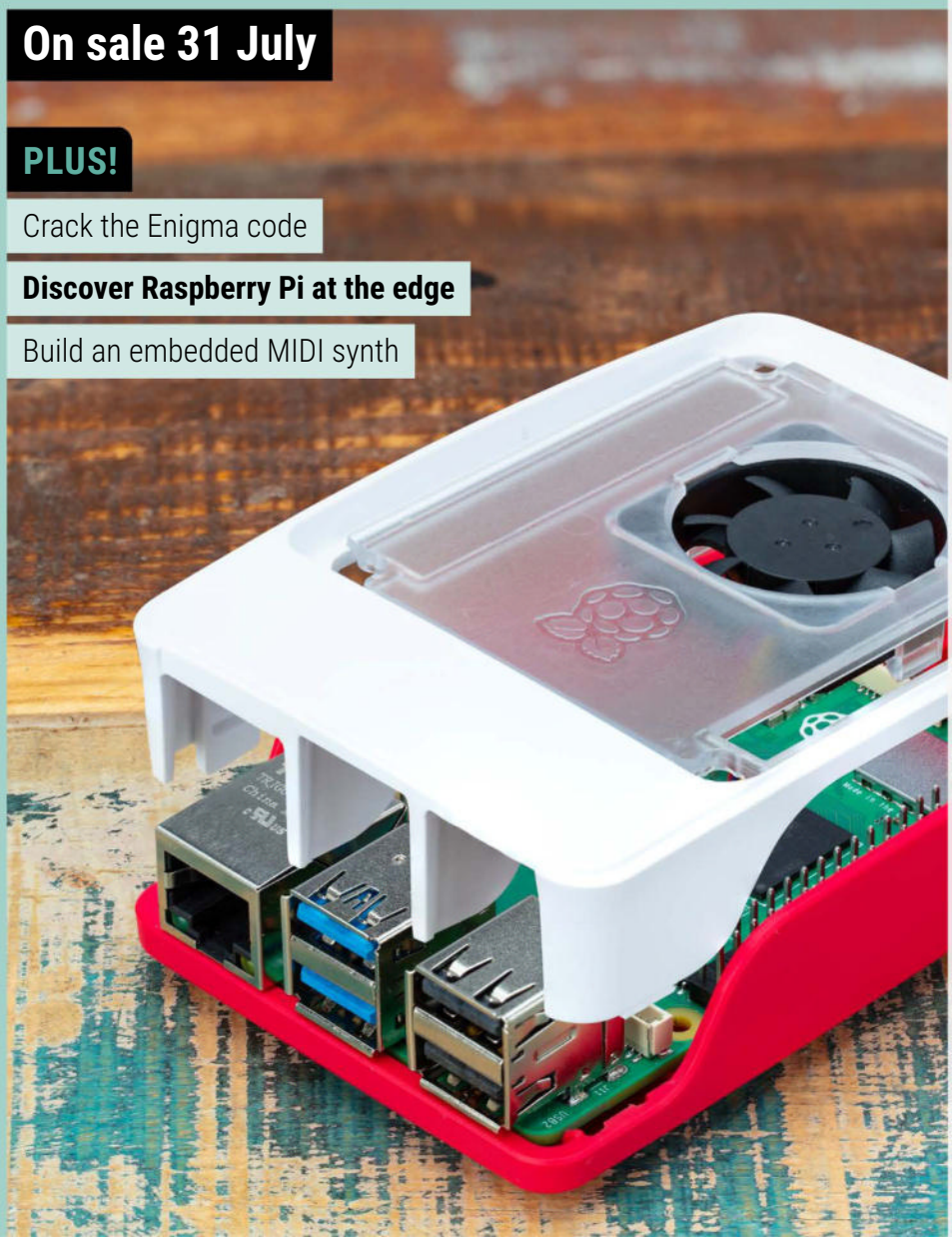
Take the next step –
advertise with us today!

Email Charlie Milligan at: **charlotte.milligan@raspberrypi.com**

Official Magazine
#156

Desktop Computer & Thin Client

Put Raspberry Pi to work in the office



On sale 31 July

PLUS!

Crack the Enigma code

Discover Raspberry Pi at the edge

Build an embedded MIDI synth

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

PJ Evans, Ben Everard, Rosemary Hattersley, Jo Hinchliffe, Nicola King, Phil King, Andrew Lewis, Chris Lowder, Ian Osborne, 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)

From the old comes forth the new

There's nothing wrong with going over old ground, says **Andrew Gregory**

> *Just swapping the computing brains of an old project for something more modern can make the whole thing feel better, faster, more useful*

I'm always struck by the endless creativity that makers have. There's something new everywhere you look, and it's amazing. But there's another side to this, and that's that the old classics keep coming up again and again. Weather stations, greenhouse monitors, time-lapse cameras... they've all been done again and again. But somehow, each one is different. Just swapping out the computing brains of an old project for something more modern can make the whole thing feel better, faster, more useful. The original idea stays in place, the computer gets an upgrade, and suddenly the whole thing feels different.

Even when you don't have new hardware to play with, it can be useful to go over old ground. Like when you re-read a book you loved as a kid and you find new depths in it as an adult, going back to the projects you built years ago will give you a better insight into how you did them. I have a guitar pedal I built years ago and when I look at the solder joints, I cringe inside. It's a wonder the thing ever worked. But it does, and now I know that if I were to do it again, I'd do a better job. Likewise, whenever I look back at old code: I can do far better now, partly because I was so bad at it before.

An old favourite

It's been a while since we've included a magic mirror project in the magazine, because it's been done so many times. But really, how hard would it be to bring a magic mirror into the age of AI? In my house it would only have to recognise two people, so training the model to recognise us both would be fairly simple. I'd get train departure times and updates on the horrendous state of our railways; my wife drives to work, so she'd get updates on the horrendous state of the local traffic, and we'd each get our own calendars.

Come to think of it, it's a brilliant idea, and I'm surprised that no-one has done it before. [Quick Google search occurs.] It's still a brilliant idea, and there is one on the market, but rather than use it to display personalised information based on the user's interests, the makers are using it to display tailored exercise plans. Oh well – I can still go ahead and build mine, even if it's not the groundbreakingly original idea I thought it was 30 seconds ago. Because it really doesn't matter – the point is that if you make it, it's completely your own, whether it's a brand-new project idea or something that's been done thousands of times before.

Andrew Gregory – Features Editor
Andrew is features editor of *Raspberry Pi Official Magazine*, and is currently using a Raspberry Pi Compute Module 5 as his desktop machine.

rpimag.co

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

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

Available at these great Pi stores:

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

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

PiKVM

Remote control redefined

Manage your servers or PCs remotely!

Powered by Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

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

PiKVM V4 Plus

The most feature-rich edition

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

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

Available at the main Raspberry Pi resellers

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

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBII · Cytron Marketplace

ROBU.IN Your ideas, Our Parts · ANTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

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

List of official resellers by country: